Mem. S.A.It. Vol. 88, 12

Memorie della

Boundary conditions in CO5BOLD

Bernd Freytag

Division of Astronomy and Space Physics, Department of Physics and Astronomy, Uppsala University, Box 516, SE-75120 Uppsala, Sweden, e-mail: `Bernd.Freytag@physics.uu.se`

Abstract. The declaration of boundary conditions is a crucial step in the setup of a `CO5BOLD` simulation (and many others) due to the physical nature of the problem, that is reflected in the mathematical description by partial differential equations, discrete versions of which are integrated by the numerical solver(s). While parameters controlling the flux of energy through the computational box are most important for all simulations of convective flows, the detailed specifications describing the behavior of energy, gas and dust densities, velocities, and magnetic fields at or just beyond the boundaries influence the flow, dynamics, and stratification within the box. Recent refinements of the treatment of boundary conditions in `CO5BOLD` resulted in reliably working implementations of open and *closed* versions for top, bottom, and "inner" boundaries even under conditions with strong velocity fields (waves, shocks, or downdrafts). They are implemented and available in the current version of `CO5BOLD` – but have to be activated properly with parameters adapted to the type of the star under consideration (by defining for instance the depth of the damping layers for the closed-bottom boundary or by specifying the damping constants for the open-bottom boundary).

Key words. Stars: atmospheres – Numerical methods: hydrodynamics

1. Introduction

The radiation-magneto-hydrodynamics code `CO5BOLD` (Freytag et al. 2012) can simulate the atmospheres of cool stars and substellar objects, together with the convective surface layers below. It has been used successfully to model the atmospheric flows of objects from A-type stars to brown dwarfs and from white dwarfs to red supergiants (Freytag et al. 2012; Tremblay et al. 2013).

There are separate modules for solving the hydrodynamics equations (with an approximate Riemann solver of Roe type, see Roe 1986) and the magneto-hydrodynamics (MHD) equations (with an HLL solver, see Harten et al. 1983). Both setups account for compressibility, ionization effects, and gravity. Two different geometrical setups are implemented: the one for local "box-in-a-star" models of small patches of the stellar surface requires the use of the `MSrad` radiation transport module, that uses a long-characteristics Feautrier scheme and accounts for horizontally periodic boundary conditions (see e.g. Wedemeyer et al. 2004). The other setup is intended for performing global "star-in-a-box" simulations of the outer stellar convective envelope of red supergiants (Freytag et al. 2002; Steffen & Freytag 2007; Chiavassa et al. 2009) and stars on the asymptotic giant branch (Freytag & Höfner 2008), and for toy models of rotating dwarfs. It uses a short-characteristics scheme (`SHORTrad` module). Both account for non-local radiation-transport, i.e., optically thin or thick conditions, and frequency-dependent opacities via an opacity-binning scheme. Various optional modules account for molecule or dust formation (e.g., Freytag & Höfner 2008; Freytag et al. 2010, 2017).

Beeck et al. (2012) compared solar models computed with the CO5BOLD, MURaM, and Stagger codes. The small discrepancies in the "look" of granules was attributed to different numerical resolutions, while velocity fields in the upper photosphere varied somewhat in behavior due to different choices of the upper boundary conditions.

An overview about the choices of boundary-condition settings in the current CO5BOLD version 003.00.2016.05.16 will be given in the following.

2. Boundary conditions

2.1. The role of boundary conditions

CO5BOLD solves a system of partial differential equations for the (magneto-) hydrodynamics part plus integro-differential equations for the radiation transport, complemented by a formula for the gravity field, a table for the equation of state (EOS), and another table for the opacities (cf. Freytag et al. 2012). Besides the initial values (the start model), the material specifications (EOS, opacities), and parameters for the gravity field, the specifications of the boundary conditions describe the physical system to be simulated. A number of additional parameters control numerical details: e.g.,the reconstruction scheme of the hydrodynamics solver (Freytag 2013), the number of rays of the radiation-transport solver, or the strength of the optional tensor viscosity.

Physically, the hydrodynamics boundary conditions describe density, internal energy, and velocity fields at or just outside the borders of the computational domain. Solutions of the MHD equations need a specification of the magnetic-field components, with the additional complication that magnetic monopoles must be absent. The radiation transport requires information about the intensity of light entering the computational box. In general, the boundary conditions control the flow of conserved quantities (mass, energy, momentum, magnetic fields) into and out of the box.

Mathematically, the boundary conditions fix some quantities to certain values (e.g., the velocities to zero) or they specify the derivatives at the border of the computational domain.

Several strategies can be followed to fulfill boundary conditions *numerically*.

2.2. Boundary conditions in CO5BOLD

Various ways are used in CO5BOLD to implement boundary conditions, sometimes separately, sometimes in combination. Often, *ghost* cells (or *halo* or *buffer* cells) are added on all sides of the computational domain and filled with values of the basic quantities (density ρ, internal energy ei, velocity $\boldsymbol{v}$) according to the boundary conditions. For instance, a model with 300^3 grid points can be stored in CO5BOLD as a number of arrays with 306^3 elements, where 3 elements on each side are needed for 2nd-order reconstruction schemes (Freytag 2013) – and less for simpler schemes. In this way, stencils crossing the borders of the computational box can be treated with the very same numerical operator as stencils located completely within the box. However, in some cases, boundary fluxes computed within the numerical solvers are set explicitly to specific values (e.g., zero). Some boundary conditions are implemented by modifying values *within* the computational domain (e.g., for the heating of material or to ensure zero overall mass flux through the bottom of the box). The radiation-transport solvers need start values for the integration of the intensity through the computational box.

In the CO5BOLD parameter file, boundary conditions are not – and cannot be – specified for all six sides of the computational box individually but only for the top, the bottom, and all the side boundaries (for which the MHD solver allows different options in the two horizontal directions). While for local models, the latter are usually just periodic, the lower and upper boundaries have to allow the stellar energy flux to pass.

However, they are often located in regions with strong velocity fields, where as few artefacts as possible should be introduced to the flow field. In addition to the "classical" open variants `bottom_bound=inoutflow` and `top_bound=transmitting` (e.g., in all models of the CIFIST grid, see Ludwig et al. 2009), there are now closed variants `bottom_bound=closedbottom` and `top_bound=closed` working even in cases where the lower boundary is located within a convection zone or when shock waves hit the upper boundary (Freytag et al. 2010, 2017).

Ideally, the boundaries in CO5BOLD simulations should behave as if they were not there. That means that the boundary conditions are supposed to mimic (essential features of) the deep interior, the chromosphere, the wind-formation zone, or the rest of the stellar surface. This is not an easy task.

All solvers in CO5BOLD that handle spatial derivatives (e.g., for hydrodynamics or long-characteristics radiation transport, but not for the treatment of dust source terms) react to the specified values.

Some values of boundary-condition parameters are kept the same for a large range of model stars (e.g., periodic side boundaries for all local models, the open lower boundary for many cool local models, the top open upper/outer boundary condition for many local and all global models). Others need some adaptation when switching from one to another class of stars (e.g., dimensionless rates) or might have to be adjusted in each individual case (e.g., scale heights, buffer-zone widths).

The following section describes the boundary conditions that are useful to compute dynamical stellar atmospheres, which always have some kind of gravity field. Additional boundary conditions are implemented for test purposes only, for instance several variants that require zero gravity or vertical periodic boundaries (see the CO5BOLD user manual: Freytag et al. 2015).

3. Hydrodynamics (Roe) solver

3.1. Side boundaries

In local models, closed – i.e., reflective – side boundaries attract downdrafts, most obviously in low-resolution 2D simulations, where downdrafts behave very simply: they can grow from small fluctuations and merge with others but never split. A downdraft merging with its mirror image on the other side of a reflective boundary results in a downdraft trapped at the boundary, because moving it away from the boundary would require an extremely unlikely splitting.

So, the standard choice for horizontal boundaries is `side_bound=periodic`. At the beginning of each time step, values from the rightmost part of the computational box with dimension (m_1:n_1, m_2:n_2, m_3:n_3) are copied into the buffer layers on the left side, e.g., for each stencil i_2=$m_2 \ldots n_2$, i_3=$m_3 \ldots n_3$ and all quantities q=ρ, ei, v_1, v_2, v_3:

$$\begin{aligned} q\,(m_1-3) &= q\,(n_1-2) \\ q\,(m_1-2) &= q\,(n_1-1) \\ q\,(m_1-1) &= q\,(n_1)\,. \end{aligned} \tag{1}$$

and analogously for the right side. The same is done for the front and rear sides. Periodic boundary conditions have some nice properties: they are easy to implement, do not require a complex model for the values outside the computational box, allow standing and travelling waves, and do not produce artefacts at the boundaries, because the boundary layers are treated in the very same way as the inside regions. Still, they are artificial as they cannot be build in the real world.

For global models, usually `side_bound=transmitting` is used as described in Sect. 3.4 for the open top boundary.

3.2. Open bottom boundary

The bottom boundary of cool local models is located well inside the convection zone, where relatively high densities, small fluctuations in the thermodynamic quantities, and small Mach numbers prevail. The material coming from below is assumed to have the entropy of the adiabat of the deeper convective envelope (Ludwig et al. 1999). As induced artefacts or waves tend to grow with height due to the decreasing density, care must be taken to alter quantities only gently (i.e., not on too short time scales by a too large amount) and to dampen fluctuations, where necessary. Grimm-Strele et al. (2015) implemented several variants of the open lower boundary conditions (among them the version from Freytag et al. 2012) into the ANTARES code and made a detailed analysis, in particular of the problem of long thermal time scales in deep models.

The appropriate boundary condition `bottom_bound=inoutflow` in CO5BOLD adjusts the entropy `s_InFlow` of the ascending material with a rate controlled with `c_sChange`, ensures a zero total vertical mass flux, and, for stability reasons, reduces pressure fluctuations (controlled with `c_pChange`) and vertical velocities (controlled with the new parameters `c_v3changelinbottom` and `c_v3changesqrbottom`). Horizontal velocities are assumed to be constant with depth. The values of ρ, ei, and the vertical velocity v_3 *inside* the lowermost grid layer are actually modified during the application of this boundary condition, before the ghost cells are filled. Therefore, the conservation laws are only fulfilled in the volume above the bottom layer. For each cell $i_1{=}m_1\ldots n_1$, $i_2{=}m_2\ldots n_2$, in the bottom layer, $i_3{=}m_3$, the following steps are performed:

The equation of state is solved,

$$\mathrm{EOS}(\rho, e\mathrm{i}) \quad\rightarrow\quad s,\ P,\ T,\ \Gamma_1,\ \Gamma_3,\ c_\mathrm{s}\ , \tag{2}$$

to get entropy, pressure, temperature, first and third adiabatic coefficients, and sound speed. Horizontal averages of the density and pressure $\langle\rho\rangle^{(0)}$, $\langle P\rangle$ over the entire bottom layer are computed, where the superscripts $^{(0)},\ldots,^{(3)}$ here and in the following equations denote the sub steps. To avoid loosing or gaining mass due to round-off errors, these averaging steps are performed in *double precision*. A characteristic time scale is estimated by

$$t_\mathrm{char} = \Delta x_3 / \langle c_\mathrm{s} + |v_3|\rangle\ . \tag{3}$$

In cells with an upflow (i.e., $v_3 > 0$), mass and energy are modified according to

$$\rho^{(1)} \;=\; \rho + C_{s\mathrm{Change}} \frac{\Delta t}{t_\mathrm{char}} \frac{-\rho^2\ T\ (\Gamma_3 - 1)}{P\ \Gamma_1}\ (s_\mathrm{InFlow} - s)\,, \tag{4}$$

$$e\mathrm{i}^{(1)} \;=\; e\mathrm{i} + C_{s\mathrm{Change}} \frac{\Delta t}{t_\mathrm{char}} T \left(1 - \frac{\Gamma_3 - 1}{\Gamma_1}\right) (s_\mathrm{InFlow} - s) \tag{5}$$

with the two external parameters $C_{s\mathrm{Change}}$ ($\sim$0.1) and s_InFlow. The latter has to be chosen individually for each model to get the desired effective temperature T_eff. To reduce deviations of the pressure from the horizontal mean, the following corrections are applied to all cells in the bottom layer:

$$\rho^{(2)} = \rho^{(1)} + C_{P\mathrm{Change}} \frac{\Delta t}{t_\mathrm{char}} \frac{1}{c_\mathrm{s}^2}\ (\langle P\rangle - P)\quad , \tag{6}$$

$$e\mathrm{i}^{(2)} = e\mathrm{i}^{(1)} + C_{P\mathrm{Change}} \frac{\Delta t}{t_\mathrm{char}} \frac{1}{\Gamma_1\ \rho}\ (\langle P\rangle - P)\quad , \tag{7}$$

adding another parameter $C_{P\mathrm{Change}}$ ($\sim$0.3). To keep the total mass in the bottom layer (and therefore the entire model volume) unaltered, the density in the bottom layer is corrected with

$$\rho^{(3)} = \rho^{(2)} + \langle\rho\rangle^{(0)} - \langle\rho^{(2)}\rangle\ . \tag{8}$$

In addition to the procedure described in Freytag et al. (2012), vertical velocities can be smoothed to suppress local instabilities, that might arise with high-order reconstruction schemes (Freytag 2013), with

$$v_3^{(1)} \;=\; v_3 - v_3 \min(1, \frac{\Delta t}{t_\mathrm{char}} (C_\mathrm{v3ChangeLinBottom} + C_\mathrm{v3ChangeSqrBottom} \frac{|v_3|}{\langle v_3^2\rangle^{1/2} + \epsilon}))\ , \tag{9}$$

with ϵ a tiny real number, and the new control parameters $C_{v_3\mathrm{ChangeLinBottom}}$ ($\sim$0.0025) and $C_{v_3\mathrm{ChangeSqrBottom}}$ ($\sim$0.002). Finally, the vertical velocity is modified to ensure a zero-average vertical mass flux,

$$v_3^{(2)} = v_3^{(1)} - \frac{\langle\rho^{(3)}\, v_3^{(1)}\rangle}{\langle\rho\rangle^{(0)}}\ . \tag{10}$$

Because of this step, this boundary condition acts as a closed boundary for plane-parallel waves. Finally, the new values are given by

$$\rho^{(\mathrm{new})} = \rho^{(3)},\ e\mathrm{i}^{(\mathrm{new})} = e\mathrm{i}^{(2)},\ v_3^{(\mathrm{new})} = v_3^{(2)}\ . \quad (11)$$

Later, during the hydrodynamics step, the ghost cells are simply filled with constantly extrapolated values from the bottom layer while keeping the gravitational potential constant in these layers for `bottom_bound=inoutflow`. An exponential density stratification, a linear gradient of ei, and non-zero gravity are assumed in the extrapolation for `bottom_bound=inoutflow2`.

3.3. Closed bottom boundary

Classically, a closed bottom is employed for hot local models with single or double shallow convection zone(s) found in A-type stars (Freytag & Steffen 2004) or hotter white dwarfs (Tremblay et al. 2013), where the energy flux is injected as radiation handled by the radiation-transport solver.

However, to avoid having to deal with the in- and outflow of dust species, to prevent any possible excitation of spurious waves by the fairly complex open lower boundary (see Sect. 3.2), and to have better control of damping of residual pressure waves in nearly incompressible small-Mach-number flows, a variant of the closed bottom boundary is used also for models of brown dwarfs, that have a very deep convection zone (Freytag et al. 2010, 2017).

Simple reflective boundaries cannot be employed due to the gravitational stratification and the need to inject energy to provide the stellar flux and drive convection.

For `bottom_bound=closedbottom`, the gravity terms in the ghost layers are set to zero and all quantities are assumed to be constant, except for the vertical velocity (perpendicular to the boundary), that is inverted. For `bottom_bound=closedbottom2`, only the horizontal velocities are assumed to be constant, while for the density an exponential increase with a scale height computed from the two bottom-most layers in the computational box and for the internal energy a linear increase are taken into account. In this case, the vertical mass flux is reversed at the boundary instead of the vertical velocity. Within the (M)HD solver, the mass flux across the bottom boundary and all fluxes proportional to it are explicitly set to zero to enforce a vanishing mass transfer across the boundary.

As the radiative flux in the convection zone of cool stars with efficient convection is negligible, the energy in the bottom layers is not injected by radiation, but is added as energy source term within a layer with thickness `r0_grav`, analogously to the central "internal" boundary used for global models (see Sect. 3.6). For `heat_mode=bottom_entropy`, the entropy is adjusted towards `s_InFlow` with a rate `C_sChange` in a way very similar to the one described for the open lower boundary in Sect. 3.2. With `heat_mode=bottom_energy1`, a certain amount of energy per time is injected to give the mean vertical energy flux according to `Teff` but this mode is not often used (and tested), because this variant can lead to a significant change of the entropy in the bottom layers – over very long (thermal) time scales.

To avoid the generation of spurious acoustic waves when downdrafts bounce against a hard lower boundary, a drag force can be applied in the bottom layers (with a thickness specified by `r0_grav`), controlled with `C_CoreDrag` ~ 0.05). Different profiles of the strength of the damping with height are used for `heat_mode=bottom_entropy`, `bottom_entropy2`, or `bottom_entropy3`.

In spite being located right within the convection zone, the closed lower boundary works very well, at least for the tested cool brown-dwarf models (Freytag et al. 2010, 2017). However, to avoid the onset of small-scale convective cells at the top of the heated bottom layers, a sufficiently large damping zone (`r0_grav`), a sufficiently small entropy-adjustment rate (`C_sChange`), and an appropriate velocity-damping rate have to be chosen (`C_CoreDrag`).

Sample values for M dwarfs and brown dwarfs, with $T_{\mathrm{eff}} = 4000 - 1400\,\mathrm{K}$ at $\log g = 5$ with a bottom pressure scale height of $200 - 25\,\mathrm{km}$ are $r_{0\mathrm{grav}} = 100 - 17\,\mathrm{km}$,

$C_{s\mathrm{Change}} = 0.003 - 0.002$, $C_{\mathrm{CoreDrag}} = 0.09 - 0.035$.

3.4. Open top/outer boundary

The top `transmitting` boundary of local models is also used for all six sides of the computational domain in global models. This boundary is usually hit by an outgoing shock wave or lets material fall back into the computational domain – often with supersonic velocities (Wedemeyer et al. 2004). Thus, there is no compelling reason to tune the formulation for an optimum transmission of small-amplitude waves. Instead, a simple prescription that lets the shocks pass turns out to be most adequate and sufficiently stable. It is implemented by assigning typically one to three layers of ghost cells (the number depending on the order of the reconstruction scheme) with boundary values. For the velocity components and the internal energy a constant extrapolation is used, while the density is assumed to decrease exponentially with height in the ghost layers, with a scale height set to a certain fraction of the local hydrostatic pressure scale height. The control parameter `C_HpTopFactor` accounts for the – a-priori unknown – fraction of the turbulent pressure on the overall pressure. Effectively, it allows for the adjustment of the mean mass flux through the open boundary (Allende Prieto et al. 2013).

For local models, the temperature of the material entering at the top of the computational box is adjusted towards `C_Tsurf*Teff` with a rate controlled with the dimensionless parameter `C_TChange` (~0.5). Sometimes used for local MHD models is the option to smooth horizontal density fluctuations, activated by setting `C_rhoChange` to any positive value. Likewise mostly for (local or global) MHD models is the option to apply a drag force to infalling material, controlled by `C_VisBound` (~0.01).

3.5. Closed top boundary

While the open upper or outer boundary condition as described in Sect. 3.4 is used as default for most simulations, it is not without drawbacks: it contains assumptions about the matter conditions outside the computational volume (e.g., the density stratification or the temperature, which need some parameterization). That is particularly cumbersome for dust, where a simple extrapolation of the composition inside the computational box is often not adequate for the layers above. Therefore, in Freytag et al. (2010), a closed upper boundary has been used for 2D models of brown dwarfs, that since then has been refined (Freytag et al. 2017).

The implementation of `top_bound=closed` and `closed2` is analogous to the corresponding closed bottom boundary (Sect. 3.3). First ghost cells are filled and later the mass flux is explicitly set to zero. Differently to the closed bottom boundary, there is no energy injected.

However, shocks can create strong gradients across the boundary. These can be damped by an optional drag force applied in a "sponge" layer below the top boundary and computed in an optional `tweak module`. The total thickness of the damping layer (~ 18 grid cells) and of the transition region (~ 10 grid cells), the damping time scale for velocities (a few seconds), the velocity components to damp (usually vertically only), and the handling of the dissipated kinetic energy (conservative or – more often – lossy) are controlled by parameters (example values are given for 3D brown-dwarf models with $\log g$=5).

3.6. Central "internal" boundary

In addition, to the "proper" boundary conditions described above, that handle the actual borders of the computational domain, there is the option to create an "inner" boundary condition handling the energy-source terms (heating) and an optional drag force in the core of global models. The (magneto-) hydrodynamics and radiation-transport solvers integrate right through this zone

The radius of the central smoothing region of the gravitational potential (see Eq. (41) in Freytag et al. 2012) is set by the parameter `r0_grav`, which is also used as the radius of the sphere within which the energy-source terms are applied, unless it is overwritten with r0_core. This boundary condition has evolved with time and has a number of options, by now. Some standard choices to control heating and drag force will be described below.

For stars with efficient convection and little contribution from radiation transport in the interior, heat_mode=core_entropy1 is the appropriate choice to adjust the entropy towards the values specified with s_InFlow. However, if radiation carries a significant fraction of the stellar luminosity close to the core causing significant deviations from a flat entropy profile, heat_mode=core_energy1 is to be preferred to add a certain luminosity specified with luminositypervolume. A combination is selectable with heat_mode=core_energyentropy1, which provides a certain luminosity and additionally smoothes entropy fluctuations in the core.

For example, for heat_mode=core_entropy1, the following steps are performed for each cell $i_1{=}m_1 \ldots n_1$, $i_2{=}m_2 \ldots n_2$, $i_2{=}m_3 \ldots n_3$ within a sphere with radius $r_{0\text{grav}}$: The equation of state is solved,

$$\mathrm{EOS}(\rho, e\mathrm{i}) \quad \rightarrow \quad s, \;\; T, \;\; c_\mathrm{s} \;\; , \tag{12}$$

to get entropy, temperature, and sound speed. A characteristic time scale for each grid point is estimated from the grid spacing and the local sound speed as

$$t_\mathrm{char} = \Delta x / c_\mathrm{s} \;\; . \tag{13}$$

Usually, the local target entropy $s_\mathrm{InFlow,eff}$ is directly taken from an input parameter,

$$s_\mathrm{InFlow,eff} = s_\mathrm{InFlow} \;\; . \tag{14}$$

However, with $s_\mathrm{InFlRadGrad}$ a radial gradient (e.g., to let the heating start less abruptly at the surface of the core) and with $s_\mathrm{InFlLatGrad}$ a gradient in latitude (to enforce or dampen bipolar flows in rotating models) can be specified resulting in

$$\begin{aligned} s_\mathrm{InFlow,eff} = \; & s_\mathrm{InFlow} \\ & + (\frac{r}{r_\mathrm{core}} - 1)\, s_\mathrm{InFlRadGrad} \\ & + \frac{r}{r_\mathrm{core}} (\frac{1}{2} - \frac{x_\mathrm{c3}^2}{r_\mathrm{core}^2 + \epsilon})\, s_\mathrm{InFlLatGrad} \;\; , \end{aligned} \tag{15}$$

using the distance from the center r and the vertical coordinate x_c3 of each grid point. Then, the change of the internal energy that is needed to get closer to the specified core entropy is computed and the internal energy is updated according to

$$e\mathrm{i}^\mathrm{(new)} = e\mathrm{i} + C_{s\mathrm{Change}} \frac{\Delta t}{t_\mathrm{char}} T \left(s_\mathrm{InFlow,eff} - s \right) \;. \tag{16}$$

For the alternative algorithm heat_mode=core_energy1, a certain input luminosity can be specified with $C_\mathrm{LuminosityPerVolume}$. For hdCoreHeatProfile= Constantdrhoei, the energy in each grid cell within the core is updated by

$$e\mathrm{i}^\mathrm{(new)} = e\mathrm{i} + \frac{\Delta t \, C_\mathrm{LuminosityPerVolume}}{\rho} \;\; , \tag{17}$$

using the local density ρ. For hdCoreHeatProfile=Constantdei, the density $\langle\rho\rangle$ averaged over the core is used for the update, instead,

$$e\mathrm{i}^\mathrm{(new)} = e\mathrm{i} + \frac{\Delta t \, C_\mathrm{LuminosityPerVolume}}{\langle\rho\rangle} \;\; . \tag{18}$$

The damping of velocities in the core region is controlled by the dimensionless parameter C_CoreDrag, that determines the damping rate, and the string hdCoreDragProfile. The latter determines the radial profile of the damping (constant, linear, cosine, cosine squared) and the velocity components the damping should be applied to (all, radial, meridional).

For the example hdCoreDragProfile= Linear-Meridional, the following steps are performed for all grid points within the core: An inverse characteristic time scale is computed – from the sound-crossing time through the core and a radial profile – by

$$\frac{1}{t_\mathrm{char}} = C_\mathrm{CoreDrag} \frac{c_\mathrm{s}}{2 r_\mathrm{core}} \max(1 - \frac{r}{r_\mathrm{core}}, 1) \;, \tag{19}$$

which gives an overall drag factor

$$f_{v,\mathrm{Drag}} = \min(\frac{\Delta t}{t_\mathrm{char}}, 1) \;\; . \tag{20}$$

The azimuthal velocity is a projection given by

$$v_\mathrm{Azi} = \frac{x_\mathrm{c1} v_1 + x_\mathrm{c2} v_2}{r + \epsilon} \;\; , \tag{21}$$

which is used to compute the change for all velocity components as

$$\Delta v_1 = f_{v,\mathrm{Drag}}\, v_{\mathrm{Azi}}\, \frac{x_{c1}}{r+\epsilon} \tag{22}$$
$$\Delta v_2 = f_{v,\mathrm{Drag}}\, v_{\mathrm{Azi}}\, \frac{x_{c2}}{r+\epsilon} \tag{23}$$
$$\Delta v_3 = f_{v,\mathrm{Drag}}\, v_3 \ . \tag{24}$$

These are then used to update velocities and internal energy,

$$\begin{aligned} ei^{(new)} &= ei + (\Delta v_1\,(v_1 - 1/2\,\Delta v_1)) \\ &\quad + (\Delta v_2\,(v_2 - 1/2\,\Delta v_2)) \\ &\quad + (\Delta v_3\,(v_3 - 1/2\,\Delta v_3)) \end{aligned} \tag{25}$$
$$v_1^{(new)} = v_1 + \Delta v_1 \tag{26}$$
$$v_2^{(new)} = v_2 + \Delta v_2 \tag{27}$$
$$v_3^{(new)} = v_3 + \Delta v_3 \ . \tag{28}$$

4. MHD (HLLE) solver

In general, the treatment of boundary conditions of the MHD solver for ρ, ei, and $\boldsymbol{v}$ is similar to that of the HD solver, as the very same subroutines are used, with an additional branch to treat the magnetic-field components, if necessary. However, the open lower (`bottom_bound=inoutflow`) and the open upper (`top_bound=transmitting`) boundary are handled in the MHD solver by adapted copies of the corresponding routines of the HD solver. Thus, not all features of one version are automatically ported to the other.

The boundary conditions for the magnetic field are controlled with a separate set of parameters, i.e., for a given “hydrodynamics” boundary-condition specification (e.g., `bottom_bound=inoutflow`), there exist several sensible options for the magnetic-field components (e.g., `bottom_bound_mag=vertical`, `constant`, `inoutflow`, etc.). In the case of `bottom_bound_mag=inoutflow`, properties of the magnetic field entering the volume have to be specified with `B1_InFlow`, `C_magthetaB`, and `C_magphiB`.

The radiation-transport boundary-condition parameters are independent of the choice of the (M)HD solver.

5. Tensor viscosity

The – optional – tensor viscosity handles all types of boundary conditions that are implemented in the HD and the MHD solver. It extrapolates all necessary quantities (i.e., e.g., not magnetic fields and no dust densities) by filling one layer of ghost cells. There are no control parameters dedicated to the boundary behavior of the tensor-viscosity solver, alone.

6. Long-characteristics radiation transport for local models

The long-characteristics radiation-transport solver `MSrad3D` is constructed for box-in-a-star models and thus has some restrictions in the possible choice of boundary conditions (only `periodic` side boundaries, no `periodic` top/bottom boundaries, no `transmitting` bottom boundary). It responds to the general boundary settings described in Sect. 3. However, the open boundaries have dedicated parameters.

6.1. Bottom boundary

For `bound_bottom=inoutflow`, the radiative flux entering the model at the bottom is computed from the local temperature gradient with the diffusion approximation. In this case, the average energy flux leaving the computational box at the top can only be determined after a simulation is finished.

In contrast, for `bound_bottom=closedbottom` and `heat_mode=''`, a radiative flux according to the parameter `Teff` is injected at the bottom, which should result in almost the same value of the emitted flux in the long-term average.

For `bound_bottom=closedbottom` and an explicitly specified heating mode like `heat_mode='bottom_energy1'`, no radiative flux enters the computational domain at the bottom and a source term to the energy provides the energy within the bottom heating layers (see Sect. 3.3).

6.2. Top boundary

The top boundary allows radiation to leave in both cases `bound_top=transmitting` and `closed`.

However, for non-zero values of `C_radhtautop`, some incoming radiation is assumed. A positive `C_radhtautop` specifies the scale height of the optical depth (in cm, in all bands) just outside the computational box. With a negative value (e.g., C_radhtautop=-1.0), this scale length is estimated from the local pressure scale height. From this, the optical depth at the topmost point is computed. The source function in the outside layers is linearly interpolated between values corresponding to the temperature in the top cell and C_Tsurf*Teff at τ=0.

Irradiation along the vertical direction can be incorporated by specifying the effective temperature of a companion star with C_radTinci and a dimensionless distance factor $(R_\star/D)^2$ with C_radDinci (see Freytag et al. 2011).

7. Short-characteristics radiation transport for global models

In star-in-a-box models, energy is injected via source terms in the model core (see Sect. 3.6) and all outer boundaries have the task to let radiation escape, typically by setting top_bound=side_bound=bottom_bound=transmitting. Incoming radiation due to optically thin matter just outside the computational box can be taken into account by setting C_radhtautop to a typical scale length of the optical depth, in a simplified version of the boundary condition described in Sect. 6.2. However, C_Tsurf is not taken into account and instead for each ray, the outside temperature is assumed to be the same as the temperature in the boundary cell.

8. Conclusions

The specification of boundary conditions is a crucial step in the setup of a CO5BOLD simulation due to the physical nature of the problem, reflected in the mathematical description by partial differential equations, discrete versions of which are integrated by the numerical solver(s). While parameters controlling the flux of energy through the computational box are most important for all simulations of convective flows, the specifications describing the behavior of energy, gas and dust densities, velocities, and magnetic fields at or just beyond the boundaries influence the flow, dynamics, and stratification within the box. It is a peculiarity of simulations of flows in stellar atmospheres (for instance, with CO5BOLD), that cut a small volume out of a large dynamical system, that ideally an extension of the computational domain by shifting boundaries further out should have no or little impact onto the conditions within.

Recent refinements of the treatment of boundary conditions in CO5BOLD were done to reduce artefacts, partly caused by the previous increase in accuracy of the hydrodynamics solver (Freytag 2013). Reliably working versions of closed and open top, bottom, and "inner" boundaries even under conditions with strong velocity fields (waves, shocks, or downdrafts) are implemented and available in CO5BOLD – but have to be activated properly with parameters adequate for the type of star under consideration. While older choices of boundary conditions likely still work with a current version of CO5BOLD, they might be not optimal to achieve best results with the current numerics.

For local models, open challenges are the further investigation of the influence of complex magnetic-field configurations outside the computational box and how to account for them with simple boundary conditions. Even in pure-hydrodynamics models of atmospheres, the effects of a chromosphere (or a stratosphere) lying above the top of the model onto the conditions within the computational domain should be accounted for. The influence of the lower boundary onto atmospheric conditions, using current versions – with optimized parameter values – of open and closed variants, should be studied and compared. For global models, the effects of the formulation of the inner boundary conditions onto pulsations and rotation need further investigation, as well as the impact of the outer boundaries onto wind properties and mass-loss rate.

Acknowledgements. The computations were performed at the *Pôle Scientifique de Modélisation Numérique* (PSMN) at the *École Normale Supérieure* (ENS) in Lyon and on resources ("milou") provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Project p2013234.

Many thanks to Matthias Steffen and Oskar Steiner for commenting on the manuscript.

References

Allende Prieto, C., et al. 2013, A&A, 550, A103
Beeck, B., Collet, R., Steffen, M., et al. 2012, A&A, 539, A121
Chiavassa, A., et al. 2009, A&A, 506, 1351
Freytag, B. 2013, MSAIS, 24, 26
Freytag, B., Steffen, M., & Dorch, B. 2002, Astron. Nachr., 323, 213
Freytag, B. & Steffen, M. 2004, in The A-Star Puzzle, ed. J. Zverko, J. Žižňovský, S. J. Adelman, & W. W. Weiss (Cambridge Univ. Press, Cambridge), IAU Symp., 224, 139

Freytag, B. & Höfner, S. 2008, A&A, 483, 571
Freytag, B., et al. 2010, A&A, 513, A19
Freytag, B., et al. 2011, in Molecules in the Atmospheres of Extrasolar Planets, ed. J. Beaulieu, S. Dieters, and G. Tinetti (ASP, San Francisco), ASP Conf. Ser., 450, 125
Freytag, B., Steffen, M., Ludwig, H.-G., et al. 2012, J. Comp. Phys., 231, 919
Freytag, B., et al. 2015, CO5BOLD User Manual, http://www.astro.uu.se/~bf/co5bold_main.html
Freytag, B., Allard, F., Homeier, D. 2017, A&A, in preparation
Grimm-Strele, H., Kupka, F., Löw-Baselli, B., et al. 2015, New Astron., 34, 278
Harten, A., Lax, P. D., & van Leer, B. 1983, SIAM Review, 25, 35
Ludwig, H.-G., Freytag, B., & Steffen, M. 1999, A&A, 346, 111
Ludwig, H.-G., Caffau, E., Steffen, M., et al. 2009, MmSAI, 80, 711
Roe, P. 1986, Annu. Rev. Fluid Mech., 18, 337
Steffen, M. & Freytag, B. 2007, Astron. Nachr., 328, 1054
Tremblay, P.-E., et al. 2013, A&A, 557, A7
Wedemeyer, S., et al. 2004, A&A, 414, 1121