Realistic Simulations of Stellar Surface Convection with ANTARES: I. Boundary Conditions and Model Relaxation

H. Grimm–Strele[a,b,*], F. Kupka[a], B. Löw-Baselli[a], E. Mundprecht[a], F. Zaussinger[c], P. Schiansky[a]

[a]Institute of Mathematics, University of Vienna, Nordbergstraße 15, A-1090 Vienna, Austria
[b]Max–Planck Institute for Astrophysics, Karl-Schwarzschild-Strasse 1, 85741 Garching, Germany
[c]BTU Cottbus, Siemens–Halske–Ring 14, 03046 Cottbus, Germany

Abstract

We have implemented open boundary conditions into the ANTARES code to increase the realism in our simulations of stellar surface convection. Even though we greatly benefit from the high accuracy due to our 5th–order numerical scheme (WENO5), the broader stencils needed for the numerical scheme complicate the implementation of boundary conditions. Moreover, one–dimensional starting models may not provide enough data to set up the simulation domain large enough in each direction, or the initial data may lead to numerical instabilities. We show how to overcome these difficulties and demonstrate to what extent numerical simulations of stellar surface convection are sensitive to the numerical setup and the boundary conditions. A sloppy choice of parameters contained in the boundary conditions can have a severe impact. Only if the initial model, the extent and position of the simulation box, and the parameters from the boundary conditions are chosen adequately, numerical simulations of stellar surface convection are (physically) meaningful and realistic throughout the entire simulation domain.

Keywords: hydrodynamics – methods: numerical – stars – Sun: granulation – convection

1. Introduction

Nowadays, realistic simulations of stellar surface convection are a mature tool of computational astrophysics, and find a wide field of applications. An extensive review on this subject is given in Nordlund et al. (2009). In Beeck et al. (2012), the usefulness and reliability of results obtained from simulations of solar surface convection with the codes CO5BOLD, MURaM and Stagger is shown. Even though numerical methods and the setup of the simulations in this comparison are quite different, the overall stratification and other basic properties of the numerical model are similar.

For any numerical simulation which requires the solution of partial differential equations, the choice of the numerical method, the simulation domain, and appropriate boundary conditions are of major importance. For the simulation of convection at the surface of solar–type stars, the top boundary of the simulation domain is in the upper photosphere, whereas the bottom boundary is situated deeply inside the convective envelope. However, due to the mixed parabolic–hyperbolic nature of the governing equations the boundary conditions influence the solution in the whole simulation domain.

The ANTARES code has originally been developed at the University of Vienna for the simulation of surface convection (Muthsam et al., 2007, 2010a; Lemmerer et al., 2013). Recently, the code was also applied to many other astrophysical problems (Kupka et al., 2009; Muthsam et al., 2010b; Kupka et al., 2012; Happenhofer et al., 2013; Zaussinger and Spruit, 2013; Mundprecht et al., 2013). In some of these applications problems arose due to the use of closed boundary conditions (Kupka et al., 2009; Mundprecht et al., 2013). Shock fronts were reflected at the top boundary, leading to instabilities in the simulation results. Boundary conditions which transmit shock waves rather than reflect them are needed to continue these simulations over a sufficiently long time interval.

Robinson et al. (2003), Kupka and Robinson (2007) and Kupka (2009a,b) discussed, amongst others, the effect of boundary conditions on statistical properties of the flow. Unrealistic boundary conditions can lead to unphysical flow patterns in a huge part of the simulation box. When the bottom boundary is situated in the convective region of the star, which usually is the case, e.g., for solar surface convection simulations, most of the energy is transported in the lower part of the simulation box by advection of enthalpy and kinetic motions. Therefore, free in– and outflow into the simulation domain is necessary to avoid an artificial flow pattern and artificial heating in the lower part of the simulation domain. The entropy of the inflowing material is unknown and has to be specified somehow. At the top boundary, shock fronts should not be reflected, but be transmitted.

[*]corresponding author
Email address: hannes.grimm-strele@univie.ac.at (H. Grimm–Strele)

Even though a lot of literature exists on stellar surface convection simulations (e.g., Nordlund, 1982; Freytag et al., 2012; Vögler et al., 2005), a dedicated investigation on the effect of the setup of boundary conditions on the simulation results appears to be missing. This paper gives a precise description of the numerical setup of our simulations with ANTARES and demonstrates the sensitivity of the simulations to changes in the choice of parameters which are introduced by specifying the boundary conditions.

As we can argue when considering the study of Asplund et al. (2000), meaningful tests of boundary conditions can ultimately only be done in three dimensions, with sufficient resolution and long relaxation time. The two-dimensional geometry leads to systematic differences in the mean stratification such that, for instance, a different inflow entropy is required to obtain the same total flux at the stellar surface. This makes this investigation very expensive in terms of computation time as well as in "real" time.

The results of the present paper provide a necessary extension to comparisons shown in Kupka (2009a) and Beeck et al. (2012), where simulation data from very different simulation runs were compared. Systematically testing every parameter combination in the setup of a numerical simulation, however, is simply not affordable, whence we have to rely on tests of the overall boundary condition with only partial coverage of the parameter space available. In this paper, we show results from several working and non–working combinations and estimate the impact of inappropriately designed boundary conditions.

The remainder of the paper is organised as follows. In Section 2.1, we show the superiority of the WENO algorithm used in ANTARES compared to ordinary first or second order schemes. The gain in accuracy justifies the broader stencils needed for any high–order method. On the other hand, the design of boundary conditions gets more complicated since they extend over several vertical layers. We have implemented several boundary conditions also used in the aforementioned codes paying special attention to the requirements of our high–order numerical scheme. Details are given in Section 2.2. The numerical setup of our simulations, i.e. the starting model and the simulation box size, is described thoroughly in Section 2.3.

We then investigate the dependence of the boundary conditions on parameters like the amount of energy flowing into the domain at the bottom boundary and compare them, taking statistical properties of the flow into account, in Sections 3.1 and 3.2. We show the difficulties arising in the setup of the simulation, its dependence on the initial model (Section 3.3) and on the grid resolution (Section 3.4), and emphasize the importance of a careful choice of parameters. In Section 3.5, we describe differences between simulations with open and with closed boundary conditions. Finally, we compare two– and three–dimensional models in Section 3.6. A discussion of the results can be found in Section 4 followed by our conclusions in Section 5.

2. Simulation Setup

In this section, we describe the implementation of several boundary conditions in the code ANTARES as well as other modifications to the code we have performed in this context.

In this document, the ANTARES convention for spatial coordinates is used, i.e. the x direction points inwards into the star and u is the vertical component of the velocity vector $(u, v, w)^T$. The grid is assumed to be equidistant in every direction.

2.1. Some Comments on the Numerical Method

For the spatial discretisation of the advective part of the Navier–Stokes equations, ANTARES uses the weighted essentially non–oscillatory (WENO) algorithm as described, e.g., in Jiang and Shu (1996) and Shu (2003). WENO schemes exist for any order of accuracy. In ANTARES, we use the fifth order variant which will be called WENO5 in the following (Muthsam et al., 2010a). For the tests discussed in this subsection, flux splitting as described in Donat and Marquina (1996) is not necessary, even though it is implemented in ANTARES and used for the simulations presented in the later sections of this paper.

The advantages of the WENO5 method are its high order of accuracy and its shock–capturing ability. This allows accurate modelling of surface convection where strong shock fronts are ubiquitous in the downflows, whereas the granules themselves are rather smooth (cf. Nordlund et al. 2009). Even when used in combination with lower order methods, e.g. for the time integration, which theoretically degrades the overall order of the method, the WENO5 scheme is superior to classical schemes or other ENO–type schemes of lower order, especially when the latter are combined with artificial diffusivities (Muthsam et al., 2007, 2010a).

We demonstrate the efficiency of the WENO5 method by solving the advection equation

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} = 0 \tag{1}$$

for $t > 0$ and $x \in [0, 1]$ with periodic boundary conditions. With the initial condition

$$\phi\left(x, 0\right) = 1 + 0.1 \sin\left(2\pi x\right), \tag{2}$$

the analytical solution stays smooth for all times. Therefore, this is an appropriate test case for determining the empirical order of accuracy and the error constants of a method. In these test calculations, the Courant number is fixed to 0.1.

From Figure 1 we deduce that the empirical order of accuracy of the WENO5 algorithm together with a second order time integration such as TVD2 (Shu and Osher, 1988) is two which is also obtained with the Lax–Friedrichs

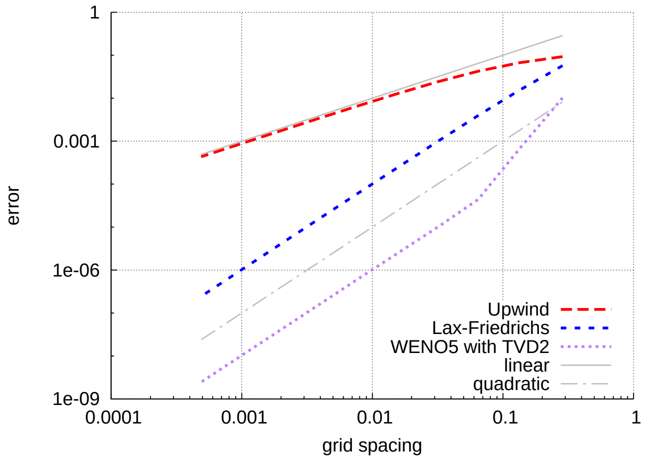

Figure 1: Empirical order of accuracy of several schemes when solving (1) and (2). The thin lines indicating first and second order convergence are given for guidance.

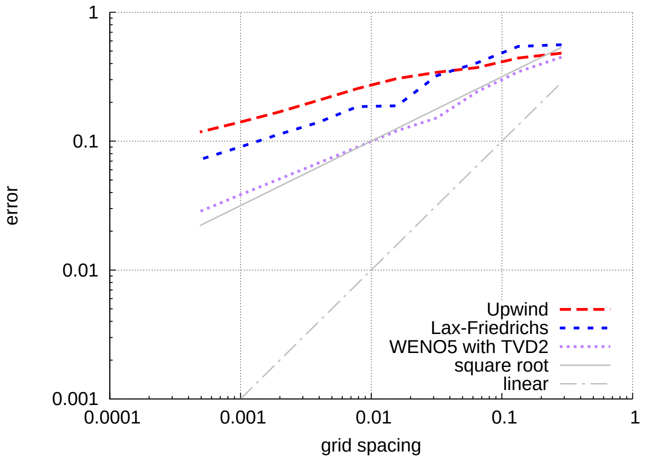

Figure 2: Empirical order of accuracy of several schemes when solving (1) and (3). The thin lines indicating square root and first order convergence are given for guidance.

method (cf. Strikwerda 1989). Nevertheless, the error is two magnitudes smaller than with the Lax–Friedrichs method except for the small region shown to the right where the resolution is very low. There, we observe third to fourth order convergence of the WENO5 method since in this region, the spatial error dominates over the temporal error and the higher–order convergence of the WENO5 scheme is visible. The error of any first order method as, e.g., the upwind method (cf. again Strikwerda 1989), is larger by several magnitudes. A very high amount of grid points is required for them to reach an acceptable error size.

Typically, the magnitude of the (relative) mean error in our simulations is of the order 10^{-3}. With the Lax–Friedrichs method, we would need about six times as many grid points compared to the WENO5 method to obtain the same accuracy. This means that in three dimensions the number of grid points must be more than 200 times higher than with the WENO5 method. Furthermore, this accuracy is only achieved with a much smaller time step size. In the end, for smooth flows we have to expect that the right–hand side has to be evaluated over a 1000 times more often when using the Lax–Friedrichs method in comparison with the combination of TVD2 with WENO5 even at the moderate resolution typical for astrophysical simulations (~ 4 grid points per smallest feature) to achieve the same accuracy. This can be summarised by the combined scheme of WENO5 with TVD2 having a much smaller error constant than the Lax–Friedrichs scheme.

Therefore, the WENO5 method is not only superior in terms of shock–handling, but also in terms of efficiency. In contrast, the disadvantages — higher computational costs per grid point and broader stencils — are negligible.

Given discontinuous initial data,

$$\phi(x,0) = \begin{cases} 1, & \text{if } 0.1 < x < 0.3, \\ 0, & \text{else}, \end{cases} \tag{3}$$

the convergence order is restricted by the smoothness of the (weak) solution. In Figure 2, the order of all three methods is worse than linear. Nevertheless, WENO5 with TVD2 is again much more efficient than the Lax–Friedrichs method: for an error of magnitude 0.1, about six times as many grid points are needed by the Lax–Friedrichs method compared to WENO5. Again, the first–order accurate upwind method performs worst and shows the largest error except for very coarse resolution.

In conclusion, for both smooth and discontinuous solutions the WENO5 method combined with TVD2 time integration is more efficient and more accurate than standard low order methods. This justifies the additional efforts required for its implementation such as adapting boundary conditions to its wider stencils.

From the comparisons done in Muthsam et al. (2007) and in Muthsam et al. (2010a), it follows that WENO5 without artificial diffusivities yields also much more accurate results than other high–order methods which, however, always require such stabilisations. Also in simulations based on the WENO5 numerical scheme unresolved scales can be modelled with a subgrid scale model (e.g., Smagorinsky, 1963). Anyhow this was not necessary for the simulations presented in this paper, since their numerical resolution was rather high and no particular stability problems have been encountered (implicit LES or iLES approach, Grinstein et al. 2007). We also note that the appropriate treatment of diffusive and viscous terms in the context of WENO5 is discussed in detail in Happenhofer et al. (2013).

When applying the WENO method to systems of conservation laws, the state variables must be transformed into the eigenstate which increases the computation time. On the other hand, methods where no transformation is needed are less accurate and artificial diffusivities are necessary to stabilise the solution, e.g. around shock fronts, which at the bottom line is less efficient (Muthsam et al., 2007, 2010a).

2.2. Boundary Conditions for Stellar Convection Simulations

In simulations of surface granulation of cool stars, the top boundary is typically placed above the photosphere whereas the bottom boundary is deep inside the convective envelope (Stein and Nordlund, 1998; Robinson et al., 2003; Wedemeyer Böhm et al., 2004; Freytag et al., 2012; Vögler et al., 2005; Muthsam et al., 2007, 2010a). Numerical boundary conditions either force the fluid to stop its vertical motion and move horizontally without friction (the so called "slip" boundary condition) and hence not cross the vertical boundary, or they allow (constant) in– and outflow. In the case of an inflow, the density and the energy of the inflowing material must be specified.

There is no physical reason why the vertical convective motions should stop at the boundary of the simulation domain as it is forced by closed boundary conditions. To increase the physical realism of the simulation it is mandatory to allow free in– and outflow at the boundaries. On the other hand, implementation and stability of open boundary conditions are non–trivial, as will be shown in the following. In particular, the number of boundary layers must correspond to the width of the stencils of the numerical method used.

For solar–like main–sequence stars, the characteristic scale of surface convection is small compared to the stellar radius and, hence, the "box–in–a–star" approximation is valid for the simulation of surface convection. If the simulation box is broad enough, periodic boundary conditions can be used on the horizontal boundaries of the domain without any negative impact on the simulation (Robinson et al., 2003).

2.2.1. Closed Boundaries

Slip (Stress–free) boundary conditions are characterised by setting the vertical velocity component and the vertical derivative of the horizontal velocities to 0, i.e.

$$u = 0, \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0. \tag{4a}$$

Furthermore, since the bottom boundary can transport energy only by radiation, an artificial source term must be introduced to feed the required amount of energy into the simulation domain. This is done by modifying the thermal conductivity κ in the lowermost layers of the simulation domain such that

$$-\kappa_{\text{modified}} \frac{\partial T}{\partial x} = F_\star = \sigma T_{\text{eff}}^4, \tag{4b}$$

and imposing $\frac{\partial \rho}{\partial x} = 0$. The detailed setup is described, e.g., in Robinson et al. (2003) and Muthsam et al. (2010a).

Closed boundary conditions have the advantage of easy implementation and high stability, at least for subsonic flows. On the other hand, they reflect shocks and disturb the velocity field in an undesirable way. In the context of simulations of stellar surface convection, their unfavourable effect concerning reflection of waves and statistical properties of the flow is discussed in detail, for instance, in Robinson et al. (2003), Kupka et al. (2009), and Kupka (2009a). Nevertheless, there are applications where the use of closed boundary conditions is not problematic, such as the numerical simulation of the superadiabatic layer of the Sun (Kim and Chan, 1998; Robinson et al., 2003; Kupka and Robinson, 2007; Kupka, 2009a).

2.2.2. Open Top Boundary Conditions

The boundary conditions presented in Cheung (2006) and Trampedach (1997) and also those discussed in Freytag et al. (2012) are not directly applicable to ANTARES due to the broader stencil of the WENO5 method and the equidistant grid. Therefore, we had to adapt their approach to our setup.

For the top boundary layers, we assume a hydrostatic and isoenergetic, i.e., in practice, isothermal stratification. As a difference to previous implementations, in the hydrostatic equilibrium we want to include the turbulent pressure since it is quite large in the photosphere. The velocities should be constantly extrapolated. Generally speaking, the boundary conditions should be designed in such a way that the numerical setup does not influence their behaviour.

The vertical and horizontal velocities u, v, and w in the boundary layers are set according to

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0. \tag{5a}$$

For a high–order method, more than one boundary layer is needed. Therefore, we add as many additional nodes on the top of the first boundary layer as necessary for our scheme. In the case of WENO5, three layers are required.

Let 0 be the index of the innermost boundary layer and 1 the one of the outermost domain layer. The mean density in the innermost boundary layer $\langle \rho \rangle_0$ is advanced in time by an averaged midpoint rule centred at the outermost cell interface of the domain, i.e.

$$\langle \rho \rangle_0^{(n+1)} = \langle \rho \rangle_0^{(n)} - \frac{\tau_{\text{stg}}}{\Delta x} \langle \rho u \rangle_{\frac{1}{2}}^{(n)}, \tag{5b}$$

where τ_{stg} is the time step size of the current Runge–Kutta stage and Δx is the (constant) vertical grid spacing. $\langle \rho u \rangle_{\frac{1}{2}}^{(n)}$, the mean density flux at the interface between the layers 0 and 1, can be computed by the spatial integration scheme of its dynamical equation given enough boundary nodes. Now, the density profile from the outermost domain layer is just copied to the innermost boundary layer scaled by the ratio $\frac{\langle \rho \rangle_0^{(n+1)}}{\langle \rho \rangle_1^{(n+1)}}$, i.e.

$$\rho^{(n+1)}(0, y, z) = \frac{\langle \rho \rangle_0^{(n+1)}}{\langle \rho \rangle_1^{(n+1)}} \rho^{(n+1)}(1, y, z). \tag{5c}$$

The specific internal energy $\langle\epsilon\rangle_0$ in all boundary layers is assumed to be constant. It is initialised by the mean specific internal energy in the outermost domain layer $\langle\epsilon\rangle_1$ and is furthermore advanced in time by

$$\langle\epsilon\rangle_0^{(n+1)} = (1.0-\delta)\langle\epsilon\rangle_0^{(n)} + \delta\langle\epsilon\rangle_1^{(n+1)}. \tag{5d}$$

Cheung (2006) set δ constantly to 10^{-3}, but we reset $\langle\epsilon\rangle_0$ only at the end of each Runge–Kutta step with δ given by the formula

$$\delta = \frac{\tau \cdot \langle v_{\mathrm{snd}}\rangle_1}{c_{\mathrm{f}} \cdot \langle H\rangle_1}, \tag{5e}$$

where τ is the time step of one Runge-Kutta integration, $\langle v_{\mathrm{snd}}\rangle_1$ is the mean sound speed in layer 1 and $\langle H\rangle_1$ the mean pressure scale height in that layer. $\delta \in [0,1]$ controls the time scale on which the heat content of the boundary layers varies. With the parameter c_{f}, this time scale can be changed: higher values of c_{f} lead to slower changes, whereas with small values of c_{f} the boundary layers relax much faster to the stratification of the inner cells. Since c_{f} has to be adjusted anyway, it is sufficient to use the gas pressure to compute $\langle H\rangle_1$ for the approximate time scale underlying (5e).

The idea behind equation (5e) is to make the procedure independent of different time integration methods and spatial resolution. With a constant value of δ, the time scale of changing $\langle\epsilon\rangle_0$ would be different, if a two– or a three–stage Runge–Kutta method is used, or if the Courant number is changed. Furthermore, we choose δ independent of the grid spacing such that the time scale does not change, if the grid is modified. Modifying c_{f} allows to control the stability and the stratification of the boundary layers.

The density in the layers above index 0 is set according to

$$\frac{\partial(p+p_{\mathrm{turb}})}{\partial x} = -\rho g, \tag{5f}$$

where $p = p_{\mathrm{rad}} + p_{\mathrm{gas}}$ and $p_{\mathrm{turb}} = \rho\,(u-\langle u\rangle)^2$ is the turbulent pressure. By the chain rule,

$$\frac{\partial(p+p_{\mathrm{turb}})}{\partial x} = \frac{\partial\,(p+p_{\mathrm{turb}})}{\partial\rho}\frac{\partial\rho}{\partial x} = \left(\frac{\partial p}{\partial\rho} + (u-\langle u\rangle)^2\right)\frac{\partial\rho}{\partial x}. \tag{5g}$$

For an ideal gas, constant ϵ implies an isothermal stratification. The slight fluctuations of T, evaluated from the realistic equation of state, therefore show the deviation from the ideal gas equation. Numerical tests of this condition show that it is reasonable to assume the simplification of an ideal gas in the boundary layers, considering the other simplifications like constant velocity and constant ϵ. In this case, $\frac{\partial p}{\partial\rho} = (\gamma-1)\,\epsilon$ is constant in the boundary layer.

Since $(u-\langle u\rangle)^2$ in the boundary layer is constant in the vertical direction due to (5a), the value $c_\rho := \frac{\partial p}{\partial\rho} + (u-\langle u\rangle)^2$ is constant for each horizontal grid index (y,z). Therefore, we can integrate equation (5g) analytically and obtain

$$\rho(i,y,z) = \rho(0,y,z)\exp\left(\frac{-|i|\Delta x \cdot g}{c_\rho}\right), \tag{5h}$$

where i indexes all boundary layers above layer 0. With $\langle\epsilon\rangle_0$ and the velocities calculated by (5a), all values in all boundary layers can be calculated.

2.2.3. Open Bottom Boundary Conditions

At the bottom boundary, we set the velocities according to

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0 \tag{6}$$

to allow constant in– and outflow. In the literature, the density (of an inflow) usually is set to keep the total mass of the model unchanged. Therefore, only an additional condition for the energy is needed. Since the bottom boundary in surface convection simulations is situated in the adiabatic region, a constant entropy value can be assumed for inflows. This idea was first presented in Nordlund and Stein (1990) and is widely used today (Freytag et al., 2012; Vögler et al., 2005).

There are two approaches how the amount of energy of an inflow at the bottom boundary is specified technically: in Nordlund and Stein (1990) and Freytag et al. (2012) for instance, an entropy value is prescribed, whereas, e.g., in Vögler et al. (2005), the specific internal energy is prescribed. Since the correct values are not known from within the problem in both cases, they must be found iteratively or be taken from some external source. For small changes, the relation between ϵ and S is nearly linear, so that it is not important to which of these variables the boundary conditions are applied to. Nevertheless, ϵ increases with depth, whereas S in the deep layers of the convection zone should approach an (almost) constant value. Therefore, prescribing an entropy value has the advantage of being essentially depth–independent, whereas values for the inflowing energy must vary with box depth.

The density in the boundary layers in both cases is set such that the total mass contained in the simulation domain stays unchanged.

Since in a relaxed simulation, the radiative flux at the top of the domain corresponds to the amount of energy flowing through the bottom boundary, Vögler et al. (2005) decided to directly correlate the two values on the Kelvin–Helmholtz time scale. When the radiative flux at the top is too high, the amount of energy flowing in at the bottom is lowered which in principle leads to a lower radiative flux. This motivated the formula

$$\epsilon_{\mathrm{inflow}}^{(n+1)} = \epsilon_{\mathrm{inflow}}^{(n)} \cdot \left(1.0 + \frac{\tau}{\tau_{\mathrm{KH}}}\left(1.0 - \frac{F_{\mathrm{rad}}^{\mathrm{top}}}{F_\star}\right)\right), \tag{7}$$

where $F_\star = \sigma T_{\text{eff}}^4$ is the energy flux of the star according to the Stefan–Boltzmann law and τ the time step size of one Runge–Kutta integration. The Kelvin–Helmholtz time scale τ_{KH} can be calculated by

$$\tau_{\text{KH}} = \frac{\int_{\text{box}} \rho\epsilon \, dV}{\int_{\text{top}} F_{\text{rad}} \, dy \, dz} \tag{8}$$

(cf. formula (19) and (20) in Vögler et al. (2005)).

On the contrary, Nordlund and Stein (1990) decided to choose a fixed value for the entropy of the inflow at the bottom boundary, although it is not discussed how in general this value of S_{inflow} is obtained. In Freytag et al. (2012), the mass and the energy of the inflowing material are corrected in two steps to keep the entropy of the inflow close to S_{inflow} and to reduce pressure fluctuations (see their equations (34) to (37)). They also do not discuss how the value S_{inflow} is actually obtained.

We implemented both approaches which define either S_{inflow} or ϵ_{inflow} into our code and will show results in Section 3.2. In those cases where we specify S_{inflow} at the bottom, we use an iterative correction procedure similar to (7), which is used for ϵ_{inflow}. In the first variant, we only change the time scale and define

$$S_{\text{inflow}}^{(n+1)} = S_{\text{inflow}}^{(n)} \cdot \left(1.0 + \frac{\tau}{\tau_{\text{S}}}\left(1.0 - \frac{F_{\text{rad}}^{\text{top}}}{F_\star}\right)\right), \tag{9}$$

where τ_{S} is the time scale of the correction. In general, $\tau_{\text{S}} \neq \tau_{\text{KH}}$ since the correction procedure is applied to S and not to ϵ as in (7). In the limits $\tau_{\text{S}} \to \infty$ and $F_{\text{rad}}^{\text{top}} \to F_\star$, we arrive at a constant value of S_{inflow} which is effectively the boundary condition that was used by Nordlund and Stein (1990) and Freytag et al. (2012) in the sense that $S_{\text{inflow}} =$ constant.

In the second variant, it is guaranteed that the total flux at the bottom boundary is close to $F_\star$. Here,

$$S_{\text{inflow}}^{(n+1)} = S_{\text{inflow}}^{(n)} \cdot \left(1.0 + \frac{\tau}{\tau_{\text{S}}}\left(1.0 - \frac{F_{\text{tot}}^{\text{bot}}}{F_\star}\right)\right), \tag{10}$$

where $F_{\text{tot}}^{\text{bot}}$ is the sum of the radiative, the kinetic, and the convective fluxes at the bottom of the domain.

None of these correction mechanisms should be applied in the first minutes of the time evolution of a model, since both the radiative flux at the top and the total flux at the bottom require some time to reach the value which reflects the current numerical setup and thermal stratification. Therefore, we do not change S_{inflow} during the first five sound crossing times of a new model started from a one–dimensional stratification.

Specifically, in the first correction step the density ρ and specific internal energy ϵ of an inflow are changed such that

$$S(\rho^{(1)}, \epsilon^{(1)}) = S_{\text{inflow}}, \tag{11}$$

where S_{inflow} is determined either with (9) or (10). Entropy, internal energy, and density of an outflow are not changed. On the other hand, in Freytag et al. (2012) the density and specific internal energy are reset by

$$\rho^{(1)} = \rho + c_{\text{Schange}} \frac{\tau}{t_{\text{char}}} \frac{-\rho^2 T(\Gamma_3 - 1)}{p\Gamma_1} \left(S_{\text{inflow}} - S\right), \tag{12a}$$

$$\epsilon^{(1)} = \epsilon + c_{\text{Schange}} \frac{\tau}{t_{\text{char}}} T \left(1 - \frac{\Gamma_3 - 1}{\Gamma_1}\right) \left(S_{\text{inflow}} - S\right), \tag{12b}$$

(cf. (34) and (35) from Freytag et al. 2012) with the parameter c_{Schange} controlling the time scale of the correction step (further quantities are explained below). In our implementation, however, the entropy of the inflowing material is fixed to $S_{\text{inflow}}^{(n+1)}$ determined by (9) or (10), therefore deleting the parameter c_{Schange} from our formulation of the boundary conditions. In the end, keeping the gas pressure unchanged, we compute $S_{\text{inflow}} := S_{\text{inflow}}^{(n+1)}$ from either (9) or (10) and obtain $\epsilon^{(1)}$ and $\rho^{(1)}$ from inverting (11) using the equation of state.

We then proceed with two additional correction steps proposed by Freytag et al. (2012) which avoid the generation of unwanted large pressure fluctuations by the inflow at the open bottom boundary, and fix the mean mass flux over the boundary to 0. Thus, as in the second correction step from Freytag et al. (2012), density and specific internal energy throughout the entire boundary layers are reset by

$$\rho^{(2)} = \rho^{(1)} + c_{\text{Pchange}} \frac{\tau}{t_{\text{char}}} \frac{1}{v_{\text{snd}}^2} \left(\langle p \rangle - p\right), \tag{13a}$$

$$\epsilon^{(2)} = \epsilon^{(1)} + c_{\text{Pchange}} \frac{\tau}{t_{\text{char}}} \frac{1}{\Gamma_1 \rho} \left(\langle p \rangle - p\right) \tag{13b}$$

(cf. (36) and (37) from Freytag et al. 2012). Here, τ is the time step size of the (Runge–Kutta) integration and t_{char} is given by

$$t_{\text{char}} = \frac{\Delta x}{\langle v_{\text{snd}} + |u| \rangle} \tag{13c}$$

(cf. (33) from Freytag et al. 2012). The velocity of sound v_{snd} as well as the adiabatic coefficients Γ_1 and Γ_3 are taken from the equation of state as a function of the uncorrected values of ρ and ϵ. The parameter c_{Pchange} governs the time scale on which pressure fluctuations are reduced.

Finally, ρ and the vertical velocity u are modified by

$$\rho^{(3)} = \rho^{(2)} + \langle \rho \rangle^{(0)} - \langle \rho^{(2)} \rangle, \tag{13d}$$

$$u^{(1)} = u - \frac{\langle \rho^{(3)} u \rangle}{\langle \rho \rangle^{(0)}} \tag{13e}$$

(cf. (38) and (39) from Freytag et al. 2012) to keep the total mass at the bottom boundary unaltered. Again, this correction is applied throughout the entire boundary layers.

As for the top boundary at the bottom three boundary layers are needed for the WENO scheme used in ANTARES. After the procedure described above is applied to the innermost boundary layer at the bottom, the two underlying layers are filled by exponentially extrapolating the density and by linearly extrapolating the specific internal energy (assuming adiabatic stratification instead would require to proceed as in Sect. 2.3.2 to recover the power law growth with depth, possibly with a correction for turbulent pressure as in (5f) and (5g), but we expect the difference to our simplified procedure along the two outer grid points to be acceptably small).

When implementing the boundary conditions which are similar to Freytag et al. (2012), care must be taken in the formulation of mass conservation. Setting the mean of ρu as a variable to 0 does not enforce mass conservation in flux–based codes such as ANTARES. Instead, the corresponding cell boundary flux must be modified. The system is not overdetermined with this condition, since it is only another numerical enforcement of the (analytical) mass conservation requirement.

Furthermore, as another available alternative to the open bottom boundary conditions specified by (6), (9) or (10), (11) and (13), we implemented the full version of boundary conditions from Vögler et al. (2005) into our code. For the present paper these are used only for the case of Model 2 in Section 3.2. In these boundary conditions, in addition to specifying the inflow internal energy by (7), the pressure is assumed to be uniform across the lower boundary. Its value p_{tot} is determined to keep the mass in the simulation box unaltered (see equations (21) to (23) in the cited reference). In downflow regions, the velocities and the entropy density S are set according to

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0, \; \frac{\partial S}{\partial x} = 0, \tag{14a}$$

(see equations (16) and (17) in the same reference).

In upflow regions, Vögler et al. (2005) force the inflow to be vertical (cf. (18) therein). In contrast,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial w}{\partial x} = 0 \tag{14b}$$

in our implementation. The specific internal energy of an inflow is found by (7) already discussed further above.

Open bottom boundaries can lead to a strong increase in horizontal momenta since, due to (6), they do not conserve mass and momentum simultaneously. To mitigate this effect we usually damp the horizontal momenta with the procedure described in Section 2.3.3 in the layers where the bottom boundary condition is applied.

Table 1 provides a summary of models and boundary conditions which we test in Section 3.2.

2.3. Simulation Domain Size and Initial Conditions

Numerical simulations of stellar surface convection are commonly initialised with one–dimensional models taken from stellar evolution calculations. These models often do not contain enough data points in the photosphere or introduce other deficiencies, which do not allow to apply them directly. In general we have to extend the starting models both at the top and the bottom. In the following, we describe the procedures used for this purpose.

2.3.1. Extending the Simulation Box in the Photosphere

To decrease the influence of the top boundary on the simulation, it is desirable to move the top boundary away from the optical surface towards the upper photosphere or even further above. Since the one–dimensional models used as an initial condition often do not contain data for sufficiently low optical depth and moreover since turbulent pressure and multidimensional radiative transfer cause an elevation of the optical surface in comparsion with one–dimensional models (Nordlund and Stein, 1999; Rosenthal et al., 1999), we developed a procedure to extend the simulation domain when setting up the initial conditions from such models.

The basic variables are re–allocated to the new number of grid points and are pointwise continued to the (new) top of the computational domain by extrapolating the existing values. First of all, the pressure p is exponentially extrapolated from the values in the initial model. Then, the density ρ is set according to the equation of hydrostatic equilibrium

$$\frac{\partial p}{\partial x} = -\rho g, \tag{15}$$

where the derivative of p is calculated numerically with fourth order centred differences (a negative sign for g compensates for x increasing with depth). By calling the equation of state, $T = T(p, \rho)$ is calculated. With the new values of T and ρ the corresponding pressure values are calculated again and the whole procedure is repeated several times to ensure consistency with the equation of state. The velocities are then set according to (5a).

In any case, the initial model for the new layers is only of minor importance, since due to the short time scales in the upper photosphere, the state variables change rapidly away from the initial values. It is indeed more important that this procedure is sufficiently robust. The additional space allows the model to relax to cooler temperatures and reach smaller optical depths.

2.3.2. Deep Models

When we tried to calculate deeper solar granulation models which extend to a depth of about 4 Mm below the surface, we experienced stability problems when the values from the one–dimensional starting model were used for initialisation of the state variables. This probably stems from the fact that the initial model is not numerically adiabatic and hydrostatic when interpolated to the numerical

name	type	detailed description	equations	time scales [h]	parameters
`BC 1`	closed	Muthsam et al. (2010a)	(4)		
`BC 2`	open	Vögler et al. (2005)	(6), (7) & (14b)	$\tau_{\mathrm{KH}} \approx 550$	
`BC 3a`	open	similar to Freytag et al. (2012)	(6), (9), (11) & (13)	$\tau_{\mathrm{S}} \approx 1000$	$c_{\mathrm{Pchange}} = 1.0$
`BC 3b`	open		(6), (10), (11) & (13)	$\tau_{\mathrm{S}} \approx 100$	$c_{\mathrm{Pchange}} = 0.1$

Table 1: Summary of bottom boundary conditions in ANTARES. In the fourth column, the number of the equations where the boundary conditions are described in this paper are given.

grid used in the simulation. The deviations from equilibrium and the ensuing instabilities get larger the deeper the model is. Robinson et al. (2003) experienced similar problems.

Therefore, in the solar case we re–integrate such initial models starting from around 2.5 Mm below the surface using the classical, explicit 4th order Runge-Kutta algorithm to solve the equations of adiabaticity and hydrostatic equilibrium, i.e.

$$\frac{\partial p}{\partial x} = -\rho g, \tag{16}$$

$$\frac{\partial T}{\partial x} = -\rho g \frac{T}{p} \nabla_{\mathrm{ad}}, \tag{17}$$

with ∇_{ad} and ρ for given p and T taken from the equation of state.

Since we have assumed the turbulent pressure to be negligibly small here, the open bottom boundary has to be located sufficiently deep below the superadiabatic layer, where p_{turb} reaches its maximum inside the convection zone (cf., for instance, Tanner et al. 2012). For very deep extrapolations of more than about two pressure scale heights, S_{inflow} can no longer be considered a constant, since it will slightly increase even inside the nearly adiabatic part of the convection zone, in which case necessary changes to S_{inflow} would have to be estimated from solar (resp. stellar) structure models.

With this procedure a stable initialisation of simulations of deep stellar surface convection such as that one of our Sun is readily possible.

2.3.3. Damping Initial Oscillations

Since initial models set up from one–dimensional stellar models numerically are not in perfect hydrostatic equilibrium, every multi–dimensional model starts to oscillate in the vertical direction. These oscillations are not the p–mode oscillations expected to be seen in the simulation of stellar surface convection, but are an artifact introduced by the setup of the initial conditions and must be removed. Therefore, we adapted the procedure described in detail in Trampedach (1997) and Trampedach et al. (2013) in our code to damp unwanted vertical oscillations. The damping term is an additional term in the velocity equation. The damping velocity v_{mode} was defined by

$$v_{\mathrm{mode}} = \frac{\langle \rho u \rangle}{\langle \rho \rangle}. \tag{18}$$

The (vertical) momentum equation takes the form

$$\frac{\partial \rho u}{\partial t} = \text{usual terms} - \rho \frac{v_{\mathrm{mode}}}{t_{\mathrm{mode}}}, \tag{19}$$

where t_{mode} is the period of the mode which should be damped. To be consistent, we also added a corresponding term to the total energy equation such that

$$\frac{\partial E}{\partial t} = \text{usual terms} - \rho u \frac{v_{\mathrm{mode}}}{t_{\mathrm{mode}}}, \tag{20}$$

which improved the stability of this procedure and removed its non–conservativeness with respect to energy (even though the conservation errors of the original method are very small).

This procedure can easily be extended to remove dispensable horizontal momenta by replacing the vertical velocity in the calculation of v_{mode} by its horizontal counterparts $v_{\mathrm{mode},y}$ or $v_{\mathrm{mode},z}$:

$$v_{\mathrm{mode},y} = \frac{\langle \rho v \rangle}{\langle \rho \rangle}, \; v_{\mathrm{mode},z} = \frac{\langle \rho w \rangle}{\langle \rho \rangle}. \tag{21}$$

The time scale t_{mode} can be used to control the rate at which the horizontal momenta are removed. With values of $t_{\mathrm{mode}} \approx 1$ sound crossing time, we found that removing of horizontal momenta, if desired, works very efficiently.

2.4. Asymmetric Stencils at the Domain Boundaries

As described in Ferziger and Perić (2002, p. 53), enforcing conditions which set derivatives to zero as in various prescriptions for the in- and outflow at top and bottom boundaries like (5a), (6), (14a), or (14b) should not be done by simply setting u to a constant value, but by exact inversion of the stencils used for the calculation of the derivatives. Especially, if higher–order methods as, e.g., the fourth order method presented in Happenhofer et al. (2013), are used, unsuitable procedures can lead to pathological behaviour of the velocity field near the boundary.

In general, boundary stencils cannot keep the order and the width of the interior stencils at the same time especially when building higher order derivatives. Either the stencils are broadened to keep the order of the method constant, or the width of the stencils is kept constant leading to a lower–order method. We decided to conserve the order of the method and present the corresponding stencils in the following. The symmetry of the stencils gets lost in any case.

Boundary conditions for ANTARES must be set on three vertical layers. All physical layers are numbered from 1 to n_x starting at the top of the simulation box, such that the boundary layers at the top of the domain get the indices 0, -1 and -2, whereas n_x+1, n_x+2 and n_x+3 correspond to the boundary layers at the bottom.

At the top boundary of the computational domain, derivatives of a scalar–valued function ϕ at the cell centres are calculated by

$$\frac{\partial\phi}{\partial x}(x_{-2}) = \frac{-25\phi_{-2}+48\phi_{-1}-36\phi_0+16\phi_1-3\phi_2}{12\Delta x}, \tag{22a}$$

$$\frac{\partial\phi}{\partial x}(x_{-1}) = \frac{-3\phi_{-2}-10\phi_{-1}+18\phi_0-6\phi_1+\phi_2}{12\Delta x}, \tag{22b}$$

following the approach presented in Happenhofer et al. (2013). For $\frac{\partial\phi}{\partial x}(x_0)$, the symmetric stencil can be used, i.e.

$$\frac{\partial\phi}{\partial x}(x_0) = \frac{\phi_{-2}-8\phi_{-1}+8\phi_1-\phi_2}{12\Delta x}, \tag{22c}$$

where $\phi_i=\phi(x_i)$ and mirrored expressions are valid for the bottom boundary. For $\frac{\partial\phi}{\partial x}(x_{\frac{1}{2}})$ and other derivatives at the cell boundary, similar expressions can be calculated with the same procedure.

Expansion in Taylor series shows that the error term ε is given by

$$\varepsilon\left(\frac{\partial\phi}{\partial x}(x_{-2})\right) = -\frac{1}{5}\left(\Delta x\right)^4\phi^{(5)}(\zeta), \tag{23a}$$

$$\varepsilon\left(\frac{\partial\phi}{\partial x}(x_{-1})\right) = \frac{1}{20}\left(\Delta x\right)^4\phi^{(5)}(\zeta), \tag{23b}$$

$$\varepsilon\left(\frac{\partial\phi}{\partial x}(x_{0})\right) = -\frac{1}{30}\left(\Delta x\right)^4\phi^{(5)}(\zeta), \tag{23c}$$

demonstrating the fourth order of the given stencils. The error constant increases the more the stencils are asymmetric.

If we try to enforce $\frac{\partial\phi}{\partial x}=0$ by setting $\phi_{-2}=\phi_{-1}=\phi_0=\phi_1$, the derivatives will not be zero numerically. Instead, we reset ϕ according to

$$\phi_{-2} = \frac{64}{55}\phi_1 - \frac{9}{55}\phi_2, \tag{24a}$$

$$\phi_{-1} = \frac{63}{55}\phi_1 - \frac{8}{55}\phi_2, \tag{24b}$$

$$\phi_{0} = \frac{64}{55}\phi_1 - \frac{9}{55}\phi_2. \tag{24c}$$

Plugging equations (24) into (22) gives 0.0 for the derivatives numerically, as desired.

While this procedure immediately applies to bottom boundary conditions as specified through (7), (8), (14a), and (14b), for conditions set by (6), (9) or (10) in conjunction with (13), the vertical velocity u in the innermost boundary layer n_x+1 is already fixed by the enforcement of mass conservation (13d). Whereas we can use the procedure from above for the horizontal velocities, the boundary condition for the vertical velocity must be changed to

$$u_{n_x+1} \text{ set to enforce mass conservation,} \tag{25a}$$

$$u_{n_x+2} = \frac{279}{197}u_{n_x+1} - \frac{99}{197}u_{n_x} + \frac{17}{197}u_{n_x-1}, \tag{25b}$$

$$u_{n_x+3} = \frac{252}{197}u_{n_x+1} - \frac{64}{197}u_{n_x} + \frac{9}{197}u_{n_x-1}. \tag{25c}$$

2.5. Calculation of the Energy Fluxes

The calculation of the energy fluxes follows Canuto (1997) which describes the procedure for the case of an ideal gas. We avoid simplifications in the equation of state in the following, and set

$$u'' = u - \frac{\langle\rho u\rangle}{\langle\rho\rangle}. \tag{26}$$

The equivalent holds for v'' and w''. A positive flux means flux towards the top of the box and vice versa.

2.5.1. Radiative Flux ($F_{\rm rad}$)

In optically thin regions, the radiative transfer equation is solved with the short characteristics method as described in Muthsam et al. (2010a). In this approach, the equation is solved by an angular integration over a discrete set of rays. The dependence on the frequency is considered by repeating the calculation several times for averages over frequency sets, to each of which the weight $\omega_{\rm bin}$ is assigned (Nordlund, 1982; Stein and Nordlund, 2003). In every computational node, the intensity $I(\text{bin},\text{ray})$ is calculated for $N_{\rm rays}$ rays, where $N_{\rm rays}$ depends on the quadrature formula chosen for the angular integration, and $N_{\rm bins}$ frequency bins. For the integration rule from Carlson (1963) which is used in the simulations presented in this paper, $N_{\rm rays}$ is 24 in three and 12 in two dimensions due to symmetries. $N_{\rm bins}=1$ corresponds to the grey approximation, and $N_{\rm bins}=4$ is used for non–grey simulations.

The vertical component of the radiative flux is calculated by

$$F_{\rm rad} = \frac{4\pi}{N_{\rm rays}}\sum_{\rm bins}\omega_{\rm bin}\sum_{\rm rays}n_x I(\text{bin},\text{ray}), \tag{27}$$

where n_x is the vertical component of the ray direction. In optically thick regions, the diffusion approximation is valid and $F_{\rm rad}$ can be calculated simply by

$$F_{\rm rad} = \kappa\frac{\partial T}{\partial x}. \tag{28}$$

The sign in front of the thermal diffusivity κ is positive due to the coordinate system chosen in ANTARES.

In most of our simulations, the transition to the diffusion approximation is done at a fixed geometrical depth. We choose this depth such that the transition is in any case performed in an optically thick region, also in case of optically thin downflows.

2.5.2. Convective Flux (F_{conv})

The vertical component of the convective flux is defined by

$$F_{\text{conv}} = \langle u''\rho \left(h - \frac{\langle e+p \rangle}{\langle \rho \rangle} \right) \rangle, \tag{29}$$

where $h = \frac{e+p}{\rho}$ is the specific enthalpy.

2.5.3. Kinetic Flux (F_{kin})

The vertical component of the kinetic flux in three dimensions is defined by

$$F_{\text{kin}} = \frac{1}{2} \langle \rho u'' \left(u''^2 + v''^2 + w''^2 \right) \rangle. \tag{30}$$

When the model is statistically stationary, these fluxes coincide with those defined in Nordlund and Stein (2001). Furthermore, the viscous flux is defined as the horizontal average of the viscous stress tensor as in equation (16) from Nordlund and Stein (2001) or equation (7b) in Canuto (1997).

3. Results

In the following, we show results from a bunch of simulations of solar surface convection with ANTARES. With these data, we demonstrate the advantages and disadvantages of the methods presented in Section 2.

These results can be generalised to other main sequence stars without any fundamental complications. Nevertheless, care must be taken in choosing the boundary condition parameters adequately. Since this paper is focussed on validating the methods from Section 2, we restrict ourselves in the following to the solar case as the best–known example of stellar surface convection.

3.1. Top Boundary Conditions

There is one free parameter in the top boundary conditions described in Section 2.2.2. Changing the parameter c_{f} controls the speed at which $\langle \epsilon \rangle_0$, the value of the specific internal energy in the upper boundary layers, is adapted to the values of ϵ in the inner domain.

In the solar case, typical values of the mean speed of sound $\langle v_{\text{snd}} \rangle_1$ and the mean pressure scale height $\langle H \rangle_1$ in the uppermost domain layer are 7 km/s and 100 km, respectively. Given a time step size per Runge–Kutta integration of around 0.06 s, the value of 10^{-3} given for δ in Cheung (2006) corresponds to $c_{\text{f}} \approx 4.0$.

In two–dimensional numerical experiments we found that the value $c_{\text{f}} = 4.0$ leads to a very slow change in $\langle \epsilon \rangle_0$. The (temporal) standard deviation of the mean temperature stratification σ_t,

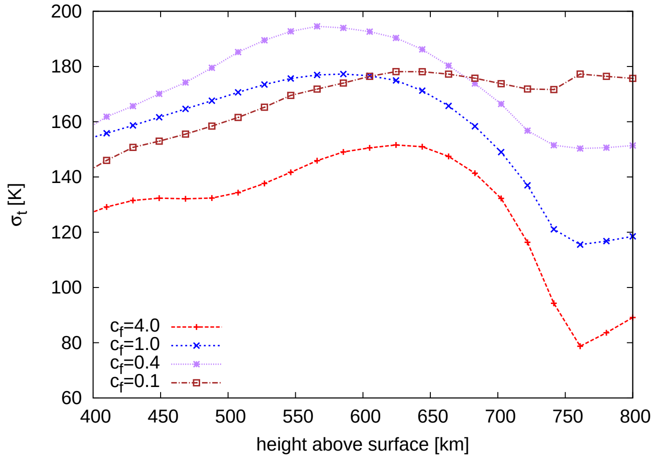

Figure 3: Vertical profiles of σ_t, quantifying the temporal variation of the mean temperature (see (31)), obtained for different values of c_{f} as defined in (5e).

$$\sigma_t \left(T \right) (x) = \langle (\langle T \rangle (x) - \langle \langle T \rangle \rangle_t (x))^2 \rangle_t, \tag{31}$$

where $\langle T \rangle$ is the horizontal and $\langle \langle T \rangle \rangle_t$ the temporal and horizontal average of the temperature T, is plotted in Figure 3. The rightmost three nodes in this figure correspond to the boundary layers. A value of $c_{\text{f}} = 4.0$ decreases the variation to an undesirably small level at the top, which eventually leads to an unrealistic bump in the average $T(\tau)$ relation for layers near the top boundary due to a forced slow relaxation, whereas $c_{\text{f}} = 0.1$ leads to much stronger variations in the same region. We suggest to use a value of $c_{\text{f}} \approx 0.4$ or less. Overall, the effect of c_{f} on the mean temperature stratification throughout the model domain is quite small and restricted to the upper layers of the simulation. Nevertheless, while the choice of large c_{f} can hinder a fast relaxation to a different statistically stationary temperature structure, equation (5e) leads to a time scale which is independent of the numerical setup of the simulation. On the other hand, very small values of c_{f} may lead to strong variations in the temperature which can decrease the numerical stability of the boundary condition.

The value of $c_{\text{f}} = 0.4$ suggested here is a compromise between fast relaxation and robustness with respect to strong shock fronts. We note here that — also for stability reasons — a very “stiff” boundary resulting from $c_{\text{f}} \geq 1$ is not desirable either. Given the numerical justification of this choice however, it has to be expected that for stars very different from the Sun, for instance, Cepheids or A–type stars, c_{f} will have to be adjusted.

3.2. Bottom Boundary Conditions

We compare three 3D solar models with the same simulation setup, which differ mainly in the bottom boundary condition. The simulation boxes extend to 5.2 Mm in the vertical and 9.0 Mm in the horizontal direction. In the vertical direction, the resolution is 15.3 km whereas in the horizontal directions, we chose a resolution of 32.1 km. The term "optical surface" in the following designates the position where the mean temperature gradient is steepest. This occurs at a geometrical depth of around 0.7 Mm to 0.8 Mm from the top of the simulation box. That region is typically located some 0.1 Mm below the layer where $\tau_{\mathrm{ross}} \approx 1$, but is more easily identified when comparing different simulations. For the radiative transfer equation, we used the grey approximation in the upper 30 % of the box, corresponding to a geometrical depth of around 1.6 Mm, and the diffusion approximation elsewhere. The time average covers about one hour.

In Model 2, the bottom boundary condition `BC 2` as in Vögler et al. (2005) was used, with the formula (7) for calculating the energy of the inflow. Model 3a uses the boundary conditions `BC 3a` and Model 3b uses `BC 3b` as summarised in Table 1. They are similar to the boundary conditions of Freytag et al. (2012), but with different values for c_{Pchange} and different formulae for the calculation of S_{inflow}. (10) is used for Model 3b and (9) for Model 3a. Whereas we use $\tau_{\mathrm{KH}} \approx 550\,\mathrm{h}$ as time scale in (7) for Model 2, we chose $\tau_{\mathrm{S}} \approx 100\,\mathrm{h}$ for Model 3b and $\tau_{\mathrm{S}} \approx 1000\,\mathrm{h}$ for Model 3a. Of course, these time scales are not universally applicable, but even for solar convection simulations have to be adjusted to the depth of the simulation box.

3.2.1. Energy Fluxes

The radiative flux at the top of the domain depends on the amount of energy flowing through the simulation domain. The idea of formula (9) therefore is to correct the (unknown) entropy of the inflow at the bottom boundary according to the radiative flux at the top of the domain until the radiative flux has reached the desired value.

Equation (20) in Vögler et al. (2005), i.e. (7) in this paper, defines a similar procedure and corrects the specific internal energy ϵ at the bottom boundary on the Kelvin–Helmholtz time scale τ_{KH}. τ_{KH} increases rapidly with box depth, as shown in Figure 8. For our standard simulations of solar surface granulation with a domain reaching two to four Mm below the optical surface, $\tau_{\mathrm{KH}} \approx 100\,\mathrm{h}$ to $600\,\mathrm{h}$. In contrast, the time scale for the surface granulation is about several minutes.

In Figure 4, the radiative flux at the top of the domain of Model 3a and S_{inflow} calculated with (9) are plotted. Due to the insufficient vertical resolution of the simulation of only 15.3 km and the short relaxation time from the one–dimensional model, among others, the radiative flux stays several per cent too high.

Figures 5 and 6 show the time–averaged convective and kinetic fluxes of Model 2, Model 3a, and Model 3b from above, with the fluxes computed by the formulae (29) and (30), respectively.

It is obvious from Figure 4 that the correction at the bottom boundary does not influence the radiative flux at the top of the domain even after more than two hours of simulation time. During this time, the entropy of the inflowing material is decreased by about 0.4% which is reflected in a completely wrong overall flux at the bottom boundary shown in Figures 5 and 6. This implies that the correction will also not work if applied on even shorter time scales than τ_{KH} because then, the unfavourable corrections due to (9) would be even larger.

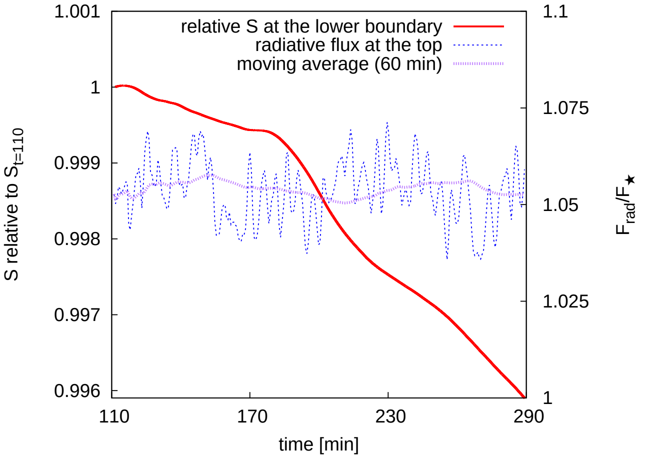

Figure 4: Effect of using a variable inflow entropy as in (9) for calculating S_{inflow} in Model 3a. Radiative flux $F_{\mathrm{rad}}^{\mathrm{top}}$ calculated as horizontal average of F_{rad} in the outermost domain layer and normalised by $F_\star = \sigma T_{\mathrm{eff}}^4$.

We deduce that the coupling of inflowing energy at the bottom boundary and the energy radiated at the top works on much longer time scales than common in simulations of solar and stellar surface granulation. The formulae (7) and (9) require a much longer pre–relaxation time of the model to be effective, or a model which is already relaxed to a suitable three–dimensional stratification.

From Figure 5 we also deduce that blindly employing formula (7) as used in `BC 2` led to a completely wrong convective flux at the bottom boundary. The flux is too large by a factor of 4. In contrast, the energy fluxes from Stein and Nordlund (2000) do not show any anomaly near the bottom boundary. We suppose that the (fixed in time) value for the energy of an inflow in Stein and Nordlund (2000) was well–chosen, whereas we started with the value from the one–dimensional initial model which cannot be expected to be in agreement with the value suited for a three–dimensional simulation, and which in the end invoked an unreasonably strong correction on S_{inflow}.

Due to the insufficient vertical resolution of the simulation, the radiative flux at the top does not approach the desired value no matter how strong the correction at the bottom boundary is — at least not on time scales affordable for a three–dimensional simulation. Since τ_{KH} is that large and the coupling between the radiative flux at the top and $\epsilon_{\mathrm{inflow}}$ or S_{inflow} too weak, the convective flux is forced to unacceptable values by the bottom boundary condition `BC 2`. Similar is valid for Model 3a where (9) is used. There, F_{conv} even changes sign.

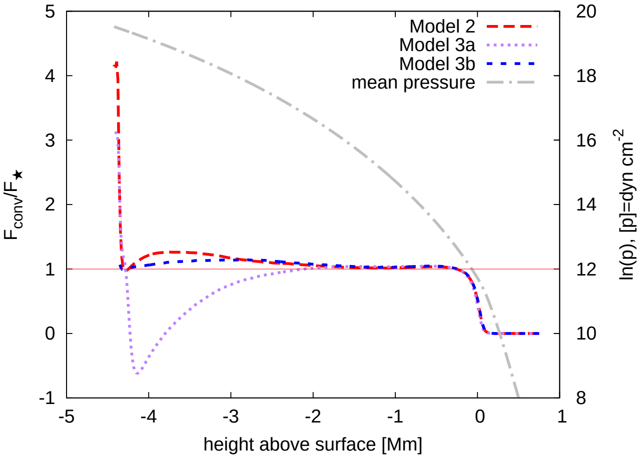

Figure 5: Convective flux of three solar models which differ in the bottom boundary condition, normalised by $F_\star = \sigma T_{\rm eff}^4$. The mean pressure profile from Model 2 is shown, too (the pressure profiles of the models do not differ significantly). The boundary conditions affect the lower 2 Mm, or two pressure scale heights, of the domain.

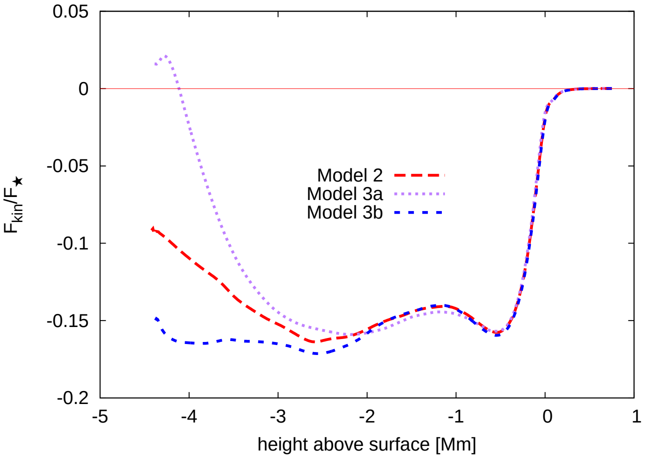

Figure 6: Kinetic flux of three solar models which differ in the bottom boundary condition, normalised by $F_\star = \sigma T_{\rm eff}^4$.

This weak coupling is also demonstrated by the fact that the radiative fluxes of all three models do not differ significantly, even though the differences in the lower part of the simulation domain shown in Figures 5 and 6 are exorbitant. To be more precise, in the lower 2 Mm of the models, or two pressure scale heights, the fluxes are influenced in a very direct and strong way by the boundary condition despite their main difference is the determination and change of $\epsilon_{\rm inflow}$ (resp. $S_{\rm inflow}$) as a function of time (Models 3a and 3b also differ in the damping rate of pressure fluctuations at the bottom boundary, $c_{\rm Pchange}$). Near the surface however, the differences between the three models are insignificant.

We note here that the simulations presented in Vögler et al. (2005) have much more shallow domains with a vertical extent of typically just 1.4 Mm of which only 0.8 Mm are located below a mean depth for which the optical depth is 1. As a result, in their case $\tau_{\rm KH} \approx 1$ to 2 h. For this special case we can expect a much tighter coupling of bottom boundary and $F_{\rm rad}^{\rm top}$ than we observe here for our much deeper models because the distance between the bottom boundary and the optical surface is only about two pressure scale heights in which case a correction procedure as given by (7) can still be effective.

We hence prefer `BC 3b` as used in Model 3b, since these boundary conditions keep the energy fluxes in the lower part of the domain at the desired level and do not produce any serious artifacts even if the initial value of $S_{\rm inflow}$ is not chosen very well. The variation over one time step is small for this procedure. If desired, a (practically) constant input entropy can be obtained by choosing $\tau_{\rm S} \gg \tau_{\rm KH}$.

Since we do not use any artificial diffusive terms in our simulations presented here, the mean viscous flow is very small such that we decided not to include it in our plots.

3.2.2. Dependence on Parameters of the Boundary Conditions

The parameter $c_{\rm Pchange}$ in Freytag et al. (2012) defines a time scale on which pressure fluctuations at the bottom boundary are damped. $\tau_{\rm S}$ controls the time scale on which $S_{\rm inflow}$ is changed. We tried two values for both parameters and found that they influence the behaviour of the boundary conditions considerably.

In Figure 6, the kinetic flux of Model 3a goes to 0 near the bottom boundary giving a completely wrong flow pattern in the lower half of the simulation domain. For Model 3a, the parameter $c_{\rm Pchange}$ from Freytag et al. (2012) was fixed to 1.0 and $\tau_{\rm S} \approx 1000$ h in (10). The only reasonable fluxes are found in Model 3b, where the formula (10) was used for the calculation of $S_{\rm inflow}$ with $\tau_{\rm S} \approx 100$ h and where $c_{\rm Pchange}$ was set to 0.1. It is striking to see how large the influence of these parameters can be. Model 2 uses `BC 2`.

In Figure 7, the distribution of the vertical velocity of the three models in three vertical layers with fixed depth and scaled by the maximum velocity in this layer is depicted, i.e. the distribution density function of the dimensionless variable u_{scaled} with

$$u_{\mathrm{scaled}}(x, y, z) = \frac{u(x, y, z)}{\max_{y,z} |u(x, y, z)|}. \tag{32}$$

In the layer corresponding to the uppermost image in Figure 7, which is situated 1.3 Mm below the optical surface, the differences between the three models are very small. As expected, the distribution has a significant skewness due to the asymmetry between the faster, narrow downflows and the slower upflows. Note the shift of the maximum induced by the broad upflows (and the fully compressible treatment of the flow). Going deeper and therefore closer to the bottom boundary the differences between the models become more prominent. In the bottom panel, the distribution of Model 3a has much narrower tails and its skewness is much smaller than in Model 2 or Model 3b, whereas the velocity distribution of Model 2 and Model 3b is in all three layers very similar (as we expect it to be). On the other hand, there are some deviations visible for Model 2 while for Model 3b, if scaled to the same maximum value, i.e.

$$u^{\star}_{\mathrm{scaled}}(x, y, z) = \frac{u_{\mathrm{scaled}}(x, y, z)}{\max_{x,y,z} |u_{\mathrm{scaled}}(x, y, z)|}, \tag{33}$$

the distribution is almost invariant as a function of depth apart from a gradual broadening observed for layers farther away from the optical surface. By modifying only the correction formula for S_{inflow} and two parameters in the bottom boundary conditions, we changed the velocity distribution of Model 3a in the lower third of the simulation domain considerably: the distribution approaches a narrow Gaussian one with a very flat, additional tail and, in the end, resulting in a kinetic energy flux of about 0.

Comparing our results with the data from Stein et al. (2009), who performed simulations of the upper part of the solar convection zone in boxes of up to 20 Mm depth, we conclude that there is no physical reason for the kinetic flux to go to 0. On the contrary, it should even slightly increase in magnitude when going deeper down into the convection zone — even though that trend in the kinetic flux in the lower part of their simulation box may also be influenced by the boundary conditions. Anyway, the kinetic fluxes of Models 2 and 3b are much more realistic than the one of Model 3a.

Furthermore, we conclude from Figure 3 in Stein et al. (2009), that the assymetry between up– and downflows persists when going deeper into the convection zone, while simultaneously the dominant horizontal scale of the convection increases. This means that the velocity distribution function in each vertical layer should have the same shape when scaled to the maximum velocity value in this layer.

Returning to the differences between Model 2 and Model 3b, in the bottom panel of Figure 7, the peak of the

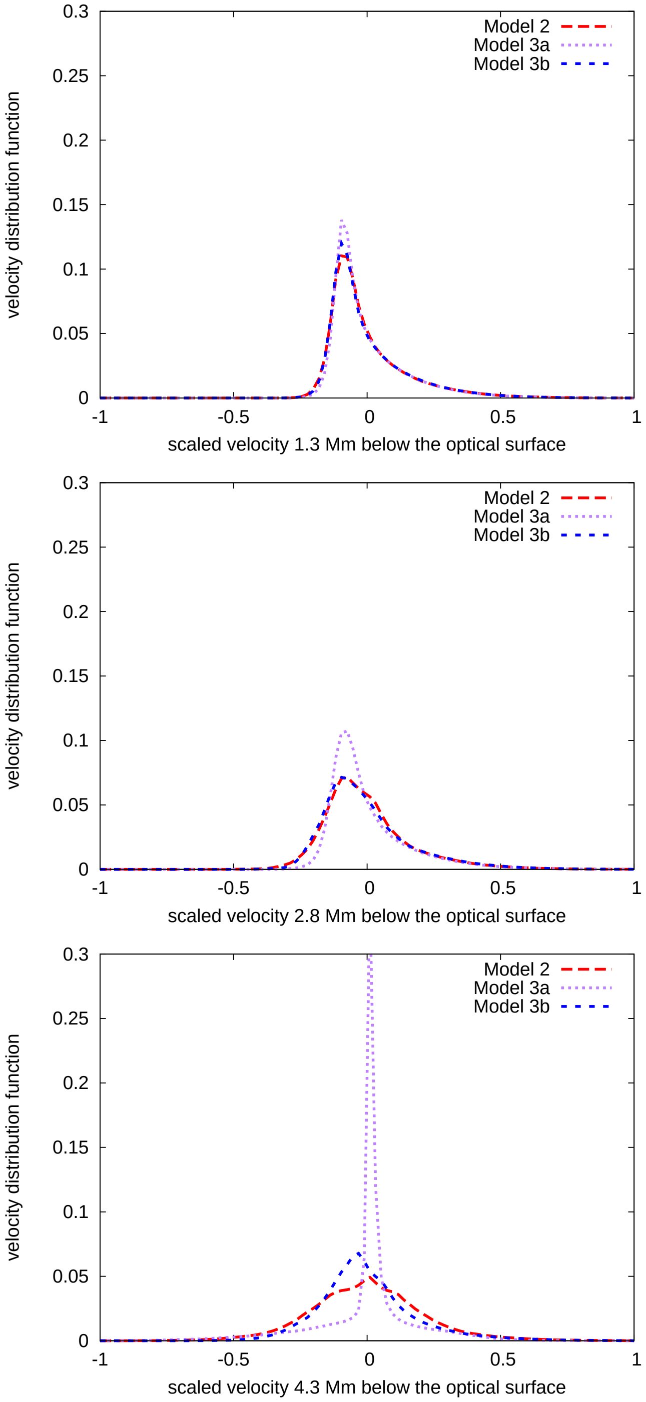

Figure 7: Scaled velocity distribution 1.3 Mm, 2.8 Mm and 4.3 Mm below the optical surface (from top to bottom). The velocity is scaled by the maximum velocity in the corresponding layer. For theses pictures, the velocities in each of the layers were grouped into 96 equal–sized bins and scaled by the maximum velocity in this layer. Finally, the distribution function was normalised by the number of nodes and time steps such that the integral from −1 to 1 is 1. Due to the coordinate system chosen for ANTARES upflows imply $u(x, y, z) < 0$.

scaled velocity distribution of Model 2 is close to a velocity of 0 whereas for Model 3b, the skewness is not changed significantly. The same can also be observed when comparing more shallow models such as those summarised in Table 2. What we observe for Model 3b is hence in agreement with Stein et al. (2009), which is not the case for Models 2 and 3a as their behaviour as a function of depth may be modified by just changing the depth of the simulation box. This makes no physical sense, as we require the open bottom boundary condition to influence the flow as little as possible. Altogether, this indicates that Model 3b, and therefore `BC 3b` is superior.

Since both values tested for τ_{S} are rather large, we assume that they do not influence the results as does the change in the value of c_{Pchange} on the one hand and the change from (9) to (10) on the other. Individual testing of the parameters, however, would have to be done also in three dimensions and with sufficient numerical resolution, which has not been affordable in terms of computation time. Anyway, the combination of $\tau_{\mathrm{S}} \approx 100\,\mathrm{h}$ and $c_{\mathrm{Pchange}} = 0.1$ yields sensible results and we choose them as new default values for these parameters.

From the comparison of the fluxes we deduce that the domain of influence of the bottom boundary condition includes the lower third of the simulation domain, in this case about 2 Mm or 1 to 2 pressure scale heights, coinciding with the findings in Kupka (2009b, p. 95) and suggested in Kupka and Robinson (2007) for the case of closed boundary conditions. The fluxes in the upper two thirds of the domain are nearly identical for all three models, demonstrating that at least the surface layers and the superadiabatic region are not strongly influenced by the boundary conditions, if the safety margin is large enough.

3.3. Dependence on Initial Model

In the following, we compare three solar 3D models which differ only in their initial conditions, three different 1D solar structure models. The first one, called *modelS* in the following, is based on a semi–empirical model. It combines an MLT model of convection (Böhm Vitense, 1958) with $\alpha_{\mathrm{MLT}} = 1.99$ with a semi-empirical model for the photosphere. It is our standard initial model and was used for all other simulations of solar surface convection in this paper. The second one is a purely theoretical model. Convection is modelled by means of an MLT model with $\alpha_{\mathrm{MLT}} = 1.766$. Finally, the third one uses the CGM model of convection (Canuto et al., 1996) with $\alpha_{\mathrm{CGM}} = 0.69$. Both theoretical models are calculated with the CESAM code and use the grey Eddington approximation for the photosphere. See Morel (1997) and Morel and Lebreton (2008) for a description of the CESAM code and Samadi et al. (2006) for the implementation of the convection model.

Since the photospheric layers contained in the stellar structure of the models computed with the CESAM code are much more shallow than *modelS*, we extended the box of all three models such that the computational domain has about the same size, i.e. 6 Mm in the horizontal and 4 Mm in the vertical direction and such that the optical surface is at about 1 Mm geometrical depth. All models treat radiative transfer with the opacity binning method with 4 bins and were relaxed for 80 to 90 min with low spatial resolution. Then, the following 30 min were used for analysis. During this time the resolution was about 11 km in the vertical and 35 km in the horizontal directions. All models use the open boundary conditions `BC 3b` at the bottom.

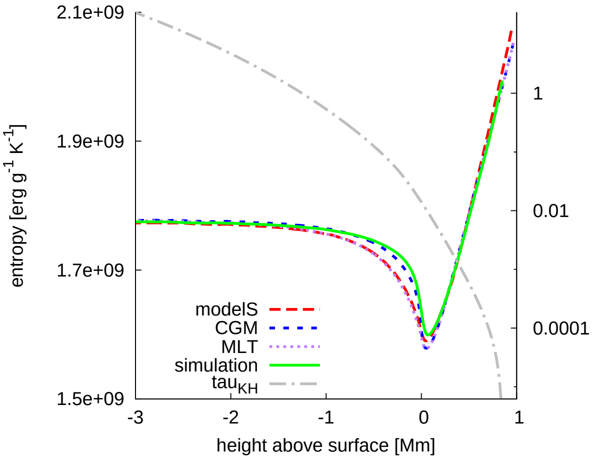

Figure 8: Entropy of the three initial models (modelS, CGM, MLT) and of the relaxed numerical simulation as well as τ_{KH} as a function of depth. The data for the initial models are obtained from an average over the first 0.6 s of each three–dimensional model. Since the simulations based on these initial conditions all feature similar relaxed profiles, only one representative simulation is shown in its relaxed state, too.

As depicted in Figure 8, the initial entropy profiles differ mainly in the layers from 1 Mm below the surface up to the surface itself. At the bottom boundary, the differences in entropy are much smaller. Immediately below the superadiabatic peak, the three profiles from the relaxed simulations, however, are very similar to the CGM initial model. As they do not differ considerably from each other, only one profile is shown in Figure 8. Since τ_{KH} is about 1 to 2 h in this region, the pre–relaxation time was long enough to allow the simulations to adjust to a common stratification. Near the surface (at about 150 km), the simulation results differ from all three one–dimensional models. On the other hand, when we look at Figure 9, we see that $\nabla - \nabla_{\mathrm{ad}}$ from the simulation is much closer to the MLT and the *modelS* profiles, a result which may be unexpected when looking at the seemingly small differences in the entropy profile near its minimum (Figure 8). We can understand this result by taking into account the elevation of the optical surface in the fully developed three dimensional simulation (Nordlund and Stein, 1999). This process moves the region of nearly constant entropy upwards while the averaged superadiabatic gradient remains less steep in the very narrow region around its maximum.

In all three models, once relaxed, the entropy profile

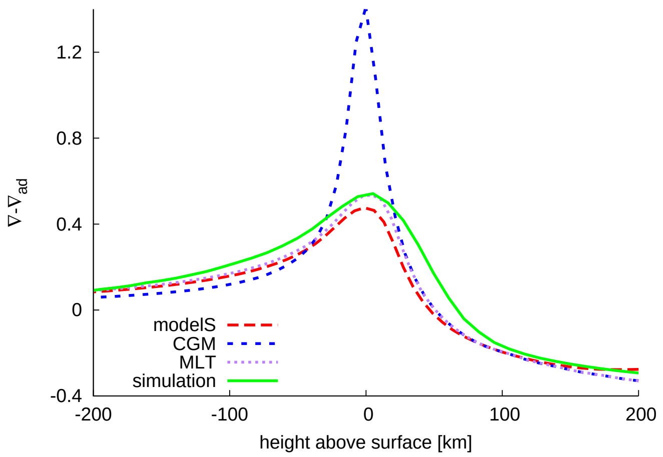

Figure 9: $\nabla - \nabla_{\rm ad}$ of the three initial models and of the numerical simulation as discussed in Fig. 8. The data for the initial models are obtained from an average over the first 0.6 s of the three–dimensional model. The integration rule from Carlson (1963) was used for the angular integration in the radiative transfer solver. The logarithmic temperature gradient ∇ is computed with respect to the total (i.e. including turbulent) pressure.

as well as the radiative flux at the top do not differ significantly from each other. Thus, differences in the superadiabatic layer exhibited by the initial models are not important, if the simulations are run long enough in terms of $\tau_{\rm KH}$ for that region. From Figure 8 we deduce that $\tau_{\rm KH}$ is about 1 to 2 h in the region where the entropy profiles of the initial models differ, which corresponds to the simulation time used for pre–relaxation.

3.4. Resolution Dependence

To reach the correct radiative flux at the top the resolution of the simulation must be high enough and the initial model must be well-suited. From numerical experiments we deduce that a vertical resolution of at least 10 to 11 km is necessary for this purpose, if the integration rule of Carlson (1963) is used for the angular integration. These numbers are slightly different depending on the type of binning used for the radiative transport, and depend also on the quadrature formula used to perform angular integration.

The resolution dependence is demonstrated in Table 2, where a bunch of grey and non–grey models are compared with respect to the radiative flux at the top of the domain. All of these models differ from the ones mentioned above in terms of simulation domain size, duration of the simulation and grid resolution. Whereas the grid resolution is specified directly in Table 2, the simulation domain is always between 3.8 Mm to 4.1 Mm in the vertical and 6.0 Mm in the horizontal direction. All models are three-dimensional and the radiative transfer equation is solved along 24 rays using the quadrature rule of Carlson (1963) for the angular integration, but the number of bins differs. The grey approximation corresponds to the case of one bin, whereas the non–grey models were calculated with four bins. The bottom boundary conditions are specified in Table 2. Finally, the model must reach far enough into the photosphere. We found that about 800 km above the optical surface are sufficient.

binning	resolution [km]	b. c.	$T_{\rm eff}$ [K]
grey	19.5×40.0^2	`BC 2`	5959.8
grey	13.0×40.0^2	`BC 2`	5835.8
grey	13.0×40.0^2	`BC 3b`	5834.7
grey	11.0×35.3^2	`BC 3b`	5813.7
non–grey	13.0×40.0^2	`BC 3b`	5884.2
non–grey	11.0×35.3^2	`BC 3b`	5868.1
non–grey	6.5×20.0^2	`BC 2`	5765.3

Table 2: Effective temperature calculated from $F_{\rm rad}^{\rm top} = \sigma T_{\rm eff}^4$ in dependence of binning method, numerical resolution and boundary conditions. The observed effective temperature of the Sun is 5777.6 K. Boundary conditions are explained in Table 1.

$F_{\rm rad}^{\rm top}$ primarily depends on the vertical resolution. In the grey case, $F_{\rm rad}^{\rm top}$ is slightly lower than in the non–grey case where we used 4 bins. We emphasize that the rate of change of $F_{\rm rad}^{\rm top}$ with numerical resolution depends on the angular quadrature rule. The rule by Carlson (1963), which does not include a vertical ray, requires a very high resolution to yield the correct radiative flux (only the model with 6.5 km vertical resolution yields a $T_{\rm eff}$ very close to the solar one).

Conversely, the dependence of $F_{\rm rad}^{\rm top}$ on the lower boundary conditions is weak. Table 2 does not show any systematic difference with respect to the bottom boundary conditions. Changes occur only on much longer time scales.

We note that the horizontal resolution of all simulations presented here is sufficient to get the correct effective temperature and radiative flux at the top of the domain (cf. Asplund et al. 2000; Robinson et al. 2003).

3.5. Comparison with Closed Boundary Conditions

The main effect of an open bottom boundary is that it permits a free vertical outflow. In the case of closed boundaries, the fluid cannot penetrate the bottom boundary and energy is transported into the domain only by an artificial radiative source term.

In Figure 10, the energy fluxes of two similar models, one with closed boundaries `BC 1` and one with open boundaries `BC 3b`, are compared. The time average extends over two hours for the model with closed boundaries, and over one hour for the model with open ones. Since the resolution of the model with open boundary conditions was higher, the data was interpolated to the coarser grid of the model with closed boundaries. The resolution changes the radiative flux near the surface, similarly to what was shown in Table 2, thus explaining the step in $F_{\rm conv}$ near a height of 0 km. In the following, we only discuss the differences in the kinetic and convective fluxes.

Compared to the model with open boundary conditions the convective and the kinetic flux stay smaller in magni-

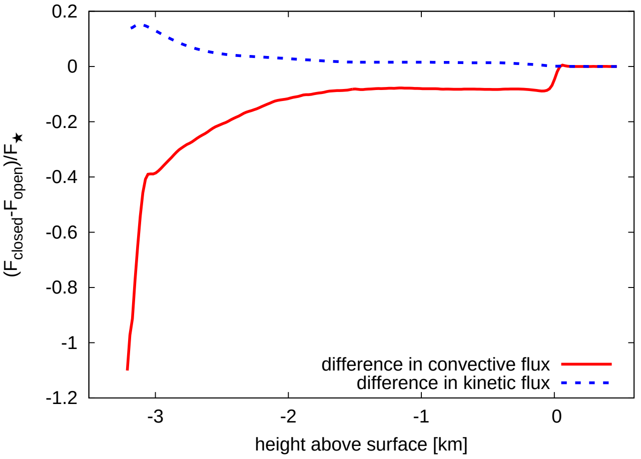

Figure 10: Difference in the energy fluxes of a model with closed and a model with open boundary conditions normalised by $F_\star = \sigma T_{\rm eff}^4$.

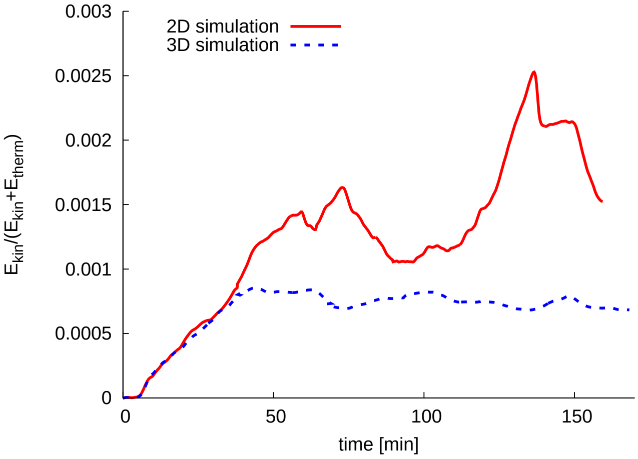

Figure 11: Temporal evolution of the normalised total kinetic energy for a two– and a three–dimensional simulation.

tude — keeping in mind their opposite sign. The simulation with closed boundary conditions is not as well relaxed as its counterpart with open boundary conditions, which is at least partially due to the insufficient resolution.

Around 1 Mm above the bottom boundary, the kinetic flux from the model with closed boundaries reaches a saturated state. The differences to the model with open boundaries is very small. On the contrary, the effect of the closed boundary on the convective flux is much larger. In about half of the simulation box, the convective flux differs considerably from the one with open boundaries (besides from the general offset stemming from the insufficient vertical resolution). The difference between the convective fluxes exhibits a slope for about 2 Mm starting from the bottom of the simulation box. The closed boundary condition hence also influences about 2 pressure scale heights, as is shown in Figure 5 for the case of open boundary conditions (Model 3a). However, in addition, at the bottom boundary itself, the stratification of the model with closed boundaries is stable and energy is transported into the domain by an artificial source term (4b) (Muthsam et al. 2010a, cf. also Robinson et al. 2003) which explains the behaviour of the fluxes near the bottom.

3.6. Two–Dimensional Models

As demonstrated in Figures 4 and 8, thermal relaxation of a simulation takes place on very long time scales, comparable in magnitude up to the local Kelvin–Helmholtz time scale (Kupka, 2009b) unless the mean thermal structure is known in advance (we return to this issue below). Thus, if a good initial model is not available, a three–dimensional simulation may not be run long enough with reasonable effort such that it can relax properly.

Since the computational costs for a two–dimensional model are much lower, one is tempted to relax the model in two dimensions first and switch to three dimensions as soon as the correct amount of energy is contained in the simulation domain. Unfortunately, we have found that two– and three–dimensional models of solar surface convection behave quite differently during relaxation.

In Figure 11, the time evolution of the total kinetic energy $E_{\rm kin}$ scaled by the sum of kinetic and thermal energy of a two– and a three–dimensional simulation both starting from the same one–dimensional model is shown. In theory, the kinetic energy should saturate as soon as the convection is fully developed, and the simulation should reach a quasi–stationary state. Obviously, this happens quite rapidly in the three–dimensional case, whereas the kinetic energy of the two–dimensional model starts to oscillate with a very high amplitude and without showing any signs of saturation.

From Figure 11 we deduce that a two–dimensional simulation intended as a precursor of a three–dimensional simulation should not be conducted for a long time and can at most be used to pre–calculate the few first sound crossing times of the model. Even then, it can be hard to get rid of the artifacts of the two–dimensionality, especially if the model reaches deep into the convection zone. When we tried to convert a two–dimensional model which was run for a long time (around 60 sound crossing times) to a three–dimensional one, we encountered serious problems destroying the symmetry in the third dimension in the deeper layers, and the relaxation to a truly three–dimensional state was not shorter than when the model were started directly from the one–dimensional initial model. On the contrary, the model inherits a lot of the deficiencies of the two–dimensional model such as statistical non–stationarity and an overall higher vorticity. Relaxing the model to a true three–dimensional state took at least as long as when starting from a one–dimensional model. This was independent of the type and strength of perturbation we used to break the two–dimensional symmetry.

Asplund et al. (2000) pointed out that a two–dimensional simulation of solar granulation leads to different stratifications than a three–dimensional simulation and that the results from two–dimensional simulations are qualitatively wrong concerning line profiles and element abundances. Furthermore, they emphasized that two– and three–dimensional simulations need different values for the entropy of the inflowing material (about 2% difference). As a consequence, it appeared advisable to us to avoid using two–dimensional simulations for testing the bottom boundary conditions since the resulting entropy profiles will differ from their three–dimensional counterparts.

Nevertheless, pre–computing a model in two dimensions can be useful if the two–dimensional phase does not take too long, and if the model is shallow such that the time scale of motion at the bottom boundary is still quite short. For instance for the solar case, the first hour after starting from the one–dimensional initial model may be calculated in two dimensions, if the model is not deeper than 4 Mm.

If the stratification of the one–dimensional model is far away from that one of the multi–dimensional model, it might take a very long time for the model to relax to the new quasi–stationary state. In this case, the horizontal and temporal average of a long two–dimensional simulation could be used as a one–dimensional starting model for the three–dimensional simulation. In this way, the problem of breaking the symmetries and high vorticity is avoided, provided one accepts to discard with any a priori information about the velocity field.

Figure 11 furthermore demonstrates that the thermal stratification of a model is quite robust against kinetic motions since the kinetic energy reaches only about 0.1% of the thermal energy. This ratio is comparable to the ratio of τ_{sound} to τ_{KH} and thus essentially the time scale of convective transport relative to the time scale of simulation relaxation from our “arbitrary” initial conditions (see Sect. 4).

Confirming the results from Section 3.3 convective motions would need a long time to change the thermal stratification of the entire model, especially since quasi–adiabaticity keeps the sizes of the fluxes of kinetic energy and enthalpy within the same order of magnitude. A significantly different stratification could of course lead to a higher total flux which in turn could reduce the relaxation time somewhat. Still, a good starting model is an essential prerequisite for cost efficient numerical simulations.

In Figure 12 another difference between two– and three–dimensional simulations is shown. There, the relative share of the horizontally averaged turbulent pressure in the innermost boundary layer of the open top boundary condition is plotted. The simulations are the same as in Figure 11. In both cases, $c_{\text{f}} = 4.0$, but we tested several different values and found no noticeable variation. The difference in the relative contribution of turbulent to total pressure is significant: in the three–dimensional case, it fluctuates around 10% whereas in two dimensions, the turbulent pressure sometimes even exceeds the sum of the gas and the radiative pressure. In the mean, its share is around 30% demonstrating once more the unstable and non–stationary nature of the two–dimensional simulation for the upper layers of the (solar) photosphere.

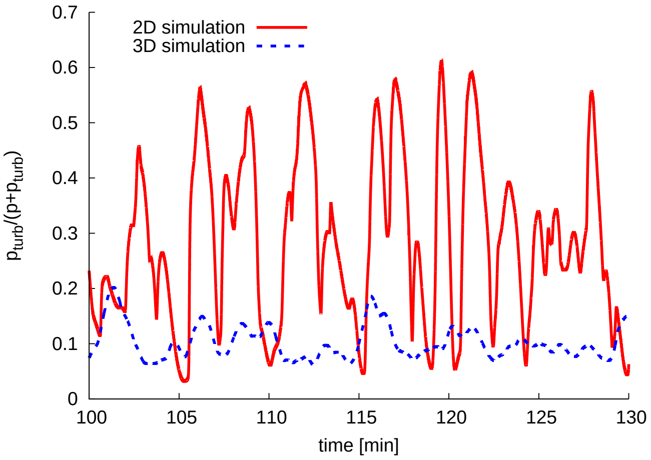

Figure 12: Temporal evolution of the ratio $p_{\text{turb}}/(p + p_{\text{turb}})$ in the innermost boundary layer for the same simulations.

In both cases, the share of the turbulent pressure in the boundary layers is not negligible. Using (5f) is therefore definitely an improvement compared to using the hydrostatic equation without turbulent pressure. Nevertheless, Figure 12 demonstrates that any static approximation of these layers is rather unrealistic.

In comparison with Figure 1.3 (bottom) in Steffen (2000), the difference between two and three dimensions is of a similar magnitude, even though the layer depicted in Figure 12 is even higher in the photosphere than the data from Steffen (2000).

4. Discussion

The physical realism of simulations of stellar surface convection depends strongly on the boundary conditions imposed at the top and the bottom of the simulation domain. In a large part of the simulation domain, the flow is influenced in a direct way by the boundary conditions. Therefore, the simulation box must be wide enough in all directions, and the boundary should be open to allow free in– and outflow. The influence of the lower boundary conditions is particularly strong on the lowermost two pressure scale heights. With closed boundary conditions, the flow pattern is disturbed such that the results are physically not relevant in this region (cf. also Kupka and Robinson 2007; Kupka 2009b). Open boundary conditions are clearly preferable as they can increase the realism of the simulation throughout the entire domain considerably, but even with open boundary conditions we recommend to ensure a distance of at least two pressure scale heights between the lower boundary and the superadiabatic layer.

At the top of the domain, time scales are very short such that the combined influence of the top boundary conditions and the initial stratification of the layers in that region on the stratification of the model throughout most of the simulation box is rather small provided that it is located sufficiently far above the optical surface (cf. Fig. 1

in Kupka 2009a). The boundary conditions (5) assume an isoenergetic (isothermal) and hydrostatic stratification, including the effect of turbulent pressure. Since the velocities are set constant in the top boundary layers, waves are not reflected, but transmitted, which increases the long–term stability of the model. The formulation of the top boundary conditions allows variations in the temperature away from the initial profile. In our implementation, the parameter c_f controls the stiffness of the open boundary conditions in terms of temporal variations of the temperature, as shown in Figure 3. The optimal value has to be found by numerical experiments and may vary depending on the specific application.

The most difficult part in designing open boundary conditions at the bottom of the simulation domain is to specify the amount of energy of an inflow. This value controls the overall stratification of a surface convection model. But the coupling with the energy radiated at the top of the domain is rather weak and for deep, nearly adiabatic convection zones it operates on time scales which are not feasible to be calculated with multi–dimensional simulations.

In deep stellar convection zones the only mechanism which can change the stratification of a model is convection. But to change the energy content of the lowest 1 Mm of one of our solar models by only 1%, convection will need about 1 h, while this time scale increases rapidly with box depth. The Kelvin–Helmholtz time scale τ_KH may not only be interpreted as the time it takes until the whole gravitational energy contained in the simulation box is radiated, but also as the time it takes convection to rearrange the complete stratification of a (in our case nearly adiabatic) model. This results from the fact that (gravitational) potential energy completely dominates the total energy balance. For radiative (and conductive) zones a derivation of why τ_KH can be used as an approximation to the *thermal adjustment time*, which is the actual quantitiy of interest here, can be found in Chapter 5.3 of Kippenhahn and Weigert (1994), where also limitations of this approximation are pointed out. For the case of convection zones the approximation $\tau_\mathrm{KH} \approx \tau_\mathrm{adj}$ is derived in Chapter 6.4 of the same reference as part of a general stability analysis for stellar material at a certain (arbitrary) depth. As illustrated with Figure 8, differences in the entropy of the initial model within the superadiabatic region disappear within the local τ_KH whereas the stratification of simulations over shorter relaxation times is determined by the initial model. The success of one–dimensional models in predicting the nearly adiabatic stratification for most but about the upper 1 Mm of the solar convection zone is essential for the efficiency of hydrodynamical simulations and for the short relaxation times of 1 to 2 hours of numerical simulation of solar granulation reported in the literature.

For simulations of solar convection, any correction mechanism which correlates the bottom boundary directly with the top boundary will be inefficient, if the simulation box spans more than 1 or 2 Mm of depth below the optical surface. On the other hand, in shallow simulation boxes the boundary conditions will directly influence the superadiabatic layer. The simulation box must be large to keep the influence of the bottom boundary small. This, however, prohibits a direct correction mechanism of the entropy at the bottom boundary based on the radiative flux on the top. Consequently, the input entropy (resp. internal energy) is much better controlled by a local criterion like (10). We prefer to impose the boundary conditions on the entropy since its value is nearly constant with depth, such that the boundary conditions can be applied at a range of geometrical depths using the same parameter values. Criteria like (7) or (9) which couple the upper boundary to the lower boundary can lead to wrong energy fluxes as shown in Figures 5 and 6, since the correction at the bottom does not affect the upper boundary in the time covered by the simulation. In terms of Table 1, our preferred boundary conditions are `BC 3b`. These boundary conditions allow free in– and outflow with an adiabatic inflow, while fixing the mean mass flux over the bottom boundary to 0. We refrain from using `BC 2` and `BC 3a` for future simulations.

In the formulation of open boundary conditions, several parameters must be set which influence the behaviour of the flow near the physical boundaries considerably. But, since the computational costs for each single simulation of surface convection are high, systematic tests of the specific effect of each parameter are expensive. Nevertheless, we found a reliable choice for all parameters at least for the solar case, which should be widely applicable to stars of the central and lower main–sequence.

The computational costs cannot be reduced by performing simulations only in two dimensions, since the results differ considerably. As stated also in Asplund et al. (2000), two–dimensional simulations lead to different stratifications because of the different nature of the kinetics of the model. Furthermore, longer simulation times are needed for statistical significance, since the area coverage of two–dimensional simulations is smaller — which dilutes the gain in computation time again. We have found that in addition two–dimensional simulations of solar surface convection do not reach an energetic equilibrium on the same time scale as three–dimensional ones and the upper photosphere is influenced by much stronger fluctuations of turbulent pressure. As the symmetries in the flow-field are also difficult to break by applying a perturbation to initialise a three–dimensional simulation, the use of two–dimensional models as precursors of three–dimensional ones should hence be limited to only five to ten sound crossing times. Alternatively, horizontal averages may be used to construct one–dimensional initial models, if the latter cannot be constructed in a reliable way otherwise.

The numerical resolution influences directly the radiative flux at the top of the domain. As we show by numerical experiments in Table 2, the accuracy of the radiative transfer solver can be improved by increasing the numerical resolution on the hydrodynamical grid. If the resolution is sufficiently high, the radiative flux and therefore the effective temperature is in agreement with the observed values. When the angular quadrature rule by Carlson (1963) is used, a vertical grid spacing of at least 10 to 11 km is necessary for the solar case, otherwise the effective temperature of the model will be too high. Therefore, simulations with lower numerical resolution will behave differently, and a realistic simulation of surface granulation must de done in sufficiently high resolution. If the aspect ratio of each computational cell is very different from 1, the magnitude of the numerical viscosity varies in dependence of the coordinate direction, and surfaces which are inclined with respect to the coordinate system are not well resolved. Therefore, increasing the vertical resolution must be accompanied by an increase in horizontal resolution as well, which makes the simulation much more expensive.

The required resolution depends on the angular integration rule used in the radiative transfer solver. Tanner et al. (2012) discussed that the local radiative flux (in a fixed grid point) is highly variable in time and in the inclination angle. Therefore, choosing a suitable integration rule for the angular dependence of the intensity function is crucial to accurately determine the radiative flux, and different formulae will give different results. Only in the limit of more and more ray directions these results will converge.

In contrast to the angular integration rule from Tanner et al. (2012), but also Stein and Nordlund (2003), the integration rule used in our simulations (Carlson, 1963) does not have a vertical ray. But at least in the centre of a granule, the radiative flux is dominated by the vertical direction. If only few rays are used in the angular integration, as it is common in stellar surface convection simulations, a vertical ray should be included in the integration scheme. We will show in a forthcoming paper, how choosing an angular integration rule changes the radiative flux of a numerical simulation of stellar surface convection. After all, the effective temperature (as well as its superadiabicity) of a model depends primarily on the radiative transfer at the surface and only weakly on changes of the entropy at a bottom boundary far away from the surface.

5. Conclusions

The design of open boundary conditions has to be done with great care, especially for high–order methods, and their influence on the simulation results can be large. In this paper, we investigate the shortcomings of several approaches and discuss how to implement boundary conditions for a high–order method which allow realistic simulations of stellar surface granulation.

In a large part of the simulation domain, the properties of the flow are strongly influenced by the boundary conditions for which closed boundary conditions give unphysical flow patterns. Hence, open boundary conditions are desirable, except for simulations made for specific applications where the flow structure is of limited interest, such as the determination of $\nabla - \nabla_{\mathrm{ad}}$ in the superadiabatic layer.

At the top of the domain, open boundary conditions can transmit waves originating at the stellar surface whereas closed boundaries reflect them. At the bottom boundary, the value of the inflowing entropy must be specified for open boundaries and it controls the overall stratification in the simulation box. Some of the proposed approaches, which couple the input flux at the bottom boundary to F_{rad} at the top of the domain (Vögler et al., 2005), work only for shallow boxes where the Kelvin–Helmholtz time scale is small. Instead, F_{rad} at the surface of a model is sensitive to the numerical resolution and, as was discussed recently in Tanner et al. (2012), to the direction and number of rays used in the radiative transfer solver. Consequently, the entropy of the inflow at the bottom boundary should be controlled by a local criterion. In a forthcoming paper, we will demonstrate the difference in radiative flux resulting from using different formulae for angular integration in the radiative transfer solver.

Due to the large influence of the boundary conditions, it is important to choose the simulation domain sufficiently large in all directions. We show how to extend the box if the one–dimensional initial models do not provide enough data, whence the simulations can be initialised in a numerically robust way. Great care also has to be taken in how to specify either the density, pressure, or momentum of the inflow at the bottom boundary, since this can modify the fluxes and even the flow topology in the convection zone to the same amount as it is done by closed boundary conditions. Open bottom boundary conditions can be expected to excel over closed ones only, if they are designed along these principles and combined with a suitable, local criterion for the inflow entropy. We have discussed one such successful setup called `BC 3b` in Section 3.2.

Numerical simulations of stellar surface convection are expensive in terms of computation time, since they are intrinsically three–dimensional and need high resolution. In two dimensions, kinetics are qualitatively different, as shown in Section 3.6 which extends earlier results on stratification discussed in Asplund et al. (2000). Finally, the simulations have to be run for a long time to relax to an intrinsically three–dimensional state. Since they must not be influenced any more by the initial model, this limits the usefulness of two–dimensional surface convection simulations of the Sun and solar–like stars as precursors of three–dimensional ones to short initialisation runs and to the computation of mean structure models useful when no reliable one–dimensional models are available.

The high computational requirements of each single simulation of stellar surface convection prohibits systematic testing of each parameter in the formulation of the open boundary conditions, even though the behaviour of the latter depends sensitively on them. A larger range of parameters and configurations of boundary conditions could also be investigated by comparing the results from different simulations codes. Nevertheless, we have found a reliable choice for both top and bottom boundaries while we have demonstrated that a poor choice can have a severe impact on a large part of the simulation domain. The best performing setup presented in this paper has become the default for numerical simulations of stellar surface convection with ANTARES.

Acknowledgements

We acknowledge financial support from the Austrian Science fund (FWF), projects P20973 and P20762. FK acknowledges support by the FWF grant P21742. HGS wants to thank H. Muthsam for helpful discussions and the MPA Garching for a grant for a research stay in Garching. Calculations have been performed at the VSC clusters of the Vienna universities and the Heraklit cluster of the TU Cottbus. The model with closed boundary conditions has been calculated at RZG. We would like to thank G. Houdek for making *modelS* and R. Samadi for making the MLT and the CGM initial models available to us. We are thankful to K. Belkacem and J. Ballot for carefully reading the manuscript and suggesting a number of improvements.

References

Asplund, M., Ludwig, H.-G., Nordlund, A., Stein, R. F., 2000. The effects of numerical resolution on hydrodynamical surface convection simulations and spectral line formation. A&A 359, 669 – 681.

Beeck, B., Collet, R., Steffen, M., Asplund, M., Cameron, R. H., Freytag, B., Hayek, W., Ludwig, H.-G., Schüssler, M., 2012. Simulations of the solar near-surface layers with the CO5BOLD, MURaM, and Stagger codes. A&A 539, A121.

Böhm Vitense, E., 1958. Über die Wasserstoffkonvektionszone in Sternen verschiedener Effektivtemperaturen und Leuchtkräfte. Zeitschrift für Astrophysik 46, 108 – 143.

Canuto, V. M., 1997. Compressible Turbulence. ApJ 482, 827 – 851.

Canuto, V. M., Goldman, I., Mazzitelli, I., 1996. Stellar Turbulent Convection: A Self-consistent Model. ApJ 473, 550 – 559.

Carlson, B. G., 1963. The numerical theory of neutron transport. Vol. 1 (Statistical Physics). Methods in Computational Physics. Advances in Research and Applications, Ch. 1, pp. 1 – 42.

Cheung, C. M. M., 2006. Magnetic flux emergence in the solar photosphere. Ph.D. thesis, Universität Göttingen.

Donat, R., Marquina, A., 1996. Capturing Shock Reflections: An Improved Flux Formula. JCP 125, 42 – 58.

Ferziger, J. H., Perić, M., 2002. Computational Methods for Fluid Dynamics, 3rd Edition. Springer, Berlin.

Freytag, B., Steffen, M., Ludwig, H.-G., Wedemeyer Böhm, S., Schaffenberger, W., Steiner, O., 2012. Simulations of stellar convection with CO5BOLD. JCP 231, 919 – 959.

Grinstein, F. F., Margolin, L. O., Rider, W. J., 2007. Implicit large eddy simulation: computing turbulent fluid dynamics. Cambridge University Press.

Happenhofer, N., Grimm Strele, H., Kupka, F., Löw Baselli, B., Muthsam, H. J., 2013. A low Mach number solver: Enhancing applicability. JCP 236, 96 – 118.

Jiang, G.-S., Shu, C.-W., 1996. Efficient Implementation of Weighted ENO Schemes. JCP 126, 202 – 228.

Kim, Y.-C., Chan, K. L., 1998. A Hydrodynamic Simulation of the Highly Superadiabatic Layer of the Sun. ApJL 496, L121.

Kippenhahn, R., Weigert, A., 1994. Stellar Structure and Evolution, 3rd Edition. Springer Berlin, Berlin.

Kupka, F., 2009a. 3D stellar atmospheres for stellar structure models and asteroseismology. Mem.S.A.It. 80, 701 – 710.

Kupka, F., 2009b. Turbulent Convection and Simulations in Astrophysics. Springer Lecture Notes in Physics 756, Ch. 3, pp. 49–105.

Kupka, F., Ballot, J., Muthsam, H. J., 2009. Effects of Resolution and Helium Abundance in A Star Surface Convection Simulations. Comm. in Asteroseismology 160, 30 – 63.

Kupka, F., Happenhofer, N., Higueras, I., Koch, O., 2012. Total–variation–diminishing implicit–explicit Runge–Kutta methods for the simulation of double–diffusive convection in astrophysics. JCP 231, 3561 – 3586.

Kupka, F., Robinson, F., 2007. On the effects of coherent structures on higher order moments in models of solar and stellar surface convection. MNRAS 374, 305 – 322.

Lemmerer, B., Utz, D., Hanslmeier, A., Veronig, A., Thonhofer, S., Grimm Strele, H., Karyappa, R., 2013. 2D Segmentation of small convective patterns in RHD simulations. Submitted to A&A.

Morel, P., 1997. CESAM: A code for stellar evolution calculations. Astron. Astrophys. Suppl. Ser. 124, 597 – 614.

Morel, P., Lebreton, Y., 2008. CESAM: a free code for stellar evolution calculations. Astrophysics and Space Science 316, 61 – 73.

Mundprecht, E., Muthsam, H., Kupka, F., 2013. Multidimensional realistic modelling of Cepheid-like variables. I: Extensions of the ANTARES code. Submitted to MNRAS, available at http://arxiv.org/abs/1209.2952.

Muthsam, H. J., Kupka, F., Löw Baselli, B., Obertscheider, C., Langer, M., Lenz, P., 2010a. ANTARES – A Numerical Tool for Astrophysical RESearch with applications to solar granulation. NewA 15, 460 – 475.

Muthsam, H. J., Kupka, F., Mundprecht, E., Zaussinger, F., Grimm Strele, H., Happenhofer, N., 2010b. Simulations of stellar convection, pulsation and semiconvection. In: Brummell, N., Brun, A., Miesch, M., Ponty, Y. (Eds.), Astrophysical Dynamics: From Stars to Galaxies. No. 271 in Proceedings IAU Symposium. pp. 179 – 186.

Muthsam, H. J., Löw Baselli, B., Obertscheider, C., Langer, M., Lenz, P., Kupka, F., 2007. High–resolution models of solar granulation: the two–dimensional case. MNRAS 380, 1335 – 1340.

Nordlund, A., 1982. Numerical simulations of the solar granulation. I - Basic equations and methods. A&A 107, 1 – 10.

Nordlund, A., Stein, R. F., 1990. 3-D simulations of solar and stellar-convection and magnetoconvection. Computer Physics Communications 59, 119 – 125.

Nordlund, A., Stein, R. F., 1999. Convection Simulations. In: Gimzénez, A., Guinan, E., Montesinos, B. (Eds.), Theory and Tests of Convection in Stellar Structure. Vol. 173 of ASP Conference Series. pp. 91 – 102.

Nordlund, A., Stein, R. F., 2001. Solar Oscillations and Convection. I. Formalism for Radial Oscillations. ApJ 546, 576 – 584.

Nordlund, A., Stein, R. F., Asplund, M., 2009. Solar Surface Convection. Living Rev. Solar Phys. 6, 1 – 117, cited on January 16, 2013.

Robinson, F., Demarque, P., Li, L. H., Sofia, S., Kim, Y.-C., Chan, K. L., Guenther, D. B., 2003. Three-dimensional convection simulations of the outer layers of the Sun using realistic physics. MNRAS 340 (3), 923 – 936.

Rosenthal, C., Christensen Dalsgaard, J., Nordlund, A., Stein, R. F., Trampedach, R., 1999. Convective contributions to the frequencies of solar oscillations. A&A 351, 689 – 700.

Samadi, R., Kupka, F., Goupil, M., Lebreton, Y., van't Veer Menneret, C., 2006. Influence of local treatments of convection upon solar p mode excitation rates. A&A 445, 233 – 242.

Shu, C.-W., 2003. High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD. International Journal of Computational Fluid Dynamics 17 (2), 107 – 118.

Shu, C.-W., Osher, S., August 1988. Efficient implementation of essentially non-oscillatory shock-capturing schemes. JCP 77 (2), 439 – 471.

Smagorinsky, J., 1963. General Circulation Experiments with the Primitive Equations. I. The Basic Experiment. MWR 91, 99 – 164.

Steffen, M., 2000. 2D numerical simulation of stellar convection – An overview. In: Cheng, L., Chau, H., Chan, K., Leung, K. (Eds.), Stellar Astrophysics. Proceedings of the Pacific

Rim Conference held in Hong Kong, 1999. pp. 25 – 34, available at http://www.aip.de/groups/sternphysik/stp/publications_neu.html.

Stein, R. F., Nordlund, A., 1998. Simulations of Solar Granulation. I. General Properties. ApJ 499, 914 – 933.

Stein, R. F., Nordlund, A., 2000. Realistic Solar Convection Simulations. Solar Physics 192, 91 – 108.

Stein, R. F., Nordlund, A., 2003. Radiative Transfer in 3D Numerical Simulations. In: Hubeny, I., Mihalas, D., Werner, K. (Eds.), Stellar Atmosphere Modeling. Vol. 288 of ASP Conference Proceedings. pp. 519 – 532.

Stein, R. F., Nordlund, A., Georgoviani, D., Benson, D., Schaffenberger, W., 2009. Supergranulation–Scale Convection Simulations. In: Dikpati, M., Arentoft, T., Hernández, I. G., Lindsey, C., Hill, F. (Eds.), Solar-Stellar Dynamos as Revealed by Helio- and Asteroseismology. Vol. 416 of ASP Conference Proceedings. pp. 421 – 426.

Strikwerda, J. C., 1989. Finite Difference Schemes and Partial Differential Equations. Wadsworth & Brooks/Cole.

Tanner, J. D., Basu, S., Demarque, P., 2012. Comparing the Effect of Radiative Transfer Schemes on Convection Simulations. ApJ 759, 120 – 131.

Trampedach, R., 1997. Convection in Stellar Atmospheres. Master's thesis, Aarhus University, available at users-phys.au.dk/art/papers/speciale.pdf.

Trampedach, R., Asplund, M., Collet, R., Nordlund, A., Stein, R. F., 2013. A Grid of 3D Stellar Atmosphere Models of Solar Metallicity: I. General Properties, Granulation and Atmospheric Expansion. ApJ 767, (2), available at http://arxiv.org/abs/1303.1780.

Vögler, A., Shelyag, S., Schüssler, M., Cattaneo, F., Emonet, T., Linde, T., 2005. Simulations of magneto-convection in the solar photosphere. A&A 429, 335 – 351.

Wedemeyer Böhm, S., Freytag, B., Steffen, M., Ludwig, H.-G., Holweger, H., 2004. Numerical simulation of the three-dimensional structure and dynamics of the non-magnetic solar chromosphere. A&A 414, 1121 – 1137.

Zaussinger, F., Spruit, H., 2013. Semiconvection: numerical simulations. A&A, in press, available at http://arxiv.org/abs/1303.4522.