TECHNISCHE UNIVERSITÄT MÜNCHEN

MAX-PLANCK-INSTITUT FÜR ASTROPHYSIK

Simulation of low Mach number astrophysical flows

Fabian Miczek

Vollständiger Abdruck der von der Fakultät für Physik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Shawn Bishop, Ph.D.
Prüfer der Dissertation:	
	1. Univ.-Prof. Dr. Friedrich Röpke, Julius-Maximilians-Universität Würzburg
	2. Univ.-Prof. Dr. Björn Garbrecht

Die Dissertation wurde am 20.12.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Physik am 29.04.2013 angenommen.

Contents

Nomenclature

x, y, z	Cartesian coordinates
ξ, η, ζ	curvilinear coordinates
t	time
$\mathbf{n} = (n_x, n_y, n_z)^T$	interface normal vector in global Cartesian coordinates
u, v, w	Cartesian velocity components
$\mathbf{q} = (u, v, w)^T$	Cartesian velocity vector
$q = \sqrt{u^2 + v^2 + w^2}$	absolute value of velocity
$q_n = n_x u + n_y v + n_z w$	velocity normal to an interface
e	internal energy per unit mass
$\epsilon = \rho e$	internal energy per unit volume
$E = e + \frac{1}{2} q^2$	total (internal+kinetic) energy per unit mass
ρ	density
p	pressure
T	temperature
ϑ	potential temperature
μ	mean molecular weight
c	speed of sound
$M = q/c$	Mach number
X	active or passive scalar / mass fraction of chemical abundances
$\mathbf{g} = -\nabla\phi$	gravitational acceleration
ϕ	gravitational potential
$\mathbf{U} = (\rho, \rho u, \rho v, \rho w, \rho E, \rho X)^T$	vector of conservative variables
$\mathbf{V} = (\rho, u, v, w, p, X)$	vector of primitive variables
$\mathbf{F}$	flux vector
R	universal gas constant
$\tilde{R} = R/\mu$	specific gas constant
γ	ratio of specific heats
σ_{sb}	Stefan-Boltzmann constant
K	thermal conductivity
κ_{op}	opacity
c_l	speed of light
V	volume
H_p	pressure scale height
$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{q} \cdot \nabla$	Lagrangian derivative
subscript r, e.g. M_{r}	reference quantity for non-dimensionalization
$F_{\mathrm{r}} = \frac{u_{\mathrm{r}}}{\sqrt{g_{\mathrm{r}} x_{\mathrm{r}}}}$	reference Froude number

1 Introduction: Principles of stellar evolution and open questions

1.1 Physical processes in stars

Stars are known to be self-luminous massive objects. They consist of a huge amount of gaseous matter which is hold together by its own gravitational field. In the following, some basic facts about stellar structure and the governing physical processes are briefly reviewed. More details can for example be found in the books by Weiss et al. (2004), Salaris & Cassisi (2005) or Maeder (2009). Three important parameters to categorize a star are the total mass of the gas, the luminosity (i.e. the radiative energy output per unit time) and the radius. For the latter, it has to be assumed that a star has a spherical shape. From the luminosity and the radius, an effective surface temperature can be derived under the assumption of a black body emission.

For the sun, these parameters are easily accessible by observations. As it is the only star whose surface can be spatially resolved by observations, its radius and shape can be determined with high precision. The oblateness of the sun is only of the order of 10^{-5} such that spherical symmetry is indeed a very good approximation here (e.g. Sofia et al. 1994). As the other parameters can also be measured very accurately, they usually serve as reference quantities for comparisons.

For other stars, the above parameters can at least be derived from observations under some further assumptions. It is seen that the parameters show a huge variability among the observable stars (e.g. Weiss et al. 2004). On the one hand, this may be explained by evolutionary processes in stars which slowly change the stellar parameters. The observed variability would thus represent a sample of different evolutionary stages. On the other hand, stars may have some intrinsic parameters distinguishing them from other stars. As the luminosity and the effective temperature of stars usually do not evolve significantly on short time scales (e.g. a human lifetime), it can be assumed that stars are in some kind of mechanical and energetic equilibrium to a very good approximation. Thus, evolutionary processes may only emerge on much longer time scales and can only be investigated by theoretical models.

The chemical composition of stars can be inferred by spectroscopic observations, but only on the stellar surface. Most stars are composed of hydrogen and helium and a minor fraction of heavier elements at the surface. However, the internal composition and structural properties such as the density and temperature profiles are a priori unknown and can only be obtained by indirect measurements such as astroseismology. They are therefore also subject to theoretical modeling efforts.

Before describing stellar evolution models in more detail, some relevant physical processes in stars are reviewed in the following. To this end, any process that may disturb a certain mechanical or energetic equilibrium within a star is considered along with its typical time scale. The mechanical equilibrium of a star is given by the assumption that the gravitational force is balanced by a non-vanishing radial pressure gradient of the gas (cf. Section 2.3.2). Any flow within a star is expected to be very close to such a *hydrostatic equilibrium* state.

If this equilibrium is globally significantly distorted, the time scale for fluid motions can be estimated by assuming that pressure forces can be neglected. The corresponding *free-fall time scale* is then given by (Weiss et al. 2004)

$$t_{\mathrm{ff}} \approx 2\sqrt{\frac{R^3}{GM}} \approx 1 \cdot \sqrt{\frac{\bar{\rho}_\odot}{\bar{\rho}}} \text{ hours.} \tag{1.1.1}$$

The variables R, G, M and $\bar{\rho}$ denote the stellar radius, the gravitational constant, the total mass and the mean density of a star. The subscript $\odot$ always denotes the corresponding solar values in the following. The free-fall time scale can also be seen as an estimate for the typical time for reinforcing a new hydrostatic equilibrium.

If the hydrostatic equilibrium is distorted only locally, the gravitational force tries to pull the corresponding fluid elements back to their equilibrium position. Due to the presence of inertial forces, this results in oscillatory motions which may build up so-called *internal gravity waves* in the stellar atmosphere. Their typical time scale is given by the (inverse) Brunt-Väisälä-frequency which is typically somewhat larger than the free-fall time scale (cf. Section 6.4). Internal gravity waves may propagate through significant parts of a stellar atmosphere within hours to days, carrying some amount of energy and (angular) momentum with them.

The global energetic equilibrium is governed by the assumption that the energy generation within a star is balanced by the energy which is radiated away from its surface. From theoretical modeling efforts in the last century, it is known that stars are mostly powered by the energy release from nuclear reactions. Since the conditions at the stellar surface are insufficient to trigger such reactions, they have to be confined to regions near the center of the star where the temperatures and densities are expected to be much higher. For example, in the solar core, hydrogen is converted to helium by several nuclear reactions. The emerging neutrino flux produced in these reactions can actually be measured on earth today. In order to get an impression of the time scale on which the nuclear burning may significantly change the composition and structure of a star by hydrogen burning, it is assumed that the corresponding energy release is in equilibrium with the energy that is radiated away at the stellar surface. Estimating that a tenth of the total mass M of a star fulfills the thermodynamic conditions for the nuclear reactions, the *nuclear time scale* can be approximated by (Weiss et al. 2004)

$$t_{\mathrm{nuc}} \approx \frac{1}{10} \cdot \frac{0.007 M c_l^2}{L} \approx 10^{10} \left(\frac{M}{M_\odot} \right) \left(\frac{L_\odot}{L} \right) \text{years}, \tag{1.1.2}$$

where L denotes the surface luminosity of the star and c_l the speed of light. The factor 0.007 is the fraction of mass that is converted to energy during the nuclear reaction. This rough estimate shows that the nuclear energy reservoir of the sun lasts for about 10 billion years. The internal chemical composition should therefore also change on this time scale. It should be noted that nuclear reactions only change the composition and internal energy locally but do not transport the reaction products to different locations within the star.

As the nuclear energy generation is assumed to be in equilibrium with the surface luminosity, the energy has to be transported from the center of a star to the surface. One possibility to achieve this is by thermal radiation. A time scale for this process can easily be estimated by assuming that nuclear reactions are absent and that all the internal energy is successively radiated away at the stellar surface. The so-called *Kelvin-Helmholtz time scale* can then be calculated as (Weiss et al. 2004)

$$t_{\mathrm{KH}} \approx \frac{3GM^2}{4LR} \approx 2 \cdot 10^7 \left(\frac{M}{M_\odot} \right)^2 \left(\frac{L_\odot}{L} \right) \left(\frac{R_\odot}{R} \right) \text{years}. \tag{1.1.3}$$

The variable G denotes the gravitational constant while R denotes the stellar radius. Thus, without an energy source, the sun would radiate away its thermal energy content in about 20 million years. However, the assumption that the luminosity and radius stays constant during this process is not justified. Therefore, the Kelvin-Helmholtz time scale should be regarded as an estimate for the thermal adjustment time which is needed to reinforce a global energetic equilibrium after a temperature excess has been produced somewhere within the star. The relatively long time scale for this process is caused by the fact that stellar material is mostly ionized. Due to the large number of free electrons, the mean free path of a thermal photon is of the order of millimeters (Maeder 2009), which is extremely short compared to the radius of a star.

Energy transport solely by thermal radiation is possible only if the temperature gradient does not exceed a certain critical value. This threshold is given by the temperature gradient of an

adiabatically expanding gas (cf. Section 6.4.1). For an over-adiabatic temperature gradient, the atmosphere becomes unstable to convective motions, where hot plumes of gas rise to some point releasing their energy content somewhere above in the atmosphere. As the gas cools again it sinks downwards to a different position. This results in a cyclic process, which is well-known from everyday life, e.g. boiling water in a pot or convective motions in earth's troposphere. Under suitable stellar conditions, the time scale of a convective turnover time can be estimated to be several days in deep interiors to several minutes in the outer layers of a star (e.g. Weiss et al. 2004). Thus, compared to thermal radiation, convection is a very fast process to transport energy. It should be noted that chemical elements are also redistributed by convective motions on the same time scale. Species generated by nuclear reactions in the core may therefore rise to higher altitude within the star by convective motions. Moreover, convection is an inherently multi-dimensional process and cannot be described directly in a one-dimensional framework assuming spherical symmetry.

Another possibly important process in the description of stars is rotation. For example, the solar surface rotates with a period of approximately 25 days, while the internal rotation structure inferred from helioseismic data shows some considerable variations (Thompson et al. 2003). The rotation of the sun can be considered to be relatively weak, because the overall shape stays nearly spherical as mentioned above. However, different kinds of stars may have much higher rotational velocities such that the spherical symmetry is broken by centrifugal forces (e.g. Beck et al. 2012). Moreover, this may induce further internal transport processes which also take place on the typical time scale of a rotational period (e.g. Maeder & Meynet 2000).

A redistribution of chemical elements may also be caused by atomic diffusion and turbulent motions. The strength of atomic diffusion can be estimated by considering mean free paths of particles and leads to diffusion time scales of the order of 10^{12} years (Maeder 2009). As this exceeds even the nuclear time scale, atomic diffusion is generally assumed to have only minor effects on the structure and evolution of stars. However, as soon as fluid elements within a star start to move, the stellar interior is expected to become very turbulent as the corresponding viscosity is very low. Describing these motions as an effective diffusion process, the corresponding time scales may be estimated to be smaller by factors of 10^5 to 10^8 compared to atomic diffusion (Maeder 2009). A physical process that can trigger such fluid motions is for example shear due to differential rotation (Maeder & Meynet 2000).

Finally, stellar atmospheres permit the propagation of sound waves as they are composed of a compressible gas (cf. Section 2.6.1). Denoting the average speed of sound in a solar-like star by $\bar{c}$, the *acoustic time scale* can be estimated by

$$t_{\mathrm{ac}} = \frac{R_\odot}{\bar{c}} \approx 1 \text{ hour}. \tag{1.1.4}$$

In total, it can be seen that stellar interiors contain a rich variety of physical processes. The corresponding time scales range from hours to billions of years, giving rise to a multi-scale problem for their theoretical investigation. The time scales given above can be ordered as

$$t_{\mathrm{ac}} \lesssim t_{\mathrm{ff}} < t_{\mathrm{rotation}} \ll t_{\mathrm{KH}} \ll t_{\mathrm{nuc}}. \tag{1.1.5}$$

It should be noted that these relations and the definitions of the corresponding time scales only apply for typical evolutionary stages where a star evolves in a relatively quiescent and quasi static way. As these stages usually make up most of the lifetime of a star, the above definitions are justified. However, there are several exceptions where the above ordering and the involved assumptions do not apply. For example, nuclear reactions may become very fast in rare situations, such that the corresponding nuclear time scale reduces even below the free-fall time scale (see Section 1.3).

1.2 Stellar evolution modeling

In order study the evolution of a star, it is desirable to cover its full life-cycle such that effects on measurable parameters like the luminosity or effective temperature become apparent. The governing time scale is thus set by the nuclear time scale which refers to the slowest physical process in the problem. However, it is obvious that the fastest physical processes can not be followed in detail over such long periods of time. This is the case in particular, if a specific stellar model is subject to numerical modeling efforts, as these generally require to resolve the fastest processes in detail. Therefore, it is standard practice to simplify the overall problem such that the nuclear time scale can be resolved. This method, referred to as stellar evolution modeling, is briefly described in the following. More details can, for example, be found in the books by Kippenhahn & Weigert (1994); Weiss et al. (2004); Salaris & Cassisi (2005).

The most simple stellar evolution models use a set of standard assumptions in order to simplify the problem such that the remaining complexity becomes solvable at least with numerical algorithms. It should be noted that in modern, more elaborate stellar evolution models, the standard assumptions are often weakened. Certain physical processes are not neglected anymore, but are treated in an approximate way. Thus, the standard assumptions presented in the following still form a common basis for many stellar evolution models, although specific models may have significant improvements on certain points.

- Stars are isolated objects. Although many observable stars can be found in binary systems, tidal and radiative effects of a companion star are assumed to be small and are thus neglected.
- Stars are spherically symmetric not only in their overall shape but also in their internal structure. This assumption greatly reduces the complexity of the problem by reducing the number of spatial dimensions from three to one.
- Stellar material is composed of gaseous matter and thermal radiation such that a *local* thermodynamic equilibrium may be defined at each point of the star. This assumption basically justifies the definition of unique temperature everywhere.
- The star is always very close to a hydrostatic equilibrium state. Evolutionary processes therefore have to be sufficiently slow.
- Atomic and turbulent diffusion as well as viscosity can be neglected.
- Acoustic phenomena and internal gravity waves can be neglected.
- Chemical elements may only be transported by convection, whereas energy is additionally be transported by thermal radiation.
- Stellar rotation is absent as centrifugal effects are considered to be small. Moreover, a spherically symmetric treatment of a star does not permit rotation anyway, as it would break up this symmetry.

It should be noted that two assumptions are not compliant to each other, namely the spherical symmetry and the possibility of convective heat and element transport as it is an inherently multi-dimensional process. On the one hand, convection is regarded as a very important process for stellar structure and thus cannot be neglected. On the other hand, it could not be resolved temporally in a multi-dimensional setup due to the large discrepancy between the convective and the nuclear time scale. In order to cure this problem, a one-dimensional theoretical model of convective transport has to be established. For stellar evolution modeling, the so-called *mixing length theory* (MLT) is commonly used (Vitense 1953; Böhm-Vitense 1958). It assumes that a rising convective fluid element gets mixed with its surroundings after traveling a fixed vertical

length and deposits its thermal energy content there. The so-called *mixing length* is a free parameter of the model. Despite its phenomenological approach, the mixing length theory may produce qualitatively reasonable results as long as the parameter is well adjusted.

With the above assumptions and MLT, it is possible to derive a rather simple set of partial differential equations describing the evolution of a star as a sequence of hydrostatic models. These equations can be solved numerically assuming that material functions such as a suitable equation of state, opacity functions and nuclear energy generation rates are given. Despite the simplicity of the model, it is very successful as it is able to explain the large variety of observable stars at least qualitatively. It is thus the standard model for stellar evolution.

1.3 Limitations and open questions

Despite the great success of the standard model for stellar evolution, there are also some stellar phases which cannot be treated within this model. Moreover, some effects seen in observations are in contradiction with the model assumptions. Some of the limitations and open questions in stellar evolution modeling and possible solutions are discussed in the following.

The assumption that the nuclear time scale is the slowest is not justified for the full life time of the star. For example, during the late phase of helium burning in stars with $0.7M_\odot \lesssim M \lesssim 2.2M_\odot$, the electrons within the helium core may become partially degenerate, leading to a massive increase of the temperature and the nuclear burning rates (see e.g. Salaris & Cassisi 2005). This process is known as core helium flash. The corresponding nuclear time scale reduces to several days here. The energy produced by the nuclear reactions is mostly carried away by convection, which has a typical turnover time of several hours. As the nuclear burning can not be treated separately from convection anymore, it can be suspected that the convective fluxes predicted by mixing length theory are rather inaccurate in this case. This can be confirmed by multi-dimensional hydrodynamic simulations of the problem (e.g. Mocák et al. 2009). Moreover, it can be seen that stellar evolution calculations performed by different research groups predict fundamentally different results on the outcome of the core helium flash. A similar example, where stellar evolution calculations become very questionable is the silicon burning phase of massive stars (see e.g. Arnett & Meakin 2011). Here, the nuclear time scale may even decrease to hours and the resulting strong convection may not be spherically symmetric on average anymore. Since the results of the silicon burning phase serve as initial conditions for core-collapse supernovae models, a more detailed theoretical investigation is very desirable.

In general, the validity of the mixing length theory can also be questioned for rather quiescent stellar evolutionary phases. According to Kupka (1996) some of the main problems of mixing length theory are the choice of a free parameter (i.e. the mixing length) and the basic assumptions about the turbulent flow field within a convective zone. In fact, the mixing length parameter may have a severe impact on the results of a stellar evolution calculation and is sometimes even be used to compensate different errors in order to get results which are more compliant to observations. Choosing the mixing length on physical grounds is hardly possible. Moreover, due to the low viscosity in stars, the convective flow field is expected to be highly turbulent. Mixing length theory contains some assumptions about turbulence which can not be justified for stellar atmospheres. In order to cure the problems of mixing length theory, several improvements have been developed in the past (see e.g. Spruit et al. 1990; Canuto 1996; Kupka 1996). However, these models either involve even more free parameters which are a priori unknown or they are hardly applicable to all kinds convection zones which might occur. Mixing length theory is, despite its deficiencies, still the most common method to model stellar convection. More recent hydrodynamic simulations of convection in deep stellar interiors which were compared to predictions of mixing length theory have actually shown that the mean energy flux is quite compatible (Meakin & Arnett 2007). However, the position of the convective boundaries where the energy flux becomes dominated by radiation again seems to be hardly predictable with mixing length theory. Moreover, additional mixing processes of chemical elements at these boundaries (e.g. by convective overshooting)

cannot be resolved by mixing length theory. Since convective boundaries may have a substantial impact on stellar evolution calculations, further improvement of convection models may become vital in stellar evolution research.

Internal gravity waves may also be excited at convective boundaries by overshooting processes (see e.g. Young et al. 2003; Talon 2008, and references in there). These waves may propagate through radiative zones within a star until they break by non-linear interactions, leading to small-scale turbulence in these regions. Apart from being an important mechanism for angular momentum transport, internal gravity waves may constitute an additional source of mixing of chemical elements in radiative regions. Such processes can only be modeled in stellar evolution calculations by additional turbulent diffusivities in radiative regions. However, to date there is no self-consistent method to estimate the strength of such a process. Comparing the solar rotation profile measured by helio-seismology to stellar evolution calculations suggests that internal gravity waves may indeed have important implications on stellar structure (Talon 2008). More sophisticated models are thus required to incorporate the additional physics.

Stellar evolution modeling becomes much more involved when rotation is considered. First of all, rotation breaks up the spherical symmetry of stars due to the effect of centrifugal forces, violating one of the basic assumptions of standard stellar evolution models. A one-dimensional treatment of the overall problem is then still possible if shellular rotation is assumed (Maeder 2009). However, the description still becomes very complex because a differentially rotating stellar interior is subject to various flow instabilities such as meridional circulation, the Solberg-Hoiland instability, the GSF instability (Goldreich & Schubert 1967; Fricke 1968) and dynamical and secular shear instabilities (see Maeder & Meynet 2000, for an overview). The latter are discussed in more detail in Chapter 7. These instabilities may lead to turbulence and may thus be an additional source of mixing. Moreover, the global angular momentum transport crucially depends on these processes and should be investigated in a self-consistent way. For stellar evolution modeling, the situation is very similar to internal gravity waves. The additional physics can only be modeled by additional turbulent diffusivities whose effective properties are rather unknown. As stars, especially in the high-mass region, are suspected to develop fast rotating cores in the late phases of their evolution, an accurate treatment of rotation is of great importance (Maeder & Meynet 2000).

In summary, these examples show that the standard stellar evolution model still has major shortcomings that might have major effects the results in some cases. It should be noted that the above presentation not complete. Additional physical processes like magnetic fields or radiation in optically thin regions might further increase the complexity of the overall problem, but are not discussed here. Moreover, some physical phenomena in stars, for example the Urca process or stellar pulsations, are basically known but can not be treated accurately in stellar evolution models yet. In total, it can be stated that stellar evolution models, although qualitatively extremely successful so far, need further improvement in order to give a comprehensive explanation of the huge variety of observable stars on a quantitive level.

1.4 Hydrodynamic treatment of stars

One potential way to improve stellar evolution models is by detailed hydrodynamic numerical simulations on a computer system. Ideally, they should capture the temporal evolution of a star incorporating all relevant physical processes from first principles on a three-dimensional domain. This also means that the fastest process should be followed in detail, which is the propagation of sound waves in this case. However, it is obvious that sound waves cannot be followed for the entire life-cycle of a star. The ratio of the nuclear time scale to the acoustic time scale actually spans about 14 orders of magnitude resulting in an extreme multi-scale problem (cf. Section 1.1). Even with a tremendous increase in computing power, such problems would be far out of reach for numerical simulations. Therefore, the detailed hydrodynamic treatment of a star has to be limited to relatively short time scales compared to the stellar life time. Nonetheless,

such simulations may still be very useful. First, they could capture very violent phases where stellar evolution models fail due to their simplifications. Examples include the aforementioned core helium flash or the silicon burning phase which only last for several days, but may have significant implications for the further evolution of the star. Second, hydrodynamic simulations may be used to investigate physical processes in more detail which can only be treated in a parametrized way in stellar evolution models. This includes convective zones in stars with turnover times suitable for a hydrodynamic treatment. The results can then be compared to the predictions of mixing length theory in order to test its validity in certain situations. Moreover, the big question of the origins of turbulent flow patterns in radiative regions in stars may be addressed by hydrodynamic calculations. Here, it might even be sufficient to simulate small parts of a star on the length scale where a certain phenomenon is expected. In a first step, suitable stellar conditions for turbulence producing processes may be explored. In a second step, effective turbulent diffusivities might be derived from simulations and compared to existing simplified model predictions. Finally, these numerical calculations may be even used to develop new models for a certain physical process which can then be incorporated into stellar evolution models.

It was seen in the last section that the most important prospects for a hydrodynamic treatment of stars are related to convection or turbulent fluid motions. The time scale for both processes is thus determined by the speed q of fluid elements. However, the governing equations (see Chapter 2) also permit sound waves which propagate at a different velocity c. The ratio of both speeds, $M = q/c$, is called *Mach number*. For typical situations in stellar interiors the Mach number is very low. This can be seen by evaluating the speed of sound for an ideal gas, which is solely determined by the temperature T of the gas:

$$c = \sqrt{\gamma \tilde{R} T} \tag{1.4.1}$$

The parameter γ denotes the ratio of specific heats while $\tilde{R}$ refers to the specific gas constant. As stellar interiors generally become hotter with increasing depth, the speed of sound also increases. Thus, it can be expected that the Mach numbers tend to decrease towards the center of a star. For example, the convective speeds predicted by mixing length theory have corresponding Mach numbers of $\approx 10^{-6}$ in deep interiors and may reach $0.1 - 1$ near the stellar surface. Hence, fluid motions in stars are generally low Mach number flows.

In order to perform hydrodynamic simulations of stellar interiors, two approaches have commonly been employed in the past. The first one uses simplified sets of equations, such as the Boussinesq approximation or the anelastic equation, to describe the flow. Although these approaches are generally well-suited for low Mach number flows, it is argued in Section 2.7 that the involved simplifications are too severe in order to simulate complex stellar interiors. For the second approach, the fully compressible Euler equations (see Section 2.1) are simulated with numerical codes, which were originally developed for high Mach number flows. As the temporal discretization is usually explicit in this case, the computational time step is limited by numerical stability requirements such that the fastest physical process, i.e. the propagation of sound waves, is well resolved in time. As this limitation is also valid in the low Mach number regime, the computational efficiency becomes extremely low. Here, it takes of the order of $1/M$ time steps, just to move a fluid element from one grid cell to another. Hence, this approach can only be used for moderately low Mach numbers (i.e. $> 10^{-2}$). Simulations of convection in deep stellar interiors is therefore far out of reach with this approach. Moreover, it can be questioned if the employed spatial discretization techniques, which were developed to capture strong high Mach number shock waves are still accurate in the low Mach number regime.

1.5 Goals and organization of this thesis

The aim of this work is to develop a new general purpose simulation code which is especially suitable for stellar interiors involving low Mach number flows. In order to keep the applicability of the code as large as possible, the simulation is also governed by the compressible Euler equations, which generally describe flows of all Mach numbers accurately. In order to maintain computational efficiency in the low Mach number regime, one main goal of this work is to employ a fully-implicit temporal discretization (see Section 5.2). With such a technique, the computational time step is not limited by the fast sound waves anymore and may be chosen according to the physical process of interest, e.g. low Mach number convection. Thus, implicit time steppers may potentially be a factor $1/M$ more efficient than an explicit temporal discretization. However, implicit time steps are computationally much more expensive than their explicit counterpart. In each implicit time step, several systems of non-linear equations have to be solved numerically. While this can easily be done for very small systems, it has often been questioned whether this approach is feasible for discretizations of the compressible Euler equations, potentially involving up to several hundreds million degrees of freedom. Thus, the solution of the huge non-linear systems of equations is critical in order to keep the implicit method computationally efficient compared to explicit time steppers. It is shown in Chapter 5, that this goal can indeed be reached with a suitable combination of modern numerical techniques. Moreover, the resulting code can even be employed on parallel computer systems with distributed memory architectures efficiently (see Section 7.5).

A second goal of this work is the systematic evaluation and improvement of the numerical accuracy of low Mach number flows in stellar interiors. It is shown in Chapter 4 that standard discretization techniques (developed for high Mach number flows) become very inaccurate in the low Mach number regime, as the numerical viscosity becomes exceedingly high. In order to cure this problem, the behavior of several low Mach number discretization techniques is analyzed in detail. Moreover, new discretization methods are proposed here, which are particularly well-suited for the simulation of stellar interiors. Besides that, special emphasis is put on the discretization of gravity. As low Mach number flows in stellar interiors can be expected to be nearly hydrostatic, it is of uttermost importance that the involved physical processes are treated with high accuracy. Due to aforementioned improvements on the computational efficiency and accuracy, it became possible to examine nearly hydrostatic flows in more detail. However, it is shown in this work that standard discretization techniques for gravity also need some improvement in order to capture the gravitational dynamics in stellar interiors accurately.

This work is organized as follows. Chapter 2 describes the physics and the governing equations which are needed in order to described stellar interiors with a hydrodynamic treatment. Moreover, the asymptotic behavior of the equations in the low Mach number regime is discussed in detail. Chapter 3 describes standard discretization techniques which form the basis of the new hydrodynamic simulation code. Special emphasis is put here on a discrete formulation in curvilinear coordinates. In Chapter 4, numerical flux functions are analyzed systematically in the low Mach number regime. The reason for the failure of the standard discretization techniques in the low Mach number regime is identified and several alternatives are presented and compared. Implicit temporal discretization methods are described in Chapter 5. Several numerical techniques are presented in order to solve the arising non-linear systems of equations efficiently. The computational performance and accuracy of the overall method is confirmed by detailed numerical tests. Chapter 6 is devoted to the discretization of gravity and nearly hydrostatic flows. The arising numerical challenges are assessed by theoretical arguments and numerical tests. Moreover, the behavior of internal gravity waves in numerical simulations is evaluated and their importance for stellar interiors is emphasized. In Chapter 7, a first application of the new simulation code is presented. Shear instabilities in stellar interiors are first analyzed by a semi-analytical model. It is then shown that numerical simulations with the new code are feasible in an accurate and efficient way. Moreover, the prospects of the new tool for the improvement of stellar evolution models are outlined.

In total, this work can be seen as a first step towards more efficient simulations of low Mach

number flows in stellar interior such that convective and turbulent fluid motions can be accurately resolved. For stellar evolution models, it would be very desirable to reach time scales in hydrodynamic simulations where the flow becomes significantly affected by radiative processes. Although beyond the scope of the efforts presented here, this work provides a significant step towards this goal.

2 Governing equations

This chapter describes the basic equations for a hydrodynamic treatment of stellar interiors. This will be done to a level of detail as needed for their discretization in the new simulation code. Moreover, some theoretical properties concerning low Mach number flows are derived.

2.1 Euler equations

Throughout this thesis, the fluid is described in the continuum approximation, neglecting the individual behavior of the particles of the fluid. This approach is well justified in case that all fluid elements of interest are much larger than the mean free path of the particles. For typical situations of gas flows within stars, this is requirement is well fulfilled. Moreover, it is assumed that the fluid is inviscid. Due to technical reasons the discretized equations contain some *numerical* viscosity, as will be shown later. The *physical* viscosity is neglected because it is typically orders of magnitude smaller within stars.

With these assumptions, the fluid can be described mathematically by its velocity field and some quantities that fully represent the thermodynamic state of the fluid. All fields are continuous functions of space and time whose temporal evolution is governed by the compressible Euler equations. They may be written as a set of conservation laws,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} + \frac{\partial \mathbf{F}_z}{\partial z} = \mathbf{S} \tag{2.1.1}$$

where the vector of conservative variables,

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho X \end{pmatrix}, \tag{2.1.2}$$

contains the density ρ, the Cartesian velocity components u, v, w, the total (internal+kinetic) energy E of the fluid and optionally some scalar values X. The corresponding flux vectors,

$$\mathbf{F}_x = \begin{pmatrix} \rho u \\ \rho uu + p \\ \rho uv \\ \rho uw \\ \rho uE + up \\ \rho uX \end{pmatrix} \quad \mathbf{F}_y = \begin{pmatrix} \rho v \\ \rho vu \\ \rho vv + p \\ \rho vw \\ \rho vE + vp \\ \rho vX \end{pmatrix} \quad \mathbf{F}_z = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho ww + p \\ \rho wE + wp \\ \rho wX \end{pmatrix}, \tag{2.1.3}$$

contain the pressure p that is related to the conservative variables by an equation of state. The vector $\mathbf{S}$ may contain source terms such as gravity, nuclear reactions or thermal radiation. In the absence of any source terms, the equations are called homogeneous Euler equations. They are said to be in strong conservation form, since they express the conservation of mass, momenta and total energy. This can easily be seen by integrating the equations over an arbitrary domain and applying Gauss' theorem. Then, the integral over the conservative variables may only change in time by fluxes through the boundaries of the domain.

It should be noted that the scalar field X, as written in the above equation is only exemplary for any number of scalar fields, which may be needed. For example, the fluid may consist of n chemical species, whose individual mass fractions X_i $(i = 1..n)$ should be distinguished. Then, the Euler equations are augmented by n scalar fields, whose temporal evolution is similar to the one shown above. Therefore X may always represent a set of scalar fields.

The Euler equations as presented above are valid on a three-dimensional domain. However, they can easily be reduced to one or two spatial dimensions by removing the dispensable momentum equations and setting the corresponding velocities and fluxes to zero. For example, the one-dimensional Euler equations are recovered by setting

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho E \\ \rho X \end{pmatrix} \qquad \mathbf{F}_x = \begin{pmatrix} \rho u \\ \rho u u + p \\ \rho u E + u p \\ \rho u X \end{pmatrix} \qquad \mathbf{F}_y = \mathbf{F}_z = 0. \tag{2.1.4}$$

2.2 Equations of state

2.2.1 General description

In order to close the Euler equations, an equation of state is needed that relates the pressure of the fluid to the conservative variables. Since the discretization of the Euler equations should not depend on any specific equation of state, its detailed form is not specified at this point. Instead of that, a general interface is described in the following, which can be used by an arbitrary equation of state.

For the discretization, it is assumed that a specific equation of state provides the pressure p and its partial derivatives as a function of density ρ, internal energy per unit volume ϵ and optionally any number of scalars X, i.e.

$$p(\rho, \epsilon, X) \qquad \left.\frac{\partial p}{\partial \rho}\right|_{\epsilon,X} \qquad \left.\frac{\partial p}{\partial \epsilon}\right|_{\rho,X} \qquad \left.\frac{\partial p}{\partial X}\right|_{\rho,\epsilon}. \tag{2.2.1}$$

It should be noted that any other combination of independent variables (e.g. ρ, T, X) for this function would be equally reasonable. The specific choice made here has the advantage that the pressure can easily be expressed by the conservative variables:

$$p = p\left(\mathbf{U}_\rho, \mathbf{U}_{\rho E} - \frac{1}{2\mathbf{U}_\rho}\left(\mathbf{U}_{\rho u}^2 + \mathbf{U}_{\rho v}^2 + \mathbf{U}_{\rho w}^2\right), \frac{\mathbf{U}_{\rho X}}{\mathbf{U}_\rho}\right) \tag{2.2.2}$$

Therefore the partial derivatives of the pressure with respect to the conservative variables can also be computed very easily by the chain rule.

With the knowledge of the pressure and its derivatives, it is possible to compute the speed of sound, which is needed for many calculations. For a general fluid, it is defined as the square root of a pressure change with density in an isentropic process (e.g. Weiss et al. 2004):

$$c = \sqrt{\left.\frac{\partial p}{\partial \rho}\right|_S} \tag{2.2.3}$$

This partial derivative can be expanded by the chain rule with the known derivatives:

$$\left.\frac{\partial p}{\partial \rho}\right|_S = \left.\frac{\partial p}{\partial \rho}\right|_{\epsilon,X} + \left.\frac{\partial p}{\partial \epsilon}\right|_{\rho,X} \cdot \left.\frac{\partial \epsilon}{\partial \rho}\right|_{S,X} \tag{2.2.4}$$

The last derivative can further be calculated with the first law of thermodynamics under the assumption of constant mass and composition of the fluid element:

$$\begin{aligned} T dS &= dU + p\, dV \\ &= V d\epsilon + (p + \epsilon) dV \\ &= V d\epsilon - \tfrac{p+\epsilon}{\rho}\, V\, d\rho \end{aligned} \tag{2.2.5}$$

The final expression for the speed of sound therefore results in

$$c = \sqrt{\left.\frac{\partial p}{\partial \rho}\right|_{\epsilon,X} + \left.\frac{\partial p}{\partial \epsilon}\right|_{\rho,X} \cdot \frac{\epsilon + p}{\rho}}. \tag{2.2.6}$$

Apart from pressure, an equation of state has to provide also the temperature T of the fluid. For the Euler equations as described in the last section, this would not be necessary since they do not contain T explicitly. However, certain boundary conditions or source terms like thermal diffusion may have such an explicit temperature dependence. In these cases, information of the temperature derivatives may also be needed. In most cases, it is more convenient to compute the derivatives of the inverse function $\epsilon = \epsilon(\rho, T, X)$ which contains the same amount of information. To summarize, the following expressions have to be provided by the equation of state for temperature depended processes:

$$T(\rho, \epsilon, X) \qquad \left.\frac{\partial \epsilon}{\partial \rho}\right|_{T,X} \qquad \left.\frac{\partial \epsilon}{\partial T}\right|_{\rho,X} \qquad \left.\frac{\partial \epsilon}{\partial X}\right|_{\rho,T} \tag{2.2.7}$$

Equation 2.2.1 and 2.2.7 form the interface for the equation of state. It was chosen such that further calculations needed in this thesis take a rather simple form. However, a disadvantage of this interface is, that many equations of state are naturally given with different independent variables, e.g.

$$p = p(\rho, T, X) \qquad \epsilon = \epsilon(\rho, T, X). \tag{2.2.8}$$

In general, their inverse functions can not be written down analytically. Given the conservative variables, it may therefore be necessary to iterate on pressure and temperature with a numerical algorithm (e.g. a Newton-Raphson method). If the equation of state provides the partial derivatives corresponding to Equation 2.2.8 only, they have to be transformed to the interface given by Equation 2.2.1. Therefore, the pressure is written as

$$p\left(\rho, T\left(\rho, \epsilon, X\right), X\right) \tag{2.2.9}$$

whose partial derivatives with respect to ρ, ϵ, X can be obtained by the chain rule:

$$\begin{array}{lcll} \left.\frac{\partial p}{\partial \rho}\right|_{\epsilon,X} & = & \left.\frac{\partial p}{\partial \rho}\right|_{T,X} & + \left.\frac{\partial p}{\partial T}\right|_{\rho,X} \cdot \left.\frac{\partial T}{\partial \rho}\right|_{\epsilon,X} \\ \left.\frac{\partial p}{\partial \epsilon}\right|_{\rho,X} & = & & \left.\frac{\partial p}{\partial T}\right|_{\rho,X} \cdot \left.\frac{\partial T}{\partial \epsilon}\right|_{\rho,X} \\ \left.\frac{\partial p}{\partial X}\right|_{\rho,\epsilon} & = & \left.\frac{\partial p}{\partial X}\right|_{\rho,T} & + \left.\frac{\partial p}{\partial T}\right|_{\rho,X} \cdot \left.\frac{\partial T}{\partial X}\right|_{\rho,\epsilon} \end{array} \tag{2.2.10}$$

The partial derivatives of the temperature can be calculated by applying the implicit function theorem:

$$\begin{array}{lcl} \left.\frac{\partial T}{\partial \rho}\right|_{\epsilon,X} & = - & \left(\left.\frac{\partial \epsilon}{\partial T}\right|_{\rho,X}\right)^{-1} \cdot \left.\frac{\partial \epsilon}{\partial \rho}\right|_{T,X} \\ \left.\frac{\partial T}{\partial \epsilon}\right|_{\rho,X} & = & \left(\left.\frac{\partial \epsilon}{\partial T}\right|_{\rho,X}\right)^{-1} \\ \left.\frac{\partial T}{\partial X}\right|_{\rho,\epsilon} & = - & \left(\left.\frac{\partial \epsilon}{\partial T}\right|_{\rho,X}\right)^{-1} \cdot \left.\frac{\partial \epsilon}{\partial X}\right|_{\rho,T} \end{array} \tag{2.2.11}$$

With this procedure, equations of state in the form (2.2.8) can easily be used with the interface described in this thesis. Similar procedures can be applied if the equation of state is given in arbitrary other variables. Moreover, it should be emphasized that no assumption about the analytic form of the equation of state is made. Therefore, even tabulated equations of state may be used as long as the values required by the interface can be computed numerically.

2.2.2 Ideal gas

The most simple equation of state is that of an ideal gas. The pressure can be related directly to the internal energy via

$$p = (\gamma - 1)\,\epsilon. \tag{2.2.12}$$

The ratio of specific heats γ is assumed to be 5/3, corresponding to a monoatomic gas, if not stated differently. The partial derivatives of this equation of state read

$$\left.\frac{\partial p}{\partial \rho}\right|_{\epsilon,X} = 0 \qquad \left.\frac{\partial p}{\partial \epsilon}\right|_{\rho,X} = \gamma - 1 \qquad \left.\frac{\partial p}{\partial X}\right|_{\rho,\epsilon} = 0. \tag{2.2.13}$$

In order to calculate the temperature of the ideal gas, the following relation can be used:

$$p = \rho R T/\mu \tag{2.2.14}$$

Here, the mean molecular weight μ can either be taken constant or it can be calculated from the mass fractions of the chemical (or nuclear) species stored in scalar fields. In the latter case, the effective mean molecular weight can be derived from the original ideal gas law,

$$p = n k_{\mathrm{B}} T \tag{2.2.15}$$

where n is the total number of particles per volume. The fluid consists of several different species with mass fractions X_i, each having a molecular weight M_i and a charge number Z_i. It is assumed that the atoms are fully ionized, corresponding to the environment in deep stellar interiors. The total particle number per volume can thus be expressed as

$$n = \sum_i (1 + Z_i) \frac{\rho X_i N_A}{M_i}. \tag{2.2.16}$$

Comparing this expression with Equation 2.2.14 leads to the definition of the mean molecular weight as

$$\mu = \left(\sum_i (1 + Z_i) \frac{X_i}{M_i} \right)^{-1}. \tag{2.2.17}$$

The remaining partial derivatives of the equation of state needed for the interface therefore read

$$\left.\frac{\partial \epsilon}{\partial \rho}\right|_{T,X} = \frac{RT}{(\gamma-1)\mu} \qquad \left.\frac{\partial \epsilon}{\partial T}\right|_{\rho,X} = \frac{\rho R}{(\gamma-1)\mu} \qquad \left.\frac{\partial \epsilon}{\partial X_i}\right|_{\rho,T} = \frac{\rho R T}{\gamma - 1} \cdot \frac{1 + Z_i}{M_i}. \tag{2.2.18}$$

However, it must be stressed that as long as the Euler equations do not contain a temperature-depended process, the hydrodynamic evolution is independent of temperature and chemical composition. A change in mean molecular weight can not be distinguished from an inverse change in temperature. Therefore, it is usually not necessary to store and evolve the chemical composition explicitly as long as an ideal gas is used.

2.2.3 Ideal gas with radiation pressure

In many cases in stellar interiors, the temperature rises to extremely high values, such that the radiation pressure becomes significant or even dominates over the gas pressure. Therefore, the ideal gas equation of state has to be extended to account for these effects. The pressure and the internal energy are given by (e.g. Weiss et al. 2004)

$$p(\rho, T, X) = \frac{\rho R T}{\mu} + \frac{4\sigma_{\mathrm{sb}}}{3 c_l} T^4, \tag{2.2.19}$$

$$\epsilon(\rho, T, X) = \frac{\rho R T}{(\gamma-1)\mu} + \frac{4\sigma_{\mathrm{sb}}}{c_l} T^4. \tag{2.2.20}$$

Here, the pressure can not be expressed through the conservative variables analytically. Therefore, given the density and the internal energy, Equation 2.2.20 is solved numerically for the temperature by the Newton-Raphson method. The initial guess for the temperature is chosen as the arithmetic

mean of the gas temperature and the radiation temperature. Convergence to machine accuracy is usually reached within a few iterations. The pressure can then be evaluated from Equation 2.2.19. The partial derivatives, needed for the interface of the equation of state are calculated with the procedure described in Section 2.2.1. Special care has to be taken in this step in order to avoid large numerical round-off errors. The treatment of the chemical composition is similar to the ideal gas.

2.3 Gravity

2.3.1 Source terms

The hydrodynamic evolution of gas flows in stars is strongly affected by gravitational forces. The gravitational potential ϕ of a star can be computed from the density distribution self-consistently by solving Poisson's equation:

$$\nabla^2 \phi = 4\pi G \rho \tag{2.3.1}$$

The universal (or Newton's) gravitational constant is denoted by G here. This equation should be solved simultaneously with the Euler equation in order to model to gas flow correctly. However, for a spherical star, the gravitational potential at a certain point is solely determined by the mass within a spherical shell below that point. Since it is not expected that this mass changes significantly within the time scales of interest, the gravitational potential is assumed to be constant in time and externally given for this work. However, there are some cases like full-star simulations of stellar pulsations where this approximation is certainly not valid anymore. The extension to a correct treatment of self-gravity is subject to future work.

Given the external potential, the gravitational acceleration can be calculated by

$$\mathbf{g} = -\nabla \phi. \tag{2.3.2}$$

The corresponding source term which has to be added to the Euler equations reads

$$\mathbf{S}_g = \begin{pmatrix} 0 \\ \rho g_x \\ \rho g_y \\ \rho g_z \\ \rho \left(u g_x + v g_y + w g_z \right) \\ 0 \end{pmatrix}. \tag{2.3.3}$$

The terms in the momentum equations simply describe the action of the external body force. Thus, the total momentum is not conserved anymore. The term in the total energy equation accounts for the corresponding change of kinetic energy. The sum of internal and kinetic equation is therefore also not conserved anymore.

However, the potential energy can also be added to the total energy, defining a new quantity

$$E_\phi = E + \phi. \tag{2.3.4}$$

Rewriting the energy equation from the Euler system in terms of this new variable results in

$$\frac{\partial}{\partial t} \left(\rho E_\phi \right) + \frac{\partial}{\partial x} \left(\rho u E_\phi + u p \right) + \frac{\partial}{\partial y} \left(\rho v E_\phi + v p \right) + \frac{\partial}{\partial z} \left(\rho w E_\phi + w p \right) = S. \tag{2.3.5}$$

The gravitational source term for this new system reads

$$\mathbf{S}_\phi = \begin{pmatrix} 0 \\ \rho g_x \\ \rho g_y \\ \rho g_z \\ 0 \\ 0 \end{pmatrix}, \tag{2.3.6}$$

which clearly shows the conservation of the sum of total and potential energy. For the continuous Euler equations, both formulations of the energy equation are equivalent. However, discretizing both systems may have an effect on the results. The system with E_ϕ as energy variable automatically conserves the energy E_ϕ, if it is discretized with a suitable scheme. In contrast to that, special care has to be taken if the original formulation with the source term (2.3.3) is used. Discretization errors in the momentum and the energy equation have to be consistent in order to inhibit a numerical violation of total energy conversation. However, the discretization methods presented in Chapter 6 fulfill this criterion.

2.3.2 Hydrostatic equilibrium

The Euler equations in the presence of gravity permit a quiet steady state, which is called hydrostatic equilibrium. This can easily be seen by setting time derivatives and velocities to zero. Then, only the momentum equations have non-vanishing terms and reduce to

$$\nabla p = \rho \, \mathbf{g}. \tag{2.3.7}$$

Thus, in a quiet atmosphere, the pressure gradients may be balanced by the gravitational acceleration. Note that the equilibrium depends on the equation of state, since both pressure and density are involved.

An important special case of hydrostatic equilibrium is that of an ideal gas with gravitational acceleration along the z-axis. The differential equation then reads

$$\frac{\partial p}{\partial z} = \frac{g_z \mu}{RT} \cdot p = \frac{1}{H_p} p. \tag{2.3.8}$$

In the last step, the so-called pressure scale height has been introduced, which is generally defined as

$$H_p = p \left(\frac{\partial p}{\partial z} \right)^{-1}. \tag{2.3.9}$$

The hydrostatic equilibrium therefore depends on the vertical profiles of gravity, temperature and mean molecular weight. Assuming that these profiles are given, the differential equation for the hydrostatic pressure can be solved by

$$p(z) = p_0 \exp \left\{ \int_0^z \frac{1}{H_p(z')} dz' \right\}. \tag{2.3.10}$$

In the simplest case, where the pressure scale height is a constant, this reduces to

$$p(z) = p_0 \exp \left\{ \frac{z}{H_p} \right\}, \tag{2.3.11}$$

showing that H_p denotes the length on which the pressure reduces by a factor $1/e$. Moreover, it should be stressed that the hydrostatic equilibrium of an ideal gas solely depend on the ratio of temperature and mean molecular weight and not on the individual quantities (see also Section 2.2.2)

2.4 Thermal radiation

Another important physical process for stellar interiors is energy transport by thermal radiation. It is assumed that the radiation field is in local thermodynamic equilibrium (LTE) with the gas. This assumption is justified by the fact that the mean free path of the photons is very small compared to the length scale of temperature variations. Only the outermost layers of a star, where the gas becomes optically thin, can not be treated within this approximation. In LTE, the

spectrum of the radiation can locally be described by a blackbody, leading to a very simple source term for the Euler equations in the energy equation (see e.g. Maeder 2009)

$$S_{\mathrm{rad},\rho E} = \frac{\partial}{\partial x}\left(K\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(K\frac{\partial T}{\partial z}\right), \tag{2.4.1}$$

where the thermal conductivity

$$K = \frac{4ac_l T^3}{3\rho\kappa_{\mathrm{op}}} \tag{2.4.2}$$

is a function of density, temperature and opacity κ_{op}. Note that the form of the source term is identical to the one for heat conduction. Only the conductivity differs in that case. Therefore, the simulation code described in this work can easily be extended to account for this process.

The opacity is usually itself a function of the local thermodynamic state and chemical composition. However, in this work the opacity is assumed to be constant. Appropriate values are taken from the OPAL opacity tables (Iglesias & Rogers 1996). An extension to dynamic opacities is straightforward, but this is subject to future work.

Special attention should be drawn to the form of the source term (2.4.1). It can be viewed as a divergence of a radiative flux $f_{\mathrm{rad},\chi} = K\frac{\partial T}{\partial \chi},\ \chi \in \{x, y, z\}$. Therefore, it can be added to the hydrodynamic fluxes of the Euler equations before the system is discretized, i.e.

$$\begin{array}{l} \mathbf{F}_{x,\rho E} \rightarrow \mathbf{F}_{x,\rho E} + f_{\mathrm{rad},x} \\ \mathbf{F}_{y,\rho E} \rightarrow \mathbf{F}_{y,\rho E} + f_{\mathrm{rad},y} \\ \mathbf{F}_{z,\rho E} \rightarrow \mathbf{F}_{z,\rho E} + f_{\mathrm{rad},z} \end{array} . \tag{2.4.3}$$

As will be seen in the next chapter, this treatment maintains the energy conservation property of the discrete system.

2.5 Non-dimensionalization

For numerical simulations of the Euler equations, as well as their analytic treatment, it is often desirable to get results which are independent of a specific choice of units. This can be achieved by non-dimensionalizing the system. Therefore, each quantity like density, velocity, etc. is decomposed into a product of a reference value (subscribed with an r) and a dimensionless number (marked with a hat), e.g. $\rho = \rho_{\mathrm{r}} \cdot \hat{\rho}$. The reference value should always be chosen such that the dimensionless value is typically of order one. Here, a reference density ρ_{r}, velocity u_{r}, speed of sound c_{r} and length scale x_{r} is chosen. Other suitable reference quantities can be derived directly from them:

$$t_{\mathrm{r}} = \frac{x_{\mathrm{r}}}{u_{\mathrm{r}}} \qquad p_{\mathrm{r}} = \rho_{\mathrm{r}} c_{\mathrm{r}}^2 \qquad E_{\mathrm{r}} = e_{\mathrm{r}} = c_{\mathrm{r}}^2 \tag{2.5.1}$$

These definitions can now be used to replace all dimensional quantities in the Euler equations (3.1.5) and the corresponding flux vectors (3.1.10). After some simple algebraic manipulations, most of the reference quantities cancel out, leading to

$$\frac{\partial \hat{\mathbf{U}}}{\partial \hat{t}} + \frac{\partial \hat{\mathbf{F}}_x}{\partial \hat{x}} + \frac{\partial \hat{\mathbf{F}}_y}{\partial \hat{y}} + \frac{\partial \hat{\mathbf{F}}_z}{\partial \hat{z}} = \hat{\mathbf{S}} \qquad \text{with} \qquad \hat{\mathbf{U}} = \begin{pmatrix} \hat{\rho} \\ \hat{\rho}\hat{u} \\ \hat{\rho}\hat{v} \\ \hat{\rho}\hat{w} \\ \hat{\rho}\hat{E} \\ \hat{\rho}X \end{pmatrix} \tag{2.5.2}$$

and the non-dimensional flux vectors

$$\hat{\mathbf{F}}_x = \begin{pmatrix} \hat{\rho}\hat{u} \\ \hat{\rho}\hat{u}\hat{u} + \frac{1}{M_\mathrm{r}}\hat{p} \\ \hat{\rho}\hat{u}\hat{v} \\ \hat{\rho}\hat{u}\hat{w} \\ \hat{\rho}\hat{u}\hat{E} + \hat{u}\hat{p} \\ \hat{\rho}\hat{u}X \end{pmatrix} \qquad \hat{\mathbf{F}}_y = \begin{pmatrix} \hat{\rho}\hat{v} \\ \hat{\rho}\hat{v}\hat{u} \\ \hat{\rho}\hat{v}\hat{v} + \frac{1}{M_\mathrm{r}}\hat{p} \\ \hat{\rho}\hat{v}\hat{w} \\ \hat{\rho}\hat{v}\hat{E} + \hat{v}\hat{p} \\ \hat{\rho}\hat{v}X \end{pmatrix} \qquad \hat{\mathbf{F}}_z = \begin{pmatrix} \hat{\rho}\hat{w} \\ \hat{\rho}\hat{w}\hat{u} \\ \hat{\rho}\hat{w}\hat{v} \\ \hat{\rho}\hat{w}\hat{w} + \frac{1}{M_\mathrm{r}}\hat{p} \\ \hat{\rho}\hat{w}\hat{E} + \hat{w}\hat{p} \\ \hat{\rho}\hat{w}X \end{pmatrix}. \tag{2.5.3}$$

Thus, the homogeneous, non-dimensional Euler equations solely depend on a single, non-dimensional reference quantity, namely the reference Mach number

$$M_\mathrm{r} = \frac{u_\mathrm{r}}{c_\mathrm{r}}. \tag{2.5.4}$$

It should be further noted that scalar fields are not non-dimensionalized because they are usually unit-free. Moreover, they are expected to have typical values of order one, at least when used for storing mass fractions for chemical abundances.

In order to transform the source term for gravity, it is necessary to define a reference gravitational acceleration g_r. This value can be used to define the so-called reference Froude number

$$F_\mathrm{r} = \frac{u_\mathrm{r}}{\sqrt{g_\mathrm{r} x_\mathrm{r}}} \tag{2.5.5}$$

which is a non-dimensional value, describing the strength of advection relative to gravity. The non-dimensional source term then reads

$$\hat{\mathbf{S}}_g = \frac{1}{F_\mathrm{r}^2} \begin{pmatrix} 0 \\ \hat{\rho}\hat{g}_x \\ \hat{\rho}\hat{g}_y \\ \hat{\rho}\hat{g}_z \\ M_\mathrm{r}^2 \hat{\rho}\left(\hat{u}\hat{g}_x + \hat{v}\hat{g}_y + \hat{w}\hat{g}_z\right) \\ 0 \end{pmatrix}. \tag{2.5.6}$$

For transforming the fluxes for thermal radiation it is further required to define a reference temperature T_r and a reference thermal conductivity K_r. These quantities can be used to define another non-dimensional reference quantity, namely the Péclet number

$$Pe_\mathrm{r} = \frac{\rho_\mathrm{r} c_\mathrm{r}^2 x_\mathrm{r} u_\mathrm{r}}{K_\mathrm{r} T_\mathrm{r}}. \tag{2.5.7}$$

The dimensionless flux for thermal radiation results in

$$\hat{f}_{\mathrm{rad},\chi} = \frac{1}{Pe_\mathrm{r}} \cdot \hat{K} \frac{\partial \hat{T}}{\partial \chi}, \ \chi \in \{\hat{x}, \hat{y}, \hat{z}\}. \tag{2.5.8}$$

Summarizing, it can be stated that the non-dimensional Euler equations, as presented here, solely depend on three reference quantities. All other reference quantities are just a scaling to specific units, but do not influence the dynamics of the system. Moreover, it should be noted that the dimensional Euler equations can easily be retained by setting all reference quantities to one.

2.6 Low Mach number asymptotics

In this section, the behavior of the Euler equations is analyzed as the Mach number is decreased to zero. This is done for the homogeneous system, i.e. in the absence of gravity and thermal radiation.

2.6.1 Sound waves

From everyday life, it is obvious that stationary fluids with no mean velocity field (i.e. at zero Mach number) support the propagation of sound waves. Their behavior can be analyzed (see e.g. Landau & Lifschitz 1991) by considering a fluid at rest with uniform density ρ_0 and pressure p_0. Small deviations from that state can be expressed in a non-dimensional way by

$$\hat{\rho} = \hat{\rho}_0 + M_\mathrm{r}\hat{\rho}_1 \qquad \hat{p} = \hat{p}_0 + M_\mathrm{r}\hat{p}_1 \qquad \hat{u} = M_\mathrm{r}\hat{u}_1, \tag{2.6.1}$$

where the reference Mach number ($M_\mathrm{r} \ll 1$) has been chosen to express the smallness. For simplicity, a one-dimensional geometry is assumed. The evolution of the disturbances is analyzed by inserting this ansatz into the non-dimensional Euler equations, keeping only terms up to order M_r. The continuity equation then results in

$$\frac{\partial\hat{\rho}_1}{\partial\hat{t}} + \hat{\rho}_0\frac{\partial\hat{u}_1}{\partial\hat{x}} = 0, \tag{2.6.2}$$

while the momentum equation reads

$$\hat{\rho}_0\frac{\partial\hat{u}_1}{\partial\hat{t}} + \frac{1}{M_\mathrm{r}^2}\frac{\partial\hat{p}_1}{\partial\hat{x}} = 0. \tag{2.6.3}$$

Instead of invoking the energy equation, the pressure fluctuations are expressed by the density fluctuations under the assumption of reversible, adiabatic thermodynamic processes:

$$\hat{p}_1 = \left.\frac{\partial\hat{p}}{\partial\hat{\rho}}\right|_S \cdot \hat{\rho}_1 = \hat{c}^2 \cdot \hat{\rho}_1 \tag{2.6.4}$$

In this step, the definition of the non-dimensional speed of sound is recognized and inserted into the formula. The continuity equation can then be transformed to

$$\frac{\partial\hat{p}_1}{\partial\hat{t}} + \hat{\rho}_0\hat{c}^2\frac{\partial\hat{u}_1}{\partial\hat{x}} = 0. \tag{2.6.5}$$

Together with the momentum equation, this forms a closed system for the evolution of $\hat{u}_1$ and $\hat{p}_1$. Combining the two equations with the definition of the velocity potential,

$$\hat{u} = \frac{\partial\hat{\varphi}}{\partial\hat{x}}, \tag{2.6.6}$$

finally results in

$$\frac{\partial^2\hat{\varphi}}{\partial\hat{t}^2} + \frac{\hat{c}^2}{M_\mathrm{r}^2}\cdot\frac{\partial^2\hat{\varphi}}{\partial\hat{x}^2} = 0. \tag{2.6.7}$$

This is the well-known linear wave-equation which permits forward and backward propagating sound waves at speeds $\pm\hat{c}/M_\mathrm{r}$.

Two important facts can be inferred form the above derivation. First of all, the Euler equations permit sound waves with arbitrary small velocity fluctuations and thus at arbitrary small (reference) Mach numbers. Secondly, the relative velocity fluctuations have the same order as the pressure fluctuations. Therefore, sound waves may be recognized by the fact that pressure fluctuations scale linearly with the reference Mach number, i.e. $\hat{p} = \hat{p}_0 + M_\mathrm{r}\hat{p}_1$.

2.6.2 Incompressible flow

In the following, the asymptotic behavior of the compressible Euler equations in the limit of zero Mach number is analyzed. The presentation mostly follows the work by Guillard & Viozat (1999). They performed an expansion of the non-dimensional equations in terms of reference

Mach number. Therefore, the non-dimensional density, velocity, pressure and energy are written in the following way:

$$\begin{array}{rclcrcrcl} \hat{\rho} & = & \hat{\rho}_0 & + & \hat{\rho}_1 M_\mathrm{r} & + & \hat{\rho}_2 M_\mathrm{r}^2 & + & \mathcal{O}\left(M_\mathrm{r}^3\right) \\ \hat{\mathbf{q}} & = & \hat{\mathbf{q}}_0 & + & \hat{\mathbf{q}}_1 M_\mathrm{r} & + & \hat{\mathbf{q}}_2 M_\mathrm{r}^2 & + & \mathcal{O}\left(M_\mathrm{r}^3\right) \\ \hat{p} & = & \hat{p}_0 & + & \hat{p}_1 M_\mathrm{r} & + & \hat{p}_2 M_\mathrm{r}^2 & + & \mathcal{O}\left(M_\mathrm{r}^3\right) \\ \hat{E} & = & \hat{E}_0 & + & \hat{E}_1 M_\mathrm{r} & + & \hat{E}_2 M_\mathrm{r}^2 & + & \mathcal{O}\left(M_\mathrm{r}^3\right) \end{array} \tag{2.6.8}$$

Each term in the expansion may be a function of space and time. The terms of zeroth order describe the zero Mach number limit. Inserting these expressions into the non-dimensional Euler equations and keeping only terms up to zeroth order results in

$$\begin{array}{l} \frac{\partial}{\partial \hat{t}} \hat{\rho}_0 + \mathrm{div}\left(\hat{\rho}_0 \hat{\mathbf{q}}_0\right) = 0 \\ \frac{\partial}{\partial \hat{t}} \left(\hat{\rho}_0 \hat{\mathbf{q}}_0\right) + \mathrm{div}\left(\hat{\rho}_0 \hat{\mathbf{q}}_0 \otimes \hat{\mathbf{q}}_0\right) + \frac{1}{M_\mathrm{r}^2} \nabla \hat{p}_0 + \frac{1}{M_\mathrm{r}} \nabla \hat{p}_1 + \nabla \hat{p}_2 = 0 \\ \frac{\partial}{\partial \hat{t}} \left(\hat{\rho}_0 \hat{E}_0\right) + \mathrm{div}\left(\hat{\rho}_0 \hat{E}_0 \hat{\mathbf{q}}_0 + \hat{p}_0 \hat{\mathbf{q}}_0\right) = 0 \end{array} . \tag{2.6.9}$$

In order to achieve convergence for $M_\mathrm{r} \to 0$, it is required that pressure is constant in space up to first order, i.e.

$$\nabla \hat{p}_0 = \nabla \hat{p}_1 = 0. \tag{2.6.10}$$

The expansion in Mach number is also applied to the equation of state. For simplicity in the presentation, an ideal gas is assumed. To zeroth order the equation of state becomes $\hat{p}_0 = (\gamma - 1)\hat{\rho}_0 \hat{E}_0$. Using the fact that $\nabla \hat{p}_0 = 0$, the zeroth order energy equation can be transformed to

$$\frac{\partial \hat{p}_0}{\partial \hat{t}} + \frac{\gamma}{\gamma - 1}\, \hat{p}_0\, \mathrm{div}\left(\hat{\mathbf{q}}_0\right) = 0. \tag{2.6.11}$$

Since the zeroth order pressure has to be constant in space, time variations may only be imposed by boundary conditions, which are assumed to be zero. Therefore, it can be concluded that the velocity field has to be divergence-free in zero Mach number limit:

$$\mathrm{div}\left(\hat{\mathbf{q}}_0\right) = 0 \tag{2.6.12}$$

Inserting this constraint into the zeroth order continuity and momentum equation results in

$$\frac{\partial \hat{\rho}_0}{\partial \hat{t}} + \hat{\mathbf{q}}_0 \cdot \nabla \hat{\rho}_0 = 0, \tag{2.6.13}$$

$$\frac{\partial \hat{\mathbf{q}}_0}{\partial \hat{t}} + \mathrm{div}\left(\hat{\mathbf{q}}_0 \otimes \hat{\mathbf{q}}_0\right) + \frac{1}{\hat{\rho}_0}\, \nabla \hat{p}_2 = 0. \tag{2.6.14}$$

The last three expressions form the well-known *incompressible Euler equations* in non-dimensional form, which are the zero Mach number limit of the compressible Euler equations. Their name originates from the fact that the density of a fluid element can not change, as it is advected with the flow (Equation 2.6.13). Moreover, it should be stressed that the asymptotic behavior of the pressure field is now known, i.e.

$$p\left(\mathbf{x}, t\right) = p_0 + p_2\left(\mathbf{x}, t\right) M_\mathrm{r}^2. \tag{2.6.15}$$

Thus, pressure fluctuations within the compressible Euler equations scale with the *square* of the reference Mach number, as the solution convergences to a solution of the incompressible equations.

2.6.3 General behavior

In the last section, it was shown that the compressible Euler equations approach the incompressible Euler equations as the Mach number is decreased to zero. However, in Section 2.6.1, it was shown that the compressible equations permit sound waves with arbitrary small velocity fluctuations.

These waves can not be described by the incompressible Euler equations, since they do not permit density fluctuations at all. So, it may be concluded that the compressible Euler equations permit two distinct solutions in the low Mach number regime, namely nearly incompressible flow and sound waves. It was proven by Schochet (1994) and Dellacherie (2010) that these two types of solutions decouple from each other as the Mach number is decreased. Therefore, setting up an incompressible flow as initial condition for the compressible Euler equations, the solutions should stay in the incompressible regime. This is an important criterion for the quality of numerical schemes, which should maintain this property.

2.6.4 Kinetic energy

Another important process in the low Mach number regime is the behavior of kinetic energy. It is defined (per unit volume) as

$$\epsilon_{\text{kin}} = \frac{1}{2}\rho q^2 = \frac{1}{2}\rho\left(u^2 + v^2 + w^2\right) \tag{2.6.16}$$

and is usually contained in the total energy ρE in the Euler equations. However, a separate evolution equation for the kinetic energy can be derived as follows. Therefore, each momentum equation is multiplied by its corresponding velocity component, e.g. for the momentum is x-direction:

$$u\frac{\partial \rho u}{\partial t} + u\frac{\partial \rho u^2}{\partial x} + u\frac{\partial \rho uv}{\partial y} + u\frac{\partial \rho uw}{\partial z} + u\frac{\partial p}{\partial x} = 0 \tag{2.6.17}$$

Summing up the three transformed momentum equation and applying some basic theorems of vector calculus results in an evolution equation for the kinetic energy:

$$\frac{\partial \epsilon_{\text{kin}}}{\partial t} + \nabla \cdot \left(\left(\epsilon_{\text{kin}} + p\right)\mathbf{q}\right) = p\ \nabla \cdot \mathbf{q} \tag{2.6.18}$$

This equation is in strong conservation form, except for the source term on the right hand side. However, as seen in Section 2.6.2, the velocity divergence vanishes for incompressible flow. Therefore, the total kinetic energy is conserved in this case. This will be a further test for numerical discretizations, whether they are able to maintain an incompressible flow regime.

2.7 Low Mach number approaches

As seen in the previous sections, the compressible Euler equations tend towards the incompressible Euler equations as the Mach number decreases. Since the simulation of sound waves is usually not of interest in this regime, it is common practice (e.g. in the engineering community) to discretize the incompressible equations for low Mach number flows. However, the previous considerations are only valid in the absence of gravity. Therefore, two very popular approaches to include this source term into the incompressible equations are briefly described the following. It will, however, be shown that their applicability for simulations of stellar interiors is very limited.

2.7.1 Boussinesq approximation

The first approach to include gravity into the incompressible Euler equations is the so-called Boussinesq approximation, which is briefly reviewed in the following. Details about the method and its derivation can be found in the book by Sutherland (2010). For gas dynamics, the equations are most conveniently derived in terms of *potential temperature* which is defined for an ideal gas by

$$\vartheta = T\left(\frac{p}{p_0}\right)^{-\frac{\gamma-1}{\gamma}}. \tag{2.7.1}$$

The pressure p_0 is a reference value at a certain height within an atmosphere. The potential temperature describes the temperature of a fluid element with is adiabatically brought to this reference height. As only adiabatic motions are considered here, the potential temperature has the same significance as the entropy of an ideal gas. Thus, the potential temperature is conserved along the motion of a fluid element, which can be expressed by

$$\frac{D\vartheta}{Dt} = 0, \tag{2.7.2}$$

where the Lagrangian derivative $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{q} \cdot \nabla$ has been introduced. It should be noted that the Boussinesq approximation is usually expressed in terms of density, such that $\frac{D\rho}{Dt} = 0$. This is a very good approximation for stratified liquids, which are nearly incompressible. However, in gaseous atmospheres the density may change due to adiabatic (reversible) motions. Therefore, the above formulation in terms of potential temperature is more reasonable here.

Next, the gravitational source term is added to the momentum equations, which can be written in the following form:

$$\rho \frac{D\mathbf{q}}{Dt} + \nabla p = \rho\, \mathbf{g} \tag{2.7.3}$$

It is assumed that a time-independent hydrostatic equilibrium (hse) state exists, which is defined by

$$\nabla p_{\text{hse}} = \rho_{\text{hse}}\, \mathbf{g}. \tag{2.7.4}$$

It should be noted that the hydrostatic equilibrium also depends on a given profile of the potential temperature because $\rho_{\text{hse}} = \rho_{\text{hse}}(p_{\text{hse}}, \vartheta_{\text{hse}})$. In the following, all thermodynamic variables are expressed in terms of deviations (marked by a tilde) from the hydrostatic values such that

$$p = p_{\text{hse}} + \tilde{p}, \qquad \vartheta = \vartheta_{\text{hse}} + \tilde{\vartheta}, \qquad \rho = \rho_{\text{hse}} + \tilde{\rho}. \tag{2.7.5}$$

Subtracting the hydrostatic balance equation (2.7.4) from the momentum equation (2.7.3) results in

$$\rho \frac{D\mathbf{q}}{Dt} + \nabla \tilde{p} = \tilde{\rho} g. \tag{2.7.6}$$

The actual Boussinesq approximation then assumes that all thermodynamic variables are very close to a spatially constant reference value at a certain height, i.e.

$$p \approx p_0, \qquad \vartheta \approx \vartheta_0, \qquad \rho \approx \rho_0. \tag{2.7.7}$$

The momentum equation can then be simplified to

$$\rho_0 \frac{D\mathbf{q}}{Dt} + \nabla \tilde{p} = \rho_0 \frac{\tilde{\vartheta}}{\vartheta_0} \mathbf{g}, \tag{2.7.8}$$

where the density fluctuations were replaced by potential density fluctuations through a linearization of the corresponding ideal gas law.

The decomposition (2.7.5) can also be used to rewrite Equation (2.7.2) in terms of potential temperature fluctuations:

$$\frac{D\tilde{\vartheta}}{Dt} + \mathbf{q} \cdot \nabla \vartheta_{\text{hse}} = 0 \tag{2.7.9}$$

In total, the Boussinesq system for a gaseous atmosphere consists of the simplified momentum equation (2.7.8), Equation (2.7.9) for the evolution of potential temperature and the requirement of a divergence-free velocity field, i.e. $\nabla \cdot \mathbf{q} = 0$. The latter was derived in Section 2.6.2 for incompressible flow fields and is responsible for the desired filtering of sound waves. However, the Boussinesq system may only be used to simulate *very* small vertical sections of an atmosphere where (2.7.7) is approximately true. It is therefore not suitable for stellar atmospheres, where the length scales of interest involve several pressure scale heights. Since the Boussinesq system is quite simple from a mathematical point of view, it is still useful to derive some theoretical properties of nearly hydrostatic flows (see Chapter 6).

2.7.2 Anelastic equations

Some of the limitations of the Boussinesq approximation can be circumvented by using the so-called anelastic approximation. It also uses a time-independent hydrostatic background state which has to be given. Moreover, the resulting equations are conveniently formulated in terms of potential temperature of an ideal gas. Using the same notation as for the Boussinesq approximation, the anelastic equations result in

$$\nabla \cdot (\rho_{\text{hse}} \mathbf{q}) = 0, \tag{2.7.10}$$

$$\frac{D\mathbf{q}}{Dt} + \nabla \left(\frac{\tilde{p}}{\rho_{\text{hse}}} \right) = \frac{\tilde{\vartheta}}{\vartheta_{\text{hse}}} \mathbf{g}, \tag{2.7.11}$$

$$\frac{D\tilde{\vartheta}}{Dt} + \mathbf{q} \cdot \nabla \vartheta_{\text{hse}} = 0. \tag{2.7.12}$$

A detailed derivation and discussion of these equations can be found in the book by Sutherland (2010). It is worth noting that velocity field is now only divergence free with respect to the background density field. This approximation still filters out the sound waves from the compressible Euler equations. Unlike the Boussinesq approximation, the anelastic equations are able to describe arbitrary large vertical regions of a stratified atmosphere.

The anelastic equations have successfully been used to study stellar atmospheres, mostly in the context of convective flow phenomena (e.g. Glatzmaier 1984; Miesch et al. 2000; Talon et al. 2003; Browning et al. 2004). However, the anelastic approach also has some drawbacks. Most importantly, the derivation of the anelastic equations assumes that the hydrostatic background state has a vanishing gradient in potential temperature. This means that the actual temperature gradient has to be very close to the adiabatic temperature gradient. For the investigation of convective phenomena, this criterion is reasonably well fulfilled. However, in radiative regions in stellar interiors, the temperature gradients may be highly non-adiabatic such that the anelastic approximation can formally not be applied. Moreover, the anelastic equations as presented here are only valid for an ideal gas and a hydrostatic background state which does not evolve in time. These latter limitations may be circumvented by some different derivations of the anelastic equations (see e.g. Miesch 2005, and references in there). However, the description of the anelastic atmosphere get much more complicated in these cases. Generally, it should be noted that there are various different anelastic approximations (see e.g. Brown et al. 2012, and references in there), which only share Equation 2.7.10 but involve different approximations on the thermodynamic quantities. Although, some formulations seem to perform well on slightly non-adiabatic atmospheres, the anelastic results remain at least very questionable for more complex temperature profiles. The anelastic equations are therefore not further investigated in this work.

2.7.3 Fully compressible approach

Due to the drawbacks of the Boussinesq and the anelastic approximation, it has been decided to develop a simulation code for the fully compressible Euler equations. This has the advantage that all flow regimes, from incompressible over moderate ($0.1 < M < 1$) Mach numbers to supersonic flows involving shock waves can in principle be modeled by the equations. Moreover, all thermal and buoyancy effects are included and the extension with other source terms should be simpler.

Unfortunately, simulations of low Mach number flows with the compressible Euler equations pose some severe numerical challenges. First of all, it has to be guaranteed that the low Mach number asymptotics as presented in Section 2.6 are maintained by the numerical scheme. Secondly, the system of equations may become very stiff due to the large disparity of time scales as the Mach number decreases. This stiffness may be further enhanced by source terms. However, it is believed that the numerical challenges are solvable and therefore suitable discretization techniques for low Mach number flows were developed, which are presented in the following chapters. These techniques have been implemented in a new simulation code which aims to simulate low Mach number flows in an accurate and efficient way.

3 Basic discretization techniques

In this chapter, some fundamental discretization techniques are presented, which form the basis for the new simulation code which is called LHC (Low Mach number Hydro Code). Since the fully compressible Euler equations are to be simulated, standard approaches for their discretization are considered as a starting point. These approaches are common techniques which are widely used in many astrophysical and engineering applications to simulate *high* Mach number flows. The suitability of these numerical schemes for *low* Mach number flows is examined in the next chapters, where some of the methods have to be modified in order to resolve these kind of flows in an accurate and efficient way.

The equations to be discretized are the compressible Euler equations in non-dimensional form as presented in Section 2.5. In order to simplify notation, it is assumed for this chapter that all quantities are non-dimensional and the hat is thus omitted.

3.1 General coordinate transformations

So far, all equations have been formulated in a Cartesian coordinate system. However, for many applications it is important to adapt the coordinates to the underlying physical problem. By selecting a suitable system, it is easier to choose an appropriate domain for the discretization and to apply suitable boundary conditions. Moreover, the spacing of the discrete mesh may be easily adapted in such a system, in order to achieve a uniform spatial resolution of the physical processes of interest.

In order to avoid the adaption of the code to many specialized coordinate systems, the equations are transformed to general curvilinear coordinates. The ideas for this treatment are adapted from the work by Kifonidis, K. & Müller, E. (2012).

The new coordinates ξ, η, ζ are introduced by transformation of a global Cartesian coordinate system using the functions

$$\xi\left(x,y,z\right), \qquad \eta\left(x,y,z\right), \qquad \zeta\left(x,y,z\right). \tag{3.1.1}$$

These functions have to be invertible, but can be chosen arbitrary; the resulting coordinate system can even be non-orthogonal. The only constraint is that the second derivatives of the transformation must commute. The first derivatives form the Jacobian matrix of the transformation, while its inverse is given by the Jacobian matrix of the inverse transformation:

$$T = \begin{pmatrix} \xi_{,x} & \xi_{,y} & \xi_{,z} \\ \eta_{,x} & \eta_{,y} & \eta_{,z} \\ \zeta_{,x} & \zeta_{,y} & \zeta_{,z} \end{pmatrix} \qquad T^{-1} = \begin{pmatrix} x_{,\xi} & x_{,\eta} & x_{,\zeta} \\ y_{,\xi} & y_{,\eta} & y_{,\zeta} \\ z_{,\xi} & z_{,\eta} & z_{,\zeta} \end{pmatrix} \tag{3.1.2}$$

Moreover, the determinants of these Jacobians are important for the further calculations and are denoted by:

$$J = \det\left(T\right) \qquad J^{-1} = \det\left(T^{-1}\right) \tag{3.1.3}$$

3.1.1 Transformation of the Euler equations

With these definitions, the derivatives of the flux vectors can be written in the following way:

$$\begin{aligned} J^{-1}\frac{\partial \mathbf{F}_x}{\partial x} &= J^{-1}\frac{\partial \mathbf{F}_x}{\partial \xi}\,\xi_{,x} + J^{-1}\frac{\partial \mathbf{F}_x}{\partial \eta}\,\eta_{,x} + J^{-1}\frac{\partial \mathbf{F}_x}{\partial \zeta}\,\zeta_{,x} \\ &= \frac{\partial}{\partial \xi}\left(J^{-1}\xi_{,x}\mathbf{F}_x\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\eta_{,x}\mathbf{F}_x\right) + \frac{\partial}{\partial \zeta}\left(J^{-1}\zeta_{,x}\mathbf{F}_x\right) - \\ &\quad -\mathbf{F}_x \cdot \left(\frac{\partial}{\partial \xi}\left(J^{-1}\xi_{,x}\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\eta_{,x}\right) + \frac{\partial}{\partial \zeta}\left(J^{-1}\zeta_{,x}\right)\right) \\ &= \frac{\partial}{\partial \xi}\left(J^{-1}\xi_{,x}\mathbf{F}_x\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\eta_{,x}\mathbf{F}_x\right) + \frac{\partial}{\partial \zeta}\left(J^{-1}\zeta_{,x}\mathbf{F}_x\right) \end{aligned} \tag{3.1.4}$$

The corresponding expressions for $\mathbf{F}_y$ and $\mathbf{F}_z$ can be obtained in a similar fashion. Inserting these expressions in the Euler equations and rearranging the terms results in

$$J^{-1}\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_\xi}{\partial \xi} + \frac{\partial \mathbf{F}_\eta}{\partial \eta} + \frac{\partial \mathbf{F}_\zeta}{\partial \zeta} = J^{-1}\mathbf{S} \tag{3.1.5}$$

with

$$\begin{aligned} \mathbf{F}_\xi &= J^{-1}\left(\xi_{,x}\mathbf{F}_x + \xi_{,y}\mathbf{F}_y + \xi_{,z}\mathbf{F}_z\right) \\ \mathbf{F}_\eta &= J^{-1}\left(\eta_{,x}\mathbf{F}_x + \eta_{,y}\mathbf{F}_y + \eta_{,z}\mathbf{F}_z\right) \\ \mathbf{F}_\zeta &= J^{-1}\left(\zeta_{,x}\mathbf{F}_x + \zeta_{,y}\mathbf{F}_y + \zeta_{,z}\mathbf{F}_z\right). \end{aligned} \tag{3.1.6}$$

These are the Euler equations in general curvilinear coordinates. The new flux functions may now be formulated in a direction-independent way.

$$\mathbf{F}_\chi = \begin{pmatrix} \rho\ Q_\chi \\ \rho u\ Q_\chi\ + J^{-1}\chi_{,x}\frac{p}{M_\mathrm{r}^2} \\ \rho v\ Q_\chi\ + J^{-1}\chi_{,y}\frac{p}{M_\mathrm{r}^2} \\ \rho w\ Q_\chi\ + J^{-1}\chi_{,z}\frac{p}{M_\mathrm{r}^2} \\ \rho(E+p/\rho)\ Q_\chi \\ \rho X\ Q_\chi \end{pmatrix} \qquad \chi \in \{\xi,\eta,\zeta\} \tag{3.1.7}$$

$$Q_\chi = J^{-1}\chi_{,x}u + J^{-1}\chi_{,y}v + J^{-1}\chi_{,z}w \tag{3.1.8}$$

This expression can be further simplified with the substitutions

$$\begin{aligned} a_\chi &= \sqrt{\left(J^{-1}\chi_{,x}\right)^2 + \left(J^{-1}\chi_{,y}\right)^2 + \left(J^{-1}\chi_{,z}\right)^2} \\ \mathbf{n}_\chi &= \tfrac{J^{-1}}{a_\chi}\begin{pmatrix} \chi_{,x} \\ \chi_{,y} \\ \chi_{,z} \end{pmatrix} \qquad \chi \in \{\xi,\eta,\zeta\}, \\ q_n &= n_x u + n_y v + n_z w \end{aligned} \tag{3.1.9}$$

leading to the final expression for the transformed flux vectors

$$\mathbf{F}_\chi = a_\chi \cdot \begin{pmatrix} \rho\ q_n \\ \rho u\ q_n\ + n_x\frac{p}{M_\mathrm{r}^2} \\ \rho v\ q_n\ + n_y\frac{p}{M_\mathrm{r}^2} \\ \rho w\ q_n\ + n_z\frac{p}{M_\mathrm{r}^2} \\ \rho(E+p/\rho)\ q_n \\ \rho X\ q_n \end{pmatrix}. \tag{3.1.10}$$

Here, $\mathbf{n}_\chi = (n_x, n_y, n_z)$ is the unit vector along the corresponding coordinate direction (expressed in global Cartesian coordinates) and q_n is the projected velocity in that direction. a_χ has the meaning of a differential surface area. A further advantage of the transformed Euler equations is the fact that the flux vectors have the same analytic form in each direction. Therefore only one implementation for the simulation code is necessary instead of three in the old coordinates.

It must be stressed that only the flux derivatives were transformed to the new coordinate system. Vector quantities such as the velocity and the gravitational acceleration stay in the global Cartesian coordinate system. This has the huge advantage that the homogeneous system (i.e. $\mathbf{S} = 0$) stays in strong conservation form. Transforming the vector quantities would lead to additional geometric source terms which would be problematic for the discretization of the equations.

3.1.2 Transformations for thermal radiation

The fluxes for thermal radiation are transformed analogous to the Euler equations. However, the Cartesian flux functions contain temperature derivatives, which need to be transformed as well:

$$\begin{aligned} f_{\mathrm{rad},\xi} = & \quad J^{-1}\left(\xi_{,x} f_{\mathrm{rad},x} + \xi_{,y} f_{\mathrm{rad},y} + \xi_{,z} f_{\mathrm{rad},z}\right) \\ = & \quad \tfrac{K}{Pe_{\mathrm{r}}} J^{-1}\Big(\\ & \tfrac{\partial T}{\partial \xi}\ \xi_{,x}\xi_{,x} + \tfrac{\partial T}{\partial \eta}\ \eta_{,x}\xi_{,x} + \tfrac{\partial T}{\partial \zeta}\ \zeta_{,x}\xi_{,x} + \\ & \tfrac{\partial T}{\partial \xi}\ \xi_{,y}\xi_{,y} + \tfrac{\partial T}{\partial \eta}\ \eta_{,y}\xi_{,y} + \tfrac{\partial T}{\partial \zeta}\ \zeta_{,y}\xi_{,y} + \\ & \tfrac{\partial T}{\partial \xi}\ \xi_{,z}\xi_{,z} + \tfrac{\partial T}{\partial \eta}\ \eta_{,z}\xi_{,z} + \tfrac{\partial T}{\partial \zeta}\ \zeta_{,z}\xi_{,z}\Big) \\ = & \quad \tfrac{K}{Pe_{\mathrm{r}}} J^{-1}\left(g^{11}\tfrac{\partial T}{\partial \xi} + g^{12}\tfrac{\partial T}{\partial \eta} + g^{13}\tfrac{\partial T}{\partial \zeta}\right) \end{aligned} \tag{3.1.11}$$

$$f_{\mathrm{rad},\eta} = \frac{K}{Pe_{\mathrm{r}}} J^{-1}\left(g^{21}\frac{\partial T}{\partial \xi} + g^{22}\frac{\partial T}{\partial \eta} + g^{23}\frac{\partial T}{\partial \zeta}\right) \tag{3.1.12}$$

$$f_{\mathrm{rad},\zeta} = \frac{K}{Pe_{\mathrm{r}}} J^{-1}\left(g^{31}\frac{\partial T}{\partial \xi} + g^{32}\frac{\partial T}{\partial \eta} + g^{33}\frac{\partial T}{\partial \zeta}\right) \tag{3.1.13}$$

In the last step, the components of the metric tensor,

$$g^{ij} = \mathbf{h}^i \cdot \mathbf{h}^j \quad \text{with} \quad \mathbf{h}^1 = \begin{pmatrix} \xi_{,x} \\ \xi_{,y} \\ \xi_{,z} \end{pmatrix}, \mathbf{h}^2 = \begin{pmatrix} \eta_{,x} \\ \eta_{,y} \\ \eta_{,z} \end{pmatrix}, \mathbf{h}^3 = \begin{pmatrix} \zeta_{,x} \\ \zeta_{,y} \\ \zeta_{,z} \end{pmatrix}, \tag{3.1.14}$$

where introduced. These expressions may then again be added to the fluxes of the transformed Euler system as given by Equation 3.1.10. It should be noted that each flux function contains temperature derivatives in all coordinate directions. However, for numerical reasons (see Section 3.5.4) this is not desirable. Therefore, coordinate transformations are limited to *orthogonal* transformation at the moment, where the off-diagonal entries of metric tensor vanish. Then, the radiative fluxes contain only temperature derivatives in the same coordinate direction again.

3.2 Finite volume discretization

The discretization of the physical domain of interest is performed in curvilinear coordinates ξ, η, ζ. There, the domain is partitioned by a regular, equidistant grid with $N_\xi \times N_\eta \times N_\zeta$ cells. Without loss of generality, the transformation functions may always be chosen such that the extents of the domain are

$$\begin{aligned} \xi &\in [1/2, N_\xi + 1/2] \\ \eta &\in [1/2, N_\eta + 1/2] \\ \zeta &\in [1/2, N_\zeta + 1/2] \end{aligned} \ . \tag{3.2.1}$$

This has the advantage that the cell widths are uniform, i.e. $\Delta\xi = \Delta\eta = \Delta\zeta = 1$, and will simplify some of the further expressions. Integer values of the coordinates refer to cell centers, e.g. $(\xi, \eta, \zeta) = (i, j, k)$. Cell faces are denoted by half integer values in the corresponding direction. For example, $(\xi, \eta, \zeta) = (i, j + 1/2, k)$ refers to the interface at the right-hand side of cell (i, j, k) in η-direction.

The discretization is performed on the non-dimensional Euler equations in curvilinear coordinates, which may be written in the following form:

$$J^{-1}\frac{\partial \mathbf{U}}{\partial t} + \nabla_{\xi\eta\zeta} \cdot \mathcal{F} = J^{-1}\mathbf{S} \qquad \text{with} \qquad \mathcal{F} = \begin{pmatrix} \mathbf{F}_\xi \\ \mathbf{F}_\eta \\ \mathbf{F}_\zeta \end{pmatrix}. \tag{3.2.2}$$

These equations are integrated over the volume $\Omega_{i,j,k}$ of a cell in curvilinear coordinates (where $d\Omega = d\xi\, d\eta\, d\zeta$):

$$\int_{\Omega_{i,j,k}} J^{-1}\frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\Omega_{i,j,k}} \nabla_{\xi\eta\zeta} \cdot \mathcal{F}\; d\Omega = \int_{\Omega_{i,j,k}} J^{-1}\mathbf{S}\; d\Omega. \tag{3.2.3}$$

After exchanging the integral and the derivative in the first term, it can be recognized that the integral represents the average of the conservative variables times the cell's volume Vi,j,k in global Cartesian space. The same holds for the integral over the source term on the right-hand side of the equation. Therefore, new cell-averaged quantities are introduced by

$$\mathbf{U}_{i,j,k} = \frac{1}{V_{i,j,k}} \int_{\Omega_{i,j,k}} J^{-1}\mathbf{U}\; d\Omega, \qquad \mathbf{S}_{i,j,k} = \frac{1}{V_{i,j,k}} \int_{\Omega_{i,j,k}} J^{-1}\mathbf{S}\; d\Omega. \tag{3.2.4}$$

Moreover, the divergence theorem can be applied to the flux integral, replacing the volume integration by an integration over the surface of the cell:

$$\frac{\partial \mathbf{U}_{i,j,k}}{\partial t} + \frac{1}{V_{i,j,k}} \oint_{\partial\Omega} \mathcal{F} \cdot \mathbf{n}\; d\mathcal{S} = \mathbf{S}_{i,j,k} \tag{3.2.5}$$

This surface integral can be further decomposed by assuming that the integrated fluxes through the six surfaces of a three dimensional cell are known. Defining these *numerical fluxes* by

$$\begin{array}{l} \mathbf{F}_{i+1/2,j,k} = \oint_{\partial\Omega(i+1/2,j,k)} \mathbf{F}_\xi\; d\mathcal{S} \\ \mathbf{F}_{i,j+1/2,k} = \oint_{\partial\Omega(i,j+1/2,k)} \mathbf{F}_\eta\; d\mathcal{S} \\ \mathbf{F}_{i,j,k+1/2} = \oint_{\partial\Omega(i,j,k+1/2)} \mathbf{F}_\zeta\; d\mathcal{S} \end{array}, \tag{3.2.6}$$

the finite volume discretization of the Euler equation finally leads to

$$\begin{array}{ll} \frac{\partial \mathbf{U}_{i,j,k}}{\partial t} + \frac{1}{V_{i,j,k}}\big(& \mathbf{F}_{i+1/2,j,k} - \mathbf{F}_{i-1/2,j,k} + \\ & \mathbf{F}_{i,j+1/2,k} - \mathbf{F}_{i,j-1/2,k} + \\ & \mathbf{F}_{i,j,k+1/2} - \mathbf{F}_{i,j,k-1/2} \big) = \mathbf{S}_{i,j,k} \end{array}. \tag{3.2.7}$$

Therefore, cell-averaged quantities are stored in the simulation code. Their values may change exclusively by fluxes through the interfaces of the corresponding cell or by source terms. Since the flux through the common interface of two neighboring cells are the same, the sum of the cell-averaged conservative variables over the whole domain can not change (as long as there are no source terms and fluxes through the boundary interfaces). Therefore, this finite volume discretization perfectly reflects the conversation properties of the continuous equations.

The remaining challenge for the discretization is to find suitable expressions for the numerical fluxes which must be constructed from the surrounding cell-averaged quantities. Moreover, it should be stressed that Equation 3.2.7 is only in semi-discretized form as the temporal derivatives are not discretized yet. The spatial and temporal discretization is performed separately for this work. This has the advantage that several discretization techniques may be combined in a very flexible way. This procedure is known as method of lines and reduces the *partial* differential equations to a set of coupled *ordinary* differential equations for the cell-averaged values, i.e.

$$\frac{\partial \mathbf{U}_{i,j,k}}{\partial t} + \mathbf{R}_{i,j,k} = 0. \tag{3.2.8}$$

where the vector $\mathbf{R}_{i,j,k}$ is denotes the *spatial residual* of a cell. In general, it may depend on all other cell-averaged values on the grid. Suitable numerical discretizations for this ordinary differential equation will be presented in Chapter 5.

Discretizations of the one and two-dimensional Euler equations can be obtained by setting the number of grid cells of dispensable dimensions to one and omitting the numerical fluxes in the corresponding directions. Moreover, the cell volumes in global Cartesian space $V_{i,j,k}$ have to be replaced by the cell widths in 1D or the cell areas in 2D.

3.3 Computation of the geometric terms

3.3.1 General strategy

The finite volume discretization presented in the last section requires the computation of numerical fluxes at the cell interfaces. It was seen in Section 3.1 that the flux computation in general curvilinear coordinates involves derivatives of the corresponding transformation functions. Specifically, numerical values for the interface normal vectors $\mathbf{n}_d$ and the interface areas are needed (see Equation 3.1.9). Moreover, discrete values of the cell volumes in global Cartesian space are required by the finite volume scheme.

In order to compute these terms, it should first be noted that the transformation functions to curvilinear coordinates are usually given by their inverse, i.e.

$$x(\xi,\eta,\zeta), \qquad y(\xi,\eta,\zeta), \qquad z(\xi,\eta,\zeta). \tag{3.3.1}$$

These functions describe the coordinates of each grid cell in global Cartesian space. They can either by given by an analytic expression or by a set of coordinates for all cell corners or centers.

From Equation 3.1.9 it can be seen that only products of the inverse Jacobian determinant J^{-1} and the partial derivatives of the transformation functions ξ,η,ζ are required. Moreover, the thermal radiative fluxes require an additional factor J^{-1}. These expressions can all be formulated in terms of derivatives of x,y,z after some trivial algebraic manipulations:

$$\begin{aligned} M = J^{-1}T \quad &= J^{-1}\begin{pmatrix} \xi_{,x} & \xi_{,y} & \xi_{,z} \\ \eta_{,x} & \eta_{,y} & \eta_{,z} \\ \zeta_{,x} & \zeta_{,y} & \zeta_{,z} \end{pmatrix} = J^{-1}\begin{pmatrix} x_{,\xi} & x_{,\eta} & x_{,\zeta} \\ y_{,\xi} & y_{,\eta} & y_{,\zeta} \\ z_{,\xi} & z_{,\eta} & z_{,\zeta} \end{pmatrix}^{-1} \\ &= \begin{pmatrix} y_{,\eta}z_{,\zeta} - y_{,\zeta}z_{,\eta} & -x_{,\eta}z_{,\zeta} + x_{,\zeta}z_{,\eta} & x_{,\eta}y_{,\zeta} - x_{,\zeta}y_{,\eta} \\ -y_{,\xi}z_{,\zeta} + y_{,\zeta}z_{,\xi} & x_{,\xi}z_{,\zeta} - x_{,\zeta}z_{,\xi} & -x_{,\xi}y_{,\zeta} + x_{,\zeta}y_{,\xi} \\ y_{,\xi}z_{,\eta} - y_{,\eta}z_{,\xi} & -x_{,\xi}z_{,\eta} + x_{,\eta}z_{,\xi} & x_{,\xi}y_{,\eta} - x_{,\eta}y_{,\xi} \end{pmatrix} \end{aligned} \tag{3.3.2}$$

$$J^{-1} = x_{,\xi}\left(y_{,\eta}z_{,\zeta} - y_{,\zeta}z_{,\eta}\right) + x_{,\eta}\left(-y_{,\xi}z_{,\zeta} + y_{,\zeta}z_{,\xi}\right) + x_{,\zeta}\left(y_{,\xi}z_{,\eta} - y_{,\eta}z_{,\xi}\right) \tag{3.3.3}$$

For analytic expressions of x,y,z, the derivatives can be calculated directly. Otherwise, the derivatives can be approximated by finite differences between the coordinates of the given set of points (see next section). The required areas and normal vectors can then be evaluated at the positions of all cell interfaces by:

$$a_\xi = \sqrt{M_{11}^2 + M_{12}^2 + M_{13}^2} \qquad \mathbf{n}_\xi = \frac{1}{a_\xi}\begin{pmatrix} M_{11} \\ M_{12} \\ M_{13} \end{pmatrix} \tag{3.3.4}$$

$$a_\eta = \sqrt{M_{21}^2 + M_{22}^2 + M_{23}^2} \qquad \mathbf{n}_\eta = \frac{1}{a_\eta}\begin{pmatrix} M_{21} \\ M_{22} \\ M_{23} \end{pmatrix} \tag{3.3.5}$$

$$a_\zeta = \sqrt{M_{31}^2 + M_{32}^2 + M_{33}^2} \qquad \mathbf{n}_\zeta = \frac{1}{a_\zeta}\begin{pmatrix} M_{31} \\ M_{32} \\ M_{33} \end{pmatrix} \tag{3.3.6}$$

It should be noted that J^{-1} corresponds to the cell volume V in global Cartesian space (because the curvilinear grid spacing is always set to unity). However, this expression is not used to compute the cell volumes when the transformation is not given analytically. Since the finite differences which are used for computing J^{-1} contain some discretization error, it is not guaranteed that the sum of the resulting cell volumes equals the total volume of the domain. Therefore, the volume of a cell is computed by connecting the cell's corners with given coordinates by straight lines as explained by Vinokur & Kordulla (1983). The resulting total volume is thus conserved.

Moreover, it should be noted that only the three-dimensional case is explained here. Expressions for the geometry terms for one- and two-dimensional grids can easily be derived in a similar way.

3.3.2 Consistency relations

Given the inverse coordinate transformations $x(\xi,\eta,\zeta), y(\xi,\eta,\zeta), z(\xi,\eta,\zeta)$, it was shown in the last section how the required geometry terms can be computed by evaluating the partial derivatives of these functions. However, it is well known in the literature that an improper calculation of these derivatives leads to *spurious* geometric source terms within the discretized equations, which may deteriorate the solution (see e.g. Thompson et al. 1982; Thompson et al. 1985; Flores et al. 1984).

There are two constraints for the numerical evaluation of the partial derivatives. The first constraint concerns the homogeneous part of the Euler equations. It was shown in Section 3.1 that the second derivatives of the transformation functions have to be symmetric. This property has to be fulfilled also in the discrete sense, e.g. for a two-dimensional transformation (Kifonidis, K. & Müller, E. 2012):

$$\begin{aligned}(y_{,\eta})_{i+1/2,j} - (y_{,\eta})_{i-1/2,j} &= (y_{,\xi})_{i,j+1/2} - (y_{,\xi})_{i,j-1/2} \\ (x_{,\eta})_{i+1/2,j} - (x_{,\eta})_{i-1/2,j} &= (x_{,\xi})_{i,j+1/2} - (x_{,\xi})_{i,j-1/2}\end{aligned} \tag{3.3.7}$$

When evaluating the partial derivative of the transformation functions by finite differences, it has to be ensured that the difference stencil fulfills the above relations. Moreover, it must be emphasized that these relations are usually not fulfilled if the derivatives are given by analytic expressions. Therefore, it is suggested that the derivatives are always computed by appropriate finite differences between the coordinates of cell corners or centers, even if the transformation functions are given analytically.

The second constraint for the partial derivatives only concerns the evaluation of the fluxes for thermal radiation. It is shown in Section 3.5.4 that the involved temperature gradients are evaluated numerically by finite differences. The corresponding stencils have to be equal to the stencils used to derive the geometric terms (see Flores et al. 1984; Bian 2008).

3.3.3 Cartesian geometry

An important special case is Cartesian geometry. Here, the geometric terms may be evaluated analytically as shown in the following. The physical domain extends over $[x_1, x_2] \times [y_1, y_2] \times [z_1, z_2]$ and is to be discretized into $N_\xi \times N_\eta \times N_\zeta$ cells. The transformation functions are thus given by

$$\begin{aligned}x(\xi,\eta,\zeta) &= x_1 + \left(\xi - \tfrac{1}{2}\right) \cdot \Delta x, \\ y(\xi,\eta,\zeta) &= y_1 + \left(\eta - \tfrac{1}{2}\right) \cdot \Delta y, \\ z(\xi,\eta,\zeta) &= z_1 + \left(\zeta - \tfrac{1}{2}\right) \cdot \Delta z,\end{aligned} \tag{3.3.8}$$

where the cell widths are denoted by

$$\Delta x = \frac{x_2 - x_1}{N_\xi}, \qquad \Delta y = \frac{y_2 - y_1}{N_\eta}, \qquad \Delta z = \frac{z_2 - z_1}{N_\zeta}. \tag{3.3.9}$$

Using the formulas from Section 3.3.1, the geometric terms result in the following expressions:

$$V = \Delta x \Delta y \Delta z \tag{3.3.10}$$

$$a_\xi = \Delta y \Delta z, \qquad a_\eta = \Delta x \Delta z, \qquad a_\zeta = \Delta x \Delta y \tag{3.3.11}$$

$$\mathbf{n}_\xi = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{n}_\eta = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \qquad \mathbf{n}_\zeta = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.3.12}$$

Inserting these expressions into the discrete curvilinear Euler equations, it can be seen that they become identical to the Euler equations which are discretized directly in Cartesian space. Therefore, the analytic evaluation of the geometric terms is justified in this particular case.

3.4 Reconstruction of interface values

3.4.1 General considerations

For the finite volume discretization present before, only the cell-averaged quantities of the grid are stored. In order to evolve these quantities, numerical fluxes have to be computed at the cell interfaces. It is a common approach to divide the flux computation into two different steps. First, interface-centered values are calculated by interpolation from the cell-centered values. In a second step, the numerical flux is calculated solely from the interpolated values at the interface (see Section 3.5). The remainder of this section briefly describes the basic interpolation routines that are used by LHC. The implementation is based on Toro (2009) where further details can be found.

The interpolation routines used here are applied in curvilinear space along a coordinate direction. Since the interpolation routines are all one dimensional, the further notation is simplified by only specifying one index. For example, $\mathbf{U}_{i+1/2}$ could denote an interface state in ξ, η or ζ direction.

The interpolation methods assume that the solution is represented on the grid by some piecewise defined functions. The integral of such a function over a grid cell has to be consistent with the cell-averaged value. Since each interface is surrounded by two cells, there are usually two different states approaching each interface from the left and the right hand side. For this work, it is assumed that two interface states (denoted by L and R) are reconstructed by a function $\mathcal{R}$ from the cell-averaged values with the following dependencies:

$$\mathbf{U}^{LR}_{i+1/2} = \mathcal{R}^{LR}_{i+1/2}\left(\mathbf{U}_{i-1},\ \mathbf{U}_i,\ \mathbf{U}_{i+1},\ \mathbf{U}_{i+2}\right) \tag{3.4.1}$$

Usually, the reconstruction is performed directly on the cell-averaged conservative variables. However, it is also possible to perform the reconstruction in the so-called primitive variables defined by:

$$\mathbf{V} = \begin{pmatrix} \rho \\ u \\ v \\ w \\ p \\ X \end{pmatrix} \tag{3.4.2}$$

After the reconstruction step, the interface centered primitive variables are transformed back to conservative variables, leading to

$$\mathbf{U}^{LR}_{i+1/2} = \mathbf{U}\left(\mathcal{R}^{LR}_{i+1/2}\left(\mathbf{V}_{i-1},\ \mathbf{V}_i,\ \mathbf{V}_{i+1},\ \mathbf{V}_{i+2}\right)\right). \tag{3.4.3}$$

Moreover, it should be noted that the first and last two interfaces along each coordinate direction may depend on cell-averaged values which are outside of the domain. Therefore, the grid is usually extended by two layers of so-called ghost cells which are filled by appropriate values (see Section 3.6). Alternatively, the reconstruction method can be altered near the boundaries in order to avoid the use of ghost cells. In this case, it is sufficient to compute the interface values at the inner side of the domain, i.e. $\mathbf{U}^R_{1/2}$ and $\mathbf{U}^L_{N+1/2}$ (see Section 3.6).

3.4.2 Constant reconstruction

The most simple reconstruction is performed by assuming that the values within a cell are constant. In this case, the reconstruction function simply reads

$$\mathcal{R}^L_{i+1/2} = \mathbf{U}_i, \qquad \mathcal{R}^R_{i+1/2} = \mathbf{U}_{i+1}. \tag{3.4.4}$$

Using this reconstruction leads to a discretization which is only first-order accurate in space.

3.4.3 Linear reconstruction

Assuming that the solution on the grid is represented by piecewise linear functions leads to the reconstruction function

$$\mathcal{R}^L_{i+1/2} = \mathbf{U}_i + \frac{1}{2}\Delta_i \tag{3.4.5}$$

$$\mathcal{R}^R_{i+1/2} = \mathbf{U}_{i+1} - \frac{1}{2}\Delta_{i+1}. \tag{3.4.6}$$

Thus, the variables within the cell have the slope Δ_i which can generally be approximated by

$$\Delta_i = \frac{1}{2}(1+\omega)(\mathbf{U}_i - \mathbf{U}_{i-1}) + \frac{1}{2}(1-\omega)(\mathbf{U}_{i+1} - \mathbf{U}_i), \tag{3.4.7}$$

where the parameter $\omega \in [-1, 1]$ determines whether the slope is weighted to the left or to the right hand side. If not stated differently, $\omega = 0$ is used which results in central difference approximation. Using this reconstruction, the overall scheme is second-order accurate in space.

If the use of ghost cells should be avoided, the slopes in the boundary cells 1 and N are modified by a one-sided finite difference approximation:

$$\Delta_1 = -\frac{3}{2}\mathbf{U}_1 + 2\mathbf{U}_2 - \frac{1}{2}\mathbf{U}_3 \tag{3.4.8}$$

$$\Delta_N = -\frac{3}{2}\mathbf{U}_N + 2\mathbf{U}_{N-1} - \frac{1}{2}\mathbf{U}_{N-2} \tag{3.4.9}$$

These slopes are constructed to be second-order accurate, but they involve an additional dependence which would not be allowed according to Equation 3.4.1. However, this is justified at the boundary cells because it does not destroy the overall numerical stencil of the scheme (see Section 5.4.3).

Moreover, it is well-known that the linear reconstruction as presented here may lead to spurious oscillations in the solution in the presence of step gradients (Toro 2009). Therefore, it is common practice to use so-called slope limiters which restrict the size of Δ_i in certain regions of the flow. For LHC, various slope limiters have been implemented (Van Leer, minmod, Superbee, etc.). However, if not stated differently, slope limiters are disabled because they do not have a continuous derivative which cause problems with implicit time stepping (see Chapter 5). However, this is usually not harmful since typical flows in the low Mach number regime are expected to be very smooth.

3.5 Numerical flux functions

3.5.1 The Riemann problem

The finite volume discretization presented in Section 3.2 defined numerical flux functions which represent the integrated fluxes over the corresponding surface. Note that the surface integration is trivial (since the curvilinear grid spacing is always unity) assuming that an average flux is given. However, the numerical fluxes have to be defined in way that the properties of the continuous

Euler equations are maintained on a discrete level as far as possible. A common way to do that (e.g. Toro 2009) is by considering one-dimensional Riemann problems at each cell interface:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_\chi}{\partial \chi} = 0 \qquad \chi \in \{\xi, \eta, \zeta\} \tag{3.5.1}$$

$$\mathbf{U}(\chi, t=0) = \begin{cases} U^L_{i+1/2} & \text{for } \chi > 0 \\ U^R_{i+1/2} & \text{for } \chi < 0 \end{cases} \tag{3.5.2}$$

It can be shown that the analytic solution of this problem with piecewise constant initial data is self-similar, i.e.

$$\mathbf{U}(\chi, t) = \mathbf{U}^*(\chi/t). \tag{3.5.3}$$

Assuming that this solution is known, a numerical flux function can be defined by

$$\mathbf{F}_{i+1/2} = \mathbf{F}(\mathbf{U}^*(0)). \tag{3.5.4}$$

In combination with constant extrapolation, this forms the well-known Godunov method. Other reconstruction algorithms lead to higher-order extensions of that method. These Godunov-type schemes are the most common way to discretize the compressible Euler equations. They were designed to capture supersonic flows involving shock waves in a proper way. Their ability to handle low Mach number flows will be analyzed in Chapter 4. Moreover, it should be noted that Godunov methods solve a homogeneous Riemann problem in the absence of source terms, which are discretized separately. Therefore, it should be kept in mind that the resulting numerical flux functions have no information about physical processes governed by source terms.

In order to solve the above Riemann problem, various methods have been developed in literature. First of all, the exact self-similar solution can be calculated numerically. However, since this involves a lot of computational effort (and cannot be done for more complex systems), many so-called approximate Riemann solvers have been developed. The resulting fluxes are easier to compute, while maintaining most (but not all) of the scheme's desirable properties. For the new simulation code LHC, several approximate Riemann solvers have been implemented which are presented in the following sections.

3.5.2 Roe's approximate Riemann solver

The most important Riemann solver in the context of this thesis was constructed by Roe (1981). The scheme approximates the (non-linear) Riemann problem by a local linearization. The solution to the simplified system is then calculated exactly. The resulting numerical flux function can be written in a simple way,

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}\left(\mathbf{U}^L_{i+1/2}\right) + \mathbf{F}\left(\mathbf{U}^R_{i+1/2}\right) - |A_{\text{roe}}|\left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\right)\right), \tag{3.5.5}$$

where A denotes the *flux Jacobian* matrix defined by

$$A = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = R \Lambda R^{-1}. \tag{3.5.6}$$

This matrix can be decomposed into a set of right eigenvectors (the columns of R) and a diagonal matrix containing the corresponding eigenvalues

$$\Lambda = \text{diag}\left(q_n - \frac{1}{M_\text{r}}c,\ q_n + \frac{1}{M_\text{r}}c,\ q_n, q_n, q_n, q_n\right). \tag{3.5.7}$$

This decomposition shows the characteristic waves present in the Euler equations, i.e. sound waves traveling in negative and positive direction and advective/shear waves traveling with the

fluid velocity. Here, the decomposition is just used to define the absolute value of the flux Jacobian matrix:

$$|A| = R\,|\Lambda|\,R^{-1} \tag{3.5.8}$$

The flux Jacobian matrix and its absolute value are evaluated at the so-called Roe-averaged state. It is mainly constructed by the requirement

$$\mathbf{F}\left(\mathbf{U}^R\right) - \mathbf{F}\left(\mathbf{U}^L\right) = A_{\text{roe}}\left(\mathbf{U}^R - \mathbf{U}^L\right) \tag{3.5.9}$$

which ensures conversation across discontinuities. However, the original work of Roe only describes the derivation of the Roe-averaged state for ideal gases. For a general equation of state, the derivation of the corresponding average is not unique anymore. Therefore, many slightly different extensions to the original Roe scheme have been proposed (see e.g. Mottura et al. 1997; Cinnella 2006). For this work, the approach of Vinokur & Montagne (1990) has been implemented. They define predictors for the Roe averages for the velocities $u_{\text{roe}}, v_{\text{roe}}, w_{\text{roe}}$, the total enthalpy H_{roe} and the partial derivatives of the pressure $\left(\frac{\partial p}{\partial \rho}\right)_{\text{roe}}$, $\left(\frac{\partial p}{\partial \epsilon}\right)_{\text{roe}}$ by

$$(.)_{\text{roe}} = \frac{\sqrt{\rho^L}(.)^L + \sqrt{\rho^R}(.)^R}{\sqrt{\rho^L} + \sqrt{\rho^R}}. \tag{3.5.10}$$

The pressure derivatives are then projected to a final state which complies with Equation 3.5.9. However, it is shown in Chapter 4 that the term involving $|A_{\text{roe}}|$ in the numerical flux function is further modified in order to achieve accurate results for low Mach number flows. Since this modification is not unique at all, the projection step for the pressure derivatives is disabled for simplicity. Numerical tests (not presented here) have shown that this has no influence on the simulation results for low Mach number flows.

Moreover, it should be noted that the term involving $|A_{\text{roe}}|$ in Equation 3.5.5 just balances the flux evaluation either towards the left or right approaching state, depending on the characteristic decomposition and its eigenvalues. This process enforces that the characteristic waves are discretized at the *upwind* state of the fluid and guarantees the numerical stability of the scheme. For example, if the flow is supersonic in the corresponding coordinate direction, all eigenvalues are positive and the resulting numerical flux becomes $\mathbf{F}_{i+1/2} = \mathbf{F}\left(\mathbf{U}^L_{i+1/2}\right)$, because the other terms cancel out (assuming Equation 3.5.9 holds). For subsonic flows, the numerical flux is thus a mixture where each characteristic wave is discretized in the upwind direction.

3.5.3 Other flux functions

In addition to the above presented Roe scheme, several other numerical flux functions are implemented in LHC. The first class of these functions can all be formulated by (Toro 2009)

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}\left(\mathbf{U}^L_{i+1/2}\right) + \mathbf{F}\left(\mathbf{U}^R_{i+1/2}\right) - s\left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\right)\right), \tag{3.5.11}$$

where s is a scalar, local signal propagation speed.

The first subtype of these schemes is called *central flux* has $s = 0$. It is just the arithmetic mean of the fluxes approaching from the left and right hand side. Although this flux is fully consistent with the Euler equations, it is known to be numerically unstable, leading to spurious growth of discretization errors which deteriorate the solution. Nonetheless, it can be used on short time scales to compare the results against other numerical flux functions.

The second subtype defines the local signal propagation speed by the maximum of the eigenvalues of the flux Jacobian matrix, i.e.

$$s = \max\left(\left|q_n^R\right| + \frac{1}{M_{\text{r}}}c^R,\ \left|q_n^L\right| + \frac{1}{M_{\text{r}}}c^L\right). \tag{3.5.12}$$

The resulting flux function is called *local Lax-Friedrich* or *Rusanov* flux. It can be seen that it is a further simplification of the Roe flux, where the upwinding is solely determined by fastest sound wave.

The last subtype of this class is schemes is the so-called *Lax-Wendroff* flux where the signal propagation speed is determined by ratio of the local grid spacing to the time step:

$$s = \frac{\Delta x}{\Delta t} \tag{3.5.13}$$

Since the discretization is performed in curvilinear space, the grid spacing is estimated by the cell volume over the interface area:

$$\Delta x = \frac{\min\left(V_i, V_{i+1}\right)}{a_{i+1/2}}. \tag{3.5.14}$$

For a Cartesian geometry, this results in the correct values for the grid spacing. However, it should be noted that the upwinding for this flux function depends on the time step. This is usually not desirable, since it is unlikely that the solution will converge with respect to the time step.

A different approach to compute the numerical flux is the *VFRoe* scheme (Masella et al. 1999; Buffard et al. 2000). It is very similar to Roe's approximate Riemann solver except that the self-similar solution of the Riemann problem is directly approximated with the following expressions:

$$\mathbf{U}^* = \frac{1}{2}\left(\mathbf{U}^L_{i+1/2} + \mathbf{U}^R_{i+1/2} - \operatorname{sgn}\left(A\right)_{\text{roe}}\left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\right)\right) \tag{3.5.15}$$

$$\operatorname{sgn}\left(A\right) = R \operatorname{sgn}\left(\Lambda\right) R^{-1} \tag{3.5.16}$$

The numerical flux is then defined by

$$\mathbf{F}_{i+1/2} = \mathbf{F}\left(\mathbf{U}^*\right). \tag{3.5.17}$$

Since the upwinding of this flux resembles the Roe-scheme, the results of this scheme are expected to be similar.

The last kind of numerical flux functions that are implemented in LHC are so-called *AUSM*$^+$ and *AUSM*$^+$*-up* schemes (Liou 1996, 2006). These fluxes belong to the class of flux-vector splittings and are derived in a very different way compared to Godunov-type schemes. Details on the implementation are postponed to Section 4.3.5 where the behavior of these fluxes in the low Mach number regime is analyzed.

3.5.4 Thermal radiation

In the following, the thermal radiative flux is discretized. However, it should be emphasized that the grid geometry is limited to *orthogonal* coordinate transformations in this case (see Section 3.1.2). The resulting flux function then involves only temperature derivatives in the direction of the flux:

$$f_{\text{rad},\chi} = \frac{K}{Pe_{\text{r}}} \cdot \frac{a_\chi^2}{J^{-1}} \cdot \frac{\partial T}{\partial \chi} \qquad \chi \in \{\xi, \eta, \zeta\} \tag{3.5.18}$$

This flux is discretized at the cell interfaces by approximating the temperature derivative by a finite difference between the neighboring cell-averaged values:

$$f_{\text{rad},i+1/2} = \frac{K_i + K_{i+1}}{2Pe_{\text{r}}} \cdot \frac{a_\chi^2}{J^{-1}} \cdot \left(T_{i+1} - T_i\right) \tag{3.5.19}$$

This central difference is second-order accurate in space. The thermal conductivity at the interface is approximated by the arithmetic mean of the neighboring cells. It should be noted that only cell-averaged values and not the reconstructed interface values are used for this discretization.

Note that if non-orthogonal grids would be allowed for thermal radiation, it would be necessary to approximate temperature derivatives in all coordinates directions at each interface. However,

all possible finite difference stencils with second-order accuracy would necessarily involve cell averaged-values next to the corners of the corresponding cell. The spatial residual of a certain grid cell would then depend on at least 27 other grid cells (in 3D). Since this is very undesirable for implicit temporal discretizations (see Chapter 5), the grid geometry is limited to orthogonal transformations at the moment.

3.6 Boundary conditions

In order to specify the behavior of the discretized Euler equations at the boundaries, two different approaches are implemented in LHC which are presented in the remainder of this section. In general, there are various classes of *physical* boundary conditions which should be reproduced by the discretized equations. The boundaries may be transmissive or reflective to certain kinds of waves. Moreover, the boundaries may have a predefined rate of inflow or outflow of mass, momentum, energy, etc. A physical boundary can even be omitted if the domain is considered to be periodic in a certain coordinate direction.

3.6.1 Ghost cells

The first approach to specify the boundary conditions is the so-called ghost cell method, where the computational grid is extended by several layers of these artificial cells. The reconstruction methods for the interface values may then depend on the ghost cells (see Section 3.4). Due to the stencil of the linear reconstruction schemes described in Section 3.4.3, two layers of ghost cells are needed. Afterwards, the usual numerical flux function is also used for the interfaces at the boundaries. However, the values within the ghost cells have to be chosen in a way that the resulting numerical flux complies with the desired physical boundary condition.

For this work, the ghost cells are always set in curvilinear space along the one-dimensional line perpendicular to the boundary. The following presentation of different boundary treatments is therefore also one-dimensional and can be applied in any coordinate direction.

The most simple case where ghost cells are applicable are periodic boundaries. The values of the ghost cells are just filled with the values for the other end of the domain:

$$\mathbf{U}_i := \mathbf{U}_{N+i} \quad \text{for} \quad i \in \{-1, 0\}, \qquad \mathbf{U}_{N+i} := \mathbf{U}_i \quad \text{for} \quad i \in \{1, 2\} \tag{3.6.1}$$

Another method to fill the ghost cells is to extrapolate the values from the interior. For constant extrapolation, this reads

$$\mathbf{U}_i := \mathbf{U}_1 \quad \text{for} \quad i \in \{-1, 0\}, \qquad \mathbf{U}_{N+i} := \mathbf{U}_N \quad \text{for} \quad i \in \{1, 2\} \tag{3.6.2}$$

whereas a linear extrapolation is obtained by

$$\mathbf{U}_i := 2\mathbf{U}_1 - \mathbf{U}_2 \quad \text{for} \quad i \in \{-1, 0\}, \qquad \mathbf{U}_{N+i} := 2\mathbf{U}_N - \mathbf{U}_{N-1} \quad \text{for} \quad i \in \{1, 2\}. \tag{3.6.3}$$

These boundary conditions are mostly transmissive for the waves within the fluid.

In order to define reflective boundaries, a reflection operator is defined as

$$\text{refl}\left(\mathbf{U}\right) = \begin{pmatrix} \rho \\ \rho u - 2n_x \cdot \rho q_n \\ \rho v - 2n_y \cdot \rho q_n \\ \rho w - 2n_z \cdot \rho q_n \\ \rho E \\ \rho X \end{pmatrix}, \tag{3.6.4}$$

which changes the sign of the momenta normal to the boundary. The geometry terms are taken from the boundary interface. The ghost cells may then be filled by a reflected state defined by

$$\mathbf{U}_i := \text{refl}\left(\mathbf{U}_{1-i}\right) \text{ for } i \in \{-1, 0\}, \qquad \mathbf{U}_{N+i} := \text{refl}\left(\mathbf{U}_{N-i+1}\right) \text{ for } i \in \{1, 2\}. \tag{3.6.5}$$

After applying a reconstruction scheme for the interface values at the boundary, the mass flux ρq_n is approximately zero, enforcing that no mass or energy enters or leaves the domain. Sound waves are typically reflected by this kind of boundary condition. However, it should be noted that depending on the reconstruction scheme, the mass flux is not exactly zero. This leads typically to a spurious behavior in the cells near the boundary. Moreover, it should be emphasized that this boundary also reflects the density and energy profile. For nearly hydrostatic flows, this leads to unphysical values at the boundary interface.

For the last boundary condition implemented by this method, the ghost cells are set to a time-independent, predefined state. This method mimics a far-field boundary condition, which is open to the waves within a flow but specifies a 'quiet' state of the fluid far away from the numerical domain.

Although there are much more elaborated methods to fill the ghost cells (see e.g. Hirsch 1990), it can be seen from the above presentation that the ghost cell method provides only an indirect way of controlling the numerical fluxes at the boundary interfaces. Especially for complicated flow structures possibly involving a hydrostatic background the ghost cell method faces various problems.

3.6.2 Flux boundary conditions

Due to the shortcomings of the ghost cell method, an alternative approach for the boundary treatment is implemented in LHC. There, the numerical flux for the boundary interfaces is computed directly. Since the ghost cells are not filled in this case, the reconstruction schemes (see Section 3.4) have to use one-sided interpolation methods near the boundary. The goal of the new boundary conditions is to solve the Riemann problem at the boundary interfaces under the assumptions and constraints of the physical boundary conditions. A very elegant and flexible way to achieve this is described by Ghidaglia & Pascal (2005), whose approach is implemented in LHC.

As an example for such a flux boundary condition, the case of a rigid wall is illustrated in the following. The first constraint for this is that the velocity normal to the interface vanishes, i.e. $q_n = 0$. The continuous flux function then reduces to

$$\mathbf{F} = \frac{1}{M_\mathrm{r}} \begin{pmatrix} 0 \\ n_x \cdot p_\mathrm{wall} \\ n_y \cdot p_\mathrm{wall} \\ n_z \cdot p_\mathrm{wall} \\ 0 \\ 0 \end{pmatrix}. \tag{3.6.6}$$

Therefore the flux is solely determined by the wall pressure, which can be calculated by considering the characteristic waves. At the boundary, only the sound wave propagating towards the boundary can leave the domain. With this assumption, the wall pressure reads

$$p_\mathrm{wall} = p \mp \frac{\rho\, q_n c^2 M_\mathrm{r}}{c \pm \left(\frac{\partial p}{\partial \epsilon}\right) q_n M_\mathrm{r}}, \tag{3.6.7}$$

This expression is evaluated with the quantities approaching the boundary (e.g. the reconstructed values at the right hand side of the first interface). The different signs are related to first and last interface in the corresponding coordinate direction. More importantly, it can be seen that this kind of boundary treatment does not permit any mass transport through the boundary. For the corresponding reflective ghost cell boundary, this property can only be fulfilled approximately.

Besides wall boundary, several other flux boundary conditions like a far-field boundary and various inlet and outlet boundary conditions with different thermodynamic assumptions have been implemented. Details can be found in (Ghidaglia & Pascal 2005). In general, it can be concluded

that the flux boundary conditions are more physically motivated than the ghost cell boundaries because they permit an easy way of controlling the characteristic waves at the boundary and the corresponding numerical flux. Moreover, the second-order accuracy of the overall scheme can be maintained by the boundary conditions as long as the one-sided reconstruction methods permit. For the ghost-cell method this is a much more complicated task. Unfortunately, it will be seen in the next chapters that the implemented flux boundary conditions also face some problems in the presence of gravity. However, it is believed that flux boundary conditions are superior to ghost cells for the future development of suitable boundary conditions which represent the physical conditions accurately.

4 Numerical flux functions in the low Mach number regime

The numerical flux functions presented in Section 3.5 were mainly developed for transsonic and supersonic flows involving shock waves. Their behavior in the low Mach number regime is analyzed in this chapter. It is shown that the asymptotic limits of the Euler equations presented in Section 2.6 can only be reproduced when the numerical flux functions are modified in some way. It should be noted that the discretization has only been performed in space so far. For the simulation results presented in this chapter, it is therefore assumed that the time derivative in the semi-discrete Euler equations is approximated by a suitable method. The results presented in the following sections do not depend on the specific choice of a time-marching scheme. The discussion of appropriate temporal discretizations for low Mach number flows is therefore postponed to Chapter 5.

4.1 Numerical viscosity

4.1.1 Gresho vortex

In order to study the behavior of the numerical discretization in the low Mach number regime, the so-called Gresho vortex is used. It is a time-independent solution of the *incompressible* Euler equations which consists of a rotating flow where the centrifugal forces are exactly balanced by the pressure gradients. The setup is taken from Liska & Wendroff (2003) and is slightly modified here in order to adjust the maximum Mach number $M_{\max}$ of the rotating fluid. Note that the Gresho vortex is initialized in dimensional units. If the setup would be considered in a non-dimensional way, the maximum Mach number could also serve as reference Mach number.

The two-dimensional physical domain in Cartesian geometry containing the vortex has the extents $[0,1] \times [0,1]$. The boundaries are assumed to be periodic. The gas of uniform density $\rho(x,y) = 1$ is governed by an ideal gas equation of state with $\gamma = 5/3$. The vortex is centered at $(x,y) = (1/2, 1/2)$ and its angular velocity reads

$$u_\phi = \begin{cases} 5r, & 0 \le r < 0.2 \\ 2 - 5r, & 0.2 \le r < 0.4 \\ 0, & 0.4 \le r \end{cases} \tag{4.1.1}$$

with the radius $r = \sqrt{\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2}$. Note that the maximum velocity $u_{\phi,\max} = 1$ is reached at $r = 0.2$. The background pressure is adjusted such that it matches the maximum Mach number $M_{\max}$:

$$p_0 = \frac{\rho\, u_{\phi,\max}^2}{\gamma M_{\max}^2}. \tag{4.1.2}$$

The total pressure is then calculated by adding a dynamic pressure such that the centrifugal forces are balanced:

$$p = \begin{cases} p_0 + \frac{25}{2} r^2, & 0 \le r < 0.2 \\ p_0 + \frac{25}{2} r^2 + 4 \cdot (1 - 5r - \ln 0.2 + \ln r), & 0.2 \le r < 0.4 \\ p_0 - 2 + 4 \ln 2, & 0.4 \le r \end{cases} \tag{4.1.3}$$

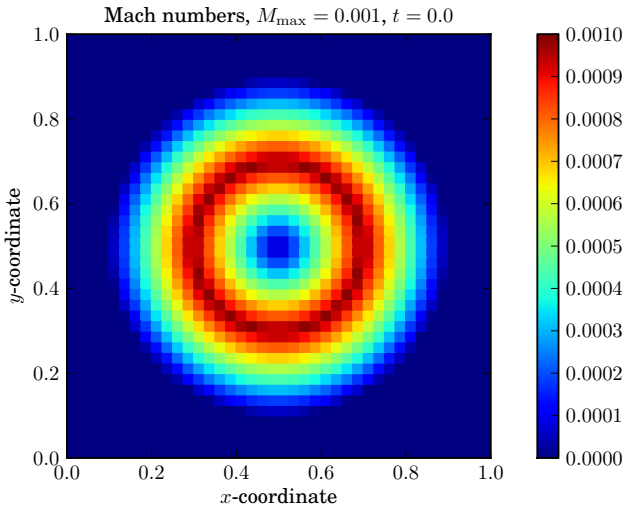


Figure 4.1: Initial Mach number distribution of the Gresho vortex

From this distribution, it is easy to show that the relative pressure fluctuations scale with the square of the Mach number, i.e.

$$\frac{p - p_0}{p_0} \propto M_{\max}^2. \tag{4.1.4}$$

According to Section 2.6.2, the flow is therefore definitely in the incompressible regime here and sound waves are absent. Since the setup is a stationary solution of the incompressible Euler equations, it should not change in time. Deviations seen in numerical simulations are therefore caused by discretization errors.

4.1.2 Godunov-like schemes

In following, it is investigated whether the Gresho vortex can be maintained on a discrete grid in time-dependent simulations. Therefore, the physical domain is discretized by 40×40 grid cells which are initialized with the values given in the previous section (see Figure 4.1). The chosen resolution seems to be rather low here. However, such a vortex might also emerge during the simulation of a more complex flow setup where the vortex takes up only a very small part of the domain. In this respect, the resolution chosen here can be considered to be rather high. The initial conditions are advanced in time with a suitable method from $t = 0$ to $t = 2$, which corresponds to approximately 1.5 rotations of the vortex. The simulations are performed with several numerical flux functions, i.e. with the central flux, Roe's approximate Riemann solver, the Rusanov flux and the AUSM$^+$ scheme. Each flux function is tested with $M_{\max} = 0.1$, $M_{\max} = 0.01$ and $M_{\max} = 0.001$.

The first tests are performed with Roe's approximate Riemann solver. The Mach number distributions at the end of the simulation time are shown in the first three plots of Figure 4.2. It can be seen that the vortex looks reasonable for $M_{\max} = 0.1$ although it is a bit smeared out. However, as the Mach number is further decreased the vortex structure becomes very faint and even unrecognizable at $M_{\max} = 0.001$. This behavior is also confirmed by the evolution of the total kinetic energy (relative to its initial value) which is shown in the first plot of Figure 4.3. As the Mach number is decreased, the dissipation of kinetic energy increases drastically. This behavior is in contradiction with the asymptotic behavior of the Euler equations, where the kinetic energy should be exactly conserved in the zero-Mach number limit. The dissipation is therefore attributed to artificial/numerical viscosity inherent of the discretization.

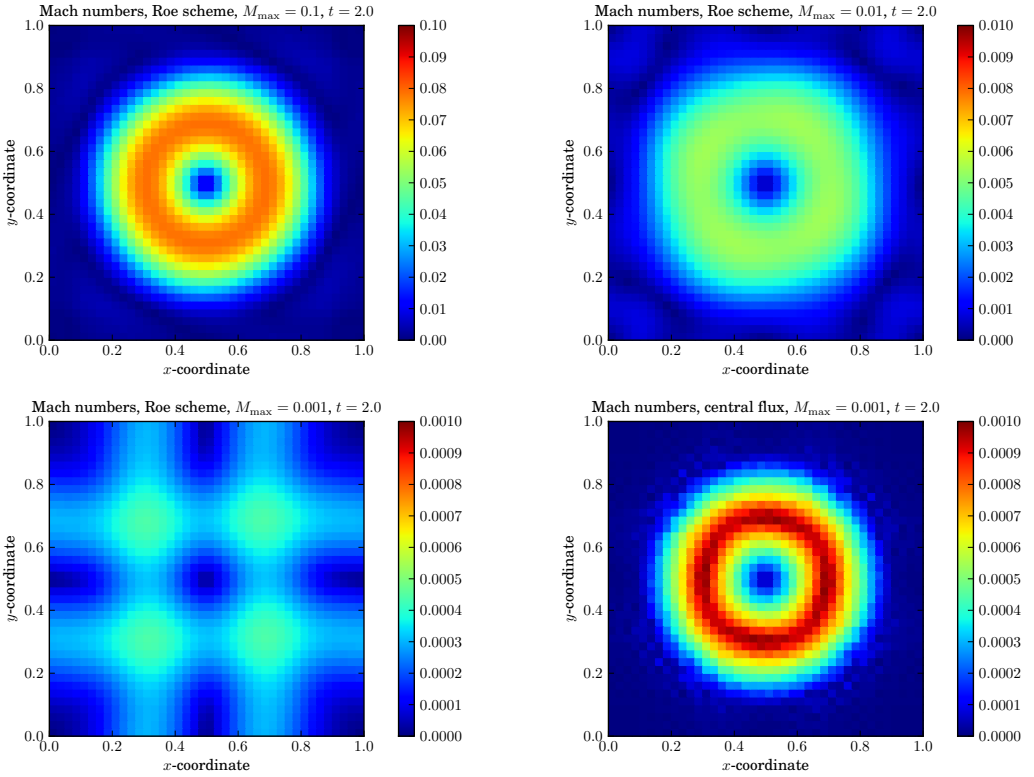


Figure 4.2: Mach number distributions at the end of the simulation time

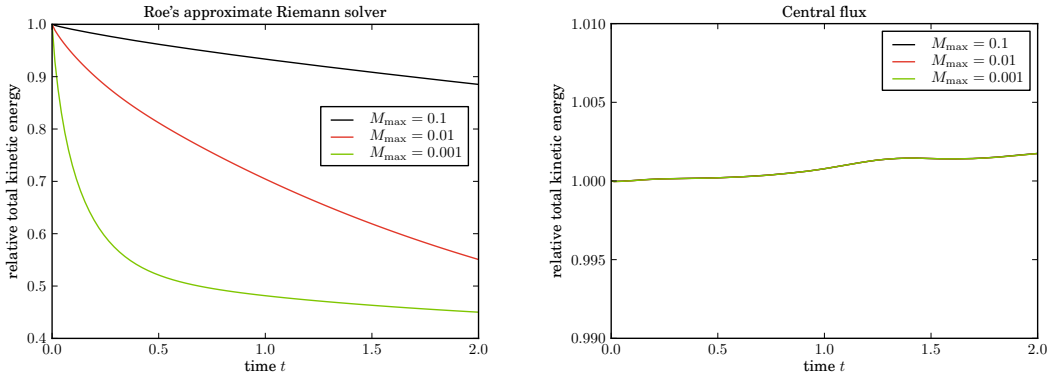


Figure 4.3: Time evolution of total kinetic energy relative to its inital value

A totally different behavior of the discretized equations is obtained when the central flux function is used. This can be seen in the last plot of Figure 4.2 where the Mach number distributions for $M_{\max} = 0.001$ are shown. Besides some noise, the vortex seems to be perfectly conserved. However, the total kinetic energy slightly *increases* during the simulation time (second plot of Figure 4.3). This absolutely unphysical behavior can be explained by the fact that this flux function is numerically unstable. Therefore, it cannot be used for practical simulations because discretization errors grow in time. However, it can also be seen from the evolution of kinetic energy that it does not depend on the Mach number. This interesting result gives a strong hint on the origins of the excessive dissipation of Roe's flux. Comparing both numerical flux functions (Equation 3.5.5 and Equation 3.5.11 with $s = 0$), it can be seen that they are equal except that Roe's flux has an additional upwinding term

$$|A_{\text{roe}}| \left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2} \right), \tag{4.1.5}$$

which can be suspected to be the origin of the numerical dissipation. This will be further investigated in the next sections. As explained in Section 3.5, the upwinding term balances the flux evaluation either to the right or to the left hand side of the interface. However, it was already shown by Dellacherie (2010) that a discretization of the compressible Euler equation only resolves the incompressible limit when the pressure terms within the flux functions are evaluated centrally at the interface. This is the case for the central flux used here. Unfortunately, the central evaluation of the pressure is only justified in the *zero* Mach number limit. For small, but non-zero Mach numbers, such a discretization usually leads to numerically unstable schemes.

Finally, the results of the other numerical flux functions are briefly discussed. The relative total kinetic energy at the end of each simulation is summarized in the following table:

numerical flux	$M_{\max} = 0.1$	$M_{\max} = 0.01$	$M_{\max} = 0.001$
Roe	0.885	0.548	0.450
central	1.002	1.002	1.002
Rusanov	0.652	0.146	0.000
AUSM$^+$	0.820	0.442	0.474

It can be seen that the other flux functions show a Mach number dependent dissipation similar to the Roe flux. The Rusanov flux has the strongest dissipation and here the vortex has completely vanished at the end of the simulation. The dissipation rate of the AUSM$^+$ flux is comparable to the Roe flux. Therefore, it can be concluded that the excessive numerical dissipation is not a peculiarity of Roe's flux function or Godunov-like methods, since flux vector splitting shows the same behavior.

4.2 Low Mach number scaling

4.2.1 Artificial sound waves

The failure of numerical flux functions for the compressible Euler equations in predicting the incompressible flow field at low Mach numbers is well-known in literature of the engineering community (see e.g. Volpe 1993; Guillard & Viozat 1999). There, simulations of steady-state flow profiles around airfoils show spurious oscillations in the pressure field as the Mach number is decreased and even overwhelm the incompressible flow field at some point. In an extensive analysis Guillard & Murrone (2004) performed an expansion in terms of reference Mach number of the Riemann problem with is solved in Godunov-like methods at the interfaces. The expansion was performed on the non-dimensional Euler equations in a similar way as presented in Section 2.6.2. They showed that the pressure of the self-similar solution of the Riemann problem takes the

following form:

$$p^* = \rho_\mathrm{r} c_\mathrm{r}^2 \left(p_0 - \frac{1}{2} M_\mathrm{r} \sqrt{\gamma p_0 \rho_0} \Delta u_1 + ... \right) \tag{4.2.1}$$

Thus, the pressure at the interface is governed by the incompressible pressure p_0 (constant in space), but contains a fluctuation which scales linearly with the reference Mach number M_r. As explained in Section 2.6.1 this scaling corresponds to sound waves, which are always produced by a first-order velocity jump Δu_1 at the interface. Therefore, the incompressible solution (scaling with M_r^2) cannot be maintained by Godunov-like methods. From a heuristic view, this behavior is not very surprising since Godunov-like methods were developed to advance the piecewise discontinuous solutions in order to capture shock waves. The smooth flow field of nearly incompressible flows is therefore not accurately represented in this picture. The excessive dissipation of the Gresho vortex presented in the last section can be related to this behavior. It is assumed that artificial sound waves are generated in this case, but quickly get dissipated into heat since they are not resolved on this grid. Moreover, it should be noted that Godunov-like methods just ensure a proper upwinding of the numerical flux. Therefore, non-Godunov methods like AUSM$^+$, which rely on similar principles for the upwinding, also show an excessive dissipation.

4.2.2 Analysis of numerical flux functions

In this work, the numerical flux functions are further analyzed in a different way. The scaling of the non-dimensional flux functions is examined in terms of reference Mach numbers. The method is inspired by the work of Turkel (1999) and is applied to all Godunov-like numerical flux functions here. First, it should be noted that the Euler equations fulfill a so-called homogeneity property. The flux function can therefore be written in the quasi-linear form

$$\mathbf{F} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \, \mathbf{U} = A \, \mathbf{U}, \tag{4.2.2}$$

where A denotes the flux Jacobian matrix. The Godunov-like numerical flux function from Section 3.5 can then be transformed to

$$\mathbf{F}_{i+1/2} = \frac{1}{2} \left(A^L \mathbf{U}^L + A^R \mathbf{U}^R - D \left(\mathbf{U}^R - \mathbf{U}^L \right) \right), \tag{4.2.3}$$

where D represents the corresponding upwinding matrix of the scheme. The goal of the analysis is to determine how the matrix elements of A and D scale with the reference Mach number. This is presented in the set of primitive variables $\mathbf{V} = (\rho, u, v, w, p, X)^T$, where the corresponding (non-dimensional) flux Jacobian matrix takes a very simple form:

$$A_\mathbf{V} = \frac{\partial \mathbf{V}}{\partial \mathbf{U}} \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{V}} = \begin{pmatrix} q_n & \rho n_x & \rho n_y & \rho n_z & 0 & 0 \\ 0 & q_n & 0 & 0 & \frac{n_x}{\rho M_\mathrm{r}^2} & 0 \\ 0 & 0 & q_n & 0 & \frac{n_y}{\rho M_\mathrm{r}^2} & 0 \\ 0 & 0 & 0 & q_n & \frac{n_z}{\rho M_\mathrm{r}^2} & 0 \\ 0 & \rho c^2 n_x & \rho c^2 n_y & \rho c^2 n_z & q_n & 0 \\ 0 & 0 & 0 & 0 & 0 & q_n \end{pmatrix} \tag{4.2.4}$$

The Jacobian matrices for this transformation can be found in Appendix C. Since the non-dimensional variables are all of order one, the scaling in the low Mach number regime can be read off directly and results in

$$A_\mathbf{V} \propto \begin{pmatrix} \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 & 0 \\ 0 & \mathcal{O}(1) & 0 & 0 & \mathcal{O}\left(\frac{1}{M_\mathrm{r}^2}\right) & 0 \\ 0 & 0 & \mathcal{O}(1) & 0 & \mathcal{O}\left(\frac{1}{M_\mathrm{r}^2}\right) & 0 \\ 0 & 0 & 0 & \mathcal{O}(1) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}^2}\right) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{O}(1) \end{pmatrix}. \tag{4.2.5}$$

The same procedure can be applied to the upwinding matrix of Roe's approximate Riemann solver. After a lengthy calculation (performed with a symbolic algebra computer program) the asymptotic scaling in the low Mach number regime was found to result in

$$D_{\mathbf{V},\mathrm{Roe}} = |A_{\mathbf{V}}| \propto \begin{pmatrix} \mathcal{O}(1) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & 0 \\ 0 & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & 0 \\ 0 & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & 0 \\ 0 & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & 0 \\ 0 & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{O}(1) \end{pmatrix}. \tag{4.2.6}$$

The discrepancy to the flux Jacobian matrix can clearly be seen. In fact, many entries of the upwinding matrix overwhelm the flux Jacobian for low Mach numbers. Therefore, the purpose of the upwinding matrix to balance the flux evaluation to the left or right hand side of an interface can not be fulfilled anymore. The numerical flux is solely governed by this matrix causing the excessive artificial viscosity of the scheme. Especially the $1/M_\mathrm{r}$ terms in the velocity equations (2nd-4th row of D) can be identified as a source of this behavior. Therefore, the upwinding matrix is modified in the next section in order to achieve a scaling which is compatible with the flux Jacobian matrix.

The Rusanov and the Lax-Wendroff flux functions can be analyzed in a similar way. Here, the upwinding matrix is diagonal with the scalar value s. For the Rusanov flux, the low Mach number scaling can be read off directly, i.e.

$$D_{\mathbf{V},\mathrm{Rusanov}} = \left(|q_n| + \frac{c}{M_\mathrm{r}}\right)\mathbb{1} \propto \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right)\mathbb{1}. \tag{4.2.7}$$

Thus, the scaling is also incompatible with the flux Jacobian matrix. A slightly different behavior is seen for the Lax-Wendroff flux where the scaling is independent of the reference Mach number. Instead, it solely depends on the ratio of the grid spacing to the time step. The latter has therefore to be adjusted such that the amount of upwinding is sufficient for a numerically stable scheme and to ensure that the overall dissipation is not to high. Since this may not be achievable in all situations, the use of this numerical flux function is usually avoided.

4.3 Low Mach number fluxes

4.3.1 Roe-Turkel scheme

The deficiencies of Godunov-like numerical flux functions for the simulation of low Mach number flows have been recognized by the engineering community since the late 1980s (e.g. Volpe 1993). However, the main interest for such applications focused on simulations of the steady-state Euler equations, where the time derivative vanishes (e.g. steady flow profiles around an airfoil). A common technique to solve these equations is to advance the solution in time until a steady state is reached. In order to speed up the convergence, the time derivative of the semi-discrete equations is multiplied by the inverse of a preconditioning matrix (e.g. Turkel 1999):

$$\frac{\partial \mathbf{U}}{\partial t} \to P^{-1}\frac{\partial \mathbf{U}}{\partial t} \tag{4.3.1}$$

This type of preconditioning does not alter steady-state solutions, where this derivative vanishes, as long as the preconditioning matrix is invertible. The goal of the preconditioning is to reduce the stiffness of the Euler equations in the low Mach number regime by equalizing the eigenvalues

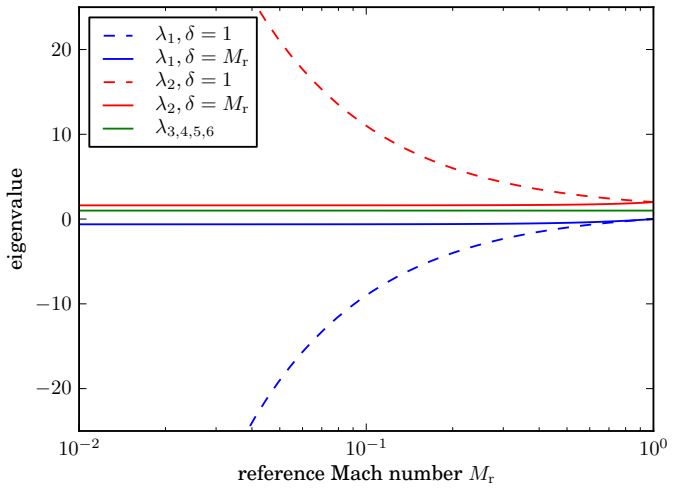


Figure 4.4: Eigenvalues of the Turkel-preconditioned flux Jacobian PA

of the preconditioned flux Jacobian $P\frac{\partial \mathbf{F}}{\partial \mathbf{U}}$. A very popular preconditioner was proposed by Weiss & Smith (1995) and can be expressed in primitive variables by

$$P_\mathbf{V} = \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{\delta^2-1}{c^2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.3.2}$$

The parameter δ should be chosen to equal the local Mach number of the fluid. However, its lower value has to be bounded to a cut-off Mach number M_cut in order to avoid that the preconditioning matrix becomes singular. Moreover, the upper value is limited to one to turn off preconditioning for supersonic flows. The parameter is therefore chosen as

$$\delta = \min\left(1, \max\left(M, M_\mathrm{cut}\right)\right). \tag{4.3.3}$$

The eigenvalues of the preconditioned flux Jacobian then read

$$\lambda_{1,2} = \frac{1}{2}\left(\left(1+\delta^2\right) q_n \mp \frac{1}{M_\mathrm{r}}\sqrt{4c^2\delta^2 + \left(\delta^2-1\right)^2 M_\mathrm{r}^2 q_n^2}\right) \tag{4.3.4}$$

$$\lambda_{3,4,5,6} = q_n. \tag{4.3.5}$$

Their behavior is shown in Figure 4.4 under the assumption that the non-dimensional variables are of order one for $\delta = 1$ (no preconditioning) and $\delta = M_\mathrm{r}$ (ideal preconditioning). This shows that only the acoustic eigenvalues (λ_1, λ_2) of the preconditioned system are modified such that they have the same order of magnitude as the advective eigenvalues.

As a side effect, it has been recognized that this kind of flux preconditioning not only speeds up the convergence to a steady state but significantly improves the accuracy of the results in the low Mach number regime. This indicates that Godunov-like numerical fluxes have to be computed according to the preconditioned Riemann problem

$$\frac{\partial \mathbf{U}}{\partial t} + P\frac{\partial \mathbf{F}_\chi}{\partial \chi} = 0 \qquad \chi \in \{\xi, \eta, \zeta\}, \tag{4.3.6}$$

$$\mathbf{U}\left(\chi, t=0\right) = \begin{cases} U^L_{i+1/2} & \text{for } \chi > 0 \\ U^R_{i+1/2} & \text{for } \chi < 0 \end{cases} . \tag{4.3.7}$$

In fact, it has been shown mathematically by Guillard & Viozat (1999) and Guillard & Murrone (2004) that preconditioned Roe-schemes and Godunov methods are able to resolve the incompressible limit of the Euler equations. This is because no artificial sound waves are generated at the interfaces and thus numerical dissipation does not increase in the low Mach number regime.

Due to the successes of the flux preconditioning technique for steady state flows, it is also desirable to use them for time-dependent flows. For Roe's approximate Riemann solver, this can be achieved by modifying only the upwinding term in the following way (Turkel 1999):

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}\left(\mathbf{U}^L_{i+1/2}\right) + \mathbf{F}\left(\mathbf{U}^R_{i+1/2}\right) - \left(P^{-1}\left|PA\right|\right)_{\text{roe}}\left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\right)\right) \tag{4.3.8}$$

Other terms, such as the time derivative are left unmodified. Therefore, the flux preconditioning only alters the way the numerical flux is balanced at the interfaces. The modified approximate Riemann in combination with the Weiss-Smith preconditioner is referred to as Roe-Turkel scheme in the following.

4.3.2 Analysis of the Roe-Turkel scheme

Despite the proven ability of the Roe-Turkel scheme to resolve nearly incompressible flows, its behavior in the low Mach number regime is further analyzed in the following. To this end, the analysis presented in Section 4.2.2 is applied here to the preconditioned upwind matrix $P^{-1}|PA|$. Assuming that the preconditioning parameter is of order $\delta = \mathcal{O}\left(M_\text{r}\right)$, it can be shown that the upwinding matrix scales as

$$D_\mathbf{V} = P^{-1}_\mathbf{V}\left|P_\mathbf{V} A_\mathbf{V}\right| \propto \begin{pmatrix} \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(\frac{1}{M_\text{r}^2}\right) & 0 \\ 0 & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(\frac{1}{M_\text{r}^2}\right) & 0 \\ 0 & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(\frac{1}{M_\text{r}^2}\right) & 0 \\ 0 & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(\frac{1}{M_\text{r}^2}\right) & 0 \\ 0 & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(1\right) & \mathcal{O}\left(\frac{1}{M_\text{r}^2}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{O}\left(1\right) \end{pmatrix} . \tag{4.3.9}$$

This behavior is much more consistent with the scaling of the flux Jacobian (Equation 4.2.5) compared to the unpreconditioned Roe scheme (Equation 4.2.6). Incorrect scaling in the low Mach number regime is only observed for the $1/M_\text{r}^2$ terms in the first and fifth row, which correspond to a dissipative flux in the density and pressure depending on pressure differences on the grid. It is expected that this behavior is a severe drawback especially for nearly hydrostatic flows, where pressure differences in the vertical direction are not necessarily small. Assuming a hydrostatic equilibrium where all velocities vanish, the upwinding term of the Roe-Turkel scheme can be written in conservative variables as

$$\frac{\partial \mathbf{U}}{\partial \mathbf{V}}\, D_\mathbf{V}\, \Delta\mathbf{V} = \frac{\Delta p}{c \delta M_\text{r}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \frac{\epsilon + p}{\rho} \\ X \end{pmatrix} . \tag{4.3.10}$$

Assuming that the preconditioning parameter δ is also of order M_r (possibly limited by M_cut), it is therefore obvious that hydrostatic pressure differences between grid cells are scaled by $1/M_\text{r}^2$. This leads to an *artificial* diffusive flux in the mass, energy and scalar equations. It is therefore expected that the Roe-Turkel scheme is unable to maintain a hydrostatic equilibrium on a discrete grid, which is confirmed by further theoretical considerations and numerical tests (see Chapter 6).

4.3.3 A new Low Mach number preconditioner

For astrophysical simulations of stellar interiors, it is vital to capture nearly hydrostatic and nearly incompressible flows accurately. The Roe-Turkel scheme presented in the last Sections resolves nearly incompressible flows, but has a potentially strong deficiency for nearly hydrostatic flows. Therefore, a new numerical flux function has been developed in the context of this thesis, which also relies on the flux preconditioning technique. Here, the numerical flux function can also be written as

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}\left(\mathbf{U}^L_{i+1/2}\right) + \mathbf{F}\left(\mathbf{U}^R_{i+1/2}\right) - \left(P^{-1}\left|PA\right|\right)_{\mathrm{roe}}\left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\right)\right). \tag{4.3.11}$$

A new preconditioning matrix is proposed which takes the following form in primitive variables:

$$P_{\mathbf{V}} = \begin{pmatrix} 1 & n_x\frac{\rho\delta M_\mathrm{r}}{c} & n_y\frac{\rho\delta M_\mathrm{r}}{c} & n_z\frac{\rho\delta M_\mathrm{r}}{c} & 0 & 0 \\ 0 & 1 & 0 & 0 & -n_x\frac{\delta}{\rho c M_\mathrm{r}} & 0 \\ 0 & 0 & 1 & 0 & -n_y\frac{\delta}{\rho c M_\mathrm{r}} & 0 \\ 0 & 0 & 0 & 1 & -n_z\frac{\delta}{\rho c M_\mathrm{r}} & 0 \\ 0 & n_x\rho c\delta M_\mathrm{r} & n_y\rho c\delta M_\mathrm{r} & n_z\rho c\delta M_\mathrm{r} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.3.12}$$

Unlike in the Roe-Turkel scheme, the preconditioning parameter δ is expected to be of order $1/M_\mathrm{r}$. Note that $\delta = 0$ corresponds to the unpreconditioned case here. With this assumption, it can be shown that the upwinding matrix scales with the reference Mach number as

$$D_{\mathbf{V}} = P^{-1}_{\mathbf{V}}\left|P_{\mathbf{V}}A_{\mathbf{V}}\right| \propto \begin{pmatrix} \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}^2}\right) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}^2}\right) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}^2}\right) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{O}(1) \end{pmatrix}. \tag{4.3.13}$$

This asymptotic behavior is fully consistent with the scaling of the flux Jacobian matrix (see Equation 4.2.5). Therefore, the new numerical flux is expected to have a numerical viscosity *independent* of the Mach number. Moreover, artificial diffusion should be limited since none of the entries of the upwinding matrix overwhelms the flux Jacobian as the Mach number decreases. The eigenvalues of the preconditioned flux Jacobian PA read

$$\lambda_{1,2} = q_n \mp \frac{1}{M_\mathrm{r}}\tau \qquad \text{with} \quad \tau = \sqrt{c^2\left(1+\delta^2\right) - \delta^2 M_\mathrm{r}^2 q_n^2} \tag{4.3.14}$$

$$\lambda_{3,4,5,6} = q_n. \tag{4.3.15}$$

The convective eigenvalues are left unmodified, whereas the speed of sound in the acoustic eigenvalues is replaced by an artificial speed of sound τ. The eigenvalues are plotted in Figure 4.5. It can be seen that the acoustic eigenvalues increase even faster than in the unpreconditioned case. Therefore, this preconditioner would be absolutely useless to accelerate convergence to a steady state. However, since this does not matter for time-dependent flows, the preconditioner can be chosen such that the accuracy of the numerical scheme is increased. However, the large disparity of the eigenvalues may lead to strong numerical round-off errors for very slow velocities. For the choice of the preconditioning parameter, the lower bound of the Mach number is therefore limited to a cut off Mach number M_cut. Moreover, it is desirable that the numerical flux recovers the unpreconditioned Roe-scheme for supersonic flows. In order to fulfill these two requirements, the preconditioning parameter is set to

$$\delta = \frac{1}{\min\left(1, \max\left(M, M_\mathrm{cut}\right)\right)} - 1. \tag{4.3.16}$$

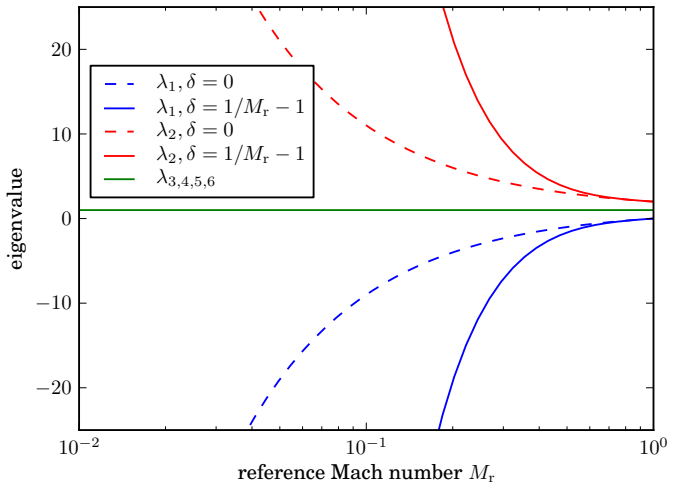


Figure 4.5: Eigenvalues of the low Mach preconditioned flux Jacobian PA

The new numerical flux function is referred to as Roe-Lowmach scheme in the following. Its detailed implementation in LHC is presented in Appendix A. Special care has to be taken in order to avoid large numerical round-off errors. However, so far it must be stressed that the new preconditioning method is neither unique nor proven to be numerically stable. There are several possibilities to choose a preconditioning matrix in order to achieve the same scaling of the upwinding term. However, other preconditioners which were formulated and tested in the context of this thesis either showed a slightly higher artificial viscosity or were numerically unstable and are therefore not presented here. The numerical stability of the new Roe-Lowmach scheme is demonstrated by numerical tests shown later in this thesis.

4.3.4 VFRoe-Lowmach scheme

Apart from Roe's approximate Riemann solver, the VFRoe scheme presented in Section 3.5.3 can also be modified in order to incorporate the low Mach number preconditioning. To this end, it is proposed to approximate the solution of the Riemann problem at the interfaces by

$$\mathbf{U}^* = \frac{1}{2}\left(\mathbf{U}^L_{i+1/2} + \mathbf{U}^R_{i+1/2} - \left(P^{-1}\mathrm{sgn}\left(PA\right)\right)_{\mathrm{roe}}\left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\right)\right). \tag{4.3.17}$$

The numerical flux is then again defined by

$$\mathbf{F}_{i+1/2} = \mathbf{F}\left(\mathbf{U}^*\right). \tag{4.3.18}$$

The preconditioning matrix is taken from the Roe-Lowmach scheme. The resulting numerical flux function is then denoted by VFRoe-Lowmach. To the author's knowledge, flux preconditioning has not been applied to the VFRoe scheme yet. Therefore, some numerical tests for this new scheme are presented in the remainder of this thesis, although its behavior is expected to be very similar to the Roe-Lowmach flux.

4.3.5 AUSM$^+$ schemes

A very popular method to calculate numerical fluxes are the so-called AUSM$^+$ schemes. They were originally developed by Liou (1996) as an alternative to Godunov-like methods. Common

to the AUSM$^+$ schemes is that they split up the continuous flux function into a pressure part in the momentum equations and a convective flux, i.e.

$$\mathbf{F} = p \begin{pmatrix} 0 \\ n_x \\ n_y \\ n_z \\ 0 \\ 0 \end{pmatrix} + \dot{m}\psi, \qquad \dot{m} = \rho q_n, \qquad \psi = \begin{pmatrix} 1 \\ u \\ v \\ w \\ E + \frac{p}{\rho} \\ X \end{pmatrix}. \tag{4.3.19}$$

Note that *this* subsection uses dimensionalized quantities in order to be consistent with the original publications of the schemes (Liou 1996, 2006). The implementation in LHC is performed in a non-dimensional way. The two parts of the flux vector are discretized separately in the following way:

$$\mathbf{F}_{i+1/2} = p_{i+1/2} \begin{pmatrix} 0 \\ n_x \\ n_y \\ n_z \\ 0 \\ 0 \end{pmatrix} + \dot{m}_{i+1/2} \begin{cases} \psi^L_{i+1/2} & \text{for } \dot{m}_{i+1/2} \geq 0 \\ \psi^R_{i+1/2} & \text{otherwise} \end{cases} \tag{4.3.20}$$

Therefore the further discretization relies on a suitable definition of the interface pressure $p_{i+1/2}$ and the interface mass flux $\dot{m}_{i+1/2}$ which determines the upwinding of the vector ψ. However, with the original definition of these interface values, it has been recognized that the numerical flux suffers from similar problems as Roe's approximate Riemann solver in the low Mach number regime. This is also confirmed by simulations of the Gresho vortex presented in Section 4.1.2. Liou (2006) developed an extension to the AUSM$^+$ scheme, called AUSM$^+$-up, which is able to resolve the incompressible limit of the compressible Euler equations. To this end, a scaling Mach number and a corresponding scaling factor were introduced by

$$M_o^2 = \min\left(1, \max\left(\bar{M}^2, M^2_{\text{cut}}\right)\right), \tag{4.3.21}$$

$$f_a = M_o\left(2 - M_o\right). \tag{4.3.22}$$

The scaling Mach number is delimited by a cut-off Mach number at the lower side and by one at the upper end, similar to the low Mach number Roe schemes. The original AUSM$^+$ scheme can be recovered from the AUSM$^+$-up scheme by simply setting $M_{\text{cut}} = 1$ and $f_a = 1$.

The interface pressure of the AUSM$^+$-up scheme is calculated with the following expression:

$$\begin{aligned} p_{i+1/2} \quad = \quad & \mathcal{P}^+_{(5)}\left(M^L_{i+1/2}\right) p^L_{i+1/2} + \mathcal{P}^-_{(5)}\left(M^R_{i+1/2}\right) p^R_{i+1/2} - \\ & K_u \mathcal{P}^+_{(5)} \mathcal{P}^-_{(5)} f_a c_{i+1/2}\left(\rho^L_{i+1/2} + \rho^R_{i+1/2}\right)\left(q^R_n - q^L_n\right) \end{aligned} \tag{4.3.23}$$

The definition and meaning of terms not explained here is given in the original publication (Liou 2006). Note that the last term (second row) corresponds to a velocity diffusion term, which scales with f_a and is thus important for the behavior in the low Mach number regime. The interface mass flux is determined by

$$\dot{m}_{i+1/2} = M_{i+1/2}\, c_{i+1/2} \begin{cases} \rho^L_{i+1/2} & \text{for } M_{i+1/2} > 0 \\ \rho^R_{i+1/2} & \text{otherwise} \end{cases}, \tag{4.3.24}$$

where the interface Mach number is defined as

$$\begin{aligned} M_{i+1/2} \quad = \quad & \mathcal{M}^+_{(4)}\left(M^L_{i+1/2}\right) + \mathcal{M}^-_{(4)}\left(M^R_{i+1/2}\right) - \\ & \frac{K_p}{f_a} \max\left(1 - \sigma \bar{M}^2, 0\right) \frac{p^R_{i+1/2} - p^L_{i+1/2}}{\rho_{i+1/2} c^2_{i+1/2}} \end{aligned}. \tag{4.3.25}$$

The last term corresponds to a pressure diffusion term, which was explicitly added in order "to enhance calculations of low Mach number or multi-phase flow" (Liou 2006). However, it can be seen that this term scales with $1/f_a \propto 1/M_o$ and the pressure difference at the interface. Therefore, the AUSM$^+$-up flux is suspected to have an excessive *artificial* diffusion for nearly hydrostatic flows, similar to the Roe-Turkel scheme (see Section 4.3.2). This is confirmed by a simple numerical test of a one-dimensional hydrostatic atmosphere with zero velocity. There, the low Mach number scaling is solely determined by the cut-off Mach number which is varied. The initial AUSM$^+$-up fluxes at the first interface result in:

M_{cut}	$F_{1/2,\rho}$	$F_{1/2,\rho u}$	$F_{1/2,\rho E}$
10^0	$-0.85 \cdot 10^{-7}$	$0.12 \cdot 10^{1}$	$-0.21 \cdot 10^{-6}$
10^{-1}	$-0.45 \cdot 10^{-6}$	$0.12 \cdot 10^{1}$	$-0.11 \cdot 10^{-5}$
10^{-2}	$-0.43 \cdot 10^{-5}$	$0.12 \cdot 10^{1}$	$-0.11 \cdot 10^{-4}$
10^{-3}	$-0.43 \cdot 10^{-4}$	$0.12 \cdot 10^{1}$	$-0.11 \cdot 10^{-3}$
10^{-4}	$-0.42 \cdot 10^{-3}$	$0.12 \cdot 10^{1}$	$-0.11 \cdot 10^{-2}$

It can clearly be seen that the numerical diffusive flux scales with $1/M_{\text{cut}}$ in the continuity and energy equation. This behavior is similar to the Roe-Turkel scheme (see Equation 4.3.10) where the diffusive flux scales with $1/\delta \propto 1/M_{\text{cut}}$.

In order to reduce the excessive numerical diffusion, a small modification of the AUSM$^+$-up scheme is proposed in this work. To this end, the low Mach number scaling is disabled in the pressure diffusion term. Thus, the scaling variables f_a is set to unity in the definition of the interface Mach number (Equation 4.3.25), while the normal value for f_a is used to calculate the interface pressure (Equation 4.3.23). The modified flux function is denoted as AUSM$^+$-Lowmach. Using this new scheme to calculate the numerical flux of the same hydrostatic atmosphere as above results in:

M_{cut}	$F_{1/2,\rho}$	$F_{1/2,\rho u}$	$F_{1/2,\rho E}$
10^0	$-0.85 \cdot 10^{-7}$	$0.12 \cdot 10^{1}$	$-0.21 \cdot 10^{-6}$
10^{-1}	$-0.85 \cdot 10^{-7}$	$0.12 \cdot 10^{1}$	$-0.21 \cdot 10^{-6}$
10^{-2}	$-0.85 \cdot 10^{-7}$	$0.12 \cdot 10^{1}$	$-0.21 \cdot 10^{-6}$
10^{-3}	$-0.85 \cdot 10^{-7}$	$0.12 \cdot 10^{1}$	$-0.21 \cdot 10^{-6}$
10^{-4}	$-0.85 \cdot 10^{-7}$	$0.12 \cdot 10^{1}$	$-0.21 \cdot 10^{-6}$

Thus, the diffusive flux does not depend on the local Mach number anymore, which is vital for the simulation of nearly hydrostatic flows. However, the modification proposed here was justified by heuristic arguments only. It cannot be excluded that the new scheme lost some of its robustness. Numerical tests presented later in this thesis therefore have to investigate whether the AUSM$^+$-Lowmach scheme provides enough stability for the desired applications.

4.3.6 Low Mach number reconstruction

A very different method to resolve low Mach number flows without excessive artificial viscosity was proposed by Thornber et al. (2008). They recognized that the solution of the Riemann problem at the interfaces (as shown by Guillard & Murrone 2004) contains a pressure term that supports the generation of artificial sound waves which are the source of the excessive numerical dissipation. This term (appearing in Equation 4.2.1) scales with the first order velocity difference with respect to the reference Mach number at the interfaces. Therefore, it was proposed to scale this difference with the local Mach number making the pressure term second order in Mach number. This method can be applied directly after the reconstruction of the interface values

where scaled velocity components (denoted by a tilde) are defined by

$$\begin{pmatrix} \tilde{u}^L \\ \tilde{v}^L \\ \tilde{w}^L \end{pmatrix} = \frac{1}{2}\left\{ \begin{pmatrix} \tilde{u}^L \\ \tilde{v}^L \\ \tilde{w}^L \end{pmatrix} + \begin{pmatrix} \tilde{u}^R \\ \tilde{v}^R \\ \tilde{w}^R \end{pmatrix} \right\} + \frac{s}{2}\left\{ \begin{pmatrix} \tilde{u}^L \\ \tilde{v}^L \\ \tilde{w}^L \end{pmatrix} - \begin{pmatrix} \tilde{u}^R \\ \tilde{v}^R \\ \tilde{w}^R \end{pmatrix} \right\}, \tag{4.3.26}$$

$$\begin{pmatrix} \tilde{u}^R \\ \tilde{v}^R \\ \tilde{w}^R \end{pmatrix} = \frac{1}{2}\left\{ \begin{pmatrix} \tilde{u}^L \\ \tilde{v}^L \\ \tilde{w}^L \end{pmatrix} + \begin{pmatrix} \tilde{u}^R \\ \tilde{v}^R \\ \tilde{w}^R \end{pmatrix} \right\} + \frac{s}{2}\left\{ \begin{pmatrix} \tilde{u}^R \\ \tilde{v}^R \\ \tilde{w}^R \end{pmatrix} - \begin{pmatrix} \tilde{u}^L \\ \tilde{v}^L \\ \tilde{w}^L \end{pmatrix} \right\}. \tag{4.3.27}$$

For supersonic flows the velocities are left unmodified. The scaling parameter is therefore defined as $s = \min(1, M)$. Numerical tests have shown that this scaling suppresses the excessive numerical dissipation in the low Mach number regime. The big advantage of the modified reconstruction is that it can principally be used in combination with any numerical flux function. Moreover, the scaling parameter does not have a lower limit in order to avoid some kind of singularity.

Here, the low Mach number reconstruction applied to Roe's approximate Riemann solver is analyzed in the same manner as the other numerical flux functions. The resulting numerical flux function is denoted by Roe-LMRC in the following. Assuming that the scaling parameter is of order M_r the modified upwinding term scales as

$$|A_\mathbf{V}|\,\Delta\tilde{\mathbf{V}} \propto \begin{pmatrix} \mathcal{O}(1) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 \\ 0 & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & \mathcal{O}(1) & 0 \\ 0 & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}(M_\mathrm{r}) & \mathcal{O}\left(\frac{1}{M_\mathrm{r}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{O}(1) \end{pmatrix} \Delta\mathbf{V}. \tag{4.3.28}$$

Compared to the scaling of the flux Jacobian (Equation 4.2.5) it can be seen that the upwinding matrix does not overwhelm the continuous flux in the velocity equations making the artificial dissipation independent of the Mach number. However, some entries have a much weaker scaling than the flux Jacobian leading to an upwinding which may be significantly to small in the low Mach number regime. Thus, the resulting numerical flux is suspicious to be numerically unstable under certain conditions. In fact, it has been reported that the low Mach number reconstruction is at least prone to checkerboard-like instabilities in the pressure field (see Rieper 2011, and references in there).

4.4 Validation and comparison

In the previous sections, several numerical flux functions were discussed which are supposed to be suitable for simulations in the low Mach number regime. Here, some of the properties of these numerical schemes are compared by simulation results of the LHC code. To this end, the Gresho vortex is set up in the same way as described in Section 4.1.2. For each numerical flux function, the solution is advanced in time to $t = 2$ with four different Mach numbers.

At first, the total kinetic energy relative to its initial value is examined at the end of the simulation time. The results are presented in the following table:

numerical flux	$M_\mathrm{max} = 0.1$	$M_\mathrm{max} = 0.01$	$M_\mathrm{max} = 0.001$	$M_\mathrm{max} = 0.0001$
Roe-Turkel	0.961	0.961	0.961	0.961
Roe-Lowmach	0.972	0.973	0.973	0.973
VFRoe-Lowmach	0.972	0.973	0.973	0.973
Roe-LMRC	0.987	0.990	0.990	0.990
AUSM$^+$-up	0.955	0.957	0.957	0.957
AUSM$^+$-Lowmach	0.966	0.969	0.969	0.969

It can be seen that all numerical fluxes have a numerical dissipation *independent* of the Mach number. Moreover, it should be noted that all schemes have a lower dissipation than the unpreconditioned Roe scheme at $M_{\max} = 0.1$ (see Section 4.1.2). The use of low Mach number methods is therefore recommendable even for moderately low Mach numbers. Moreover, the Roe-Lowmach and the VFRoe-Lowmach scheme behave as expected in precisely the same way. Moreover, the numerical viscosities of all numerical flux functions tested here are comparable.

In the next step of the analysis, the global pressure fluctuations $p_{\text{fluc}} = \frac{p_{\max} - p_{\min}}{p_{\max}}$ are calculated on the grid at the end of each simulation:

numerical flux	$M_{\max} = 0.1$	$M_{\max} = 0.01$	$M_{\max} = 0.001$	$M_{\max} = 0.0001$
Roe-Turkel	$1.2 \cdot 10^{-02}$	$1.3 \cdot 10^{-04}$	$1.3 \cdot 10^{-06}$	$1.3 \cdot 10^{-08}$
Roe-Lowmach	$1.3 \cdot 10^{-02}$	$1.3 \cdot 10^{-04}$	$1.3 \cdot 10^{-06}$	$1.3 \cdot 10^{-08}$
VFRoe-Lowmach	$1.3 \cdot 10^{-02}$	$1.3 \cdot 10^{-04}$	$1.3 \cdot 10^{-06}$	$1.3 \cdot 10^{-08}$
Roe-LMRC	$1.3 \cdot 10^{-02}$	$1.3 \cdot 10^{-04}$	$1.3 \cdot 10^{-06}$	$1.3 \cdot 10^{-08}$
AUSM$^+$-up	$1.2 \cdot 10^{-02}$	$1.3 \cdot 10^{-04}$	$1.3 \cdot 10^{-06}$	$1.3 \cdot 10^{-08}$
AUSM$^+$-Lowmach	$1.3 \cdot 10^{-02}$	$1.3 \cdot 10^{-04}$	$1.3 \cdot 10^{-06}$	$1.3 \cdot 10^{-08}$

These results prove that the pressure fluctuations scale exactly with $M_{\max}^2$. The flow is therefore in the incompressible regime (see Section 2.6.2), which can be resolved by all numerical flux functions.

The last part of the analysis concerns the numerical stability of the different discretizations. All schemes seem to be stable because plots of the flow fields do not show any apparent spurious features. Since a rigorous mathematical stability analysis of the numerical flux functions is hardly possible, a heuristic measure of stability is developed in the following. To start with, it should be noted that numerical instabilities manifest mostly in the high-frequency modes on the discrete grid. Therefore, the two-dimensional, discrete Fourier transformation (DFT2D) of the pressure field is calculated at the end of each simulation. The component with the largest frequency is the so-called Nyquist frequency which describes checkerboard-like modes on the discrete grid. The strength of this component is compared to the $(0,0)$-mode which just specifies the pressure offset. Since an incompressible flow field admits relative pressure fluctuations of order $M_{\max}^2$, only fluctuations above this level are considered. Therefore, the final measure for numerical stability is defined as

$$\alpha = \frac{1}{M_{\max}^2} \cdot \frac{\left|\text{DFT2D}\left(p_{i,j}\right)\right|_{\text{Nyquist}}}{\left|\text{DFT2D}\left(p_{i,j}\right)\right|_{0,0}}. \tag{4.4.1}$$

At the beginning of each simulation, this value is always zero. Since the Gresho vortex is a time-independent solution of the incompressible Euler equations, it is expected that the amplitude of high-frequency modes stays zero. However, it will be shown in Chapter 5, that the temporal evolution of the grid is only performed with a predefined relative accuracy. For the simulations of the Gresho vortex the *minimum* accuracy was set to 10^{-7}. Therefore, values of α *may* also contain random fluctuations up to this level. Only significantly larger values originate from the spatial discretization. The following table summarizes the values of α for each simulation:

numerical flux	$M_{\max} = 0.1$	$M_{\max} = 0.01$	$M_{\max} = 0.001$	$M_{\max} = 0.0001$
Roe-Turkel	$5.4 \cdot 10^{-14}$	$1.1 \cdot 10^{-12}$	$2.9 \cdot 10^{-11}$	$1.2 \cdot 10^{-10}$
Roe-Lowmach	$2.7 \cdot 10^{-07}$	$2.4 \cdot 10^{-07}$	$5.9 \cdot 10^{-08}$	$9.1 \cdot 10^{-08}$
VFRoe-Lowmach	$3.3 \cdot 10^{-11}$	$2.2 \cdot 10^{-13}$	$3.9 \cdot 10^{-09}$	$5.6 \cdot 10^{-10}$
Roe-LMRC	$3.5 \cdot 10^{-16}$	$1.2 \cdot 10^{-07}$	$1.1 \cdot 10^{-04}$	$3.5 \cdot 10^{-04}$
AUSM$^+$-up	$2.6 \cdot 10^{-16}$	$1.1 \cdot 10^{-13}$	$1.3 \cdot 10^{-11}$	$4.3 \cdot 10^{-10}$
AUSM$^+$-Lowmach	$1.6 \cdot 10^{-15}$	$1.8 \cdot 10^{-07}$	$3.0 \cdot 10^{-05}$	$3.5 \cdot 10^{-04}$

It can be seen that the amplitude of the high-frequency mode is significantly above the relative accuracy of the simulation for the Roe-LMRC and the AUSM$^+$-Lowmach scheme. Therefore, it is inferred that the fluctuations are presumably supported by the respective numerical flux

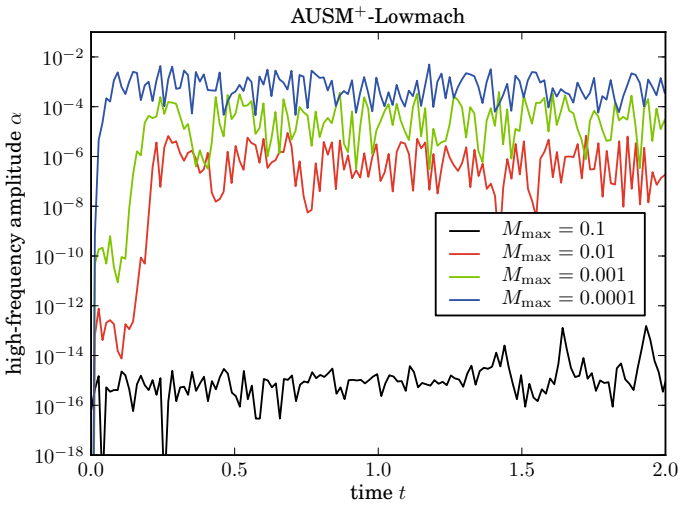


Figure 4.6: Time evolution of the high-frequency pressure mode

functions. Fortunately, these fluctuations do not further grow in time after they have reached a saturation level. This can be seen in Figure 4.6, where the temporal evolution of α is exemplary plotted for the AUSM$^+$-Lowmach scheme. Thus, these schemes can also be considered to be numerically stable. However, all other numerical flux functions have high-frequency pressure fluctuations which are orders of magnitude smaller and should be preferred from this point of view.

Summarizing the theoretical and numerical properties of the low Mach number flux functions, it can be stated that their use is essential in order to capture the asymptotic properties of the compressible Euler equations. However, the discussion of artificial diffusion in the presence of a nearly hydrostatic background shown in Section 4.3.2 and Section 4.3.5 revealed that the Roe-Turkel and the AUSM$^+$-up flux are presumably not suited for this kind of flow. Numerical simulations given in Chapter 6 support this result. From theoretical considerations presented in Section 4.3.6, the Roe-LMRC scheme with the modified reconstruction method is suspicious to have an insufficient upwinding in order to be numerically stable under all flow conditions. This statement is supported by the results presented in this section. The AUSM$^+$-Lowmach scheme revealed a similar behavior. Contrary to that, the Roe-Lowmach and VFRoe-Lowmach seem to be quite robust so far, have a small Mach number independent dissipation and are theoretically well-suited for nearly hydrostatic flows. Thus, the new flux preconditioning technique developed in the context of this thesis is the most promising method for an accurate simulation of stellar interiors. Therefore, further investigations presented in this work are mainly focused on the (VF)Roe-Lowmach scheme.

5 Temporal discretization

So far, the discretization of the compressible Euler equations was only performed in space leading to semi-discrete equation of the form

$$\frac{\partial \mathbf{U}_{i,j,k}}{\partial t} + \mathbf{R}_{i,j,k}\left(\mathbf{U}\right) = 0. \tag{5.0.1}$$

It should be noted that the spatial residual $\mathbf{R}_{i,j,k}$ at a certain grid point may generally depend on the conservative variables at all other grid points. Therefore, $\mathbf{U}$ does not have any indices in $\mathbf{R}_{i,j,k}(\mathbf{U})$ in the above equation. The separation of the spatial and temporal discretization is known as *method of lines* and allows to combine several discretization methods in a very flexible way.

In order to solve the remaining system of ordinary differential equations, several approaches are discussed in this chapter. In general, the temporal evolution of the cell-averaged variables $\mathbf{U}_{i,j,k}$ is approximated by a series of discrete steps $\mathbf{U}^n_{i,j,k}$ starting from the given initial conditions $\mathbf{U}^0_{i,j,k}$. The temporal spacing between two steps is denoted by Δt and does not have to be uniform.

5.1 Explicit time stepping

The simplest way to approximate Equation 5.0.1 is the so-called *forward Euler* method:

$$\frac{\mathbf{U}^{n+1}_{i,j,k} - \mathbf{U}^n_{i,j,k}}{\Delta t} + \mathbf{R}_{i,j,k}\left(\mathbf{U}^n\right) = 0 \tag{5.1.1}$$

Here, the time derivative is approximated by a simple finite difference at each grid point, where the spatial residual is evaluated at the old time step. The conservative variables at the new time step can be directly obtained by solving Equation 5.1.1 for $\mathbf{U}^{n+1}_{i,j,k}$. Since this is possible analytically, the forward Euler scheme belongs to the class of so-called *explicit time steppers*. However, the above discretization is only first-order accurate in time, meaning that the approximation error of the finite difference scales with $(\Delta t)^1$ which is not satisfactory for most practical applications.

However, a huge variety of higher-order explicit time steppers can be found in literature, each with slightly different properties. For the use in LHC, a so-called explicit Runge-Kutta developed by Shu & Osher (1988) has been implemented. It advances the solution to the next time step in a three-stage process:

$$\begin{aligned}
\mathbf{U}^{(1)}_{i,j,k} &= \mathbf{U}^n_{i,j,k} - \Delta t\ \mathbf{R}_{i,j,i}\left(\mathbf{U}^n\right) \\
\mathbf{U}^{(2)}_{i,j,k} &= \frac{3}{4}\mathbf{U}^n_{i,j,k} + \frac{1}{4}\mathbf{U}^{(1)}_{i,j,k} - \frac{1}{4}\Delta t\ \mathbf{R}_{i,j,k}\left(\mathbf{U}^{(1)}\right) \\
\mathbf{U}^{n+1}_{i,j,k} &= \frac{1}{3}\mathbf{U}^n_{i,j,k} + \frac{2}{3}\mathbf{U}^{(2)}_{i,j,k} - \frac{2}{3}\Delta t\ \mathbf{R}_{i,j,k}\left(\mathbf{U}^{(2)}\right)
\end{aligned} \tag{5.1.2}$$

This scheme is denoted by RK3 as it is formally third-order in time. Special emphasis in its derivation has been put on non-linear stability as it fulfills the total variation diminishing (TVD) property.

In general, the advantage of explicit time stepping methods is that they are easy to implement and the resource usage (on the computer) is very low per time step. However, they have stringent conditions on the maximum allowable time step δt. Violating these conditions leads to numerical

instabilities which quickly deteriorate the solution. For the homogeneous Euler equations, the time step is usually determined by the (acoustic) CFL condition

$$\Delta t_{uc} = \frac{\mathrm{CFL}_{uc}}{N_{\mathrm{dim}}} \ \min\left(\frac{\Delta x}{|q_n| + c/M_{\mathrm{r}}}\right), \tag{5.1.3}$$

where the minimum has to be taken over all coordinate directions and grid cells. N_{dim} denotes the spatial dimension of the grid. The condition limits the propagation of fastest waves relative to the width Δx of grid cells. For the homogeneous Euler equations, the restriction applies to sound waves traveling at the velocity $|q_n| + c/M_{\mathrm{r}}$. The dimensionless CFL_{uc} number determines how many cells a sound wave may travel per time step. However, all explicit time steppers have an upper limit of the CFL_{uc} number which is of the order one. Therefore, the time step scales with the Mach number, i.e. $\Delta t \propto M_{\mathrm{r}}$. In the low Mach number regime, this is a big disadvantage since $1/M_{\mathrm{r}}$ time steps are required to advect the fluid from one cell to another. This restriction applies even when no sound waves are present on the grid, which is a typical case for low Mach number flows. Moreover, it has been shown by Birken & Meister (2005) that using the Roe-Turkel scheme results in an even more stringent time step limitation:

$$\Delta t_{uc,\mathrm{lm}} = \frac{\mathrm{CFL}_{uc}}{N_{\mathrm{dim}}} \ \min\left(M \cdot \frac{\Delta x}{|q_n| + c/M_{\mathrm{r}}}\right) \tag{5.1.4}$$

Here, M denotes the local Mach number. Thus, the resulting time step scales as $\Delta t \propto M_{\mathrm{r}}^2$. Numerical experiments with LHC have shown that this restriction also applies to some other low Mach number flux functions. Therefore, it can be concluded that explicit time stepping methods are highly inefficient in the low Mach number regime. Even with strongly optimized codes for the flux evaluation, practical simulations for Mach numbers $\lesssim 0.01$ are hardly feasible.

5.2 Implicit time stepping

The time step limitations of explicit schemes can be avoided by using implicit schemes. Their properties and their implementation are described in the remainder of this chapter.

5.2.1 The Backward Euler scheme

The simplest implicit scheme is the commonly used *backward Euler* method. The solution is advanced in time with the following formula:

$$\frac{\mathbf{U}_{i,j,k}^{n+1} - \mathbf{U}_{i,j,k}^{n}}{\Delta t} + \mathbf{R}_{i,j,k}\left(\mathbf{U}^{n+1}\right) = 0 \tag{5.2.1}$$

The fundamental difference to the forward Euler method is that the spatial residual is evaluated at the new time step. The solution $\mathbf{U}^{n+1}$ is only given by this implicit equation, which can not be solved analytically in general. The reason why this temporal discretization admits larger time steps can be illustrated by a simple heuristic arguments as shown in Figure 5.1. There, the analytic solution of some variable (e.g. the pressure) at some grid point is sketched in gray. It consists of a wave with large amplitude and a large wavelength which symbolizes a slow advection wave. Superimposed to that is a wave with a small amplitude and wavelength which refers to a fast acoustic wave. Starting from a known solution at $t = 0$, an approximation for the solution at $t = 1$ should be found in one step. Using the explicit forward Euler, the derivative is calculated at the initial point and the solution is extrapolated to the final point. From the figure, it can be seen that this results in a huge approximation error because the derivative is solely determined by the fast acoustic wave, but the time step is chosen on the time scale of the advective wave. This explains the stringent time step limitation of explicit methods. However, the backward Euler method performs the extrapolation in the other direction. Therefore, a point at $t = 1$ has to

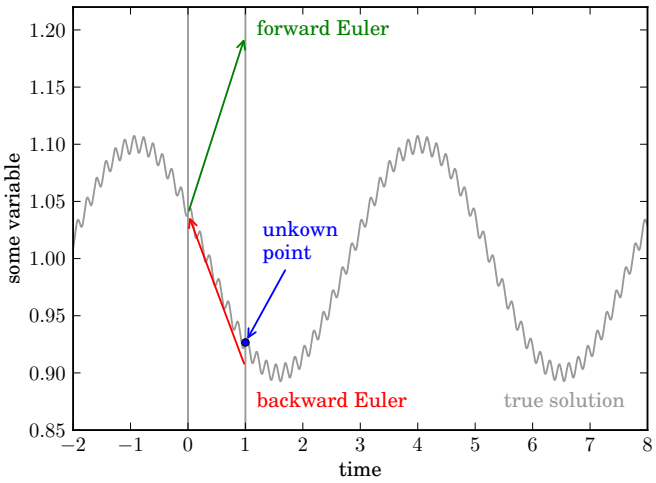


Figure 5.1: Simplified illustration of implicit time stepping

be found where its derivative points backwards to the initial value. Although the details of the fast acoustic wave are not captured by this method, the result is a rather good approximation to the slow advective wave. However, it was shown in Section 2.6 that the nearly incompressible flow field and the sound waves decouple in the low Mach number regime. Since sound waves are usually not of interest, implicit time stepping algorithms are well suited for low Mach number flows.

The stability of implicit schemes can also be analyzed in a mathematically rigorous way (Hairer & Wanner 1996). It can be shown that the backward Euler method applied to *linear* differential equations is numerically stable for arbitrary large time steps. Although the discretization of the Euler equations forms a *non-linear* system of differential equations, implicit time steppers are expected to be very stable in this case.

Moreover, it must be stressed that the numerical stability of implicit methods for very large time steps does not imply that the approximation of the solution fulfills any accuracy requirements. The simplistic illustration from above suggests that the time step has to be chosen such that the physical time scale of interest is well resolved. This is confirmed by numerical experiments presented in Section 5.6. Moreover, it should be noted that the backward Euler method is only first order accurate in time. Since this is usually not desirable for practical simulations, higher order implicit methods are presented in the next section.

Considering the computational efficiency of the backward Euler scheme, the solution of the implicit equation 5.2.1 must be taken into account. Since an analytic inversion of the spatial residual is not possible, the system has to be solved numerically (see Section 5.4). Therefore, the implicit equation is rewritten in the standardized form

$$\mathbf{D}_{i,j,k}\left(\mathbf{U}^{n+1}\right) = \mathbf{R}_{i,j,k}\left(\mathbf{U}^{n+1}\right) + c_t\mathbf{U}^{n+1}_{i,j,k} + \mathbf{C}_{i,j,k} = 0, \tag{5.2.2}$$

which is suitable for the numerical solvers. Here, the so-called defect $\mathbf{D}$ has to be zero for all variables at all grid points. For the backward Euler method the two constants within the defect are defined in the following way:

$$c_t = \frac{1}{\Delta t}, \qquad \mathbf{C}_{i,j,k} = -\frac{1}{\Delta t}\mathbf{U}^{n}_{i,j,k} \tag{5.2.3}$$

The numerical solution of this non-linear system may require substantial computational effort compared to an explicit time step. Therefore, implicit time stepping can only be efficient if the

increase of the size of the time step is large enough to compensate the additional computational work.

5.2.2 ESDIRK schemes

It was seen in the last section that the backward Euler method is only first-order accurate in time. Among the various implicit methods of higher order described in literature (e.g. Hairer & Wanner 1996), a special class of implicit Runge-Kutta methods has been chosen for the implementation in LHC and is briefly presented in the following. In general, a N_s-stage Runge-Kutta method requires the computation of N_s intermediate steps by

$$\mathbf{U}^{(s)}_{i,j,k} = \mathbf{U}^{n}_{i,j,k} - \Delta t \sum_{r=1}^{N_s} \alpha_{s,r} \mathbf{R}_{i,j,k} \left(t + \tau_r \Delta t,\ \mathbf{U}^{(r)} \right). \tag{5.2.4}$$

The solution at the new time step can then be computed with

$$\mathbf{U}^{n+1}_{i,j,k} = \mathbf{U}^{n}_{i,j,k} - \Delta t \sum_{r=1}^{N_s} \beta_{r} \mathbf{R}_{i,j,k} \left(t + \tau_r \Delta t,\ \mathbf{U}^{(r)} \right). \tag{5.2.5}$$

Thus, a Runge-Kutta method is characterized by the coefficients α ($N_s \times N_s$ matrix), β (row vector of size N_s) and τ (column vector of size N_s) which are conveniently written in a so-called Butcher-tableau:

τ	α
	β

Note that a certain Runge-Kutta scheme is explicit if α contains non-zero values only below the diagonal. If α is a full matrix, all intermediate steps are implicitly coupled, resulting in a huge non-linear system to be solved. A good alternative are so-called ESDIRK schemes (Explicit first stage, Singly Diagonally Implicit Runge-Kutta). They require α to have only zeros above the diagonal and that he first stage is explicit ($\alpha_{1,1} = 0$). Moreover, the values for β have to be equal to the last row of α. Assuming that the spatial residual contains no explicit time dependence, an ESDIRK scheme may then be written in the following form:

$$\mathbf{U}^{(1)}_{i,j,k} = \mathbf{U}^{n}_{i,j,k}, \tag{5.2.6}$$

$$\mathbf{U}^{(s)}_{i,j,k} = \mathbf{U}^{n}_{i,j,k} - \Delta t \sum_{r=1}^{s} \alpha_{s,r} \mathbf{R}_{i,j,k} \left(\mathbf{U}^{(r)} \right) \qquad s = \{2, \ldots, N_s\}, \tag{5.2.7}$$

$$\mathbf{U}^{n+1}_{i,j,k} = \mathbf{U}^{(N_s)}_{i,j,k}. \tag{5.2.8}$$

It can be seen that the intermediate stages $(2, ..., N_s)$ are given by an implicit equation which only depends on the results of the previous stages. Hence, an N_s-stage ESDIRK scheme requires the *subsequent* solution of $N_s - 1$ non-linear systems. These systems can also be written in the standardized form which was already used for the backward Euler scheme:

$$\mathbf{D}_{i,j,k} \left(\mathbf{U}^{(s)} \right) = \mathbf{R}_{i,j,k} \left(\mathbf{U}^{(s)} \right) + c_t \mathbf{U}^{(s)}_{i,j,k} + \mathbf{C}_{i,j,k} = 0 \tag{5.2.9}$$

The only difference is that the two constants are now defined as

$$c_t = \frac{1}{\alpha_{s,s} \Delta t}, \qquad \mathbf{C}_{i,j,k} = -\frac{1}{\alpha_{s,s} \Delta t} \mathbf{U}^{n}_{i,j,k} + \sum_{r=1}^{s-1} \frac{\alpha_{s,r}}{\alpha_{s,s}} \mathbf{R}_{i,j,k} \left(\mathbf{U}^{(r)} \right). \tag{5.2.10}$$

Thus, the interface for the non-linear solver does not have to be rewritten for these implicit Runge-Kutta schemes.

Several ESDIRK schemes described by Hosea & Shampine (1996) and Kennedy & Carpenter (2001) have been implemented in LHC. They are denoted by ESDIRK23, ESDIRK34, ESDIRK46

and ESDIRK58 where the first digit represents the formal order of accuracy while the second digit denotes the number of stages N_s. The corresponding Butcher tableaux are listed in Appendix B. Theoretical considerations discussed by the author's of the corresponding ESDIRK schemes show that they share the advantages of the backward Euler method concerning numerical stability. Their accuracy for the simulation of the compressible Euler equations is compared in Section 5.6.

5.3 Time step determination

5.3.1 CFL-like criteria

Previously, it has been shown that the time step is restricted by a so-called acoustic CFL criterion for explicit time steppers which limits the propagation of sound waves to a certain number of grid cells. Although implicit time steppers are not subject to such a restriction in terms of numerical stability, accuracy requirements on the solution may limit the time step. Therefore the concept of a limited signal propagation is generalized here. The time step is thus determined by

$$\Delta t = \frac{\mathrm{CFL}}{N_{\mathrm{dim}}} \min\left(\frac{\Delta x}{s}\right), \tag{5.3.1}$$

where s is the speed of some kind of signal which is to be resolved temporally. Note that the minimum has to be taken over all grid cells and in all curvilinear coordinate directions. Since the geometry terms are only evaluated at cell interfaces, the cell width and the direction of the coordinate line in global Cartesian space are approximated by simple arithmetic means between the right (+) and left (−) interface of a cell:

$$\Delta x = \frac{2V_{i,j,k}}{\left(a_\chi^+ + a_\chi^-\right)} \tag{5.3.2}$$

$$\mathbf{n} = \frac{1}{2}\left(\mathbf{n}_\chi^+ + \mathbf{n}_\chi^-\right) \tag{5.3.3}$$

For Cartesian geometries, these formulas are exact whereas they may be only a rough estimate for strongly deformed grids. However, since the proposed CFL-like criteria do not limit the temporal accuracy in an absolute sense anyway, such estimates are expected be sufficient.

The acoustic CFL_{uc} criterion can be recovered by setting the signal speed to

$$s = |q_n| + \frac{c}{M_\mathrm{r}}. \tag{5.3.4}$$

In this case, it is expected that sound waves are well resolved even with implicit schemes although the computational efficiency may be poor.

For low Mach number flows, it is proposed to set the signal speed to the local fluid velocity, i.e.

$$s = |q_n|. \tag{5.3.5}$$

Thus, the time step is determined such that the fluid moves $\mathrm{CFL} \lesssim 1$ grid cells per time step. The criterion, which is independent of the Mach number, is referred to as advective or CFL_u criterion. It is expected that implicit time steppers smear out sound waves because they are not resolved temporally, but capture the advective flow accurately.

In the presence of gravity, further physical time scales may be relevant for the flow field (see Chapter 6). Since the exact requirements on the time step are not quite clear in this case, a new CFL-like criterion is proposed here which takes gravity into account in its extreme limiting case. To this end, the strength of gravity (compared to advection) along a certain coordinate direction (given by the unit vector $\mathbf{n}$) is defined as

$$g_n = \frac{\mathbf{n} \cdot \mathbf{g}}{F_\mathrm{r}^2}. \tag{5.3.6}$$

It is assumed that a fluid element with initial velocity q_n is not influenced by hydrodynamic processes. Therefore, it is in free fall with its relative position after a time Δt given by

$$x = g_n \left(\Delta t\right)^2 + q_n \Delta t. \tag{5.3.7}$$

Solving this equation for Δt and applying some algebraic manipulations results in

$$\Delta t = \frac{2x}{q_n \pm \sqrt{q_n^2 + 4 g_n x}}. \tag{5.3.8}$$

Limiting the relative position change to

$$|x| = \text{CFL } \Delta x, \tag{5.3.9}$$

this formula fits into the schemes for CFL-like criteria defined above. A free fall signal propagation speed can therefore be defined as

$$s = \frac{1}{2}\left(a q_n + \sqrt{q_n^2 + 4a \; g_n \; \text{CFL } \Delta x}\right). \tag{5.3.10}$$

The constant a is determined by evaluating the various cases given by different directions of gravity and velocity:

condition	a
$q_n > 0, g_n > 0$	$+1$
$q_n < 0, g_n < 0$	-1
$q_n > 0, g_n < 0, \frac{q_n^2}{4g_n} + \text{CFL } \Delta x \leq 0$	$+1$
$q_n > 0, g_n < 0, \frac{q_n^2}{4g_n} + \text{CFL } \Delta x > 0$	-1
$q_n < 0, g_n > 0, \frac{q_n^2}{4g_n} - \text{CFL } \Delta x > 0$	-1
$q_n < 0, g_n > 0, \frac{q_n^2}{4g_n} - \text{CFL } \Delta x < 0$	$+1$

(5.3.11)

The new criterion is referred as free-fall or CFL_{ug} criterion. It is expected that it ensures gravitational effects to be well resolved temporally, although it might be a bit too restrictive since it does not take into account hydrodynamic effects caused by pressure gradients. It should also be noted that the advective CFL criterion becomes singular if all velocities are zero. This is the case when starting a simulation from a quiet hydrostatic atmosphere. However, the free-fall CFL criterion provides a meaningful time step for this case. Moreover, it approaches the advective CFL criterion as gravity becomes weaker.

5.3.2 ESDIRK error estimator

A very different way to calculate the time step is inherently provided by the ESDIRK schemes. To see this, it should be noted that these schemes calculate the solution at the new time step by combining the results from the intermediate stages applying the corresponding set of coefficients β. These coefficients where derived such that the overall method has a certain order of accuracy. However, it is also possible to derive a second set of coefficients $\tilde{\beta}$ that result in a lower order approximation of the solution (see Hairer & Wanner 1996):

$$\tilde{\mathbf{U}}_{i,j,k}^{n+1} = \mathbf{U}_{i,j,k}^{n} - \Delta t \sum_{r=1}^{N_s} \tilde{\beta}_r \mathbf{R}_{i,j,k}\left(\mathbf{U}^{(r)}\right) \tag{5.3.12}$$

Note that this does not involve much computational overhead, since the spatial residuals at the intermediate steps need to be stored anyway. Comparing the low order approximation to the high order approximation provides an estimate of the absolute error for the conservative variables in each grid cell:

$$\mathbf{Q}_{i,j,k}^{n+1} = \tilde{\mathbf{U}}_{i,j,k}^{n+1} - \mathbf{U}_{i,j,k}^{n+1} \tag{5.3.13}$$

Strictly speaking, this is an error estimate for the lower order approximation. However, the high-order approximation must have an error well below that value, which serves as an upper bound in this case. For each conservative variable, the maximum norm of the error is computed over the whole grid. These values are scaled by a typical value of the corresponding conservative variable in order to obtain a relative error estimate:

$$\mathbf{Q}_{\max} = \frac{1}{||\mathbf{U}^n||} \max_{i,j,k} \left| \mathbf{Q}_{i,j,k}^{n+1} \right| \tag{5.3.14}$$

If the relative error in any conservative variable exceeds a given threshold, the time step is rejected. The size of the time step is then reduced until the required accuracy is reached. In case a time step was successful, the size of the next time step is calculated by (Hairer & Wanner 1996)

$$\Delta t_{\text{new}} = \Delta t_{\text{old}} \left(\frac{\text{threshold}}{\max\left(\mathbf{Q}_{\max}\right)} \right)^{1/5}. \tag{5.3.15}$$

This ensures that the size of the time step may also grow, if the error is well below the threshold.

With the above algorithm, a suitable time step is determined automatically. However, experiments with low Mach number flows have shown that this method often reduces the time step to unnecessary small values. This behavior can be attributed to the appearance of very weak sound waves. The error estimator recognizes that they are better resolved with smaller time steps, although it is not desired to follow them. A possible solution to this problem is described by Hairer & Wanner (1996) where a *stiff* error estimator is proposed. However, this has not been implemented in LHC yet. Instead, the allowed range of automatically determined time steps is limited by some kind of CFL criterion, e.g. limiting the CFL_u number to values between 0.1 and 10. Numerical experiments have shown that this kind of treatment is usually sufficient.

5.4 Non-linear equations

The implicit time steppers introduced in this chapter require the solution of one or more non-linear algebraic systems per time step. In the engineering community, similar systems occur in the context of steady-state flow problems and are commonly solved by non-linear multigrid (FAS) or pseudo-transient methods. However, it has been shown in the author's diploma thesis (Miczek 2008) that it is very challenging to extend these methods for time-dependent problems in the low Mach number regime. Therefore, a different approach is used here which is presented in this section.

5.4.1 General properties and the treatment of scalar fields

Before solving the actual equations, their structure is analyzed in order to simplify the solution process. It has been shown in the previous section that the non-linear equations may always be written in the following form:

$$\mathbf{D}_{i,j,k}\left(\mathbf{U}\right) = \mathbf{R}_{i,j,k}\left(\mathbf{U}\right) + c_t \mathbf{U}_{i,j,k} + \mathbf{C}_{i,j,k} = 0 \tag{5.4.1}$$

Assuming that the vector of conserved quantities $\mathbf{U}$ contains N_{cons} variables per grid cell, the total number of equations (and unknowns) for a grid with $N_\xi \cdot N_\eta \cdot N_\zeta$ cells is given by

$$N_{\text{eqn}} = N_{\text{cons}} \cdot N_\xi \cdot N_\eta \cdot N_\zeta. \tag{5.4.2}$$

The spatial residual $\mathbf{R}$ couples each equation to the variables in the other grid cells. The dependencies may be analyzed by computing the Jacobian matrix $\frac{\partial \mathbf{D}}{\partial \mathbf{U}}$ of the defect $\mathbf{D}$. Since this matrix needs to be evaluated numerically in order to solve the equations (see next section), the number of dependencies per equation and thus the number of non-zero elements of the matrix is

of vital importance. If each equation would be coupled to the unknowns in all other grid cells, this Jacobian matrix would be dense, i.e. having N_{eqn}^2 entries. For reasonable three-dimensional grids, the amount of data would exceed the memory limits of any modern computer system. Fortunately, the coupling between the equations is very limited making the matrix sparse. This can be seen by analyzing the flux evaluation. The spatial residual at a certain grid cell depends on the flux differences between the interfaces in each coordinate direction. The corresponding interface values are reconstructed along one-dimensional lines of cell-averaged values from the two grid cells in positive and negative direction. Therefore, the spatial residual at a certain grid cell depends on a stencil of 13 grid cell (in three dimensions). The coupling between grid cells is always dense. Hence, for a general N_{dim}-dimensional grid, the number of non-zero entries in the Jacobian matrix can be written as

$$N_{\text{matrix}} = N_{\text{cons}}^2 \cdot (4 \cdot N_{\text{dim}} + 1) \cdot N_\xi \cdot N_\eta \cdot N_\zeta. \tag{5.4.3}$$

Assuming a memory requirement of 8 bytes (double precision float) per entry, the defect Jacobian matrix for a three dimensional grid of size 512^3 with 5 unknowns per grid cell would require 325 GB of memory. Although this is quite substantial, it is manageable on modern computer systems.

Unfortunately, a severe issue is encountered at this point. If a simulation requires a lot of scalar fields, the increased number of conservative variables leads to a quadratic growth of the size of the defect Jacobian matrix. Therefore, a new technique is proposed here in order to handle such situations efficiently. To start with, it can be seen from the Euler equations that scalar fields only appear in their corresponding advection equation. However, if scalar fields are used to describe chemical abundances, they may also influence the pressure via the equation of state. In this case, all conservative variables are fully coupled and the scalar field is called *active scalar*. Alternatively, when a scalar field is only advected with the fluid, but does not influence the hydrodynamic evolution, it is called *passive scalar*. Then the coupling between the passive scalar and the remaining conservative variables is only in one direction. The conservative variables influence the evolution of the passive scalar but not in opposite way.

Therefore, it is proposed to split the vector of conservative variables into a hydrodynamic part containing the active scalars and a passive scalar part:

$$\mathbf{U}_h = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho X_{a,1} \\ \rho X_{a,2} \\ \cdots \end{pmatrix} \qquad \mathbf{U}_p = \begin{pmatrix} \rho X_{p,1} \\ \rho X_{p,2} \\ \cdots \end{pmatrix} \tag{5.4.4}$$

The same splitting is applied to the discretized Euler equations. For the implicit time steppers, this results in a non-linear system of equations for the hydrodynamic part:

$$\mathbf{D}_h\left(\mathbf{U}_h\right) = \mathbf{R}_h\left(\mathbf{U}_h\right) + c_t \mathbf{U}_h + \mathbf{C}_h = 0. \tag{5.4.5}$$

Having a large amount of passive scalars N_p, this system is drastically reduced in size. The number of entries of the corresponding Jacobian matrix is

$$N_{\text{hmatrix}} = \left(N_{\text{cons}} - N_p\right)^2 \cdot (4 \cdot N_{\text{dim}} + 1) \cdot N_\xi \cdot N_\eta \cdot N_\zeta. \tag{5.4.6}$$

More importantly, the hydrodynamic part can be solved without any knowledge of passive scalars. *After* obtaining a solution for $\mathbf{U}_h$, the remaining equation for each passive scalar $\mathbf{U}_{p,i}$ reads

$$\mathbf{D}_{p,i}\left(\mathbf{U}_{p,i}\right) = \mathbf{R}_{p,i}\left(\mathbf{U}_h, \mathbf{U}_{p,i}\right) + c_t \mathbf{U}_{p,i} + \mathbf{C}_{p,i} = 0 \qquad i \in \{1, .., N_p\}. \tag{5.4.7}$$

Since passive scalars are usually not coupled to each other, these equations can be solved independently. Therefore each of these non-linear equations is scalar, having a Jacobian matrix of

size

$$N_{\text{pmatrix}} = (4 \cdot N_{\text{dim}} + 1) \cdot N_\xi \cdot N_\eta \cdot N_\zeta. \tag{5.4.8}$$

Hence, the total complexity of the system is greatly reduced by applying the splitting in a hydrodynamic and a passive scalar part. The computational overhead for all passive scalars scales *linearly* with their total number. Moreover, it should be stressed that the solution remains equal to the original system.

However, scalar fields are often used to describe chemical abundances which may have an impact on the hydrodynamic pressure. The scalar fields therefore need to be active because they enter the equation of state. However, in most cases it can be assumed that this influence is only very weak. The chemical abundances may then also be described by passive scalars. The hydrodynamic part of the equations is then solved under the assumption that the passive scalars remain constant on their previous values (i.e. at the last Runge-Kutta stage or at the last time step):

$$\mathbf{D}_h\left(\mathbf{U}_h\right) = \mathbf{R}_h\left(\mathbf{U}_h, \mathbf{U}_p^{\text{old}}\right) + c_t \mathbf{U}_h + \mathbf{C}_h = 0 \tag{5.4.9}$$

Having obtained the solution of the hydrodynamic part, the passive scalars are advected with the fluid as usual. Although the solution only approximates the solution of the original system, it can be shown by numerical experiments that the differences are usually *very* small, justifying this approach. For rare situations where this approximation causes significant errors, it may be sufficient to treat *some* abundances or thermodynamic properties like the mean molecular weight actively while keeping all other abundances passively. In LHC, any scalar value can either be treated actively or passively. In the latter case, the approximation described here is automatically used.

5.4.2 Newton-Raphson method

In order to solve the non-linear systems of equations, the so-called Newton-Raphson method is used here. This is an iterative process, starting from an initial guess $\mathbf{U}^0$ for the solution. The subsequent iterations are then calculated by

$$\mathbf{U}^{k+1} = \mathbf{U}^k - \left(\lambda \frac{\partial \mathbf{D}^k}{\partial \mathbf{U}}\right)^{-1} \mathbf{D}^k. \tag{5.4.10}$$

Instead of inverting the Jacobian matrix of the defect (evaluated at the previous iteration step), the following system of *linear* equations can be solved for the change $\Delta\mathbf{U} = \mathbf{U}^{k+1} - \mathbf{U}^k$ of the conservative variables in each iteration step:

$$\lambda \frac{\partial \mathbf{D}^k}{\partial \mathbf{U}}\ \Delta\mathbf{U} = -\mathbf{D}^k \tag{5.4.11}$$

Thus, the solution of the non-linear equations is reduced to the subsequent solution of several linear systems of equations. This Newton-Raphson process is known to converge quadratically to the solution under certain conditions. Most importantly, the initial guess must not be too far away from the solution. This is accomplished by choosing the initial guess to the solution from the last Runge-Kutta stage (ESDIRK) or to the last time step (backward Euler), because changes in one time step are expected to be moderate. Other requirements for a good convergence of the Newton-Raphson method are that the Jacobian matrix is evaluated accurately and that the linear systems are solved to machine precision. If one of the requirements for quadratic convergence is not met, the method may also converge but at a slower rate. In case the method does not converge, it may be helpful to set the scalar damping parameter λ to values > 1. Otherwise, this parameter is set to unity corresponding to an undamped scheme.

In order to monitor the convergence of the Newton-Raphson method, the L2-norm of the defect is computed after every iteration for each conservative variable $c \in \{\rho, \rho u, \rho v, ...\}$ separately:

$$\|\mathbf{D}\|^c = \sqrt{\sum_{i,j,k} \left(\mathbf{D}^c_{i,j,k}\right)^2}. \tag{5.4.12}$$

The resulting values would ideally decrease to zero during the iteration. However, from the non-linear equations it can be seen that they contain a constant vector $\mathbf{C}$:

$$\mathbf{D} = \mathbf{R}(\mathbf{U}) + c_t \mathbf{U} + \mathbf{C} = 0. \tag{5.4.13}$$

This addition is a source of numerical cancellation errors. Therefore, the convergence is always monitored in relation to the norm of the constant vector, i.e.

$$\frac{\|\mathbf{D}\|^c}{\|\mathbf{C}\|^c}. \tag{5.4.14}$$

The lowest values achievable in this scaled norm are related to the machine precision ($\approx 10^{-16}$ for double precision floating point values). Moreover, this scaling ensures that the convergence is monitored independently of the units in which the equations are formulated.

However, reaching machine precision is not always possible. Further numerical round-off errors in the discretization (e.g. by the low Mach number flux function) or in the solution process (e.g. linear solvers) may limit the achievable precision. Moreover, the convergence may be limited by slightly inexact defect Jacobian matrices or by an inaccurate solution of the linear systems. However, reaching machine precision is not always desirable at all. The discretization of the Euler equations contains an inherent discretization error, depending on the numerical schemes and on the grid resolution. Therefore, solving the non-linear systems to a higher precision than the discretization admits may be a waste of computational resources. This is underlined by the fact that a single Newton-Raphson iteration may be sufficient in order to maintain second-order accuracy in time (see e.g. Keppens et al. 1999). However, since higher-order time stepping schemes are used here and the discretization error is usually not known, it is hard to set a precise limit for the desired accuracy. If the ESDIRK error estimators are used, the accuracy of the solution of the non-linear system should be at least one order of magnitude below the ESDIRK tolerance. For this work, the Newton-Raphson iteration is usually terminated when the scaled norm drops below $\approx 10^{-7}$ in each conservative variable.

Besides monitoring the scaled norm at each iteration step, the convergence history is also analyzed automatically. If the dominant component of the scaled norm does not change for several iteration steps, the Newton-Raphson method is also terminated. Nevertheless, the simulation is continued in this case and the reasons for the stalled convergence are analyzed in the post processing. If they can be attributed to round-off errors and the iteration terminated at a fairly low level, the results may still be useful.

5.4.3 Computation of the defect Jacobian matrix

The Newton-Raphson method presented in the previous section relies on an accurate evaluation of Jacobian matrix of the defect. Due to the complexity of the numerical discretization, a direct derivation of formulas for the entries of this matrix is nearly impossible. Therefore, the evaluation of the defect Jacobian matrix is split into several logical parts which closely follow the algorithmic structure of the defect evaluation. Afterwards, the pieces are combined by applying the chain rule of differentiation.

It should be recalled that the numerical flux is computed along one-dimensional lines in curvilinear space. Therefore, the evaluation of the corresponding derivatives is also presented in a one-dimensional way in the following, but it can be applied to any coordinate direction. In the first step, the derivatives of interface reconstruction algorithms presented in Section 3.4 are computed. Since these algorithms are fairly simple, this can be done analytically. It should be noted that the left and right states approaching an interface may depend on the cell-averaged values of two grid cells in positive and negative direction. Therefore, the corresponding derivatives need to be computed and stored separately which results in

$$\frac{\partial \mathbf{U}^L_{i+1/2}}{\partial \mathbf{U}_l}, \qquad \frac{\partial \mathbf{U}^R_{i+1/2}}{\partial \mathbf{U}_l} \quad \text{with} \quad l \in \{i-1, i, i+1, i+2\}. \tag{5.4.15}$$

Note that each expression denotes a $N_{\text{cons}} \times N_{\text{cons}}$ matrix block. For example, the density $\rho^L_{i+1/2}$ may generally depend on the total energy $(\rho E)_i$ of the neighboring cell. This is not the case if the reconstruction is performed in conservative variables, where the matrix blocks become diagonal. For a reconstruction scheme in primitive variables, the matrix blocks can be evaluated by two successive matrix multiplications:

$$\frac{\partial \mathbf{U}^{LR}_{i+1/2}}{\partial \mathbf{U}_l} = \frac{\partial \mathbf{U}^{LR}_{i+1/2}}{\partial \mathbf{V}^{LR}_{i+1/2}} \cdot \frac{\partial \mathbf{V}^{LR}_{i+1/2}}{\partial \mathbf{V}_l} \cdot \frac{\partial \mathbf{V}_l}{\partial \mathbf{U}_l} \tag{5.4.16}$$

The last factor on the right hand side denotes the Jacobian matrix of the primitive variables with respect to the conservative variables in a certain grid cell (see Appendix C). This is multiplied by the diagonal matrix of the reconstruction algorithm in primitive variables. Finally, the Jacobian of the reverse transformation evaluated at the interface state is multiplied in order to determine the dependencies between the conservative variables. This procedure is the first example where the derivative of the algorithm "transformation to primitive variables, reconstruction, reverse transformation" is computed by splitting up the calculation in several logical parts which are combined by the chain rule.

In the next step, the partial derivatives of the numerical fluxes with respect to the interface values are evaluated, i.e.

$$\frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}^L_{i+1/2}}, \qquad \frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}^R_{i+1/2}}. \tag{5.4.17}$$

It should be recalled that most numerical flux functions defined in this work can be written as

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}\left(\mathbf{U}^L_{i+1/2}\right) + \mathbf{F}\left(\mathbf{U}^R_{i+1/2}\right) - D\ \left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\right)\right), \tag{5.4.18}$$

where D denotes the upwinding matrix which usually contains very complicated expressions. Therefore, the derivative of this flux with respect to the interface values is approximated by

$$\frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}^{LR}_{i+1/2}} = \frac{1}{2}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}^{LR}_{i+1/2}} \pm D\right). \tag{5.4.19}$$

Hence, the partial derivative of the upwinding matrix is neglected here. In addition to its algorithmic simplicity, numerical experiments have shown that this approximation can be quite useful. It is sufficiently accurate if the upwind state does not change significantly over one implicit time step. Otherwise, the resulting defect Jacobian matrix also gets inaccurate and the Newton-Raphson method does not converge anymore. For low Mach number flows, this is a good indicator that the advection is not resolved accurately in time. Therefore, the size of the time step has to be reduced.

Another method for computing the derivatives of the numerical flux is obtained by using finite differences. To this end, the conservative variables at each side of the interface are successively distorted by a small fraction (typically 10^{-7}). Modifying the c-th conservative variable (e.g. at the left hand side of the interface) then results in a distorted state $\tilde{\mathbf{U}}^L_c$ and a corresponding distorted numerical flux $\tilde{\mathbf{F}}_{i+1/2}$. The c-th column of the Jacobian matrix of the numerical flux can then be approximated by

$$\left(\frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}^L_{i+1/2}}\right)_c = \frac{\left(\tilde{\mathbf{F}}_{i+1/2} - \mathbf{F}_{i+1/2}\right)_r}{\tilde{\mathbf{U}}^L_c - \mathbf{U}^L}. \tag{5.4.20}$$

A similar expression can be obtained for the derivative with respect to the right approaching state. Although this method is quite general, it is only used for comparison with other evaluations of the derivative, because the finite difference always contains an approximation error which may be hard to control. In particular, if the velocity field is nearly zero in some regions, the above expression tends to become singular.

For the numerical fluxes of the AUSM$^+$ and VFRoe families, the corresponding derivatives are calculated directly by using a technique called automatic differentiation. For this, each expression in the implementation is augmented by the corresponding expression for its derivative combined by the chain rule of differentiation. However, since the derivatives are hard-coded in LHC, the method is not automatic anymore.

In the next step towards the evaluation of the defect Jacobian matrix, the derivatives of the numerical flux with respect to the cell-centered values are computed by using the chain rule again, i.e.

$$\frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}_l} = \frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}^L_{i+1/2}} \cdot \frac{\partial \mathbf{U}^L_{i+1/2}}{\partial \mathbf{U}_l} + \frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}^R_{i+1/2}} \cdot \frac{\partial \mathbf{U}^R_{i+1/2}}{\partial \mathbf{U}_l}. \tag{5.4.21}$$

The derivatives of the spatial residual along a certain coordinate line may then be assembled by

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_l} = \frac{1}{V_i}\left(\frac{\partial \mathbf{F}_{i+1/2}}{\partial \mathbf{U}_l} - \frac{\partial \mathbf{F}_{i-1/2}}{\partial \mathbf{U}_l}\right) - \frac{\partial \mathbf{S}_i}{\partial \mathbf{U}_l} \qquad l \in \{i, i \pm 1, i \pm 2\}. \tag{5.4.22}$$

This results in five $N_{\text{cons}} \times N_{\text{cons}}$ matrix blocks per dimension. Since the diagonal block ($l = i$) is shared by each dimension, the corresponding entries have to be added up in order to obtain the complete Jacobian matrix of the spatial residual. Finally, the defect Jacobian matrix reads

$$\frac{\partial \mathbf{D}_{i,j,k}}{\partial \mathbf{U}_{l,m,n}} = \frac{\partial \mathbf{R}_{i,j,k}}{\partial \mathbf{U}_{l,m,n}} + c_t \, \mathbb{1} \, \delta^{l,m,n}_{i,j,k}, \tag{5.4.23}$$

where only a single block has to be added to the diagonal block in each row.

5.5 Solvers for the linear systems

The Newton-Raphson method presented in the previous section reduced the solution of the non-linear equations for the implicit temporal discretization to the solution of several linear systems of the form

$$\lambda \frac{\partial \mathbf{D}^k}{\partial \mathbf{U}} \, \Delta \mathbf{U} = -\mathbf{D}^k. \tag{5.5.1}$$

It has also been shown that the matrix describing the linear system is very sparse. However, the inverse matrix or the LU-decomposition of the system is usually not sparse anymore. Since it is not possible (nor desirable) to store such a non-sparse matrix for moderately large three-dimensional grids, alternative methods are explored in this section. However, it is shown that the solution of the linear systems still consumes most of the computational time for low Mach number simulations. Therefore, essential requirements for these algorithms are robustness, efficiency and the possibility of being parallelized on modern distributed or shared-memory computer systems.

5.5.1 Direct solvers

The first class of algorithms are direct sparse matrix solvers. They actually perform a LU-decomposition (or similar) on the linear system, but ensure that the result is as sparse as possible. This is achieved by using complex reordering mechanisms described by graph theories. Such linear solvers are not implemented in LHC directly but are used though external software libraries. Specifically, interfaces to the PARDISO library (Schenk & Gärtner 2004), the SuperLU library (Li & Demmel 2003) and IBM's WSMP library have been implemented. However, numerical tests with all libraries have shown that the memory requirements during the solution process are quite substantial limiting their use to rather small grids. Moreover, the absolute performance is quite disappointing for the linear systems described here. Although all libraries are parallelized, the actual scaling on parallel systems is rather poor. Since the total memory and computing time requirements scale super-linearly with the grid size, direct solvers are not suited for production runs of low Mach number simulations. However, the libraries are very robust, providing solutions

up to machine accuracy. Therefore, they are at least on small grids an excellent tool to investigate situations where the Newton-Raphson method does not converge.

5.5.2 Iterative solvers

The other class of algorithms for large, sparse linear systems are iterative solvers. Starting from an initial guess, the solution is refined by several iteration steps until the desired accuracy is reached. Since the solution of the linear system describes a change in conservative variables, the iteration is always started from the zero vector. The convergence is measured by computing the linear residual,

$$\mathbf{L} = \lambda \frac{\partial \mathbf{D}^k}{\partial \mathbf{U}} \Delta \mathbf{U} + \mathbf{D}^k, \tag{5.5.2}$$

and calculating its L2-norm (separately for each conservative variable) relative to the norm of the right hand side:

$$\frac{\|\mathbf{L}\|}{\|\mathbf{D}^k\|} \tag{5.5.3}$$

Ideally, this scaled norm should reach machine accuracy ($\approx 10^{-16}$) in every component. However, the solution is only used to update the surrounding Newton-Raphson process. Even if machine accuracy is reached for the solution of the linear system, the non-linear error (the scaled defect norm) typically reduces by a factor of α only. Therefore, it is usually also sufficient to reach an accuracy of α in the linear process. Any further linear iterations only consume computing time without speeding up the non-linear convergence. Moreover, it should be noted that the defect Jacobian matrix contains an approximation for the flux derivatives. Therefore, the Newton-Raphson method rarely convergences quadratically but only linearly with typical convergence rates of 10^{-2}. It is thus sufficient in most cases to also solve the linear systems to an accuracy of 10^{-2}.

The disadvantage of iterative solvers is that they are usually not very robust. These algorithms usually do not guarantee that the iteration process convergences to the solution at all. In case the iteration converges, the rate of convergence (i.e. the reduction of the linear residual per iteration step) may be very close to unity, rendering the specific algorithm so inefficient that it is not usable for practical applications. Therefore, a suitable iterative linear solver has to be found specifically for each application. For the linear systems described here, an iterative solver should ideally provide a robust convergence with a rate independent of the specific flow field (e.g. the Mach number), the grid resolution and the details of the spatial discretization.

In the context of this thesis, many iterative solvers have been implemented and tested. Algorithms which totally failed to meet the requirements are not further explained and tested in this work. They include the Richardson method (e.g. Meister 2008), block symmetric Gauss-Seidel with over-relaxation (e.g. Meister 2008), line relaxation (e.g. Trottenberg et al. 2001), non-overlapping domain decomposition (e.g. Toselli & Widlund 2005), block ILU(0) (e.g. Meister 2008), sparse approximate inverses (e.g. Kallischko 2008) and induced dimension reduction IDR(s) (Sonneveld & van Gijzen 2008). All of these methods were tested in combination with suitable preconditioners (see next section) and multigrid but did not provide an acceptable rate of convergence for low Mach number flows.

More promising iterative linear solvers belong to the class of Krylov-subspace methods (see Meister (2008) for an overview). Specifically, the following solvers have been implemented in LHC: GMRES(r) (Saad & Schultz 1986; Meister 2008), Bi-CGSTAB(ℓ) (Sleijpen & Fokkema 1993), TFQMR (Freund 1993), QMRCGSTAB (Chan et al. 1994). Especially, the Bi-CGSTAB(ℓ) method usually shows very fast convergence for low Mach number flows. Only in rare situations, this method does not converge. However, the GMRES method seems to converge quite well for such rare events. Therefore, these two methods are usually applied successively. After a few steps with GMRES, the iteration is continued with Bi-CGSTAB(ℓ). Numerical tests of the

efficiency of this approach are presented in Section 5.6. A further beneficial property of Krylov-subspace methods is that they basically only require the computation of matrix-vector products and vector-vector scalar products during the solution process. Therefore, they are relatively easy to parallelize.

A drawback of the Krylov-subspace methods is that the convergence rate sometimes decreases as the grid resolution is increased. In order to circumvent such problems, a linear multigrid solver is implemented in LHC. Here, another iterative solver, which is called *smoother* in this context is used to obtain an approximate solution. Then the linear system is scaled down in size by a restriction matrix R. The coarser system corresponds to a discretization where the number grid cells is halved in each dimension. On this coarse mesh, a correction to the solution on the fine mesh is computed. This coarsening is recursively repeated until the solution of the system becomes trivial. Details on such multigrid methods can be found in the book by Trottenberg et al. (2001). Here, a so-called V-cycle and a Galerkin coarse grid discretization is used. The latter choice seems to be substantial for the linear systems discussed here in order to obtain good convergence rates for the overall method. Assuming that the linear system is written as $Ax + b = 0$, the pseudocode of the algorithm in use is given by:

$$
\begin{aligned}
&\text{function multigrid}(A, x_0, b) : \\
&\quad x := \text{smoother}\,(x_0) \\
&\quad \text{if coarsest mesh is reached then return } x \\
&\quad L^c := R\,(Ax + b) \\
&\quad A^c := RAR^T \\
&\quad x^c := \text{multigrid}(A^c, 0, L^c) \\
&\quad x := x + R^T x^c \\
&\quad x := \text{smoother}\,(x) \\
&\quad \text{return } x
\end{aligned}
\tag{5.5.4}
$$

The smoothing algorithm is usually set to the combination of GMRES and Bi-CGSTAB(ℓ) described above. For the restriction operator, the piecewise constant and the Kwak interpolation described by Mohr & Wienands (2004) are implemented. However, for most simulations the piecewise constant interpolation is used because it avoids a complicated handling of the boundaries of the grid. Note that the adjoined operator R^T is used to prolongate the solution from the coarse to the fine grid. The transfer operators R and R^T are both described by rectangular sparse matrices. In order to obtain the matrix A^c on the coarse grid, two multiplications between sparse matrices need to be performed. Suitable algorithms for this task were described by Pissanetzky (1984), which were parallelized for the use in LHC. The efficiency of the multigrid algorithm described here is demonstrated in Section 5.6.

5.5.3 Matrix preconditioning

The iterative solution of the linear systems consumes most of the computational resources and is especially challenging for low Mach number simulations. In order to support the iterative solvers, a technique called matrix preconditioning is applied to the linear system (see Meister (2008) for an overview on this topic). It should be noted that this technique is completely independent of the flux preconditioning technique that was described in Chapter 4. Assuming that the linear system is written in the form $Ax + b = 0$, two different kinds of matrix preconditioning can be devised. The first kind is left preconditioning, where the linear system is multiplied by a matrix P_L from the left:

$$P_L Ax + P_L b = 0 \tag{5.5.5}$$

As long as the preconditioning matrix P_L is invertible, the solution x of the linear system does not change. However, the preconditioned system may be more easy to solve with an iterative

method. The other kind of matrix preconditioning is the so called right preconditioning, where a matrix P_R and its inverse are multiplied to the original matrix A from the right hand side:

$$(AP_R)\left(P_R^{-1}x\right) + b = 0 \tag{5.5.6}$$

This defines a new linear system for a new unknown variable $\tilde{x}$:

$$(AP_R)\,\tilde{x} + b = 0, \tag{5.5.7}$$

After solving this system, the solution of the original system can be obtained by

$$x = P_R\tilde{x}. \tag{5.5.8}$$

The effect of the preconditioning can be seen if one of the preconditioning matrices is set to the inverse of A. In this case, the solution becomes trivial. The inverse of A is clearly not known and its computation would require more computational work than solving the unpreconditioned system. However, if the preconditioning matrices are set to a very rough approximation of the inverse, an iterative solver may achieve a better convergence rate. The success of such an approach depends on the specific linear system and the iterative solver in use. For the linear systems described in this work, only block diagonal preconditioning matrices are used. This has the advantage that the thus arising matrix-matrix and matrix-vector products are trivial to compute. The methods to choose the diagonal blocks of the preconditioning matrices are described in the following. The best method for accelerating the convergence of the iterative linear solvers may depend on the specific setup of a simulation and has to be determined by numerical experiments.

The first preconditioning method is applied from the left and is called block Jacobi method. The diagonal blocks in P_L are set to the inverse of the corresponding diagonal block of A. The diagonal blocks of the preconditioned system therefore become identity matrix blocks.

The next method belongs to the class of matrix scalings and can be applied as right or left preconditioning. To this end, the maximum absolute values of density, momentum and energy are determined over the whole grid. The diagonal blocks of the preconditioning matrix are all set to the same value. For left preconditioning each block is given by

$$\frac{1}{\Delta t}\begin{pmatrix} \rho_{\max} & & & & & & \\ & (\rho u)_{\max} & & & & & \\ & & (\rho u)_{\max} & & & & \\ & & & (\rho u)_{\max} & & & \\ & & & & (\rho E)_{\max} & & \\ & & & & & \rho_{\max} & \end{pmatrix}. \tag{5.5.9}$$

For right preconditioning the inverse of this matrix is used. The intention of this preconditioning is to scale the entries of the linear system such that the biggest components are of order one. This may be especially be useful if a simulation is performed in dimensionalized units. In this case, numerical values for the density and the energy may be orders of magnitude different in size. The preconditioner scales these values to the same order of magnitude during the solution of the linear system, which may be advantageous for the overall convergence rate. This kind of preconditioning is also referred as non-dimensional preconditioning in the following.

The last class of preconditioners described here are transformations to different sets of variables. For left preconditioning, the block diagonal entries may be set to the Jacobian matrix $\frac{\partial \mathbf{V}}{\partial \mathbf{U}}\big|_{i,j,k}$ of the transformation from conservative variables $\mathbf{U}$ to primitive variables $\mathbf{V}$. These Jacobians are evaluated at the grid point which corresponds to a certain row in the preconditioning matrix. For right preconditioning, the Jacobian of the inverse transformation $\frac{\partial \mathbf{U}}{\partial \mathbf{V}}\big|_{i,j,k}$ can be used. Moreover, transformations to primitive variables containing the temperature instead of the pressure and to so-called symmetrizing variables may be used (see Appendix C). If such a transformation is applied from the left, the preconditioned matrix system corresponds to system which is formulated

in these variables. For right preconditioning, the independent variables in x are transformed during the solution of the linear system.

The left preconditioners for different sets of variables may be further improved by a technique developed in the context of this thesis which is especially useful for low Mach number flows. It was shown in Section 4.3.1 that the Roe-Turkel flux preconditioning was originally developed in order to accelerate the convergence of steady-state flow problems. This is achieved by equalizing the eigenvalues of the preconditioned flux. Therefore, it is proposed here to use the Roe-Turkel flux preconditioner as matrix preconditioning. The diagonal blocks of a left preconditioning matrix are thus filled with

$$P_{\mathbf{V},\text{Turkel}} \left.\frac{\partial \mathbf{V}}{\partial \mathbf{U}}\right|_{i,j,k}. \tag{5.5.10}$$

Similar preconditioning methods may be applied with transformations to other sets of variables. It must be stressed that this kind of matrix preconditioning is independent of the numerical flux function in use and does not change the discretization. However, it is shown in the next section that the convergence of the iterative linear solvers is accelerated for low Mach number flow problems.

5.6 Validation

5.6.1 Accuracy

In order to investigate the behavior of the numerical discretization and especially the influence of different time marching schemes, a very simple test problem is constructed here, where the analytic solution is approximately known. It consists of a one-dimensional sound wave traveling to the right and an advective wave traveling to the left direction. Note that all quantities are assumed to be non-dimensional in the following.

In Section 2.6.1, the asymptotic theory for linear sound waves was presented for a fluid at rest. The theory holds for very small pressure fluctuations which are measured in terms of the reference Mach number M_r here. It was shown that an arbitrary profile for the velocity potential φ fulfills the wave equation 2.6.7 and may thus travel in positive or negative direction with the speed of sound $\pm c/M_\text{r}$. Here, a Gaussian profile traveling to the right is chosen. The analytic solution of the wave equation therefore reads

$$\varphi\left(x,t\right) = \frac{1}{M_\text{r}} \exp\left(-\left(x - x_0 - \frac{ct}{M_\text{r}}\right)^2\right). \tag{5.6.1}$$

The first order (with respect to the reference Mach number M_r) velocity, pressure and density fluctuations are then given by

$$u_1 = \frac{\partial \varphi}{\partial x}, \qquad p_1 = -\rho_0 M_\text{r}^2 \frac{\partial \varphi}{\partial t}, \qquad \rho_{1,\text{sound}} = \frac{p_1}{c^2}. \tag{5.6.2}$$

Note that φ was normalized such that the pressure and density fluctuations are of order one.

In order to construct an advective wave, it should be noted that arbitrary density profiles are a solution to the compressible Euler equations for a zero velocity field and a constant pressure field. Hence, the first order density field is disturbed by a Gaussian function located at a position different from that of sound wave, i.e.

$$\rho_{1,\text{advection}} = \exp\left(-\left(x - x_1\right)^2\right). \tag{5.6.3}$$

It must be emphasized that the corresponding pressure field is not modified. Since the fluid is at rest, the density fluctuation is a stationary solution and does not propagate with time.

For the test problem which is constructed here, the fluctuations of the sound wave and the advective density fluctuation are superposed. In the low Mach number regime ($M_\mathrm{r} \ll 1$), it is expected that the sound wave passes the advective density fluctuation without being modified, because the incompressible flow field decouples from the sound waves (see Section 2.6.3). The superposition of the two waves is thus considered as an analytic solution of full problem for small Mach numbers.

So far, the analytic solution of the test problem was constructed for a fluid at rest. However, since the compressible Euler equations are Galilean invariant, the frame of reference may be changed such the fluid globally moves to the left direction with a Mach number M_u. The advective density fluctuation thus moves with the corresponding velocity $-M_u/M_\mathrm{r}c$. The sound wave then propagates at the speed $c/M_\mathrm{r} - M_u/M_\mathrm{r}c$. Adding the velocity offset, the analytic solution in the moving frame of reference is given by replacing the x coordinate by

$$x \to x + \frac{M_u}{M_\mathrm{r}} ct \tag{5.6.4}$$

in the above expressions. Note that the Mach number M_u does not need to be small, although only values ≤ 0.1 are considered here.

The final one-dimensional setup is denoted as sound-advection test problem in the following. The analytic solution for the velocity field is given by

$$u(x,t) = \frac{M_u}{M_\mathrm{r}} c - 2x' \exp\left(-\left(x'\right)^2\right), \tag{5.6.5}$$

where the scaled coordinate

$$x' = x - x_0 - \frac{ct}{M_\mathrm{r}} + \frac{M_u}{M_\mathrm{r}} ct \tag{5.6.6}$$

has been introduced. With the first order pressure fluctuation

$$p_1(x,t) = -2\rho_0\, c\, x' \exp\left(-\left(x'\right)^2\right), \tag{5.6.7}$$

the total pressure field and the density field are given by the following expressions:

$$p(x,t) = p_0 + M_\mathrm{r} p_1(x,t) \tag{5.6.8}$$

$$\rho(x,t) = \rho_0 + M_\mathrm{r} \left(\frac{p_1(x,t)}{c^2} + \exp\left(-\left(x - x_1 + M_u ct\right)^2\right)\right) \tag{5.6.9}$$

For numerical simulations with LHC, a physical domain of size $[0, 20]$ containing an ideal gas is chosen. The sound wave is initially located at $x_0 = 5$ while the advective wave is located at $x_1 = 15$. The non-dimensional density and pressure offsets are set to $\rho_0 = 1$ and $p_0 = 1/\gamma$ such that the non-dimensional speed of sound becomes $c = 1$. The reference Mach number is chosen as $M_\mathrm{r} = 10^{-6}$ to ensure that the linear theory of sound applies to a very good approximation and that the advective wave decouples from the sound wave. The physical domain is discretized by N_x grid cells, which are initialized with the analytic solution at $t = 0$. The boundaries are treated by the far-field flux boundary conditions, which are expected to be perfectly transmissive for both kinds of waves. The initial density and pressure fluctuations are plotted in Figure 5.2.

At two instants in time, the numerical solution computed with LHC is compared with the analytic solution. At the first time step of interest,

$$t_1 = \frac{+10}{c/M_\mathrm{r} - M_u/M_\mathrm{r}c}, \tag{5.6.10}$$

the sound wave has propagated 10 non-dimensional units to the right. Here, the error in the first order pressure fluctuations are computed by

$$\Delta p_1 = \max\left(\left| \frac{p - p_0}{M_\mathrm{r}} - p_{1,\mathrm{analytic}} \right| \right). \tag{5.6.11}$$

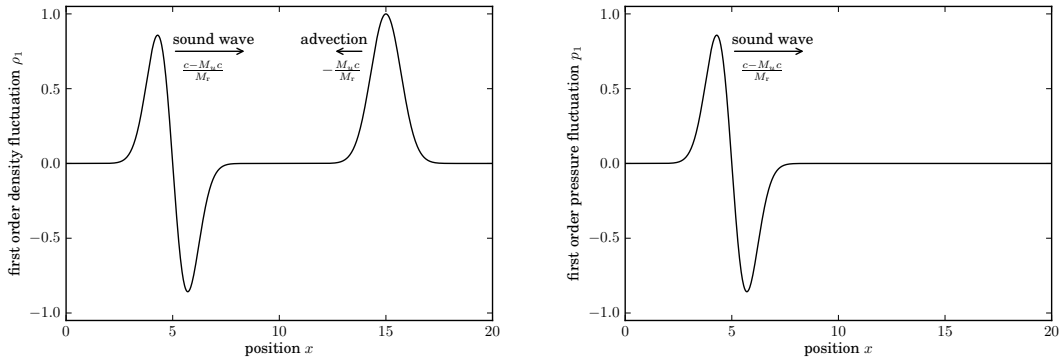


Figure 5.2: Initial conditions for the sound-advection test problem

The second time step of interest,

$$t_2 = \frac{-10}{-M_u/M_\mathrm{r} c}, \tag{5.6.12}$$

denotes the point where the advective wave has traveled -10 non-dimensional units to the left. The first order density fluctuations are compared to the analytic solution here, i.e.

$$\Delta\rho_1 = \max\left(\left|\frac{\rho - \rho_0}{M_\mathrm{r}} - \rho_{1,\mathrm{analytic}}\right|\right). \tag{5.6.13}$$

Both error estimates do not need to be further normalized as the non-dimensional first order fluctuations are of order unity.

In the following, the numerical results of different time marching schemes are compared for $M_u = 0.1$ and $M_u = 0.001$. All simulations use Roe's approximate Riemann solver combined with a linear interface reconstruction which is second order accurate in space. At first, numerical solutions are computed at different grid resolutions N_x using the *explicit* 3rd order Runge-Kutta method presented in Section 5.1. The time step is chosen according to the acoustic CFL criterion with $\mathrm{CFL}_{uc} = 0.5$ here. Figure 5.3 shows the computed first order pressure and density fluctuations at $t = t_1$ and $t = t_2$ respectively (with $N_x = 256$ and $M_u = 0.1$). It can be seen that the advective and the sound wave are both captured accurately. In the right panel the sound wave has already left the domain, showing that the boundary conditions work very well as the do not reflect any part of the sound wave. The maximum deviations from the analytic solution are shown in the following table for different grid resolutions for $M_u = 0.1$:

N_x	$N_\mathrm{steps}(t_2)$	$\Delta p_1(t_1)$	conv.	$\Delta\rho_1(t_2)$	conv.
128	1408	$2.065 \cdot 10^{-1}$	–	$7.377 \cdot 10^{-2}$	–
256	2816	$5.995 \cdot 10^{-2}$	0.29	$2.008 \cdot 10^{-2}$	0.27
512	5632	$1.529 \cdot 10^{-2}$	0.25	$4.995 \cdot 10^{-3}$	0.25
1024	11264	$3.818 \cdot 10^{-3}$	0.25	$1.244 \cdot 10^{-3}$	0.25

(5.6.14)

It can clearly be seen that the error decreases by a factor of $\approx 1/4$ every time the number of grid cells is doubled. This is fully consistent with the expectation for a second-order spatial discretization. However, this also shows that the error is not dominated by the temporal discretization here. Thus, taking smaller time steps would not increase the accuracy in this case. The number of time steps N_steps necessary to reach $t = t_2$ is also listed in the above table. Each time the grid spacing is halved the number of steps is doubled, which is a result of the acoustic CFL criterion for the time step. Assuming that the computational work to calculate one time step scales linearly with the number of grid cells, the overall work increases thus by a factor of four as the resolution

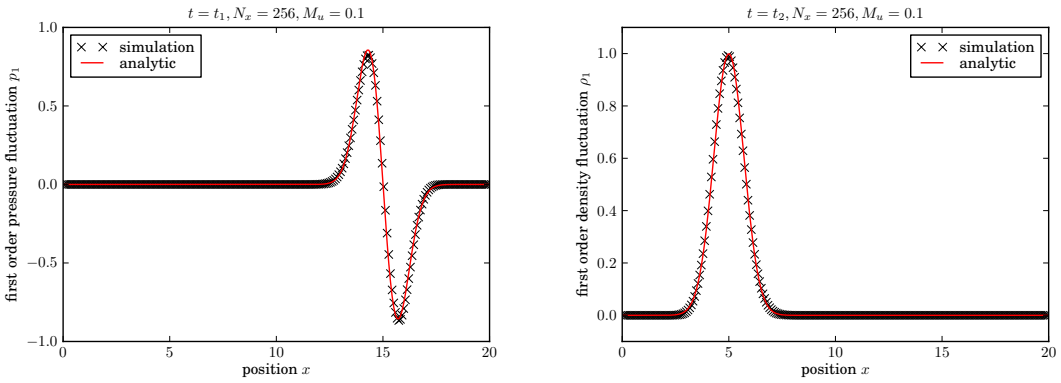


Figure 5.3: First order pressure and density fluctuations for 3rd order explicit time stepping

is increased by a factor of two. In the next set of simulations the advective Mach number is decreased to $M_u = 0.001$. The results are shown in the following table:

N_x	$N_{\text{steps}}(t_2)$	$\Delta p_1(t_1)$	conv.	$\Delta \rho_1(t_2)$	conv.
128	128128	$2.068 \cdot 10^{-1}$	–	$7.377 \cdot 10^{-2}$	–
256	256256	$6.032 \cdot 10^{-2}$	0.29	$2.008 \cdot 10^{-2}$	0.27
512	512512	$1.530 \cdot 10^{-2}$	0.25	$4.983 \cdot 10^{-3}$	0.25
1024	1025024	$3.818 \cdot 10^{-3}$	0.25	$1.271 \cdot 10^{-3}$	0.26

(5.6.15)

It can be seen that the errors are nearly equal to the errors obtained for $M_u = 0.1$. However, the number of time steps needed to advect the density fluctuation to $t = t_2$ has nearly increased by a factor of 100. This is a result of the stringent CFL criterion for the time step which follows the time scale of the propagation of the sound wave. This criterion has also be fulfilled after the sound wave has left the domain in order to ensure numerical stability. For example, the simulation with $M_u = 0.001$ and $N_x = 1024$ reaches the time $t = t_1$ after only 1026 steps. Shortly after that, the sound wave leaves the domain and the remaining one million steps are needed just to advect the density fluctuation. The overall computing time for this simple setup is already several hours on a modern desktop computer. This again demonstrates that explicit time stepping is highly inefficient for low Mach number advective flows. Hence, multidimensional simulations at such Mach numbers quickly become impractical even on large supercomputer systems.

In order to circumvent the stringent time step limitations, implicit temporal discretizations were presented in this chapter and are tested in the following. At first, the third-order ESDIRK34 scheme is used to simulate the sound-advection problem with the same parameters as above. Even the time step is chosen according to the *acoustic* CFL criterion with $\text{CFL}_{uc} = 0.5$, although this is not required to ensure numerical stability. The maximum errors for $M_u = 0.1$ result in

N_x	$N_{\text{steps}}(t_2)$	$\Delta p_1(t_1)$	conv.	$\Delta \rho_1(t_2)$	conv.
128	1408	$2.058 \cdot 10^{-1}$	–	$7.377 \cdot 10^{-2}$	–
256	2816	$5.987 \cdot 10^{-2}$	0.29	$2.008 \cdot 10^{-2}$	0.27
512	5632	$1.528 \cdot 10^{-2}$	0.26	$4.995 \cdot 10^{-3}$	0.25
1024	11264	$3.817 \cdot 10^{-3}$	0.25	$1.244 \cdot 10^{-3}$	0.25

(5.6.16)

whereas the simulations with $M_u = 0.001$ result in

N_x	$N_{\text{steps}}(t_2)$	$\Delta p_1(t_1)$	conv.	$\Delta \rho_1(t_2)$	conv.
128	128128	$2.054 \cdot 10^{-1}$	–	$7.377 \cdot 10^{-2}$	–
256	256256	$6.010 \cdot 10^{-2}$	0.29	$2.008 \cdot 10^{-2}$	0.27
512	512512	$1.528 \cdot 10^{-2}$	0.25	$4.996 \cdot 10^{-3}$	0.25

(5.6.17)

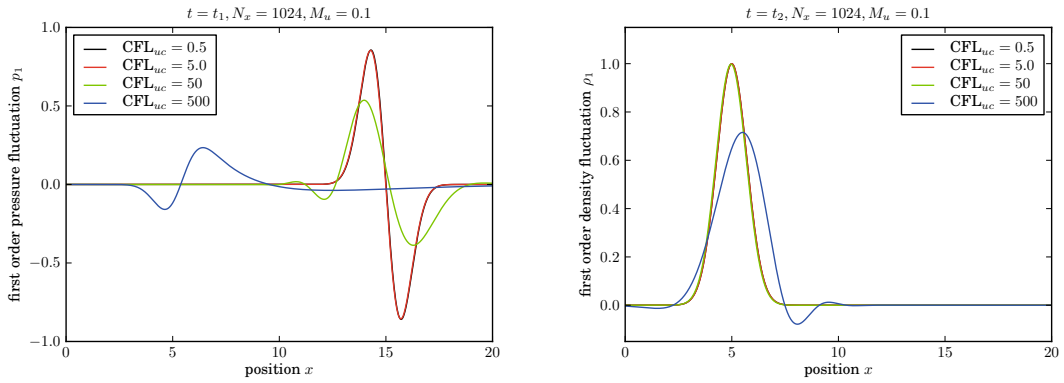


Figure 5.4: ESDIRK34 time stepping at different acoustic CFL numbers

These tables show that the errors are nearly equal to the errors obtained with explicit time stepping. Thus, the implicit time marching scheme is able to resolve acoustic and advective waves with the same accuracy as the explicit scheme if the same size of the time step is used. However, these implicit calculations are even more inefficient for low Mach number flows, because the computational work per time step is much higher. The simulation with $N_x = 1024$ and $M_u = 0.001$ could therefore not be performed in a reasonable amount of computing time and is thus omitted here.

More interesting is the behavior of the implicit time marching schemes as the size of the time step is increased. For this, the grid resolution is fixed to $N_x = 1024$ and the acoustic CFL number is successively increased by factors of 10. For $M_u = 0.1$ the results obtained with the ESDIRK34 time stepper are shown in Figure 5.4. It can be seen that neither the acoustic wave at $t = t_1$ shown in the left panel nor the advective wave at $t = t_2$ shown in the right panel is modified as the CFL_{uc} number is increased from 0.5 to 5. For $\mathrm{CFL}_{uc} = 50$, the sound wave gets heavily damped while the advective wave still looks reasonable accurate. Note that the size of the time step corresponds to an *advective* CFL number of 5 in this case. If the time step is further increased by a factor of 10, the advective wave is also strongly damped, while the acoustic wave is not even propagated to the right position anymore. The detailed results of this test with the ESDIRK34 scheme at $M_u = 0.1$ are shown in the following table:

CFL_{uc}	CFL_u	$N_{\mathrm{steps}}(t_2)$	$\Delta p_1(t_1)$	$\Delta \rho_1(t_2)$
$0.5 \cdot 10^3$	$0.5 \cdot 10^2$	11	$8.954 \cdot 10^{-1}$	$3.110 \cdot 10^{-1}$
$0.5 \cdot 10^2$	$0.5 \cdot 10^1$	113	$6.076 \cdot 10^{-1}$	$2.361 \cdot 10^{-3}$
$0.5 \cdot 10^1$	$0.5 \cdot 10^0$	1126	$5.826 \cdot 10^{-3}$	$1.243 \cdot 10^{-3}$
$0.5 \cdot 10^0$	$0.5 \cdot 10^{-1}$	11264	$3.817 \cdot 10^{-3}$	$1.244 \cdot 10^{-3}$

(5.6.18)

Performing similar simulations at with an advective Mach number $M_u = 0.001$ and the ESDIRK34 scheme results in the following data:

CFL_{uc}	CFL_u	$N_{\mathrm{steps}}(t_2)$	$\Delta p_1(t_1)$	$\Delta \rho_1(t_2)$
$0.5 \cdot 10^5$	$0.5 \cdot 10^2$	10	n/a	$3.412 \cdot 10^{-1}$
$0.5 \cdot 10^4$	$0.5 \cdot 10^1$	103	n/a	$2.943 \cdot 10^{-3}$
$0.5 \cdot 10^3$	$0.5 \cdot 10^0$	1025	$8.834 \cdot 10^{-1}$	$1.244 \cdot 10^{-3}$
$0.5 \cdot 10^2$	$0.5 \cdot 10^{-1}$	10250	$6.882 \cdot 10^{-1}$	$1.244 \cdot 10^{-3}$
$0.5 \cdot 10^1$	$0.5 \cdot 10^{-2}$	102502	$8.644 \cdot 10^{-3}$	$1.244 \cdot 10^{-3}$

(5.6.19)

Here, the acoustic CFL number is even increased to $0.5 \cdot 10^5$ which corresponds to the same maximum advective CFL number of $0.5 \cdot 10^2$. It can be seen that the advective wave is accurately

resolved at both Mach numbers for $\mathrm{CFL}_u \leq 5$, whereas an accurate treatment of the sound wave requires $\mathrm{CFL}_{uc} \leq 5$. Note that it is not possible to determine the error of the sound wave at $t = t_1$ for $M_u = 0.001$ and $\mathrm{CFL}_{uc} >= 0.5 \cdot 10^4$, because the sound wave leaves the computational domain within the first time step. However, the behavior of the advective wave is not influenced by that, emphasizing that the boundary conditions work very well in this case. Beside that, it should be noted that the ESDIRK34 time stepper is third order accurate in time. Reducing the time step by a factor of 10 should therefore decrease the errors by a factor of 1000. This behavior is not observed in the simulations presented here, because the error stagnates at some early point where it gets dominated by the spatial discretization error.

The sound-advection test problem was also simulated with different implicit time marching schemes. For the backward Euler method with $N_x = 1024$ and $M_u = 0.001$ the errors result in:

CFL_{uc}	CFL_u	$N_{\mathrm{steps}}(t_2)$	$\Delta p_1(t_1)$	$\Delta \rho_1(t_2)$
$0.5 \cdot 10^5$	$0.5 \cdot 10^2$	10	n/a	$7.774 \cdot 10^{-1}$
$0.5 \cdot 10^4$	$0.5 \cdot 10^1$	103	n/a	$4.185 \cdot 10^{-1}$
$0.5 \cdot 10^3$	$0.5 \cdot 10^0$	1025	$8.651 \cdot 10^{-1}$	$8.532 \cdot 10^{-2}$
$0.5 \cdot 10^2$	$0.5 \cdot 10^{-1}$	10250	$8.632 \cdot 10^{-1}$	$9.838 \cdot 10^{-3}$
$0.5 \cdot 10^1$	$0.5 \cdot 10^{-2}$	102502	$6.409 \cdot 10^{-1}$	$1.782 \cdot 10^{-3}$

(5.6.20)

It can be seen that the sound wave is not well resolved in any simulation. In order to get accurate results for the advective wave, very small time steps with acoustic CFL numbers of order one are required here. This shows that although the backward Euler method is very stable, it is also very inaccurate as it is only first order in time. Contrary to that the ESDIRK58 method is fifth order accurate. Simulations with this time marching method at $M_u = 0.001$ result in the following errors:

CFL_{uc}	CFL_u	$N_{\mathrm{steps}}(t_2)$	$\Delta p_1(t_1)$	$\Delta \rho_1(t_2)$
$0.5 \cdot 10^5$	$0.5 \cdot 10^2$	10	n/a	$3.811 \cdot 10^{-2}$
$0.5 \cdot 10^4$	$0.5 \cdot 10^1$	103	n/a	$1.250 \cdot 10^{-3}$
$0.5 \cdot 10^3$	$0.5 \cdot 10^0$	1025	$9.202 \cdot 10^{-1}$	$1.244 \cdot 10^{-3}$
$0.5 \cdot 10^2$	$0.5 \cdot 10^{-1}$	10250	$1.207 \cdot 10^{-1}$	$1.244 \cdot 10^{-3}$
$0.5 \cdot 10^1$	$0.5 \cdot 10^{-2}$	102502	$3.831 \cdot 10^{-3}$	$1.244 \cdot 10^{-3}$

(5.6.21)

For an advective CFL number of 50, the error of the advective wave is approximately ten times smaller than in the computations with the ESDIRK34 scheme, whereas the sound wave is only slightly better resolved here. However, with the ESDIRK58 method the errors become quickly dominated by the spatial discretization error. Therefore, the ESDIRK58 scheme provides only slightly better results than the ESDIRK34 scheme for advective CFL number of order unity. Moreover, the fifth order method needs eight Runge-Kutta stages whereas the third order method only needs four of them. The computational work per ESDIRK58 time step is thus approximately twice the work needed for an ESDIRK34 time step. Therefore, it is usually preferred to use the ESDIRK34 method for practical computations.

In summary, it can be stated that the implicit time marching schemes provide accurate solutions for the advective waves for CFL_u numbers of order unity. The size of the time step is independent of the Mach number of the problem. Sound waves are not well resolved with such large time steps by the implicit schemes. However, the propagation of sound is usually not of particular interest for low Mach number simulations as discussed in Section 2.6. Therefore, implicit time marching schemes are an excellent way to capture the incompressible flow regime of low Mach number flows. The under-resolved sound waves get damped out very quickly and do not influence the incompressible flow. However, it has also been shown by the sound-advection test that implicit methods are able to resolve waves on any time scale, as long as the dynamics of the wave of interest is temporally resolved.

5.6.2 Efficiency

Implicit time stepping methods admit time steps based on an advective CFL criterion and are therefore well suited to resolve the incompressible regime of low Mach number flows. However, the computation of each implicit time step is computationally very expensive, because several non-linear and linear algebraic systems need to be solved in each step. In contrast, explicit time stepping methods are computationally very cheap but require very small time steps based on an acoustic CFL criterion in order to maintain numerical stability. Therefore, the question arises whether implicit time stepping methods are really superior in terms of computational efficiency.

Here, the performance of an implicit method is analyzed in detail for a simple test case. To this end, the Gresho vortex defined in Section 4.1.1 is simulated at different Mach numbers and grid resolutions. First, the third-order explicit Runge-Kutta method (Equation 5.1.2) with an *acoustic* CFL number of 0.5 is used. The numerical flux is computed with Roe's approximate Riemann solver. Due to the excessive numerical dissipation in the low Mach number regime, the results are very inaccurate. Using a low Mach number numerical flux function would require an even more stringent time step limitation (see Section 5.1), where the size of the time steps is reduced by a factor of the Mach number. Therefore, the efficiency of the explicit time marching scheme would decrease by the same amount in this case. However, since explicit time stepping in combination with Roe's approximate Riemann solver is very common for many existing hydrodynamic codes, this setup will serve as a reference case in terms of computational performance.

Next, the Gresho vortex is simulated with the implicit ESDIRK34 scheme. The time step is chosen according to an *advective* CFL number of 0.5. The numerical flux is computed with the Roe-Lowmach scheme here, which is much less dissipative then the original Roe method. Therefore, the results of the implicit calculations are much more accurate than the results from the explicit method. The arising non-linear systems are solved by the Newton-Raphson method with a relative accuracy of 10^{-6}. In order to solve the linear equations, four different strategies are explored in the following:

- direct solver PARDISO
- BiCGSTAB(5) with an accuracy goal of 10^{-2}
- BiCGSTAB(5) with an accuracy goal of 10^{-2} and block Jacobi preconditioning
- Multigrid (V11 cycle) with block Jacobian preconditioning; smoothing: GMRES(20) followed by BiCGSTAB(5), both with a relative accuracy goal of 0.25 on each multigrid level

The simulations are performed on a single processor on a dedicated machine. For each setup, the grid resolution is successively doubled in both directions, ranging from 64×64 to 1024×1024. Each simulation is stopped after a few time steps where the physical time has evolved to t_{phys}. The computational performance is then measured by the ratio of the computational time t_{comp} (i.e. the wall-clock time) needed for the simulation to the physical time:

$$r = \frac{t_{\text{comp}}}{t_{\text{phys}}} \tag{5.6.22}$$

The values for this number have no significance in an absolute sense, because they strongly depend on the physical time scale and the units chosen for the simulation. Only relative changes between different simulations provide further insight. Therefore, the ratio r is always compared to the value of the corresponding simulation of the explicit time stepper (at the same grid resolution and Mach number):

$$s = \frac{r_{\text{explicit}}}{r} \tag{5.6.23}$$

This quantity describes the computational speedup which is gained by using the implicit time marching scheme. Values below unity denote the case when explicit time stepping would have been more efficient (although less accurate here), whereas values above unity indicate the increase

in computational efficiency achieved by the implicit method. Another interesting quantity is the value of r compared to the corresponding value where the grid resolution is halved in each direction (at the same Mach number and with the same time stepping method):

$$c = \frac{r}{r_{\text{half res}}} \tag{5.6.24}$$

Doubling the grid resolution in each direction, the amount of work should increase by a factor of four. However, due to the smaller grid spacing, the time step determined by a CFL-like criterion halves. Therefore, the ratio r should ideally increase by a factor of eight in total. In the case $c > 8$, the computational efficiency deteriorates as the grid gets finer, indicating that the overall method is not suited for practical applications.

The results of all simulations are presented in Table 5.6.2. For the explicit time stepping scheme, it can be seen that the ratio of computational to physical time increases by factors of 10 as the Mach number is decreased. For this reason, the use of explicit schemes is highly undesirable for low Mach number flows. However, all values of c are close to eight. Thus, the efficiency does not break down further as the grid resolution is increased. For the implicit time stepping with the PARDISO solver, the ratio r of computational to physical time does not depend on the Mach number for a given grid resolution. This method becomes superior to explicit time stepping for Mach numbers below 0.01 in this test. However, the speedup decreases as the grid resolution is increased. The values for c are significantly above eight, indicating that the efficiency breaks down for larger grids. In fact, the simulations could only be performed up to a grid resolution of 256×256. For larger grids, the internal memory requirements of the PARDISO solver exceeded the 12GB memory limit of the machine on which the simulations were performed. This is a result of the fact that the LU decomposition of a sparse matrix is generally not sparse anymore. The use of direct solvers is therefore limited to very small grids. Due to this poor efficiency, direct solvers are inapplicable for three-dimensional simulations and are only used for diagnostic purposes in LHC.

From the results presented in the table, it can be seen that BiCGSTAB(5) without matrix preconditioning is also not suited for low Mach number simulations. The ratio r of computing to physical time usually increases by factors larger than 10 as the Mach number is decreased by factors of 10. The computational speedup shows that this method is not significantly more efficient than explicit time stepping. Moreover, in some simulations at $M = 10^{-4}$ the linear solver did not converge at all. However, the situation changes drastically if block Jacobi matrix preconditioning is used. For a given grid resolution, the ratio of computational to physical time is nearly independent of the Mach number. For $M = 10^{-2}$ the method starts to become superior to explicit time stepping and it achieves large speedups for very low Mach numbers. Only the scaling of the computational work with the grid resolution is not satisfactory. The linear solver needs successively more iterations to reach the desired accuracy as the grid is refined, deteriorating the overall computational efficiency.

Finally, the results of the implicit time marching scheme with the multigrid solver are evaluated. From Table 5.6.2, it can be seen that the ratio r of computational to physical time stays nearly constant as the Mach number is decreased, which is highly desirable. Moreover, the speedup compared to explicit time stepping nearly increases by factors of 10 as the Mach number is decreased. The implicit method is superior even for moderately low Mach numbers in the range between $M = 0.1$ and $M = 0.01$. Beside that, it can be seen that the quantity c is mostly *below* eight. Thus, the iterative linear solver needs *less* steps to converge at higher grid resolutions. However, it must be emphasized that this curious behavior is most likely related to the fact that the Gresho vortex is a very simple test problem. As the grid resolution is increased, no additional flow structures on very small scales need to be resolved. The solution consists of successively lower wave numbers compared to the grid spacing, which can efficiently be solved on the coarser levels of the multigrid procedure, whereas the smoothing on the fine grid levels becomes more and more trivial.

In total, this test of the computational efficiency of implicit time stepping methods has shown

that they can indeed be superior to explicit time stepping methods. It should also be emphasized that the implicit calculations provide much more accurate results due to the use of the Roe-Lowmach scheme. Using this numerical flux for explicit calculations would require even smaller time steps, such that the speedup s of the implicit method would further increase by a factor $1/M$. However, it has also been shown that implicit time stepping strongly depends on a suitable combination of matrix preconditioning and a proper linear solver. The multigrid/block Jacobi combination presented here may not be optimal for all kinds of flow structures. According to the author's experience, it is always necessary to select a suitable matrix preconditioning method and tune the accuracy goals for the multigrid smoother for each problem setup differently in order to gain optimal performance. For very complicated flow problems it is sometimes not possible to find suitable parameters to reach the same computational efficiency as presented in this section. However, the tests of the Gresho vortex showed that implicit time stepping methods can be highly efficient and are therefore well suited for low Mach number flows.

time stepper	3rd order explicit			ESDIRK34			ESDIRK34			ESDIRK34			ESDIRK34		
linear solver	-			PARDISO			BiCGSTAB(5)			BiCGSTAB(5)			Multigrid		
matrix preconditioning	-			-			-			block Jacobi			block Jacobi		
parameters	r	s	c	r	s	c	r	s	c	r	s	c	r	s	c
$M = 10^{-1}, N = 64$	$1.18 \cdot 10^2$	1.0	-	$1.34 \cdot 10^3$	0.1	-	$1.59 \cdot 10^2$	0.7	-	$2.16 \cdot 10^2$	0.5	-	$2.09 \cdot 10^2$	0.6	-
$M = 10^{-2}, N = 64$	$1.08 \cdot 10^3$	1.0	-	$1.34 \cdot 10^3$	0.8	-	$5.67 \cdot 10^2$	1.9	-	$4.74 \cdot 10^2$	2.3	-	$3.63 \cdot 10^2$	3.0	-
$M = 10^{-3}, N = 64$	$1.07 \cdot 10^4$	1.0	-	$1.27 \cdot 10^3$	8.4	-	$2.24 \cdot 10^3$	4.8	-	$5.04 \cdot 10^2$	21.3	-	$3.54 \cdot 10^2$	30.3	-
$M = 10^{-4}, N = 64$	$1.07 \cdot 10^5$	1.0	-	$1.28 \cdot 10^3$	83.9	-	$4.74 \cdot 10^3$	22.7	-	$4.97 \cdot 10^2$	215.7	-	$3.55 \cdot 10^2$	302.1	-
$M = 10^{-1}, N = 128$	$9.34 \cdot 10^2$	1.0	7.9	$1.32 \cdot 10^4$	0.1	9.8	$1.23 \cdot 10^3$	0.8	7.7	$1.52 \cdot 10^3$	0.6	7.1	$1.31 \cdot 10^3$	0.7	6.3
$M = 10^{-2}, N = 128$	$8.53 \cdot 10^3$	1.0	7.9	$1.37 \cdot 10^4$	0.6	10.2	$4.50 \cdot 10^3$	1.9	7.9	$4.67 \cdot 10^3$	1.8	9.9	$2.47 \cdot 10^3$	3.5	6.8
$M = 10^{-3}, N = 128$	$8.62 \cdot 10^4$	1.0	8.0	$1.36 \cdot 10^4$	6.3	10.7	$3.07 \cdot 10^4$	2.8	13.7	$5.23 \cdot 10^3$	16.5	10.4	$2.31 \cdot 10^3$	37.3	6.5
$M = 10^{-4}, N = 128$	$8.36 \cdot 10^5$	1.0	7.8	$1.32 \cdot 10^4$	63.6	10.3	$7.27 \cdot 10^4$	11.5	15.4	$4.92 \cdot 10^3$	170.1	9.9	$2.36 \cdot 10^3$	354.9	6.6
$M = 10^{-1}, N = 256$	$7.34 \cdot 10^3$	1.0	7.9	$1.89 \cdot 10^5$	0.0	14.3	$8.48 \cdot 10^3$	0.9	6.9	$1.05 \cdot 10^4$	0.7	6.9	$1.01 \cdot 10^4$	0.7	7.7
$M = 10^{-2}, N = 256$	$6.71 \cdot 10^4$	1.0	7.9	$1.80 \cdot 10^5$	0.4	13.2	$3.20 \cdot 10^4$	2.1	7.1	$4.41 \cdot 10^4$	1.5	9.4	$1.67 \cdot 10^4$	4.0	6.8
$M = 10^{-3}, N = 256$	$6.64 \cdot 10^5$	1.0	7.7	$1.79 \cdot 10^5$	3.7	13.1	$5.30 \cdot 10^5$	1.3	17.3	$5.90 \cdot 10^4$	11.2	11.3	$1.67 \cdot 10^4$	39.8	7.2
$M = 10^{-4}, N = 256$	$6.58 \cdot 10^6$	1.0	7.9	$1.81 \cdot 10^5$	36.3	13.8	$4.18 \cdot 10^6$	1.6	57.4	$5.66 \cdot 10^4$	116.3	11.5	$1.69 \cdot 10^4$	389.5	7.2
$M = 10^{-1}, N = 512$	$5.83 \cdot 10^4$	1.0	7.9	n/a	n/a	n/a	$6.10 \cdot 10^4$	1.0	7.2	$7.53 \cdot 10^4$	0.8	7.1	$7.85 \cdot 10^4$	0.7	7.8
$M = 10^{-2}, N = 512$	$5.35 \cdot 10^5$	1.0	8.0	n/a	n/a	n/a	$2.00 \cdot 10^5$	2.7	6.3	$3.98 \cdot 10^5$	1.3	9.0	$1.24 \cdot 10^5$	4.3	7.4
$M = 10^{-3}, N = 512$	$5.28 \cdot 10^6$	1.0	8.0	n/a	n/a	n/a	$4.34 \cdot 10^6$	1.2	8.2	$7.04 \cdot 10^5$	7.5	11.9	$1.20 \cdot 10^5$	43.9	7.2
$M = 10^{-4}, N = 512$	$5.26 \cdot 10^7$	1.0	8.0	n/a	n/a	n/a	n/a	n/a	n/a	$5.76 \cdot 10^5$	91.3	10.2	$1.27 \cdot 10^5$	413.8	7.5
$M = 10^{-1}, N = 1024$	$4.72 \cdot 10^5$	1.0	8.1	n/a	n/a	n/a	$4.43 \cdot 10^5$	1.1	7.3	$5.20 \cdot 10^5$	0.9	6.9	$6.35 \cdot 10^5$	0.7	8.1
$M = 10^{-2}, N = 1024$	$4.33 \cdot 10^6$	1.0	8.1	n/a	n/a	n/a	$1.39 \cdot 10^6$	3.1	6.9	$2.91 \cdot 10^6$	1.5	7.3	$8.28 \cdot 10^5$	5.2	6.7
$M = 10^{-3}, N = 1024$	$4.31 \cdot 10^7$	1.0	8.1	n/a	n/a	n/a	$2.47 \cdot 10^7$	1.7	5.7	$6.50 \cdot 10^6$	6.6	9.2	$9.24 \cdot 10^5$	46.6	7.7
$M = 10^{-4}, N = 1024$	$4.35 \cdot 10^8$	1.0	8.3	n/a	n/a	n/a	n/a	n/a	n/a	$9.28 \cdot 10^6$	46.8	16.1	$9.52 \cdot 10^5$	456.5	7.5

Table 5.1: Gresho vortex, r: ratio of computing time to physical time, s: speedup compared to explicit time stepping, c: amount of work relative to work at half resolution

6 Nearly hydrostatic flows

So far, the discretization of the compressible Euler equations was performed in the absence of the gravitational source term. It was shown in Section 2.3.2 that the inclusion of this term permits a quiet steady state, called hydrostatic equilibrium, where a non-vanishing pressure gradient is balanced by gravity. For the simulation of stellar atmospheres, it is expected that the flow fields are very close to such a hydrostatic state because stars usually do not evolve significantly on a dynamical time scale. For a numerical simulation of stellar processes, it is therefore of major importance that the discretization is able to maintain a hydrostatic equilibrium and also resolves small deviations from it accurately.

6.1 Discretization of gravity

6.1.1 Pointwise gravity

In order to discretize the gravitational source term in the finite-volume framework used in this work, the volume integral

$$\mathbf{S}_{i,j,k} = \frac{1}{V_{i,j,k}} \int_{\Omega_{i,j,k}} J^{-1}\mathbf{S} \; d\Omega. \tag{6.1.1}$$

has to be approximated on the discrete grid. The easiest way to do that is by recognizing that the conservative variables stored on the grid are in fact volume averages. With these values, the discrete gravitational source term may be written as

$$\mathbf{S}^g_{i,j,k} = \frac{1}{F_\mathrm{r}^2} \begin{pmatrix} 0 \\ \rho g_x \\ \rho g_y \\ \rho g_z \\ M_\mathrm{r}^2 \rho \; \mathbf{q}\cdot\mathbf{g} \\ 0 \end{pmatrix}_{i,j,k} . \tag{6.1.2}$$

The vector of the gravitational acceleration $\mathbf{g}$ has to be provided as a proper cell-averaged value at each grid point. For implicit time marching schemes, the derivative of the discrete source term with respect to the conservative variables is also needed. Here, this derivative can be evaluated analytically and results in

$$\frac{\partial \mathbf{S}^g_{i,j,k}}{\partial \mathbf{U}_{l,m,n}} = \frac{1}{F_\mathrm{r}^2} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ g_x & 0 & 0 & 0 & 0 & 0 \\ g_y & 0 & 0 & 0 & 0 & 0 \\ g_z & 0 & 0 & 0 & 0 & 0 \\ 0 & M_\mathrm{r}^2 g_x & M_\mathrm{r}^2 g_y & M_\mathrm{r}^2 g_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{i,j,k} \delta^{l,m,n}_{i,j,k}. \tag{6.1.3}$$

This term has to be added to the diagonal block of the Jacobian matrix of the spatial residual.

In the following, the ability of this discretization to maintain a hydrostatic equilibrium is investigated theoretically. As such an analysis quickly becomes very complicated, it is restricted to a one-dimensional setup of a hydrostatic atmosphere of an ideal gas with constant gravitational acceleration g. Moreover, the analysis is performed with dimensional quantities in this section

(the reference quantities are all assumed to be unity). The spatial residual at a certain grid cell i can then be written as

$$\mathbf{R}_i = \frac{1}{\Delta x}\left(\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}\right) - \mathbf{S}_i^g. \tag{6.1.4}$$

For a perfect hydrostatic equilibrium, the time derivative of the conservative variables has to be zero everywhere (compare Equation 5.0.1). Thus, the spatial residual has to vanish everywhere on the domain. Any deviations from zero will lead to non-stationary behavior within the first time step and some fluid motions set in.

In order to further simplify the analysis, it is assumed that the interface values are computed by constant extrapolation (see Section 3.4.2) and a flux-preconditioned version of Roe's approximate Riemann solver (e.g. the Roe-Lowmach or Roe-Turkel scheme). The numerical flux can then be written in the form

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}\left(\mathbf{U}_{i+1}\right) + \mathbf{F}\left(\mathbf{U}_i\right) - D_{i+1/2}\left(\mathbf{U}_{i+1} - \mathbf{U}_i\right)\right), \tag{6.1.5}$$

where $D_{i+1/2}$ denotes the corresponding upwinding matrix. Evaluating the flux difference under hydrostatic conditions (i.e. with zero velocity), the spatial residual at a certain grid cell i can be written as

$$\mathbf{R}_i = \begin{pmatrix} 0 \\ \frac{p_{i+1}-p_{i-1}}{2\Delta x} - \rho_i g \\ 0 \end{pmatrix} + \frac{1}{2\Delta x}\left(-d_{i+1/2} + d_{i-1/2}\right), \tag{6.1.6}$$

$$d_{i+1/2} = D_{i+1/2} \cdot \left(\mathbf{U}_{i+1} - \mathbf{U}_i\right). \tag{6.1.7}$$

The first term corresponds to a discrete approximation of the hydrostatic balance equation $\frac{\partial p}{\partial x} = \rho g$. Note that the pressure difference in the spatial residual is just a central finite difference approximation of the derivative, emphasizing the consistency with the continuous equations. This approximation can be further evaluated by assuming that the initial grid values are set up with a hydrostatic equilibrium of an isothermal atmosphere,

$$\rho(x) = \rho_0 \exp\left(-\frac{g\,x}{\tilde{R}T}\right) \qquad p(x) = \rho(x)\tilde{R}T, \tag{6.1.8}$$

which is simply discretized on a regular spaced grid by

$$x_i = i\ \Delta x \qquad \rho_i = \rho\left(x_i\right) \qquad p_i = p\left(x_i\right) \qquad u_i = 0. \tag{6.1.9}$$

The discrete pressure gradient then be simplified in the following way:

$$\frac{p_{i+1} - p_{i-1}}{2\Delta x} = \frac{\rho_0 \tilde{R}T}{2\Delta x}\left(\exp\left(-\frac{g\,\Delta x(i+1)}{\tilde{R}T}\right) - \exp\left(-\frac{g\,\Delta x(i-1)}{\tilde{R}T}\right)\right) \tag{6.1.10}$$

$$= \frac{\rho_0 \tilde{R}T}{\Delta x}\exp\left(-\frac{g\,\Delta x i}{\tilde{R}T}\right)\sinh\left(-\frac{g\,\Delta x}{\tilde{R}T}\right) \tag{6.1.11}$$

$$= \rho_i g\left(1 + \frac{\Delta x^2}{6} + \ldots\right). \tag{6.1.12}$$

In the last step, a Taylor expansion of hyperbolic sine was introduced. It can be seen that the pressure gradient equals the gravitational source term up to first order in the grid spacing Δx. The first deviation from the discrete hydrostatic equilibrium is thus second order in Δx. This behavior is remarkable as the constant interface reconstruction method is only first-order accurate in space. However, these small deviations from the discrete hydrostatic balance may cause non-stationary movements of the fluid, if the initial conditions are advanced in time. Ideally, these movements damp out after a while such that a new discrete hydrostatic equilibrium is formed which is only slightly different from the initial conditions. However, such a behavior can only be investigated by numerical simulations.

So far, only the first term of Equation 6.1.6 has been analyzed. The second term depends on the upwinding matrix of a specific numerical flux function. Under the assumption of zero velocity, the upwinding term at an interface can be evaluated for the Roe-Lowmach numerical flux function:

$$d_{i+1/2} = \frac{\Delta p_{i+1/2}}{\sqrt{1+\delta^2}} \begin{pmatrix} \frac{1}{c} \\ \delta \\ \frac{c}{\gamma-1} \end{pmatrix} \approx \Delta p_{i+1/2} \begin{pmatrix} \frac{M_{\text{cut}}}{c} \\ 1 - M_{\text{cut}} \\ \frac{cM_{\text{cut}}}{\gamma-1} \end{pmatrix}. \tag{6.1.13}$$

Note that since the velocity is assumed to be zero, the preconditioning parameter is solely determined by the cut-off Mach number, i.e. $\delta = 1/M_{\text{cut}} - 1$. The second term of the spatial residual then reads

$$\frac{1}{2\Delta x}\left(-d_{i+1/2} + d_{i-1/2}\right) = -\frac{\Delta x}{2} \cdot \frac{p_{i+1} - 2p_i + p_{i-1}}{\Delta x^2} \cdot \begin{pmatrix} \frac{M_{\text{cut}}}{c} \\ 1 - M_{\text{cut}} \\ \frac{cM_{\text{cut}}}{\gamma-1} \end{pmatrix} \tag{6.1.14}$$

$$\approx -\frac{\Delta x}{2} \cdot \frac{\partial^2 p(x)}{\partial x^2} \cdot \begin{pmatrix} \frac{M_{\text{cut}}}{c} \\ 1 - M_{\text{cut}} \\ \frac{cM_{\text{cut}}}{\gamma-1} \end{pmatrix}. \tag{6.1.15}$$

It can be seen that the term contains a pressure difference, which is a finite difference approximation of the second derivative of the pressure. The overall expression is thus an artificial pressure diffusion term. It scales linearly with the grid spacing rendering the overall discretization first order accurate in space (as expected for constant interface reconstruction). However, the artificial diffusion is considered to be very harmful for the discrete hydrostatic equilibrium as it affects all three equations. Since the hydrostatic equilibrium can be constructed for arbitrary temperature profiles, it is expected that such a profile can not be maintained as the initial conditions are advanced in time. However, it can also be seen that the diffusion coefficient scales with the cut-off Mach number, which is introduced by the preconditioning matrix of the newly developed Roe-Lowmach scheme. Setting this parameter to very low values, the artificial diffusion can be greatly inhibited in the continuity and the energy equation. It is therefore expected that the new numerical flux function can maintain a discrete hydrostatic equilibrium much better than Roe's original approximate Riemann solver, which corresponds to $M_{\text{cut}} = 1$. Unfortunately, some additional artificial diffusion is introduced in the momentum equation, which does not vanish as $M_{\text{cut}} \to 0$. Although this analysis reveals some basic properties of the numerical discretization concerning the stability of hydrostatic atmospheres, the detailed behavior can only be explored by numerical experiments.

The artificial diffusion term can also be computed analytically for the Roe-Turkel numerical flux function (Section 4.3.2). The final expression for this term reads

$$\frac{1}{2\Delta x}\left(-d_{i+1/2} + d_{i-1/2}\right) = -\frac{\Delta x}{2} \cdot \frac{p_{i+1} - 2p_i + p_{i-1}}{\Delta x^2} \cdot \begin{pmatrix} \frac{1}{cM_{\text{cut}}} \\ 0 \\ \frac{c}{(\gamma-1)M_{\text{cut}}} \end{pmatrix}. \tag{6.1.16}$$

This is very similar to the Roe-Lowmach scheme, except that the diffusion coefficient scales with the *inverse* of the cut-off Mach number. Thus, the diffusion is enhanced for low Mach number flows and it is even more complicated to maintain the hydrostatic equilibrium. This is basically the reason which triggered the development of the Roe-Lowmach scheme, because the Roe-Turkel scheme can not be used for stratified flows where the second derivative of the pressure is not necessarily small in the vertical direction.

Finally, it must be emphasized that the above analysis is only valid for hydrostatic initial conditions where the velocity field is identically zero. If the grid contains finite velocities, additional terms may arise in the spatial residual which may either further deteriorate the hydrostatic equilibrium or enforce a slightly modified equilibrium state. Such a behavior can only be investigated

by numerical experiments which are presented in the following sections. Nevertheless, the analysis reveals the basic properties of the numerical flux functions in the presence of the discrete gravitational source term. While the Roe-Turkel flux is considered to be very harmful to the hydrostatic equilibrium, the Roe-Lowmach flux lowers the artificial diffusion terms significantly compared to the original Roe method. However, as none of the numerical flux functions seems to be able to maintain a hydrostatic equilibrium on the discrete level exactly, these schemes are called *non well-balanced*. Numerical discretizations of the gravitational source term which are potentially well-balanced are presented in the following two subsections.

6.1.2 F-wave Riemann solver

Numerical discretizations which are potentially able to maintain hydrostatic equilibria exactly on a discrete level are for example described in the book by Bouchut (2004). In the following, the derivation of one particular method, named *F-wave Riemann solver*, is briefly reviewed. Further details can be found in the book and references therein. Moreover, an extension of this method for low Mach number flows is proposed here.

Assuming a one-dimensional Cartesian geometry, the system of Euler equations including the gravitational source can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S}^g. \tag{6.1.17}$$

The system is now augmented by an auxiliary variable Z extending the vector of conserved quantities to

$$\mathbf{W} = \begin{pmatrix} \mathbf{U} \\ Z \end{pmatrix}. \tag{6.1.18}$$

Defining $Z = x$, Equation 6.1.17 can be rewritten in the following form:

$$\frac{\partial \mathbf{W}}{\partial t} + \underbrace{\begin{pmatrix} A & -\mathbf{S}^g \\ 0 & 0 \end{pmatrix}}_{=B} \frac{\partial \mathbf{W}}{\partial x} = 0 \qquad \text{with} \qquad A = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}. \tag{6.1.19}$$

The system is now in quasi-linear form where the source term has been absorbed into the matrix. Note that since the last row of B is identically zero, the auxiliary variable Z may not change in time. In the transformed system, a Riemann problem can be defined by piecewise constant initial data, i.e.

$$\mathbf{W}(\chi) = \begin{cases} \mathbf{W}^L_{i+1/2} & \text{for } \chi > 0 \\ \mathbf{W}^R_{i+1/2} & \text{for } \chi < 0 \end{cases}. \tag{6.1.20}$$

The main idea of the new discretization technique is to solve such a Riemann problem of the augmented system at each cell interface in order to define numerical fluxes for the original system, which take the following form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{1}{\Delta x}\left(\mathbf{F}^-_{i+1/2} - \mathbf{F}^+_{i-1/2}\right) = 0. \tag{6.1.21}$$

The source term is now absorbed in the numerical fluxes, which are necessarily not conservative anymore, i.e. $\mathbf{F}^+_{i+1/2} \neq \mathbf{F}^-_{i+1/2}$. However, the intention of this procedure is that the discretization errors of the hydrodynamic fluxes and the source term are now much more consistent. It is shown in the book by Bouchut (2004) that such a scheme can indeed be well-balanced.

The remaining task is to find an appropriate Riemann solver for the non-conservative system. As an exact Riemann solver is very hard to derive for the system considered here, an approximate solver is used instead. The combination of the non-conservative system approach and the Roe solver is actually called *F-wave Riemann solver* then. Besides the well-balancing properties of the

overall scheme, the numerical viscosity in the low Mach number regime is also of importance here (cf. Chapter 4). In order to account for this, the low Mach number preconditioning technique described in Section 4.3.3 is extended to the non-conservative Riemann solver in the following.

It is shown by Bouchut (2004) that the F-wave fluxes can be constructed by the following expressions:

$$\mathbf{F}^-_{i+1/2} = \mathbf{F}\left(\mathbf{U}^L_{i+1/2}\right) + \sum_{\lambda_k<0} \delta\mathbf{F}_k, \tag{6.1.22}$$

$$\mathbf{F}^+_{i+1/2} = \mathbf{F}\left(\mathbf{U}^R_{i+1/2}\right) - \sum_{\lambda_k>0} \delta\mathbf{F}_k. \tag{6.1.23}$$

These expressions are only valid for a *constant* interface reconstruction, rendering the overall method first-order accurate in space. An extension to higher order interface reconstruction methods is possible, but is beyond the scope of this thesis. In order to define the split fluxes $\delta\mathbf{F}_k$ in the above expressions, the eigensystem of the *preconditioned* quasi-linear system is computed:

$$PB = R\Lambda L. \tag{6.1.24}$$

Here, R denotes the matrix whose k-th column contains the right eigenvector of PB, whereas the k-th row of L contains the corresponding left eigenvector belonging to the eigenvalue λ_k. The flux preconditioning matrix is defined similarly to the Roe-Lowmach method (see Equation 4.3.12) with an additional identity row/column for the auxiliary variable Z. The split fluxes are then computed by

$$\delta\mathbf{F}_k = \lambda_k \left(L_k \cdot \left(\mathbf{W}^R_{i+1/2} - \mathbf{W}^L_{i+1/2}\right)\right) P^{-1} R_k. \tag{6.1.25}$$

It can be shown that the numerical fluxes defined by Equation 6.1.22 are equal to fluxes of the Roe-Lowmach scheme in the absence of gravity, as long as the split fluxes are evaluated at the Roe-averaged state. Moreover, it can be shown by some complicated algebraic transformations (not presented here), that the presence of the gravitational source term only adds the following two terms to the numerical flux compared to the original Roe-Lowmach flux:

$$\mathbf{S}^{q_n-\tau}_{i+1/2} = -\frac{g\,\Delta x}{2\tau\, F_\mathrm{r}^2}\,\frac{\partial \mathbf{U}}{\partial \mathbf{V}} \cdot \begin{pmatrix} \frac{\rho}{c}\left(c + \delta M_\mathrm{r} q_n\right) \\ \frac{1}{M_\mathrm{r}}\left(c\delta - \tau M_\mathrm{r}\right) \\ \rho c \left(c + \delta M_\mathrm{r} q_n\right) \end{pmatrix} \tag{6.1.26}$$

$$\mathbf{S}^{q_n+\tau}_{i+1/2} = +\frac{g\,\Delta x}{2\tau\, F_\mathrm{r}^2}\,\frac{\partial \mathbf{U}}{\partial \mathbf{V}} \cdot \begin{pmatrix} \frac{\rho}{c}\left(c + \delta M_\mathrm{r} q_n\right) \\ \frac{1}{M_\mathrm{r}}\left(c\delta + \tau M_\mathrm{r}\right) \\ \rho c \left(c + \delta M_\mathrm{r} q_n\right) \end{pmatrix} \tag{6.1.27}$$

For a detailed explanation of the quantities refer to Appendix A. With these expressions, the non-conservative numerical fluxes can be computed by the following algorithm:

if $q_n \pm \tau < 0$ then

$$\mathbf{F}^-_{i+1/2} = \mathbf{F}^{g=0}_{i+1/2} + \mathbf{S}^{q_n\pm\tau}_{i+1/2} \tag{6.1.28}$$

else

$$\mathbf{F}^+_{i+1/2} = \mathbf{F}^{g=0}_{i+1/2} - \mathbf{S}^{q_n\pm\tau}_{i+1/2} \tag{6.1.29}$$

end if

It should be noted that the sum of the of the two additional terms equals the original source times the grid spacing:

$$\mathbf{S}^{q_n-\tau}_{i+1/2} + \mathbf{S}^{q_n+\tau}_{i+1/2} = \frac{g\,\Delta x}{F_\mathrm{r}^2}\,\frac{\partial \mathbf{U}}{\partial \mathbf{V}} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \frac{g\,\Delta x}{F_\mathrm{r}^2} \begin{pmatrix} 0 \\ \rho \\ \rho u M_\mathrm{r}^2 \end{pmatrix} = \Delta x\, \mathbf{S}^g \tag{6.1.30}$$

The F-wave Riemann solver can thus be seen as a method to split the source term at the cell interfaces, which is consistent with the upwinding of the original Roe-type solver. Beside that, the F-wave Riemann solver without flux preconditioning can indeed be well-balanced if the average values for the evaluation of the split fluxes $\delta\mathbf{F}_k$ are chosen in a proper way (see Bouchut 2004). As explained before, the consistency with the $g = 0$ fluxes requires that the split fluxes are evaluated at the Roe average. The only additional variable that needs to be defined at the cell interfaces is the gravitational acceleration g. For simplicity, it is assumed in the following tests, that g is spatially constant and thus takes this value also at the cell interfaces. The unpreconditioned scheme is thus expected to be well-balanced. Whether the flux preconditioned F-wave solver is also well-balanced is theoretically not quite clear at this point. The behavior of this scheme is examined by numerical experiments presented in Section 6.2.

The F-wave Riemann solver can generally be extended to two or three spatial dimensions. However, this has only been implemented yet for the special case of Cartesian geometries where the gravitational acceleration is aligned with one coordinate axes. The non-conservative fluxes along this axes are then computed similar to the method described above, whereas the numerical fluxes in the other directions are computed with the original Roe-Lowmach scheme (cf. Bouchut 2004).

6.1.3 Hydrostatic reconstruction

A very different method to obtain a well-balanced numerical scheme for the gravitational source term is described by Fuchs et al. (2010), which is briefly reviewed in the following. The main idea of this method is to change the interface reconstruction method in a way that only deviations from a discrete hydrostatic equilibrium may produce waves which are computed by the unmodified numerical flux function.

As in the previous section, the method has only been implemented for spatially first order scheme on a Cartesian grid, where the gravitational acceleration is aligned with one of the coordinate axes. Moreover, it has to be assumed that an ideal gas equation of state is used. Extensions to more general conditions are possible but are beyond the scope of this work.

For the coordinate axes not aligned with the gravitational acceleration, the piecewise constant reconstruction is performed in the usual manner as described in Section 3.4.2. For the other coordinate axis (denoted by z in the following), piecewise constant profiles are assumed for the density and the velocities. The corresponding interface values are thus given by

$$\rho^L_{i+1/2} = \rho_i \qquad \rho^R_{i+1/2} = \rho_{i+1} \tag{6.1.31}$$

$$\mathbf{q}^L_{i+1/2} = \mathbf{q}_i \qquad \mathbf{q}^R_{i+1/2} = \mathbf{q}_{i+1}. \tag{6.1.32}$$

In order to compute the pressure at the cell interfaces, a hydrostatic pressure profile is assumed within each cell:

$$\tilde{p}_i(z) = p_i \ \exp\left(-\frac{z}{H_{p,i}}\right) \tag{6.1.33}$$

Here, the (non-dimensional) pressure scale height is computed by assuming a constant temperature and species profile within each cell:

$$H_{p,i} = \frac{F_\mathrm{r}^2}{M_\mathrm{r}^2} \cdot \frac{p_i}{g_{z,i}\rho_i} \tag{6.1.34}$$

Note that the reference Froude number F_r and the reference Mach number M_r used for non-dimensionalization (see Section 2.5) are globally constant. The pressure at the cell interface is then evaluated by:

$$p^L_{i+1/2} = \tilde{p}_i\,(+\Delta z/2) \qquad p^R_{i+1/2} = \tilde{p}_{i+1}\,(-\Delta z/2) \tag{6.1.35}$$

With the interface values calculated with this method, the numerical fluxes are computed by an arbitrary scheme. It should be noted that the pressure difference at a certain interface is greatly

reduced compared to the original constant reconstruction scheme and thus inhibits the creation of potentially unphysical waves.

The gravitational source term is then discretized in the momentum equation of the corresponding coordinate axes by

$$\mathbf{S}^g_{i,\rho w} = \frac{1}{M_\mathrm{r}^2\, F_\mathrm{r}^2} \cdot \frac{p^R_{i+1/2} - p^L_{i-1/2}}{\Delta z}, \tag{6.1.36}$$

which is a second-order approximation, although the overall scheme is only first-order accurate. The source term in the energy equation is discretized similar to the pointwise gravity, i.e.

$$\mathbf{S}^g_{i,\rho E} = \frac{M_\mathrm{r}^2}{F_\mathrm{r}^2} \cdot (\rho w\, g_z)_i. \tag{6.1.37}$$

Although this discretization is slightly inconsistent to the momentum equation, very good results concerning the well-balancing of the overall method are reported by Fuchs et al. (2010). The behavior of the hydrostatic reconstruction in combination with the Roe-Lowmach numerical flux is examined by numerical tests in the following sections.

6.2 Stability of 1D hydrostatic atmospheres

The first numerical test for the various discretization methods presented above is a one-dimensional quiet hydrostatic atmosphere, which should ideally be maintained without any distortions as it is advanced in time. As explained in Section 2.3.2, a hydrostatic equilibrium is described by the differential equation

$$\frac{\partial \hat{p}}{\partial \hat{y}} = -\hat{g} \frac{M_\mathrm{r}^2}{F_\mathrm{r}^2} \hat{\rho}, \tag{6.2.1}$$

which is written in non-dimensional units here (the hat is not omitted in this section for clarity). The reference pressure and temperature are chosen to their corresponding values at zero height, i.e. $p_\mathrm{r} = p_0$ and $T_\mathrm{r} = T_0$. The gravitational acceleration is assumed to be constant here and also chosen as corresponding reference value, such that $\hat{g} = 1$. The reference length is chosen to the pressure scale height at $\hat{y} = 0$, i.e. $y_\mathrm{r} = \frac{\tilde{R} T_0}{g}$. Moreover, an ideal gas equation of state is used here, which provides the necessary additional relation between the density and the pressure. With these assumptions, the hydrostatic balance equation can be simplified to

$$\frac{\partial \hat{p}}{\partial \hat{y}} = -\frac{\hat{p}}{\hat{T}}. \tag{6.2.2}$$

Note that the hydrostatic pressure can be constructed for an arbitrary temperature profile, which has to be given. For the numerical tests presented here, a hyperbolic tangent temperature profile is chosen:

$$\hat{T} = 1 + \hat{\alpha} \tanh\left(\hat{y}/\hat{w}\right) \tag{6.2.3}$$

The parameters $\hat{\alpha}$ and $\hat{w}$ denote the non-dimensional strength and width of the temperature gradient. The hydrostatic pressure can then be integrated analytically and results in

$$\hat{p} = \exp\left(-\frac{\hat{y} - \hat{\alpha}\hat{w} \log\left(\cosh\left(\hat{y}/\hat{w}\right) + \hat{\alpha} \sinh\left(\hat{y}/\hat{w}\right)\right)}{1 - \hat{\alpha}^2} \right). \tag{6.2.4}$$

The temperature and pressure profiles serve as initial conditions for the numerical tests. The corresponding density profile can be obtained by the equation of state. The initial velocity field is set to zero identically. The Brunt-Väisälä-frequency at zero height for this setup is given by

$$N_0 = \sqrt{\frac{\tilde{R} T_0}{H_{p,0}^2} \left(\frac{\gamma - 1}{\gamma} + \frac{\hat{\alpha}}{\hat{w}} \right)}, \tag{6.2.5}$$

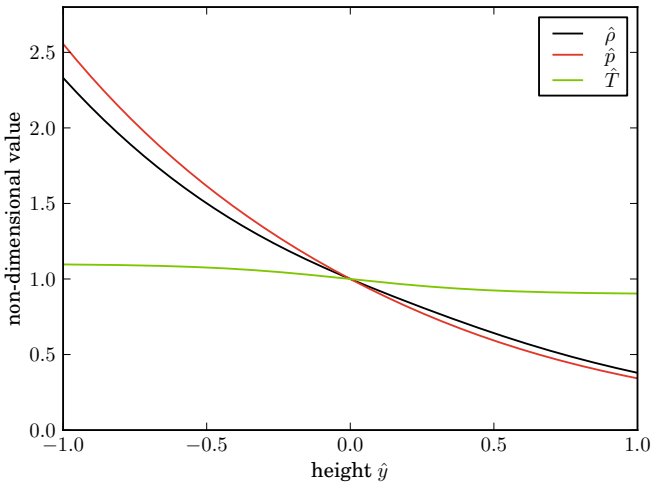


Figure 6.1: Initial density, pressure and temperature profiles

which is used to define the reference time $t_\mathrm{r} = 2\pi/N_0$ for this non-dimensional setup (cf. Section 6.4).

For the numerical tests, the computational domain is chosen to $[-1, 1]$ which is discretized by $N \in \{64, 128, 256\}$ grid cells. The parameters for the temperature profile are set to $\hat{\alpha} = -0.1$ and $\hat{w} = 0.5$. The initial density, pressure and temperature profiles are shown in Figure 6.1. The numerical tests are performed with the implicit backward Euler method at fixed time step $\Delta\hat{t} = 0.1$ until $\hat{t} = 100$ is reached. This integration method is known to be rather inaccurate in resolving transient phenomena because it damps out waves rather quickly. However, for the tests performed here, this behavior is even advantageous because the main interest lies on the long term stability of the discrete hydrostatic atmosphere, where transient phases are not important anymore.

In order to analyze the results, three different quantities are measured at the end of each simulation. First, the maximum Mach number on the grid

$$M_f = \max_i |M_i| \tag{6.2.6}$$

is computed. As explained in Section 2.3.2, the Euler equations reduce to the hydrostatic balance equations if the velocity field is identically zero. Thus, the maximum Mach number measures how non-hydrostatic a result is. Ideally, M_f should reduce to values of the order of the machine precision, which is approximately 10^{-16} here. The second quantity that is computed at the end of each simulation is the average distortion in the temperature field compared to the initial profile and relative to the strength of the temperature gradient:

$$\sigma_T(\hat{t}) = \frac{1}{\hat{\alpha} N} \sum_i \left| \hat{T}_i(t) - \hat{T}_i(0) \right| \tag{6.2.7}$$

As explained above, hydrostatic equilibria can be constructed for arbitrary temperature profiles. Once the initial hydrostatic equilibrium gets distorted (e.g. by discretization errors), a velocity field builds up. Even if these motions damp out again, it is not guaranteed that the initial temperature profile is maintained during this relaxation process. This is a result of the fact that the hydrostatic equilibrium is not uniquely defined by the Euler equations. Therefore, σ_T is an important measure for the quality of a numerical discretization for hydrostatic atmospheres.

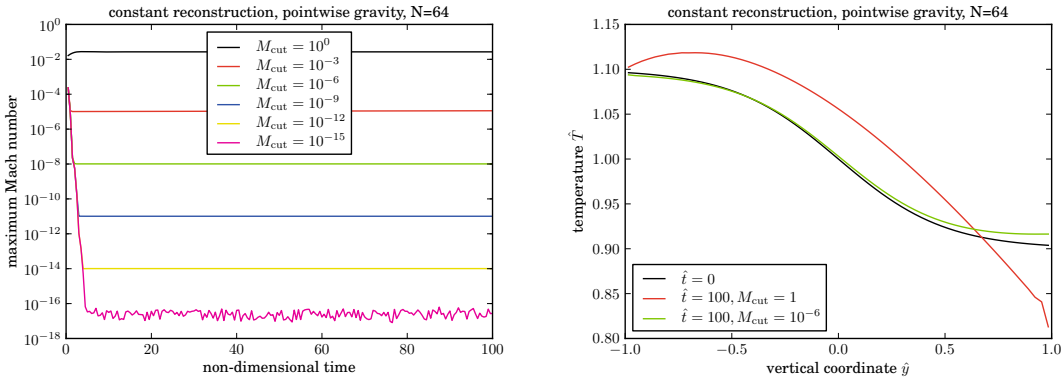


Figure 6.2: time evolution of the maximum Mach number (left), temperature distributions compared to the initial profile (right)

Finally, the time derivative of σ_T is computed at the end of each simulation, which is simply approximated by

$$\dot{\sigma}_T(\hat{t}) = \frac{\sigma_T(\hat{t}) - \sigma_T(\hat{t} - 5)}{5}. \tag{6.2.8}$$

With this value, it is possible to estimate the temperature changes on time scales which exceed the simulation time.

The first numerical tests are performed with constant interface reconstruction, the Roe-Lowmach flux with different cut-off Mach numbers M_{cut} and the pointwise discretization of gravity described in Section 6.1.1. The following table summarizes the results:

	M_f			σ_T			$\dot{\sigma}_T$		
M_{cut} \ N_x	64	128	256	64	128	256	64	128	256
1e0	2.7e-02	1.4e-02	7.4e-03	3.8e-01	3.4e-01	3.1e-01	4.4e-15	-1.8e-08	-5.8e-06
1e-3	1.1e-05	5.9e-06	3.0e-06	5.5e-02	2.8e-02	1.4e-02	2.2e-04	1.1e-04	5.7e-05
1e-6	1.0e-08	5.2e-09	2.7e-09	4.1e-02	2.1e-02	1.0e-02	1.3e-08	1.3e-08	6.6e-09
1e-9	1.0e-11	5.2e-12	2.7e-12	4.1e-02	2.1e-02	1.0e-02	1.3e-11	1.3e-11	6.6e-12
1e-12	1.0e-14	5.2e-15	2.7e-15	4.1e-02	2.1e-02	1.0e-02	1.3e-14	1.3e-14	6.6e-15
1e-15	1.6e-17	8.6e-18	3.4e-17	4.1e-02	2.1e-02	1.0e-02	-1.7e-17	0.0e+00	-7.6e-17

It can be seen that the final Mach numbers with the original Roe flux ($M_{\text{cut}} = 1$) are of the order 10^{-2}. Although, doubling the grid resolution approximately halves the final Mach number, the values are much too high for practical applications. When flux preconditioning is switched on the final Mach numbers decrease approximately linear with the cut-off Mach number. Figure 6.2 shows the temporal evolution of the maximum Mach number for $N = 64$, depending on the amount of flux-preconditioning. It can be seen that the Mach number increases from zero to approximately 10^{-2} within the first time step. With the unpreconditioned ($M_{\text{cut}} = 1$) flux, the Mach number stays at this level. With flux preconditioning, the Mach number quickly decreases again, until a saturation level is reached, which linearly scales with the cut-off Mach number. These results fully comply with the theoretical considerations presented in Section 6.1.1 about the artificial diffusion terms produced by the numerical flux function. Thus, the saturation of the maximum Mach number can be attributed to a balance between the artificial diffusion terms and a finite velocity field.

In the above table, it can further be seen that the temperature distortions, measured by σ_T, are rather high for the unpreconditioned flux ($M_{\text{cut}} = 1$). Increasing the grid resolution improves the

results only marginally. Using the preconditioned flux reduces the temperature fluctuations by approximately a factor of ten. Unfortunately, the distortions are not reduced significantly as the cut-off Mach number is further decreased. However, the temperature distortions decrease linearly as the grid resolution is increased in the flux preconditioned case. The actual temperature profiles at the beginning and the end of the simulations are shown in the right panel of Figure 6.2 for $M_{\text{cut}} = 1$ and $M_{\text{cut}} = 10^{-9}$ in the $N = 64$ case. It can be seen that the initial temperature profile gets completely distorted with the unpreconditioned numerical flux. This kind of discretization is thus not suited at all to compute nearly hydrostatic flows. Contrary to that, the final temperature profile is much closer to the initial profile in the preconditioned case, although the deviations are quite substantial.

Looking at the time derivative of the temperature distortions, $\dot{\sigma}_T$, it can be seen that the values decrease linearly with the cut-off Mach number. The temperature deviations are expected to increase substantially on a time scale of the order $1/\dot{\sigma}_T$. For $M_{\text{cut}} = 10^{-3}$ this would correspond to a timescale of the order of 10^4 non-dimensional units, which is already quite large. For $M_{\text{cut}} = 10^{-15}$ the derivative $\dot{\sigma}_T$ nearly vanishes. Therefore, it can be concluded that the preconditioned Roe-Lowmach flux is able to maintain a discrete hydrostatic equilibrium state if the cut-off Mach number is low enough, although the temperature profile gets somewhat distorted during the initial relaxation phase.

The next group of tests is also performed with the Roe-Lowmach flux and a pointwise discretization of gravity, but uses the linear interface reconstruction method, which is second order accurate in space. The other parameters of the setup are equal to those in the tests described before. The results are shown in the following table:

	M_f			σ_T			$\dot{\sigma}_T$		
M_{cut} \ N_x	64	128	256	64	128	256	64	128	256
1e0	1.8e-06	2.5e-07	3.3e-08	6.4e-03	8.3e-04	1.1e-04	6.1e-05	7.9e-06	9.9e-07
1e-3	4.6e-08	1.1e-08	2.8e-09	9.6e-05	2.3e-05	5.7e-06	-8.4e-07	-2.1e-07	-5.1e-08
1e-6	5.7e-11	1.4e-11	3.6e-12	2.0e-04	4.8e-05	1.2e-05	-1.2e-09	-3.0e-10	-7.4e-11
1e-9	5.7e-14	1.4e-14	3.6e-15	2.0e-04	4.8e-05	1.2e-05	-1.2e-12	-3.0e-13	-7.4e-14
1e-12	5.7e-17	2.0e-17	1.7e-17	2.0e-04	4.8e-05	1.2e-05	-8.8e-16	-3.1e-17	-2.0e-16
1e-15	3.0e-17	4.6e-17	7.1e-18	2.0e-04	4.8e-05	1.2e-05	2.1e-16	2.3e-17	8.6e-17

The behavior of the final Mach number, M_f, is similar to the tests with constant reconstruction. However, the actual values are much lower for the second-order scheme. Even the unpreconditioned flux reaches fairly low Mach numbers. Interestingly, the final Mach numbers decrease by a factor of > 7 as the grid resolution is doubled in this case, whereas only a factor of four would be expected for a second-order scheme. When flux preconditioning is switched on, the final Mach numbers again decrease with the cut-off Mach number and quickly approach values of the order of the machine precision. Doubling the grid resolution decreases the final Mach numbers by roughly a factor of four in this case, which is the expected behavior.

The results for the temperature distortions and its temporal derivative show the same behavior as the Mach number. The actual values are again much lower compared to the results with constant reconstruction and show the same scaling with the grid resolution as the final Mach numbers. Thus, even the unpreconditioned numerical flux is roughly able to maintain the hydrostatic equilibrium with linear reconstruction, although the deviations are still a bit too high for practical applications, where a long-term stability is very desirable. This goal is quite accurately reached when the preconditioned flux is used with cut-off Mach numbers of the order of $\approx 10^{-10}$.

For the next numerical tests the F-wave Riemann solver described in Section 6.1.2 was used instead of the Roe-Lowmach flux. It should be noted that this scheme is only first order accurate in space as it can only be used with constant reconstruction at the moment. Thus, the results presented in the following table should be compared to first order results shown above.

M_{cut} \ N_x	M_f			σ_T			$\dot{\sigma}_T$		
	64	128	256	64	128	256	64	128	256
1e0	7.8e-16	6.9e-16	8.5e-16	4.1e-04	1.0e-04	2.5e-05	1.2e-15	1.7e-15	2.7e-16
1e-3	2.5e-08	6.2e-09	1.5e-09	5.4e-05	1.4e-05	3.5e-06	-9.8e-08	-2.5e-08	-6.4e-09
1e-6	3.1e-11	7.8e-12	1.9e-12	9.3e-05	2.4e-05	5.9e-06	-6.2e-10	-1.6e-10	-3.9e-11
1e-9	3.1e-14	7.8e-15	1.9e-15	9.3e-05	2.4e-05	5.9e-06	-6.2e-13	-1.6e-13	-3.8e-14
1e-12	7.1e-17	1.4e-16	2.3e-17	9.3e-05	2.4e-05	5.9e-06	-9.2e-16	2.3e-16	2.3e-16
1e-15	4.0e-17	3.6e-17	9.5e-17	9.3e-05	2.4e-05	5.9e-06	-5.7e-16	2.2e-16	1.3e-15

The behavior of the F-wave Riemann solver is especially interesting when it is used without the flux preconditioning. The final Mach numbers quickly reach machine precision, whereas the temperature distortions decreased by a factor of 10^3 compared to the results with the Roe-Lowmach flux. Moreover, σ_T decreases by a factor of four as the grid resolution is increased although the method is only first order accurate in space. The temporal derivative of the temperature distortions reaches machine precision with every grid resolution. Thus, it should be possible to maintain this discrete hydrostatic equilibrium forever. Only the temperature profile gets slightly disturbed during the relaxation process. From these results, the F-wave Riemann solver seems to be indeed well-balanced.

From the above table, it can be seen that the final Mach numbers do *not* reach machine precision anymore, if the flux preconditioning is used with a cut-off Mach number of $M_{\text{cut}} = 10^{-3}$. Although the temperature deviations are slightly smaller than with the unpreconditioned flux, the corresponding temporal derivative is substantially higher. Thus, the flux preconditioning somewhat conflicts with well-balancing provided by the basic method. However, the situation changes again as the cut-off Mach number is further decreased. For $M_{\text{cut}} < 10^{-9}$ the preconditioned F-wave Riemann solver reaches the same accuracy as in the unpreconditioned case.

For the last group of tests, the hydrostatic reconstruction presented in Section 6.1.3 which is also first order accurate is used in combination with the Roe-Lowmach flux. The following table summarizes the results:

M_{cut} \ N_x	M_f			σ_T			$\dot{\sigma}_T$		
	64	128	256	64	128	256	64	128	256
1e0	3.7e-16	3.4e-16	5.4e-16	2.4e-05	6.0e-06	1.5e-06	7.6e-16	4.0e-16	1.1e-16
1e-3	1.2e-09	3.1e-10	7.6e-11	2.6e-05	6.3e-06	1.6e-06	-3.1e-08	-7.6e-09	-1.9e-09
1e-6	1.5e-12	3.8e-13	9.5e-14	2.9e-05	7.2e-06	1.8e-06	-3.9e-11	-9.7e-12	-2.4e-12
1e-9	1.6e-15	3.8e-16	3.8e-16	2.9e-05	7.2e-06	1.8e-06	-3.8e-14	-1.0e-14	2.9e-17
1e-12	1.7e-16	6.6e-16	1.9e-16	2.9e-05	7.2e-06	1.8e-06	8.0e-16	-1.6e-17	4.3e-18
1e-15	1.5e-16	5.1e-16	4.4e-16	2.9e-05	7.2e-06	1.8e-06	5.8e-16	7.8e-16	9.4e-16

Although the hydrostatic reconstruction is very different from the approach with the F-wave Riemann solver, the results and the overall behavior is very similar. The scheme seems to be well-balanced for the unpreconditioned Riemann solver, while preconditioning requires low cut-off Mach numbers in order to reach this goal. Moreover, the temperature deviations are even slightly smaller for the hydrostatic reconstruction compared to the F-wave Riemann solver.

In order to conclude this section, a few results should be recapitulated. First of all, the use of a pointwise discretization of gravity combined with an unpreconditioned Riemann solver does not seem to suitable for nearly hydrostatic flows. The residual Mach numbers and the temperature deviations from the initial conditions are too high for practical applications in the low Mach number regime. Contrary to that, using the flux preconditioned Roe-Lowmach solver with very low cut-off Mach numbers, the hydrostatic equilibrium can be maintained very accurately for very long time scales. Thus, the method seems to be mostly well-balanced, although no formal proof on this property can be given. The F-wave Riemann solver and the hydrostatic reconstruction, which are intended to be well-balanced show very promising results in maintaining the hydrostatic

atmosphere. If these methods are used in combination with flux preconditioning, the cut-off Mach number needs to very low in order to maintain this property. Moreover, it is expected that an extension of these methods to second-order accurate versions would further improve their ability to maintain hydrostatic temperature profile with even greater accuracy. It should further be noted that the flux preconditioning is expected to be very important although the one-dimensional results do not indicate a need for them, because the unpreconditioned well-balanced methods also provide very accurate results. However, in multi-dimensional simulations, it is expected that the numerical viscosity for low Mach number flows is still too high with an unpreconditioned flux function, especially in the directions normal to the gravitational acceleration which are not modified by the special treatment of the gravitational source term.

6.3 Stability of 2D hydrostatic atmospheres

6.3.1 Basic setup

In the last section the ability of several numerical discretizations to maintain a hydrostatic equilibrium state was investigated. Here, these tests are extended to a two-dimensional setup in order to check whether the conclusions drawn in the last section also hold in this case.

For the setup considered here, the same assumptions on the gas properties and the gravitation as in Section 6.2 are chosen. The gravitational acceleration is assumed to be aligned with the y-axis of a Cartesian coordinate system. Moreover, similar reference quantities for the non-dimensionalization are used here (with the exception that $\hat{\alpha}/\hat{w}$ has to be replaced by $\hat{\alpha}$ in Equation 6.2.5 in the following for consistency). However, a different vertical temperature profile is chosen here as it illustrates the effects presented in this section more clearly. It is assumed that the temperature varies linearly with height and is thus given by

$$\hat{T} = 1 + \hat{\alpha} \cdot \hat{y}, \tag{6.3.1}$$

where the parameter $\hat{\alpha}$ denotes a non-dimensional temperature gradient. For this profile, the hydrostatic balance equation 6.2.2 can also be integrated analytically. The resulting hydrostatic pressure is given by

$$\hat{p} = (1 + \hat{\alpha} \cdot \hat{y})^{-1/\hat{\alpha}}. \tag{6.3.2}$$

The corresponding density profile is computed by the ideal gas law. These profiles are used as initial conditions for the following simulations. Note that the profiles have no dependence on the horizontal coordinate $\hat{x}$ and are thus constant in this direction. The initial velocity field is set to zero.

The Cartesian computational domain is chosen to $[-0.1, 0.1] \times [-0.1, 0.1]$ which is discretized by $N_x \times N_y$ grid cells. If not stated otherwise, $N_x = N_y = 64$ is used. Note that length scales are measured in units of the pressure scale height at $\hat{y} = 0$. The vertical extent of the domain is thus relatively small. However, for the temperature gradient $\hat{\alpha} = 4.0$ is chosen which is quite large. The initial vertical density, pressure and temperature profiles for this setup are illustrated in Figure 6.3. The boundary conditions are periodic in the horizontal direction, while constant ghost cells with the hydrostatic values are used in the vertical direction. So far, the setup is perfectly symmetric with respect to the horizontal direction. Thus every column should exactly behave like a one-dimensional hydrostatic atmosphere as no fluxes can be generated in the horizontal direction, because the velocity field is zero everywhere. However, this symmetry between the grid cells in a horizontal layer is purely numerical as it is caused by the alignment of the gravitational acceleration with the $\hat{y}$ axes. Using a slightly rotated coordinate system would, for example, break up this numerical symmetry. Here, the symmetry is broken by adding a tiny random velocity field in the horizontal direction, where the corresponding Mach numbers are limited to $\pm 10^{-15}$. The velocity noise is thus just slightly above the machine precision. It should also be noted that noise of such a level could also be instantaneously created during (more practical) simulations,

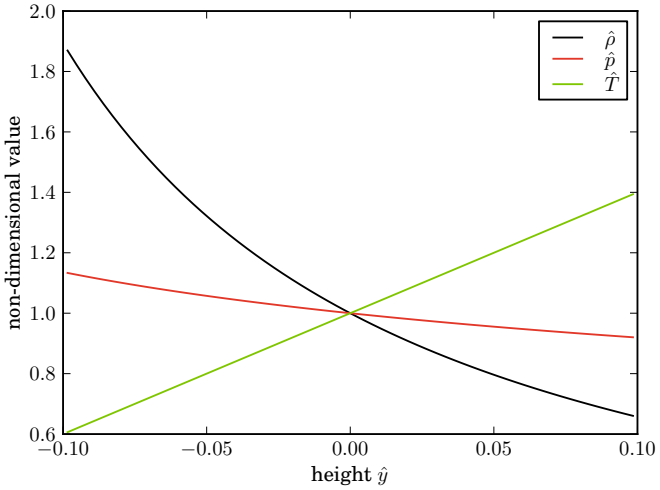


Figure 6.3: Initial density, pressure and temperature profiles

e.g. by numerical round-off errors, more complicated equations of state which need some kind of internal iteration or errors resulting from the imperfect solution of the non-linear systems for implicit time stepping. In total, the addition of the initial velocity noise is well justified. Ideally, the noise should just stay at its initial level or even damp out again and thus should not affect the simulation results at all.

6.3.2 Hydrostatic numerical instability

The first numerical tests of the two-dimensional hydrostatic atmosphere are performed with the implicit backward Euler time stepper at a fixed time step of $\Delta\hat{t} = 0.05$ until $\hat{t}_{\max} = 50$ is reached. The gravitational source term is discretized with the pointwise method. The numerical flux is computed with the Roe-Lowmach method, first without flux preconditioning ($M_{\text{cut}} = 1$) and then with flux preconditioning at $M_{\text{cut}} = 10^{-12}$. In the latter case, the hydrostatic atmosphere is expected to be very stable according to the one-dimensional results from the last section. For both cut-off Mach numbers, a constant (first order) and a linear (second order) interface reconstruction is tested. In order to analyze the results, the maximum Mach number (over all grid cells) is computed as a function of time. The results for the four test cases are illustrated in Figure 6.4.

For the unpreconditioned numerical fluxes, the behavior of the maximum Mach number is very similar to the one-dimensional results. In the first time step, the Mach number increases to a certain value and basically stays at this level as the numerical scheme is not well-balanced at all. Similarly to the one-dimensional results, the linear reconstruction performs much better as it achieves lower Mach numbers. It should also be noted that the absolute values of the final Mach numbers are somewhat smaller here than in the one-dimensional tests. This can be attributed to the fact that the vertical extent of the domain is by a factor of ten smaller here leading to a better resolution of the initial vertical profiles relative to the pressure scale height.

On the contrary, it can be seen from Figure 6.4 that the temporal evolution of the maximum Mach number changes dramatically compared to the one-dimensional results if flux preconditioning is switched on. In the first time step, the Mach number also increases to a certain value and decreases again until $\hat{t} \approx 2$ where it reaches fairly low values. However, instead of staying at this low level, the Mach number starts growing exponentially. For the constant interface re-

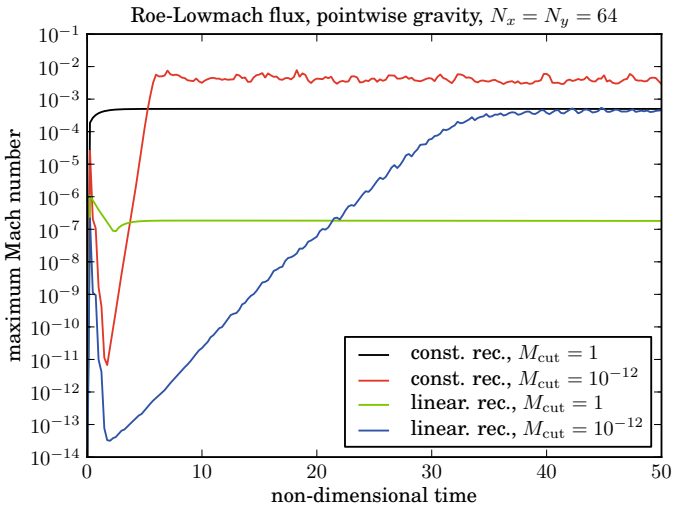


Figure 6.4: Temporal evolution of the maximum Mach number

construction, this exponential growth is very fast until it reaches a saturation level at $\approx 10^{-2}$. For the linear interface reconstruction, the behavior is similar although the growth rate and the saturation level are somewhat smaller.

The unexpected growth of the Mach number for the preconditioned fluxes was not observed in one-dimensional tests. It can be suspected that the growth is related to the new horizontal dimension in space. In order to analyze this behavior in a more detailed way, the *horizontal average* of a quantity s is defined in the following way:

$$< s >_{i,j}^{\text{hor}} = \frac{1}{N_x} \sum_{i'=1}^{N_x} s_{i',j} \tag{6.3.3}$$

The quantity s can be any variable that is defined on the grid cells, like the density, temperature or pressure. With this average, relative *horizontal fluctuations* of a quantity can measured by

$$\text{hf}_{i,j}(s) = \frac{s_{i,j} - < s >_{i,j}^{\text{hor}}}{< s >_{i,j}^{\text{hor}}}. \tag{6.3.4}$$

The horizontal density and temperature fluctuations are plotted in Figure 6.5 for the test with constant reconstruction and $M_{\text{cut}} = 10^{-12}$ at $\hat{t} = 2.5$ where the maximum Mach number just starts to grow exponentially. It can be seen that the fluctuations show a very pronounced checkerboard-like structure which is nearly equal in the density and the temperature field. Figure 6.7 (left panel) shows the temporal evolution of the maximum of the horizontal density, temperature and pressure fluctuations compared to the maximum Mach number. It can clearly be seen that all horizontal fluctuations grow exponentially until a certain saturation level is reached. Interestingly, the amplitude of the Mach number starts to rise at exactly the same point where the density (or temperature) fluctuations reach a similar amplitude. Thus, the growth of the velocity field seems to be triggered by the density fluctuations. The pressure fluctuations are generally a bit smaller in magnitude than the other quantities, but show qualitatively the same behavior. At time $\hat{t} = 10$, where the saturation level is reached in all quantities, the density fluctuations and the Mach number distribution is shown in Figure 6.6. Here, the checkerboard structures have vanished and some large-scale chaotic velocity field with corresponding density fluctuations has established. As these motions do not damp out during the simulation time, the atmosphere starts to build

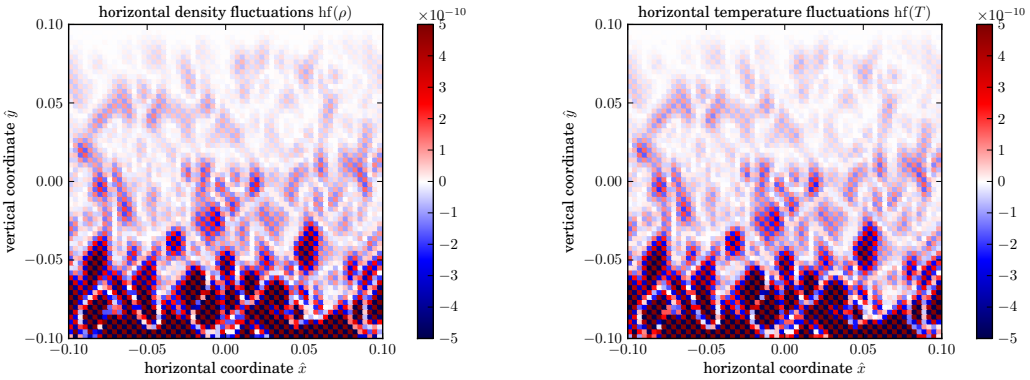


Figure 6.5: horizontal density and temperature fluctuations at $\hat{t} = 2.5$ with $M_{\text{cut}} = 10^{-12}$ and constant reconstruction

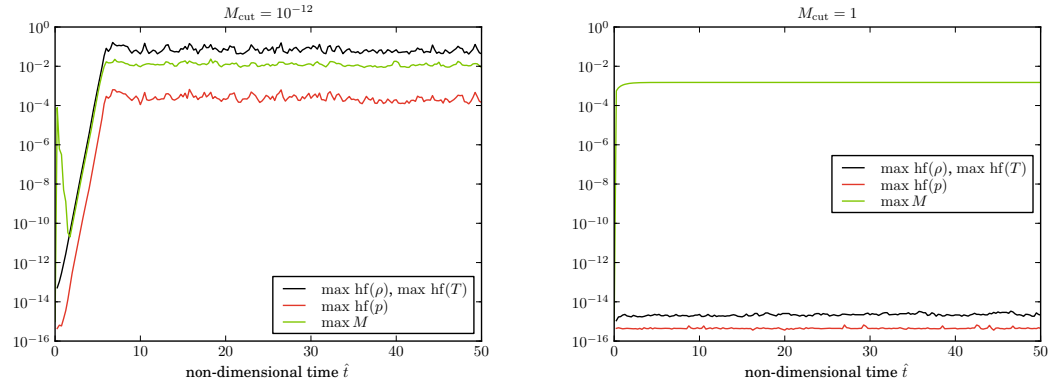


Figure 6.6: temporal evolution of horizontal fluctuations for the preconditioned (left) and the unpreconditioned (right) flux with constant reconstruction

up large scale motions which mix the atmosphere on relatively short time scales. The initial hydrostatic equilibrium thus gets completely destroyed. This behavior is clearly unphysical and can not be explained by the continuous Euler equations. The checkerboard-like structures during the exponential growth of the horizontal fluctuations is also very typical for a *numerical instability*. Such instabilities are usually powered by non-linear interactions between waves which produce higher-order harmonics that cannot be resolved on the grid anymore (see e.g. Riddaway & Hortal 2001; Hirsch & Hirsch 1991). These harmonics thus get aliased and may lead to a resonant self-interaction if the numerical discretization does not prevent such a behavior. In Figure 6.6 (right panel) the maximum horizontal fluctuations and the maximum Mach number are also plotted for the test with the unpreconditioned Roe-Lowmach solver. There, the horizontal fluctuations stay at a constant level near the machine precision and do not show any sign of a growth. The Mach number saturates at $M \approx 10^{-3}$ which is compatible with the one-dimensional tests. Thus, the velocity field stays nearly vertically orientated in this case.

From the tests presented in this section, the preconditioned Roe-Lowmach flux seems to be the origin of the numerical instability at first sight. However, such a behavior never showed up during the development of this method and the tests presented in the previous chapters if the numerical flux is used in the absence of the gravitational source term. Therefore, the discretization of gravity

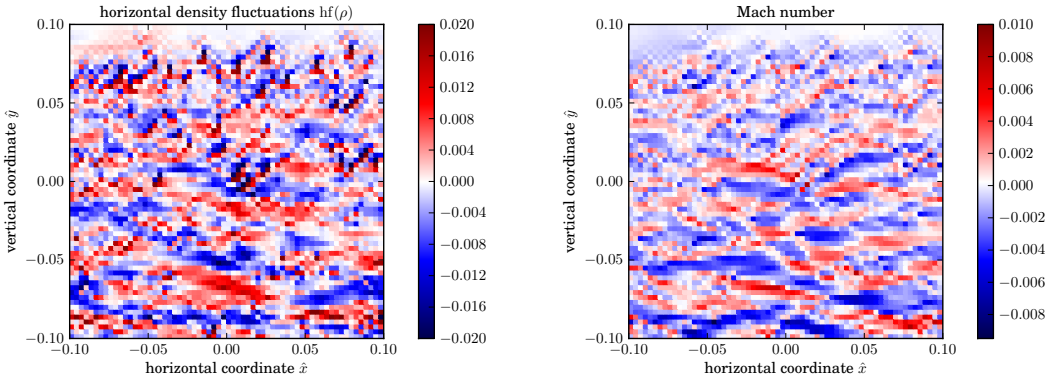


Figure 6.7: horizontal density fluctuations and Mach number at $\hat{t} = 10$ with $M_{\text{cut}} = 10^{-12}$ and constant reconstruction

and its interplay with the numerical flux function may also be potential sources for the numerical instability. Moreover, it is not very remarkable the unpreconditioned Roe solver is numerically stable as it suffers from an unphysically high artificial viscosity in the low Mach number regime (cf. Chapter 4) which effectively damps out any kind of fluid motion. For these reasons, the stability of the hydrostatic atmosphere is examined with several different numerical discretizations in order to shed some light on the origins of the numerical instability.

6.3.3 Different numerical discretizations

In the last subsection a numerical instability was discovered in a hydrostatic atmosphere in combination with the Roe-Lowmach flux. It has been shown that the instability develops with constant and linear reconstruction. It can thus be excluded that the instability is caused by a missing limiter on the linear reconstruction. Moreover, the instability was even more pronounced for constant reconstruction. This subsection investigates several modifications of the first-order discretization only in order to gain some more insight in the behavior of the numerical instability. This is done by a series of numerical experiments where only one part of the original discretization from the last subsection is modified. Other methods and parameters, not mentioned in a certain test case are left unmodified compared to the original setup. Moreover, it can be seen from the last subsection that the numerical instability can be identified by analyzing the horizontal fluctuations of the density, temperature or pressure field. As these fluctuations grow qualitatively in the same way for each field, only the behavior of the horizontal density fluctuations are presented in the following tests in order to identify the numerical instability.

The first interesting question concerning the numerical instability is the influence of the cut-off Mach number of the Roe-Lowmach flux. With $M_{\text{cut}} = 10^{-12}$ the setup was unstable, whereas the original Roe flux corresponding to $M_{\text{cut}} = 1$ is stable. Therefore, it can be speculated that limiting the flux preconditioner to some intermediate cut-off Mach number might keep the hydrostatic atmosphere stable. In order to test this hypothesis, the original setup is simulated with several different values for M_{cut}. The results are presented in Figure 6.8 (left panel). It can be seen that limiting the flux preconditioner to $M_{\text{cut}} = 10^{-3}$ has nearly no influence on the numerical instability, whereas a value of $M_{\text{cut}} = 10^{-2}$ slightly decreases the growth rate. Stability on the time-scale of the simulation can be achieved if the cut-off Mach number is further increased to $M_{\text{cut}} = 10^{-1}$. However, such a high value for the cut-off Mach number contradicts the aim of the Roe-Lowmach flux to reduce the artificial viscosity for low Mach number flows. With $M_{\text{cut}} = 10^{-1}$ velocity fields below this Mach number would get damped out on relatively short time scales. In total, limiting the cut-off Mach number does not seem to be a reasonable way to

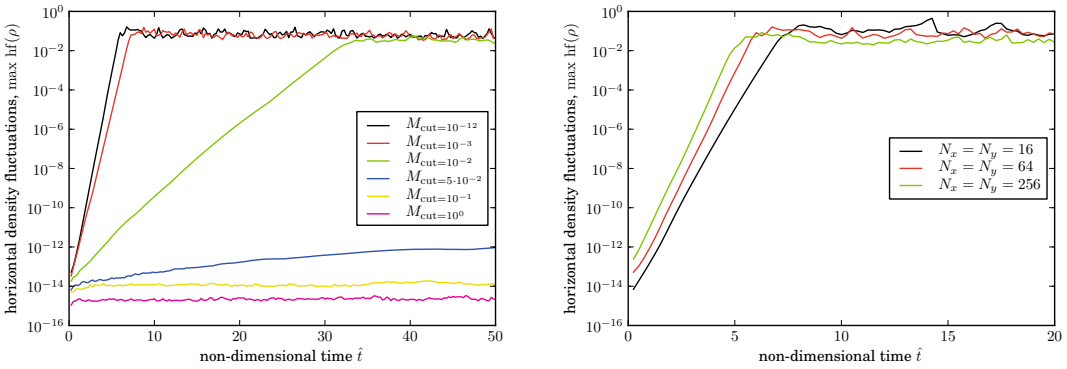


Figure 6.8: horizontal density fluctuations for different cut-off Mach numbers (left) and different grid resolutions (right)

inhibit the numerical instability.

In the next numerical experiment, the grid resolution is varied in order to see if the numerical instability is caused by a physical process which is not resolved accurate enough in the original setup. In order to investigate this, additional simulations with $N_x = N_y = 16$ and $N_x = N_y = 256$ were performed whose results are shown in Figure 6.8 (right panel). It can be seen that the numerical instability grows nearly in the same way for every grid resolution. If the instability would be caused by an unresolved process, it would be expected that the instability becomes weaker as the grid gets finer. However, the numerical instability grows even a bit faster at the highest resolution compared to the lowest resolution here. It can thus be concluded that the grid spacing only has a minor influence on the numerical instability for this test case.

In a next step, the influence of different numerical flux functions on the stability of the hydrostatic atmosphere is investigated. However, only 'low Mach number' flux functions are considered here as they do not show an excessive artificial viscosity in the low Mach number regime. Specifically, the central flux (cf. Section 3.5.3), the Roe flux in combination with low Mach number reconstruction (cf. Section 4.3.6) and the AUSM$^+$-Lowmach (cf. Section 4.3.5) are compared to the Roe-Lowmach flux. The results on the horizontal density fluctuations are shown in Figure 6.9 (left panel). It can be seen that the Roe flux with low Mach number reconstruction quickly becomes unstable and even results in a failure of the simulation due to the occurrence of NaN values. The central flux shows a similar behavior, although the growth rate of the instability is slightly lower. The central flux is generally known to be numerically unstable, although this effect is usually not as severe for implicit calculations as for explicit ones. However, it has been used for testing purposes here in order to demonstrate that the hydrostatic numerical instability also develops in the same manner as for other flux functions. As the central flux does not contain any upwinding terms, this is at least a small hint that the numerical instability is not directly caused by a deficiently modeled upwinding term within the other numerical flux functions. Moreover, the simulation results show that the numerical instability is somewhat inhibited with the AUSM$^+$-Lowmach flux compared to the Roe-Lowmach flux. However, the growth rate and the saturation level are still too high for practical applications. In total, it can be stated that the hydrostatic numerical instability develops with any 'low Mach number' flux function tested here. Particularly, the instability is not specific to the Roe-Lowmach flux, which was derived within this work in a somewhat ad-hoc fashion. Especially the AUSM$^+$-Lowmach flux shows a similar behavior but is based on a very different discretization concept. Therefore, the above results seem to indicate that the numerical instability *appears* solely due to the reduction of the artificial viscosity by the 'low Mach number' fluxes. However, it should be noted that this does not necessarily imply that these flux functions are also the direct source of the numerical instability.

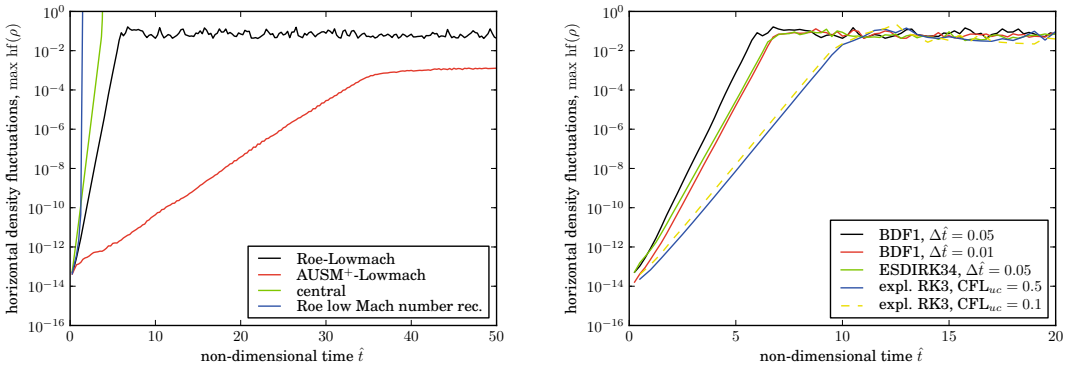


Figure 6.9: horizontal density fluctuations for different numerical flux functions (left) and different time stepping methods (right)

For the next numerical experiments, the influence of different temporal discretizations is examined. To this end, the size of the time step of the backward Euler method (BDF1) is first reduced from $\Delta\hat{t} = 0.05$ to $\Delta\hat{t} = 0.01$. Besides that, the hydrostatic setup is evolved with the implicit ESDIRK34 method at $\Delta\hat{t} = 0.05$. Moreover, the explicit 3-stage Runge-Kutta (RK3) is tested with an acoustic CFL numbers of 0.5 and 0.1 which corresponds $\Delta\hat{t} \approx 1.7 \cdot 10^{-4}$ and $\Delta\hat{t} \approx 3.4 \cdot 10^{-5}$. The results on the evolution of the horizontal density fluctuations are shown in Figure 6.9 (right panel). It can be seen that the numerical instability still grows in all test cases. The BDF1 method with the reduced time step and the ESDIRK34 shows nearly the same behavior with an exponential growth rate which is only slightly smaller than with the original setup. Although the instability grows somewhat slower with the explicit RK3 stepper at $\text{CFL}_{uc} = 0.5$, it can be seen that a further reduction of the time step has no influence on the stability of the hydrostatic atmosphere. These results show that the numerical instability is not caused by the use of implicit time stepping methods nor by an inaccurate solution of non-linear and linear algebraic systems involved in the solution process.

The next numerical tests are performed with different vertical boundary conditions. Besides the constant ghost cells filled with the hydrostatic values from the original setup, solid wall and far-field boundary conditions are tested in several combinations. The results are shown in the left panel of Figure 6.10, where the legend first denotes the lower and then the upper boundary condition. It can be seen that the boundaries only have a minor impact on the growth of the horizontal density fluctuations and are thus not considered to origin of the numerical instability.

Finally, it is investigated whether the well-balanced discretizations of gravity presented in Section 6.1 are able to inhibit the numerical instability within the hydrostatic atmosphere. To this end, the hydrostatic reconstruction method is examined in combination with the Roe-Lowmach and the AUSM$^+$-Lowmach flux functions. Moreover, the preconditioned F-wave Riemann solver is compared to the original setup with the Roe-Lowmach flux and the pointwise discretization of gravity. The temporal evolution of the resulting horizontal density fluctuations is shown in Figure 6.10 (right panel) for each method. The F-wave solver and the hydrostatic reconstruction with the Roe-Lowmach method show a similar growth of the numerical instability which is somewhat smaller than with the original setup. With the hydrostatic reconstruction and the AUSM$^+$-Lowmach flux the instability is significantly reduced. Due to this very interesting behavior, this test case is further evolved in time until $\hat{t} = 400$ is reached. The result is illustrated in Figure 6.11. It can be seen that after 50 units of time, the growth rate of the numerical instability even flattens further down. However, the instability does not seem to be completely resolved. This is also confirmed by the plot of the horizontal density fluctuations at the end of the simulation. The checkerboard-like pattern, typical for numerical instabilities can also be recognized

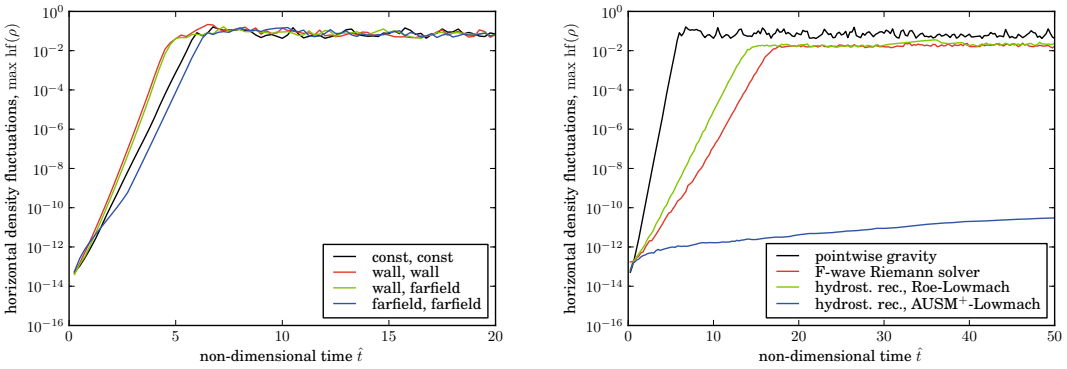


Figure 6.10: horizontal density fluctuations for different vertical boundary conditions (left) and different discretizations of gravity (right)

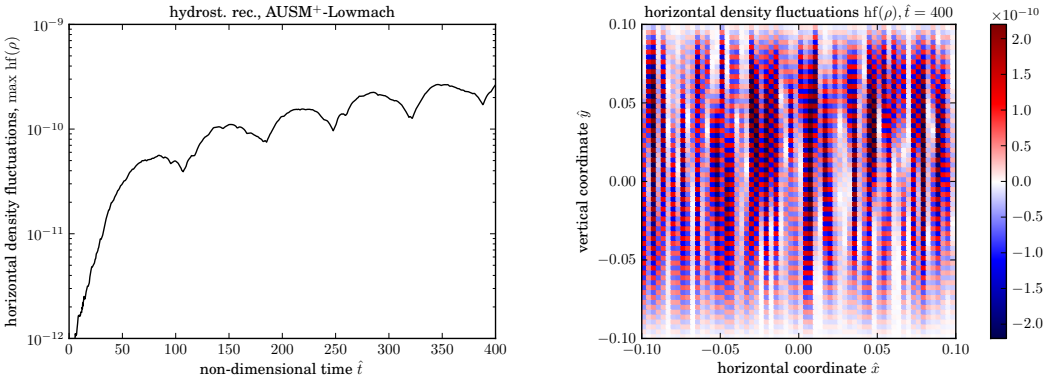


Figure 6.11: horizontal density fluctuations with hydrostatic reconstruction and the AUSM$^+$-Lowmach flux

in this test case. In total, the well-balanced schemes tested here, seem to have some potential in suppressing the numerical instability. However, none of them reaches this goal at the moment.

In summary, none of the variations of the numerical discretization tested in this subsection is able to resolve the problem of the hydrostatic numerical instability. It should be noted that in the tests presented above only one discretization feature was modified at each test. However, during the work for this thesis, many more combinations of numerical discretization method were tested, but none of them resolved the problem. Moreover, no particular part of the discretization can be identified here which is solely responsible for the numerical instability. It seems to be vital for the stability of the atmosphere that the numerical flux function and the discretization of the gravitational source term are properly adapted to each other. In this respect, the failure of the F-wave Riemann solver is somewhat disappointing, because the method approximates the solution of the *inhomogeneous* Riemann problem at each cell interface. Thus the numerical upwinding should reflect the coupled hydrodynamic and gravitational problem. On the one hand, it is still possible that the specific preconditioning approach to reduce the artificial viscosity in the low Mach number regime chosen here is not suitable for the problem. On the other hand, different low Mach number approaches which are based on different concepts show similar problems. In total, the exact reasons for the failure of the numerical discretization methods remain unclear.

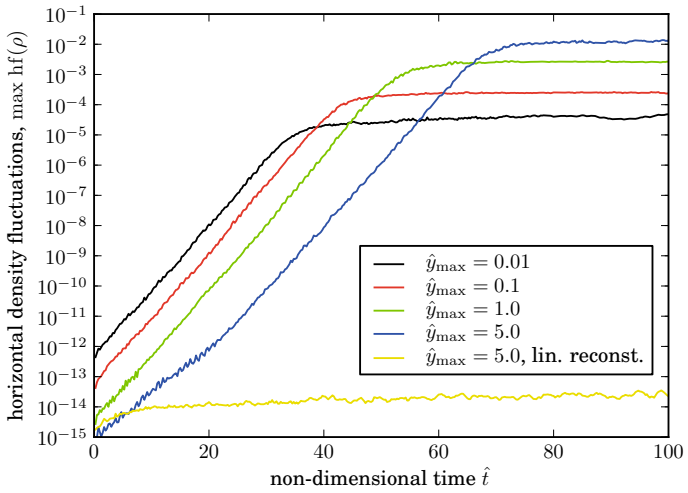


Figure 6.12: Horizontal density fluctuations for an isothermal atmosphere

6.3.4 Different atmospheres

In this subsection, it is shown that the numerical instability presented above is not bound to the specific choice of the hydrostatic atmosphere used in the previous tests. First, an isothermal atmosphere of an ideal gas with constant gravitational acceleration is considered. Using the same reference quantities for non-dimensionalization as in the last subsections (setting $\hat{\alpha} = 0$), the hydrostatic initial conditions can be written as

$$\hat{g}(\hat{x}, \hat{y}) = -1, \qquad \hat{T}(\hat{x}, \hat{y}) = 1, \qquad \hat{p}(\hat{x}, \hat{y}) = \hat{\rho}(\hat{x}, \hat{y}) = \exp(-\hat{y}). \tag{6.3.5}$$

Thus, the only parameter left for this setup is the size of the computational domain which is chosen to $[-\hat{y}_{\max}, \hat{y}_{\max}] \times [-\hat{y}_{\max}, \hat{y}_{\max}]$ here. Note that $\hat{y}_{\max}$ is measured in units of the pressure scale height, which is constant here. The domain is discretized by 64×64 grid cells with periodic horizontal and constant ghost cell vertical boundary conditions. The interface reconstruction is performed with constant extrapolation (if not stated differently) and the numerical fluxes are computed with the Roe-Lowmach method at $M_{\text{cut}} = 10^{-12}$. The initial horizontal velocity field is initialized with a tiny random noise as in the previous subsections. The setup is advanced in time with the implicit backward Euler method at fixed time step $\Delta\hat{t} = 0.05$. The temporal evolution of the horizontal density fluctuations is shown in Figure 6.12 for different sizes of the computational domain. It can be seen that the hydrostatic numerical instability also develops in all test cases considered here. The growth rates of the numerical instability are significantly lower compared to the setup with a linear temperature gradient from Section 6.3.1 and do not seem to depend on the box size. Thus, temperature gradients seem to strengthen the numerical instability. Moreover, it can be seen from the results, that the saturation level of the horizontal density fluctuations depends on the size of the computational domain. Smaller boxes tend to weaken the numerical instability. For the largest box size, where $2 \cdot 5 = 10$ pressure scale heights where resolved, the numerical test is also performed with linear interface reconstruction, which results in a second-order accurate scheme in space. The numerical instability is nearly completely inhibited in this case. In total, it can be stated that the numerical instability is also present in the most simple hydrostatic atmosphere, although the effects are much weaker than in the tests before.

The hydrostatic atmospheres considered before had only a model character to demonstrate the effects of the numerical instability but were not very realistic from the position of real stellar

atmospheres. A model of a main-sequence star with a total mass of 20 solar masses which was thankfully provided by Weiss (2009) is thus considered next. The model is an output of the 1D stellar evolution code GARSTEC (Weiss & Schlattl 2008) in which the star was evolved up to an age of 4.65 Myr. As the star is in its main-sequence stage, it basically consists of a core in which hydrogen (H) is converted to helium (He) by nuclear reactions, surrounded by a hydrogen rich envelope. At the age of the star considered here, the core is already substantially enriched by helium. A relatively small region of the star at the interface between the core and the envelope is chosen for the numerical tests in LHC. The corresponding radial profiles of density, pressure, gravity and mass fraction are mapped from the 1D stellar evolution code into a plain-parallel Cartesian grid in LHC by a seconder-order interpolation function. Each quantity is thus constant along the x-axes of the two-dimensional grid. The initial vertical profiles are plotted in the left panel of Figure 6.13, where the quantities are normalized to their values at the center (denoted by subscript c) of the box for better readability. It should be noted that the gravity is not spatially constant for this setup. In LHC, a fully ionized ideal gas with radiation pressure is used as equation of state (see Section 2.2.3). This allows to reproduce the temperature profile of the 1D stellar evolution code up to a relative accuracy of $\approx 10^{-3}$. The radiation pressure makes up about 10% of the total pressure within the box considered here and can thus not be neglected. Moreover, it should be noted that the setup should reflect a hydrostatic equilibrium state to very high accuracy. Within the stellar evolution model, the hydrostatic equilibrium may only change due to nuclear reactions and thermal radiation whose time scales are orders of magnitude higher than the time scales considered in the following tests.

The two-dimensional computational domain is chosen with an aspect ratio of $2 : 1$ and is discretized into 128×128 grid cells. For the further discretization, the interface values are reconstructed with piecewise linear functions here and the numerical fluxes are computed with the Roe-Lowmach method at $M_{\text{cut}} = 10^{-12}$. The initial hydrostatic atmosphere is evolved with the implicit backward Euler method. The time step is chosen by the free-fall CFL criterion described in Section 5.3.1 with $\text{CFL}_{ug} = 0.5$ which corresponds to a time step of $\Delta t \approx 41.3\,\text{s}$. In order to test the stability the hydrostatic atmosphere, the maximum Mach number and the maximum horizontal density fluctuations on the two-dimensional grid are recorded as a function of time. The results are illustrated in the right panel of Figure 6.13. It can be seen that the horizontal density fluctuations also grow exponentially in this test case and reach their saturation level after a physical simulation time of approximately two hours only. The spatial distribution of the horizontal density fluctuations is shown in Figure 6.14 at a certain instant in time. A checkerboard-like pattern can also be observed in this case. Interestingly, this numerical instability is concentrated at the vertical center of the box and is very closely aligned with the gradients of the mass faction of hydrogen and helium. Thus, besides the temperature gradients discussed before, concentration gradients also seem to support the numerical instability. Moreover, it can be seen from Figure 6.13 that the maximum Mach number increases to about $5 \cdot 10^{-4}$ within the first time step and slowly decreases for some time as the atmosphere tries to relax to its discrete equilibrium state. However, the Mach number rises again with the density fluctuations and finally reaches a saturation level of $\approx 5 \cdot 10^{-2}$ at the end of the simulation. Such high velocity fields are certainly too large to study any real physical processes which might occur in this atmosphere. Plotting the spatial distribution of the hydrogen mass fractions at the end of the simulation (see right panel of Figure 6.14), it can also be seen that the abundances heavily start to mix. The initially hydrostatic atmosphere thus gets completely destroyed solely due the numerical instability. In total, this test demonstrates that better discretization techniques are needed in order to study physical processes in stellar atmospheres. Moreover, it should again be emphasized that switching to numerical flux functions not designed for low Mach number flows may suppress the numerical instability reported here. However, the excessive artificial viscosity involved in these methods would also suppress important physical processes.

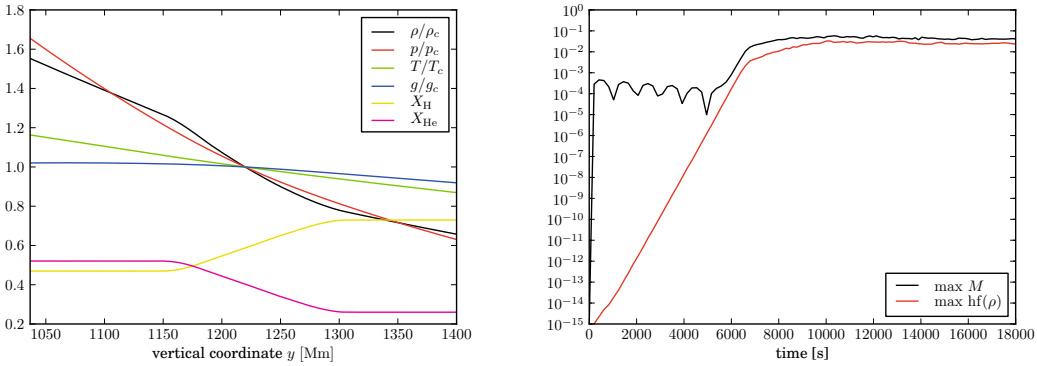


Figure 6.13: main sequence star: initial vertical profiles (left) normalized to their value at the center of the box; temporal evolution of the maximum Mach number and horizontal density fluctuations (right)

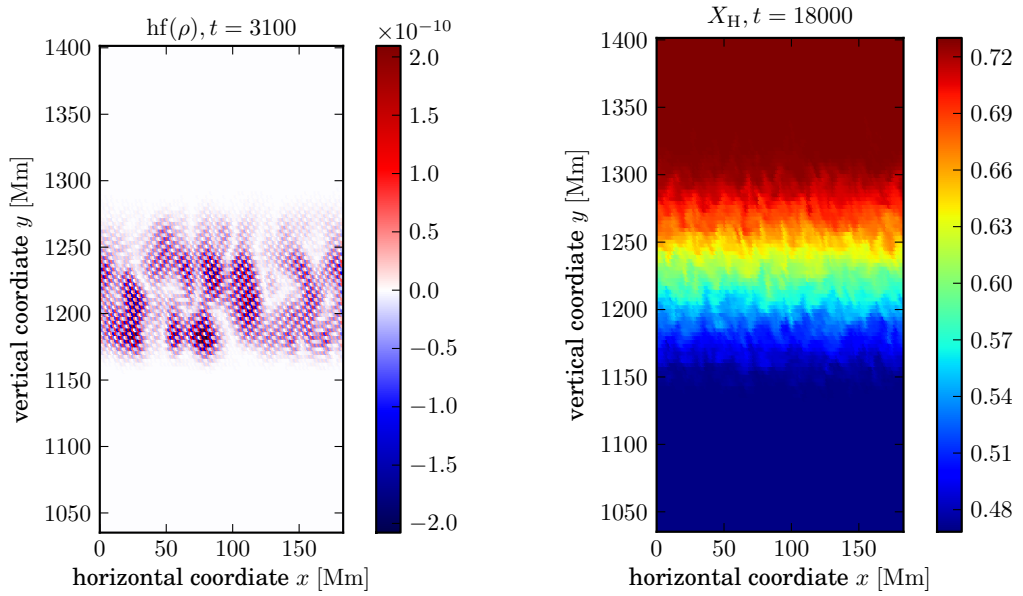


Figure 6.14: main sequence star: horizontal density fluctuations during the growth of the numerical instability (left); hydrogen mass fraction at the end of the simulation time (right)

6.3.5 Artificial physical viscosity

In Section 6.3.3, it was shown that the hydrostatic numerical instability can not be suppressed sufficiently by any of the discretization methods available in LHC so far. A general method to stabilize numerical discretizations in fluid mechanics is by adding some amount *physical* viscosity. Here, this approach seems to be somewhat counterproductive because the only aim for the invention of the low Mach number numerical functions was to reduce the artificial viscosity which is normally inherent in ordinary schemes for low Mach number flows. However, adding physical viscosity to the fluid might inhibit the hydrostatic numerical instability, while having full control on the amount of viscosity. Moreover, an additional viscosity does not depend on the Mach number of the flow (as it is the case for an unpreconditioned Roe solver). It is tested in this subsection whether such an *artificial physical viscosity* is able to stabilize a hydrostatic equilibrium.

The additional viscous source term to the Euler equations is taken from the *incompressible* Navier-Stokes equations. Although such a term is only strictly physically valid for divergence-free flows, it is a good approximation of the compressible viscosity in the low Mach number regime. Apart from that, the additional viscosity is artificial anyway and is only intended to suppress numerical oscillations. The incompressible viscous source term may be written in primitive variables for the velocity equations as (e.g. Drikakis & Rider 2005)

$$\mathbf{S}^{\mu}_{\mathbf{V},\mathrm{mom}} = \mu/\rho \; \nabla \cdot \nabla \mathbf{q}. \tag{6.3.6}$$

Assuming a two-dimensional Cartesian geometry, this source term may be rewritten in conservative variables in divergence form:

$$\mathbf{S}^{\mu} = \frac{\partial \mathbf{F}^{\mu}_{x}}{\partial x} + \frac{\partial \mathbf{F}^{\mu}_{y}}{\partial y} \tag{6.3.7}$$

The viscous fluxes are then defined by:

$$\mathbf{F}^{\mu}_{x} = \mu \begin{pmatrix} 0 \\ \partial_x u \\ \partial_x v \\ u \cdot \partial_x u + v \cdot \partial_x v \\ 0 \end{pmatrix}, \qquad \mathbf{F}^{\mu}_{y} = \mu \begin{pmatrix} 0 \\ \partial_y u \\ \partial_y v \\ u \cdot \partial_y u + v \cdot \partial_y v \\ 0 \end{pmatrix} \tag{6.3.8}$$

The variable μ denotes a *dynamic viscosity* which is assumed to be spatially and temporally constant in the following.

The divergence form of the viscous source term has the advantage that it is very easy to discretize in the finite-volume context used here. The viscous fluxes can be discretized similarly to the fluxes used for thermal radiation (cf. Section 3.5.4). The resulting numerical flux (exemplary for the x-direction)

$$\mathbf{F}^{\mu}_{x,i+1/2} = \frac{\mu}{\Delta x} \begin{pmatrix} 0 \\ u_{i+1} - u_i \\ v_{i+1} - v_i \\ \frac{(u_i+u_{i+1})}{2} (u_{i+1} - u_i) + \frac{(v_i+v_{i+1})}{2} (v_{i+1} - v_i) \\ 0 \end{pmatrix} \tag{6.3.9}$$

is second-order accurate in space and can simply be added to the numerical hydrodynamic fluxes at each cell interface. It should be noted that this numerical flux is only valid for two-dimensional Cartesian grids for dimensionalized calculations. More general cases are not yet implemented in LHC as the above formulation is sufficient for testing purposes.

With the above formulation of the viscous source term, the hydrostatic atmosphere of the main-sequence star presented in the last subsection is re-simulated. In doing so, the dynamic viscosity (measured in poise, $1\,\mathrm{P} = 1\frac{\mathrm{g}}{\mathrm{cm\ s}}$) is successively increased in each run until the hydrostatic atmosphere becomes stable. Figure 6.15 shows the corresponding evolution of the horizontal density

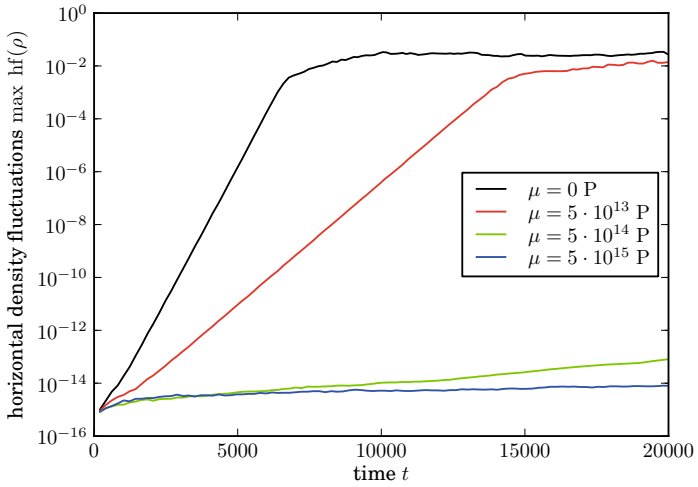


Figure 6.15: main sequence star: horizontal density fluctuations with different values for the dynamic viscosity

fluctuations compared to the original setup. It can be seen that a dynamic viscosity of $5 \cdot 10^{13}$ P is needed in order to substantially decrease the growth rate of the numerical instability. At $\mu = 5 \cdot 10^{15}$ the hydrostatic atmosphere finally becomes stable, at least on the time scales considered here. While such a high dynamic viscosity is clearly unphysical, as it rather corresponds to a solid material than to a gaseous fluid, it should be kept in mind that the viscosity is purely artificial here. It is intended to remove the density oscillations at the grid scale. Due to the relatively large grid spacing, the dynamic viscosity takes such high values. In order to get an impression of the actual amount of viscosity, a different measure is used in the following. To this end, it is assumed that the atmosphere is intended to be used to study large-scale flows at very low Mach number, like some wave-propagation phenomena, shear flows or convection. For such a flow, a dimensionless measure of the relative strength of viscosity, named *Reynolds number*, can be computed by

$$Re = \frac{\bar{\rho}\bar{q}L}{\mu} = \frac{\bar{\rho}\bar{M}\bar{c}L}{\mu}, \tag{6.3.10}$$

where the bars denote typical average quantities of the setup. Setting the typical length scale L to the vertical size of the computational domain and assuming a mean Mach number of $\bar{M} = 10^{-4}$ for the flows of interest, the Reynolds number for a dynamic viscosity of $\mu = 5 \cdot 10^{15}$ P results in

$$Re = 0.07. \tag{6.3.11}$$

In general, Reynolds numbers below unity denote flows in which even the inertial forces of the fluid elements are overwhelmed by viscous effects. For Reynolds numbers up to ≈ 1000 the flow regime is laminar whereas even higher Reynolds numbers describe turbulent flows. For the stellar atmosphere considered here, the viscosity required to stabilize the atmosphere is thus far to high for any practical application. Ideally, the turbulent flow regime should be reached in order to gain new insights into stellar evolutionary processes. As this is totally out of scope here, the use of artificial physical viscosity to resolve the hydrostatic numerical instability is dismissed in the following. In total, this test revealed how strong the numerical instability is for this atmosphere. Besides that, it is interesting to note that the atmosphere is numerically stable if the unpreconditioned numerical Roe flux is used. Thus, this test also gives an impression about

the strength of the artificial viscosity which is inherent in numerical fluxes not designed for low Mach number flows.

6.3.6 Fire Dynamics Simulator

After the presentation of the hydrostatic numerical instability in the last subsections, two questions may arise. First, is it possible that the various low Mach number numerical flux functions or even the basic discretization techniques used in LHC are just not suitable for hydrostatic atmospheres? Secondly, can the instability be caused by a simple programming bug in LHC? The second question seems somewhat unlikely as the numerical instability shows up with various different discretization techniques which are coded in well-separated modules of the source code. However, the possibility of a severe bug can never be excluded entirely.

In order to answer the two questions, a cross-check to the results of another simulation code which is based on very different discretization techniques is very helpful. In the search for such a different code a few technical requirements should be kept in mind. At first, the other simulation code should be based on the same set of continuous equations. For the hydrostatic atmospheres considered here, this means that at least two independent thermodynamic variables should be evolved in time (e.g. density and energy or pressure and temperature) and may contain arbitrary large fluctuations. This guarantees that the hydrostatic equilibrium is not bound to a predefined temperature profile. Moreover, the other code should have an accurate treatment of gravity. The last two requirements basically exclude all codes which make use of the Boussinesq approximation. Finally, a different code should resolve low Mach number flows accurately without an inherent excessive artificial viscosity.

To the author's knowledge, the only code which meets the above requirements and is additionally publicly available is the so-called *Fire Dynamics Simulator*[1] (FDS) developed by the National Institute of Standards and Technology (NIST). This code is intended to be used to simulate the propagation of fires inside buildings. The code is based on the compressible Navier-Stokes equations and has an accurate treatment of thermally-driven flows. Moreover, it seems to be well-balanced as it subtracts a hydrostatic equilibrium state from the governing equations and evolves the hydrostatic state and deviations from it separately. Moreover, the code employs a low Mach number assumption which is strictly valid only in the limit of $M \to 0$. This limits FDS to low-speed flows, which is sufficient for the tests considered here. As the flow field becomes divergence-free in the absence of source terms in FDS, the code should not contain an excessive artificial viscosity for low Mach number flows. In general, FDS is a finite-difference code with second-order accuracy in space and time. Further details about the numerical discretization can be found in the corresponding technical reference guide. It should be noted that FDS is used by a broad community and is subject excessive verification and validation tests. In total, FDS seems to be perfectly suitable for cross-checks against LHC.

As FDS is not intended to simulate large scale hydrostatic atmospheres, a very simple setup is chosen here in order to facilitate the grid setup. For convenience with FDS, this subsection also uses SI units. For the following numerical test an isothermal atmosphere of an ideal gas with a mean molecular weight of 1 g/mol at $T = 303.15$ K is considered under the action of the constant gravitational acceleration of $g = 9.81$ m/s^2. The two-dimensional Cartesian computational domain of size $[100\text{km}, 100\text{km}] \times [100\text{km}, 100\text{km}]$ is discretized with 64×64 nodes. The boundary conditions are solid walls at each side of the box. It should be noted that the more sophisticated features of FDS, like humidity, radiation, viscosity, turbulence models and chemical reactions are all switched off. Besides the initial hydrostatic atmosphere, a random velocity field with a peak Mach number of $\approx 10^{-14}$ is also added to this atmosphere in order to break up the perfect numerical symmetry in the horizontal direction. The setup is evolved in time until $t = 60000$ s is reached. As in the previous hydrostatic setups with LHC, the results are also analyzed by recording the maximum of the horizontal density fluctuations as a function of time.

[1] available via http://fire.nist.gov/fds/index.html

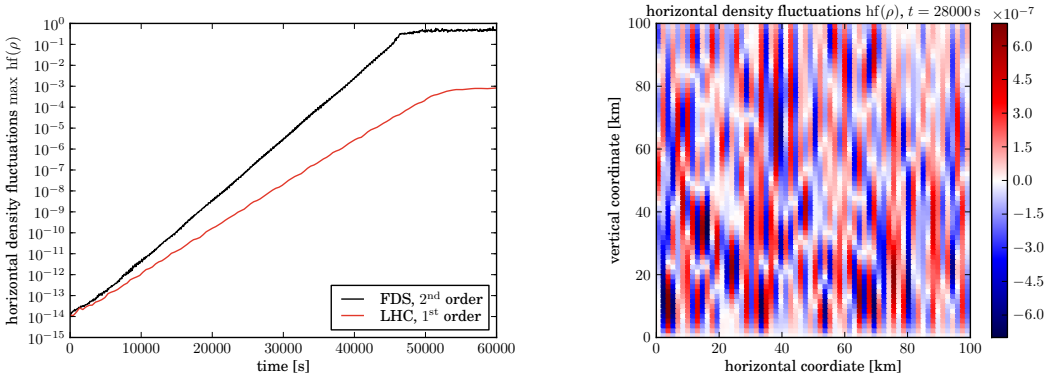


Figure 6.16: comparison of growth rates of the hydrostatic numerical instability (left); spatial structure of the numerical instability in FDS (right)

The results are illustrated in Figure 6.16 (left panel). For comparison, an identical hydrostatic atmosphere was evolved with LHC (Roe-Lowmach, constant reconstruction) which is also shown in the figure. It can be seen that the horizontal density fluctuations also grow exponentially in FDS. The hydrostatic numerical instability seems also to be present here. The corresponding spatial structure of the horizontal density fluctuations at a certain instant in time is shown in the right panel of Figure 6.16. Interestingly, the instability does not show a checkerboard-like pattern as in LHC but only a horizontal zig-zag pattern between the grid nodes. The growth rate of the numerical instability is even a bit higher in FDS than in LHC. It should be noted that the LHC results are first-order accurate in space. Using a linear interface reconstruction for second-order accuracy would nearly inhibit the numerical instability at all for this atmosphere.

In summary, it can be concluded that the hydrostatic numerical instability also shows up with FDS. The reason that this failure was not discovered before, is probably due to the fact that FDS is mainly used to simulate fires inside buildings. The typical height of computational domains is thus not greater than ≈ 500 m, which is only a tiny fraction compared to the corresponding pressure scale of 257 km. In order to answer the questions from the beginning of this subsection, it can be stated that the numerical methods used in LHC *may* be inappropriate for nearly hydrostatic flows. However, they may just be as inappropriate as the methods used by FDS which are very different. Thus, it seems unlikely that a *specific* choice of a certain discretization technique used in LHC is responsible for the hydrostatic numerical instability. Moreover, the cross-check with FDS revealed that it is very unlikely that the numerical instability is caused by a programming error in LHC.

6.3.7 Conclusions

In the following, the results from the previous sections are briefly recapitulated. Concerning the stability of one-dimensional hydrostatic atmospheres, it can be stated that well-balanced discretizations of gravity are clearly advantageous compared to a pointwise discretization of gravity. Using non low Mach number flux functions with pointwise gravity results in large residual Mach numbers and substantial distortions of the initial temperature profile in a hydrostatic atmosphere. Therefore, such a discretization is clearly not suited for nearly hydrostatic flows. However, using a low Mach number flux function in combination with pointwise gravity, the hydrostatic atmospheres seem to be nearly well balanced. The residual Mach number reach very low values down to the machine precision while distortions of the initial temperature profile are still in an acceptable range. Using well-balanced discretizations of gravity, the initial hydrostatic setup can be maintained with even greater accuracy. Low Mach number flux functions do not interfere with the well

balancing as long as the cut-off Mach number is set low enough. Thus, extending the well-balanced discretizations of gravity to more general situations, like second-order accuracy, non-ideal gases and non-Cartesian geometries should be a goal in the design of future developments in LHC.

Unfortunately, the hydrostatic numerical instability reported here limits the use of LHC to simple atmospheres at the moment, where the growth of the instability can be slowed down by a proper choice of discretization techniques such that the instability does not become dominant on the time scales of interest. It has been shown that the numerical instability occurs with all kinds of numerical discretizations suitable for low Mach number flows. From the results presented above, the numerical instability does not seem to be caused by an erroneous design of the low Mach number flux functions itself. The instability seems to emerge as soon as the numerical viscosity of ordinary flux functions is reduced in order capture low Mach number flows accurately. The specific choice the low Mach number technique only seems to have a minor influence on the stability, which is clearly emphasized by the results presented from the Fire Dynamics Simulator. It should be stressed that the use of a well-balanced discretization of gravity in combination with an ordinary numerical flux function does not seem to be an option for practical applications although hydrostatic atmospheres are stable in this case. The numerical viscosity, especially in directions perpendicular to the gravitational acceleration (which are not influenced by the well-balancing at all) is still excessive such that low Mach number flow fields are damped out on rather short time scales. Thus, for a proper modeling of turbulent stratified flows in stellar atmospheres, a low Mach number flux function seems to be vital. However, it has also been shown that certain combinations of well-balanced methods for gravity and low Mach number flux functions may substantially decrease the growth rate of the numerical instability. Thus, future research should probably concentrate on this topic in order to find a discretization of gravity and hydrodynamics which are more compliant with each other.

6.4 Internal gravity waves

6.4.1 Motion of a displaced fluid element

So far, only hydrostatic atmospheres and their stability in numerical simulations has been investigated. Assuming that a fluid element of density ρ_{int} of such an atmosphere is slightly vertically displaced from its equilibrium position r_0, it will fulfill an adiabatic motion described by the following equation (Maeder 2009):

$$\rho_{\text{int}} \frac{d^2 r}{dr^2} + g\left(\rho_{\text{int}} - \rho_{\text{ext}}\right) = 0 \tag{6.4.1}$$

The index ext denotes the density of the unperturbed atmosphere at the corresponding position. Linearizing the densities about the equilibrium position r_0, the equation of motion may be written as

$$\rho_{\text{int}} \frac{d^2 r}{dr^2} + g\left(\left.\frac{d\rho_{\text{int}}}{dr}\right|_{r_0} - \left.\frac{d\rho_{\text{ext}}}{dr}\right|_{r_0}\right)(r - r_0) = 0. \tag{6.4.2}$$

This differential equation just describes a harmonic oscillator with a frequency N defined by

$$N^2 = \frac{g}{\rho}\left(\left.\frac{d\rho_{\text{int}}}{dr}\right|_{r_0} - \left.\frac{d\rho_{\text{ext}}}{dr}\right|_{r_0}\right). \tag{6.4.3}$$

Thus, the displaced fluid element fulfills a periodic motion around its equilibrium position which is driven by the restoring force of gravity. The frequency N is called *Brunt-Väisälä-frequency* and is generally defined for an ideal gas as (Maeder 2009)

$$N^2 = \frac{g}{H_p}\left(\nabla_{\text{ad}} - \nabla_{\text{ext}} + \nabla_{\mu}\right), \tag{6.4.4}$$

where the variables

$$\nabla_{\mathrm{ad}} = \frac{d\ln T_{\mathrm{ad}}}{d\ln p} = \frac{\gamma - 1}{\gamma}, \qquad \nabla_{\mathrm{ext}} = \frac{d\ln T_{\mathrm{ext}}}{d\ln p}, \qquad \nabla_{\mu} = \frac{d\ln \mu_{\mathrm{ext}}}{d\ln p}, \tag{6.4.5}$$

denote the adiabatic, external logarithmic temperature gradients and the logarithmic gradient of the mean molecular weight. The index ext denotes the profiles of the unperturbed hydrostatic equilibrium state here. It should be noted that the Brunt-Väisälä-frequency may also contain a non-vanishing imaginary part if $N^2 < 0$. A displaced fluid element is then not forced back to its equilibrium position, but the displacement grows exponentially until non-linear processes stop this growth. This is the basic explanation for convection. The corresponding hydrostatic atmosphere is said to be *convectively unstable*. However, the atmospheres used in this work are all stable in this respect.

More importantly, it can be shown that oscillations of displaced fluid elements are also possible in directions not aligned with the direction of gravity (see e.g. Sutherland 2010). Only a purely horizontally displaced fluid element will not perform any periodic motions because there is no restoring force in this direction. Moreover, if the fluid is not displaced at a single point, but in a continuous way, it is possible that the oscillatory motions build up waves propagating through the hydrostatic atmosphere. These waves are called *internal gravity waves* and are described more mathematically in the following subsection.

6.4.2 Linear theory in Boussinesq approximation

Internal gravity waves are most easily described mathematically in the context of the Boussinesq approximation (cf. Section 2.7.1). The following derivation of the basic properties of internal gravity waves basically follows the presentation given in the book by Sutherland (2010) which is briefly repeated here. More details can be found in this reference. It is assumed that the fluid is described on a two-dimensional Cartesian domain in which the gravitational acceleration $g > 0$ is pointing towards the negative y-axes. The Boussinesq equations for the deviations from the hydrostatic equilibrium are then given by:

$$\frac{D\tilde{\vartheta}}{Dt} + v\frac{d\vartheta_{\mathrm{hse}}}{dy} = 0 \tag{6.4.6}$$

$$\frac{Du}{Dt} + \frac{1}{\rho_0}\cdot\frac{\partial\tilde{p}}{\partial x} = 0 \tag{6.4.7}$$

$$\frac{Dv}{Dt} + \frac{1}{\rho_0}\cdot\frac{\partial\tilde{p}}{\partial y} = -\frac{g}{\vartheta_0}\tilde{\vartheta} \tag{6.4.8}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{6.4.9}$$

For unknown variables, the ansatz of a two-dimensional, plain wave is chosen:

$$\begin{pmatrix} \tilde{\vartheta} \\ u \\ v \\ \tilde{p} \end{pmatrix} = \begin{pmatrix} A_{\vartheta} \\ A_u \\ A_v \\ A_p \end{pmatrix} \cdot \exp\left(i\left(k_x x + k_y y\right) - i\omega t\right) \tag{6.4.10}$$

The wave vector is denoted by (k_x, k_y) whereas the temporal frequency is denoted by ω. The amplitudes A may be arbitrary complex constants and thus permit a different phase in each component of the solution. The only requirement on the amplitudes is that they are reasonably small in magnitude as only small deviations of the hydrostatic equilibrium are subject of this

investigation. Inserting the ansatz into the Boussinesq system and omitting terms with products of the (small) amplitudes results in the following system of linear equations:

$$\begin{pmatrix} -i\omega & 0 & \frac{d\vartheta_{\mathrm{hse}}}{dy} & 0 \\ 0 & -i\omega & 0 & \frac{ik_x}{\rho_0} \\ \frac{g}{\rho_0} & 0 & -i\omega & \frac{ik_y}{\rho_0} \\ 0 & ik_x & ik_y & 0 \end{pmatrix} \cdot \begin{pmatrix} A_\vartheta \\ A_u \\ A_v \\ A_p \end{pmatrix} = 0 \tag{6.4.11}$$

These equations only have a non-trivial solution for the complex amplitudes if the determinant of the matrix is zero. Evaluating this expression results in the dispersion relation for Boussinesq internal gravity waves,

$$\omega^2 = -\frac{d\vartheta_{\mathrm{hse}}}{dy} \cdot \frac{g}{\rho_0} \cdot \frac{k_x^2}{k_x^2 + k_y^2}, \tag{6.4.12}$$

which relates the frequency ω to a given wave vector (k_x, k_y). This expression can further be simplified by recognizing that the first to factors just denote the square of the Brunt-Väisälä-frequency of the corresponding atmosphere. It should be noted that the N_0^2 is always positive here, because the gradient in potential temperature has to be negative for the atmosphere to be stable against convection. The last factor in the dispersion relation describes the square of the cosine of the angle of the wave vector with respect to the horizontal direction. The dispersion relation can thus be written as

$$\omega = N_0 \cos\Theta. \tag{6.4.13}$$

This relation has some remarkable properties. First, the frequency does not depend on the magnitude of the wave vector, but only on its angle with respect to the horizontal direction. Moreover, the propagation of internal gravity waves is strongly anisotropic. While horizontal plane waves oscillate at the maximum frequency, given by N_0, vertical plane waves do not propagate at all as the frequency approaches zero in this case.

In order to further solve Equation 6.4.11, it should be noted that the linear system is still under-determined. This just reflects the fact that one amplitude can be chosen arbitrary. Only the three remaining amplitudes are fixed by the linear system. For example, the amplitude of the vertical velocity A_v can be chosen. The corresponding column of the linear system can then be written on the right hand side of the linear system, resulting in

$$\begin{pmatrix} -i\omega & 0 & 0 \\ 0 & -i\omega & \frac{ik_x}{\rho_0} \\ \frac{g}{\rho_0} & 0 & \frac{ik_y}{\rho_0} \\ 0 & ik_x & 0 \end{pmatrix} \cdot \begin{pmatrix} A_\vartheta \\ A_u \\ A_p \end{pmatrix} = -A_v \begin{pmatrix} \frac{d\vartheta_{\mathrm{hse}}}{dy} \\ 0 \\ -i\omega \\ ik_y \end{pmatrix}. \tag{6.4.14}$$

With the above dispersion relation, this reduced system has the unique solution given by:

$$A_\vartheta = -\frac{i}{\omega}\frac{d\vartheta_{\mathrm{hse}}}{dy} A_v \tag{6.4.15}$$

$$A_u = -\frac{k_y}{k_x} A_v \tag{6.4.16}$$

$$A_p = -\rho_0 \omega \frac{k_y}{k_x^2} A_v \tag{6.4.17}$$

These solutions are called polarization relations of the internal gravity waves. Together with the dispersion relation, they may be inserted into the plane wave ansatz given by Equation 6.4.10. It can then be seen that both velocity components and the pressure fluctuations are either in phase or inversely phased, while the potential temperature fluctuations are always out of phase by $\pi/2$. It should be noted that if the above ansatz is used to describe a physical internal gravity wave,

the real parts of the ansatz functions has to be taken. Moreover, it can be seen that a plane internal gravity wave is fully described by a given wave vector and the amplitude for one field.

The above description treats internal gravity waves in the linear regime which is reflected by the linear system given in Equation 6.4.11. Thus, any superposition of plane waves is also a solution of this equation. Specifically, this fact permits the construction of monochromatic wave packets, where the amplitude of a plane wave is for example modulated by a Gaussian function. The maximum of such a wave packet travels with the corresponding group velocity which is generally defined by

$$\mathbf{c}_g = \nabla_{\mathbf{k}} \omega. \tag{6.4.18}$$

For the above dispersion relation the Cartesian components of the group velocity can be evaluated to

$$c_{gx} = \frac{N_0}{k_x} \cos\Theta \sin^2\Theta, \tag{6.4.19}$$

$$c_{gy} = -\frac{N_0}{k_x} \cos^2\Theta \sin\Theta. \tag{6.4.20}$$

Interestingly, the group velocity is perpendicular to the wave vector and thus also to the phase velocity. This curious behavior is a result of the anisotropic propagation of internal gravity waves.

6.4.3 Simulation of a wave packet

In this subsection, it is investigated whether internal gravity waves can be resolved in numerical simulations with LHC. To this end, a very simple hydrostatic atmosphere in a two-dimensional Cartesian domain is considered. The fluid consists of an ideal gas at constant temperature $T_0 = 300$ with a constant mean molecular weight $\mu = 1$. The constant gravitational acceleration $g = 1000$ is pointing towards the negative y-axes. Choosing a density $\rho_0 = 1$ at zero height, the hydrostatic density and pressure profiles are then given by

$$\rho_{\text{hse}}(x, y) = \rho_0 \exp(-y/H_p), \qquad p_{\text{hse}}(x, y) = \rho_0 \tilde{R} T_0 \exp(-y/H_p). \tag{6.4.21}$$

The resulting pressure scale height and the Brunt-Väisälä-frequency are also spatially constant and are given by

$$H_p = \frac{R T_0}{g\,\mu}, \qquad N_0 = \sqrt{\frac{g}{H_p} \cdot \frac{\gamma - 1}{\gamma}}. \tag{6.4.22}$$

The corresponding potential temperature profile of the hydrostatic atmosphere can be obtained by Equation 2.7.1 and results in

$$\vartheta_{\text{hse}} = T_0 \exp\left(\frac{y}{H_p} \cdot \frac{\gamma - 1}{\gamma}\right). \tag{6.4.23}$$

In order to be able to compare the simulation results to the Boussinesq theory of internal gravity waves discussed in Section 6.4.2, the vertical extents of the physical domain are chosen very small compared to the pressure scale height:

$$y_1 = -0.01 \cdot H_p, \qquad y_2 = 0.02 \cdot H_p \tag{6.4.24}$$

The values of the hydrostatic density and potential temperature profiles thus stay very close to their corresponding reference values at zero height, i.e. $\rho_{\text{hse}}(x, y) \approx \rho_0$, $\vartheta_{\text{hse}} \approx T_0$.

The hydrostatic atmosphere is distorted by a monochromatic wave packet of an internal gravity wave as described by the linear Boussinesq theory. The wave vector is chosen such that the wave length corresponds to a tenth of the physical domain, while the wave vector is inclined by $-60°$ with respect to the horizontal direction:

$$|\mathbf{k}| = \frac{2\pi}{(y_2 - y_1)/10}, \qquad \Theta = -60°/180°\pi \tag{6.4.25}$$

The corresponding Cartesian components of the wave vector are then given by

$$k_x = |\mathbf{k}| \cos\Theta, \qquad k_y = |\mathbf{k}| \sin\Theta. \tag{6.4.26}$$

With these values, the extents of the physical domain are chosen as $[0, \frac{2\pi}{|k_x|}] \times [y_1, y_2]$, such that one horizontal wave length fits into the domain. With periodic horizontal boundary conditions, the plane wave approximation can then at least be maintained in the horizontal direction. In the vertical direction, the amplitude of the internal gravity wave is modulated by a Gaussian function. Setting the peak Mach number for vertical velocity fluctuations to 10^{-6}, the amplitude function of the vertical velocity is then given by

$$A_v(y) = 10^{-6} \cdot \sqrt{\gamma \tilde{R} T_0} \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{y}{|y_1|/4}\right)^2\right). \tag{6.4.27}$$

With the polarization relations given in Section 6.4.2, the initial conditions for the internal gravity wave setup are given by the following expressions:

$$\begin{aligned}
\vartheta\,(x, y, t=0) &= \vartheta_{\text{hse}} + \Re\left\{-\frac{i}{\omega}\frac{d\vartheta_{\text{hse}}}{dy} A_v e^{i(k_x x + k_y y)}\right\} \\
u\,(x, y, t=0) &= \Re\left\{-\frac{k_y}{k_x} A_v e^{i(k_x x + k_y y)}\right\} \\
v\,(x, y, t=0) &= \Re\left\{A_v e^{i(k_x x + k_y y)}\right\} \\
p\,(x, y, t=0) &= p_{\text{hse}} + \Re\left\{-\rho_0 \omega \frac{k_y}{k_x^2} A_v e^{i(k_x x + k_y y)}\right\}
\end{aligned} \tag{6.4.28}$$

The corresponding initial density distribution can be obtained via Equation 2.7.1 and the ideal gas law.

From the Boussinesq theory, it is expected that the above wave packet propagates with its group velocity in the vertical direction:

$$c_{gy} = +\frac{\sqrt{3}}{4} \cdot \frac{N_0}{|\mathbf{k}|} \tag{6.4.29}$$

Theoretically, the amplitude function (6.4.27) should just be shifted as a function of time without changing its shape or relative size:

$$A_v(y, t) = A_v(y - c_{gy} t) \tag{6.4.30}$$

This property will be an important measure for the quality of different numerical discretizations.

For the numerical tests, the physical domain is discretized by 128×128 grid cells. The vertical boundary conditions are given by constant ghost cell values filled with the hydrostatic equilibrium profiles. For the temporal discretization, the implicit ESDIRK23 stepper is chosen which is second-order accurate. The time step is held fixed and is chosen such that one oscillation period with the Brunt-Väisälä-frequency is resolved by 20 steps, i.e. $\Delta t = \frac{1}{20} \cdot \frac{2\pi}{N_0}$.

Before simulating the actual gravity wave, it is tested whether the hydrostatic equilibrium is stable on the grid or if the hydrostatic numerical instability described in the previous section is also present here. In order to investigate this, the hydrostatic equilibrium profiles without the wave disturbances are advanced in time with different spatial discretization techniques. The Roe-Lowmach and the AUSM$^+$-Lowmach (at $M_{\text{cut}} = 10^{-12}$) numerical flux functions are each tested with the constant and linear interface reconstruction schemes in combination with a pointwise discretization of gravity. Moreover, the well-balanced hydrostatic reconstruction method is tested with both flux functions. As in Section 6.3, the initial horizontal velocity field is initialized with random noise with a peak Mach number of 10^{-15}. The numerical stability of the hydrostatic atmosphere is again be determined by the temporal evolution of the horizontal density fluctuations,

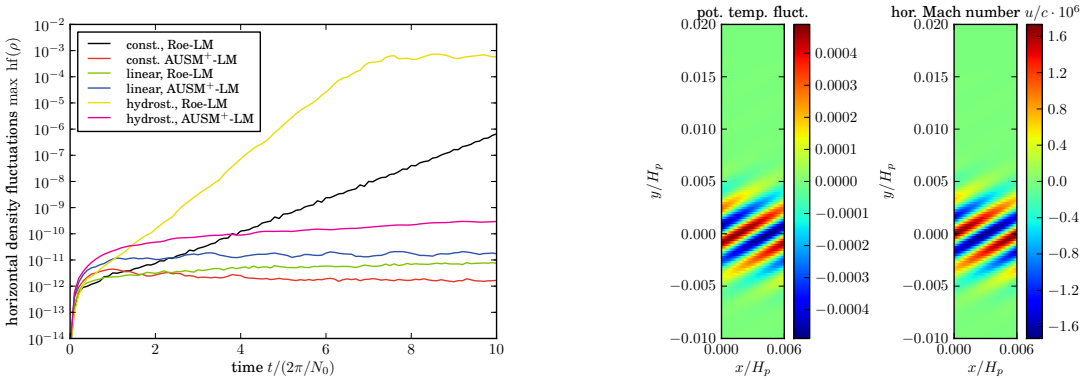


Figure 6.17: temporal evolution of horizontal density fluctuations on a quiet hydrostatic atmosphere (left); initial conditions for the internal gravity wave packet (right)

which is shown in the left panel of Figure 6.17 for each discretization method. It can be seen that the AUSM$^+$-Lowmach flux is stable for constant and linear reconstruction. Interestingly, the hydrostatic reconstruction shows a very small growth of the density fluctuations here. In contrast, the Roe-Lowmach flux exhibits a rather strong numerical instability for the constant and the hydrostatic reconstruction. Only the linear reconstruction scheme is stable here.

Despite the problems with the hydrostatic numerical instability which is also present in the atmosphere under consideration, the performance of different spatial discretizations in resolving the internal gravity wave packet is evaluated next. To this end, the grid is set up with the initial conditions described by Equations 6.4.28. The corresponding potential temperature fluctuations and the horizontal velocity field is exemplary shown in the right panel of Figure 6.17. The setup is evolved in time with the same spatial and temporal discretization techniques that were used in the hydrostatic stability test before. In order to analyze the results, it is desirable to extract the time-dependent amplitude function of the wave packet from the simulation data which can then be compared to the theoretical result given by Equation 6.4.30. In order to develop such an extraction technique, it is assumed that the horizontal velocity field at a certain time can be decomposed into a product of a vertical profile $u(y)$ and a periodic sinusoidal horizontal fluctuation:

$$u(x, y) = u(y) \sin\left(k_x x + \varphi(y)\right) \tag{6.4.31}$$

As the phase φ may also depend on y, this expression is valid as long as the horizontal fluctuation is close to a sine. In particular, the analytic solution for the velocity field given by the Boussinesq theory fulfills this criterion exactly. Integrating the square of Equation 6.4.31 over the horizontal box size then eliminates the sinusoidal x-dependence:

$$\int_0^{2\pi/k_x} u(x, y)^2 dx = \frac{\pi}{k_x} u(y)^2 \tag{6.4.32}$$

With this expression, the vertical amplitude profile $u(y)$ can be recovered. For the analysis of the simulation data, the above integral is evaluated numerically and the resulting amplitude profile is normalized to its maximum initial value:

$$r_j = \frac{\sqrt{\frac{k_x}{\pi} \sum_i u_{i,j}^2 \Delta x}}{\left| \frac{k_y}{k_x} A_v(y = 0) \right|} \tag{6.4.33}$$

Thus, r_j can be used to measure the relative amplitude of the internal gravity wave from the simulation data even after the initial conditions were evolved in time and are not necessarily given by the analytic Boussinesq solution.

The amplitude reconstruction procedure is performed for each simulation setup at three points in time, after $2, 4$ and 6 oscillation periods of the Brunt-Väisälä-frequency. The results for the different spatial discretization techniques are shown in Figure 6.18. The theoretical positions of the maximum of the amplitude function, given by $c_{gy} \cdot t$, is marked by dashed lines within each plot. Moreover, the spatial distribution of the velocity field at the end of each simulation ($t = 6 \cdot 2\pi/N_0$) is shown separately for each discretization. Examining the plots for the Roe-Lowmach flux with constant reconstruction, it can be seen that the amplitude function initially propagates at the estimated group velocity, although the amplitude is heavily damped out. However, already at $t = 4 \cdot 2\pi/N_0$, it can be seen that a second maximum forms in the lower part of the atmosphere, which finally dominates the velocity field at the end of the simulation. The spatial distribution of the velocity field clearly shows that internal gravity wave is scrambled, which is not compliant with the theoretical considerations. In contrast to that, the AUSM$^+$-Lowmach flux with constant reconstruction shows a very different behavior. The internal gravity wave propagates continuously and does not show any signs for a second maximum in the amplitude function. However, besides that the wave packet also gets heavily damped out here, it does not propagate with the estimated speed at all.

The Roe-Lowmach and the AUSM$^+$-Lowmach flux exhibit a very similar behavior when they are used in combination with the linear interface reconstruction method. The internal gravity wave propagates exactly with the theoretically predicted speed, while the amplitude decreases only mildly compared to the previous tests. As the linear reconstruction method provides second-order accuracy in space, the corresponding results seem to be more reliable than the first-order results.

With the well-balanced hydrostatic reconstruction scheme combined with the Roe-Lowmach flux, the amplitude function of the internal gravity waves quickly *increases* (notice the different scale in the amplitude function plot here). The simulation even aborts with unphysical values shortly after $t = 4 \cdot 2\pi/N_0$. As an increase in the amplitude is not compliant with energy conservation principles (cf. Sutherland 2010) for internal gravity waves, this behavior is clearly unphysical and must be attributed to the numerical discretization. Opposed to that, the AUSM$^+$-Lowmach flux with hydrostatic reconstruction shows an excellent behavior at the beginning of the simulation. The internal gravity wave propagates exactly with the predicted speed and the damping is very low. It must be emphasized that the hydrostatic reconstruction is only first-order accurate in space, but exhibits a damping behavior similar to the second-order scheme without well-balancing. Thus, it can be suspected that a second-order extension of the hydrostatic reconstruction would provide even better results. Unfortunately, a second maximum also builds up here in the amplitude function at the end of the simulation. As this spurious maximum is constantly increasing, the results would similar to the Roe-Lowmach scheme with hydrostatic reconstruction if the simulation would be run on longer time scales.

In total, only the linear reconstruction scheme with a pointwise discretization of gravity shows reasonable results for the low Mach number flux functions. All other discretizations show some kind of spurious behavior, which does not seem to be physical. Comparing these results with the growth rates of the hydrostatic numerical instability presented in Figure 6.17 shows a very interesting correlation. The faster the hydrostatic numerical instability grows for a given discretization method, the faster the spurious amplitudes rise for the internal gravity waves. This is a strong hint, that both numerical problems are actually caused by internal gravity waves, which are not properly resolved by the numerical discretizations. This is underlined by the fact that the hydrostatic numerical instability typically grows on time scales given by the Brunt-Väisälä-frequency. This can be seen from the non-dimensional numerical tests presented in Section 6.3, where the reference time was always chosen according to N_0. However, the internal gravity wave tests presented here also show that such a wave may spuriously rise in amplitude without exhibiting a checkerboard-like pattern as it was obtained for the hydrostatic numerical instability. Unfortunately, this also means that the use of LHC for stellar atmospheres is very limited at the moment. It is not possible to perform more complex simulations with the existing discretization methods and just abort them when a spurious checkerboard pattern shows up. The internal grav-

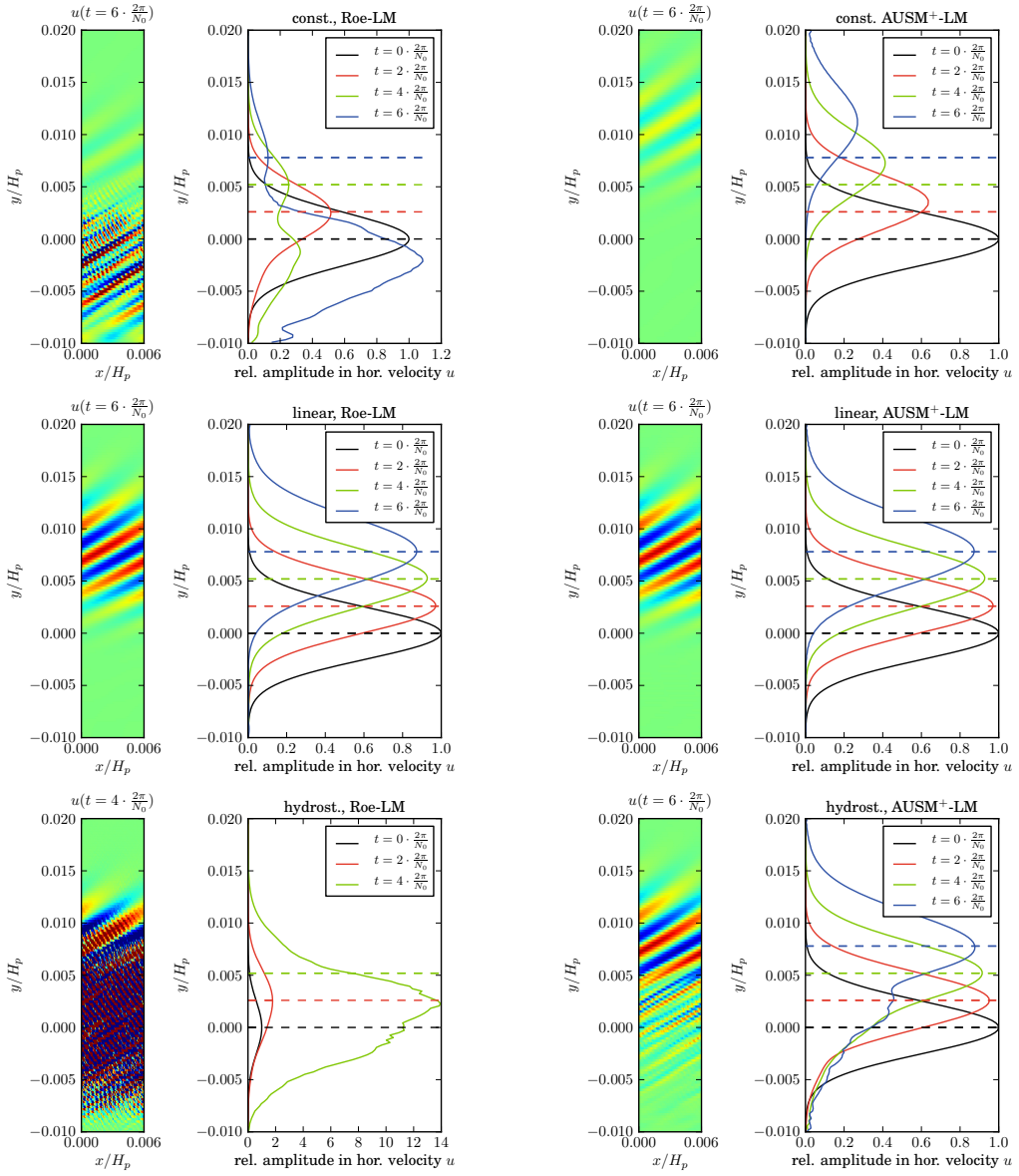


Figure 6.18: horizontal velocity fields (pseudo-color plots) at the end of each simulation and extracted amplitude functions of the internal wave packet at different points in time (line plots); dashed lines denote the theoretically expected position of the wave packet

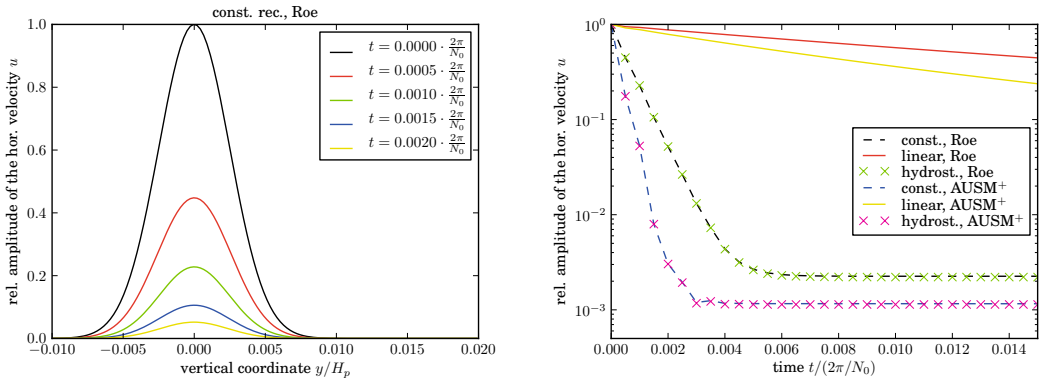


Figure 6.19: Amplitudes of the internal gravity wave packet for non-low Mach number flux functions

ity wave dynamics may also show spurious artifacts on more resolved scales, which can probably not be distinguished from physical flow features in more complex situations. Thus, it is of major importance for the future development on LHC to design new discretization techniques that are not prone to the numerical instabilities presented here. However, the internal gravity wave test also showed up the prospects of well-balanced schemes as they capture internal gravity waves with much greater accuracy.

Finally, it should be noted that the internal gravity wave tests were only performed with low Mach number flux functions, which generally seem to strengthen the numerical instabilities presented in this chapter. Hence, the above tests are repeated with the ordinary versions of the corresponding numerical flux functions (setting $M_{\text{cut}} = 1$). The only difference is that the size of the time step is reduced by a factor of 100 to $\Delta t = \frac{1}{2000} \cdot \frac{2\pi}{N_0}$. The temporal evolution of the amplitude function of the wave packet is shown in the left panel of Figure 6.19 for Roe's approximate Riemann solver with the constant interface reconstruction method. It can be seen that the amplitude quickly decreases to very low values on extremely short time scales. The temporal evolution of the maximum of the relative amplitude is shown in the right panel of Figure 6.19 for all discretizations. The amplitude decreases exponentially for all discretization methods until a certain saturation level is reached. Although the decay of the internal gravity wave is slower with linear interface reconstruction, the damping is still way too high for any practical application. Moreover, the results show that the use of the well-balanced hydrostatic reconstruction does not provide any advantage here. The damping can easily be explained by the excessive numerical viscosity of the flux functions in the low Mach number regime (cf. Chapter 4). Thus, a low Mach number flux function combined with a proper discretization of gravity seems to vital in order to resolve internal gravity wave accurately.

6.4.4 Non-linear behavior and astrophysical relevance

The plain wave solutions for internal gravity waves discussed in Section 6.4.2 are only valid within the Boussinesq approximation and for sufficiently small wave amplitudes such that non-linear effects are insignificant. For internal gravity waves traveling over large parts of the pressure scale height of the corresponding atmosphere, an analytic linear theory can also be derived for the anelastic equations (cf. Section 2.7.2). The plain wave solutions then contain an additional amplitude variation with the vertical coordinate depending on the density scale height (see Sutherland 2010). However, the general behavior of these linear internal gravity waves is very similar to the results derived within the Boussinesq approximation.

For internal gravity waves whose amplitude is large, non-linear effects have to be considered.

Remarkably, it can easily be shown that a single plain wave, as described by Equation 6.4.10 and the corresponding dispersion and polarization relations, is in fact a solution of the *non-linear* Boussinesq equations. However, as soon as some kind of wave packet is considered, this statement does not hold anymore. It can be shown that the amplitude function of a Gaussian wave packet may either broaden or even steepen as it advanced in time (see Sutherland 2010). In the latter case, which occurs for $\cos\Theta > \sqrt{2/3}$, the wave packet successively narrows and increases in amplitude. This effect is known as *modulational instability* but is only expected to be visible for rather large initial amplitudes of the wave. It should be noted that the results presented in the previous section are not affected by this phenomenon because the amplitude is very small and the above condition on the angle of the wave is not fulfilled here anyway.

The weakly non-linear theory of internal gravity waves also predicts that different plain waves may interact by resonant triad interactions (see e.g. Sutherland 2010). The wave vectors and frequencies of the involved waves must satisfy the following resonance conditions,

$$\mathbf{k}_1 = \mathbf{k}_2 \pm \mathbf{k}_3, \qquad \omega_1 = \omega_2 \pm \omega_3, \tag{6.4.34}$$

which simply express the conservation of momentum and energy during the wave interaction. Under suitable conditions, it is therefore possible to transfer energy from one wave to a different wave of different wave length. An important special case of these triad interactions concerns the stability of a single wave. It can be shown that a plane internal gravity wave is inherently unstable and may decay into two other waves of different wave numbers. A theoretical Floquet analysis of this problem reveals that the maximum growth rates for this instability are obtained for $\omega_2 = \omega_3 = -\omega_1/2$ and $k_2 \approx k_3 \gg k_1$ (Klostermeyer 1982, 1991; Staquet & Sommeria 2002). Due to its frequency dependence, the process is called *parametric subharmonic instability* (PSI). It provides a mechanism of direct energy transfer from large to small scales. Interestingly, the instability is not confined to rather strong waves, but may also occur for waves with arbitrary small amplitudes (Drazin 1977). The PSI has been subject to various theoretical and numerical investigations in literature (see Bouruet-Aubertot et al. 1995; Staquet & Sommeria 2002, and references in there) and has even been confirmed experimentally (e.g. Benielli & Sommeria 1998).

In the author's opinion, it seems to be possible that the PSI is also responsible for the numerical stability problems described in the last sections. The PSI may generate waves at successively smaller length scales, which cannot be resolved on the discrete grid at some level. Due to a failure of the numerical discretization, the unresolved waves do not get dissipated but get shifted in wave number by numerical aliasing effects. If this process is self-sustained by some kind of resonant interaction, this could be an explanation for the numerical stability problems. However, it should be stressed that these arguments cannot be further reasoned or even proved at the current stage of the investigation. In any case, a better numerical treatment of hydrostatic equilibria and internal gravity waves seems to be vital in order to resolve the numerical problems.

The fate of non-linear internal gravity waves is generally governed by two effects. First, if the amplitude of a wave reaches a certain critical value, the wave may break by various processes (see Staquet & Sommeria 2002; Sutherland 2010, for an overview). For example, the amplitude of internal waves may increase drastically due to the modulational instability or by effects of non-uniform stratification. The density or velocity fluctuations of the wave may become so strong that the wave breaks into a turbulent flow field by local Rayleigh-Taylor or Kelvin-Helmholtz instabilities. The turbulence then decays to smaller scales until it is dissipated by viscous forces. Second, if the amplitudes are not strong enough for wave breaking, the long term behavior of internal gravity waves is governed by the PSI. As the subharmonic wave components may again be subject to resonant triad interactions, the wave energy may be transferred to successively smaller scales. The number of involved wave components may tremendously increase during this evolution. Finally, the wave lengths may become so small, that these waves are dissipated by viscous processes. As a detailed theoretical description of successive triad interaction is very complex due to the large number of involved waves, the overall behavior is usually described by statistical models (see Staquet & Sommeria 2002; Sutherland 2010, and references therein). This approach

is very similar to turbulence modeling although the properties of internal wave turbulence may be quite different. The corresponding wave spectra can actually be measured for atmospheric and oceanic flows on earth, but are still subject to ongoing research in the corresponding scientific disciplines.

In stellar interiors, internal gravity waves are expected to be an important transport mechanism for chemical elements and angular momentum in radiative zones (see e.g. Schatzman 1993). The internal gravity waves are expected to be generated at the boundaries between convective and radiative regions by convective overshooting. The spectra of the excited waves can actually predicted by analytic models (e.g. Garcia Lopez & Spruit 1991) or numerical simulations (e.g. Dintrans et al. 2005; Rogers & Glatzmaier 2005). Although there is a rough agreement on the global properties of the spectra, their details remain controversial. The propagation of gravity waves within radiative zones has attracted much attention for the sun because they may explain two important effects (see Garcia Lopez & Spruit 1991; Charbonnel & Talon 2005; Rogers & Glatzmaier 2005). First, helioseismic measurements revealed that the inner radiative zone in the sun is in nearly in solid-body rotation which is in contradiction to simple solar models which include rotation. Internal gravity waves seem to provide a suitable mechanism for angular momentum transport in order to explain this effect. Second, the solar surfaces abundances show a significant depletion of lithium. This may only be explained by a mechanism which mixes the lithium sufficiently deep into the stellar interior where the temperatures are high enough for a destruction of this element by election captures. However, the outer convective zone of the sun does not extent to such depths. Therefore, an additional mechanism for mixing into the inner radiative zone is necessary. Mixing by internal gravity waves seem to be a very good candidate in order to explain this process.

In general, it can be seen that internal gravity waves may have an important impact on transport and mixing processes in stellar interiors and may be vital in order to explain stellar rotation profiles in a self-consistent way. Numerical simulations have mostly been performed for solar-like stars and are often either not very suitable for low Mach number flows or use the anelastic approximation, which restricts simulations to stellar regions near convective zones. Therefore, it can be suspected that low Mach number codes as presented in this work may provide valuable new insights in the modeling of mixing and transport processes by internal gravity waves.

7 Shear instabilities

7.1 Introduction

In this chapter, the prospects of hydrodynamic modeling of stellar interiors are shown by the example of shear flows, whose stability may have a significant impact on the evolution of certain stars. The investigation is focused on shear flows with vertical gradients in the horizontal velocity components. This is the typical situation in radiative zones in differentially rotating stars. If such a shear flow is unstable, small disturbances grow in time until the flow field becomes fully turbulent. This process is a possible source for mixing and angular momentum transport in stellar radiative region and can only be treated by effective turbulent diffusivities in stellar evolution models (see e.g. Hirschi et al. 2004; Mathis et al. 2004; Maeder 2009). Here, the stellar conditions under which shear flows may become unstable are investigated by a theoretical model and detailed multi-dimensional numerical simulations. The results may give some insight when an additional turbulent diffusion should actually be used in stellar evolution models. In principle, the effective diffusivities could also be derived from such numerical simulations and compared to existing prescriptions. However, this is subject to future work.

In the absence of gravity, inviscid shear flows are always unstable to arbitrary small fluctuations leading to the well-known Kelvin-Helmholtz instability. In a stratified atmosphere, the conditions under which a shear flow may become unstable can be illustrated by very simple energetic arguments (e.g. Chandrasekhar 1961). Consider two neighboring fluid elements in a stratified atmosphere with different horizontal velocities as shown in Figure 7.1. In order to exchange the fluid elements, some potential energy has to be provided. Raising the lower (and denser) fluid element upwards needs more energy than the lowering of the upper (and lighter) fluid element can provide. The difference in potential energy thus reads

$$\Delta E_{\mathrm{pot}} = -g\,\delta\rho\,\delta y, \tag{7.1.1}$$

where $\delta\rho$ denotes the density difference between the two fluid elements. The only energy source available to the system is the kinetic energy of the horizontal motion. The kinetic energies of the fluid elements before the exchange are

$$E_{\mathrm{kin},1} = \frac{1}{2}\rho u^2 \qquad E_{\mathrm{kin},2} = \frac{1}{2}\rho\left(u+\delta u\right)^2. \tag{7.1.2}$$

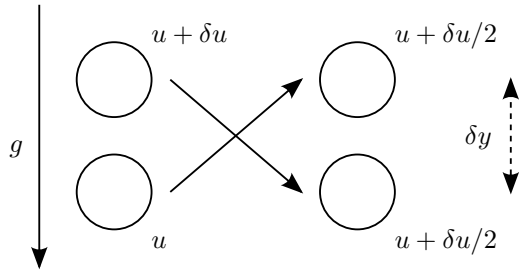


Figure 7.1: Exchange of two fluid elements in a stratified medium

Assuming that the fluid elements mix during the exchange, they both end up at the same average velocity and kinetic energy

$$\bar{u} = u + \frac{\delta u}{2} \qquad \bar{E}_{\text{kin}} = \frac{1}{2}\rho\bar{u}^2. \tag{7.1.3}$$

The difference in kinetic energy thus reads

$$\Delta E_{\text{kin}} = \frac{1}{4}\,\rho\,\delta u^2. \tag{7.1.4}$$

This energy is available to overcome the potential energy and trigger vertical motions which lead to the shear instability. Therefore, it is obvious that if the sum of the energy differences is negative, i.e.

$$\Delta E_{\text{kin}} + \Delta E_{\text{pot}} < 0, \tag{7.1.5}$$

the flow must be stable. Rewriting this condition leads to the so-called *Richardson criterion*:

$$Ri > 1/4 \quad \text{where} \quad Ri = \frac{g}{\rho}\frac{\delta\rho}{\delta y}\frac{1}{\left(\frac{\delta u}{\delta y}\right)^2} \tag{7.1.6}$$

It states that the flow is stable if the Richardson number is greater than the so-called *critical Richardson number* (1/4). In a continuously stratified medium the Richardson number can be defined by

$$Ri\,(y) = \frac{g\ \rho'\,(y)}{\rho\,(y)\,(u'\,(y))^2}. \tag{7.1.7}$$

It should be emphasized that if the Richardson criterion is violated somewhere within an atmosphere, the flow may become unstable but it does not have to.

A mathematically rigorous derivation of this result was given by Howard (1961). He performed a linear stability analysis on the Boussinesq equations (see Section 2.7.1). He found that a necessary condition for instability is that the local Richardson number, as defined in Equation 7.1.7, has to be below 1/4 somewhere in the flow. This result holds for arbitrary density and velocity profiles.

The above stability criterion is only valid in the context of the Boussinesq approximation, where density and temperature fluctuations have to be small. However, inspecting the above definition of the Richardson number it can be seen that it contains the Brunt-Väisälä frequency for Boussinesq flow

$$N^2\,(y) = g\frac{\rho'(y)}{\rho(y)}. \tag{7.1.8}$$

This suggests to extend the definition of the Richardson number also for non-Boussinesq flows,

$$Ri\,(y) = \frac{N^2\,(y)}{(u'(y))^2}, \tag{7.1.9}$$

where the Brunt-Väisälä-frequency is defined in a different way (e.g. as in Equation 6.4.4 for an ideal gas). This extension is similar to the approach followed in Section 6.4 for internal gravity waves. It should be strongly emphasized that Equation (7.1.7) and (7.1.9) are *not* equivalent for non-Boussinesq flows. Concerning the stability of non-Boussinesq shear flows, the value of the critical Richardson number has sometimes be questioned in literature. For example, Brüggen & Hillebrandt (2001) reported shear instabilities in numerical simulations for Richardson numbers up to 1.5. However, the increased critical Richardson number can be solely attributed to the use of the Boussinesq definition of the Richardson number, which is not appropriate in this case. Thus, stability results should always be carefully compared regarding the actual definition of the Richardson number, which is unfortunately not unique. Moreover, Canuto (2002) derived a critical Richardson number from turbulence theory. The result basically states that turbulence by shear instabilities can be maintained up to Richardson numbers of unity. This finding is not necessarily

in conflict with a critical Richardson number of 1/4 which assumes that the flow field is initially not turbulent. However, Canuto's result may raise the question if a critical Richardson number of 1/4 is generally valid for non-Boussinesq flows. Instabilities could, for example, be excited by non-linear effects which are usually not captured by a linear stability analysis. Moreover, thermal effects or small but finite Mach numbers may alter the stability criterion. In order to investigate the general behavior of shear flows, an extended stability analysis is performed in the following. Moreover, the results are compared to hydrodynamic simulations with LHC as they are not affected by the potential limitations of the analytic treatment.

It should be noted that the stability criterion for stratified shear flows may further be altered by physical processes which are not discussed here. For example, thermal radiation may significantly reduce the effective Richardson number (or alternatively increase the critical Richardson number) in the case of small Péclet numbers (Lignières et al. 1999). Moreover, the interaction of internal gravity waves with a mean shear flow may give rise to additional interesting phenomena (see e.g. Staquet & Sommeria 2002; Sutherland 2010). These physical processes may be treated adequately with hydrodynamic simulations in LHC as soon as the remaining numerical challenges are resolved.

7.2 Model problem

7.2.1 Setup and governing equations

In this work, the stability criteria for shear flows are examined in a fully compressible approach. In order to simplify the analysis, a model atmosphere with well-defined control parameters is constructed. The response of this equilibrium state to small disturbances is then investigated in the following sections. If fluctuations grow in time, the equilibrium state is said to be unstable otherwise it is stable.

The shear flow is examined on a two-dimensional Cartesian plane with infinite extents. The gravitational acceleration $g > 0$ is assumed to be constant over the whole domain and is pointing towards the negative y-axis.

The velocity profile is modeled by a hyperbolic tangent of a certain width L_u. The Cartesian velocity components of the equilibrium state are

$$u_{\mathrm{eq}}(x,y) = u_o \ \tanh\left(\frac{y}{L_u}\right), \qquad v_{\mathrm{eq}}(x,y) = 0. \tag{7.2.1}$$

The fluid consists of an ideal gas with constant mean molecular weight. The pressure p is thus related to the density ρ and temperature T by $p = \rho\tilde{R}T$. The ratio of specific heats is assumed to be $\gamma = 5/3$ if not stated differently.

The vertical temperature profile is also modeled by a hyperbolic tangent, but may have a width L_T different to the width of the velocity profile:

$$T(y) = T_0 - \theta \ \tanh\left(\frac{y}{L_T}\right) \tag{7.2.2}$$

The fluid is supposed to be in hydrostatic equilibrium, which is described by the differential equation

$$\frac{\partial p}{\partial y} = -\rho g. \tag{7.2.3}$$

For an ideal gas and the above temperature profile, this equation can be integrated analytically, resulting in the equilibrium pressure profile

$$p_{\mathrm{eq}}(y) = p_0 \exp\left\{-\frac{g}{R}\frac{T_0 y + \theta L_T \ln\left(\cosh\left(\frac{y}{L_T}\right) - \frac{\theta}{T_0}\sinh\left(\frac{y}{L_T}\right)\right)}{T_0^2 - \theta^2}\right\}. \tag{7.2.4}$$

The equilibrium density structure is then determined by the equation of state. For this atmosphere, the Brunt-Väisälä frequency is given by

$$N^2 = \frac{g}{H_p}\left(\nabla_{\text{int}} - \nabla_{\text{ext}}\right) = \frac{g}{H_p}\left(\frac{\gamma - 1}{\gamma} + H_p \frac{T'(y)}{T(y)}\right) \tag{7.2.5}$$

with the pressure scale height

$$H_p = -\frac{p_{\text{eq}}(y)}{p'_{\text{eq}}(y)} = \frac{\tilde{R}T(y)}{g}. \tag{7.2.6}$$

The Richardson number then can be calculated by

$$Ri(y) = \frac{N^2(y)}{\left(u'_{\text{eq}}(y)\right)^2}, \tag{7.2.7}$$

resulting in a lengthy expression not shown here. The stability of the equilibrium state will strongly depend on the minimal Richardson number. For most parameter ranges $Ri(y)$ has a unique minimum at $y = 0$ where it approaches

$$Ri_0 = \frac{g^2 L_u^2}{\tilde{R}T_0 u_0^2}\left(\frac{\gamma - 1}{\gamma} - \frac{\tilde{R}\theta}{gL_T}\right). \tag{7.2.8}$$

This Richardson number is used as a control parameter for the following analysis.

The dynamics of the system are governed by the compressible Euler equations. For an ideal gas, they can be written in the non-conservative form

$$\begin{array}{rcrl} \frac{D\rho}{Dt} & + & \rho\nabla\cdot\vec{u} & = 0 \\ \rho\frac{Du}{Dt} & + & \frac{\partial p}{\partial x} & = 0 \\ \rho\frac{Dv}{Dt} & + & \frac{\partial p}{\partial y} & = -\rho g \\ \frac{Dp}{Dt} & + & p\gamma\,\nabla\cdot\vec{u} & = 0 \end{array} \tag{7.2.9}$$

with the Lagrangian derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}. \tag{7.2.10}$$

The following analysis is based on these equations and is therefore not limited to the Boussinesq or anelastic approximations, as it is commonly done in literature. The influence of divergent velocity fields and entropy gradients on the stability of shear flows are thus potentially captured by this analysis.

7.2.2 Non-dimensionalization

The setup of the equilibrium state depends on many parameters. In order to find the relevant ones, it is very insightful to non-dimensionalize the system. Therefore, each variable is decomposed into a product of its reference quantity (with subscript r) and its dimensionless value (marked with a hat), e.g. $\rho = \rho_{\text{r}} \cdot \hat{\rho}$. The reference quantities should be chosen such that typical dimensionless values are of order unity. Here, the reference density, length scale, temperature, gravity and velocity are set to the following values:

$$\rho_{\text{r}} = \frac{p_0}{RT_0} \quad x_{\text{r}} = L_u \quad T_{\text{r}} = T_0 \quad g_{\text{r}} = g \quad u_{\text{r}} = u_0 \tag{7.2.11}$$

Other reference quantities like the reference time, sound speed, pressure, Mach number and Froude number are derived from them:

$$t_{\text{r}} = \frac{x_{\text{r}}}{u_{\text{r}}} \quad c_{\text{r}} = \sqrt{\gamma R T_{\text{r}}} \quad p_{\text{r}} = \rho_{\text{r}} c_{\text{r}}^2 \quad M_{\text{r}} = \frac{u_{\text{r}}}{c_{\text{r}}} \quad F_{\text{r}} = \frac{u_{\text{r}}}{\sqrt{g_{\text{r}} x_{\text{r}}}} \tag{7.2.12}$$

Moreover, it is useful to define another dimensionless quantity β, which characterizes the ratio of the length scales of the velocity and temperature gradients:

$$\beta = \frac{L_u}{L_T} \tag{7.2.13}$$

With these definitions, the setup of the equilibrium state can be casted to the simple non-dimensional form

$$\hat{p}_{\mathrm{eq}}\left(\hat{y}\right) = \frac{1}{\gamma}\exp\left\{-\frac{\gamma M_{\mathrm{r}}^2}{F_{\mathrm{r}}^2}\frac{\hat{y}+\beta\hat{\theta}\ln\left(\cosh\left(\frac{\hat{y}}{\beta}\right)-\hat{\theta}\sinh\left(\frac{\hat{y}}{\beta}\right)\right)}{1-\hat{\theta}^2}\right\}, \tag{7.2.14}$$

$$\hat{\rho}_{\mathrm{eq}}\left(\hat{y}\right) = \frac{\gamma\hat{p}_{\mathrm{eq}}\left(\hat{y}\right)}{1-\hat{\theta}\tanh\left(\hat{y}/\beta\right)}, \tag{7.2.15}$$

$$\hat{u}_{\mathrm{eq}}\left(\hat{y}\right) = \tanh\left(\hat{y}\right) \quad \hat{v}_{\mathrm{eq}}\left(\hat{y}\right) = 0. \tag{7.2.16}$$

The non-dimensional Euler equations read

$$\begin{array}{rcll} \frac{D\hat{\rho}}{D\hat{t}} & + & \hat{\rho}\,\hat{\nabla}\cdot\vec{\hat{u}} & = 0 \\ \hat{\rho}\frac{D\hat{u}}{D\hat{t}} & + & \frac{1}{M_{\mathrm{r}}^2}\frac{\partial\hat{p}}{\partial\hat{x}} & = 0 \\ \hat{\rho}\frac{D\hat{v}}{D\hat{t}} & + & \frac{1}{M_{\mathrm{r}}^2}\frac{\partial\hat{p}}{\partial\hat{y}} & = -\frac{1}{F_{\mathrm{r}}^2}\hat{\rho} \\ \frac{D\hat{p}}{D\hat{t}} & + & \hat{p}\gamma\,\hat{\nabla}\cdot\vec{\hat{u}} & = 0 \end{array}. \tag{7.2.17}$$

Note that the setup only depends on five parameters, namely $\beta, \gamma, F_{\mathrm{r}}, M_{\mathrm{r}}, \hat{\theta}$. It is thus clear that the temporal evolution of the system and therefore the stability properties of the shear flow may only depend on these five dimensionless numbers. Any additional parameters are just a scaling to specific units, but leave the results unaltered otherwise.

The non-dimensional parameters of the setup each have a distinct physical relevance. However, the meaning of the Froude number F_{r} and the non-dimensional depth of the temperature gradient $\hat{\theta}$ are not very instructive in this context. Therefore, these parameters are transformed to new quantities which emphasize their physical relevance.

At first, the Froude number which is a measure for the strength of gravity is replaced by the Richardson number at zero height as defined in equation (7.2.8). Next, the external temperature gradient

$$\nabla_{\mathrm{ext}}\left(y\right) = -\frac{\tilde{R}T'\left(y\right)}{g} = \frac{\theta\tilde{R}}{gL_T\cosh\left(y/L_T\right)^2} \tag{7.2.18}$$

is considered, which has a maximum at $y = 0$. This value is related to the adiabatic temperature gradient by a new dimensionless factor

$$\alpha = \frac{\nabla_{\mathrm{ext}}\left(0\right)}{\nabla_{\mathrm{int}}} \qquad \alpha < 1. \tag{7.2.19}$$

Here $\alpha = 0$ corresponds to a constant temperature on the whole domain and $\alpha = 1$ denotes the case where the equilibrium is about to become unstable to convection at $y = 0$. For this work α is limited to values below 1 corresponding to radiative regions in stars.

The new parameters α and Ri_0 can now be used to replace $\hat{\theta}$ and F_{r}. After non-dimensionalizing all involved quantities and some algebraic transformations the corresponding expressions read

$$\hat{\theta} = M_r\alpha\beta\sqrt{\frac{Ri_0\left(\gamma-1\right)}{1-\alpha}} \tag{7.2.20}$$

$$F_{\mathrm{r}}^2 = M_{\mathrm{r}} \sqrt{\frac{(1-\alpha)(\gamma-1)}{Ri_0}}. \tag{7.2.21}$$

The $\hat{y}$-dependence of the Richardson number can also be expressed by the new parameters:

$$Ri(\hat{y}) = Ri_0 \, \frac{\cosh^4(\hat{y}) \left(\alpha \, \mathrm{sech}^2(\hat{y}/\beta) - 1\right)}{(\alpha - 1)\left(1 - M_{\mathrm{r}} \alpha\beta \sqrt{Ri_0 \frac{\gamma-1}{1-\alpha}} \tanh(\hat{y}/\beta)\right)} \tag{7.2.22}$$

Since it is not possible to derive the minimum of this function in closed form, it is always computed numerically for a given set of parameters. If not stated differently, the minimum is Ri_0 at $\hat{y} = 0$.

7.3 Semi-analytical stability analysis

7.3.1 Method

The first approach to study the stability of above defined equilibrium state is to analyze the governing equations analytically. The following calculations are based on the work by Blumen (1970), who studied the influence of finite Mach numbers on the stability of a hyperbolic tangent shear flow. His work is extended here to the model defined in Section 7.2 in order to examine the effects of temperature gradients, gravity and stratification.

In order to simplify the notation, all quantities in this section are assumed to be non-dimensional. The hat is thus omitted in the following. The stability of the shear flow can be investigated by considering small disturbances to the equilibrium state. To this end, a normal-mode approach for the solution of the non-dimensional Euler equations is considered. The solution is split into a sum of the time-independent equilibrium state a small periodic disturbance:

$$\begin{pmatrix} \rho(x,y,t) \\ u(x,y,t) \\ v(x,y,t) \\ p(x,y,t) \end{pmatrix} = \begin{pmatrix} \rho_{\mathrm{eq}}(y) \\ u_{\mathrm{eq}}(y) \\ 0 \\ p_{\mathrm{eq}}(y) \end{pmatrix} + \begin{pmatrix} \tilde{\rho}(y) \\ \tilde{u}(y) \\ \tilde{v}(y) \\ \tilde{p}(y) \end{pmatrix} e^{ikx - i\omega t} \tag{7.3.1}$$

The disturbance is composed of a complex coefficient marked by a tilde that only has a y-dependence and an oscillating part containing the x and t dependence. The spatial wave number of the disturbance is denoted by k. The complex-valued temporal frequency is denoted by $\omega = \omega_r + i\omega_i$. The imaginary part ω_i is called *growth rate*. If it is positive for a given mode, the disturbance will grow exponentially in time. The equilibrium state is said to be unstable in this case. For negative values of ω_i the disturbance decays and the equilibrium state is stable. The goal of the analysis is to compute the dispersion relation $\omega(k)$ for any given set of parameters.

For this purpose, the ansatz (7.3.1) is inserted into the non-dimensional Euler equations (7.2.17). Non-linear terms with products of $e^{ikx-i\omega t}$ which would excite higher harmonics are omitted. The analysis is therefore only valid in the linear regime, where the solution stays on the given spatial wave number k. In this case, any unstable mode grows exponentially in time without any limits. It is therefore clear that for real flows, a non-linear coupling between the modes has to stop the growth at some time. However, the linear approach used here is sufficient to determine the stability of the flow.

An additional requirement for the analysis is that the equilibrium state fulfills the non-dimensional hydrostatic balance equation,

$$\frac{\partial p_{\mathrm{eq}}(y)}{\partial y} = -\frac{M_{\mathrm{r}}^2}{F_{\mathrm{r}}^2} \, \rho_{\mathrm{eq}}(y), \tag{7.3.2}$$

which is the case for the model defined in the previous section. With these assumptions, the Euler equations can be transformed to a set of coupled linear differential equations for the coefficients

of the disturbance:

$$\begin{array}{rcl} i\,(ku_{\mathrm{eq}}-\omega)\;\tilde{\rho} & + & \tilde{v}\;\rho'_{\mathrm{eq}}+\rho_{\mathrm{eq}}\,(ik\tilde{u}+\tilde{v}')=0 \\ i\,(ku_{\mathrm{eq}}-\omega)\;\tilde{p} & + & \tilde{v}\;p'_{\mathrm{eq}}+p_{\mathrm{eq}}\gamma\,(ik\tilde{u}+\tilde{v}')=0 \\ i\rho_{\mathrm{eq}}\,(ku_{\mathrm{eq}}-\omega)\;\tilde{u} & + & \frac{1}{M_{\mathrm{r}}^2}\;ik\tilde{p}+\rho_{\mathrm{eq}}\tilde{v}u'_{\mathrm{eq}}=0 \\ i\rho_{\mathrm{eq}}\,(ku_{\mathrm{eq}}-\omega)\;\tilde{v} & + & \frac{1}{M_{\mathrm{r}}^2}\;\tilde{p}'+\frac{1}{F_{\mathrm{r}}^2}\tilde{\rho}=0 \end{array} \tag{7.3.3}$$

Variables with a dash denote the derivative with respect to y. With some elementary algebraic transformations, the system of equations can be converted to a single differential equation for the disturbance in the vertical velocity component:

$$\tilde{v}''(y)+A(y)\;\tilde{v}'(y)+B(y)\;\tilde{v}(y)=0 \tag{7.3.4}$$

The derivation of this equation and the corresponding computational procedure for its solution is described in Appendix D. It should be noted that it is necessary to impose boundary conditions in the vertical direction in order to solve Equation 7.3.4. These can easily be obtained by requiring that the velocity disturbance has to approach zero at infinity, i.e. $\lim_{y\to\pm\infty}\tilde{v}(y)=0$, posing a boundary-value problem for Equation 7.3.4.

The overall challenge of this stability analysis is to find a possibly complex frequency ω such that the above boundary value problem has a unique solution. This has to be done for any given set of parameters for the equilibrium state and the spatial wave number k. Since this problem cannot be solved analytically, the differential equation is solved numerically by a shooting method augmented by a root-finding method for ω. The procedure closely follows the work by Blumen (1970), but is implemented here with the built-in numerical functions from the commercial software package Mathematica. It should be noted that the root-finding method for ω requires a good initial guess which should not be too far away from the solution. Otherwise the numerical algorithm does not converge at all. In most situations the initial guess is extrapolated from known solutions with slightly different parameters, e.g. a different wave number k. If this approach fails, a suitable range for the initial value of ω is tested by trial and error. However, it can not be excluded that the numerical algorithm misses some solutions. Therefore, the absence of a solution should not be regarded as proof for its non-existence. With this method, unstable modes with a positive growth rate $\omega_i > 0$ are searched for various parameter values of the equilibrium state.

7.3.2 Results

The semi-analytical stability analysis is used to compute the dispersion relation $\omega(k)$ for various sets of parameters of the equilibrium state. As the main goal of the analysis is the identification of the parameter range for shear instabilities, the following presentation of the results is limited to setups with $\omega_i > 0$. Some examples of such unstable modes, as computed with the stability analysis, are presented in Figure 7.2. The left panel shows the growth rates $\omega_i(k)$ where the parameters are fixed to $\alpha = 0.1, \beta = 100, M_{\mathrm{r}} = 0.001$ while the minimum Richardson number is additionally varied. This setup corresponds to a rather weak temperature gradient which varies on much larger scales than the shear velocity field. For the smallest Richardson number considered here, $Ri_0 = 0.001$, the form of the dispersion relation is very close to the results obtained in the absence of gravity and temperature gradients (cf. Blumen 1970; Miczek 2008). However, as the Richardson number is increased, the growth rates successively decrease. Extrapolating the Richardson number where the maximum growth rate becomes zero results in 0.250. This is in perfect agreement with the critical Richardson number of 1/4, derived within the Boussinesq approximation (cf. Section 7.1). Moreover, it can be seen from the results that the (non-dimensional) horizontal wave numbers of the unstable modes are limited to values in the range between zero and unity. As the wave numbers are non-dimensionalized by the inverse of the length scale of the shear velocity profile, i.e. L_u, it can be concluded that the typical horizontal length scale of shear instabilities is also given by this value. However, this is not necessarily the case for different parameters of the equilibrium state. This can be seen in the right panel of Figure 7.2, where the parameters are

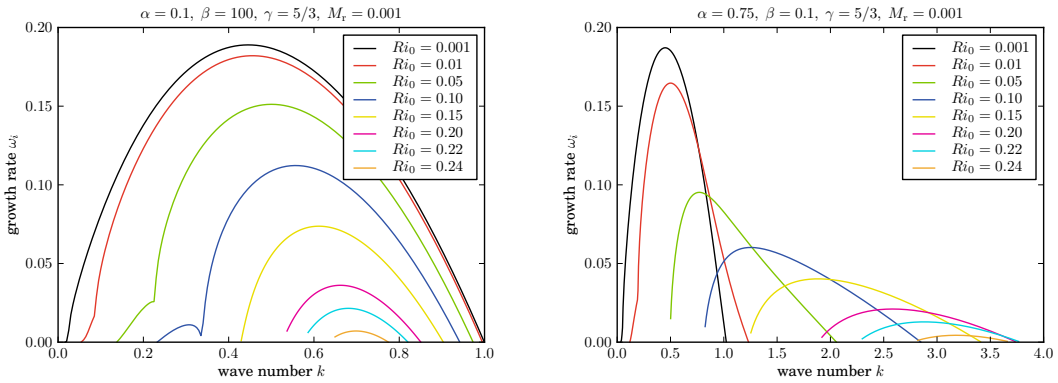
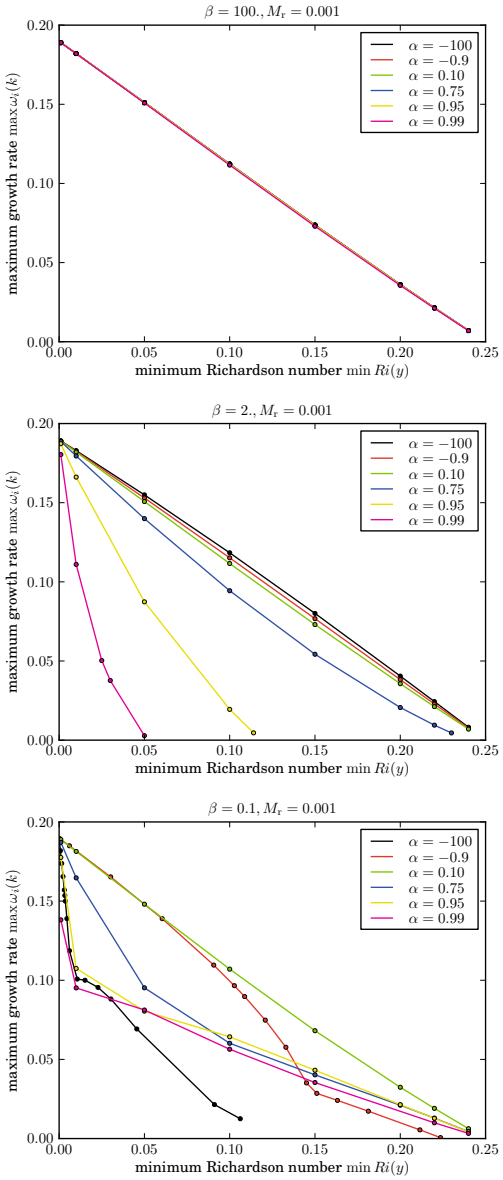


Figure 7.2: Examples of computed growth rates

chosen to $\alpha = 0.75, \beta = 0.1, M_r = 0.001$. The length scale of the temperature gradient (β) is ten times smaller than the length scale of the shear velocity profile here. With increasing Richardson number, the unstable modes have wave numbers well above unity. Therefore, the typical horizontal length scale of shear instability is rather determined by the temperature gradient here. Inspecting the maximum growth rates for this setup, the critical Richardson number can also be extrapolated to 0.250. Thus, no violation of the Boussinesq Richardson criterion is found here.

In order to examine the conditions for instability in a broader way, similar dispersion relations $\omega_i(k)$ have been computed for a large variety of parameters of the equilibrium state. At first, only the reference Mach number is held fixed at $M_r = 0.001$ corresponding to a rather incompressible flow. The strength of the temperature gradient α is chosen from $\{-100, -0.9, 0.1, 0.75, 0.95, 0.99\}$, whereas the corresponding length scale β is varied independently within $\{100, 10, 2, 1, 0.1\}$. For each setup, the dispersion relation $\omega_i(k)$ of unstable modes is computed starting with $Ri_0 = 0.001$. The Richardson number is then successively increased until no unstable modes are found anymore. For each set of parameters, the maximum growth rate with respect to the wave number is identified for the results. The corresponding values are illustrated in the plots shown in Figure 7.3. Each plot corresponds to a certain value of β. For $\beta = 100$, it can be seen that the maximum growth rates have nearly no α dependence. The extrapolated Richardson numbers where the maximum growth rates becomes zero are always 0.250 here. For $\beta = 10$, the maximum growth rates decrease more rapidly with the Richardson number for large values of α. It should be noted that α is limited to values below unity for the setup to be stable against convection. Thus, near the convective stability boundary, the growth rates of shear instabilities tend to decrease here. Contrary to that, strong negative temperature gradients, i.e. $\alpha = -100$, do not influence the growth rates significantly here. The same effects can be seen for $\beta = 2$, where the growth rates and the critical Richardson numbers become even smaller near the convective instability limit. However, the situation changes again for $\beta = 1$. Although the growth rates decrease with increasing α, the critical Richardson number approaches 1/4 in all cases. For $\beta = 0.1$, a similar behavior can be observed for $\alpha > 0$. For negative temperature gradients, the critical Richardson number decreases with the strength of the temperature gradients. It should be noted that in the latter case, the minimum Richardson number of the setup is *not* located at $y = 0$ anymore. However, this effect was taken into account for the analysis of the data. In total, it can be summarized that for $M_r = 0.001$, all growth rates of shear instabilities tend towards zero with critical Richardson numbers below or equal to one fourth. The Boussinesq Richardson criterion is thus not violated here.

In order to investigate the effects of small but finite Mach numbers, the above stability calculated are repeated for $M_r = 0.1$ corresponding to a weakly compressible case. The results are presented in Figure 7.4. It can be seen that the results are qualitatively similar to the previous results at



Extrapolated Richardson numbers where the maximum growth rate approaches zero:

α \ β	100	10	2	1	0.1
−100	0.250	0.250	0.250	0.250	0.127
−0.9	0.250	0.250	0.250	0.250	0.225
0.1	0.250	0.250	0.250	0.250	0.250
0.75	0.250	0.249	0.239	0.250	0.250
0.95	0.250	0.219	0.118	0.249	0.250
0.99	0.250	0.159	0.052	0.249	0.250

Figure 7.3: Maximum growth rates of shear instabilities at $M_r = 0.001$

lower Mach number. In general, the maximum growth rates for shear instabilities seem to be only slightly smaller here. The critical Richardson numbers are also somewhat smaller for certain sets of parameters and do not exceed the 1/4 limit in any case. A similar behavior was found by Blumen (1970) in the absence of gravity and temperature gradients. There, the growth rates of shear instabilities decrease significantly only for Mach numbers well above 0.1. A similar behavior can be expected here within the stratified atmosphere. However, since such high Mach numbers are probably not relevant for differentially rotating stellar interiors, the current analysis is limited to $M_\mathrm{r} = 0.1$.

In total, it can be concluded from the results of the semi-analytical stability analysis that the Boussinesq Richardson criterion is not violated for any parameters tested here. The maximum growth rates always tend towards zero for Richardson numbers below or equal to one fourth. However, it should be kept in mind that computational method used for this analysis may miss unstable modes if the initial guess for the frequency ω is too far away from the solution. In addition to the results presented here, the parameter space for the setup and ω has been excessively sampled in an automated way for $Ri_0 > 1/4$. No unstable modes with $\omega_i > 0$ could be identified with this procedure. With the results presented here, this is at least a strong indication that the Boussinesq Richardson criterion is also valid in fully compressible stratified flows involving possibly strong temperature gradients. Moreover, it should be noted that the results presented here can also be transferred to atmospheres with gradients of the mean molecular weight. As the Euler equations combined with an ideal gas equation of state are only sensitive to the ratio T/μ, the temperature gradients analyzed here can also be interpreted as an inverse gradient in mean molecular weight.

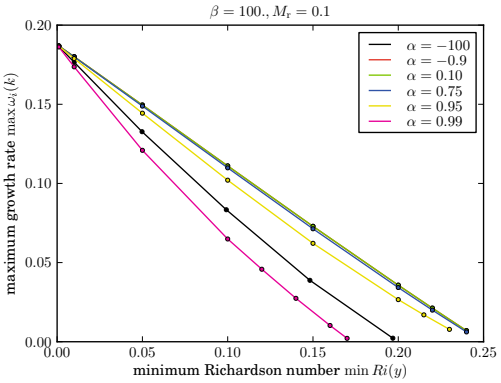
7.4 2D simulations

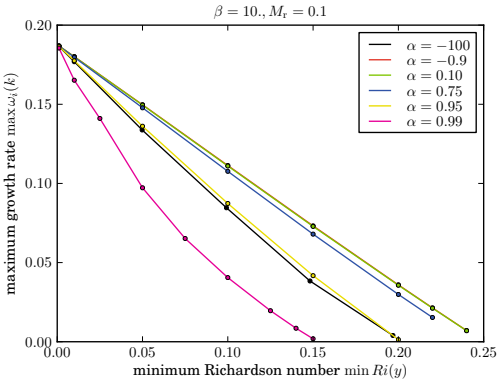
The results of the semi-analytical stability analysis for shear instabilities are only valid in a linear regime where normal modes do not interact with each other by non-linear couplings. In order to test the validity of this analysis, results of two-dimensional numerical simulations of shear instabilities are presented in this section. This is done despite the numerical problems with hydrostatic atmospheres and internal gravity waves described in Chapter 6. The results have therefore be treated with some care, as will be further explained below. For the simulations, the non-dimensional model problem described in Section 7.2 is set up in LHC on a two-dimensional Cartesian grid. As the simulations are intended to resolve a single normal mode of given wave number k, the horizontal dimension of the box is chosen to the corresponding wave length, i.e. $[-\pi/k, +\pi/k]$. Periodic boundary conditions are applied in this direction. In order to facilitate the growth of a single normal mode, the initial profile of the vertical velocity component is slightly disturbed by

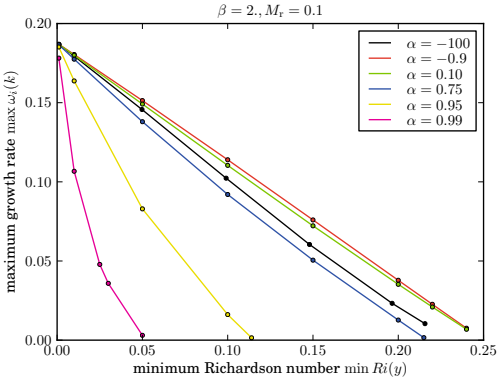
$$v_\mathrm{eq}(x, y) = v_0 \sin(kx) \exp\left(-y^2\right). \tag{7.4.1}$$

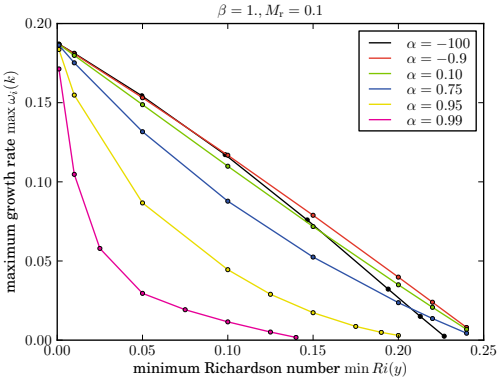
The parameter v_0 is chosen such that it corresponds to a Mach number of 10^{-8} at $y = 0$ which is well below the horizontal shear velocities under consideration. The vertical dimension of the box is chosen to $[-10, +10]$ such that the boundaries are sufficiently far away from the shear layer. The vertical boundary conditions are described by constant ghost-cell values filled with the undisturbed equilibrium profiles. The computational domain is discretized by 64×192 grid cells if not stated differently. The interface reconstruction is performed by linear interpolation in primitive variables. The numerical fluxes are computed with the Roe-Lowmach method at $M_\mathrm{cut} = M_\mathrm{r}$, while gravity is discretized point-wise. The initial conditions are evolved in time with the ESDIRK34 stepper where the size of the time steps is determined with the adaptive error estimator described in Section 5.3.2 with a relative accuracy goal of 10^{-3}. Allowable time steps are limited to advective CFL numbers between 0.25 and 1000.

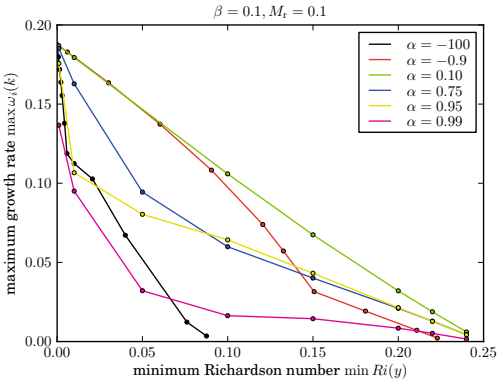
For the first simulation, the parameters of the equilibrium state are chosen to $\alpha = 0.95, \beta = 10, Ri_0 = 0.05, M_\mathrm{r} = 0.001$. For this setup, the semi-analytical stability analysis predicts a maximum in the growth rate at $k = 0.705$, which is used here to define the horizontal extents of the box. The initial conditions for this setup are illustrated in Figure 7.5. The vertical velocity











Extrapolated Richardson numbers where the maximum growth rate approaches zero:

α \ β	100	10	2	1	0.1
−100	0.200	0.202	0.231	0.230	0.092
−0.9	0.250	0.250	0.250	0.250	0.228
0.1	0.250	0.250	0.250	0.250	0.249
0.75	0.249	0.241	0.217	0.249	0.250
0.95	0.243	0.202	0.115	0.216	0.250
0.99	0.173	0.153	0.052	0.148	0.249

Figure 7.4: Maximum growth rates of shear instabilities at $M_r = 0.1$

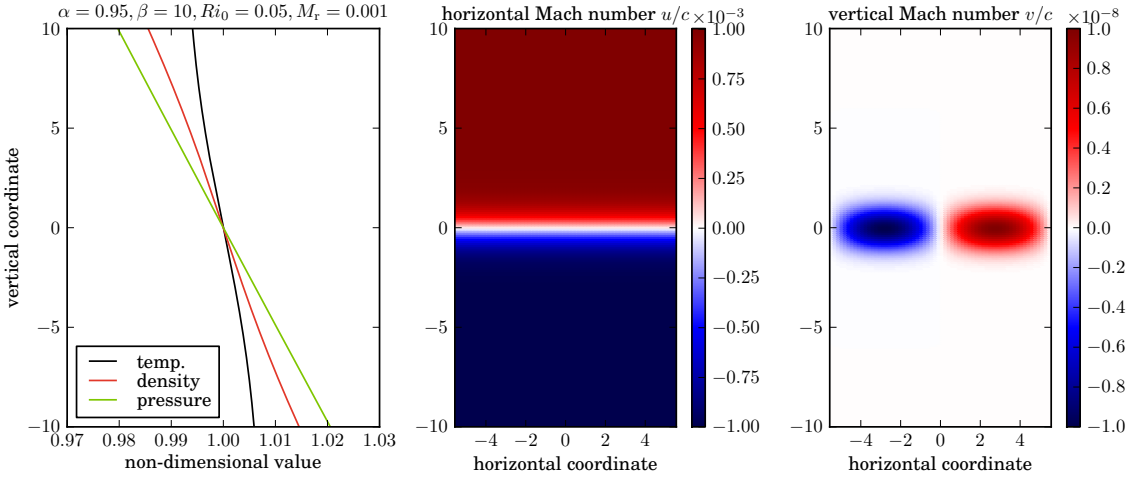


Figure 7.5: Initial conditions for a certain simulation of a single mode shear instability

field after 27 non-dimensional units in time is plotted in the left panel of Figure 7.6. It can seen that the vertical velocities already increased by a factor of ten while the shape of the disturbance changed slightly. As this shape remains constant up to $t \approx 80$, it can be suspected that it corresponds to the normal mode under consideration. In order to quantify the growth of the normal mode, the total kinetic energy in the vertical direction defined by

$$E_{\text{kin},y} = \Delta x \Delta y \sum_{i,j} \frac{1}{2} \rho_{i,j} v_{i,j}^2 \tag{7.4.2}$$

is computed as a function of time. The result is illustrated in the right panel of Figure 7.6. It can be seen that this energy grows exponentially as it is expected from linear theory. In fact, the semi-analytical stability analysis revealed a growth rate of $\omega_{i,\text{stab}} = 0.1426$ for the vertical velocity component. The vertical kinetic energy is thus expected to grow with twice this rate. This can be confirmed by fitting an exponential function to the simulation data. The exponential growth rate in the vertical kinetic energy results in $2 \cdot \omega_{i,\text{sim}} = 2 \cdot 0.1429$. The simulation data thus complies with the stability analysis with a relative difference of only $2.1 \cdot 10^{-3}$. This is a very impressive verification of the semi-analytical stability analysis, at least for the set of parameters used in this test. Moreover, it should be noted that no signs of checkerboard-like patterns or spurious internal gravity waves can be observed in this simulation. This is most likely caused by the fact that these unphysical phenomena grow on a time scale which is significantly larger than the simulation time. These results may therefore be seen as a verification of the employed discretization methods. However, it should be emphasized that such a behavior may not be achievable with arbitrary sets of parameters for this setup.

The exponential growth of the shear instability only lasts up to $t \approx 80$. The vertical kinetic energy starts to saturate at this point. This behavior can be attributed to non-linear effects which are not captured by the stability analysis. In order to visualize the flow field in the non-linear regime, a passive scalar field is evolved with the fluid. It is initialized with the equilibrium shear velocity profile at $t = 0$ and is passively advected with the fluid during the simulation. In the saturation regime of the shear instability, the passive scalar field is illustrated in the left panel of Figure 7.7. It can be seen that the initial shear layer rolls up, building a vortex-like structure. If the simulation would not have been confined to a single normal mode, several vortices would emerge in the non-linear regime, possibly interacting with each other. Although the resulting flow field would look very chaotic, it does not describe turbulent mixing which is an inherently three-dimensional process as explained in the next section.

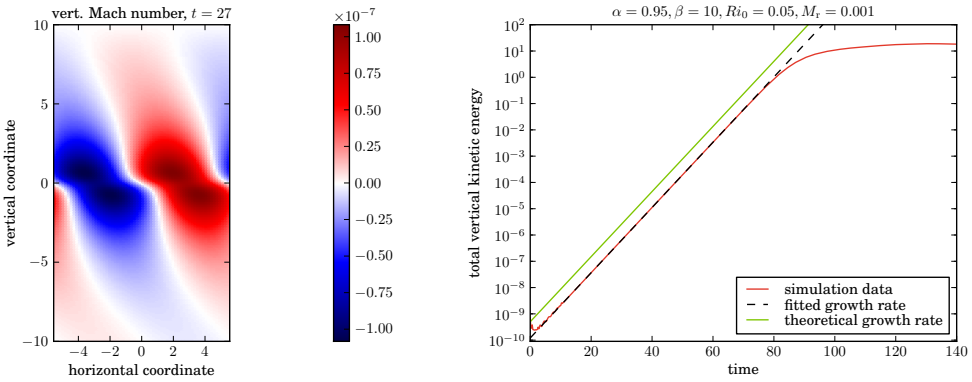


Figure 7.6: vertical Mach number during the growth of the instability (left); evolution of total vertical kinetic energy (right)

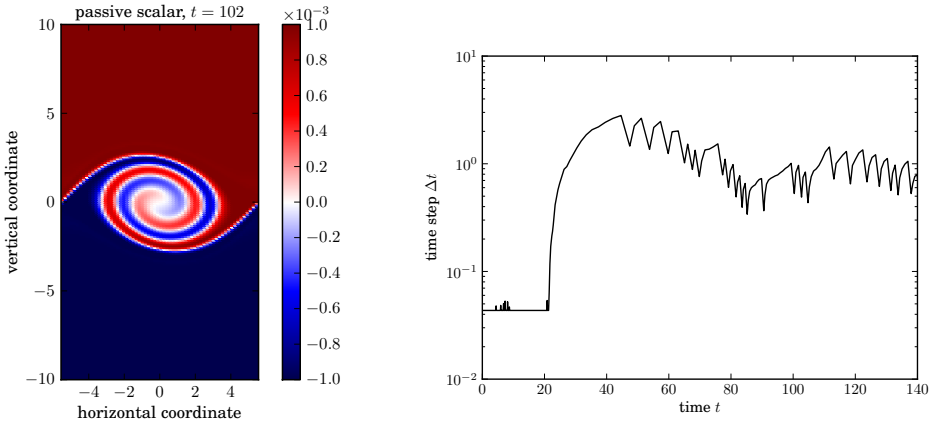


Figure 7.7: Passive scalar field in the non-linear stage at $t = 102$ which was initialized with the horizontal Mach number at $t = 0$ (left); time steps provided by the adaptive ESDIRK error estimator (right)

Another interesting result of the simulation presented here concerns the performance of the adaptive time step chooser which is provided by the error estimator of the ESDIRK schemes. The actual time steps used in the simulation are shown in the right panel of Figure 7.7. It can be seen that the time step is nearly constant at the beginning of the simulation. Here, the time step is limited by the lower advective CFL restriction of 0.25. During the exponential growth of the shear instability, the time step is raised by more than a factor of ten. In the following evolution, the estimated error often hits the relative accuracy goal of 10^{-3} where the time step is halved again. In total, it can be seen that the adaptive time stepping is very efficient here compared to a time step with some fixed CFL number.

Two-dimensional numerical simulations of shear instabilities were also performed with various different sets of parameters for the model problem. In order to cover the parameter range of the stability analysis at least roughly, the parameter combinations $\{\alpha = 0.1, Ri_0 = 0.2\}$, $\{\alpha = 0.75, Ri_0 = 0.15\}$ and $\{\alpha = 0.95, Ri_0 = 0.05\}$ are independently used with $\beta \in \{100, 10, 2, 1, 0.1\}$ and $M_r \in \{0.001, 0.1\}$. For each setup, the wave number $k_{\max}$ with the maximum growth rate is identified within the results of the stability analysis. This wave number is used for the cor-

			$M_\mathrm{r} = 0.001$			$M_\mathrm{r} = 0.1$		
α	β	Ri_0	k_max	$\omega_{i,\mathrm{stab}}$	$\omega_{i,\mathrm{sim}}$	k_max	$\omega_{i,\mathrm{stab}}$	$\omega_{i,\mathrm{sim}}$
0.10	100	0.20	0.660	0.0361	0.0361	0.660	0.0357	0.0388
0.75	100	0.15	0.610	0.0737	0.0740	0.615	0.0713	0.0716
0.95	100	0.05	0.500	0.1511	0.1512	0.505	0.1444	0.1445
0.10	10	0.20	0.665	0.0361	0.0365	0.660	0.0357	0.0392
0.75	10	0.15	0.635	0.0713	0.0716	0.640	0.0680	0.0683
0.95	10	0.05	0.565	0.1426	0.1429	0.560	0.1361	0.1362
0.10	2.0	0.20	0.670	0.0357	0.0355	0.670	0.0353	0.0384
0.75	2.0	0.15	0.815	0.0543	0.0540	0.825	0.0505	0.0551
0.95	2.0	0.05	0.900	0.0874	0.0880	0.885	0.0829	0.0826
0.10	1.0	0.20	0.680	0.0354	0.0353	0.680	0.0350	0.0347
0.75	1.0	0.15	0.920	0.0526	0.0533	0.920	0.0525	0.0571
0.95	1.0	0.05	1.105	0.0909	0.0911	1.120	0.0867	0.0879
0.10	0.1	0.20	0.705	0.0324	0.0312	0.705	0.0320	0.0365
0.75	0.1	0.15	1.885	0.0402	0.0404	1.880	0.0401	0.0404
0.95	0.1	0.05	3.500	0.0805	0.0747	3.500	0.0804	0.0741

Table 7.1: Comparison of growth rates from the semi-analytical stability analysis ($\omega_{i,\mathrm{stab}}$) to the growth rates extracted from 2D numerical simulations ($\omega_{i,\mathrm{sim}}$)

responding numerical simulation with LHC. The setup and the analysis procedure is similar to the one described above. For each setup, a growth rate of the shear instability is extracted from the simulation data and compared to the theoretical prediction from the stability analysis. The results are presented in Table 7.1. It can be seen that the growth rates are in very good agreement for all sets of parameters. Typical deviations are below a one percent level. It should be noted that the simulations with $\beta = 0.1$ were performed with an increased grid resolution of 64 in order to resolved to relatively sharp temperature gradient accurately.

In total, the results of the stability analysis can be fully confirmed by the numerical simulations performed here. Non-linear effects which should be captured by the simulations, do not seem to be important during the initial phase of the shear instabilities and to not alter the growth rates predicted by the stability analysis. Moreover, the good agreement between both methods shows that the numerical discretization techniques used here seem to describe the shear instabilities very accurately. Therefore, it is very desirable to extend the numerical simulations to parameter ranges which are not captured by the stability analysis. In particular, it could be tested whether shear flows at $Ri > 1/4$ are really stable under all circumstances. However, due to the numerical problems described in Chapter 6, such simulations cannot be performed yet. A possibly emerging physical shear instability cannot be distinguished from a spuriously growing internal gravity wave within the simulation data. Numerical tests (not presented here) have shown that this case can indeed occur. Therefore, it is of great importance to resolve these numerical problems in order to perform simulations that may also have predictive power on the physical behavior of a system. However, the above results also show that the discretization techniques used are generally able to capture shear instabilities in the low Mach number regime.

7.5 3D simulation

Shear instabilities may provide a path to turbulent mixing in differentially rotating stellar interiors (e.g. Maeder & Meynet 2000). Since turbulence is an inherently three-dimensional process (see e.g. Pope 2000), it is clear that two-dimensional simulations as presented in the last section are not sufficient for modeling turbulent mixing although growth rates within the linear regime may be accurately calculated. Here, a single numerical simulation of a three-dimensional shear layer is presented and analyzed in detail. The aim of this analysis is to show the path from an unstable

configuration to a fully turbulent flow regime and to demonstrate the basic ability of LHC to handle such flows.

The numerical simulation described in the following is based on the non-dimensional model problem of a shear layer presented in Section 7.2. The two-dimensional equilibrium profiles are mapped into three-dimensional space with no variations in the new coordinate axis. The vertical coordinate is denoted by z here, whereas horizontal coordinate in the direction of the shear is denoted by x. The setup is thus symmetric in the new horizontal coordinate y which may also be referred to as span-wise coordinate. The parameters for the equilibrium state are chosen to $\alpha = 0.75, \beta = 5.0, Ri_0 = 0.1, M_\mathrm{r} = 0.001$. The semi-analytical stability analysis presented in Section 7.3.1 predicts a maximum growth rate of $\omega_i = 0.105$ at $k = 0.615$. However, the simulation presented here is not intended to be restricted to a single normal mode. The horizontal extents of the computational domain are therefore chosen by lower wave numbers. The $x-y$ plane is spanned by $[-\pi/0.125, +\pi/0.125]\times[-\pi/0.1, +\pi/0.1]$ and periodic boundary conditions are assumed in both directions. The vertical extent of the domain is chosen to $[-10, +10]$ as in the two-dimensional simulations. The corresponding boundary conditions are set by constant ghost-cell values filled with the undisturbed equilibrium profiles. The computational domain is discretized by 256^3 grid cells which are initialized with the equilibrium profiles. In order to facilitate the growth of shear instabilities without forcing a specific normal mode, the horizontal velocity components are additionally initialized with random noise at a peak Mach number of 10^{-8}. The interface values are reconstructed by linear interpolation in primitive variables whereas the numerical fluxes are computed with the Roe-Lowmach method at $M_\mathrm{cut} = 10^{-4}$. Gravity is discretized with the point-wise method. The temporal evolution is again computed with the implicit ESDIRK34 scheme with adaptive time stepping. For visualization purposes of the three-dimensional flow field, a passive scalar X, initialized with the horizontal shear velocity, is additionally advected with the fluid.

The simulation was performed at the Rechenzentrum Garching of the Max Planck Society on the IBM Power6 system. Each of the computing nodes is equipped with 32 processors. The parallelization was performed with the Message Passing Interface (MPI), using a decomposition of the computational grid into $n_x \times n_y \times n_z$ equally-sized blocks which are each handled by a single processor. Before performing the actual computation of the shear layer, the parallel performance is evaluated by calculating a single implicit time step of the setup with different numbers of processors. Due the large memory requirements of the grid and the data structures for the linear solvers, the smallest number of processors that can be used here is 32. The execution time of this run serves as a reference case to calculate the speed-up factor with higher numbers of processors. The results are summarized in the following table:

nodes	domain decomposition	processors	execution time	speed-up
1	$4 \times 4 \times 2$	32	2408s	
2	$4 \times 4 \times 4$	64	1191s	2.02
16	$8 \times 8 \times 8$	512	141.3s	17.04

It can be seen that the execution time halves as the number of processors is doubled here. The parallel efficiency is thus optimal. Increasing the number of processors to 512, an optimal speed-up of 16 is expected. However, the corresponding execution time is even somewhat *below* this value. The speed-up is thus super linear. This remarkable result may have several origins. First, the amount of processor cache memory per local grid block increases with the number of processors favoring such a behavior. Second, the solvers for the linear systems are not perfectly deterministic anymore in parallel computations. This is caused by global reduction operations which are for example needed to calculate scalar products between two vectors. Thus, it might happen that the internal number of iterations within the linear solvers varies slightly for different parallel configurations. The observed super linear speed-up would then be just incidental. In total, this performance test shows that it is indeed possible to perform time-implicit simulations on parallel computer architectures in an efficient way. For the actual simulation of the shear layer, the domain

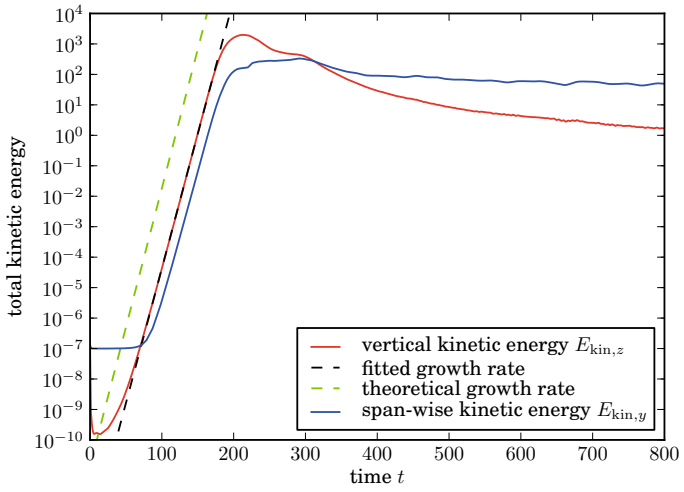


Figure 7.8: Evolution of total vertical and span-wise kinetic energy

decomposition was chosen to $8 \times 8 \times 8$ blocks, corresponding to 512 processors. The simulation was carried out until $t = 820$ in physical time was reached with needed approximately 48 hours of computing time.

In order to analyze the simulation results, the total vertical kinetic energy, defined by

$$E_{\mathrm{kin},z} = \Delta x \Delta y \Delta z \sum_{i,j,k} \frac{1}{2} \rho_{i,j,k} w_{i,j,k}^2, \tag{7.5.1}$$

is computed as a function of time. The results are shown in Figure 7.8. It can be seen that the vertical kinetic energy grows exponentially until $t \approx 200$. The corresponding growth rate is fitted and leads to $2 \cdot \omega_i = 2 \cdot 0.102$. This is in very good agreement with the theoretical growth rate of the fastest growing mode for this setup, given by $\omega_i = 0.105$, which can be obtained by the (two-dimensional) semi-analytical stability analysis. This is particularly remarkable as neither the box geometry nor the initial disturbances of the simulation setup force the growth of this specific normal mode. Thus, so far the simulation results are fully compatible with the two-dimensional investigation. In order to analyze three-dimensional effect, the total kinetic energy

$$E_{\mathrm{kin},y} = \Delta x \Delta y \Delta z \sum_{i,j,k} \frac{1}{2} \rho_{i,j,k} v_{i,j,k}^2 \tag{7.5.2}$$

is also computed in span-wise (y) direction. The result is also shown in Figure 7.8. It can be seen that the span-wise kinetic energy also grows exponentially with approximately the same rate. However, compared to the vertical kinetic energy, the growth is a bit delayed here. Only in the non-linear regime of the shear instability after $t \approx 200$, the span-wise kinetic energy overtakes the vertical kinetic energy. Both energies are slowly decreasing in this regime.

In order to get an impression about the three-dimensional evolution of the flow field, the passive scalar X is visualized in Figure 7.9 at various instants in time. Each plot shows the isosurface $X = 0$ which is colored by the background pressure for better readability. Initially, the isosurface is flat corresponding to the $z = 0$ layer. At $t = 187$, the total kinetic energy in vertical direction has nearly reached its maximum. It can be seen that the fluid rolls up as in the two-dimensional case (cf. Figure 7.7). However, the setup is already not perfectly symmetric in the span-wise direction anymore. In the further evolution, the vortex-like rolls become unstable and successively break

up. At $t = 319$, the passive scalar field has become completely chaotic. It can be suspected that the flow field becomes fully turbulent at this point. However, the turbulence seems to decay in the further evolution as the passive scalar field gets more calm again. The overall behavior of the flow described here is typical for the transition of a three-dimensional shear layer to a turbulent flow regime. The two-dimensional vortex-like structures are in fact unstable to span-wise disturbances which can be confirmed by three-dimensional stability analyses (see e.g. Peltier 2003).

In the following, the properties of the turbulent flow within the shear layer are analyzed in more detail by examining the spectral distribution of kinetic energy. The analysis is performed within a horizontal layer located at $z = 0$ where turbulence is expected to be strongest. As the boundary conditions of the simulation in the horizontal directions are periodic, spectral properties can be easily obtained by Fourier transformations. To this end, a two-dimensional discrete Fourier transformation (DFT2D) is calculated for each component of the velocity field separately:

$$\begin{aligned} \hat{u}\left(k_x, k_y\right) &= \mathrm{DFT2D}\left(u\left(x, y, z = 0\right)\right) \\ \hat{v}\left(k_x, k_y\right) &= \mathrm{DFT2D}\left(v\left(x, y, z = 0\right)\right) \\ \hat{w}\left(k_x, k_y\right) &= \mathrm{DFT2D}\left(w\left(x, y, z = 0\right)\right) \end{aligned} \tag{7.5.3}$$

A measure for the spectral kinetic energy can then be obtained by (Schmidt 2004)

$$\hat{E}_{\mathrm{kin}}\left(k_x, k_y\right) = \frac{1}{2}\left(u\, u^{\star} + v\, v^{\star} + w\, w^{\star}\right), \tag{7.5.4}$$

where the star denotes the complex conjugate. In order to further simplify the analysis, the spectral kinetic energy is averaged on circles with constant $|k| = \sqrt{k_x^2 + k_y^2}$ in the two-dimensional wave number space. The corresponding computational procedure closely follows the work by Schmidt (2004) and leads to the spectral kinetic energy $\bar{E}_{\mathrm{kin}}\left(|k|\right)$ which only depends on the magnitude of the horizontal wave number. Double logarithmic plots of kinetic energy spectra are shown in Figure 7.10 (left panel) at three instants in time. It can be seen that the overall kinetic energy decreases with time, which is consistent with the results shown in Figure 7.8. Moreover, below a certain threshold wave number, the spectra tend towards straight lines in double logarithmic space, corresponding to a $k^{-5/3}$ scaling. This is exactly the famous theoretical scaling found by Kolmogorov (1941) for incompressible turbulence at high Reynolds numbers. Thus, at least in the horizontal plane under consideration, the simulation code is able to resolve a turbulent cascade where large eddies successively decay towards smaller scales in a self-similar way. This result also implies that the turbulence is isotropic in the horizontal direction. Only at wave numbers above $k = 2\pi/(8\Delta y)$, the Kolmogorov scaling and thus the turbulent cascade breaks down. This is the expected behavior in the regime where turbulent eddies get dissipated into heat (see e.g. Pope 2000). It should be noted that the simulation does not contain any physical viscosity. Therefore, the turbulent cascade should ideally reach wave numbers corresponding to the size of the grid cells. In the context of large-eddy simulations, such a behavior would be favorable. Here, the turbulent cascade breaks down at length scales smaller than $8\Delta y$ (the span-wise grid spacing is the coarsest here). On these length scales, the kinetic energy of the turbulent flow gets converted into heat solely by the numerical dissipation inherent in the discretization scheme. It should be noted that this eight grid cell limit observed here is quite good for a numerical method which does not involve a sub-grid scale model for turbulence (cf. Schmidt et al. 2004). The simulation results obtained with LHC in turbulent flow regimes can be regarded as an implicit large-eddy simulation (see e.g. Grinstein et al. 2007) on scales larger than eight grid cells.

Finally, the Richardson number as a function of the vertical coordinate is calculated from the simulation data. Defining horizontal averages of the temperature and the velocity field, the local Richardson number can be approximated by

$$Ri(z) = \frac{g^2}{\tilde{R} < T > \left(\frac{d<u>}{dz}\right)^2}\left(\frac{\gamma - 1}{\gamma} + \frac{\tilde{R}}{g} \cdot \frac{d < T >}{dz}\right). \tag{7.5.5}$$

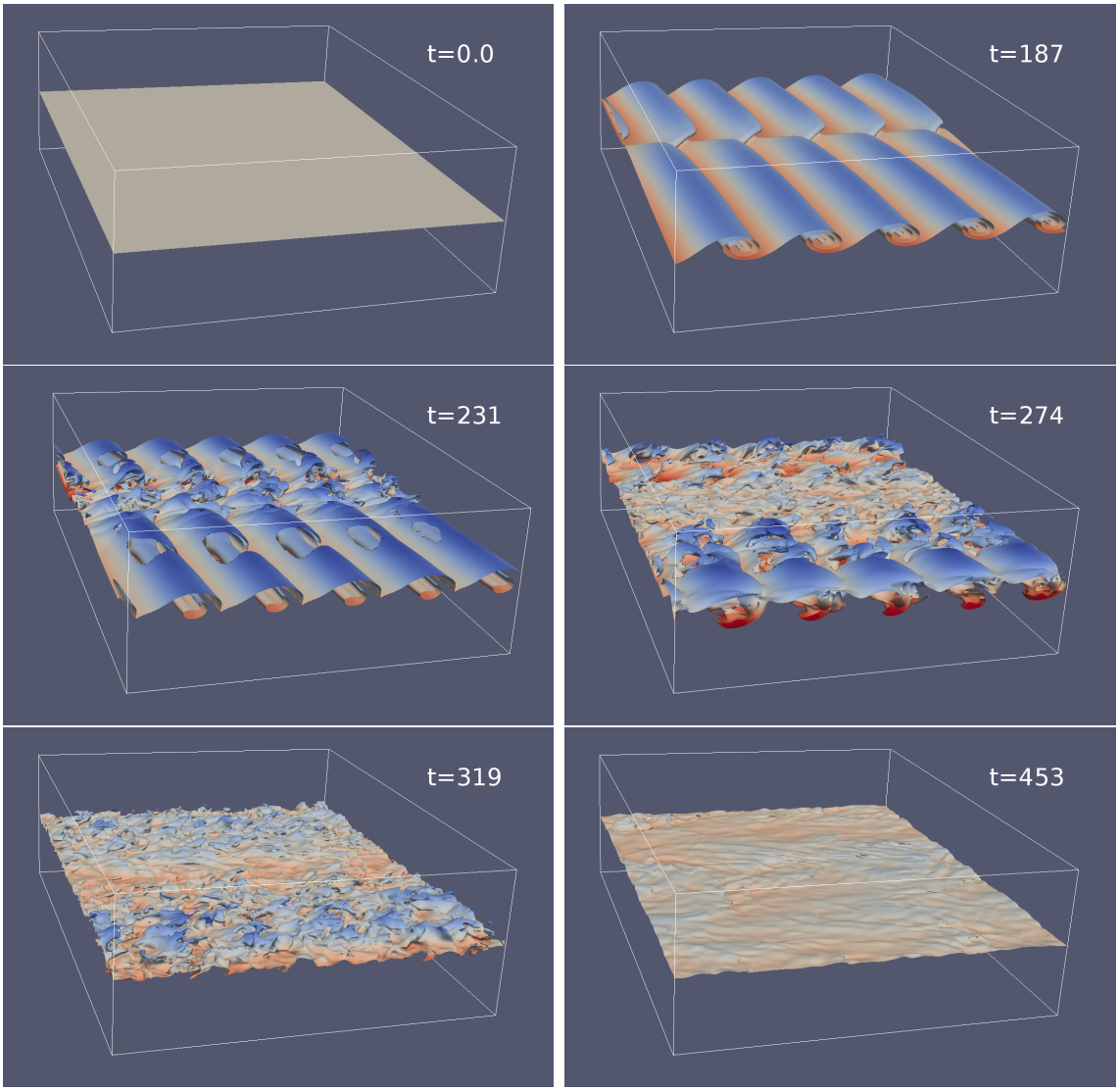


Figure 7.9: Temporal evolution of a passive scalar field X initialized with the horizontal shear velocity. The plots show the isosurface $X = 0$.

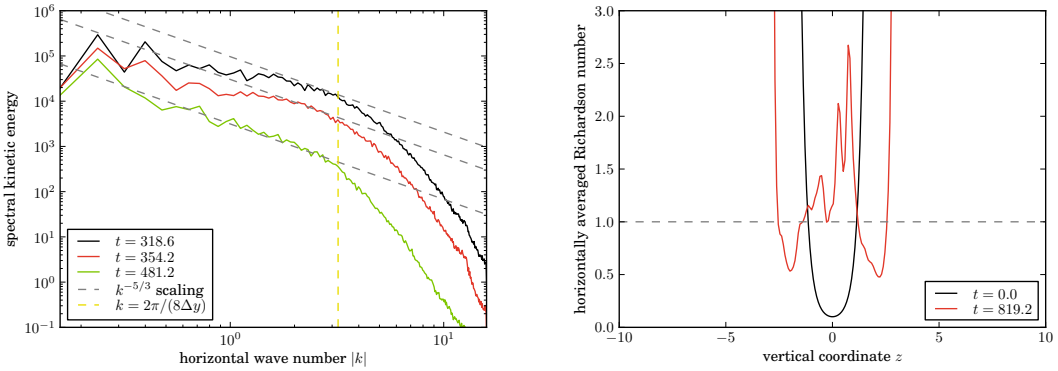


Figure 7.10: kinetic energy spectrum in the $z = 0$ plane (left); horizontally averaged local Richardson numbers at the beginning and the end of the simulation (right)

This function is illustrated in Figure 7.10 at the beginning and the end of the simulation at $t = 819.2$. It can be seen that the initial Richardson number distribution has a rather sharp minimum at $z = 0$ where it approaches $Ri_0 = 0.1$ as given by the setup for this simulation. At the end, the Richardson numbers are somewhat noisy which probably corresponds to the fact that turbulence has not fully decayed at this point. However, it can be seen that the Richardson number distribution has become somewhat broader. Averaging the Richardson number between $z = -2.5$ and $z = +2.5$ results in

$$\frac{1}{5}\int_{-2.5}^{+2.5} Ri(z)dz \approx 1.08. \tag{7.5.6}$$

This is in good agreement with the results given by Canuto (2002) who theoretically derived that shear turbulence can be maintained up to a critical Richardson number of unity. From the results obtained here, it seems that this is a natural final state of a shear instability which is not in contradiction with the classical Richardson criterion. However, it should be noted that the present results are still too noisy in order to fully confirm this statement.

From the results presented in this chapter, it can be concluded that there is no indication that the Boussinesq Richardson criterion is violated in fully compressible stratified atmospheres. However, further numerical simulations in the $Ri > 1/4$ regime are needed in order to verify this statement. Moreover, the three-dimensional simulation presented here revealed that LHC is basically able to capture the turbulent flow field which may be produced within shear layers. Therefore, an accurate derivation of turbulent diffusivities from numerical simulations seems to be possible, although this is subject to future work. The simulation performed here also emphasized that turbulence produced by shear instabilities is an inherently three-dimensional process which cannot be modeled in a two-dimensional way. Although three-dimensional simulations of low Mach number flows are computationally quite expensive, it has been shown that they are generally affordable on modern parallel computer systems.

8 Summary and outlook

Stellar evolution modeling poses extreme multi-scale problems as the disparity of involved time scales is very large. In order to capture hydrodynamic processes in stars even on very short periods of time by multi-dimensional numerical simulations, it is necessary to resolve low Mach number flows in these stratified atmospheres in an accurate and efficient way. It has been shown in Chapter 2 that the compressible Euler equations permit two classes of solutions as the Mach number tends towards zero, namely sound waves and incompressible flow. The latter is of special importance as incompressible flow fields stay in this regime as they are evolved in time. Opposed to that, sound waves do not seem to be of great importance for stellar evolution modeling. However, it has been argued in Section 2.7 that it is still desirable to simulate the fully compressible Euler equations without restricting them to a submodel, such as the Boussinesq approximation or the anelastic approximation which do not permit sound waves at all.

In order to perform numerical simulations of low Mach number flows, a new code called LHC has been developed in the context of this work. In Chapter 3, the basic methods for the spatial discretization of the Euler equations were presented which are based on a finite volume approach. Common methods to compute numerical fluxes include Godunov-like schemes which solve Riemann problems at each interface between two discrete grid cells. However, these methods were originally designed for high Mach number flows where an accurate capturing of shock waves is essential. As such discretizations are consistent with the Euler equations, they should reveal the same continuous solutions as the mesh size tends towards zero. However, it has been shown in Chapter 4 by numerical tests, that discrete solutions obtained in the low Mach number regime are very inaccurate, even for moderately high grid resolutions. In particular, Godunov-like numerical fluxes show an excessive numerical viscosity which increases with decreasing Mach number. Thus, kinetic energy is converted to internal energy on very short time scales which is in severe contradiction with an incompressible flow field where kinetic energy should be conserved. It has been shown that this behavior can be attributed to the upwinding terms in the numerical discretization which are generally necessary to maintain numerical stability. However, in the low Mach number regime, the Godunov-like upwinding terms act mainly as numerical dissipation which becomes so strong that they even overwhelm any physically motivated flux.

A widespread method to cure this problem is provided by the Roe-Turkel scheme. It uses a technique called flux preconditioning which alters the upwinding terms in order to reduce the excessive numerical viscosity. However, it has been shown by an asymptotic scaling analysis that the Roe-Turkel scheme introduces an additional numerical diffusion term in the energy equation. This term does not seem to be harmful for typical applications in the engineering community where the Roe-Turkel scheme was originally developed to simulate low Mach number flows. However, it has been shown that the additional numerical diffusion becomes very excessive for nearly hydrostatic atmospheres. The potentially large pressure gradient in the vertical direction supports the numerical diffusion and may easily lead to unphysical changes of the hydrostatic temperature profile. In order to cure this problem, a new numerical flux function, called Roe-Lowmach was proposed in this work. It is also based on a combination of the Roe scheme and the flux preconditioning technique, but uses a different preconditioning matrix such that the resulting upwinding terms show the same asymptotic scaling with the Mach number as the continuous fluxes of the Euler equations. In addition to that, the new Roe-Lowmach scheme should be more appropriate for nearly hydrostatic flows. Beside that, two different approaches from literature to construct numerical fluxes in the low Mach number regime were discussed. First, it has been shown that the so-called low Mach number reconstruction exhibits an asymptotic scaling of the upwinding terms which may give rise to numerical instabilities. Second, the AUSM$^+$-up scheme, which is

also designed for low Mach number flows, has been analyzed in more detail. It has been shown here that the scheme exhibits the same behavior as the Roe-Turkel scheme in the low Mach number regime and is thus potentially not suitable for nearly hydrostatic flows. Therefore, a small modification of the scheme is proposed here in order to reduce the excessive numerical diffusion in the energy equation. The resulting flux is called AUSM$^+$-Lowmach here. Finally, it has been shown that all numerical flux functions for low Mach number flows are generally able to resolve the incompressible flow regime as the dissipation of kinetic energy does not depend on the Mach number anymore. From theoretical considerations, the Roe-Lowmach and the AUSM$^+$-Lowmach fluxes are considered to be most appropriate for low Mach number flows in nearly hydrostatic atmospheres.

In Chapter 5, it has been shown that explicit temporal discretizations are computationally very inefficient in the low Mach number regime. Typical time steps are limited by an acoustic CFL criterion in order to maintain numerical stability. Thus, even if sound waves are not present in a specific setup, it takes tremendous amounts of time steps to advance a discrete solution in the low Mach number regime. In contrast to that, the time step is not limited in this way if implicit temporal discretizations are used. In particular, the time step can be chosen based on an advective CFL criterion such that fluid elements within the discrete solution may propagate of the order of one grid cell per time step. Compared to explicit methods, suitable time steps for implicit methods may therefore be a factor of $1/M$ larger which may become quite substantial for very subsonic flows. For the new simulation code LHC, the backward Euler method and a family of implicit Runge-Kutta (ESDIRK) schemes were implemented. The latter provide numerical solutions which are up to fifth-order accurate in time. Moreover, ESDIRK schemes provide an inherent method to estimate the temporal discretization error. This can be used to choose the size of the time step in an adaptive way.

The general drawback of implicit methods is that the computational work per time step is much higher than for explicit methods. The reason for this is that huge systems of non-linear algebraic equations have to be solved in each step. In the context of low Mach number flows, it is very challenging to solve these equations with common techniques such as non-linear multigrid or pseudo-transient methods (Miczek 2008). Therefore, a Newton-Raphson approach is proposed here which reduces the non-linear problem to a sequence of linear problems. However, the corresponding matrices of these huge linear systems can occupy substantial amounts of memory of modern computer systems. For a large number of scalar fields, such as chemical element abundances the memory requirements may even become impracticable. Therefore, a new method is proposed here to split the non-linear equations into a hydrodynamic part and a passive scalar part. The first one can be solved without the knowledge of the solution of the second one. The equations for each passive scalar can then be solved independently which greatly reduces the computational complexity. The overall solution is fully compliant with the original system. Moreover, it has been argued that this method may also be applied for active scalar fields. The overall solution is only an approximation to the original system in this case. However, as long as active scalars only have minor impact on the hydrodynamics, the numerical solution may still be very accurate.

In order to compute the matrices for the linear equations, it has been shown that it is possible to derive mostly analytic expressions for the matrix elements. This can be achieved by splitting up the computation into logical parts corresponding to individual discretization steps with the use of the chain rule of differentiation. The actual solution of the linear equations is then the most critical part concerning the computational efficiency. It has been argued that direct solvers are not able to achieve this goal as memory and computing time requirements quickly become excessive. However, their implementation is still desirable for diagnostic purposes on very small problems. More promising iterative linear solvers include Krylov-subspace methods such as GMRES(r) and Bi-CGSTAB(ℓ) combined with a linear multigrid solver. A particular advantage of these methods is that they can easily be parallelized for the use on large-scale computing systems by domain decomposition methods. In the context of this work, all parts of the LHC simulation code have been parallelized in this way for computing architectures with shared and/or distributed memory

systems. For the latter case, the parallel efficiency has been demonstrated in Section 7.5 for a moderately large three-dimensional setup. In order to further facilitate the iterative solution of the linear equations, several matrix preconditioning strategies were discussed in this work. The overall efficiency of implicit time stepping compared to explicit time stepping was shown in Section 5.6.2. In the present version of LHC, implicit time stepping becomes superior in terms of computational work for flows with Mach numbers below ≈ 0.05. It has also been shown that the performance does not deteriorate on large grids. These promising results may critically depend on a suitable choice of linear solvers and matrix preconditioners. However, the "standard approach" presented here seems to be quite robust for various different kinds of numerical setups. Moreover, it should be noted that there is even some potential to further improve the computational efficiency. This could be achieved by an automatic determination of the best parameters for the linear solvers depending on the specific flow situation. However, in terms of computational efficiency, LHC can readily be used for practical applications for simulations of stellar interiors.

Implicit time stepping methods have also been evaluated in terms of accuracy. It has been shown that implicit methods are generally able to resolve sound waves. However, this requires that the size of the time step is again limited by an acoustic CFL criterion (which is not desirable in terms of computational efficiency). The accuracy is then similar to an explicit method of equal order in this case. As the size of the implicit time steps is increased, sound waves are successively damped out in the numerical solution. This behavior does not affect the accuracy of advection waves also present in such a flow. In fact, the size of the time step can be increased up to advective CFL numbers of the order of one. Then, advection waves also become inaccurate. In total, it can be seen that implicit methods do not restrict the size of the time step in order to maintain numerical stability. The time step can be chosen based on the physical process of interest which has to be accurately resolved in time. Such processes may also be given by additional physics such as gravity, thermal radiation or nuclear reactions. Thus, implicit time stepping methods may open a path towards the investigation of further interesting flow regimes by numerical simulations.

Hydrodynamic flows in stellar atmospheres are expected to be very close to a hydrostatic equilibrium. The discretization of gravity is therefore of particular importance for numerical simulations. A suitable scheme should ideally be able to maintain hydrostatic equilibria up to machine precision and resolve small deviations from this state accurately. In Chapter 6, the properties of several discretizations of gravity and their interplay with the numerical flux function were discussed in detail. For a pointwise discretization of gravity, a modified equation analysis has been conducted in order to investigate numerical diffusion terms for hydrostatic initial states. It has been shown that the magnitude of these terms scales with the cut-off Mach number of the Roe-Lowmach flux. The original version of Roe's approximate Riemann solver is therefore expected to deteriorate hydrostatic equilibria while the new Roe-Lowmach solver should exhibit a much better behavior. Apart from that, two different discretizations of gravity were presented which are theoretically able to maintain hydrostatic equilibria exactly and are therefore called well-balanced schemes. Extensions of these methods were proposed here in order to reduce the numerical viscosity for low Mach number flows.

All discretizations of gravity were first tested on a one-dimensional hydrostatic atmosphere. The pointwise gravity combined with the Roe-Lowmach flux seems to be nearly well-balanced although this result cannot be proved theoretically at this point. Without flux preconditioning, the hydrostatic atmosphere deteriorates on very short time scales as expected. Contrary to that, both well-balanced schemes show very good behavior in this test. The proposed extensions for low Mach number flows are also able to maintain the hydrostatic equilibrium under the condition that the corresponding cut-off Mach number is set to a very low value.

These promising results could unfortunately not be confirmed in numerical tests of a two-dimensional hydrostatic atmosphere. During its temporal evolution, horizontal density fluctuations build up in an unphysical way and grow exponentially. This hydrostatic numerical instability manifests itself in a checkerboard-like structure on the grid which eventually becomes so strong that the hydrostatic atmosphere is completely destroyed. The instability occurs with all kinds of low Mach number numerical fluxes. Contrary to that, ordinary flux functions seem to be stable,

which can be attributed to the fact that they contain an excessive numerical viscosity. These flux functions are therefore not suitable for the simulation of nearly hydrostatic flows anyway. Moreover, it has been shown that the hydrostatic numerical instability is neither caused by a specific part of the numerical discretization nor by a specific low Mach number numerical flux. The instability can be seen in a broad range of hydrostatic atmospheres, but the corresponding growth rates may differ significantly. It has further been shown that the inclusion of additional artificial viscosity generally resolves the problem. However, the required viscosities are far too high for any practical application. Moreover, it has been shown that the hydrostatic numerical instability can also be seen in another simulation code which relies on a very different discretization method. Therefore, this instability seems to be a rather general problem for numerical discretizations in hydrostatic atmospheres with low (numerical) viscosity. However, to the author's knowledge, this problem has not been described in literature yet. Future investigations should therefore concentrate on the exact origins of this instability in order to design a numerical discretization which is more suitable for such flows. It should be noted that well-balanced schemes combined with low Mach number methods seem to weaken the instability. It can be suspected that a discretization of gravity whose discretization error is more compatible to the corresponding error of the hydrodynamic flux may provide a path to a numerically stable scheme. However, the necessary conditions for such a method remain unknown so far and are subject to future work.

Beside maintaining a hydrostatic equilibrium, a proper numerical discretization should also be able to resolve small deviations from this state accurately. It has been shown that such fluctuations propagate as internal gravity waves through the stratified atmosphere. They can be described theoretically within the Boussinesq approximation. Numerical tests of internal gravity waves showed that ordinary numerical flux functions are too dissipative in order to resolve such waves even when they are combined with a well-balanced scheme. In contrast to that, low Mach number methods generally show a very promising behavior. Unfortunately, internal gravity waves are also plagued by a spurious rise of their amplitudes which is in contradiction with theoretical expectations. As the corresponding growth rates seem to be directly related to the growth rates of the hydrostatic numerical instability, it can be suspected that both problems share the same origin. However, in an internal gravity wave setup, the numerical instability does not manifest as a checkerboard-like pattern. Thus, using the present discretization techniques for more complicated problems, it is generally not possible to identify spurious numerical instabilities. Practical applications of LHC for stellar atmospheres should therefore be treated with much care at the moment. Apart from that, the numerical tests also revealed that implicit time marching schemes are generally suitable for resolving internal gravity waves. The size of the time step should then be determined by the Brunt-Väisälä-frequency of the setup or by the free-fall CFL criterion proposed in this work. As such time steps are generally much larger than time steps based on the acoustic CFL criterion, implicit methods are also superior here in terms of computational efficiency compared to explicit methods. In general, it can be seen from (non-linear) theoretical considerations and investigations of earth's atmosphere that internal gravity waves show a large variety of new physical phenomena. In particular, they may provide a direct mechanism for energy transfer from large to small scales and may therefore contribute to turbulent mixing and angular momentum transport in stellar interiors. As such investigations have not attracted much attention in the astrophysical community, future numerical simulations of internal gravity waves may provide import new insights.

In the last chapter, the prospects of hydrodynamic modeling of stellar interiors were shown by the example of shear flows. The conditions and the growth rates for shear instabilities were first investigated by a compressible stability analysis in a stratified atmosphere. The results are fully compatible with the classical Richardson criterion with a critical Richardson number of one fourth. This could also be confirmed by two-dimensional numerical simulations of shear flows. The extracted growth rates of the shear instabilities actually equal the growth rates of the stability analysis up to a relative accuracy of less than one percent. Fortunately, the simulation results do not seem to be plagued by spurious numerical instabilities which probably grow on time scales much longer than the typical growth time of the physical shear instability. At the moment, this limits numerical simulations to setups which can also be treated with the semi-analytic stability

analysis. It is very desirable to extent the numerical simulations to shear layers at Richardson numbers above one fourth in order to test the physical stability of the flow. Such simulations can be performed once the numerical problems are solved and thus remain subject of future work.

A single shear layer has also been investigated with a three-dimensional simulation at moderately high grid resolution of 256^3. In accordance to theoretical predictions, the two-dimensional vortex-like structures of the unstable shear layer break up in the span-wise direction, leading to a fully turbulent flow field. The corresponding kinetic energy spectrum extracted from the simulation data actually shows a Kolmogorov scaling in the horizontal plane. The turbulent cascade breaks down at length scales of only eight grid cells. This remarkably good result emphasizes that the present discretization techniques are well-suited to capture turbulence in the low Mach number regime. Thus, once the numerical problems are solved, LHC can be readily used to investigate turbulent mixing processes in stellar interiors. The present results also emphasize that it is necessary to perform such simulations in three dimensions. Although the computational demand is very high in this case, it has been shown that three-dimensional simulations are affordable due to the good performance of LHC on parallel computing systems. It should also be noted that future simulations of shear layers can easily be extended to setups involving strong thermal radiation whose source terms are already discretized. It should then easily be possible to examine the theoretical predictions given by Lignières et al. (1999) with numerical simulations. This could be a first case where the thermal time scale of stellar interiors can be resolved by implicit calculations.

In total, this work showed the prospects of a hydrodynamic treatment of stellar interiors with numerical simulations. As the Mach numbers of the involved flows are typically very low, it is necessary to use adequate spatial discretization techniques which are able to resolve flows without introducing an excessive numerical viscosity. In this respect, the low Mach number flux functions presented in this work show a very promising behavior. This is also a very instructive example that existing numerical codes should not be applied to new flow regimes without evaluating the suitability of the numerical discretization by comprehensive numerical tests. Although a numerical discretization may be consistent with the original equations, an accurate treatment of certain flow regimes may require a numerical resolution that is practically not affordable. It could further be shown in this work that temporally implicit calculations are able to resolve stellar flow structures of interest without restricting the time step to the dynamics of sound waves. The implementation of the implicit solver in LHC has been proven to be computationally efficient, even on parallel computers. Interesting future applications of the code include the various hydrodynamic and thermal instabilities in differentially rotating stars, stellar convection and the rich phenomena introduced by internal gravity waves. Resolving the remaining numerical challenges, such simulations may provide important new insights for stellar evolution modeling.

A Implementation of the Roe-Lowmach scheme

As explained in Section 4.3.3, the implementation of the Roe-Lowmach flux is very prone to numerical round-off errors. In fact, evaluating the expressions for the numerical flux in a straight forward way quickly results in NaN (Not a Number) values on the computer. Therefore, the detailed implementation of the Roe-Lowmach flux in LHC is presented in the following, where many terms where hand-optimized by common methods in order to avoid round-off errors.

The starting point for the flux evaluation are the reconstructed conservative variables $\mathbf{U}^L_{i+1/2}$, $\mathbf{U}^R_{i+1/2}$ approaching an interface $i+1/2$ from the left and right hand side. For both states, the corresponding pressure and its derivatives are calculated by the equation of state. With these values, the Roe-averaged quantities as defined in Section 3.5.2 are calculated. Moreover, the Roe-averaged interface velocity, speed of sound and Mach number are computed by

$$q_n = n_x u_{\mathrm{roe}} + n_y v_{\mathrm{roe}} + n_z w_{\mathrm{roe}}, \tag{A.1}$$

$$c = \sqrt{\left.\frac{\partial p}{\partial \rho}\right|_{\mathrm{roe}} + \left.\frac{\partial p}{\partial \epsilon}\right|_{\mathrm{roe}} \cdot \frac{\epsilon_{\mathrm{roe}} + p_{\mathrm{roe}}}{\rho_{\mathrm{roe}}}}, \tag{A.2}$$

$$M = \frac{q_n}{c}. \tag{A.3}$$

The preconditioning parameter and the preconditioned acoustic eigenvalues then read

$$\delta = \frac{1}{\min\left(1, \max\left(M, M_{\mathrm{cut}}\right)\right)} - 1, \tag{A.4}$$

$$\tau = \sqrt{\frac{c^2}{M_{\mathrm{r}}^2} + \left(\frac{c^2}{M_{\mathrm{r}}^2} - q_n^2\right)\delta^2}, \tag{A.5}$$

$$\lambda_1 = q_n - \tau, \tag{A.6}$$

$$\lambda_2 = q_n + \tau. \tag{A.7}$$

Analyzing the structure of the upwinding matrix in primitive variables $D_{\mathbf{V}} = P_{\mathbf{V}}^{-1}\left|P_{\mathbf{V}} A_{\mathbf{V}}\right|$, it can be shown that the matrix can be written in the following form:

$$D_{\mathbf{V}} = a_\chi \begin{pmatrix} |q_n| & n_x S_1 & n_y S_1 & n_z S_1 & \frac{S_3 - |q_n|}{c^2} & 0 \\ 0 & |q_n|\left(n_y^2 + n_z^2\right) + S_3 n_x^2 & n_x n_y\left(S_3 - |q_n|\right) & n_x n_z\left(S_3 - |q_n|\right) & n_x S_2 & 0 \\ 0 & n_y n_x\left(S_3 - |q_n|\right) & |q_n|\left(n_x^2 + n_z^2\right) + S_3 n_y^2 & n_y n_z\left(S_3 - |q_n|\right) & n_y S_2 & 0 \\ 0 & n_z n_x\left(S_3 - |q_n|\right) & n_z n_y\left(S_3 - |q_n|\right) & |q_n|\left(n_x^2 + n_y^2\right) + S_3 n_z^2 & n_z S_2 & 0 \\ 0 & n_x c^2 S_1 & n_y c^2 S_1 & n_z c^2 S_1 & S_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & |q_n| \end{pmatrix} \tag{A.8}$$

Here, a set of auxiliary variables is defined by:

$$\omega = \frac{\delta}{1 + \delta^2} \tag{A.9}$$

$$S_1 = -\frac{\rho}{2c\tau}\left(\left(c + \omega M_{\mathrm{r}} \lambda_2\right)\left|\lambda_1\right| - \left(c + \omega M_{\mathrm{r}} \lambda_1\right)\left|\lambda_2\right|\right) \tag{A.10}$$

$$S_2 = -\frac{1}{2\rho c \tau M_\mathrm{r}^2} \left((c - \omega M_\mathrm{r} \lambda_2) |\lambda_1| - (c - \omega M_\mathrm{r} \lambda_1) |\lambda_2| \right) \tag{A.11}$$

$$S_3 = \frac{|\lambda_1| + |\lambda_2|}{2\left(1 + \delta^2\right)} + \frac{\omega \delta q_n \left(|\lambda_2| - |\lambda_1| \right)}{2\tau} \tag{A.12}$$

These variables are very sensitive to round-off errors and have thus been strongly optimized. With these definitions, the upwinding matrix is evaluated numerically in primitive variables. Moreover, the transformation matrices to conservative variables are also evaluated numerically at the Roe-averaged state. The upwinding matrix in conservative variables is then computed by two matrix-matrix multiplications, i.e.

$$D_\mathbf{U} = \left(\frac{\partial \mathbf{U}}{\partial \mathbf{V}} \cdot D_\mathbf{V} \cdot \frac{\partial \mathbf{V}}{\partial \mathbf{U}} \right)_\mathrm{roe} . \tag{A.13}$$

Finally, the numerical flux of the Roe-Lowmach scheme is computed by

$$\mathbf{F}_{i+1/2} = \frac{1}{2} \left(\mathbf{F}\left(\mathbf{U}^L_{i+1/2} \right) + \mathbf{F}\left(\mathbf{U}^R_{i+1/2} \right) - D_\mathbf{U} \left(\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2} \right) \right) . \tag{A.14}$$

B Butcher tableaus for ESDIRK schemes

ESDIRK23 (Hosea & Shampine 1996):

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 2-\sqrt{2} & \frac{2-\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} & 0 \\ 1 & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{2-\sqrt{2}}{2} \\ \hline \beta & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{2-\sqrt{2}}{2} \\ \hline \tilde{\beta} & \frac{4-\sqrt{2}}{12} & \frac{4+3\sqrt{2}}{12} & \frac{2-\sqrt{2}}{6} \end{array} \tag{B.1}$$

ESDIRK34 (Kennedy & Carpenter 2001):

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1767732205903}{2027836641118} & \frac{1767732205903}{4055673282236} & \frac{1767732205903}{4055673282236} & 0 & 0 \\ \frac{3}{5} & \frac{2746238789719}{10658868560708} & \frac{-640167445237}{6845629431997} & \frac{1767732205903}{4055673282236} & 0 \\ 1 & \frac{1471266399579}{7840856788654} & \frac{-4482444167858}{7529755066697} & \frac{11266239266428}{11593286722821} & \frac{1767732205903}{4055673282236} \\ \hline \beta & \frac{1471266399579}{7840856788654} & \frac{-4482444167858}{7529755066697} & \frac{11266239266428}{11593286722821} & \frac{1767732205903}{4055673282236} \\ \hline \tilde{\beta} & \frac{2756255671327}{12835298489170} & \frac{-10771552573575}{22201958757719} & \frac{9247589265047}{10645013368117} & \frac{2193209047091}{5459859503100} \end{array} \tag{B.2}$$

ESDIRK46 (Kennedy & Carpenter 2001):

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{83}{250} & \frac{8611}{62500} & \frac{-1743}{31250} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{31}{50} & \frac{5012029}{34652500} & \frac{-654441}{2922500} & \frac{174375}{388108} & \frac{1}{4} & 0 & 0 \\ \frac{17}{20} & \frac{15267082809}{155376265600} & \frac{-71443401}{120774400} & \frac{730878875}{902184768} & \frac{2285395}{8070912} & \frac{1}{4} & 0 \\ 1 & \frac{82889}{524892} & 0 & \frac{15625}{83664} & \frac{69875}{102672} & \frac{-2260}{8211} & \frac{1}{4} \\ \hline \beta & \frac{82889}{524892} & 0 & \frac{15625}{83664} & \frac{69875}{102672} & \frac{-2260}{8211} & \frac{1}{4} \\ \hline \tilde{\beta} & \frac{4586570599}{29645900160} & 0 & \frac{178811875}{945068544} & \frac{814220225}{1159782912} & \frac{-3700637}{11593932} & \frac{61727}{225920} \end{array} \tag{B.3}$$

ESDIRK58 (Kennedy & Carpenter 2001):

$$
\begin{array}{c|ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{41}{100} & \frac{41}{200} & \frac{41}{200} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{2935347310677}{11292855782101} & \frac{41}{400} & \frac{-567603406766}{11931857230679} & \frac{41}{200} & 0 & 0 & 0 & 0 & 0 \\
\frac{1426016391358}{7196633302097} & \frac{683785636431}{9252920307686} & 0 & \frac{-110385047103}{1367015193373} & \frac{41}{200} & 0 & 0 & 0 & 0 \\
\frac{92}{100} & \frac{3016520224154}{10081342136671} & 0 & \frac{30586259806659}{12414158314087} & \frac{-22760509404356}{11113319521817} & \frac{41}{200} & 0 & 0 & 0 \\
\frac{24}{100} & \frac{218866479029}{1489978393911} & 0 & \frac{638256894668}{5436446318841} & \frac{-1179710474555}{5321154724896} & \frac{-60928119172}{8023461067671} & \frac{41}{200} & 0 & 0 \\
\frac{3}{5} & \frac{1020004230633}{5715676835656} & 0 & \frac{25762820946817}{25263940353407} & \frac{-2161375909145}{9755907335909} & \frac{-211217309593}{5846859502534} & \frac{-4269925059573}{7827059040749} & \frac{41}{200} & 0 \\
1 & \frac{-872700587467}{9133579230613} & 0 & 0 & \frac{22348218063261}{9555858737531} & \frac{-1143369518992}{8141816002931} & \frac{-39379526789629}{19018526304540} & \frac{32727382324388}{42900044865799} & \frac{41}{200} \\
\hline
\beta & \frac{-872700587467}{9133579230613} & 0 & 0 & \frac{22348218063261}{9555858737531} & \frac{-1143369518992}{8141816002931} & \frac{-39379526789629}{19018526304540} & \frac{32727382324388}{42900044865799} & \frac{41}{200} \\
\hline
\tilde{\beta} & \frac{-975461918565}{9796059967033} & 0 & 0 & \frac{78070527104295}{32432590147079} & \frac{-548382580838}{3424219808633} & \frac{-33438840321285}{15594753105479} & \frac{3629800801594}{4656183773603} & \frac{4035322873751}{18575991585200}
\end{array}
\tag{B.4}
$$

C Transformations between variables

In the following, the Jacobian matrices of the transformation from conservative to primitive and symmetrizing variables are briefly derived for non-dimensional quantities and a general equation of state. To this end, the non-dimensional vector of conservative variables is written in the following form:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho X \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}_3 \\ \mathbf{U}_4 \\ \mathbf{U}_5 \\ \mathbf{U}_6 \end{pmatrix} \tag{C.1}$$

Assuming that the equation of state is given by $p = p(\rho, \epsilon, X)$ (cf. Section 2.2.1), the vector of primitive variables can be expressed through the conservative variables by

$$\mathbf{V} = \begin{pmatrix} \rho \\ u \\ v \\ w \\ p \\ X \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2/\mathbf{U}_1 \\ \mathbf{U}_3/\mathbf{U}_1 \\ \mathbf{U}_4/\mathbf{U}_1 \\ p\left(\mathbf{U}_1, \mathbf{U}_5 - \frac{1}{2\mathbf{U}_1} M_\mathrm{r}^2 \left(\mathbf{U}_2^2 + \mathbf{U}_3^2 + \mathbf{U}_4^2\right), \mathbf{U}_6/\mathbf{U}_1\right) \\ \mathbf{U}_6/\mathbf{U}_1 \end{pmatrix}. \tag{C.2}$$

The Jacobian matrix of the transformation from conservative to primitive variables can then directly be calculated as follows:

$$\frac{\partial \mathbf{V}}{\partial \mathbf{U}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ p_{,\rho} + \frac{1}{2} M_\mathrm{r}^2 p_{,\epsilon} \left(u^2 + v^2 + w^2\right) - \frac{p_{,X} X}{\rho} & -p_{,\epsilon} M_\mathrm{r}^2 u & -p_{,\epsilon} M_\mathrm{r}^2 v & -p_{,\epsilon} M_\mathrm{r}^2 w & p_{,\epsilon} & \frac{p_{,X}}{\rho} \\ -\frac{X}{\rho} & 0 & 0 & 0 & 0 & \frac{1}{\rho} \end{pmatrix} \tag{C.3}$$

The derivatives of the equation of state are abbreviated by

$$p_{,\rho} = \left.\frac{\partial p}{\partial \rho}\right|_{\epsilon, X} \qquad p_{,\epsilon} = \left.\frac{\partial p}{\partial \epsilon}\right|_{\rho, X} \qquad p_{,X} = \left.\frac{\partial p}{\partial X}\right|_{\rho, \epsilon}. \tag{C.4}$$

The Jacobian matrix of the transformation from primitive to conservative variables can be calculated by inverting the above matrix:

$$\frac{\partial \mathbf{U}}{\partial \mathbf{V}} = \left(\frac{\partial \mathbf{V}}{\partial \mathbf{U}}\right)^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 & 0 \\ w & 0 & 0 & \rho & 0 & 0 \\ \frac{1}{2} M_\mathrm{r}^2 \left(u^2 + v^2 + w^2\right) - \frac{p_{,\rho}}{p_{,\epsilon}} & \rho u M_\mathrm{r}^2 & \rho v M_\mathrm{r}^2 & \rho w M_\mathrm{r}^2 & \frac{1}{p_{,\epsilon}} & -\frac{p_{,X}}{p_{,\epsilon}} \\ X & 0 & 0 & 0 & 0 & \rho \end{pmatrix} \tag{C.5}$$

Deriving the Jacobians in this way has the advantage, that the derivative of the equation of state do not need to be inverted. The derivatives given in Equation C.4 are computed anyway.

The so-called *symmetizing variables* $\mathbf{W}$ are actually only defined by their Jacobian matrix (see e.g. Turkel 1999):

$$\frac{\partial \mathbf{W}}{\partial \mathbf{V}} = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{\rho c} & 0 \\ 0 & M_\mathrm{r} & 0 & 0 & 0 & 0 \\ 0 & 0 & M_\mathrm{r} & 0 & 0 & 0 \\ 0 & 0 & 0 & M_\mathrm{r} & 0 & 0 \\ -\frac{c}{\rho} & 0 & 0 & 0 & \frac{1}{\rho c} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{C.6}$$

The variable c denotes speed of sound as defined in Equation 2.2.6. The transformation to conservative variables can be obtained with the chain rule of differentiation:

$$\frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \frac{\partial \mathbf{W}}{\partial \mathbf{V}} \cdot \frac{\partial \mathbf{V}}{\partial \mathbf{U}} \tag{C.7}$$

Jacobian matrices of the inverse transformations can then again be calculated as the inverse of the corresponding matrix. It should be noted that the symmetrizing variables are actually called that way because they may be used to diagonalize the Euler equations locally.

D Details of the semi-analytical stability analysis

The semi-analytical stability analysis for shear flows presented in Section 7.3.1 yields the following equations for the coefficients of the normal mode:

$$\begin{array}{rcl} i\,(ku_{\mathrm{eq}}-\omega)\;\tilde{\rho} & + & \tilde{v}\;\rho'_{\mathrm{eq}}+\rho_{\mathrm{eq}}\,(ik\tilde{u}+\tilde{v}')=0 \\ i\,(ku_{\mathrm{eq}}-\omega)\;\tilde{p} & + & \tilde{v}\;p'_{\mathrm{eq}}+p_{\mathrm{eq}}\gamma\,(ik\tilde{u}+\tilde{v}')=0 \\ i\rho_{\mathrm{eq}}\,(ku_{\mathrm{eq}}-\omega)\;\tilde{u} & + & \frac{1}{M_{\mathrm{r}}^2}\;ik\tilde{p}+\rho_{\mathrm{eq}}\tilde{v}u'_{\mathrm{eq}}=0 \\ i\rho_{\mathrm{eq}}\,(ku_{\mathrm{eq}}-\omega)\;\tilde{v} & + & \frac{1}{M_{\mathrm{r}}^2}\;\tilde{p}'+\frac{1}{F_{\mathrm{r}}^2}\tilde{\rho}=0 \end{array} \tag{D.1}$$

In order to solve these equations, various algebraic transformations have to be applied which are briefly outlined in the following. The actual computations were performed with the symbolic computer algebra software Mathematica. The method closely follows the work by Blumen (1970) where further details can be found.

In a first step, the third equation is solved for $\tilde{u}$. The result is inserted into to first and second equation. Next, the first equation is solved for $\tilde{\rho}$ and inserted into the fourth equation. Then, the fourth equation is solved for $\tilde{p}$ whose derivative with respect to y is computed analytically. These results are inserted into the third equation which can then be casted into a equation for the vertical velocity disturbance $\tilde{v}$:

$$\tilde{v}''(y)+A(y)\;\tilde{v}'(y)+B(y)\;\tilde{v}(y)=0 \tag{D.2}$$

With this method, the original system of equations has been reduced to a single, second-order differential equation at the expense that the coefficients $A(y)$ and $B(y)$ take a rather complicated form.

Equation D.2 has to be solved under the constraint that the velocity disturbances vanish at infinity, i.e. $\lim_{y\to\pm\infty}\tilde{v}(y)=0$, yielding a boundary-value problem. It should be noted that the coefficients contain the complex frequency ω which is an unknown parameter for this problem. The overall challenge is to find such a frequency that the boundary-value problem has a unique solution. As the detailed form of $\tilde{v}(y)$ is thus not of importance here, the differential equation can be further simplified by introducing a new depended variable by

$$\tilde{w}(y)=\frac{\tilde{v}'(y)}{\tilde{v}(y)}. \tag{D.3}$$

The boundary-value problem then becomes a first-order differential equation again:

$$\tilde{w}'(y)+\tilde{w}^2(y)+A(y)\tilde{w}(y)+B(y)=0 \tag{D.4}$$

The corresponding boundary conditions for the new variable $\tilde{w}$ can easily be obtained by assuming that the coefficients of Equation D.2 tend towards constant values at infinity:

$$\tilde{v}''_{\infty}(y)+A_{\infty}\tilde{v}'_{\infty}(y)+B_{\infty}\tilde{v}_{\infty}(y)=0 \tag{D.5}$$

This linear differential equation can be solved analytically leading to solutions of the form

$$\tilde{v}_{\pm\infty}(y)\propto\exp\left\{\frac{1}{2}\left(-A_{\pm\infty}\pm\sqrt{\left(A_{\pm\infty}^2-4B_{\pm\infty}\right)}\right)y\right\}. \tag{D.6}$$

Transforming these solutions to the new dependent variable $\tilde{w}$ results in the final form of the boundary conditions:

$$\lim_{y\to\pm\infty} \tilde{w}(y) = \frac{1}{2}\left(-A_{\pm\infty} \pm \sqrt{\left(A_{\pm\infty}^2 - 4B_{\pm\infty}\right)}\right) \tag{D.7}$$

$$\mathrm{Re}\left(\lim_{y\to\pm\infty} \tilde{w}(y)\right) \lesseqgtr 0 \tag{D.8}$$

As the coefficients of Equation D.4 are rather complicated, the problem can only be solved numerically. However, the integration range from minus infinity to plus infinity is computationally problematic. Therefore, a new independent variable is defined by

$$z = \mathrm{arctanh}\,(y)\,, \tag{D.9}$$

which ranges only from -1 to 1. The corresponding boundary-value problem then reads

$$\tilde{w}'(z) = -\frac{\tilde{w}^2(z) + A(z)\tilde{w}(z) + B(z)}{1 - z^2}. \tag{D.10}$$

Inserting the equilibrium states of the shear flow problem defined in Section 7.2 and the above transformations into the coefficients $A(y)$ and $B(y)$, the differential equation can be solved numerically. As the problem contains the unknown frequency ω, an initial (complex) value is guessed or taken from the results with slightly different parameters. Equation D.10 is then integrated numerically with Mathematica's built-in methods from $z = -1$ to $z = 0$ and from $z = +1$ to $z = 0$. The difference of both results at $z = 0$ is then taken as an input parameter for a root-finding method which takes ω as independent variable. Convergence is assumed if the relative difference of both integration steps drops below 10^{-6}. This method is automated in a way that it produces the dispersion relation $\omega(k)$ for a given set of model parameters.

Bibliography

Arnett, W. D. & Meakin, C. 2011, ApJ, 733, 78

Beck, P. G., Montalban, J., Kallinger, T., et al. 2012, Nature, 481, 55

Benielli, D. & Sommeria, J. 1998, Journal of Fluid Mechanics, 374, 117

Bian, X. 2008, Master's thesis, Technical University Munich

Birken, P. & Meister, A. 2005, BIT Numerical Mathematics, 45, 463

Blumen, W. 1970, Journal of Fluid Mechanics, 40, 769

Böhm-Vitense, E. 1958, ZAp, 46, 108

Bouchut, F. 2004, Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources, Frontiers in Mathematics (Birkäuser)

Bouruet-Aubertot, P., Sommeria, J., & Staquet, C. 1995, Journal of Fluid Mechanics, 285, 265

Brown, B. P., Vasil, G. M., & Zweibel, E. G. 2012, ArXiv e-prints

Browning, M. K., Brun, A. S., & Toomre, J. 2004, The Astrophysical Journal, 601, 512

Brüggen, M. & Hillebrandt, W. 2001, MNRAS, 320, 73

Buffard, T., Gallout, T., & Hrard, J.-M. 2000, Computers & Fluids, 29, 813

Canuto, V. M. 1996, ApJ, 467, 385

Canuto, V. M. 2002, A&A, 384, 1119

Chan, T. F., Gallopoulos, E., Simoncini, V., Szeto, T., & Tong, C. H. 1994, SIAM Journal on Scientific Computing, 15, 338

Chandrasekhar, S. 1961, Hydrodynamic and hydromagnetic stability (International Series of Monographs on Physics, Oxford: Clarendon, 1961)

Charbonnel, C. & Talon, S. 2005, Science, 309, 2189

Cinnella, P. 2006, Computers & Fluids, 35, 1264

Dellacherie, S. 2010, Journal of Computational Physics, 229, 978

Dintrans, B., Brandenburg, A., Nordlund, Å., & Stein, R. F. 2005, A&A, 438, 365

Drazin, P. G. 1977, Royal Society of London Proceedings Series A, 356, 411

Drikakis, D. & Rider, W. 2005, High-Resolution Methods For Incompressible And Low-Speed Flows, Computational Fluid and Solid Mechanics (Springer)

Flores, J., Batiste, D. M., Holst, T. L., & Kwak, D. 1984, AIAA Journal, 22, 1027

Freund, R. W. 1993, SIAM Journal on Scientific Computing, 14, 470

Fricke, K. 1968, ZAp, 68, 317

Fuchs, F., McMurry, A., Mishra, S., Risebro, N., & Waagan, K. 2010, Journal of Computational Physics, 229, 4033

Garcia Lopez, R. J. & Spruit, H. C. 1991, ApJ, 377, 268

Ghidaglia, J. & Pascal. 2005, European Journal of Mechanics B Fluids, 24, 1

Glatzmaier, G. 1984, Journal of Computational Physics, 55, 461

Goldreich, P. & Schubert, G. 1967, ApJ, 150, 571

Grinstein, F., Margolin, L., & Rider, W. 2007, Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics (Cambridge University Press)

Guillard, H. & Murrone, A. 2004, Computers & Fluids, 33, 655

Guillard, H. & Viozat, C. 1999, Computers & Fluids, 28, 63

Hairer, E. & Wanner, G. 1996, Solving Ordinary Differential Equations II (Berlin: Springer Verlag)

Hirsch, C. 1990, Numerical computation of internal and external flows. Vol. 2 - Computational methods for inviscid and viscous flows (Chichester, England and New York: John Wiley & Sons)

Hirsch, C. & Hirsch, C. 1991, Numerical Computation of Internal and External Flows: Fundamentals of numerical discretization, Wiley series in numerical methods in engineering (Wiley)

Hirschi, R., Maeder, A., & Meynet, G. 2004, in IAU Symposium, Vol. 215, Stellar Rotation, ed. A. Maeder & P. Eenens, 510

Hosea, M. & Shampine, L. 1996, Applied Numerical Mathematics, 20, 21 , method of Lines for Time-Dependent Problems

Howard, L. N. 1961, Journal of Fluid Mechanics, 10, 509

Iglesias, C. A. & Rogers, F. J. 1996, ApJ, 464, 943

Kallischko, A. 2008, Dissertation, Technische Universität München, München

Kennedy, C. A. & Carpenter, M. H. 2001, Additive Runge-Kutta Schemes for Convection-Diffusion-Reaction Equations, Tech. rep., NASA Technical Memorandum

Keppens, R., Toth, G., Botchev, M., & Van der Ploeg, A. 1999, International Journal for Numerical Methods in Fluids, 30, 335

Kifonidis, K. & Müller, E. 2012, A&A, 544, A47

Kippenhahn, R. & Weigert, A. 1994, Stellar Structure and Evolution

Klostermeyer, J. 1982, Journal of Fluid Mechanics, 119, 367

Klostermeyer, J. 1991, Geophysical and Astrophysical Fluid Dynamics, 61, 1

Kolmogorov, A. 1941, Akademiia Nauk SSSR Doklady, 30, 301

Kupka, F. 1996, in Astronomical Society of the Pacific Conference Series, Vol. 108, M.A.S.S., Model Atmospheres and Spectrum Synthesis, ed. S. J. Adelman, F. Kupka, & W. W. Weiss, 73

Landau, L. D. & Lifschitz, E. M. 1991, Lehrbuch der theoretischen Physik, Vol. 6, Hydrodynamik (Berlin: Akademie Verlag)

Li, X. S. & Demmel, J. W. 2003, ACM Trans. Mathematical Software, 29, 110

Lignières, F., Califano, F., & Mangeney, A. 1999, A&A, 349, 1027

Liou, M. 1996, Journal of Computational Physics, 129, 364

Liou, M.-S. 2006, Journal of Computational Physics, 214, 137

Liska, R. & Wendroff, B. 2003, SIAM Journal on Scientific Computing, 25, 995

Maeder, A. 2009, Physics, formation and evolution of rotating stars, Astronomy and astrophysics library (Springer)

Maeder, A. & Meynet, G. 2000, ARA&A, 38, 143

Masella, J.-M., Faille, I., & Gallouet, T. 1999, International Journal of Computational Fluid Dynamics, 12, 133

Mathis, S., Palacios, A., & Zahn, J.-P. 2004, A&A, 425, 243

Meakin, C. A. & Arnett, D. 2007, ApJ, 667, 448

Meister, A. 2008, Numerik linearer Gleichungssysteme, 3rd edn. (Wiesbaden: vieweg)

Miczek, F. 2008, diploma thesis, Technische Universität München

Miesch, M. S. 2005, Living Reviews in Solar Physics, 2, 1

Miesch, M. S., Elliott, J. R., Toomre, J., et al. 2000, The Astrophysical Journal, 532, 593

Mocák, M., Müller, E., Weiss, A., & Kifonidis, K. 2009, A&A, 501, 659

Mohr, M. & Wienands, R. 2004, Computing and Visualization in Science, 7, 129

Mottura, L., Vigevano, L., & Zaccanti, M. 1997, Journal of Computational Physics, 138, 354

Peltier, W. R. 2003, Annual Review of Fluid Mechanics, 35, 135

Pissanetzky, S. 1984, Sparse matrix technology (Academic Press)

Pope, S. B. 2000, Turbulent Flows

Riddaway, R. & Hortal, M. 2001, Numerical methods, `http://www.ecmwf.int/newsevents/training/rcourse_notes/NUMERICAL_METHODS/NUMERICAL_METHODS/Numerical_methods.html`, (accessed 2012/07/24)

Rieper, F. 2011, Journal of Computational Physics, 230, 5263

Roe, P. L. 1981, Journal of Computational Physics, 43, 357

Rogers, T. M. & Glatzmaier, G. A. 2005, MNRAS, 364, 1135

Saad, Y. & Schultz, M. H. 1986, SIAM Journal on Scientific and Statistical Computing, 7, 856

Salaris, M. & Cassisi, S. 2005, Evolution of Stars and Stellar Populations

Schatzman, E. 1993, A&A, 279, 431

Schenk, O. & Gärtner, K. 2004, Future Generation Computer Systems, 20, 475

Schmidt, W. 2004, Dissertation, Technische Universität München, München

Schmidt, W., Hillebrandt, W., & Niemeyer, J. C. 2004, ArXiv Astrophysics e-prints

Schochet, S. 1994, Journal of differential equations, 114, 476

Shu, C.-W. & Osher, S. 1988, Journal of Computational Physics, 77, 439

Sleijpen, G. & Fokkema, D. 1993, Electronic Transactions on Numerical Analysis, 1, 11

Sofia, S., Heaps, W., & Twigg, L. W. 1994, ApJ, 427, 1048

Sonneveld, P. & van Gijzen, M. B. 2008, SIAM Journal on Scientific Computing, 31, 1035

Spruit, H. C., Nordlund, A., & Title, A. M. 1990, ARA&A, 28, 263

Staquet, C. & Sommeria, J. 2002, Annual Review of Fluid Mechanics, 34, 559

Sutherland, B. 2010, Internal Gravity Waves (Cambridge University Press)

Talon, S. 2008, in EAS Publications Series, Vol. 32, EAS Publications Series, ed. C. Charbonnel & J.-P. Zahn, 81–130

Talon, S., Vincent, A., Michaud, G., & Richer, J. 2003, Journal of Computational Physics, 184, 244

Thompson, J., Warsi, Z., & Mastin, C. 1985, Numerical grid generation: foundations and applications (North-Holland)

Thompson, J. F., Warsi, Z. U. A., & Mastin, C. W. 1982, Journal of Computational Physics, 47, 1

Thompson, M. J., Christensen-Dalsgaard, J., Miesch, M. S., & Toomre, J. 2003, ARA&A, 41, 599

Thornber, B., Mosedale, A., Drikakis, D., Youngs, D., & Williams, R. 2008, Journal of Computational Physics, 227, 4873

Toro, E. 2009, Riemann solvers and numerical methods for fluid dynamics: a practical introduction (Springer)

Toselli, A. & Widlund, O. 2005, Domain decomposition methods–algorithms and theory, Springer series in computational mathematics (Springer)

Trottenberg, U., Oosterlee, C., & Schueller, A. 2001, Multigrid (London: Academic Press)

Turkel, E. 1999, Annual Review of Fluid Mechanics, 31, 385

Vinokur, M. & Kordulla, W. 1983, AIAA Journal, 21, 917

Vinokur, M. & Montagne, J.-L. 1990, Journal of Computational Physics, 89, 276

Vitense, E. 1953, ZAp, 32, 135

Volpe, G. 1993, AIAA Journal, 31, 49

Weiss, A. 2009, private communication

Weiss, A., Hillebrandt, W., Thomas, H.-C., & Ritter, H. 2004, Cox and Giuli's Principles of Stellar Structure

Weiss, A. & Schlattl, H. 2008, Ap&SS, 316, 99

Weiss, J. M. & Smith, W. A. 1995, AIAA Journal, 33, 2050

Young, P. A., Knierman, K. A., Rigby, J. R., & Arnett, D. 2003, ApJ, 595, 1114

Acknowledgments

At this point, I would like to thank my supervisor Fritz Röpke for giving me the opportunity to work in such an interesting field of scientific research and for his constant encouragement and guidance.

I also thank all my colleagues at MPA and especially the people from the type Ia supernova group. They created a very pleasant and inspiring working atmosphere. The countless scientific (and especially non-scientific) discussions really helped during the work for this thesis. I want to thank all people from room 009 who shared an office with me during the last years. In particular, I want to mention Rüdiger Pakmor, Stephan Hachinger, Philipp Edelmann, Franco Ciaraldi-Schoolmann, Michael Klauser, Janina von Groote, Ulrich Noebauer and Matthias Herzog.

Finally, I want to thank my girlfriend Binne and her tomcat Schnitti wholeheartedly for their continuous encouragement, support and patience for my work (and for trying to improve this thesis by lumbering on the keyboard). They made life worth living during the last years.