

Modelling of solar granulation - Implementation and comparison of numerical schemes

Dissertation
aus Mathematik

zur Erlangung des akademischen Grades
„Doktor der Naturwissenschaften“
an der Fakultät für Mathematik
der Universität Wien

Eingereicht von:
Christof Obertscheider
Michaelerstraße 4/22
1180 Wien

Betreuer: Univ.Prof. Dr. Herbert J. Muthsam

Wien, im April 2007

Abstract

The mathematical part covers the implementation and comparison of different numerical schemes to solve the equations of radiation hydrodynamics. One and the same two-dimensional model of solar granulation is simulated with different high resolution numerical methods (essentially non-oscillatory and convex non-oscillatory) and non-grey short characteristic radiative transfer. Furthermore the influence of artificial diffusivities is studied.

The second part covers two astrophysical applications. A two-dimensional non-grey high resolution simulation of a downflowing plume in the sun and a grey three-dimensional simulation of the quiet sun are presented.

Contents

Introduction	3
1 The equations of radiation hydrodynamics (RHD)	5
1.1 The hydrodynamical equations	5
1.2 The equation of radiative transfer	6
1.3 The equation of state	8
2 The ANTARES code	9
2.1 The structure of the ANTARES code	9
2.2 Modelling a box in the sun	10
2.2.1 Initial condition	10
2.2.2 Numerical grid	11
2.2.3 Domain decomposition for parallelization	12
2.2.4 Hydrodynamical boundary conditions	13
2.2.5 Numerical problems	14
2.2.6 Equation of state	14
3 Implemented numerical schemes	16
3.1 Numerical schemes for conservation laws	16
3.1.1 Essentially non-oscillatory (ENO) and Weighted essentially non-oscillatory (WENO) schemes	17
3.1.2 Essentially non-oscillatory schemes with Marquina's flux splitting and entropy fix	23
3.1.3 Convex ENO (CNO) schemes	25
3.1.4 Discussion	27
3.2 Discretization of the viscous fluxes	31
3.2.1 Viscous hydrodynamical fluxes	31
3.2.2 Artificial diffusivities	31
3.3 Interpolation tool	32
3.4 Numerical radiative transfer	33
3.4.1 Numerical Scheme	33

3.4.2	Frequency dependent radiative transfer	40
3.4.3	2D test problem	41
3.5	Temporal discretization	45
4	Comparison of the numerical schemes	46
4.1	Results	47
4.1.1	Up- and downflows at 600km	47
4.1.2	Comparison of horizontally averaged quantities	49
4.2	Discussion	51
4.2.1	Effects using different numerical schemes for conservation laws	51
4.2.2	Effects using artificial diffusivities	52
4.2.3	Different spatial resolution	53
4.2.4	Conclusion	53
5	A two-dimensional high resolution study of solar surface flows	55
6	A grey three-dimensional simulation of the quiet sun	65
	Bibliography	68
	Danksagung	69
	Lebenslauf	70
	Sacherschließung	71

Introduction

'Under good conditions of atmospheric seeing, with a telescope of at least intermediate size ($d \geq 20\text{cm}$, say), one sees a cellular pattern which covers the entire solar surface, except in sunspots. Bright isolated elements, the granules, appear on a dark background of multiply connected intergranular lanes. [...] The bright granules are upwards-moving, hot parcels of gas, while the dark intergranular lanes represent cooler, downwards-moving, material.' (Stix [34])

Solar granulation is the visible manifestation of the convection in the upper regions of the sun. Generally, convective motions in the sun 'together with rotation are believed to drive the dynamo that reverses the surface magnetic field with an eleven year timescale. Convective motions generate the p -mode oscillations that are used to probe the solar interior. They also produce magneto-hydrodynamic waves that supply energy to the solar chromosphere, and they move the footpoints of the coronal magnetic flux tubes, which leads to magnetic reconnection that heats the corona.' (Stein and Nordlund [32])

'Granules evolve because their diverging flow pushes against neighboring granules and because overlying cool fluid splits them apart. Above a granule, high pressure, due to higher temperature and larger scale height, diverts the flow horizontally, producing a fountain-like topology. The horizontal flow encounters surrounding expanding granules against which it pushes. Granules with the largest pressure driving their expansion win the competition and grow, while others with less pressure excess have their growth limited and may even shrink.' (Stein and Nordlund [32])

The study of solar granulation is a very active field of research. Thus the main goal of this PhD project was to extend an existing hydrodynamic code for realistic simulations of the granulation in the quiet sun, i.e. regions without magnetic activity.

The mathematical part covers the implementation and comparison of different numerical schemes to solve the equations of radiation hydrodynamics. In the astrophysical part results of two realistic simulations of the sun (a two-dimensional high resolution study of a downflowing plume and a three-dimensional grey simulation) are presented.

In chapter 1 the equations of radiation hydrodynamics are presented. Chapter 2 describes the ANTARES (Advanced Numerical Tool for Astrophysical RESearch) code and in chapter 3 the used numerical schemes for spatial and temporal discretization are presented.

In chapter 4 the differences between the numerical methods are analyzed. For the numerical solution of the conservation laws are used:

- a fourth order convex non-oscillatory scheme ([16]),
- a fifth order weighted essentially non-oscillatory scheme (Harten et al. [9], Shu and Osher [29]) and
- a third order essentially non-oscillatory scheme with Marquina's flux splitting and entropy-fix (Fedkiw et al. [8]).

The 'real' viscosity in the sun is too small for stable numerical simulations. Either the numerical viscosity of the methods or the explicit addition of stabilizing viscosities must ensure stability. The viscous fluxes either can be discretized by a fourth order scheme or can be replaced by artificial diffusivities (Stein and Nordlund [32], Caunt and Korpi [5]) which remove short-wavelength noise without damping longer wavelengths and diffuse strong discontinuities in order to stabilize the numerical code.

All simulations different in numerics are performed with the same code starting from the same initial state.

Modelling of solar granulation describes reality in many details (Stein and Nordlund [33]). Nevertheless, important issues remain unchecked, particularly processes occurring on small scales which neither observations nor previous models resolve. Small scale features are important because they may influence the global outcome and the overall physical structure. Chapter 5 presents results of a two-dimensional high resolution simulation of a downflowing plume.

Chapter 6 briefly shows some intermediate results of a grey three-dimensional simulation in order to show that the ANTARES code also performs well for the 3D case. This three-dimensional simulation contains grey radiative transfer since this reduces the computational time.

Chapter 1

The equations of radiation hydrodynamics (RHD)

1.1 The hydrodynamical equations

The equations governing the dynamics of solar granulation are the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{u}] = 0, \quad (1.1)$$

the momentum equation

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot [\rho \mathbf{u} \mathbf{u} + p \underline{I}] = \rho \mathbf{g} + \nabla \cdot \underline{\tau}, \quad (1.2)$$

and the total energy equation

$$\frac{\partial e}{\partial t} + \nabla \cdot [\mathbf{u}(e + p)] = \rho(\mathbf{g} \cdot \mathbf{u}) + \nabla \cdot (\mathbf{u} \cdot \underline{\tau}) + Q_{\text{rad}}. \quad (1.3)$$

All physical quantities are functions of space $\mathbf{x} = (x, y, z)^T$ and time t , i.e. $\rho = \rho(\mathbf{x}, t)$. The x -direction points in the vertical direction. y - and z -direction are the horizontal coordinates.

The relations between the thermodynamical quantities are described by an equation of state.

ρ	gas density
\mathbf{u}	flow velocity
$\rho \mathbf{u}$	momentum density
$\mathbf{u} \mathbf{u}$	dyadic product
p	gas pressure
\underline{I}	identity matrix
$\mathbf{g} = (g, 0, 0)^T$	gravitational acceleration
$\underline{\tau}$	viscous stress tensor
	$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} (\nabla \cdot \mathbf{u}) \right) \quad i, j = 1, 2, 3$
μ	dynamic viscosity
e	'total' energy density per volume, the sum of internal and kinetic energy densities $e = e_{\text{int}} + e_{\text{kin}}$ (without potential energy)
Q_{rad}	radiative source term
T	temperature

Table 1.1: Meaning of the variables.

length	cm
time	s
solid angle	sr
gas density	g/cm ³
flow velocity	cm/s
energy density	erg/cm ³
temperature	K
gas pressure	dyn/cm ²
specific heat capacity	erg/g/K
dynamic viscosity	g/cm/s
thermal conductivity	erg/cm/K/s
frequency	hz=1/s
opacity	1/cm
intensity, radiative source function	erg/cm ² /s/hz/sr
mean intensity	erg/cm ² /s/hz
radiative heating rate	erg/cm ³ /s
(radiative) energy flux density	erg/cm ² /s

Table 1.2: Unities of used physical quantities. The unities of all quantities are in the cgs-system if nothing else is specified.

$g = 274\text{m/s}^2$	gravity acceleration at the solar surface
$c = 2.998 \cdot 10^{10}\text{cm/s}$	speed of light in vacuum
$h = 6.626 \cdot 10^{-27}\text{erg s}$	Planck's constant
$k = 1.381 \cdot 10^{-16}\text{erg/K}$	Boltzmann's constant
$\sigma = 5.670 \cdot 10^{-5}\text{erg/s/cm}^2/\text{K}^4$	Stefan-Boltzmann constant
$\pi = 3.1415927$	

Table 1.3: Values and unities of used constants.

1.2 The equation of radiative transfer

Any realistic simulation of solar surface (photosphere and convection zone) flows must include the energy exchange between gas and radiation. This energy exchange is described by the radiative heating rate Q_{rad} which is an additive term in the energy equation.

Starting point for determining Q_{rad} is the time- and frequency-dependent radiative transfer equation

$$\left(\frac{1}{c} \frac{\partial}{\partial t} + \mathbf{r} \cdot \nabla\right) I_{\nu} = \rho \chi_{\nu} (S_{\nu} - I_{\nu}) \quad (1.4)$$

where S_{ν} is the source function and χ_{ν} is the opacity of the material. The specific Intensity I_{ν} at position \mathbf{x} travelling in direction \mathbf{r} with frequency ν at time t is defined (Mihalas [21]) such that the amount of energy transported by radiation of frequencies $(\nu, \nu + d\nu)$ across an element of area dS into a solid angle $d\omega$ in a time interval dt is

$$dE_{\text{rad}} = I(\mathbf{x}, \mathbf{r}, \nu, t) dS \cos \theta d\omega d\nu dt \quad (1.5)$$

where θ is the angle between the direction of the beam and the normal to the surface (i.e., $dS \cos \theta = \mathbf{n} \cdot d\mathbf{S}$) (see figure 1.1).

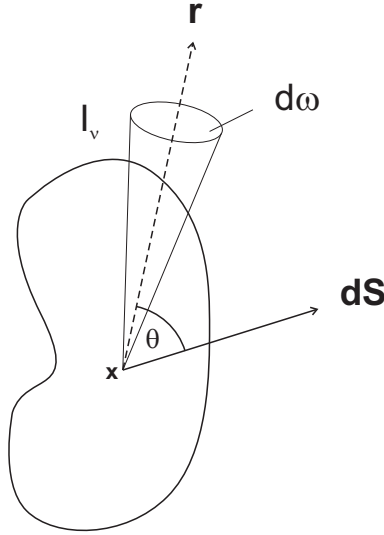


Figure 1.1: The unit vector \mathbf{r} is the direction of propagation of the radiation, while $d\mathbf{S}$ is perpendicular to the element of area.

Since the travel time of a photon through the photosphere is much shorter than any other relevant timescale and mainly regions of relatively high density are considered, the radiation field can be assumed to adjust instantaneously to any change of the thermodynamical state of the gas. Therefore we can neglect the time derivative in (1.4) and we obtain the time-independent radiative transfer equation (RTE)

$$\mathbf{r} \cdot \nabla I_\nu = \rho \chi_\nu (S_\nu - I_\nu). \quad (1.6)$$

The 1D equation of radiative transfer. For the development of the short characteristic algorithm only the one-dimensional RTE is used. The region of interest is a plane parallel model in direction x . Thus the 1D RTE is

$$r \frac{\partial I_\nu}{\partial x} = \rho \chi_\nu (S_\nu - I_\nu), \quad (1.7)$$

where $r = \mathbf{r}_1$. Introducing the optical depth of a path element at frequency ν as the independent coordinate

$$d\tau_\nu = \chi_\nu \rho dx \quad (1.8)$$

the 1D RTE can be written as

$$r \frac{\partial I_\nu}{\partial \tau_\nu} = S_\nu - I_\nu. \quad (1.9)$$

For $r = 1$, this equation has the formal solution

$$I(\tau_\nu) = I(\tau_{\nu,1})e^{(\tau_{\nu,1}-\tau_\nu)} + \int_{\tau_{\nu,1}}^{\tau_\nu} S_\nu(t)e^{-(\tau_\nu-t)} dt. \quad (1.10)$$

If the value $I(\tau_{\nu,1})$ and the values of $S(t)$ along the direction of the ray between optical depth $\tau_{\nu,1}$ and τ_ν are known, the value $I(\tau_\nu)$ can be determined.

Local thermodynamic equilibrium (LTE). 'Thermodynamic equilibrium prevails when a single value T of the temperature is sufficient to describe the thermodynamic state everywhere. The particles then have Maxwellian velocity distributions for that T , the states of ionization and excitation of the atoms are distributed according to the Saha and Boltzmann equation for that same T , and the radiation field has the homogeneous and isotropic black-body form given by the Kirchhoff-Planck function for this T , namely

$$B_\nu(T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{h\nu/kT} - 1}. \quad (1.11)$$

No temperature gradient exists in thermodynamic equilibrium. It is obvious that this situation is realized virtually nowhere.' (Stix [34])

'Local thermodynamic equilibrium means that at a certain place, a single temperature T does suffice to describe the statistical particle velocities, the population of the atomic states, and the local ratio of emission to absorption of radiation. In LTE the most important simplification of the radiative transfer problem is the relation $S_\nu = B_\nu(T)$.' (Stix [34]) (Scattering is not considered here.)

'Whether or not LTE can be assumed depends on the thermalization length. This is the distance over which a particle or photon emitted in a certain collision or transition has undergone sufficient further collisions or absorption/emission processes so that it can no longer be distinguished within the respective distribution. In LTE the thermalization length must be shorter than the distance over which the temperature of the gas changes markedly.' (Stix [34])

'Significant departures from LTE must be expected in the upper photosphere, especially in strong lines where scattering dominates over thermal emission and disturbs the detailed energy balance of LTE. This effect can be neglected as long as these lines do not contribute significantly to the total radiative heating rate Q_{rad} .' (Vögler [35])

Radiative heating rate. If the radiation field is known, the mean intensity is

$$J_\nu = \frac{1}{4\pi} \int I_\nu(\mathbf{r}) d\omega \quad (1.12)$$

and the radiative energy flux is

$$F_\nu = \int I_\nu(\mathbf{r}) \mathbf{r} d\omega. \quad (1.13)$$

The radiative heating rate, which represents the difference between absorption and emission, can be determined either from

$$Q_{\text{rad}} = 4\pi\rho \int_\nu \chi_\nu (J_\nu - S_\nu) d\nu \quad (1.14)$$

or from the equivalent expression

$$Q_{\text{rad}} = - \int_\nu (\nabla \cdot F_\nu) d\nu. \quad (1.15)$$

1.3 The equation of state

The conservations laws for radiation hydrodynamic (1.1), (1.2), (1.3) are completed by the equation of state which describes the relation between the thermodynamical quantities. It depends on the physical properties of the gas under study. At the uppermost few Mm of the the convection zone the solar plasma is partly ionized and the simple thermodynamical relations for an ideal gas do not apply.

Realistic microphysics is included by the OPAL equation of state and opacities (Rogers, Swenson and Iglesias [26]) and the Alexander low-temperature Rosseland opacities (Alexander and Ferguson [1]) for grey radiative transfer. For non-grey radiative transfer parts of the ATLAS 9 package (Kurucz [14]) are used (i.e. opacity distribution functions, model atmosphere) to determine bin-averaged opacities and source functions.

Chapter 2

The ANTARES code

The ANTARES (Advanced Numerical Tool for Astrophysical RESearch) code as used here can perform compressible hydrodynamic simulations with full radiative transfer in 1D, 2D and 3D cases on a rectangular grid, all encompassed in one parallelized Fortran90 program. Various high resolution numerical schemes (essentially non-oscillatory and convex non-oscillatory (Liu, Osher and Chan [17], Fedkiw et al. [8], Liu and Osher [16])) are implemented. The order of spatial and temporal discretization can be chosen arbitrarily to a considerable extent. Since the 'real' viscosity often is too small to stabilize the simulation either the numerical viscosity of the methods or the explicit addition of stabilizing viscosities must ensure stability. The viscous terms can either be discretized by a fourth order accurate scheme or be replaced by artificial diffusivities (Stein and Nordlund [32], Caunt and Korpi [5]). Furthermore the ANTARES code allows grid-refinement in rectangular patches, even recursively. Data from the refinement are projected up to the coarser level.

Part of this PhD project were:

- The development of a non-local and non-grey radiative transfer solver. All other terms in the hydrodynamic equations can be expressed locally by the physical quantities and their derivatives, the radiative transfer is non-local. Therefore the numerical treatment is very different. For details see section 3.4.
- Parallelization of the whole ANTARES Code via MPI (Message Passing Interface).
- The development of artificial diffusivities. The viscosity tensor in the momentum and energy equations is replaced by artificial equivalents, in order to remove short-wavelength noise and to diffuse discontinuities.
- The development and discretization of appropriate boundary conditions for solar convection.

2.1 The structure of the ANTARES code

The physical state \mathbf{q}^{n+1} at time step $n + 1$ is calculated from the physical state \mathbf{q}^n at time step n by a Runge-Kutta scheme (section 3.5) using either one or two intermediate states. The spatial derivatives are computed according to the finite volume method. Source terms are evaluated at the cell centers.

Each (intermediate) state is calculated by the following algorithm:

- Start with a physical state given as cell averages.
- Apply hydrodynamical boundary conditions (section 2.2.4).
- Calculate the inviscid cell boundary fluxes in all directions using a one-dimensional ENO- or CNO-scheme (section 3.2).
- Calculate the radiative heating rate Q_{rad} (section 3.4).

- Calculate the viscous cell boundary fluxes by fourth order interpolation or by the use of artificial diffusivities (section 3.2).
- Determine the new values for $\rho, \rho \mathbf{u}, e$ according to the discretization of the conservation laws.
- Call the equation of state to get all other used quantities.

2.2 Modelling a box in the sun

The system of equations (1.1), (1.2), (1.3) models the quiet sun, i.e. regions without magnetic activity.

The region of interest D for the numerical simulation is a rectangular domain (rectangle or cuboid) at the solar surface (see figure 2.1). The dimensions of this domain are approximately 2800km in the vertical direction and 11200km in the horizontal direction(s). Continuum optical depth unity (Rosseland mean) is at spatial depth about 600km from the top¹.

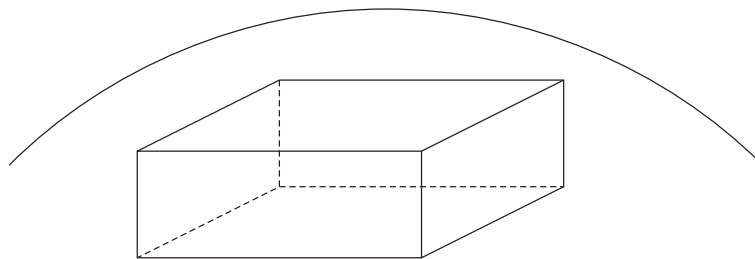


Figure 2.1: Box in the sun.

The x -direction points in the vertical direction, $x = 0$ is at the top of the domain. y - and z -direction are the horizontal coordinates.

2.2.1 Initial condition

First, a plane parallel model of the solar atmosphere by Christensen-Dalsgaard et al. [6] between 4350K and 20000K (see figure 2.2) with slightly disturbed x -momentum is evolved in time. Afterwards the two-dimensional physical state is converted to a three-dimensional one by putting the two-dimensional model side-by-side. Here the existing y -momentum (ρv) and the new z -momentum ($\rho w = 0$) are slightly disturbed. Then the model is evolved in time.

¹The average continuum optical depth unity level of the fully developed convection in its statistically steady state is shifted upwards by approximately 80-100km. This effect is typical for hydrodynamical simulations of convection. It is a result of the turbulent pressure of the convective flows (see e.g. Stein and Nordlund [32]).

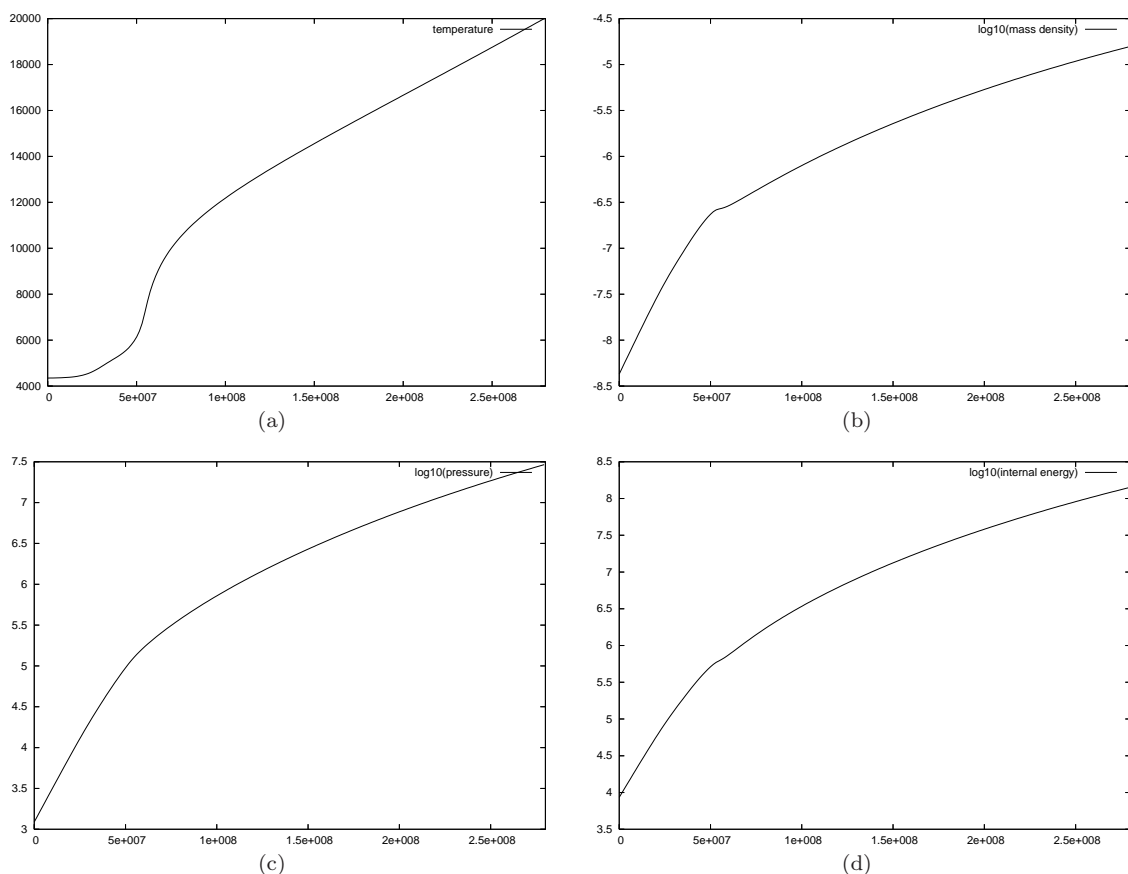


Figure 2.2: One-dimensional model of the sun (Christensen-Dalsgaard et al. [6]): temperature (a), logarithm of the mass density (b), pressure (c) and the internal energy density (d).

2.2.2 Numerical grid

The rectangular computational domain is equipped with a uniform numerical grid (see figure 2.3): N_x grid points in x-direction, N_y grid points in y-direction and N_z grid points in z-direction with up to seven ghost cells at each boundary, depending on the order of the spatial discretizations. The mesh sizes are Δx , Δy and Δz .

The x -range covers $x \in [0, x_{\max}]$ and the horizontal ranges cover $y \in [0, y_{\max}]$ and $z \in [0, z_{\max}]$. Thus the numerical grid becomes:

$$\begin{aligned} x_i &= (i-1) \cdot \Delta x & (i = 1, \dots, N_x), & \text{ with } \Delta x = x_{\max}/(N_x - 1) \\ y_j &= (j-1) \cdot \Delta y & (j = 1, \dots, N_y), & \text{ with } \Delta y = y_{\max}/(N_y - 1) \\ z_k &= (k-1) \cdot \Delta z & (k = 1, \dots, N_z), & \text{ with } \Delta z = z_{\max}/(N_z - 1) \end{aligned}$$

This defines cell boundary values

$$\begin{aligned} x_{i \pm \frac{1}{2}} &= x_i \pm \frac{\Delta x}{2} \\ y_{j \pm \frac{1}{2}} &= y_j \pm \frac{\Delta y}{2} \\ z_{k \pm \frac{1}{2}} &= z_k \pm \frac{\Delta z}{2} \end{aligned}$$

and cells

$$I_{i,j,k} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \times [z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}]$$

for $i = 1, \dots, N_x$, $j = 1, \dots, N_y$ and $k = 1, \dots, N_z$.

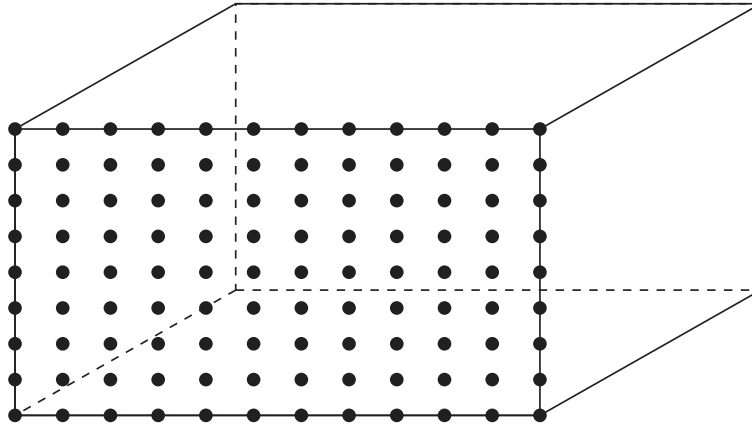


Figure 2.3: Numerical grid: Copies of the shown two-dimensional grid are put side by side such that the whole computational domain is equipped with grid points.

2.2.3 Domain decomposition for parallelization

Parallelization is done with MPI according to the distributed memory concept. Each processor performs a simulation on a rectangular subdomain. The number of required processors is specified by the number of subdivisions in x -, y - and z -direction. Each subdomain is equipped with ghost cells in each direction. If an edge area of a subdomain is border to an other subdomain the ghost cells are filled with the values from the neighboring subdomain. If an edge area represents a physical boundary then the the physical boundary conditions are applied. Corner values (see figure 2.4) in any $(y \times z)$ -plane are interpolated if required.

The domain decomposition procedure for the one-dimensional case is illustrated in figure 2.5. The whole domain (physical (white) and ghost (grey) cells) is split up. For each subdomain additional ghost cells are introduced. These ghost cells are filled with the values of the adjacent domain.

Each (intermediate) state is distributed to the adjacent subdomains.

Radiative transfer. Since radiative transfer is a non-local phenomenon each subdomain must wait until information about a directed (one-dimensional) ray from the corresponding neighboring subdomain is available. (For details see section 3.4 about the short characteristic method and its implementation for numerical radiative transfer.)

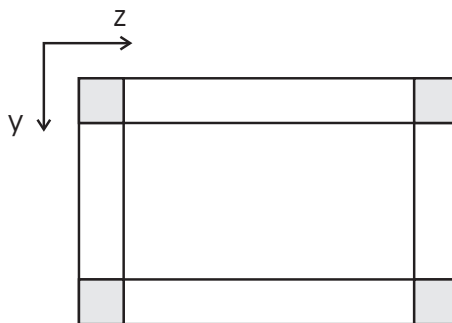


Figure 2.4: Corners in the $(y \times z)$ -plane.

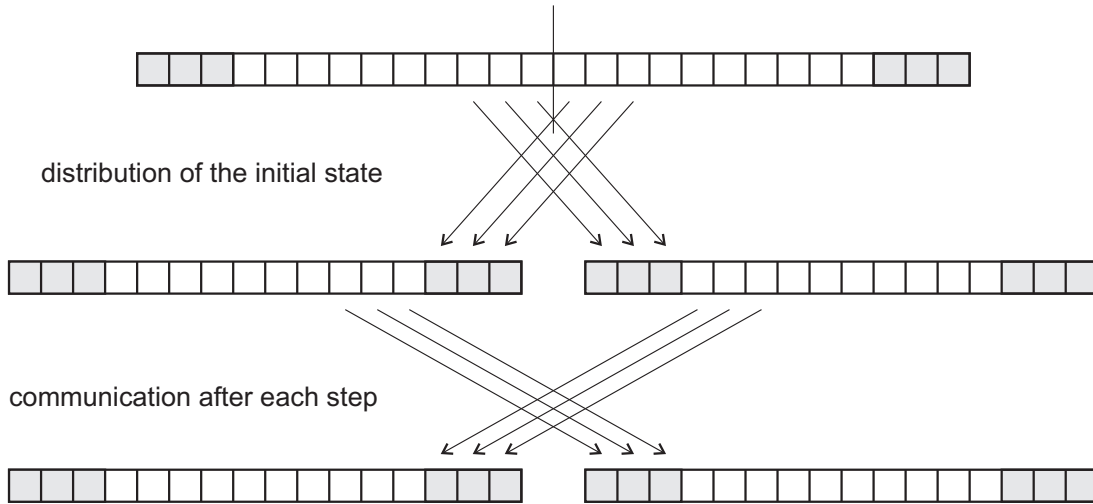


Figure 2.5: Domain decomposition in two subdomains for the one-dimensional case.

2.2.4 Hydrodynamical boundary conditions

Horizontal boundary conditions. All quantities are assumed to be periodic in both horizontal directions.

Upper and lower boundary. The simulations of solar surface flows are performed with closed boundary conditions at the upper and lower boundary of the computational domain. These boundary conditions lead to unphysical reflections of waves and shocks. Since the region of interest is around optical depth unity this is a reasonable assumption. Closed boundaries only have small influence at this depth. (Note that $\mathbf{u} = (u, v, w)^T$.)

For closed boundaries at the top of the computational domain (similar to Vögler et al. [36]) it is assumed that the vertical convective fluxes of mass, horizontal momentum and energy (except radiative energy) vanish. This is achieved by setting the vertical velocity and the vertical gradients of mass density and energy density to zero:

$$\begin{aligned} u|_{\text{top}} &\equiv 0 \\ \frac{\partial \rho}{\partial x} \Big|_{\text{top}} &\equiv 0 \\ \frac{\partial e}{\partial x} \Big|_{\text{top}} &\equiv 0 \end{aligned}$$

On the horizontal velocity components stress-free boundary conditions are applied:

$$\begin{aligned} \frac{\partial v}{\partial x} \Big|_{\text{top}} &\equiv 0 \\ \frac{\partial w}{\partial x} \Big|_{\text{top}} &\equiv 0 \end{aligned}$$

At the bottom again the boundary conditions for the momentum read:

$$\begin{aligned} u|_{\text{bot}} &\equiv 0 \\ \frac{\partial v}{\partial x} \Big|_{\text{bot}} &\equiv 0 \\ \frac{\partial w}{\partial x} \Big|_{\text{bot}} &\equiv 0 \end{aligned}$$

Since there is no fluid motion through the bottom of the domain the convective and the kinetic energy fluxes are zero. Hence the incoming energy transport is adjusted by introducing a radiative flux density $F_r = \kappa \nabla T$ where κ is the thermal conductivity. F_r is set to the appropriate value for the sun.

Numerical implementation. At the top of the domain three ghost cells are used to implement the boundary conditions. The physical state of the horizontal layer with $i = 1$ is continued vertically to the ghost cells for density, energy, y - and z -momentum. The ghost cell value for the x -momentum is set to 0.

At the bottom of the domain there is one ghost cell to implement the momentum boundary conditions. y - and z -momentum are continued vertically, the x -momentum is set to zero. To adjust the radiative flux the vertical temperature gradient at each point is constant in time and κ is increased at three grid points to get the desired radiative flux.

Conserved quantities. Due to the implemented boundary conditions mass and energy are conserved, the simulations are save for radiative gains or losses.

Spatial discretization. If not enough ghost cells in the vertical direction are available for determining the cell boundary values, depending on the numerical method, the order of the spatial discretization is reduced, the required values are extrapolated or the order of the method is preserved using asymmetric stencils.

2.2.5 Numerical problems

The maximal allowed time step is determined by the CFL-condition. In a single time step information transport by flow velocities must not be longer than the mesh width. Thus we get a time step restriction

$$\Delta t_{\text{CFL}} \leq C_{\text{Courant}} \frac{\min_i(\Delta x_i)}{\max(\|\mathbf{u}\|) + \max(c_{\text{sound}})} \quad (2.1)$$

where C_{Courant} is the Courant number and c_{sound} is the sound velocity. The max in the denominator is taken over the whole domain. For all simulations presented here a Courant number $C_{\text{Courant}} = \frac{1}{4}$ is selected.

Stronger diffusion results in smaller time-steps. The fourth order discretization of the viscous stress tensor (see section 3.2.2) results in the time step restriction:

$$\Delta t_{\text{diffusive}} \leq C_{\text{diffusive}} \frac{\min_i(\Delta x_i)^2}{\frac{4}{3} \max \mu} \quad (2.2)$$

If artificial diffusivities (see section 3.2.2) are used an additional diffusive time step restriction is applied:

$$\Delta t_{\text{diffusive}} \leq C_{\text{diffusive}} \frac{\min_i(\Delta x_i)^2}{\max \nu} \quad (2.3)$$

$C_{\text{diffusive}}$ also is $\frac{1}{4}$. Again the max in the denominator is taken over the whole domain. In the latter the argument of the max-function are all artificial diffusivities $\nu = \nu^{\text{shk}} + \nu^{\text{hyp}}$.

Consequently, the used time step Δt for the simulations presented in the chapters 4, 5 and 6 is

$$\Delta t = \min(\Delta t_{\text{CFL}}, \Delta t_{\text{diffusive}}). \quad (2.4)$$

This choice is adequate to ensure that the code remains stable.

2.2.6 Equation of state

The equation of state is included by precomputed values for a logarithmic density-temperature-grid. 500 grid points cover the logarithmic mass density range and 1000 grid points cover the logarithmic temperature range. For a given pair (ρ, t) one gets all required thermodynamical quantities through interpolation in the logarithmic density-temperature-grid.

Since the system of conservation laws (1.1), (1.2), (1.3) for ρ , $\rho\mathbf{u}$ and e is solved in time the internal energy (total minus kinetic energy) and the temperature must be calculated to apply the equation of state. The latter is interpolated in a precomputed logarithmic density-internal energy grid with values for the temperature. Here 500 grid points cover the logarithmic mass density range and 1000 cover the logarithmic internal energy range.

The bin-averaged quantities for non-grey radiative transfer are interpolated in a precomputed logarithmic density-pressure-grid, where 56 grid points cover the logarithmic temperature range and 21 cover the logarithmic pressure range.

Chapter 3

Implemented numerical schemes

The spatial derivatives of the system of conservation laws (1.1), (1.2), (1.3) are calculated according to the finite volume method. The values of the fluxes are interpolated to the cell boundaries (see section 3.1 for the inviscid fluxes and section 3.2 for the viscous fluxes) and then the divergence for each cell is computed. The radiative heating rate is the most expensive part of the whole algorithm and determined with the short-characteristic method to solve the radiative transfer equation (see section 3.4). All other source terms are simply evaluated at the cell centers. Furthermore the used interpolation tool (see section 3.3) and the temporal discretization (3.5) are described.

3.1 Numerical schemes for conservation laws

The hyperbolic part of the system of conservation laws (1.1), (1.2), (1.3) can be written as

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ e \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \mathbf{u} + p \underline{I} \\ (e + p) \mathbf{u} \end{bmatrix} = 0. \quad (3.1)$$

Let

$$\begin{aligned} \mathbf{u} &= [u, v, w]^T \\ \mathbf{q} &= [\rho, \rho u, \rho v, \rho w, e]^T \\ f(\mathbf{q}) &= [\rho u, \rho u^2 + p, \rho uv, \rho uw, u(e + p)]^T \\ g(\mathbf{q}) &= [\rho v, \rho vu, \rho v^2 + p, \rho vw, v(e + p)]^T \\ h(\mathbf{q}) &= [\rho w, \rho wu, \rho wv, \rho w^2 + p, w(e + p)]^T. \end{aligned}$$

Then (3.1) can be rewritten as

$$\mathbf{q}_t + [f(\mathbf{q})]_x + [g(\mathbf{q})]_y + [h(\mathbf{q})]_z = 0 \quad (3.2)$$

or

$$\mathbf{q}_t + \sum_{i=1}^3 [f_i(\mathbf{q})]_{x_i} = 0 \quad (3.3)$$

where $x_1 = x$, $x_2 = y$, $x_3 = z$, $f_1 = f$, $f_2 = g$ and $f_3 = h$.

Each Jacobian $f'_i(\mathbf{q})$ ($i = 1, 2, 3$) at each point has got five eigenvalues

$$\lambda_1(\mathbf{q}) \leq \lambda_2(\mathbf{q}) \leq \lambda_3(\mathbf{q}) \leq \lambda_4(\mathbf{q}) \leq \lambda_5(\mathbf{q})$$

with corresponding right and left eigenvectors

$$\begin{aligned} &\mathbf{r}_1(\mathbf{q}), \mathbf{r}_2(\mathbf{q}), \mathbf{r}_3(\mathbf{q}), \mathbf{r}_4(\mathbf{q}), \mathbf{r}_5(\mathbf{q}) \text{ and} \\ &\mathbf{l}_1(\mathbf{q}), \mathbf{l}_2(\mathbf{q}), \mathbf{l}_3(\mathbf{q}), \mathbf{l}_4(\mathbf{q}), \mathbf{l}_5(\mathbf{q}). \end{aligned}$$

Let

$$R(\mathbf{q}) = (\mathbf{r}_1(\mathbf{q}), \mathbf{r}_2(\mathbf{q}), \mathbf{r}_3(\mathbf{q}), \mathbf{r}_4(\mathbf{q}), \mathbf{r}_5(\mathbf{q}))$$

then

$$R^{-1}(\mathbf{q})f'(\mathbf{q})R(\mathbf{q}) = \Lambda(\mathbf{q}) = \begin{pmatrix} \lambda_1(\mathbf{q}) & & & & 0 \\ & \lambda_2(\mathbf{q}) & & & \\ & & \lambda_3(\mathbf{q}) & & \\ & & & \lambda_4(\mathbf{q}) & \\ 0 & & & & \lambda_5(\mathbf{q}) \end{pmatrix}.$$

The rows of $R^{-1}(\mathbf{q})$ are the left eigenvalues.

The eigenvalues and eigenvectors are different for different states \mathbf{q} . Using the transformation $\mathbf{p} = R^{-1}(\mathbf{q})\mathbf{q}$ respectively $\mathbf{q} = R(\mathbf{q})\mathbf{p}$ one can change to the local characteristic variables.

Dimensional splitting. In the ANTARES code the spatial discretization is performed component-wisely for each direction separately, for $[f(\mathbf{q})]_x$, $[g(\mathbf{q})]_y$ and $[h(\mathbf{q})]_z$. Numerical approximations $\hat{f}_{i\pm\frac{1}{2},j,k}$, $\hat{g}_{i,j\pm\frac{1}{2},k}$ and $\hat{h}_{i,j,k\pm\frac{1}{2}}$ at the cell boundaries are calculated in order to get approximations for

$$\begin{aligned} [f(\mathbf{q})]_x &\approx \frac{1}{\Delta x} (\hat{f}_{i+\frac{1}{2},j,k} - \hat{f}_{i-\frac{1}{2},j,k}) \\ [g(\mathbf{q})]_y &\approx \frac{1}{\Delta y} (\hat{g}_{i,j+\frac{1}{2},k} - \hat{g}_{i,j-\frac{1}{2},k}) \\ [h(\mathbf{q})]_z &\approx \frac{1}{\Delta z} (\hat{h}_{i,j,k+\frac{1}{2}} - \hat{h}_{i,j,k-\frac{1}{2}}) \end{aligned}$$

at point $x_{i,j,k}$. Thus the description of the numerical schemes for conservation laws needs only to consider one space dimension.

One-dimensional spatial grid. Let the cell centers x_i ($i = 1, \dots, N$) define the one-dimensional equidistant numerical grid with mesh size Δx . For $i = 1, \dots, N$ the cell boundaries are $x_{i\pm\frac{1}{2}} = x_i \pm \frac{\Delta x}{2}$ and the cells are $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. Boundary conditions are not considered in this section, all values are available for $i \leq 0$ and $i > N$ if needed.

3.1.1 Essentially non-oscillatory (ENO) and Weighted essentially non-oscillatory (WENO) schemes

First the algorithms to reconstruct cell boundary values from given cell averages are described and then their application for scalar, one- and multi-space-dimensional systems of conservation laws is shown.

Essentially non-oscillatory (ENO) schemes

Numerical methods for systems of nonlinear conservation laws must capture steep gradients (shocks and contact discontinuities) that may develop spontaneously and then persist in the solution. Classical numerical schemes either produce oscillations near steep gradients or smear out these gradients as well as fine details (see e.g. LeVeque [15]).

Essentially non-oscillatory (ENO) methods (Harten et al. [9], Shu and Osher [29]) use adaptive stencils for polynomial interpolation (for the evaluation of the cell wall fluxes) to avoid high order polynomial interpolation across steep gradients which leads to oscillations in the computed solution.

For systems of conservation laws this interpolation must be done in the local characteristic field since the Riemann invariants properly propagate in various directions.

The presentation of the (weighted) ENO algorithms here follows the presentation in Shu [28].

Reconstruction from cell averages. Given the cell averages $\bar{v}_i \equiv v(x_i)$ ($i = 1, \dots, N$) of a function

$v(x)$, find a numerical flux function $\hat{v}_{i+\frac{1}{2}} \equiv \hat{v}(v_{i-r}, \dots, v_{i+s})$ ($i = 0, \dots, N$) such that the flux difference approximates the derivative $v'(x)$ to k -th order accuracy where v is sufficiently smooth:

$$\frac{1}{\Delta x} \left(\hat{v}_{i+\frac{1}{2}} - \hat{v}_{i-\frac{1}{2}} \right) = v'(x_i) + O(\Delta x^k) \quad (i = 1, \dots, N) \quad (3.4)$$

O is a Landau symbol¹. For each $i = 1, \dots, N$ consider a stencil $S(i) = \{I_{i-r}, \dots, I_{i+s}\}$, $r + s + 1 = k$, $r, s \geq 0$. Then there exists a unique polynomial $p(x)$ of degree $k - 1 = r + s$ with

$$\frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p(\xi) d\xi = \bar{v}_i \quad (3.5)$$

and

$$p(x) = v(x) + O(\Delta x^k) \text{ inside } I_i. \quad (3.6)$$

To construct $p(x)$ define the primitive function $V(x) = \int_{-\infty}^x v(\xi) d\xi$. Then

$$V(x_{i+\frac{1}{2}}) = \sum_{j=-\infty}^i \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} v(\xi) d\xi = \sum_{j=-\infty}^i \bar{v}_j \Delta x.$$

$P(x)$ is the unique polynomial of degree $\leq k$ which interpolates $V(x)$ at the following $k + 1$ points $x_{i-r-\frac{1}{2}}, \dots, x_{i+s+\frac{1}{2}}$. Let $p(x) = P'(x)$. Then (3.5) is valid since

$$\begin{aligned} \frac{1}{\Delta x} \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} p(\xi) d\xi &= \frac{1}{\Delta x} \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} P'(\xi) d\xi \\ &= \frac{1}{\Delta x} \left(P(x_{j+\frac{1}{2}}) - P(x_{j-\frac{1}{2}}) \right) \\ &= \frac{1}{\Delta x} \left(V(x_{j+\frac{1}{2}}) - V(x_{j-\frac{1}{2}}) \right) \\ &= \frac{1}{\Delta x} \left(\int_{-\infty}^{x_{j+\frac{1}{2}}} v(\xi) d\xi - \int_{-\infty}^{x_{j-\frac{1}{2}}} v(\xi) d\xi \right) \\ &= \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(\xi) d\xi \\ &= \bar{v}_j \quad \forall j = i - r, \dots, i + s. \end{aligned}$$

Since $P(x) = V(x) + O(\Delta x^{k+1})$ standard approximations theory (e.g. Neumaier [23]) yields $p(x) = v(x) + O(\Delta x^k)$ and (3.6) is valid.

The polynomial $p(x)$ yields approximations:

$$\begin{aligned} \hat{v}_{i+\frac{1}{2}}^- &= p(x_{i+\frac{1}{2}}) = v(x_{i+\frac{1}{2}}) + O(\Delta x^k) \\ \hat{v}_{i-\frac{1}{2}}^+ &= p(x_{i-\frac{1}{2}}) = v(x_{i-\frac{1}{2}}) + O(\Delta x^k) \end{aligned}$$

Differentiating $v(x_i) \equiv \bar{v}_i = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(\xi) d\xi$ yields $v'(x_i) = \frac{1}{\Delta x} \left(v(x_{i+\frac{1}{2}}) - v(x_{i-\frac{1}{2}}) \right)$. Replacing $v(x_{i+\frac{1}{2}})$ and $v(x_{i-\frac{1}{2}})$ by the approximations $\hat{v}_{i+\frac{1}{2}}^-$ and $\hat{v}_{i-\frac{1}{2}}^+$ the accuracy requirement (3.4) is achieved as long as the $O(\Delta x^k)$ terms in the approximations $\hat{v}_{i+\frac{1}{2}}^-$ and $\hat{v}_{i-\frac{1}{2}}^+$ are smooth. In practice, this assumption is true. Hence the difference in (3.4) gives an extra $O(\Delta x)$, just to cancel the one in the denominator.

The mapping from the given cell averages $\{\bar{v}_j\}$ in the stencil $S(i)$ to the values $\hat{v}_{i+\frac{1}{2}}^-$ and $\hat{v}_{i-\frac{1}{2}}^+$ is linear. Hence there are constants c_{rj} and \tilde{c}_{rj} which depend on the left shift r of the stencil $S(i)$ and the accuracy order k but not on the function $v(x)$ itself, such that:

$$\begin{aligned} \hat{v}_{i+\frac{1}{2}}^- &= \sum_{j=0}^{k-1} c_{rj} \bar{v}_{i-r+j} \\ \hat{v}_{i-\frac{1}{2}}^+ &= \sum_{j=0}^{k-1} \tilde{c}_{rj} \bar{v}_{i-r+j} \end{aligned}$$

¹In order to characterize the asymptotic behavior of the functions f and g in the neighborhood of $r^* \in \mathbb{R} \cup \infty$ (a value usually apparent from the context) one writes $[...] f(r) = O(g(r))$ when $\frac{f(r)}{g(r)}$ remains bounded as $r \rightarrow r^*$. In particular, $[...] f(r) = O(g(r))$, $\lim_{r \rightarrow r^*} g(r) = 0 \Rightarrow \lim_{r \rightarrow r^*} f(r) = 0$. (Neumaier [23]).

Note that $\tilde{c}_{rj} = c_{r-1,j}$. Manipulating the Lagrange form of the interpolation polynomial $P(x)$ and building the derivative to get $p(x)$ one gets these coefficients c_{rj} and \tilde{c}_{rj} which can be prestored. See table 3.1 for the coefficients up to order $k = 6$ (Shu [28]).

k	r	j=0	j=1	j=2	j=3	j=4	j=5
1	-1	1					
	0	1					
2	-1	3/2	-1/2				
	0	1/2	1/2				
	1	-1/2	3/2				
3	-1	11/6	-7/6	1/3			
	0	1/3	5/6	-1/6			
	1	-1/6	5/6	1/3			
	2	1/3	-7/6	11/6			
4	-1	25/12	-23/12	13/12	-1/4		
	0	1/4	13/12	-5/12	1/12		
	1	-1/12	7/12	7/12	-1/12		
	2	1/12	-5/12	13/12	1/4		
	3	-1/4	13/12	-23/12	25/12		
5	-1	137/60	-163/60	137/60	-21/20	1/5	
	0	1/5	77/60	-43/60	17/60	-1/20	
	1	-1/20	9/20	47/60	-13/60	1/30	
	2	1/30	-13/60	47/60	9/20	-1/20	
	3	-1/20	17/60	-43/60	77/60	1/5	
	4	1/5	-21/20	137/60	-163/60	137/60	
6	-1	49/20	-71/20	79/20	-163/60	-31/30	-1/6
	0	1/6	29/20	-21/20	37/60	-13/60	1/30
	1	-1/30	11/30	19/20	-23/60	7/60	-1/60
	2	1/60	-2/15	37/60	37/60	-2/15	1/60
	3	-1/60	7/60	-23/60	19/20	11/30	-1/30
	4	1/30	-13/60	37/60	-21/20	29/20	1/6
	5	-1/6	31/30	-163/60	79/20	-71/20	49/20

Table 3.1: The constants c_{rj}

ENO approximation. Near discontinuities in the solution of hyperbolic conservation laws oscillations can occur because the stencil $S(i)$ contains the discontinuity. Therefore an adaptive stencil is taken for the interpolation of the cell boundary fluxes, the left shift r changes with the location x_i . The main idea is to exclude discontinuous cells from the stencil $S(i)$.

To decide whether a stencil contains a discontinuity divided differences of $V(x)$ are considered. The 0-th degree divided difference is

$$V[x_{i-\frac{1}{2}}] \equiv V(x_{i-\frac{1}{2}}).$$

The j -th degree divided difference is defined recursively by

$$V[x_{i-\frac{1}{2}}, \dots, x_{i+j-\frac{1}{2}}] \equiv \frac{V[x_{i+\frac{1}{2}}, \dots, x_{i+j-\frac{1}{2}}] - V[x_{i-\frac{1}{2}}, \dots, x_{i+j-\frac{3}{2}}]}{x_{i+j-\frac{1}{2}} - x_{i-\frac{1}{2}}} \quad (j \geq 1).$$

Similarly, the divided differences of the cell averages \bar{v} are defined by

$$\begin{aligned}\bar{v}[x_i] &\equiv \bar{v}(x_i), \\ \bar{v}[x_i, \dots, x_{i+j}] &\equiv \frac{\bar{v}[x_{i+1}, \dots, x_{i+j}] - \bar{v}[x_i, \dots, x_{i+j-1}]}{x_{i+j} - x_i} \quad (j \geq 1).\end{aligned}$$

Note that

$$V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] = \frac{V(x_{i+\frac{1}{2}}) - V(x_{i-\frac{1}{2}})}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} = \bar{v}_i,$$

the 0-th degree divided differences of \bar{v} are equal to the first degree divided differences of $V(x)$.

The Newton form of the interpolation polynomial $P(x)$ which interpolates $V(x)$ at the $k+1$ points of the stencil $S(i)$ can be expressed as

$$P(x) = \sum_{j=0}^k V[x_{i-r-\frac{1}{2}}, \dots, x_{i-r+j-\frac{1}{2}}] \prod_{m=0}^{j-1} (x - x_{i-r+m-\frac{1}{2}}). \quad (3.7)$$

Hence

$$p(x) = P'(x) = \sum_{j=1}^k V[x_{i-r-\frac{1}{2}}, \dots, x_{i-r+j-\frac{1}{2}}] \sum_{m=0}^{j-1} \prod_{\substack{l=0 \\ l \neq m}}^{j-1} (x - x_{i-r+l-\frac{1}{2}}). \quad (3.8)$$

In the polynomial $p(x)$ only first and higher order divided differences of $V(x)$ appear, hence $p(x)$ can be expressed completely by the divided differences of \bar{v} .

Divided differences are a measurement of the smoothness of a function within a given stencil (e.g. Neumaier [23]): If V is smooth in the stencil $S(i)$ then

$$V[x_{i-\frac{1}{2}}, \dots, x_{i+j-\frac{1}{2}}] = \frac{V^{(j)}(\xi)}{j!}$$

for some $x_{i-\frac{1}{2}} < \xi < x_{i+j-\frac{1}{2}}$ inside the stencil. If the stencil $S(i)$ contains a discontinuity of $V(x)$ then

$$V[x_{i-\frac{1}{2}}, \dots, x_{i+j-\frac{1}{2}}] = O(1/\Delta x^j).$$

ENO Algorithm. Given the cell averages $\{\bar{v}_i\}$ of a function $v(x)$, a piecewise polynomial reconstruction of degree $\leq k-1$ is obtained by the following algorithm:

- Calculate the divided differences of the primitive function $V(x)$ for degrees 1 to k using \bar{v} . If the grid is uniform use undivided differences in order to save computational time and reduce round-off errors.
- For each cell I_i start with the two-point-stencil $\tilde{S}_2(i) = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$ for $V(x)$ which is equivalent to the stencil $S_1(i) = \{I_i\}$.
- Recursively add a point to the stencil $\tilde{S}_l(i) = \{x_{j+\frac{1}{2}}, \dots, x_{j+l-\frac{1}{2}}\}$ where $l = 2, \dots, k$.
 - If $|V[x_{j-\frac{1}{2}}, \dots, x_{j+l-\frac{1}{2}}]| < |V[x_{j+\frac{1}{2}}, \dots, x_{j+l+\frac{1}{2}}]|$ add $x_{j-\frac{1}{2}}$ to the stencil $\tilde{S}_l(i)$ and get $\tilde{S}_{l+1}(i) = \{x_{j-\frac{1}{2}}, \dots, x_{j+l-\frac{1}{2}}\}$.
 - Otherwise add $x_{j+l+\frac{1}{2}}$ to the stencil $\tilde{S}_l(i)$ and get $\tilde{S}_{l+1}(i) = \{x_{j+\frac{1}{2}}, \dots, x_{j+l+\frac{1}{2}}\}$.
- Use the Lagrange or Newton form of the interpolation polynomial and the stencil defined above to obtain $p(x)$, which is a polynomial of degree $\leq k-1$. Use $p(x)$ to get the approximations

$$\hat{v}_{i+\frac{1}{2}}^- = p(x_{i+\frac{1}{2}})$$

and

$$\hat{v}_{i-\frac{1}{2}}^+ = p(x_{i-\frac{1}{2}}).$$

Weighted essentially non-oscillatory (WENO) schemes

Instead of choosing the 'smoothest' stencil for the interpolation polynomial in the ENO reconstruction, a convex combination of all candidates is used to achieve the essentially non-oscillatory property (Liu, Osher and Chan [17]).

WENO approximation. Performing a k -th order ENO scheme there are k candidate stencils

$$S_r(i) = \{x_{i-r}, \dots, x_{i-r+k-1}\}, \quad r = 0, \dots, k-1$$

which produce k different reconstructions to the value $v_{i+\frac{1}{2}}$:

$$\hat{v}_{i+\frac{1}{2}}^{(r)} = \sum_{j=0}^{k-1} c_{rj} \bar{v}_{i-r+j}, \quad r = 0, \dots, k-1$$

The WENO approach chooses a convex combination of these values $\hat{v}_{i+\frac{1}{2}}^{(r)}$ as a new approximation for $\bar{v}_{i+\frac{1}{2}}$.

$$\hat{v}_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r \hat{v}_{i+\frac{1}{2}}^{(r)}, \quad r = 0, \dots, k-1.$$

The weights ω_r must serve $\omega_r \geq 0 \forall r = 1, \dots, k-1$ and $\sum_{r=0}^{k-1} \omega_r = 1$.

For a smooth function $v(x)$ there are constants d_r (Liu, Osher and Chan [17]) such that

$$\begin{aligned} \hat{v}_{i+\frac{1}{2}} &= \sum_{r=0}^{k-1} d_r \hat{v}_{i+\frac{1}{2}}^{(r)} = v(x_{i+\frac{1}{2}}) + O(\Delta x^{2k-1}) \text{ and} \\ \sum_{r=0}^{k-1} d_r &= 1. \end{aligned}$$

For $1 \leq k \leq 3$ these constants are given by:

$$\begin{aligned} k=1: \quad d_0 &= 1 \\ k=2: \quad d_0 &= \frac{2}{3}, d_1 = \frac{1}{3} \\ k=3: \quad d_0 &= \frac{3}{10}, d_1 = \frac{3}{5}, d_2 = \frac{1}{10} \end{aligned}$$

If $\omega_r = d_r + O(\Delta x^{k-1}) \forall r = 1, \dots, k-1$ $\hat{v}_{i+\frac{1}{2}}$ is $(2k-1)$ -th order accurate:

$$\hat{v}_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r \hat{v}_{i+\frac{1}{2}}^{(r)} = v(x_{i+\frac{1}{2}}) + O(\Delta x^{2k-1})$$

because

$$\begin{aligned} \sum_{r=0}^{k-1} \omega_r \hat{v}_{i+\frac{1}{2}}^{(r)} - \sum_{r=0}^{k-1} d_r \hat{v}_{i+\frac{1}{2}}^{(r)} &= \sum_{r=0}^{k-1} (\omega_r - d_r) \left(\hat{v}_{i+\frac{1}{2}}^{(r)} - v(x_{i+\frac{1}{2}}) \right) \\ &= \sum_{r=0}^{k-1} O(\Delta x^{k-1}) O(\Delta x^k) = O(\Delta x^{2k-1}). \end{aligned}$$

In Liu, Osher and Chan [17] weights of the form

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{k-1} \alpha_s}, \quad r = 0, \dots, k-1 \quad (3.9)$$

with

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}.$$

are proposed. $\epsilon > 0$ is the machine accuracy, which is introduced here to keep the denominator from becoming 0. β_r are smoothness indicators of the stencil $S_r(i)$, which should be essentially 0 if $S_r(i)$ contains a discontinuity.

Through Taylor expansion analysis, weights of the form

$$\beta_r = D(1 + O(\Delta x^{k-1})), \quad \forall r = 1, \dots, k-1 \quad (3.10)$$

where D is a nonzero quantity independent of r , satisfy the accuracy requirement $\omega_r = d_r + O(\Delta x^{k-1}) \quad \forall r = 1, \dots, k-1$ (Jiang and Shu [10]). A robust choice of smoothness indicators is defined by (Jiang and Shu [10])

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial^l x} \right)^2 dx \quad (3.11)$$

where $p_r(x)$ is the reconstruction polynomial on the stencil $S_r(i)$. The smoothness indicators β_r are a measure for the total variation in the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$.

With this choice, for $k=2$, the smoothness measurement coefficients are

$$\begin{aligned} \beta_0 &= (\bar{v}_{i+1} - \bar{v}_i)^2, \\ \beta_1 &= (\bar{v}_i - \bar{v}_{i-1})^2. \end{aligned} \quad (3.12)$$

For $k=3$, the smoothness measurement coefficients are

$$\begin{aligned} \beta_0 &= \frac{13}{12}(\bar{v}_i - 2\bar{v}_{i+1} + \bar{v}_{i+2})^2 + \frac{1}{4}(3\bar{v}_i - 4\bar{v}_{i+1} + \bar{v}_{i+2})^2, \\ \beta_1 &= \frac{13}{12}(\bar{v}_{i-1} - 2\bar{v}_i + \bar{v}_{i+1})^2 + \frac{1}{4}(\bar{v}_{i-1} + \bar{v}_{i+1})^2, \\ \beta_2 &= \frac{13}{12}(\bar{v}_{i-2} - 2\bar{v}_{i-1} + \bar{v}_i)^2 + \frac{1}{4}(\bar{v}_{i-2} - 4\bar{v}_{i-1} + 3\bar{v}_i)^2. \end{aligned} \quad (3.13)$$

WENO algorithm. Given the cell averages $\{\bar{v}_i\}$ of a function $v(x)$, $(2k-1)$ -th order accurate approximations at the cell boundaries to the function $v(x)$ are obtained in the following way:

- Calculate the k approximations $\hat{v}_{i+\frac{1}{2}}^{(r)}$ and $\hat{v}_{i-\frac{1}{2}}^{(r)}$ of order k for $r = 0, \dots, k-1$.
- Determine the coefficients d_r and \tilde{d}_r such that

$$\begin{aligned} \hat{v}_{i+\frac{1}{2}} &= \sum_{r=0}^{k-1} d_r \hat{v}_{i+\frac{1}{2}}^{(r)} = v(x_{i+\frac{1}{2}}) + O(\Delta x^{2k-1}) \text{ and} \\ \hat{v}_{i-\frac{1}{2}} &= \sum_{r=0}^{k-1} \tilde{d}_r \hat{v}_{i-\frac{1}{2}}^{(r)} = v(x_{i-\frac{1}{2}}) + O(\Delta x^{2k-1}). \end{aligned}$$

Note that $\tilde{d}_r = d_{k-1-r}$.

- Determine the smoothness indicators β_r (3.10)-(3.13) for $r = 0, \dots, k-1$.
- Form the weights ω_r (3.9) and $\tilde{\omega}_r$ for $r = 0, \dots, k-1$.
- Calculate the $(2k-1)$ -th order accurate reconstructions

$$\begin{aligned} \hat{v}_{i+\frac{1}{2}}^- &= \sum_{r=0}^{k-1} \omega_r v_{i+\frac{1}{2}}^{(r)} \text{ and} \\ \hat{v}_{i-\frac{1}{2}}^+ &= \sum_{r=0}^{k-1} \tilde{\omega}_r v_{i-\frac{1}{2}}^{(r)}. \end{aligned}$$

Upwinding, One- and Multi-space-dimensional systems.

Upwinding. Consider a one-dimensional scalar conservation law

$$q(x, t)_t + f(q(x, t))_x = 0$$

equipped with suitable initial ($q(x, 0) = q_0(x)$) and boundary conditions. To advance a solution in time one considers the conservative discretization

$$\frac{d\bar{q}_i(t)}{dt} = -\frac{1}{\Delta x} \left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}} \right),$$

where $\bar{q}_i(t)$ is the numerical approximation to the cell average $\bar{q}(x_i, t)$. The numerical flux $\hat{f}_{i+\frac{1}{2}}$ is obtained by the ENO algorithm with $\bar{v}(x) = f(q(x, t))$. The ENO algorithm yields two values $\hat{v}_{i+\frac{1}{2}}^-$ and $\hat{v}_{i+\frac{1}{2}}^+$ resulting from the different starting stencils $S(i)$ and $S(i+1)$. For stability it is important that upwinding is used in constructing $\hat{f}_{i+\frac{1}{2}}$. This is for example achieved by computing the Roe speed $\bar{a}_{i+\frac{1}{2}} \equiv \frac{f(u_{i+1}) - f(u_i)}{u_{i+1} - u_i}$.

- If $\bar{a}_{i+\frac{1}{2}} \geq 0$, the wind blows from left to right. Thus the numerical flux $\hat{f}_{i+\frac{1}{2}} = \hat{v}_{i+\frac{1}{2}}^-$.
- If $\bar{a}_{i+\frac{1}{2}} < 0$, the wind blows from right to left. Thus the numerical flux $\hat{f}_{i+\frac{1}{2}} = \hat{v}_{i+\frac{1}{2}}^+$.

One-dimensional systems of conservation laws. The one-dimensional hydrodynamic hyperbolic systems of conservation laws can be written as

$$\mathbf{q}_t + [f(\mathbf{q})]_x = 0 \tag{3.14}$$

where $\mathbf{q} = [\rho, \rho u, e]^T$ and $f(\mathbf{q}) = [\rho u, \rho u^2 + p, u(e + p)]^T$. For every point x_i one calculates the eigenvalues and eigenvectors of the Jacobian $f'(\mathbf{q}_i)$ and then transforms the flux function $f(\mathbf{q})$ in a sufficiently large neighborhood $U(i)$ of x_i to the local characteristic variables:

$$\tilde{f}_j = R^{-1}(\mathbf{q}_i) f(\mathbf{q}_j), \quad j \in U(i)$$

After this transformation one performs the ENO algorithm for every component \tilde{f} separately. The Roe speed $\bar{a}_{i+\frac{1}{2}}$ is replaced by the eigenvalue for the corresponding characteristic variable. Then the calculated cell boundary values are transformed back into the physical space.

Multi-space-dimensional systems of conservation laws. The procedure for one-dimensional systems of conservation laws is performed for the flux function $f(\mathbf{q})$ in x-direction, for the flux function $g(\mathbf{q})$ in y-direction and for the flux function $h(\mathbf{q})$ in z-direction.

Thus one gets approximations for

$$\begin{aligned} [f(\mathbf{q})]_x &\approx \frac{1}{\Delta x} (\hat{f}_{i+\frac{1}{2},j,k} - \hat{f}_{i-\frac{1}{2},j,k}), \\ [g(\mathbf{q})]_y &\approx \frac{1}{\Delta y} (\hat{g}_{i,j+\frac{1}{2},k} - \hat{g}_{i,j-\frac{1}{2},k}), \\ [h(\mathbf{q})]_z &\approx \frac{1}{\Delta z} (\hat{h}_{i,j,k+\frac{1}{2}} - \hat{h}_{i,j,k-\frac{1}{2}}) \end{aligned}$$

at point $x_{i,j,k}$.

3.1.2 Essentially non-oscillatory schemes with Marquina's flux splitting and entropy fix

For some test problems ENO methods produce spurious oscillatory results (Fedkiw et al. [8]). These oscillations are due to the manner in which the transformation to local characteristic variables is evaluated (Donat and Marquina [7]). A cell wall may separate two very different states. Marquina then proposes a flux splitting technique where the transformation to characteristic variables is performed unambiguously on the left and on the right. Thus the subsequent flux calculations are well defined. The results are combined

in an upwind fashion to get the cell wall flux. When Marquina's flux splitting technique is used with standard ENO flux calculation, it fixes all problematic cases in Fedkiw et al. [8].

Let λ be the p -th eigenvalue of the Jacobian of the flux function in x -direction. Let q be the p -th characteristic variable and f be the flux function in x -direction for the characteristic variable q . Let H be the primitive function of f and let $D_k^j H$ denote the j -th order divided difference of the primitive function H at point x_k .

The third order accurate ENO-Roe discretization. At the cell wall $x_{i+\frac{1}{2}}$ the numerical flux function $\hat{f}_{i+\frac{1}{2}}$ for the characteristic variable q is calculated as follows:

- If $\lambda(\mathbf{q}_{i+\frac{1}{2}}) > 0$ set $k = i$. Otherwise set $k = i + 1$. Then define $Q_1(x) = (D_k^1 H)(x - x_{i+\frac{1}{2}})$.
- If $|D_{k-\frac{1}{2}}^2 H| \leq |D_{k+\frac{1}{2}}^2 H|$ then $c = D_{k-\frac{1}{2}}^2(H)$ and $k^* = k - 1$. Otherwise $c = D_{k+\frac{1}{2}}^2(H)$ and $k^* = k$. Then define $Q_2(x) = c(x - x_{k-\frac{1}{2}})(x - x_{k+\frac{1}{2}})$.
- If $|D_{k^*}^3 H| \leq |D_{k^*+1}^3 H|$ then $c^* = D_{k^*}^3(H)$. Otherwise $c^* = D_{k^*+1}^3(H)$. Then define $Q_3(x) = c^*(x - x_{k^*-\frac{1}{2}})(x - x_{k^*+\frac{1}{2}})(x - x_{k^*+\frac{3}{2}})$.

Then $\hat{f}_{i+\frac{1}{2}} = H'(x_{i+\frac{1}{2}}) = Q_1'(x_{i+\frac{1}{2}}) + Q_2'(x_{i+\frac{1}{2}}) + Q_3'(x_{i+\frac{1}{2}})$ which simplifies to

$$\hat{f}_{i+\frac{1}{2}} = D_k^1 H + c(2(i - k) + 1)\Delta x + c^*(3(i - k^*)^2 - 1)(\Delta x)^2. \quad (3.15)$$

The entropy fix. The ENO-Roe discretization allows entropy violating expansion shocks near sonic points, i.e. where a characteristic velocity changes sign. There it is possible that stationary 'expansion shock' solutions show up. At cell walls $x_{i+\frac{1}{2}}$ with a nearby sonic point, a high order dissipation is added in the calculation of $\hat{f}_{i+\frac{1}{2}}$. This correction is extremely small if the solution is locally smooth, but is large enough to break up an expansion shock.

Consider two primitive functions H^+ and H^- and compute a divided difference table for each of them. The first divided differences at point x_i are

$$D_i^1 H^\pm = \frac{1}{2}f(x_i) \pm \frac{1}{2}\alpha_{i+\frac{1}{2}}q(x_i) \quad (3.16)$$

where $\alpha_{i+\frac{1}{2}}$ is defined later. Normally the first order difference of the primitive function is the value of the flux function at the cell centers, i.e. $D_i^1 H = f(x_i)$.

For H^+ set $k = i$. Replace H with H^+ and determine $Q_1(x)$, $Q_2(x)$, $Q_3(x)$ and $\hat{f}_{i+\frac{1}{2}}^+$ using the algorithm above. For H^- set $k = i + 1$. Replace H with H^- and determine $Q_1(x)$, $Q_2(x)$, $Q_3(x)$ and $\hat{f}_{i+\frac{1}{2}}^-$ using the algorithm above. Then

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}_{i+\frac{1}{2}}^+ + \hat{f}_{i+\frac{1}{2}}^- \quad (3.17)$$

is the numerical flux function with added high order dissipation. This choice eliminates entropy violating expansion shocks.

Construction of Marquina's left and right Jacobians. Consider a cell wall $x_{i+\frac{1}{2}}$. $\mathbf{q}_{i+\frac{1}{2}}^L = \mathbf{q}_i$ is the estimate from the left and $\mathbf{q}_{i+\frac{1}{2}}^R = \mathbf{q}_{i+1}$ is the estimate from the right. With these two choices two Jacobian matrices $J^L = J(\mathbf{q}_{i+\frac{1}{2}}^L)$ and $J^R = J(\mathbf{q}_{i+\frac{1}{2}}^R)$ and their associated eigensystems are computed. For each of the two Jacobians a numerical flux in the p -th characteristic field at $x_{i+\frac{1}{2}}$ is determined: $\hat{f}_{i+\frac{1}{2}}^L$ and $\hat{f}_{i+\frac{1}{2}}^R$. Then $\hat{f}_{i+\frac{1}{2}} = \hat{f}_{i+\frac{1}{2}}^L + \hat{f}_{i+\frac{1}{2}}^R$.

ENO algorithm with Marquina's flux splitting. Consider a cell wall $x_{i+\frac{1}{2}}$. If the left and right eigenvalues at this cell wall agree on the upwind direction then there is no sonic point nearby and the ENO-Roe discretization is used. If the eigenvalues disagree a sonic point is nearby and the ENO with the entropy fix is used:

- If $\lambda^L > 0$ and $\lambda^R > 0$, upwind is from the left and $\hat{f}_{i+\frac{1}{2}}^L$ using ENO-Roe is calculated. $\hat{f}_{i+\frac{1}{2}}^R$ is set to 0.
- If $\lambda^L < 0$ and $\lambda^R < 0$, upwind is from the right and $\hat{f}_{i+\frac{1}{2}}^R$ using ENO-Roe is calculated. $\hat{f}_{i+\frac{1}{2}}^L$ is set to 0.
- If $\lambda^L \lambda^R \leq 0$, the signs of the eigenvalues disagree. The entropy fix version of ENO is used with dissipation coefficient $\alpha_{i+\frac{1}{2}} = \max(|\lambda^L|, |\lambda^R|)$. In the evaluation of $\hat{f}_{i+\frac{1}{2}}^L$, $\hat{f}_{i+\frac{1}{2}}^+$ is evaluated normally and $\hat{f}_{i+\frac{1}{2}}^- = 0$. Thus $\hat{f}_{i+\frac{1}{2}}^L = \hat{f}_{i+\frac{1}{2}}^+$. In the evaluation of $\hat{f}_{i+\frac{1}{2}}^R$, $\hat{f}_{i+\frac{1}{2}}^-$ is evaluated normally and $\hat{f}_{i+\frac{1}{2}}^+ = 0$. Thus $\hat{f}_{i+\frac{1}{2}}^R = \hat{f}_{i+\frac{1}{2}}^-$.

One- and multi-space-dimensional system. This procedure is performed for all components of all flux functions f , g and h since the spatial discretization is done by dimensional splitting. First the flux functions are transformed to the characteristic variables, then the ENO scheme with Marquina's flux splitting and entropy fix is applied to all components, and then the cell boundary values are transformed back to the physical variables.

3.1.3 Convex ENO (CNO) schemes

Convex ENO schemes (Liu and Osher [16]) use a reference flux, typically second order local Lax-Friedrichs, and choose the convex combination of the interpolation values of the different candidate stencils which is nearest to the reference flux. No transformation to the characteristic variables and hence no evaluation of the Jacobians of the flux functions is necessary.

(Local) Lax-Friedrichs flux splitting. Consider the one-dimensional scalar conservation law $q_t + [f(q)]_x = 0$. An example for a conservative scheme of the form

$$q_j^{n+1} = q_j^n - \frac{\Delta t}{\Delta x} \left(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}} \right) \quad (3.18)$$

with a consistent numerical flux

$$\hat{f}_{j+\frac{1}{2}} = \hat{f}(q_{j-l}, \dots, q_{j+k}), \quad \hat{f}(q, \dots, q) = f(q)$$

is the first order monotone Lax-Friedrichs central scheme

$$f^+(q) = \frac{1}{2}(f(q) + \alpha q), \quad f^-(q) = \frac{1}{2}(f(q) - \alpha q) \quad (3.19)$$

where $\alpha \geq \max |f'(q)|$. Thus:

$$\begin{aligned} f^{+'}(q) &\geq 0, \quad f^{-'}(q) \leq 0 \\ f^+(q) + f^-(q) &= f(q) \end{aligned}$$

The Lax-Friedrichs scheme then is (3.18) with:

$$\hat{f}_{j+\frac{1}{2}}^{LF} = f_j^+ + f_{j+1}^- = f^+(q_j) + f^-(q_{j+1}). \quad (3.20)$$

An alternative, less dissipative type is obtained by

$$f_{j+\frac{1}{2}}^+(q) = \frac{1}{2}(f(q) + \alpha_{j+\frac{1}{2}} q), \quad f_{j+\frac{1}{2}}^-(q) = \frac{1}{2}(f(q) - \alpha_{j+\frac{1}{2}} q) \quad (3.21)$$

where $\alpha_{j+\frac{1}{2}} = \max_{\min(q_j, q_{j+1}) \leq q \leq \max(q_j, q_{j+1})} |f'(q)|$. Then the local Lax-Friedrichs scheme is defined to be (3.18) with

$$\hat{f}_{j+\frac{1}{2}}^{LLF} = f_{j+\frac{1}{2}}^+(q_j) + f_{j+\frac{1}{2}}^-(q_{j+1}) \quad (3.22)$$

which is also a monotone scheme.

Reference flux. The standard second order ENO local Lax-Friedrichs flux is defined by (Shu and Osher [30], Liu and Osher [16])

$$\begin{aligned} \hat{f}_{j+\frac{1}{2}}^{LLF,2} &= \frac{1}{2} \left(f(q_{j+1}) + f(q_j) - \alpha_{j+\frac{1}{2}}(q_{j+1} - q_j) \right) \\ &+ \frac{1}{4} \text{m} \left[\Delta_+ f(q_j) + \alpha_{j+\frac{1}{2}} \Delta_+ q_j, \Delta_- f(q_j) + \alpha_{j+\frac{1}{2}} \Delta_- q_j \right] \\ &- \frac{1}{4} \text{m} \left[\Delta_+ f(q_{j+1}) - \alpha_{j+\frac{1}{2}} \Delta_+ q_{j+1}, \Delta_- f(q_{j+1}) - \alpha_{j+\frac{1}{2}} \Delta_- q_{j+1} \right] \end{aligned} \quad (3.23)$$

where

$$\Delta_{\pm} p_j = \pm(p_{j\pm 1} - p_j)$$

and

$$\text{m}(x, y) = \begin{cases} x & \text{if } |x| \leq |y| \\ y & \text{otherwise} \end{cases}$$

$\hat{f}_{j+\frac{1}{2}}^{LLF,2}$ is the sum of $\hat{f}_{j+\frac{1}{2}}^{LLF,2,+}$ and $\hat{f}_{j+\frac{1}{2}}^{LLF,2,-}$ where

$$\begin{aligned} \hat{f}_{j+\frac{1}{2}}^{LLF,2,+} &= \frac{1}{2} \left(f(q_j) + \alpha_{j+\frac{1}{2}} q_j \right) \\ &+ \frac{1}{4} \text{m} \left[\Delta_+ f(q_j) + \alpha_{j+\frac{1}{2}} \Delta_+ q_j, \Delta_- f(q_j) + \alpha_{j+\frac{1}{2}} \Delta_- q_j \right] \\ \hat{f}_{j+\frac{1}{2}}^{LLF,2,-} &= \frac{1}{2} \left(f(q_{j+1}) - \alpha_{j+\frac{1}{2}} q_{j+1} \right) \\ &- \frac{1}{4} \text{m} \left[\Delta_+ f(q_{j+1}) - \alpha_{j+\frac{1}{2}} \Delta_+ q_{j+1}, \Delta_- f(q_{j+1}) - \alpha_{j+\frac{1}{2}} \Delta_- q_{j+1} \right] \end{aligned}$$

The reference flux used in the ANTARES code is exactly the same as (3.23) except that the function m is replaced by the minmod limiter

$$\text{mm}(x, y) = \begin{cases} (\text{sign } x) \min(|x|, |y|), & \text{if } xy > 0 \\ 0 & \text{otherwise} \end{cases}$$

The minmod limiter selects the one which is smaller in magnitude if x and y have the same sign, else it returns zero.

CNO algorithm. The CNO algorithm used in the ANTARES code is a slightly modified version of the algorithm proposed by Liu and Osher [16].

Given the cell averages $\{\bar{v}_j\}$ of a function $v(x)$, we obtain for each cell I_j approximations to the function $v(x)$ at the cell boundaries in the following way:

- Calculate the k approximations $f_{j+\frac{1}{2}}^{+(r)}$ and $f_{j+\frac{1}{2}}^{-(r)}$ of order k for $r = 0, \dots, k-1$ for the local Lax-Friedrichs fluxes $f_{j+\frac{1}{2}}^+(q_j)$ and $f_{j+\frac{1}{2}}^-(q_{j+1})$ using the reconstruction for the k -th order ENO scheme. For all simulations presented here $\alpha_{j+\frac{1}{2}} = \frac{3}{2} c_{\text{sound}j+\frac{1}{2}}$.
- Calculation of $\hat{f}_{j+\frac{1}{2}}^+$:
 - If $\text{sign}(f_{j+\frac{1}{2}}^{+(r)} - \hat{f}_{j+\frac{1}{2}}^{LLF,2,+})$ for $r = 0, \dots, k-1$ are different then $\hat{f}_{j+\frac{1}{2}}^+ = \hat{f}_{j+\frac{1}{2}}^{LLF,2,+}$.
 - Otherwise choose the $f_{j+\frac{1}{2}}^{+(r)}$ which is nearest to $\hat{f}_{j+\frac{1}{2}}^{LLF,2,+}$, i.e. take the $f_{j+\frac{1}{2}}^{+(r)}$ which minimizes $|f_{j+\frac{1}{2}}^{+(r)} - \hat{f}_{j+\frac{1}{2}}^{LLF,2,+}|$.

- Calculation of $\hat{f}_{j+\frac{1}{2}}^-$:
 - If $\text{sign}(f_{j+\frac{1}{2}}^{-(r)} - \hat{f}_{j+\frac{1}{2}}^{\text{LLF},2,-})$ for $r = 0, \dots, k-1$ are different then $\hat{f}_{j+\frac{1}{2}}^- = \hat{f}_{j+\frac{1}{2}}^{\text{LLF},2,-}$.
 - Otherwise choose the $f_{j+\frac{1}{2}}^{-(r)}$ which is nearest to $\hat{f}_{j+\frac{1}{2}}^{\text{LLF},2,-}$, i.e. take the $f_{j+\frac{1}{2}}^{-(r)}$ which minimizes $|f_{j+\frac{1}{2}}^{-(r)} - \hat{f}_{j+\frac{1}{2}}^{\text{LLF},2,-}|$.
- The cell boundary value is then $\hat{f}_{j+\frac{1}{2}} = \hat{f}_{j+\frac{1}{2}}^+ + \hat{f}_{j+\frac{1}{2}}^-$.

One- and multi-space-dimensional system. Similar to the ENO- and WENO-algorithm for one-dimensional systems the CNO Algorithm is performed component-wisely and for multi-space dimensions the one-dimensional algorithm for systems is performed for the flux function f in x-direction, for the flux function g in y-direction and for the flux function h in z-direction.

3.1.4 Discussion

To show the main difference between the fifth order WENO scheme, the third order ENO scheme with Marquina's flux splitting and entropy fix and a fourth order CNO scheme the one-dimensional version of the system of conservations laws (3.1) with the following initial conditions is considered:

$$\mathbf{q}_l = \begin{bmatrix} \rho_l \\ u_l \\ T_l \end{bmatrix} = \begin{bmatrix} 5.0 \cdot 10^{-7} \\ 0 \\ 11000 \end{bmatrix} \quad \text{for } x \leq \frac{x_{\max}}{2} \quad (3.24)$$

$$\mathbf{q}_r = \begin{bmatrix} \rho_r \\ u_r \\ T_r \end{bmatrix} = \begin{bmatrix} 3.0 \cdot 10^{-7} \\ 0 \\ 9000 \end{bmatrix} \quad \text{for } x > \frac{x_{\max}}{2} \quad (3.25)$$

This state at 60 spatial grid points is evolved in time: 100 second order Runge-Kutta steps with $\Delta t = 0.7s$ are performed. The resulting state is very similar for all this three high resolution methods (figure 3.1).

A closer look at some regions with steep gradients shows some differences between the three high resolution methods. The WENO scheme produces the steepest gradient. The CNO scheme is the most smearing one because the calculated flux is nearest to a local Lax-Friedrichs flux. The ENO scheme with Marquina's flux splitting and entropy fix is between the two other in regions of steep gradients (figures 3.2, 3.3). The two different ENO schemes show a different behavior since the weighted ENO scheme is fifth order accurate and the ENO scheme with Marquina's flux splitting and entropy fix is third order accurate.

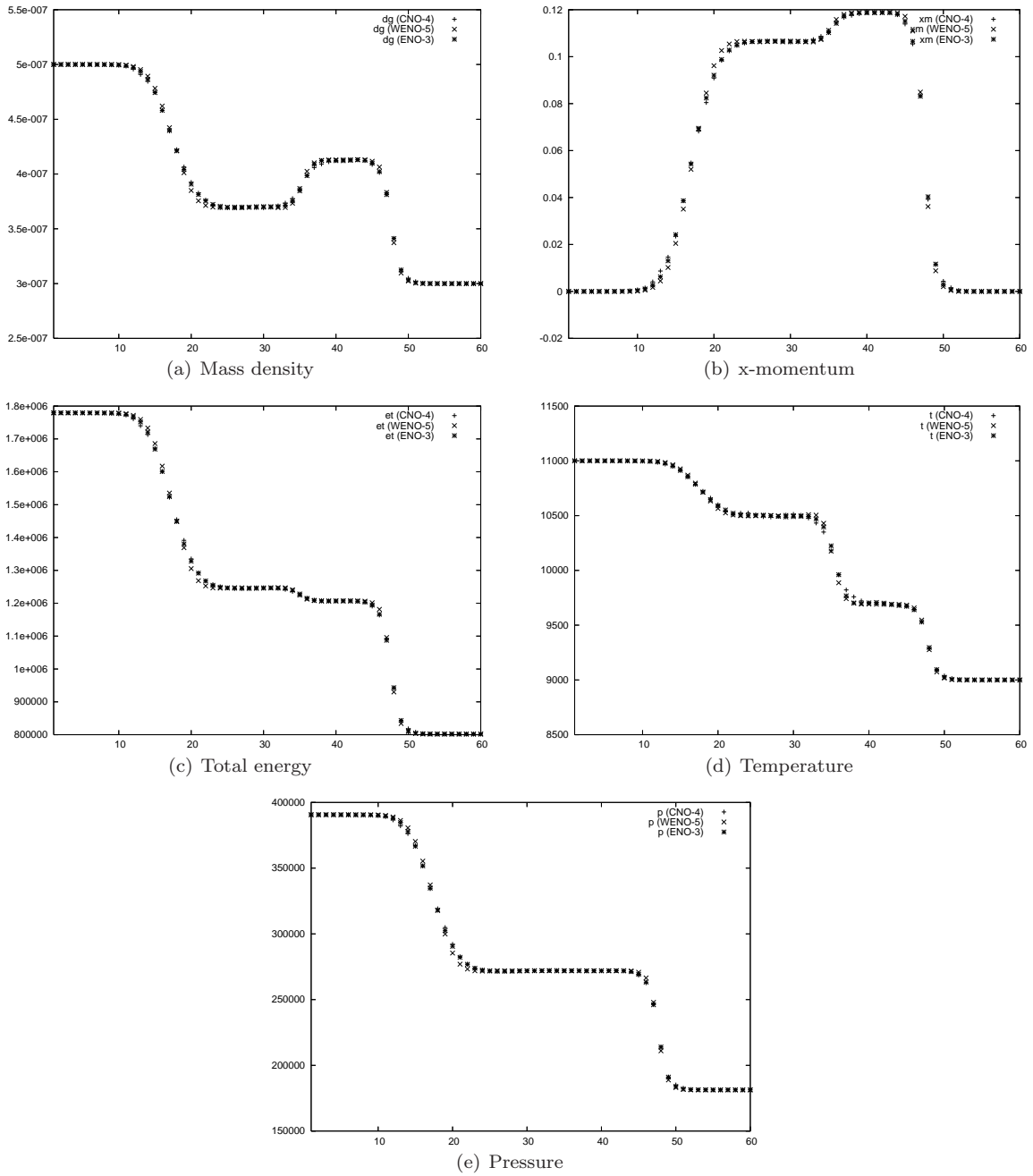
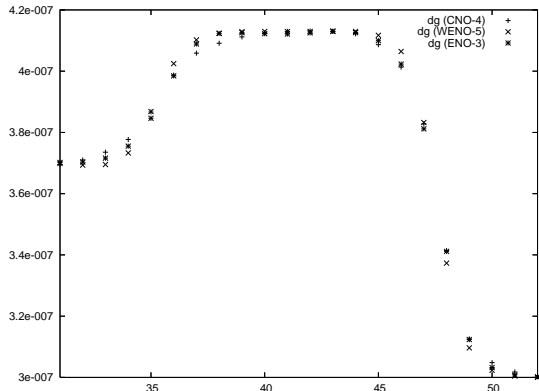
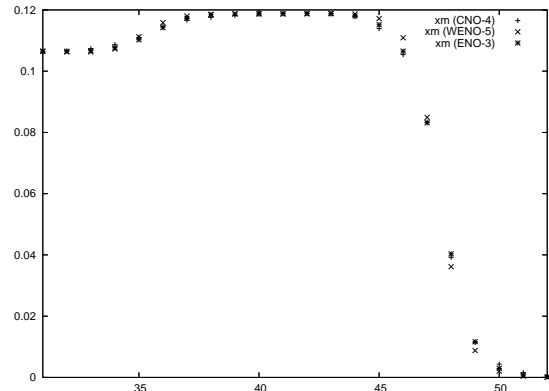


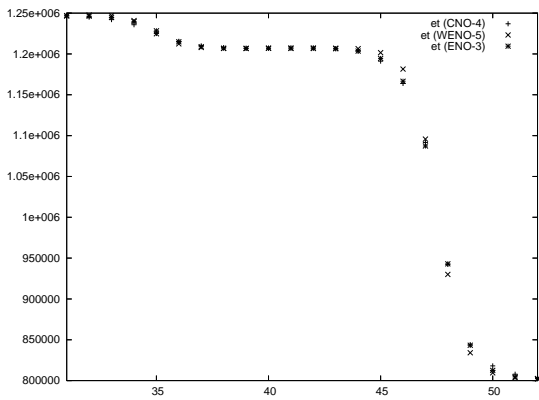
Figure 3.1: Comparison of numerical schemes for conservation laws.



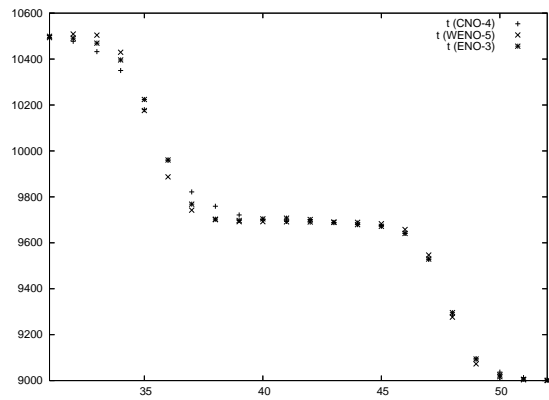
(a) Mass density



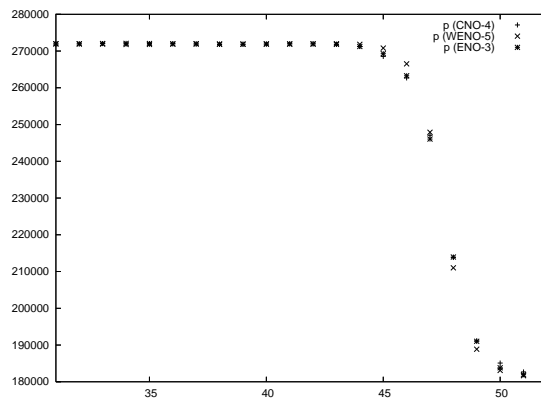
(b) x-momentum



(c) Total energy

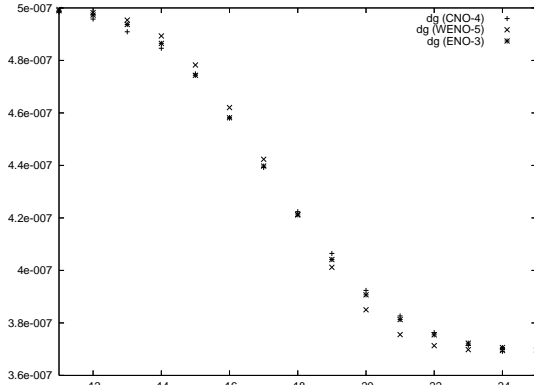


(d) Temperature

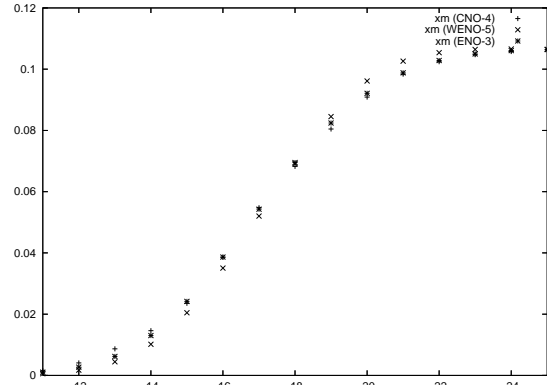


(e) Pressure

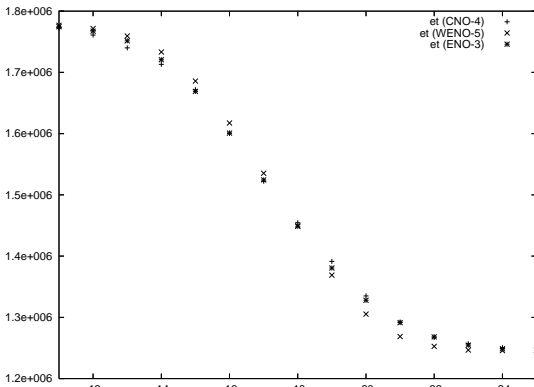
Figure 3.2: Comparison of numerical schemes for conservation laws - region with steep gradient 1.



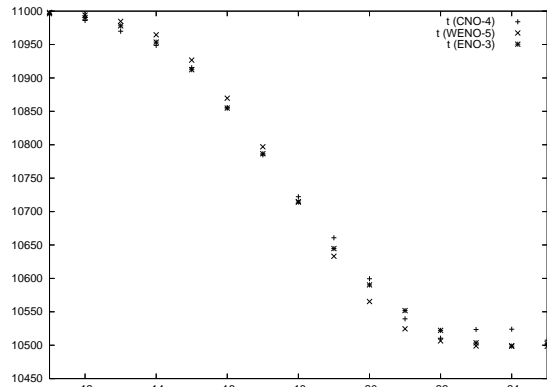
(a) Mass density



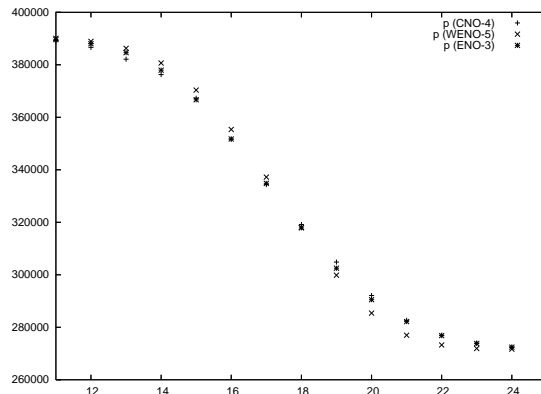
(b) x-momentum



(c) Total energy



(d) Temperature



(e) Pressure

Figure 3.3: Comparison of numerical schemes for conservation laws - region with steep gradient 2.

3.2 Discretization of the viscous fluxes

3.2.1 Viscous hydrodynamical fluxes

The derivatives in the viscous stress tensor $\underline{\tau}$ are calculated by the fourth order scheme

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{1}{12 \Delta x} (u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}).$$

Then the values for $\underline{\tau}$ and $(\mathbf{u} \cdot \underline{\tau})$ at the cell centers in the momentum and energy equations are interpolated to the cell boundaries by the following scheme

$$f_{i+\frac{1}{2}} = \frac{1}{16} (-f_{i-1} + 9f_i + 9f_{i+1} - f_{i+2})$$

to get the the contribution by $\nabla \cdot \underline{\tau}$ and $\nabla \cdot (\mathbf{u} \cdot \underline{\tau})$ through

$$\frac{1}{\Delta x} (f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}).$$

3.2.2 Artificial diffusivities

Artificial diffusivities (Stein and Nordlund [32], Caunt and Korpi [5]) remove short-wavelength noise without damping longer wavelengths and diffuse strong discontinuities in order to stabilize the numerical code.

The viscous stress tensor

$$\tau_{kl} = \mu \left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k} - \frac{2}{3} \delta_{kl} (\nabla \cdot \mathbf{u}) \right) \quad (3.26)$$

is replaced by artificial equivalents of the form

$$\tau_{kl} = \frac{1}{2} \rho \left(\nu_k(u_l) \frac{\partial u_l}{\partial x_k} + \nu_l(u_k) \frac{\partial u_k}{\partial x_l} \right) \quad (3.27)$$

where $\mathbf{u} = (u_1, u_2, u_3)^T$ and $k, l = 1, 2, 3$.

The coefficients ν_k for direction k consists of two parts, a shock resolving ν_k^{shk} and a hyperdiffusive ν_k^{hyp} :

$$\nu_k(u_l) = \nu_k^{\text{shk}} + \nu_k^{\text{hyp}}(u_l) \quad (3.28)$$

The shock resolving part in direction k is defined by

$$\nu_k^{\text{shk}} = \begin{cases} C_{\text{shk}} \Delta x_k^2 |\nabla \cdot \mathbf{u}| & \nabla \cdot \mathbf{u} < 0 \\ 0 & \nabla \cdot \mathbf{u} \geq 0 \end{cases} \quad (3.29)$$

which is only applied in regions undergoing compression, i.e. where $\nabla \cdot \mathbf{u} < 0$. The hyperdiffusive part in direction k is defined by

$$\nu_k^{\text{hyp}}(f) = C_{\text{hyp}} \Delta x_i c_{\text{tot}} \frac{\max_3 |\Delta_k^3 f|}{\max_3 |\Delta_k^1 f|} \quad (3.30)$$

where

$$c_{\text{tot}} = \|\mathbf{u}\| + c_{\text{sound}}$$

and

$$\begin{aligned} (\Delta_k^3 f)_i &= |3(u_{i+1} - u_i) - (u_{i+2} - u_{i-1})| \\ (\Delta_k^1 f)_i &= |u_{i+1} - u_i|. \end{aligned}$$

For each grid point the index i runs in direction k . For all simulations presented here $C_{\text{shk}} = 1$ and $C_{\text{hyp}} = 0.05$.

3.3 Interpolation tool

In the radiative transfer solver (section 3.4) and in the implementation of the equation of state (section 2.2.6) numerous interpolations are performed. In this section the basic interpolation procedure used in the ANTARES code is described.

For one-dimensional interpolations using four points, the approximations at point P of two interpolation polynomials with different stencils (see figure 3.4) are combined by a weighted sum. The weight for polynomial 1 with stencil 1 is $\frac{d_2}{d_1+d_2}$ and for polynomial 2 with stencil 2 is $\frac{d_1}{d_1+d_2}$.

For the two-dimensional interpolations, 4×4 grid points are used to get the interpolation values. First four one-dimensional interpolations in direction 1 and then one one-dimensional interpolation in direction 2 are performed to get the value at point P . The successive interpolation procedure is illustrated in figure 3.5.

For three-dimensional interpolations $4 \times 4 \times 4$ grid points are used and the procedure described above is performed starting with 16 interpolations in direction 1, followed by 4 interpolations in direction 2 and then one interpolation in direction 3.

In the radiative transfer solver a method for monotonic interpolation in one dimension proposed by Steffen [31] is applied which assures that the values for density, opacity, intensity and radiative source function always are positive.

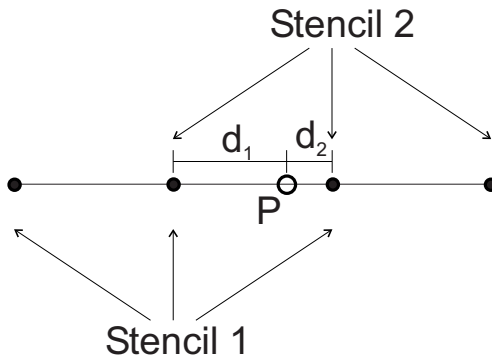


Figure 3.4: Stencils for one-dimensional interpolation.

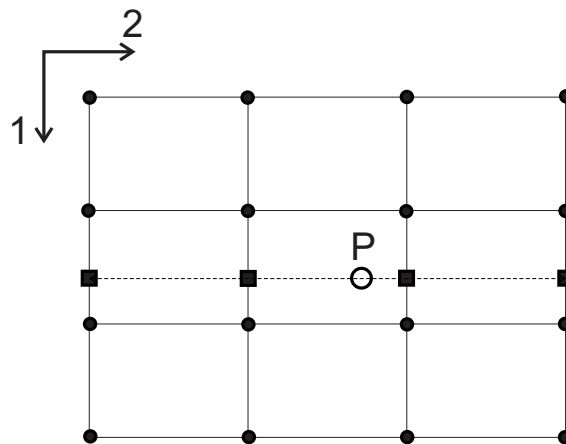


Figure 3.5: Two-dimensional interpolation procedure.

3.4 Numerical radiative transfer

Following the short characteristic approach (Mihalas et al. [22], Kunasz and Auer [13]), instead of solving the three-dimensional RTE (1.6) the one-dimensional RTE (1.7) with $r = 1$ is solved along several ray directions.

Boundary conditions. To solve the RTE in the computational domain boundary conditions for incoming radiation must be specified.

All quantities are assumed to be periodic in all horizontal directions. Where periodic radiative boundary conditions for time step $n + 1$ are applied the solution for the intensity at time step n is used.

At the top of the computational domain it is assumed that there is no incoming radiation, i.e. $I_{\text{rad}}(\mathbf{x}, \mathbf{r})|_{\text{top}} = 0 \quad \forall \mathbf{r}_1 > 0$.

At the bottom of the computation domain the diffusion approximation $I_{\text{rad}}(\mathbf{x}, \mathbf{r})|_{\text{bot}} = B_\nu(\mathbf{x})$ is valid. The diffusion approximation is even valid at optical depths greater than 10. Therefore the lower boundary condition is applied at a fixed optical depth $\tau_{\text{DA}} > 10$ (calculated with Rosseland mean opacities). In regions with optical depth greater than τ_{DA} the radiative heating rate Q_{rad} is calculated according to the diffusion approximation $Q_{\text{rad}} = \nabla \cdot (\kappa \nabla T)$.

3.4.1 Numerical Scheme

To apply the short characteristic algorithm N_{rays} ray directions are specified along which for every grid point $x_{i,j,k}$ the radiative transfer equation is solved. In this section only grey radiative transfer is considered. $S = B(T) = \frac{\sigma}{\pi} T^4$ is the frequency-independent Planck function (σ is the Stefan-Boltzmann constant), χ is the Rosseland mean opacity and τ is the corresponding optical depth.

Long characteristic method. The intensity $I(\tau_P)$ along ray direction \mathbf{r} can be calculated along long characteristics where the boundary values $I(\tau_U)$ and τ_U are known and the right hand side of equation

$$I(\tau_P) = I(\tau_U)e^{\tau_U - \tau_P} + \int_{\tau_U}^{\tau_P} S(t)e^{-(\tau_P - t)} dt \quad (3.31)$$

can be approximated (see figure 3.6).

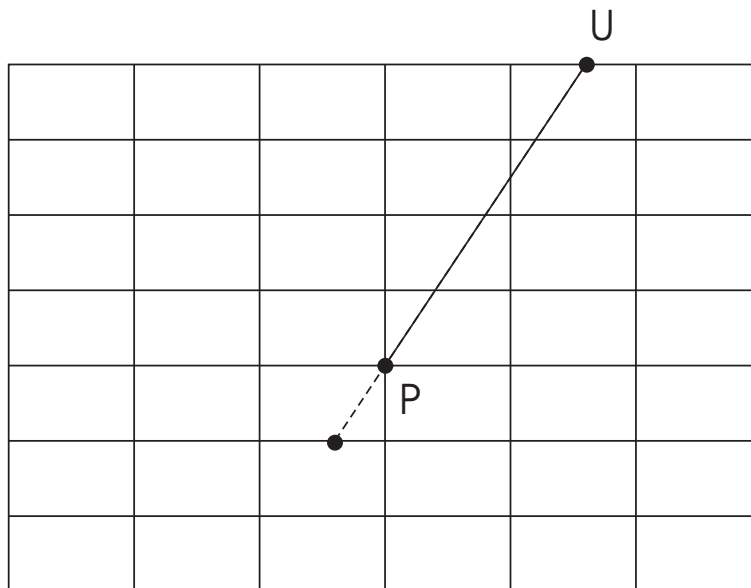


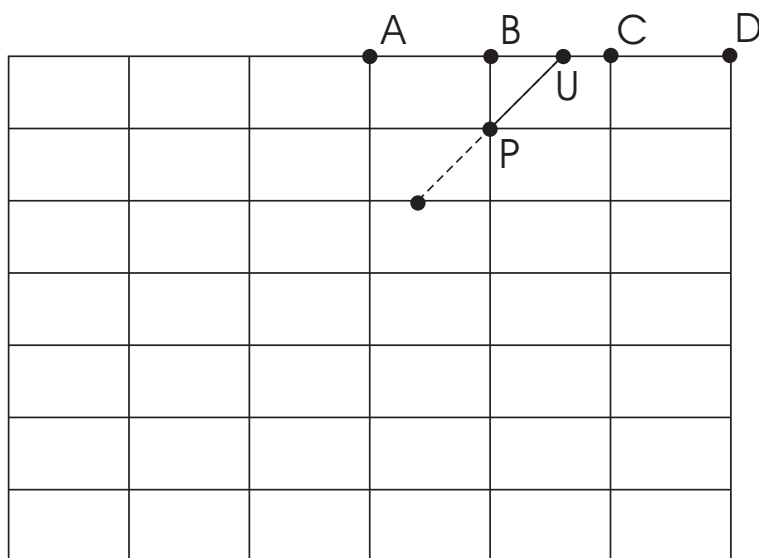
Figure 3.6: Long characteristic method.

Short characteristic method. More efficient and accurate is the short characteristic method (Kunasz and Auer [13]). At every grid point $x_{i,j,k}$ along each of the N_{rays} ray directions information is transported locally in x-, y- or z-direction.

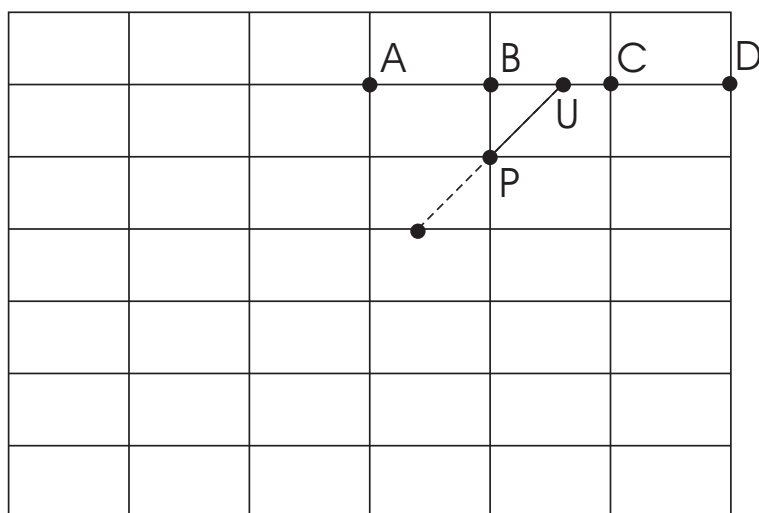
For incoming radiation, at the boundaries the values for $I(\tau_U)$ along ray direction \mathbf{r} are known. The optical depth τ_U there is set to zero. To determine $I(\tau_P)$ along this ray direction \mathbf{r} (see figure 3.7) one interpolates the values of the required physical quantities to point U using 16 points in 3D respectively 4 points in 2D. Then one evaluates equation

$$I(\tau_P) = I(\tau_U)e^{\tau_U - \tau_P} + \int_{\tau_U}^{\tau_P} S(t)e^{-(\tau_P - t)} dt \quad (3.32)$$

numerically to get $I(\tau_P)$. This procedure is repeated recursively since after step 1 the intensities for the first layer are determined (see figure 3.7).



(a) step 1



(b) step 2

Figure 3.7: Short characteristic method.

Numerical integration. To perform the numerical integration on the right hand side of (3.32) one has to determine τ_P and $\tau_{P+\overrightarrow{UP}}$. (Note that $\tau_U = 0$.) The optical depth at point P is defined as $\tau_P = \int_U^P \rho\chi dx$, at point $P + \overrightarrow{UP}$ is defined as $\tau_{P+\overrightarrow{UP}} = \int_U^{P+\overrightarrow{UP}} \rho\chi dx$.

Constructing a quadratic interpolation polynomial through the three points $(U, \rho(U)\chi(U))$, $(P, \rho(P)\chi(P))$ and $(P + \overrightarrow{UP}, \rho(P + \overrightarrow{UP})\chi(P + \overrightarrow{UP}))$ and performing an exact integration of this interpolation polynomial one gets approximations for τ_P and $\tau_{P+\overrightarrow{UP}}$.

A more stable but less accurate approximation is calculated by

$$\begin{aligned}\tau_P &\approx \frac{1}{2}(\rho(U)\chi(U) + \rho(P)\chi(P)) \cdot |\overrightarrow{UP}|, \\ \tau_{P+\overrightarrow{UP}} &\approx \tau_P + \frac{1}{2}(\rho(P)\chi(P) + \rho(P + \overrightarrow{UP})\chi(P + \overrightarrow{UP})) \cdot |\overrightarrow{UP}|.\end{aligned}$$

Here, an integral is twice evaluated numerically by the trapezoidal rule.

The integration on the right hand side of (3.32) is performed by a quadrature rule proposed by Olson and Kunasz [25]. Let τ_1 be the optical depth along the path from U to P and let τ_2 be the optical depth along the path from P to $P + \overrightarrow{UP}$. A numerical approximation to the integral $\int_{\tau_U}^{\tau_P} S(t)e^{-(\tau_P-t)} dt$ is given by

$$\int_{\tau_U}^{\tau_P} S(t)e^{-(\tau_P-t)} dt \approx \alpha S(U) + \beta S(P) + \gamma S(P + \overrightarrow{UP}) \quad (3.33)$$

where

$$\begin{aligned}e_0 &= 1 - e^{\tau_1} \\ e_1 &= \tau_1 - e_0 \\ e_2 &= \tau_1^2 - 2e_1 \\ \alpha &= \frac{e_0 + (e_2 - (2\tau_1 + \tau_2)e_1)}{\tau_1(\tau_1 + \tau_2)} \\ \beta &= \frac{(\tau_1 + \tau_2)e_1 - e_2}{\tau_1\tau_2} \\ \gamma &= \frac{e_2 - \tau_1 e_1}{\tau_2(\tau_2 + \tau_1)}.\end{aligned}$$

Interpolation. The values for the physical quantities at point U are interpolated from the neighboring grid points. In the two-dimensional case 4 grid points ($A = (x_{i-1}, y_{i-1})$, $B = (x_i, y_i)$, $C = (x_{i+1}, y_{i+1})$, $D = (x_{i+2}, y_{i+2})$) are used. A simple method for monotonic interpolation in one dimension proposed by Steffen [31] is applied to calculate the approximation at point U . The method gives exact results if the data points correspond to a second order polynomial. The application of this method assures that the values for density, opacity, intensity and radiative source function always are positive.

In the three-dimensional case 4×4 grid points are used for the interpolation at point U . First, four one-dimensional (in direction 1) interpolations are performed and then a one-dimensional interpolation with the results of the previous four interpolations is performed in direction 2. The interpolation procedure is illustrated in section 3.3.

Monotonic interpolation. The interpolation function in the interval (x_i, x_{i+1}) reads

$$f_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i. \quad (3.34)$$

If at each grid point x_i the value y_i and the slope y'_i are known the coefficients a_i, b_i, c_i, d_i are uniquely determined:

$$\begin{aligned}
a_i &= \frac{y'_i + y'_{i+1} - 2s_i}{\Delta x^2} \\
b_i &= \frac{3s_i - 2y'_i - y'_{i+1}}{\Delta x} \\
c_i &= y'_i \\
d_i &= y_i
\end{aligned} \tag{3.35}$$

$s_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$ is the slope of the secant through the points (x_i, y_i) and (x_{i+1}, y_{i+1}) .

This piecewise cubic function (Hermite-type interpolation) constructed from given y_i and y'_i automatically has a continuous first order derivative over the whole data set. To get monotonic behavior of the interpolation curve a special calculation of the first order derivative is necessary. Steffen [31] proposes the choice

$$y'_i = \begin{cases} 0 & \text{if } s_{i-1}s_i \leq 0 \\ 2a \min(|s_{i-1}|, |s_i|), a = \text{sign}(s_{i+1}) = \text{sign}(s_i) & \text{if } |p_i| > 2|s_{i-1}| \text{ or } |p_i| > 2|s_i| \\ p_i & \text{otherwise.} \end{cases} \tag{3.36}$$

$p_i = \frac{s_{i-1}h_i + s_i h_{i-1}}{h_{i-1} + h_i}$ is the slope at the internal point x_i from the unique parabola passing through the points (x_{i-1}, y_{i-1}) , (x_i, y_i) and (x_{i+1}, y_{i+1}) (Bessel's method), which for an equidistant x -grid is the same as the slope of the secant through the points (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}) .

The slope at x_i must not be greater in absolute value than twice the minimum of the absolute value of the slopes given by the left- and right-handed finite differences. If these have different signs, y'_i must be zero.

Solid angle integration. For every grid point the 1D RTE is solved along N_{rays} ray directions. To get the mean intensity J an integration over the solid angle is performed.

In the two-dimensional case the RTE is solved along 20 ray directions. These are chosen according to a Gaussian quadrature rule

$$\int_{-1}^1 f(x) dx \approx a_1 f(x_1) + \dots + a_N f(x_N) \tag{3.37}$$

for $N = 10$ since the two-dimensional integral $J = \frac{1}{4\pi} \int I(\mathbf{r}) d\omega$ can be considered as

$$J = \frac{1}{4\pi} \left(\int_{\mathbf{r}_2 \geq 0} I(\mathbf{r}) d\omega + \int_{\mathbf{r}_2 < 0} I(\mathbf{r}) d\omega \right). \tag{3.38}$$

Each of these two integrals can be calculated by a Gaussian quadrature rule where x_i is the first component of the i -th direction \mathbf{r}^i (see figure 3.8). The third component is zero, the second is determined such that all \mathbf{r}^i have length one. The weights a_i must be multiplied by π since $\int_{\mathbf{r}_2 \geq 0} d\omega + \int_{\mathbf{r}_2 < 0} d\omega$ is required to be 4π and each of these two integrals over a function equal one (i.e. $\int_{\mathbf{r}_2 \geq 0} d\omega$) corresponds to an integral evaluated by the Gaussian quadrature rule with $f(x_i) = 1 \forall i = 1, \dots, 10$ resulting $\int_{-1}^1 dx = 2$.

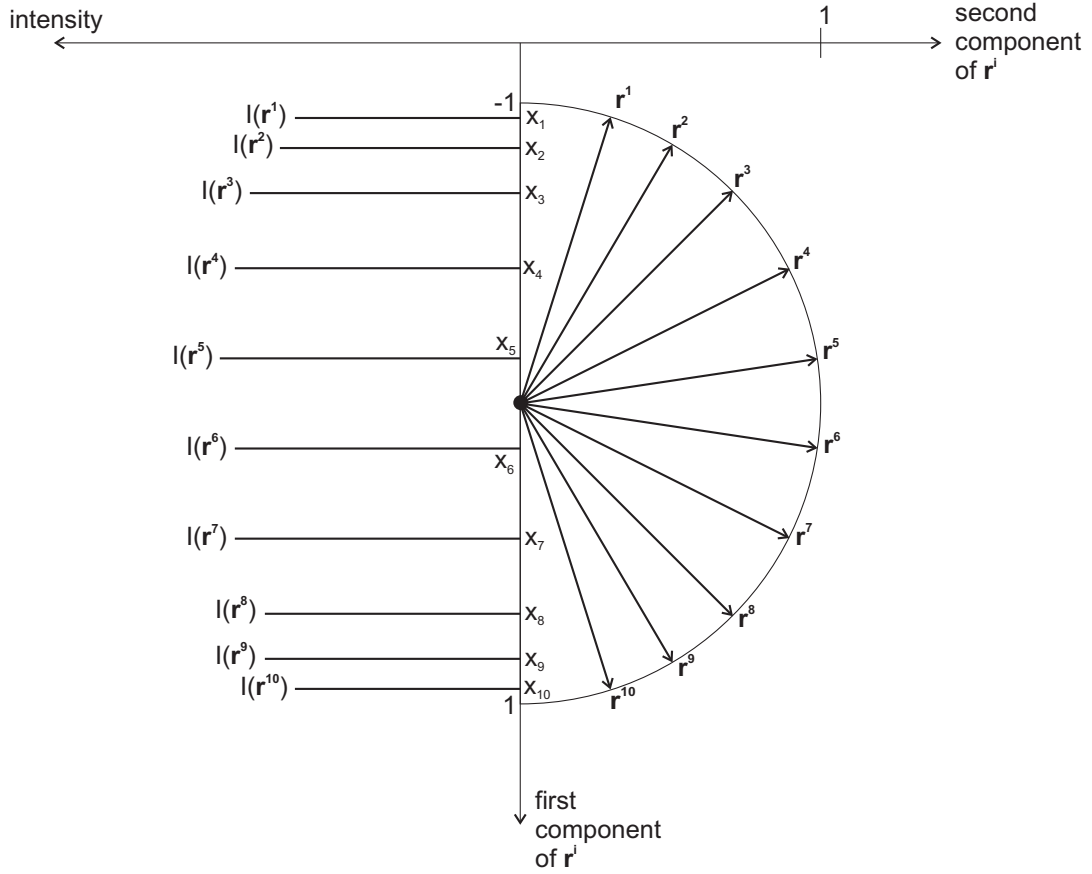


Figure 3.8: Interpretation of the solid angle integration at a fixed grid point in (3.38) with $r_2 \geq 0$ as Gaussian quadrature rule. The right hand side shows the ray directions \mathbf{r}^i in the two-dimensional computational domain. The left hand side shows the calculated intensities depending on x_i , the first components of the ray directions.

The support points x_i and the weights a_i for the Gaussian quadrature rule for $N = 10$ (Lowan, Davids and Levenson [18]) are:

i	x_i	a_i
1	-0.974	0.067
2	-0.865	0.150
3	-0.680	0.219
4	-0.433	0.270
5	-0.149	0.296
6	0.149	0.296
7	0.433	0.270
8	0.680	0.219
9	0.865	0.150
10	0.974	0.067

With these choices the two-dimensional solid angle is provided with 20 ray directions.

For three-dimensional simulations the 1D RTE is solved along 24 ray directions which are chosen according to the angular quadrature formulae of type A of Carlson [4]. The directions in each octant are arranged in a triangular pattern and the quadrature is invariant under rotations over multiples of $\pi/2$ around any

coordinate axis. A summary of the construction procedure is given in Bruls, Vollmöller and Schüssler [3]. For the three-dimensional simulations the A4 quadrature set with three directions per octant is used.

For this choice each ray has got the weight $\omega_m = \frac{1}{24}$. The ray directions in the first octant are calculated as

$$\mathbf{r}^1 = \begin{pmatrix} \sqrt{\frac{7}{9}} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}, \quad \mathbf{r}^2 = \begin{pmatrix} \frac{1}{3} \\ \sqrt{\frac{7}{9}} \\ \frac{1}{3} \end{pmatrix}, \quad \mathbf{r}^3 = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \sqrt{\frac{7}{9}} \end{pmatrix}.$$

The ray directions in the other octants are obtained by applying three-dimensional rotations into the other octants.

With these quadrature weights $\omega_m = \frac{1}{24}$ and the index m running over the set of directions, the mean intensity is

$$J = \sum_m \omega_m I(\mathbf{r}^m). \quad (3.39)$$

Radiative heating rate. Since J is calculated at each grid point the radiative heating rate is given by

$$Q_{\text{rad}} = 4\pi\rho\chi(J - S). \quad (3.40)$$

Efficient implementation. Radiative transfer is a non-local phenomenon. Simulations with N_p processors are carried out by the domain decomposition approach. To calculate the intensity for a specific ray direction (with the short characteristic method) in a subdomain, the corresponding processor has to wait until the intensity for this ray direction is calculated in the adjacent subdomain from which the ray enters.

For simulations with numerous processors one has to ensure that as many processors as possible are busy. For each ray travelling through the grid point $x_{i,j,k}$ the surface element of the grid cell $I_{i,j,k}$ through which the ray enters is determined. Since the grid is equidistant there is no grid point dependence in this classification. The rays are put into groups described by integer numbers:

1	Rays entering at the surface perpendicular to the x -direction at $x_{i-\frac{1}{2}}$.
-1	Rays entering at the surface perpendicular to the x -direction at $x_{i+\frac{1}{2}}$.
2	Rays entering at the surface perpendicular to the y -direction at $y_{j-\frac{1}{2}}$.
-2	Rays entering at the surface perpendicular to the y -direction at $y_{j+\frac{1}{2}}$.
3	Rays entering at the surface perpendicular to the z -direction at $z_{j-\frac{1}{2}}$.
-3	Rays entering at the surface perpendicular to the z -direction at $z_{j+\frac{1}{2}}$.

I.e. rays with code 1 transport information from the top to the bottom.

In the following the procedure for an efficient implementation of the radiative transfer solver is described for the rays with code 1 and -1. The procedures for the other two directions are similar.

The processors representing the upper half of the computational domain first calculate the intensities for the rays with code 1 in the following way: In the first step the processors representing the first layer from the top calculate the intensity for the first ray with code 1. In the second step these processors calculate the intensity for the second ray with code 1 and the processors representing the second layer from the top calculate the intensity for the first ray with code 1. This procedure is continued until all processors representing the upper half of the computational domain have calculated the intensities for all rays with code 1.

The processors representing the lower half of the computational domain first calculate the intensities for the rays with code -1. The procedure is similar to the one described above, only the start is at the processors representing the first layer from the bottom and then moving forward to the top.

After that, the processors representing the upper half of the computational domain have to continue from

the bottom to the top with the rays with code -1 and the processors representing the lower half of the computational domain have to continue from the top to the bottom with the rays with code 1 in the way described above.

Dividing the x -direction in $N_{p,x}$ parts for the domain decomposition the efficiency of this grouping is for even $N_{p,x}$

$$\frac{\frac{N_{\text{rays},x}}{2}}{\frac{N_{\text{rays},x}}{2} + \left(\frac{N_{p,x}}{2} - 1\right)} \quad (3.41)$$

and for odd $N_{p,x}$

$$\frac{\frac{N_{\text{rays},x}}{2}}{\frac{N_{\text{rays},x}}{2} + \left(\frac{N_{p,x}+1}{2}\right)} \quad (3.42)$$

where $N_{\text{rays},x}$ is the number of rays with code ± 1 .

Algorithm. The algorithm requires the specification of the following variables:

n_p	Number of subdomains in x -direction, $n_p = N_{p,x}$.
n_rays	Number of rays with code ± 1 , $n_rays = N_{\text{rays},x}$. $N_{\text{rays},x}$ is even since every direction is passed in both possible ways.
p	Position of the subdomain in x -direction, $p \in \{1, \dots, n_p\}$.
R(1 : n_rays)	The array R contains the numbers of the rays arranged in the following way: If $p \leq \begin{cases} \frac{n_p}{2} & \text{for even } n_p \\ \frac{n_p+1}{2} & \text{for odd } n_p \end{cases}$ the first half of the array is filled with the numbers of the rays with code 1 and the second half with the numbers of the rays with code -1. Else the first half of the array is filled with the numbers of the rays with code -1 and the second half with the numbers of the rays with code 1.
i_max	Maximal number of steps for the algorithm, $i_max = \begin{cases} n_rays + n_p - 2 & \text{for even } n_p \\ n_rays + n_p - 1 & \text{for odd } n_p \end{cases}$
C(1 : n_p, 1 : i_max)	The rows of the control field C represent the different values for p . The columns represent the steps of the algorithm. In total i_max steps are necessary. Possible entries in this array are 0, 1 and -1. $C(i, j) = \pm 1$ indicates that in step j all processors with $p = i$ calculate the intensity of a ray with code ± 1 . If $C(i, j) = 0$ the processors with $p = i$ pause in step j .

Following the description above, the entries in the control field C for $N_{p,x} = 6$ and $N_{\text{rays},x} = 8$ are

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 & -1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (3.43)$$

and for $N_{p,x} = 5$ and $N_{\text{rays},x} = 8$ the entries in the control field C are

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 \\ 0 & -1 & -1 & -1 & -1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \quad (3.44)$$

After determining the values for the variables n_p , n_rays , p , R , i_max and C the following algorithm is performed to get the intensities at all subdomains using the short characteristic method:

```

counter=1
for i=1,i_max
  if ( C(p,i) = 1 )
    i_ray=R(counter)
    INTENSITY(i_ray) !calculate intensity for ray with number i_ray
                      !for all points in the subdomain starting at the
                      !top of the subdomain and moving downwards

    counter=counter+1
  end if
  if ( C(p,i) = -1 )
    i_ray=R(n_rays)
    INTENSITY(i_ray) !calculate intensity for ray with number i_ray
                      !for all points in the subdomain starting at the
                      !bottom of the subdomain and moving upwards

    counter=counter+1
  end if
  DATA_TRANSFER !put boundary data to the adjacent subdomain
end for

```

First, the radiative transfer equation is solved for all rays with code ± 1 , followed by those with code ± 2 and those with code ± 3 . For each of these three cases the algorithm is arranged as described above. For code ± 2 respectively ± 3 the starting points for the algorithm are the lower and upper boundaries in y -respectively in z -direction instead of the top and the bottom of the computational domain.

Using mesh size $\Delta x = \Delta y = \Delta z$ and dividing the domain into $4 \times 4 \times 4$ subdomains (i.e. performing the simulation on 64 processors) then one gets an efficiency of 80% for the 24 ray directions defined above, omitting the transition to the diffusion approximation and neglecting data transfer between the processors.

3.4.2 Frequency dependent radiative transfer

The frequency dependent radiative transfer which is important in the solar photosphere is implemented by the opacity binning method (Nordlund [24], Ludwig [19]). Frequencies ν which reach optical depth $\tau_\nu = 1$ at similar geometric depth are grouped in the same bin.

Roughly speaking, about $10^6 - 10^7$ frequency points are necessary to model the detailed frequency dependence. Through grouping these frequencies into N_{bins} (4 or 12) frequency bins Ω_i and calculating average opacities $\bar{\chi}_i$, source functions \bar{S}_i and weights ω_i for these bins one can reduce the computational effort to treat non-grey radiative transfer.

The frequencies are grouped in the bins according to their optical depth scale as follows: First a one-dimensional reference atmosphere is chosen. Then $\chi_\nu \rho$ is integrated along x for all frequencies and the geometrical height corresponding to $\tau_\nu = 1$ is calculated. Threshold heights τ_{ref}^i are chosen and the height in terms of a reference optical depth scale τ_{ref} is measured. Then ν belongs to bin Ω_i if $\tau_\nu = 1$ is reached within the corresponding interval of the reference optical depth:

$$\nu \in \Omega_i \text{ if } \tau_{\text{ref}}^{i+1} \geq \tau_{\text{ref}}(\tau_\nu = 1) > \tau_{\text{ref}}^i. \quad (3.45)$$

It is assumed that frequencies which reach optical depth unity at similar geometrical depth have got a similar behavior in the radiation field.

For 4 bins the logarithmic values for the threshold heights τ_{ref}^i are: -0.5, -1.5, -2.5

For 12 bins these values are: 0.25, 0, -0.25, -0.5, -1, -1.5, -2, -2.5, -3, -4, -5

For each frequency set Ω_i Rosseland $\bar{\chi}_{R,i}$ and Planck $\bar{\chi}_{P,i}$ mean opacities are computed for a preset pressure-temperature domain which covers the values occurring during the hydrodynamical simulation. For each pair (T, p) a mean opacity is calculated by

$$\bar{\chi}_i = e^{-\tau_{\text{ref}}^i} \bar{\chi}_{P,i} + (1 - e^{-\tau_{\text{ref}}^i}) \bar{\chi}_{R,i}. \quad (3.46)$$

Thus for every frequency set the opacity is given by the Planck mean in the optically thin and by the Rosseland mean in the optically thick layers.

The radiative transfer equation is solved for each bin Ω_i , resulting $I_i(\mathbf{r})$. Then the mean intensity J_i for each bin is calculated and then the radiative heating rate is given by

$$Q_{\text{rad}} = 4\pi\rho \sum_{i=1}^{N_{\text{bins}}} \omega_i \bar{\chi}_i (J_i - \bar{S}_i). \quad (3.47)$$

3.4.3 2D test problem

To test the radiative transfer solver the radiative heating rate Q_{rad} for different two-dimensional plane-parallel models is determined.

Grey radiative transfer and diffusion approximation. On the one hand, the grey radiative heating rate for a two-dimensional plane-parallel model (see section 2.2.1) with calculated opacities and source function is determined and compared with the result of the MURaM code (Vögler et al. [36]) with a different 1D atmosphere by Maltby et al. [20]. On the other hand, the grey radiative heating rate for the crucial region of the 1D atmosphere given by Maltby et al. [20] is also calculated with the ANTARES code to compare the two different radiative transfer solvers directly.

The first plot in figure 3.9 shows the radiative heating rate as a function of depth for a 1D atmosphere by Maltby et al. [20] calculated with the MURaM code (Vögler [35]) and the second plot shows the horizontal average of the radiative heating rate for the corresponding spatial part using a different atmosphere by Christensen-Dalsgaard et al. [6], calculated with the two-dimensional algorithm of the ANTARES code. Small differences (shape of Q_{rad} and location of the ionization zone) arise from different model atmospheres and differences in the radiative transfer solver and the numerical resolution.

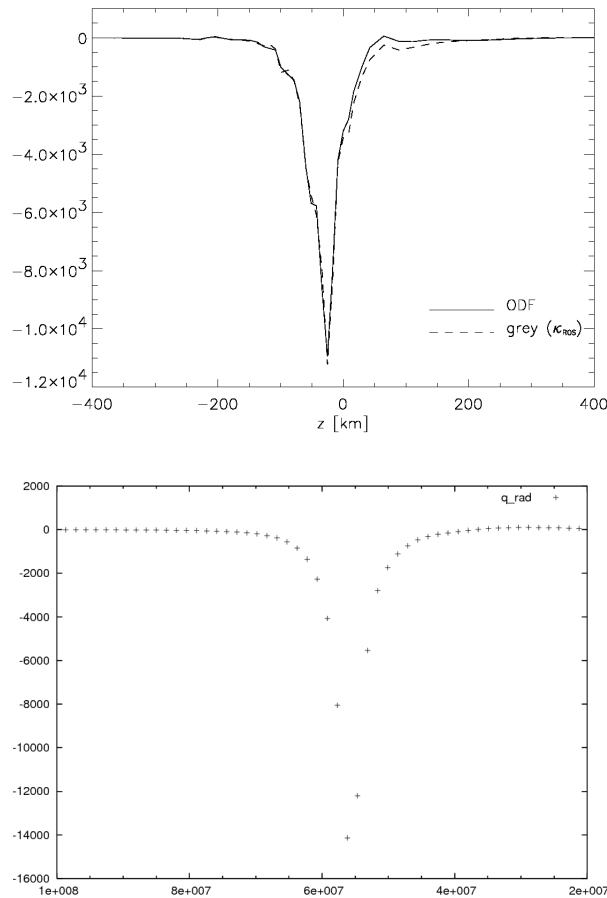


Figure 3.9: Solution for Q_{rad} for different atmospheres.

Figure 3.10 shows the horizontal average of the radiative heating rate for a part of the 1D atmosphere by Maltby et al. [20] calculated with the two-dimensional algorithm of the ANTARES code. This part corresponds to the depth interval $[-75,325]$ km in the first plot of figure 3.9. The shapes of these two grey solutions are very similar.

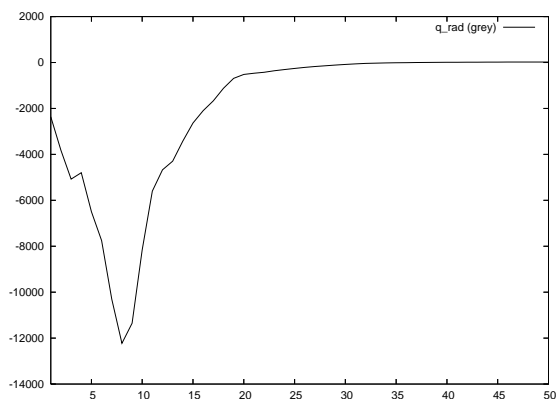


Figure 3.10: Grey solution for Q_{rad} for the crucial region of the atmosphere by Maltby et al. [20].

The whole horizontal average of Q_{rad} for the model atmosphere by Christensen-Dalsgaard et al. [6] and the maximal horizontal error $e_i = \max_j |\bar{Q}_{\text{rad}_i} - Q_{\text{rad}_{i,j}}|$, $\forall i$ are shown in figure 3.11. Comparing the solutions for the model the maximal error is larger near the ionization zone but negligible in magnitude everywhere.

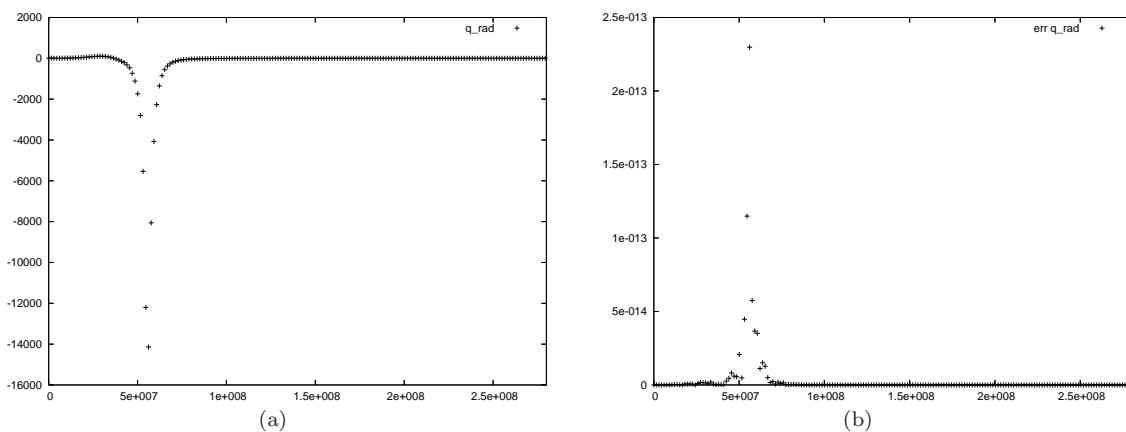


Figure 3.11: Horizontal average of the grey solution for Q_{rad} and the maximal horizontal error.

Figure 3.12 shows the difference between Q_{rad} and the diffusion approximation $\nabla \cdot (\kappa \nabla T)$. Starting at depth 1000km one can change to the diffusion approximation which reduces the computational time markedly.

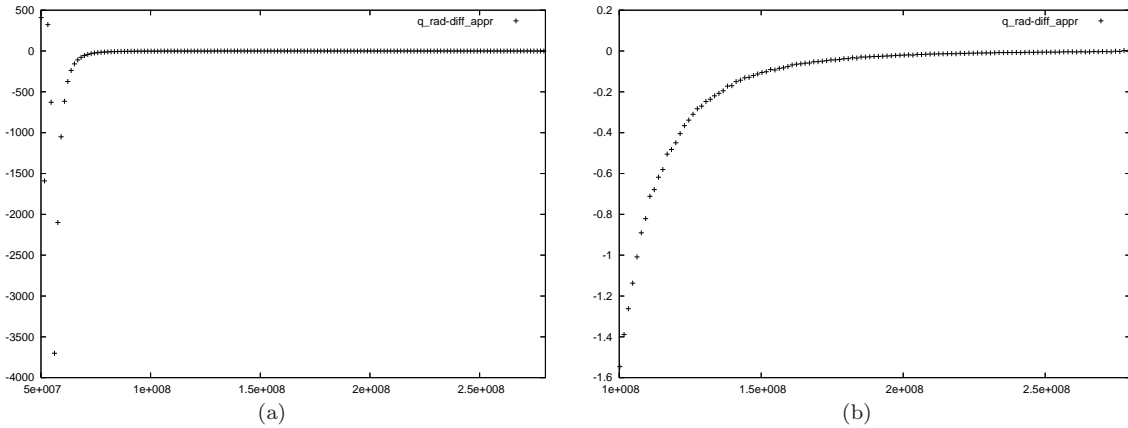
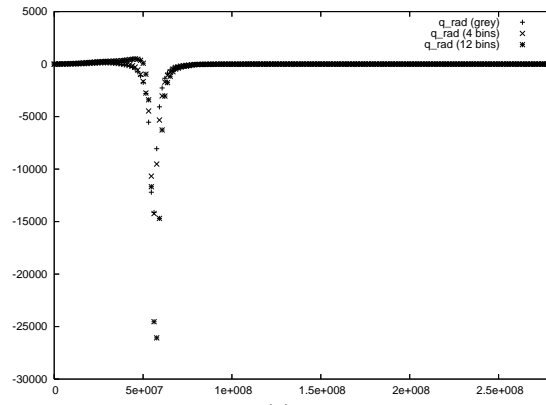


Figure 3.12: Difference between the radiative heating rate Q_{rad} and the diffusion approximation for $x \geq 500\text{km}$ (a) and $x \geq 1000\text{km}$ (b).

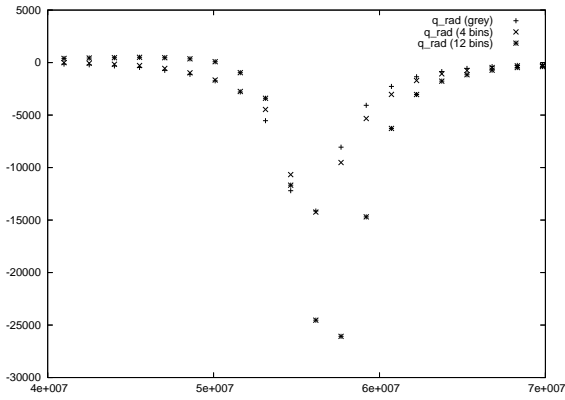
Non-grey radiative transfer. Figure 3.13 shows the differences between grey and non-grey radiative transfer. In the surrounding of the ionization zone differences are obvious.

All three solutions (grey, non-grey with 4 and 12 bins) for the radiative heating rate show a different behavior. Between 300 and 600km especially the non-grey solution with 12 bins leads to a back-warming effect relative to the two others.

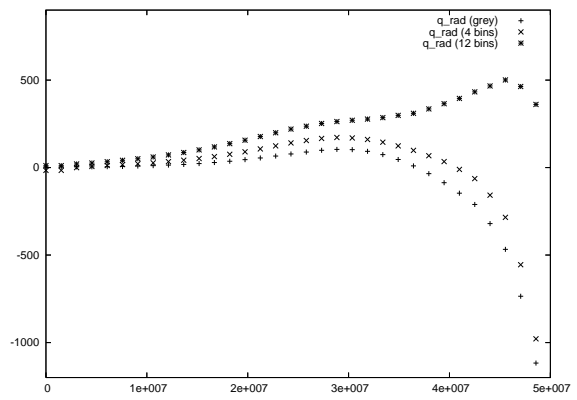
It is assumed that the solution is more accurate the more bins are used in the calculation of the mean intensity. Therefore the two-dimensional simulations are performed with 12 bins. Three-dimensional simulations are more time expensive. Thus three-dimensional simulations are carried out with four bins or even grey, depending on the the problem to be tackled.



(a)



(b)



(c)

Figure 3.13: Horizontal average of the grey and non-grey solutions for Q_{rad} in the whole domain (a), and in the regions $400 \leq x \leq 700\text{km}$ (b) and $0 \leq x \leq 500\text{km}$ (c).

3.5 Temporal discretization

To advance the numerical solution in time a second or third order accurate Runge-Kutta scheme (Shu and Osher [29]) is used.

Runge-Kutta schemes solve ordinary differential equations of the form

$$q_t = L(q)$$

for given initial conditions. For the system of conservation laws (1.1), (1.2), (1.3) $q = (\rho, \rho \mathbf{u}, e)$ and the vector $L(q)$ contains the spatial derivatives and source terms.

Given the state q^n at time t . Using a second order Runge-Kutta scheme the state q^{n+1} at time $t + \Delta t$ is calculated in two steps:

$$\begin{aligned} q_0 &= q^n \\ q_1 &= q_0 + \Delta t \cdot L(q_0) \\ q_2 &= \frac{1}{2}(q_0 + q_1) + \frac{\Delta t}{2}L(q_1) \\ q^{n+1} &= q_2 \end{aligned} \tag{3.48}$$

Using a third order Runge-Kutta scheme the solution is calculated in three steps:

$$\begin{aligned} q_0 &= q^n \\ q_1 &= q_0 + \Delta t \cdot L(q_0) \\ q_2 &= \left(\frac{3}{4}q_0 + \frac{1}{4}q_1 \right) + \frac{\Delta t}{4}L(q_1) \\ q_3 &= \left(\frac{1}{3}q_0 + \frac{2}{3}q_2 \right) + \frac{2\Delta t}{3}L(q_2) \\ q^{n+1} &= q_3 \end{aligned} \tag{3.49}$$

Chapter 4

Comparison of the numerical schemes

In this chapter different schemes for the numerical solution of conservation laws and different implementations of the viscous stress tensor are compared to find differences and to specify the optimal numerical scheme. For the comparisons in this chapter only two-dimensional simulations are considered. Since all spatial derivatives are calculated by dimensional splitting the results of the two-dimensional simulations are assumed to be valid also for three-dimensional simulations.

The vertical and horizontal extents of the computational domain are $2749\text{km} \times 11179\text{km}$. The computational domain is provided with 182×485 grid points which yields a mesh size of about 15km in the vertical and 23km in the horizontal direction. This resolution is common for simulations of the surface layers of the sun (Stein and Nordlund [32], Kim and Chan [11], Vögler et al. [36]).

Beginning with a typical state for the solar photosphere and convection zone (calculated with WENO-5 and artificial diffusivities) one hour of solar time is calculated using the second order Runge-Kutta scheme (3.48).

With the choices $C_{\text{Courant}} = C_{\text{diffusive}} = \frac{1}{4}$ the time step Δt is 0.25 seconds. Using the fourth order discretization of the viscous stress tensor the Prandtl number $Pr = \frac{c_p \mu}{\kappa}$ varies from $1.4 \cdot 10^{-8}$ at the bottom to $1.0 \cdot 10^{-10}$ at the top. Using artificial diffusivities instead, the coefficients are set to $C_{\text{shk}} = 1$ and $C_{\text{hyp}} = 0.05$.

In the ANTARES code different numerical schemes for conservation laws are implemented. All of them are of essentially non-oscillatory type but show a different behavior. Thus the influence of the numerical schemes for conservations laws to the solution is studied. For the simulations presented in this chapter,

- a fourth order convex non-oscillatory scheme (CNO-4),
- a fifth order weighted essentially non-oscillatory scheme (WENO-5) and
- a third order essentially non-oscillatory scheme with Marquina's flux splitting and entropy-fix (ENO-3)

are used. For this comparison the viscous terms are replaced by the artificial diffusivities.

On the other hand side the influence of the use of artificial diffusivities is studied. Using the fifth order weighted essentially non-oscillatory scheme the algorithm is also stable for a fourth order discretization of the viscous terms in the momentum and energy equation. Thus it is possible to study the influence of the use of artificial diffusivities with the same code on the same starting model.

In the first section some results of the simulations with different numerical methods are presented which are discussed in the second section.

Granulation. Solar granulation is the visible manifestation of the convection in the upper regions of the sun. Convective motions are described by the velocity field and the number of convection cells and their temporal variations. At depth 600km the up- and downflows are studied. Furthermore horizontal

profiles of mass density, temperature and mean outgoing intensity are considered.

Additionally to the layer at depth 600km horizontally averaged quantities are calculated to see differences in the velocity field and the turbulent character of the model.

4.1 Results

4.1.1 Up- and downflows at 600km

At a depth of 600km the granulation in the two-dimensional simulation is comparable with the observations for determining the mean size of the granules. The mean distance d between granular centers, the mean cell size, is $1.94''$ (Roudier and Muller [27]) respectively $1.76''$ (Bray and Loughland [2]), which is 1400 respectively 1270 kilometers. Figure 4.1 shows the temporal variation of up- and downflows.

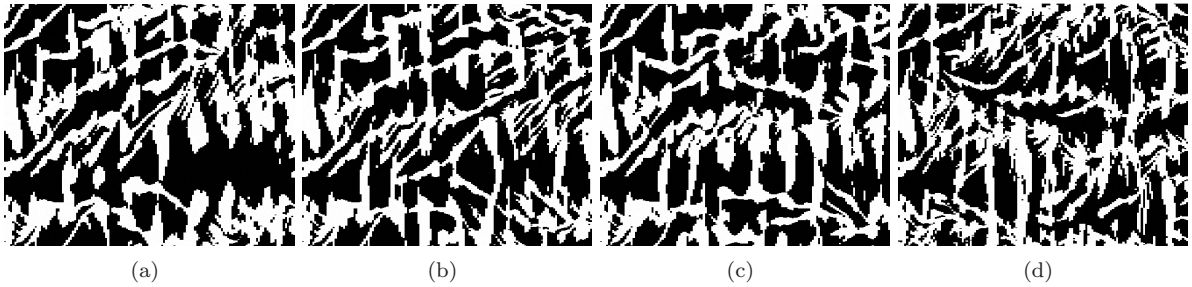


Figure 4.1: Temporal variation of up- and downflows at 600km calculated with CNO-4 (a), WENO-5 (b), ENO-3 (c) with artificial diffusivities and WENO-5 without artificial diffusivities (d). The slices are put side by side horizontally. White colored regions indicate downflows, black ones indicate upflows.

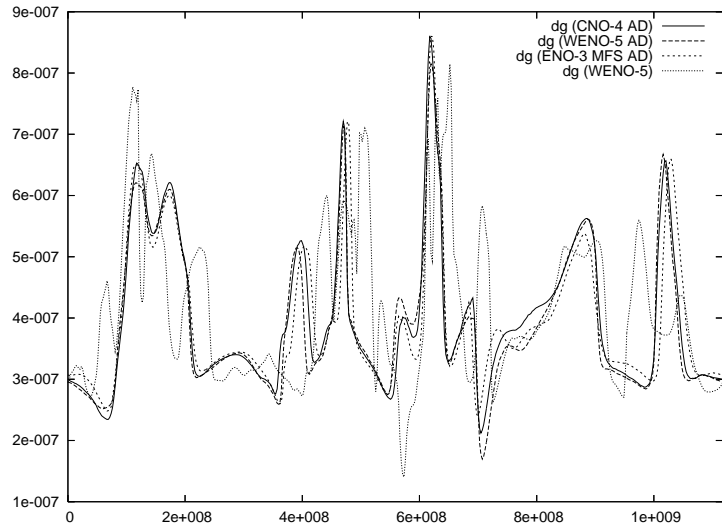
In a depth of 600km, the mean number of downflows N_{DF} and thus the mean cell size d are about:

	N_{DF}	d [km]
CNO-4 with artificial diffusivities	6.20	1806
WENO-5 with artificial diffusivities	7.25	1545
ENO-3 with artificial diffusivities	6.40	1750
WENO-5 without artificial diffusivities	8.00	1400

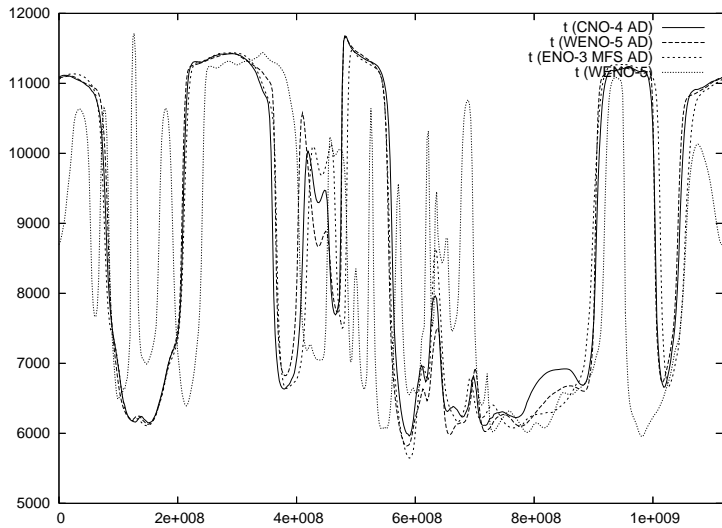
The velocity ranges for these up- and downflows are:

	$\mathbf{u}_{x,\min}$ [cm s^{-1}]	$\mathbf{u}_{x,\max}$ [cm s^{-1}]
CNO-4 with artificial diffusivities	$-8.9 \cdot 10^5$	$9.5 \cdot 10^5$
WENO-5 with artificial diffusivities	$-10.6 \cdot 10^5$	$10.3 \cdot 10^5$
ENO-3 with artificial diffusivities	$-10.2 \cdot 10^5$	$9.9 \cdot 10^5$
WENO-5 without artificial diffusivities	$-10.2 \cdot 10^5$	$13.2 \cdot 10^5$

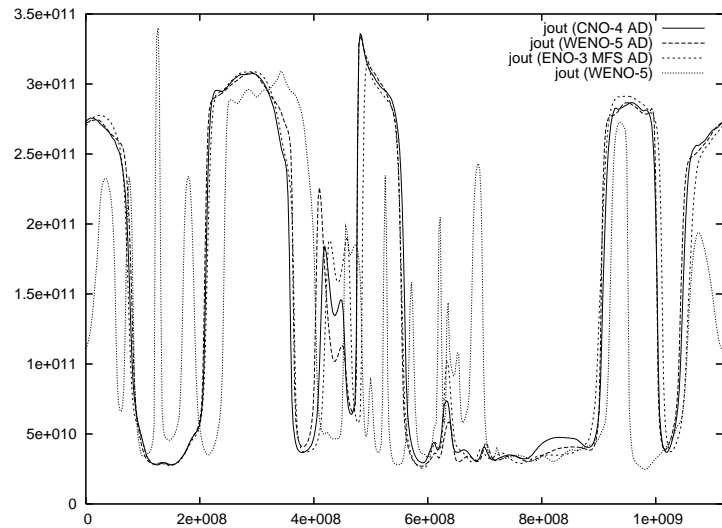
Figure 4.2 shows the horizontal slice at depth 600km for the mass density, the temperature and the mean outgoing intensity after seven minutes.



(a)



(b)



(c)

Figure 4.2: Horizontal slice of the temperature and the outgoing mean intensity at depth 600km after 7 minutes.

4.1.2 Comparison of horizontally averaged quantities

Figures 4.3 and 4.4 show the temporal mean of the following quantities:

- Horizontally averaged temperature.
- Horizontally averaged absolute value of the velocity.
- Horizontally averaged absolute value of the square of the vertical velocity.
- Horizontally averaged absolute value of the vertical velocity.
- Horizontally averaged absolute value of the horizontal velocity.
- Maximal value of the absolute value of the vertical velocity.

- Horizontal average $\left(\sum_{i,j} \left(\frac{\partial u_i}{\partial x_j} \right)^2 \right)^{\frac{1}{2}}$.

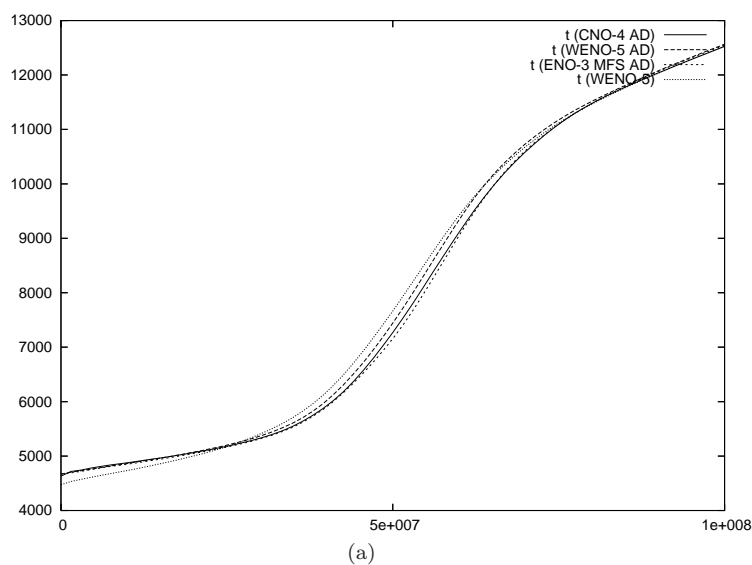


Figure 4.3: Temporal mean of the mean horizontal temperature calculated with CNO-4, WENO-5, ENO-3 with artificial diffusivities and WENO-5 without artificial diffusivities.

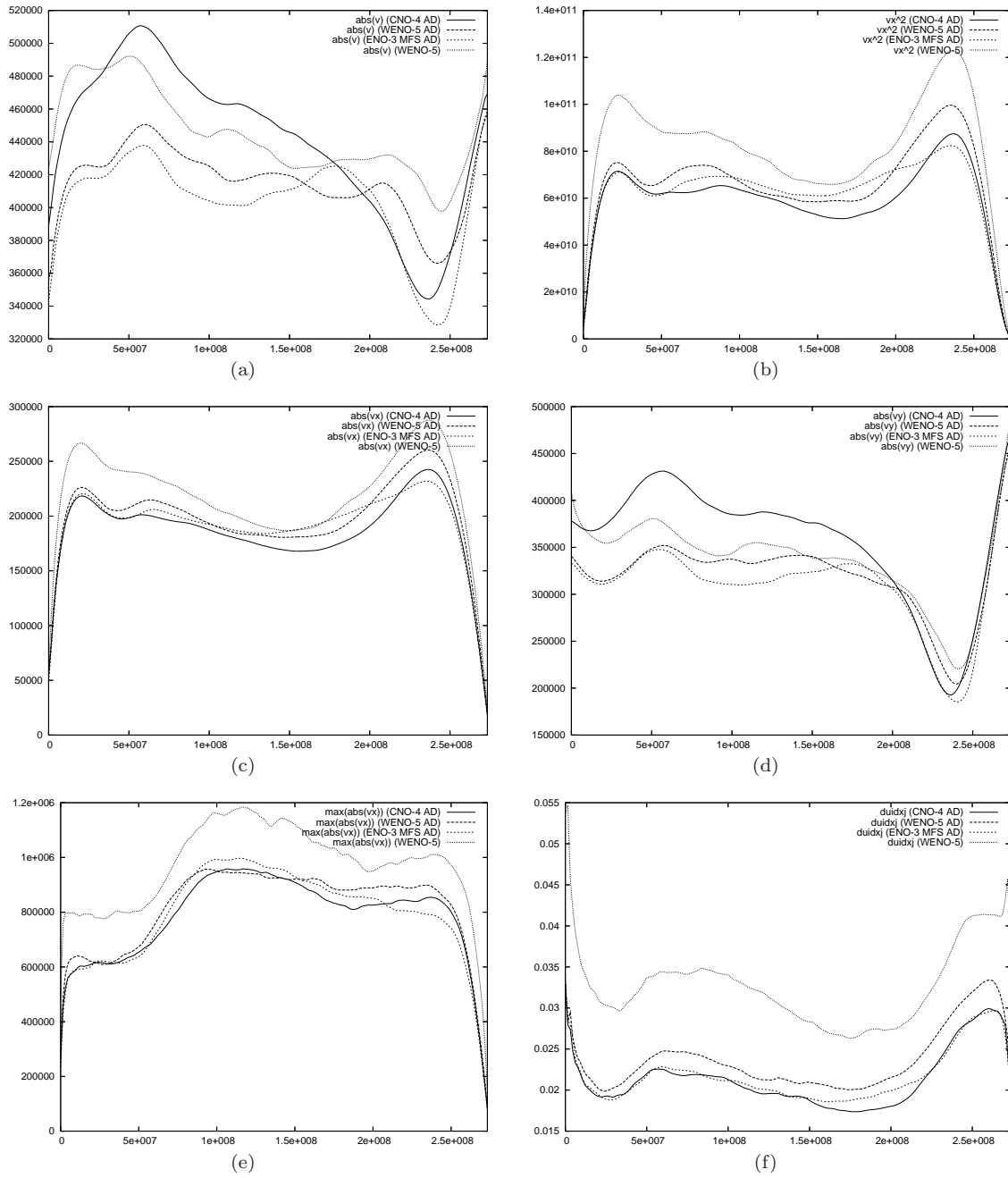


Figure 4.4: Temporal mean of the horizontal mean values of the velocity (a), of the square of the vertical velocity (b), of the absolute value of the vertical velocity (c), of the absolute value of the horizontal velocity (d), of the maximal value of the absolute value of the vertical velocity (e) and of $\left(\sum_{i,j} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right)^{\frac{1}{2}}$ (f) for the different numerical schemes.

4.2 Discussion

4.2.1 Effects using different numerical schemes for conservation laws

One main difference between the simulations with the three numerical schemes for conservation laws

- a fourth order convex non-oscillatory scheme (CNO-4),
- a fifth order weighted essentially non-oscillatory scheme (WENO-5) and
- a third order essentially non-oscillatory scheme with Marquina's flux splitting and entropy-fix (ENO-3),

where the viscous terms are replaced by the artificial diffusivities is the temporal average of the mean number of downflows N_{DF} . For the chosen resolution (cell size $15 \times 23 \text{ km}$) none of the three methods yields an appropriate number of downflows. The fifth order weighted ENO scheme is closest to the observational values.

The horizontal profile of the mass density (figure 4.2) is similar for all of these three methods. Here the mass density is considered since in its conservation law no viscous terms are present, effects of the artificial diffusivities only enter via the flux function (x - and y -momentum).

Since the quantities ρ , $\rho \mathbf{u}$ and e don't show notable differences it is clear that the quantities received by the equation of state and by the radiative transfer solver also are similar. See figure 4.2 for the horizontal profile of the temperature and the mean outgoing intensity.

The temporal mean of the horizontally averaged temperature (figure 4.3 displays this quantity in the upper half of the domain) shows small differences. In a way the ENO based schemes produce steeper gradients which corresponds to the observations in the test problem. This effect is obvious comparing the solutions for CNO-4 and ENO-3.

Considering the temporal mean of the horizontally averaged velocity quantities (figure 4.4) one gets differences between these three methods:

- There are significant differences in the horizontal averages of $\|\mathbf{u}\|$. The horizontal averages of $|\mathbf{u}_x|$ are similar with small differences in the deeper half of the computational domain. The horizontal averages of $|\mathbf{u}_y|$ are very different, the resulting horizontal velocity for the model with CNO-4 is faster. Consequently the differences in the horizontal averages of $\|\mathbf{u}\|$ are due to the different horizontal velocity profiles.
- The quantity \mathbf{u}_x^2 is proportional to the vertical kinetic energy flux which describes the vertical energy transport through velocity. The shapes of the curves for the horizontal averages of \mathbf{u}_x^2 are similar but different, the same differences as in the horizontal averages of $|\mathbf{u}_x|$ are present.
- The quantity $\left(\sum_{i,j} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right)^{\frac{1}{2}}$ is a measure for the turbulent behavior in the system. The shape of the horizontal averages of this quantity is similar for all three methods, the magnitudes are slightly different.
- The shape of the three curves of the horizontally averaged quantities for $|\mathbf{u}_x|$, \mathbf{u}_x^2 and the quantity measuring the turbulent behavior are similar for the three numerical methods.
- The vertical velocity range is similar at depth 600km and the maximal value of the vertical velocity is similar in the whole domain.

Since the velocity fields for the up- and downflows are similar for all three methods, since the vertical velocity ranges are similar and since the turbulent behaviors are similar and the differences in the horizontal profile after seven minutes are small, the optimal method is the one which produces granules of the appropriate size. Hence using artificial diffusivities the fifth order weighted ENO scheme is preferred.

4.2.2 Effects using artificial diffusivities

Prandtl number for artificial diffusivities. The Prandtl number Pr is an important similarity parameter for fluid flow:

$$Pr = \frac{c_p \mu}{\kappa} \quad (4.1)$$

μ is the dynamic viscosity, κ is the thermal conductivity and c_p is the specific heat at constant pressure. The right hand side of (4.1) can be rewritten as $\frac{\mu/\rho}{\kappa/\rho c_p}$. Hence the Prandtl number is the ratio of the kinematic viscosity to the thermal diffusivity. The viscous behavior of the the hydrodynamical equations (1.1), (1.2), (1.3) can be described by the Prandtl number.

Through the comparison of the entries \mathcal{T}_{kl} of the viscous tensor and the artificial viscous tensor

$$Pr \frac{\kappa}{c_p} \left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k} - \frac{2}{3} \delta_{kl} (\nabla \cdot \mathbf{u}) \right) = \frac{1}{2} \rho \left(\nu_k(u_l) \frac{\partial u_l}{\partial x_k} + \nu_l(u_k) \frac{\partial u_k}{\partial x_l} \right) \quad (4.2)$$

one gets an estimation for the Prandtl number for artificial diffusivities.

Figure 4.5 shows this estimate for the starting configuration, at which the two-dimensional numerical comparisons begin. At the top the Prandtl number is about 10^{-5} and the Prandtl number increases to 10^5 at the bottom. Since the Prandtl number in the solar photosphere is about 10^{-8} (Komm, Mattig and Nesis [12]) the artificial diffusivities only stabilize the code, they don't describe the viscous behavior. Therefore a comparison of simulations with and without artificial diffusivities is essential.

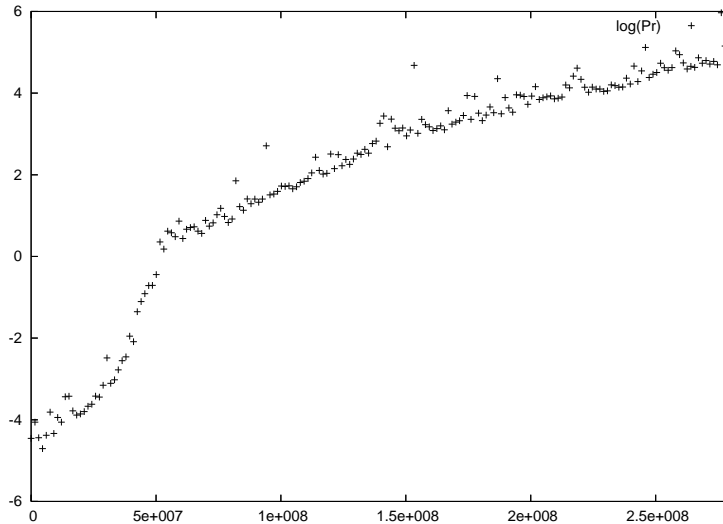


Figure 4.5: Estimate for \log_{10} of the Prandtl number using artificial diffusivities.

For constant Pr the contribution by the viscous stress tensor \mathcal{T}_{kl} declines with increasing depth since the fraction $\frac{\kappa}{c_p}$ declines rapidly and the fields of the velocity gradients don't vary significantly inside the computational domain. The entries in the artificial diffusion tensor increase since the value in the brackets which shows the same behaviour inside the computational domain is multiplied by the mass density.

Comparison. In this section the simulations with the fifth order weighted ENO scheme for the conservation laws and either a fourth order discretization of the 'proper' viscous terms or with the use of artificial diffusivities are compared.

The horizontal profiles after seven minutes (figure 4.2) show some differences. This indicates that the temporal development without the use of artificial diffusivities is different. In regions with lower temperature higher temperature regions arise and conversely regions with higher temperature divide. Since

high temperature regions are equivalent to upstreams and low temperature regions are equivalent to downstreams figure 4.2 suggests that the velocity field changes more rapidly using a fourth order discretization of the viscous tensor and that the mean life-time of a granule is shorter.

The temporal mean of the horizontally averaged temperature (see figure 4.3) shows that the temperature gradient for simulations with artificial diffusivities is steeper. This could be correlated with a different horizontally averaged temperature profile: A steeper gradient in the first quarter and a flatter gradient in the second and the third quarter of the domain.

Figure 4.4 shows differences in every subfigure. Generally speaking, the values for the simulation without artificial diffusivities are significantly higher. I.e. this means that the kinetic energy flux is higher and that the behaviour is more turbulent.

Furthermore, using a fourth order discretization for the viscous terms one gets an acceptable mean number of downflows at depth 600km, namely 8, resulting in a cell size of 1400km.

4.2.3 Different spatial resolution

Since the number of downflows for the simulations with artificial diffusivities and some horizontally averaged quantities are far from the desired values, the resolution of the model is increased (7.5km in the vertical and 11.5km in the horizontal direction) to figure out if the solution then 'converges' to the solution without artificial diffusivities.

Up- and downflows at 600km. In depth 600km, the number of downflows N_{DF} and the mean cell size d are about:

	N_{DF}	d [km]
WENO-5 without artificial diffusivities	8.00	1400
WENO-5 with artificial diffusivities	7.25	1545
WENO-5 with artificial diffusivities (high resolution)	7.95	1408

Comparison. Using higher resolution (7.5km×11.5km) for the simulation with the fifth order weighted ENO scheme for the conservation laws and the use of artificial diffusivities the solution 'converges' to that without the use of artificial diffusivities such that the magnitude is comparable. Nevertheless, it has to be stated that the velocity field will always be different since the viscous behaviour is described by different expressions.

The shapes of the horizontal averages curves (i.e. the location of minima and maxima) don't vary much through increasing the resolution, except for the horizontal average of $\|\mathbf{u}\|$. The magnitude of these curves increases significantly for all these horizontally averaged curves.

The mean number of downflows is 7.95, resulting in a cell size 1408km which is appropriate.

4.2.4 Conclusion

Using higher resolution (i.e. 7.5km×11.5km) the use of artificial diffusivities is feasible. Three-dimensional simulations at this higher resolution are computing time-expensive. If one doesn't want to forego the damping and diffusing properties of the artificial diffusivities lower resolution simulations should be performed with the fifth order weighted ENO scheme which minimizes the differences.

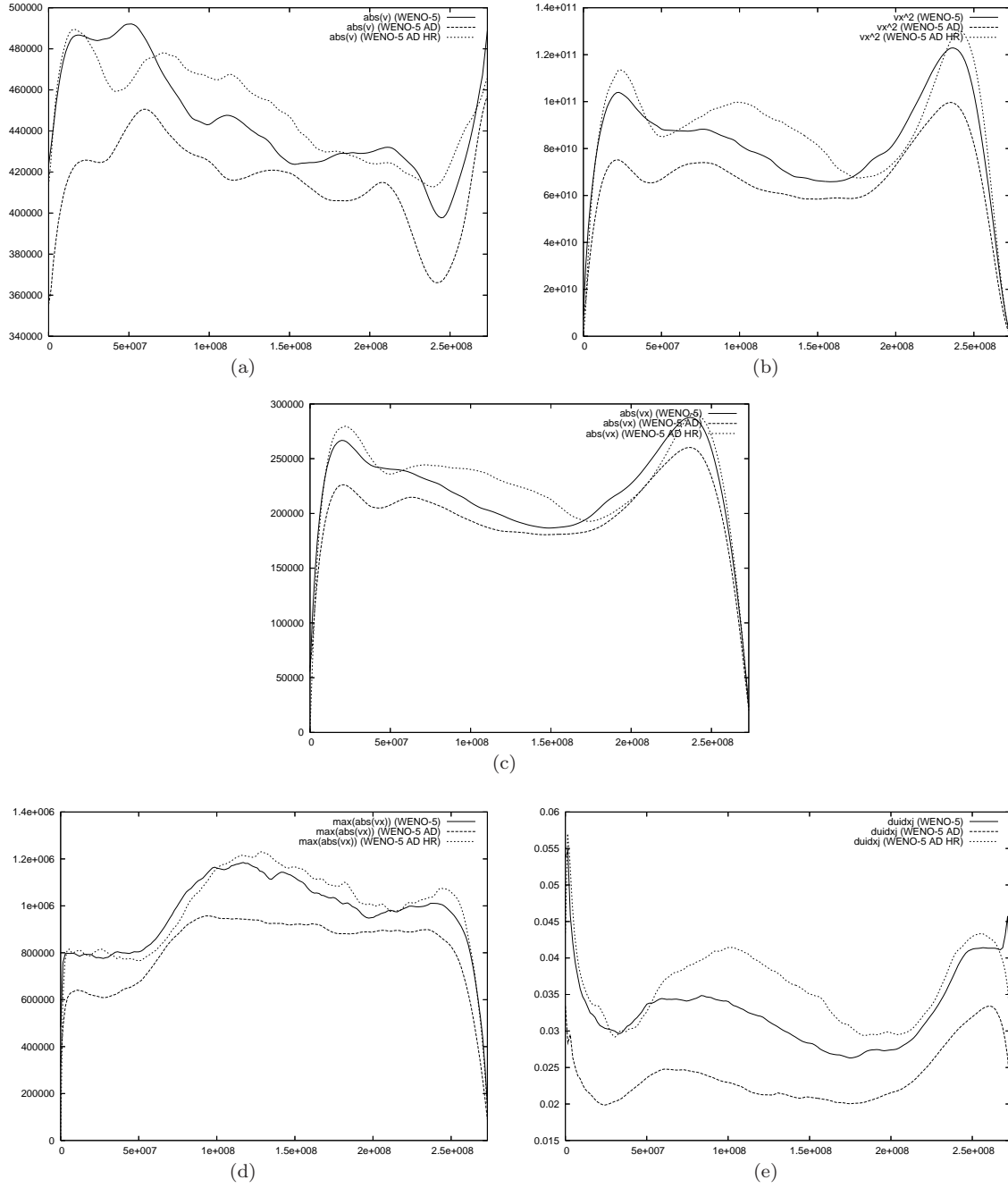


Figure 4.6: Temporal mean of the horizontal mean values of the velocity (a), of the square of the vertical velocity (b), of the absolute value of the vertical velocity (c) of the maximal value of the absolute value of the vertical velocity (d) and of $\left(\sum_{i,j} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right)^{\frac{1}{2}}$ (e) for the different numerical schemes and resolutions.

Chapter 5

A two-dimensional high resolution study of solar surface flows

Modelling of solar granulation describes reality in many details (Stein and Nordlund [33]). Nevertheless, important issues remain unchecked, particularly processes occurring on small scales which neither observations nor models did resolve. Small scale features are important because they may influence the global outcome and the overall physical structure. Thus the study of small scales is important.

Grid refinement. For a detailed study of the region near a downflowing plume local grid refinement is used. First, the region of interest for the refined area (dark grey region in figure 5.1) and the (integer) grid refinement factor (for each direction) must be specified. The refined area is surrounded by ghost cells (light grey regions in figure 5.1) to allow regular interpolation and symmetric differentiation near the boundaries. Then the values of the physical quantities at the coarse grid are interpolated to the fine grid and the ghost cells (light and dark grey region). This yields the starting state for the simulations. Furthermore a time step for the time evolution at the fine grid $\Delta t_{GR} = \frac{\Delta t}{N}$ for an integer N is calculated, which fulfils all time step restrictions. Typically, Δt is divided by the grid refinement factor in the vertical direction, if this is no contradiction to the diffusive time step restrictions.

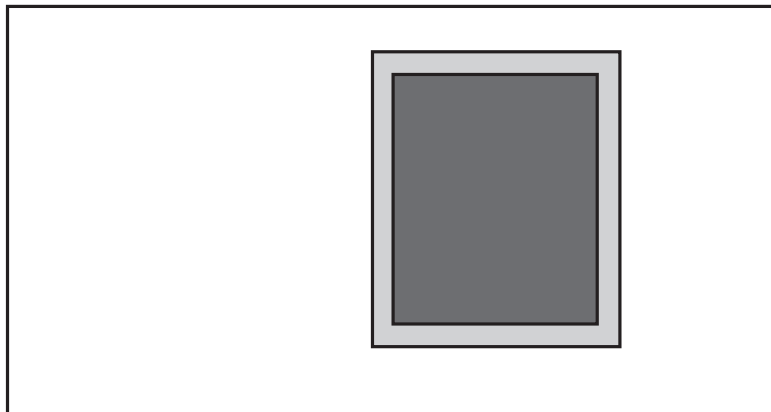


Figure 5.1: Domains for a two-dimensional grid refinement.

Each step of a simulation with a grid refinement region consists of a Runge-Kutta time step at the whole coarse grid. Then the initial data at the coarse grid are interpolated to the ghost cells surrounding the refined area and N Runge-Kutta steps at the fine grid are performed. Before every (intermediate) Runge-Kutta step, which corresponds to a time $t_{GR} \in [t, t + \Delta t]$, the ghost cells are assigned with linear interpolation values from the physical states $\mathbf{q}(t)$ and $\mathbf{q}(t + \Delta t)$ from the coarse grid. The resulting solution after N

Runge-Kutta steps $\mathbf{q}_{GR}(t + \Delta t)$ at the fine grid is projected to the corresponding grid points of the coarse grid.

For two-dimensional simulations, 4×4 grid points are used to get the interpolation values. First four one-dimensional interpolations in direction 1 and then one one-dimensional interpolation in direction 2 are performed to get the interpolation value. The successive interpolation procedure is described in section 3.3.

For three-dimensional interpolations $4 \times 4 \times 4$ grid points are used and the procedure described above is performed starting with 16 interpolations in direction 1, followed by four interpolations in direction 2 and then one interpolation in direction 3.

For parallelization both the coarse and the fine grid are distributed to the processors according to the distributed memory concept. Hence the computational effort is uniformly distributed among all processors.

High resolution simulation of a downflowing plume. The computational domain extends $2773\text{km} \times 11179\text{km}^1$. The refined range covers approximately 10-80% in x -direction and 57-80% in y -direction. The refinement factor is four for both directions. This means a cell in the coarse grid is $7.3\text{km} \times 11.4\text{km}$ and in the fine grid $1.8\text{km} \times 2.8\text{km}$. The presented results are from a simulation with the fifth order weighted ENO scheme for the conservation laws with artificial diffusivities and a second order Runge-Kutta scheme (3.48) for the temporal discretization. The required constants are set to $C_{\text{Courant}} = C_{\text{diffusive}} = \frac{1}{4}$, $C_{\text{shk}} = 1$ and $C_{\text{hyp}} = 0.05$. The resulting time step is $\Delta t = 0.125\text{s}$.

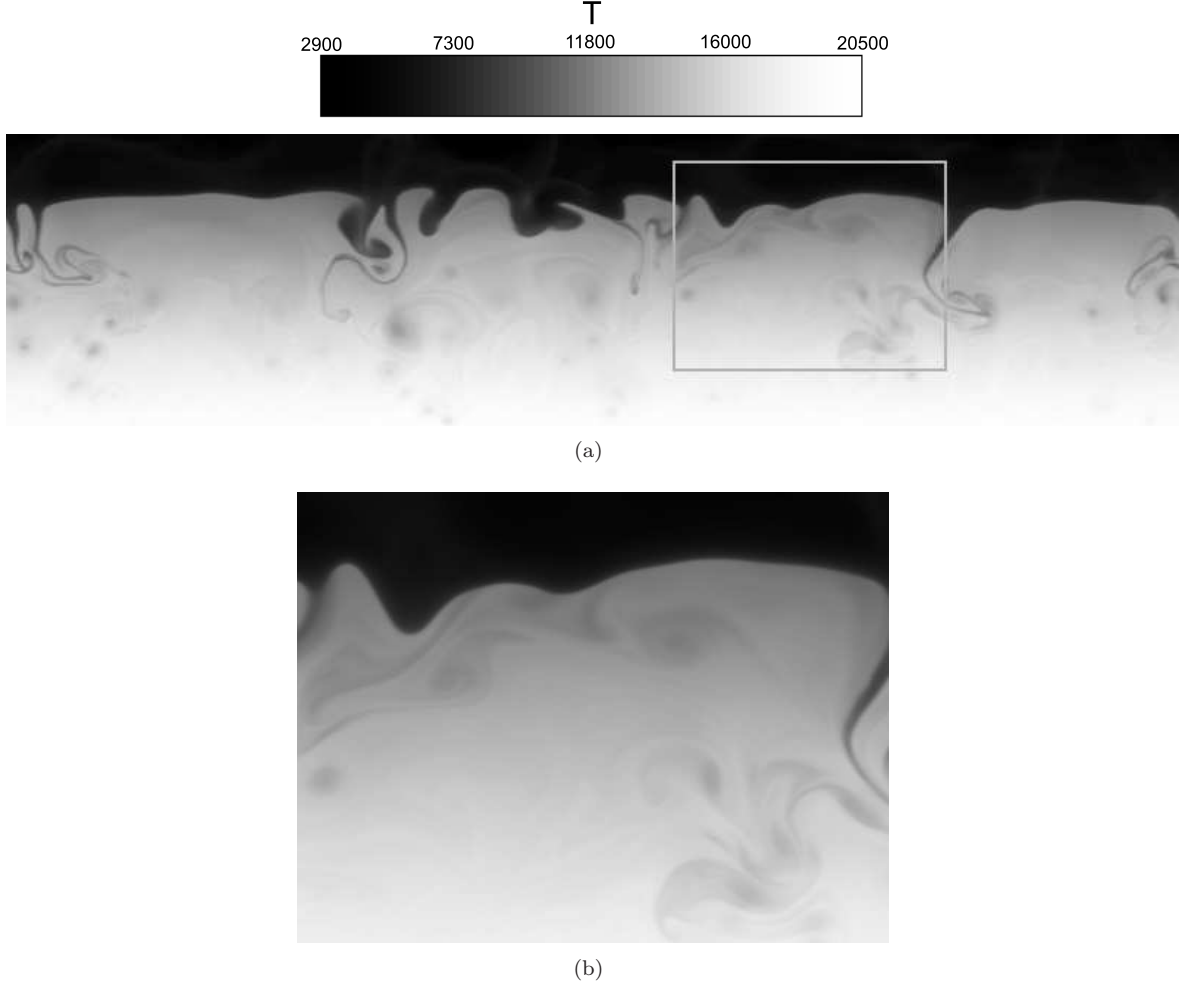


Figure 5.2: Start temperature distribution for the coarse (a) and the fine (b) grid. The temperature range in the whole domain covers $t \in [2900, 20500]$ Kelvin, the one for the refined range is $t \in [4400, 18000]$ Kelvin.

¹The differences in the vertical extent between this model and the simulations in chapter 4 are due to the implementation of the boundary conditions: The basic ANTARES code uses one ghost cell to implement the boundary conditions. For the simulations of the solar granulation at the top of the domain three ghost cells are introduced. In the present development these three ghost cells reduce the vertical extent. Hence the vertical extent depends on the spatial resolution.

Evolution of the downdraft. Figure 5.3 shows the evolution of the downdraft. The downflowing plume splits into finer ones which remain connected to the original plume. This behaviour is also visible in figure 5.4. Small channels with higher density develop.

Comparing the high resolution simulation with a coarser one (figure 5.5) the diameter of the plumes' cores is much smaller, low temperature channels extend deeper. The main downflowing low temperature channel has got a diameter between 20 and 25km.

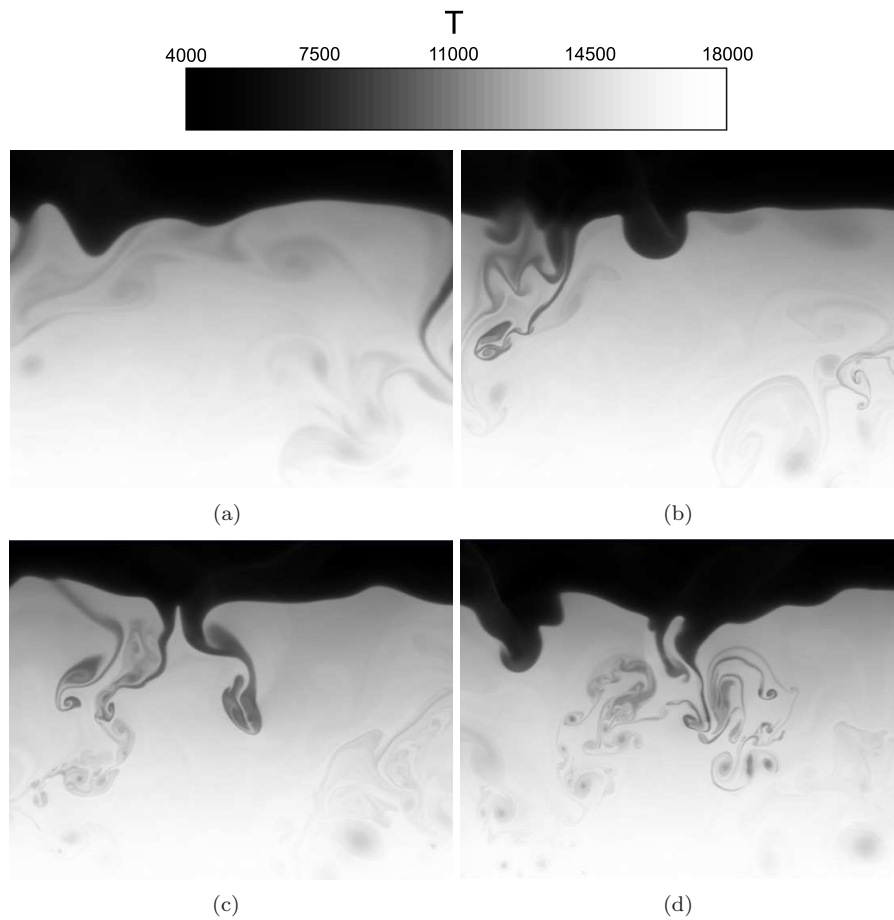


Figure 5.3: Development of the downflowing plume. Temperature distribution after 0 (a), 2 (b), 4 (c), 6 (d) minutes.

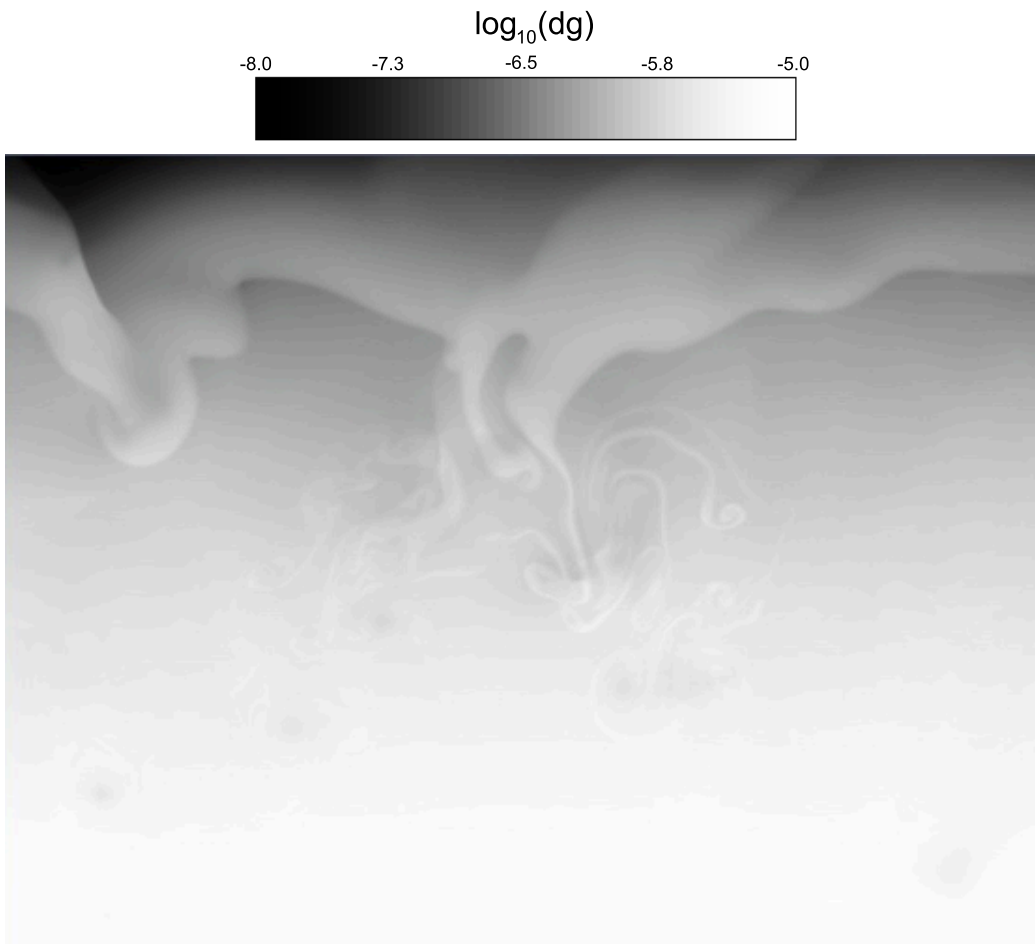


Figure 5.4: Density distribution after 6 minutes.

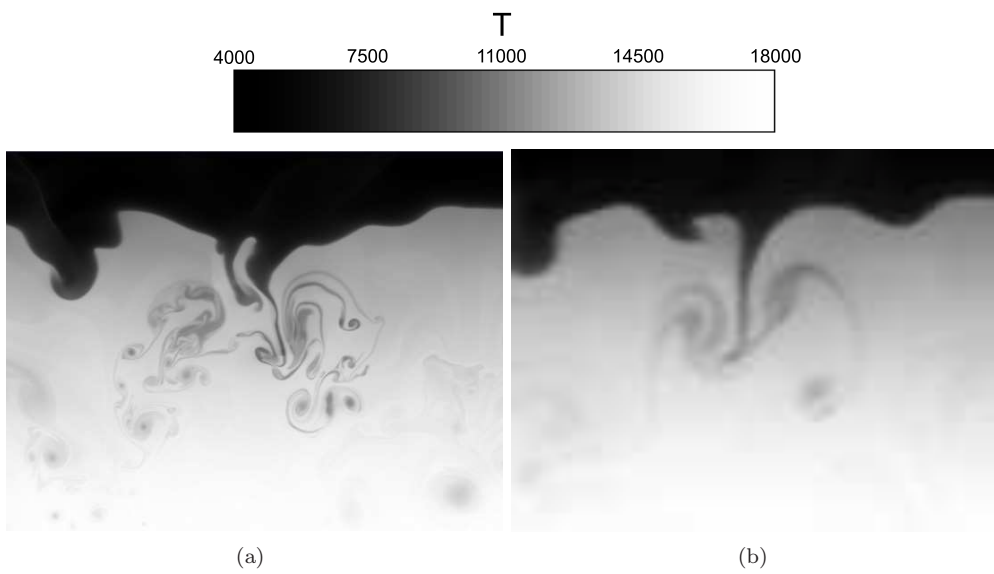


Figure 5.5: Temperature distribution after 6 minutes for simulations with cell size $1.8\text{km} \times 2.8\text{km}$ (e) and with cell size $15\text{km} \times 23\text{km}$ (f).

Horizontal averages. The higher the resolution the more details of a strong downdraft are visible. Different spatial resolutions also influence the temporal means of horizontally averaged quantities. In the simulations without the grid refinement zone the horizontal averages are taken only over the region which is refined. Thus all horizontal averages are comparable to one another.

The temporal mean of the horizontal average $\left(\sum_{i,j} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right)^{\frac{1}{2}}$ (see figure 5.6) shows the differences in the turbulent behavior of the model depending on the resolution, which is obvious considering the comparison of the temperature distribution after six minutes (see figure 5.5).

Obviously all temporal means of the horizontally averaged quantities shown in figure 5.6 converge. The finest resolution is not necessary if one is only interested in horizontal averages, it is only important to get as many small scale features as possible.

The finer the resolution the faster the absolute value of the vertical velocity and hence the vertical kinetic energy flux.

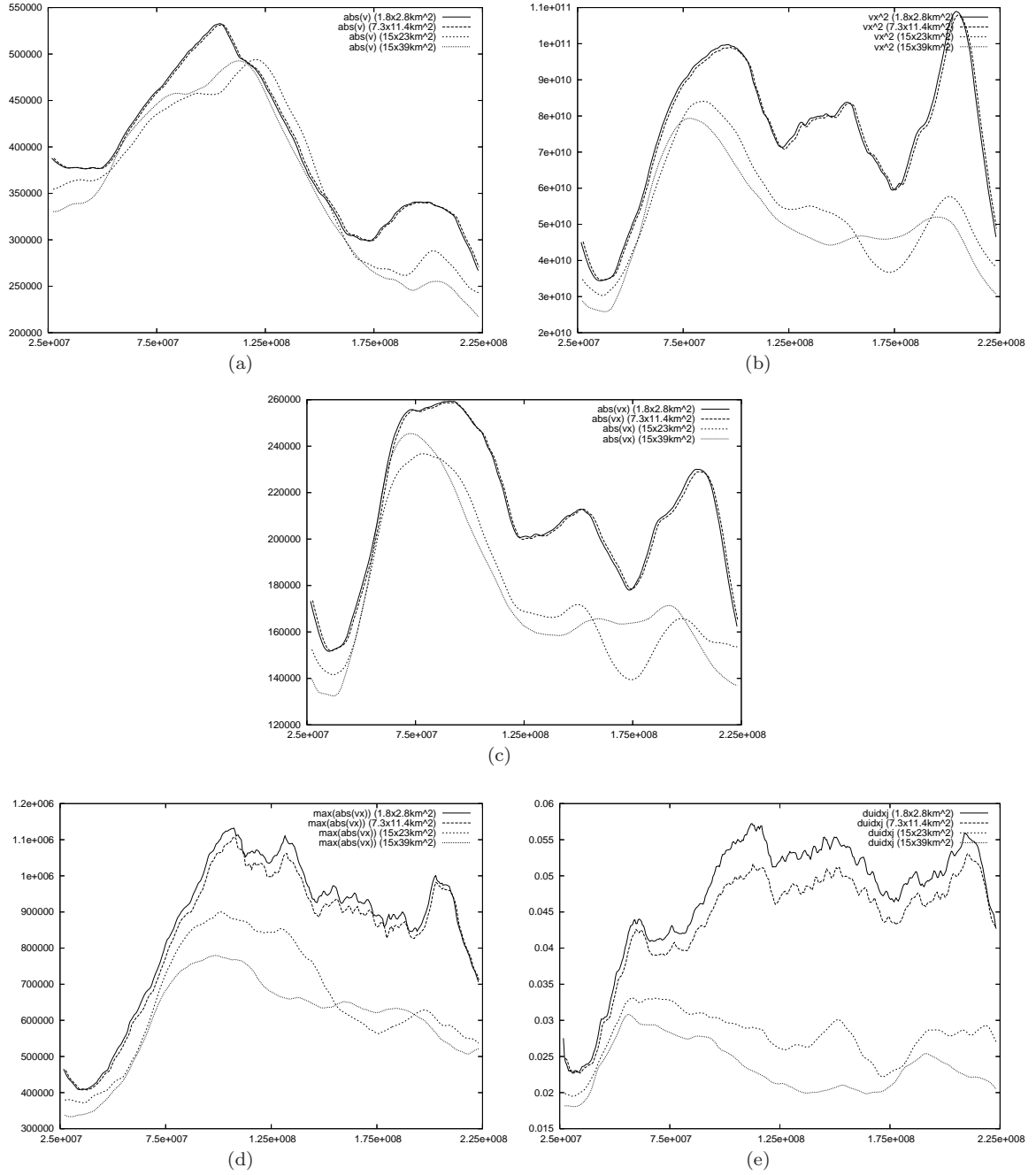


Figure 5.6: Temporal mean of the horizontal mean values of the velocity (a), of the square of the vertical velocity (b), of the absolute value of the vertical velocity (c), of the maximal value of the absolute value of the vertical velocity (d) and of the horizontal mean values of $\left(\sum_{i,j} \left(\frac{\partial u_i}{\partial x_j}\right)^2\right)^{\frac{1}{2}}$ (e) for the different spatial resolutions.

Acoustic pulses. The downflowing plumes generate acoustic pulses. The figures 5.7 and 5.8 plot the logarithm of the pressure with horizontal averages subtracted. Acoustic pulses arise in the downdrafts and then propagate through the domain. Figure 5.8 shows that these pulses propagate in all directions.

Following the pulse is a region of relatively high density and temperature, at least near its origin. Density and temperature are next to discontinuous across the pulse. During the origin of these pulses perturbations in the velocity (see the left upper corner of figure 5.9) arise but generally the velocity field doesn't map the pressure perturbations. These pulses can move against the local flow field and are therefore not advection phenomena but more akin to sound waves.

The pressure figures partly show small horizontal perturbations which are due to the transition from the radiative heating rate to the diffusion approximation and do not influence the observed behaviour. In subsequent runs these perturbations are minimized by choosing a smoother transition between them.

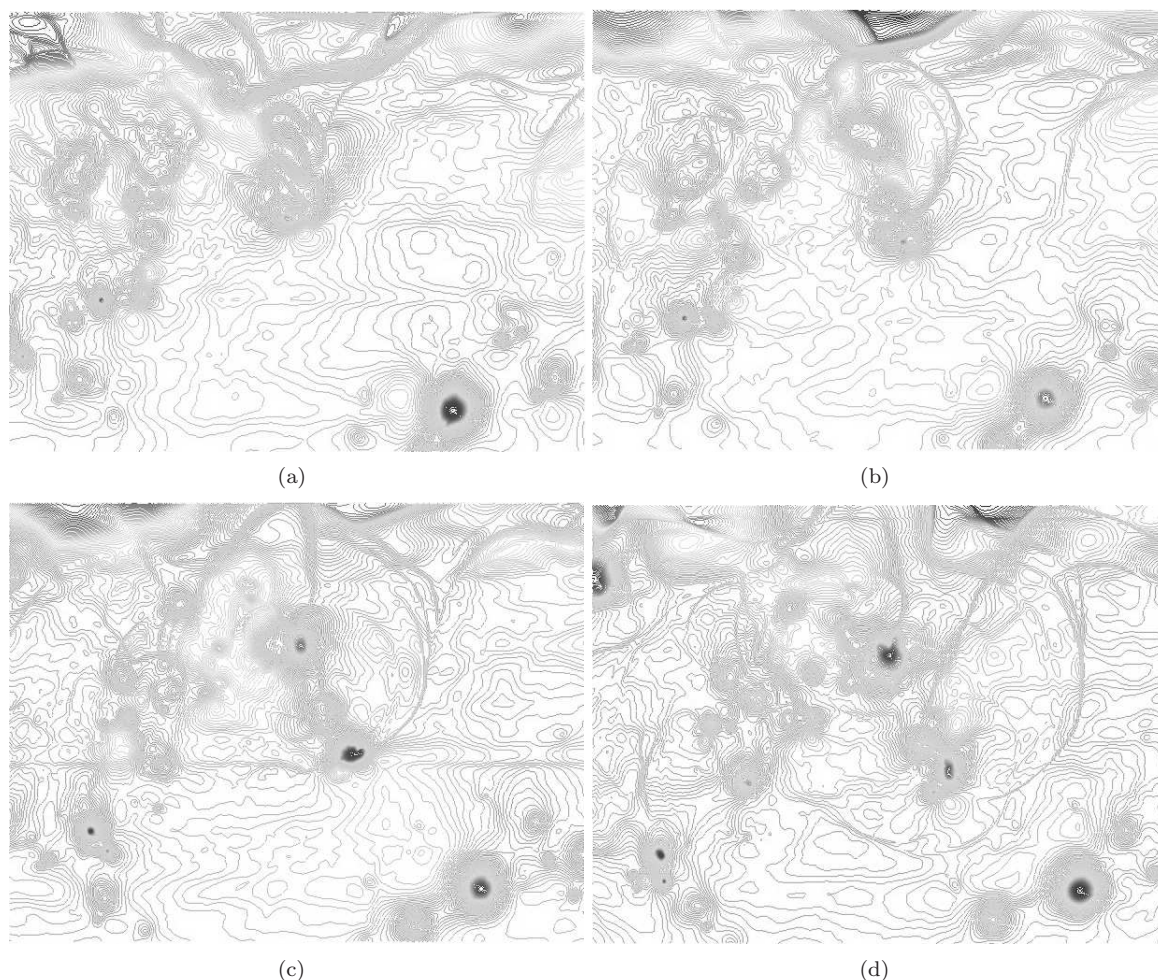


Figure 5.7: Acoustic pulses (logarithm of the pressure with horizontal averages subtracted) arise and develop in the downflowing plume. The time between two subsequent snapshots is 25 seconds and the logarithmic pressure range is $[4,7.1]$.

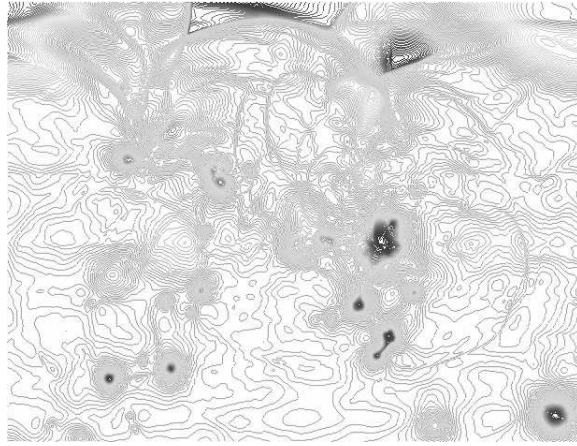


Figure 5.8: Acoustic pulses (logarithm of the pressure with horizontal averages subtracted) after 7 minutes.

Kelvin-Helmholtz instabilities. In fluids Kelvin-Helmholtz instabilities in particular occur when different densities move at various speeds. This is true for the interface between the up- and downflows. Rotating regions of lower density and higher temperature develop. This is illustrated in figure 5.9.

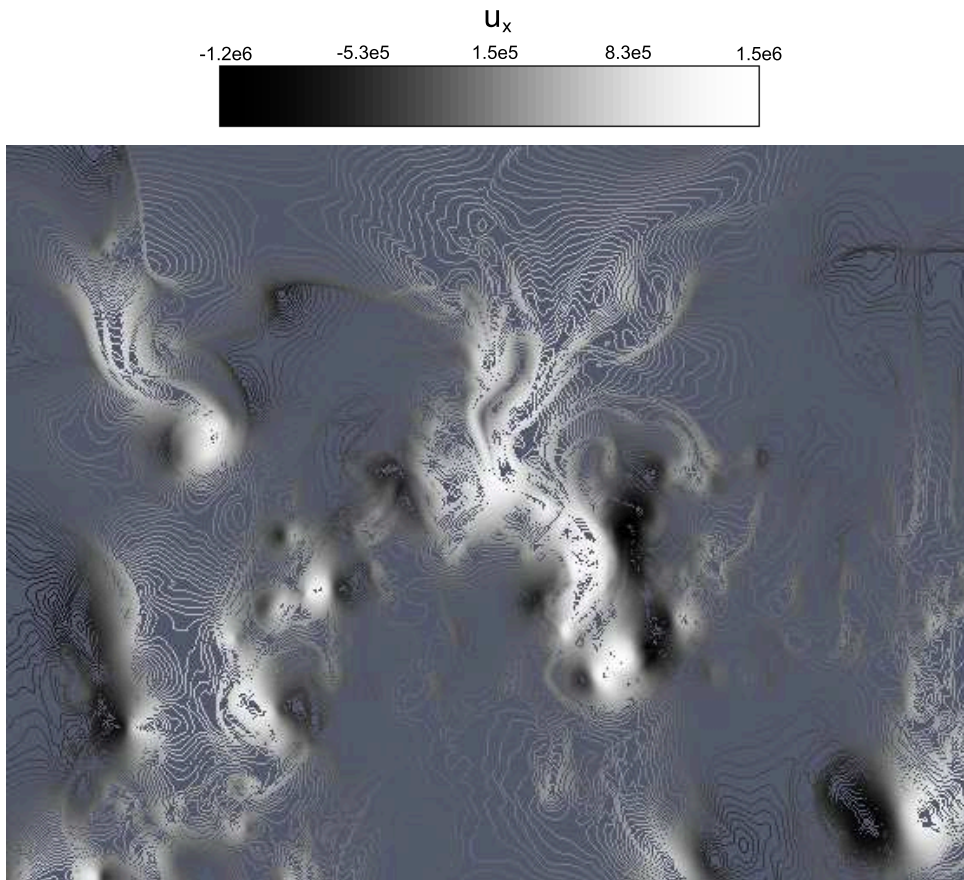


Figure 5.9: x -velocity after 6.5 minutes.

Influence of the viscous terms. Performing the same simulation with the fourth order discretization of the viscous tensor the results are nearly the same. Locally small differences in the structure of the downflowing plume are visible.

Chapter 6

A grey three-dimensional simulation of the quiet sun

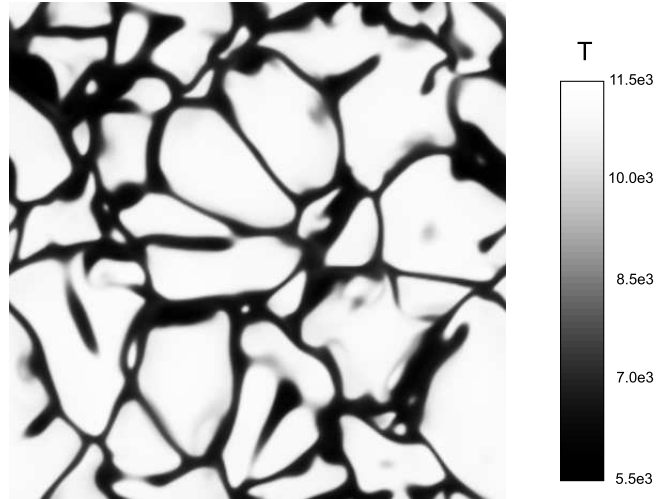
This chapter briefly shows some intermediate results of a grey three-dimensional simulation in order to show that the ANTARES code also performs well for the 3D case. This model will yield a starting point for a high resolution study of a downflowing plume in the 3D case.

First a low resolution two-dimensional model representing $2749\text{km} \times 11179\text{km}$ is evolved in time using 82×85 grid points. Then this two-dimensional model is converted to a three-dimensional one by putting replica of the two-dimensional model side-by-side. This low resolution three-dimensional model is evolved in time and the resolution is increased successively resulting in a model with extents of $2749\text{km} \times 11179\text{km} \times 11179\text{km}$ using $182 \times 285 \times 285$ grid points. Thus the resulting resolution is approximately $15\text{km} \times 39\text{km} \times 39\text{km}$.

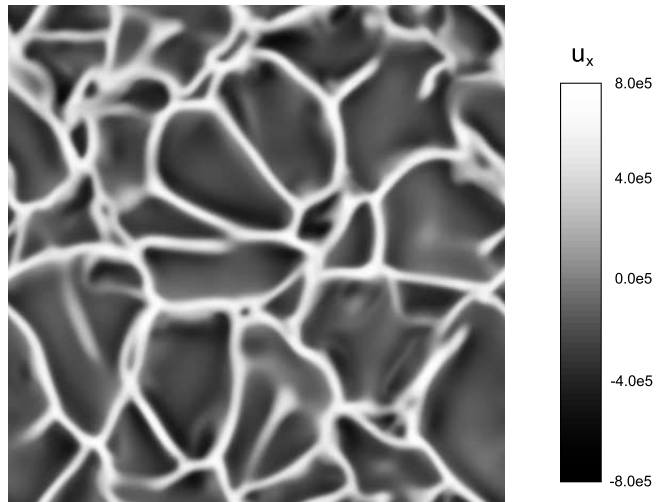
The results present a grey simulation with the fifth-order weighted ENO scheme for the conservation laws with artificial diffusivities and a second order Runge-Kutta scheme (3.48) for the temporal discretization. Again $C_{\text{Courant}} = C_{\text{diffusive}} = \frac{1}{4}$, $C_{\text{shk}} = 1$ and $C_{\text{hyp}} = 0.05$. The resulting time step is $\Delta t = 0.25\text{s}$.

Figure 6.1 shows temperature, vertical velocity and outgoing intensity at the horizontal layer corresponding to optical depth zero. The structure for these distributions is similar: Warm upflows are brighter than cool downflows.

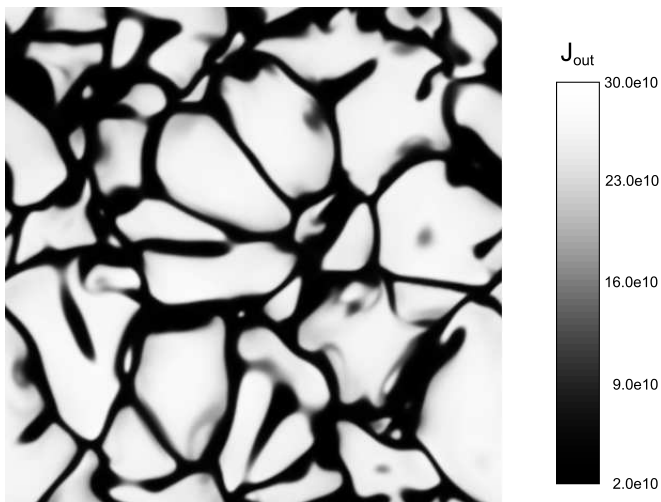
The number of granules respectively their sizes correspond to the results of the two-dimensional simulation with the same numerical setup.



(a)



(b)



(c)

Figure 6.1: Temperature, vertical velocity and outgoing intensity for the horizontal cut corresponding to Rosseland mean optical depth unity.

Bibliography

- [1] D. R. Alexander and J. W. Ferguson, *Low-temperature Rosseland opacities*, The Astrophysical Journal **437** (1994), 879–891.
- [2] R. J. Bray and R. E. Loughland, *A new determination of the granule/intergranule contrast*, Solar Physics **54** (1977), 319–326.
- [3] J. H. M. J. Bruls, P. Vollmöller, and M. Schüssler, *Computing radiative heating on unstructured spatial grids*, Astronomy and Astrophysics **348** (1999), 233–248.
- [4] B. G. Carlson, *The numerical theory of neutron transport*, In Alder B., Fernbach S. (eds.) Methods in Computational Physics, 1963, pp. 1–42.
- [5] S. E. Caunt and M. J. Korpi, *A 3D MHD model of astrophysical flows: Algorithms, tests and parallelisation*, Astronomy and Astrophysics **369** (2001), 706–728.
- [6] J. Christensen-Dalsgaard, W. Dappen, S. V. Ajukov, E. R. Anderson, H. M. Antia, S. Basu, V. A. Baturin, G. Berthomieu, B. Chaboyer, S. M. Chitre, A. N. Cox, P. Demarque, J. Donatowicz, W. A. Dziembowski, M. Gabriel, D. O. Gough, D. B. Guenther, J. A. Guzik, J. W. Harvey, F. Hill, G. Houdek, C. A. Iglesias, A. G. Kosovichev, J. W. Leibacher, P. Morel, C. R. Proffitt, J. Provost, J. Reiter, E. J. Rhodes, Jr., F. J. Rogers, I. W. Roxburgh, M. J. Thompson, and R. K. Ulrich, *The Current State of Solar Modeling*, Science **272** (1996), 1286–1292.
- [7] R. Donat and A. Marquina, *Capturing Shock Reflections: An Improved Flux Formula*, Journal of Computational Physics **125** (1996), 42–58.
- [8] R. Fedkiw, B. Merriman, R. Donat, and S. Osher, *The penultimate scheme for systems of conservation laws: Finite difference ENO with marquina’s flux splitting*, Progress in Numerical Solutions of Partial Differential Equations, Arachon, France (M. Hafez, ed.), July 1998.
- [9] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III*, Journal of Computational Physics **71** (1987), 231–303.
- [10] G. Jiang and C.-W. Shu, *Efficient implementation of weighted eno schemes*, Journal of Computational Physics **126** (1996), 202–228.
- [11] Y.-C. Kim and K. L. Chan, *A hydrodynamic simulation of the highly superadiabatic layer of the sun*, The Astrophysical Journal **496** (1998), L121–L124.
- [12] R. Komm, W. Mattig, and A. Nesis, *The height dependence of velocity-intensity fluctuations and several non-dimensional parameters in the solar photosphere*, Astronomy and Astrophysics **252** (1991), 812–820.
- [13] G. L. Kunasz and L. H. Auer, *Short characteristic integration of radiative transfer problems: Formal solution in two-dimensional slabs*, Journal of Quantitative Spectroscopy and Radiative Transfer **39** (1988), 67–79.
- [14] R. L. Kurucz, *Smithsonian astrophysical observatory, CD-ROM*, 1993.
- [15] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser Verlag, Boston, USA, 1992.
- [16] X. Liu and S. Osher, *Convex eno high order multi-dimensional schemes without field by field decomposition or staggered grids*, Journal of Computational Physics **142** (1998), 304–330.

- [17] X. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics **115** (1994), 200–212.
- [18] A. N. Lowan, N. Davids, and A. Levenson, *Table of the zeros of the Legendre polynomials of order 1-16 and the weight coefficients for Gauss' mechanical quadrature formula*, Bulletin of the American Mathematical Society, 1942, pp. 739–743.
- [19] H. G. Ludwig, *Nichtgrauer Strahlungstransport in numerischen Simulationen stellarer Konvektion*, PhD thesis, Universität Kiel, 1992.
- [20] Maltby, P. et.al., *A new sunspot umbral model and its variation with the solar cycle*, The Astrophysical Journal **306** (1986), 284–303.
- [21] D. Mihalas, *Stellar Atmosphere*, second ed., W. H. Freeman and Company, 1978.
- [22] D. Mihalas, L. H. Auer, and B. R. Mihalas, *Two-dimensional radiative transfer. I - Planar geometry*, The Astrophysical Journal **220** (1978), 1001–1023.
- [23] A. Neumaier, *Introduction to Numerical Analysis*, Cambridge University Press, 2001.
- [24] A. Nordlund, *Numerical simulations of the solar granulation. I - Basic equations and methods*, Astronomy and Astrophysics **107** (1982), 1–10.
- [25] L. H. Olson and P. B. Kunasz, *Short characteristic solution of the non-LTE line transfer problem by operator perturbation - I. the one-dimensional planar slab*, Journal of Quantitative Spectroscopy and Radiative Transfer **38** (1987), 325–336.
- [26] F. J. Rogers, F. J. Swenson, and C. A. Iglesias, *OPAL Equation-of-State Tables for Astrophysical Applications*, The Astrophysical Journal **456** (1996), 902–908.
- [27] T. Roudier and R. Muller, *Structure of the solar granulation*, Solar Physics **107** (1986), 11–26.
- [28] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, Technical Report NASA CR-97-206253 ICASE Report No. 97-65, Institute for Computer Applications in Science and Engineering, 1997.
- [29] C.-W. Shu and S. Osher, *Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes*, Journal of Computational Physics **77** (1988), 439–471.
- [30] ———, *Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, II*, Journal of Computational Physics **83** (1989), 32–78.
- [31] M. Steffen, *A Simple Method for Monotonic Interpolation in One Dimension*, Astronomy and Astrophysics **239** (1990), 443–450.
- [32] R. F. Stein and A. Nordlund, *Simulations of Solar Granulation. I. General Properties*, The Astrophysical Journal **499** (1998), 914–933.
- [33] ———, *Realistic solar convection simulations*, Solar Physics **192** (2000), 91–108.
- [34] M. Stix, *The Sun - An Introduction*, Springer-Verlag Berlin Heidelberg, 1989.
- [35] A. Vögler, *Three-dimensional simulations of magneto-convection in the solar photosphere*, PhD thesis, Universität Göttingen, 2003.
- [36] A. Vögler, S. Shelyag, M. Schüssler, F. Cattaneo, T. Emonet, and T. Linde, *Simulations of magneto-convection in the solar photosphere. Equations, methods, and results of the MURaM code*, Astronomy and Astrophysics **429** (2005), 335–351.

Danksagung

Die vorliegende Arbeit wurde im Rahmen des FWF-Projekts P17024 (*High Resolution Studies of Solar Surface Flows*) erstellt. Dem Projektleiter, Betreuer dieser Dissertation und Architekt des Softwarepakets ANTARES, Herrn Professor Herbert J. Muthsam, danke ich für die ausgezeichnete Betreuung, für die ständige Bereitschaft zu Gesprächen und die daraus resultierenden zahlreichen Anregungen.

Viele Gespräche, insbesondere über numerische und physikalische Probleme, führte ich mit meinem Kollegen Bernhard Löw-Baselli. Dafür, und für seine Weiterentwicklungen des ANTARES-Codes bedanke ich mich. Bedanken möchte ich mich auch bei Herrn Friedrich Kupka für seine Anregungen und seine physikalischen Beiträge zum ANTARES-Code.

Durch Gespräche mit diesen und anderen Wissenschaftlern, wie zum Beispiel Herrn Professor Arnold Hanslmeier, habe ich sehr viel über das wissenschaftliche Arbeiten sowie über das Arbeitsgebiet am Schnittpunkt zwischen Mathematik und Astrophysik gelernt.

Zur Visualisierung der Daten wurde hauptsächlich das von den Kollegen Patrick Lenz und Mathias Langer entwickelte Softwarepaket VIVAT verwendet, für dessen Entwicklung ich mich bedanke.

Die aufwendigen Simulationen wurden an den Hochleistungsrechnern der Fakultät für Mathematik und des Zentralen Informatikdienstes der Universität Wien durchgeführt. Dem Leiter der hiesigen Computergruppe, Herrn Andreas Németh, danke ich für das zur Verfügung stellen und Administrieren der Rechner sowie das Beantworten meiner Fragen.

Voraussetzung für mein Mathematik- und Physikstudium war die Unterstützung meiner Eltern Josef und Martha, einerseits finanziell andererseits durch Anteilnahme und Ratschläge, wofür ich mich herzlich bedanke.

Ohne Freunde, die einem im Leben unterstützen oder einfach nur den Alltag angenehmer gestalten, geht nichts. Danke! Ein besonderer Dank gilt natürlich meiner Freundin Lisi.

Lebenslauf

Name	Christof Obertscheider
Geboren	31. Juli 1978 in Lienz/Osttirol
Nationalität	Österreich
September 1984 - Juli 1988	Volksschule Ainet/Osttirol
September 1988 - Juli 1996	BG/BRG Lienz, Matura mit ausgezeichnetem Erfolg
Oktober 1996 - Mai 1997	Präsenzdienst (Franz-Josephs-Kaserne Lienz)
Oktober 1997 - dato	Studium an der Universität Wien
November 2002	Abschluss des Lehramtsstudiums für Mathematik und Physik mit der Diplomarbeit <i>Essentially Non-Oscillatory Verfahren zur numerischen Lösung der Diffusionsgleichung</i> (Betreuer Prof. Herbert J. Muthsam)
Mai 2004	Abschluss des Diplomstudiums Mathematik
Mai 2004 - dato	Doktoratsstudium der Mathematik im Rahmen des FWF Projekts <i>High Resolution Studies of Solar Surface Flows</i> (Betreuer Prof. Herbert J. Muthsam)
Februar 2004 - April 2004	ERASMUS Aufenthalt an der Università degli Studi di Firenze

Sacherschließung

Sacherschließung