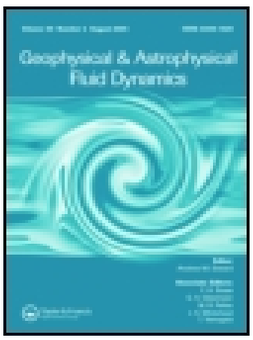

Taylor & Francis
Taylor & Francis Group

Geophysical & Astrophysical Fluid Dynamics

ISSN: 0309-1929 (Print) 1029-0419 (Online) Journal homepage: https://www.tandfonline.com/loi/ggaf20

Driving solar coronal MHD simulations on high-performance computers

Philippe-A. Bourdin

To cite this article: Philippe-A. Bourdin (2019): Driving solar coronal MHD simulations on high-performance computers, Geophysical & Astrophysical Fluid Dynamics, DOI: 10.1080/03091929.2019.1643849

To link to this article: https://doi.org/10.1080/03091929.2019.1643849

Published online: 29 Jul 2019.

GEOPHYSICAL & ASTROPHYSICAL FLUID DYNAMICS
https://doi.org/10.1080/03091929.2019.1643849

Driving solar coronal MHD simulations on high-performance computers

Philippe-A. Bourdin

Space Research Institute, Austrian Academy of Sciences, Graz, Austria

ABSTRACT

The quality of today's research is often tightly limited to the available computing power and scalability of codes to many processors. For example, tackling the problem of heating the solar corona requires a most realistic description of the plasma dynamics and the magnetic field. Numerically solving such a magneto-hydrodynamical (MHD) description of a small active region (AR) on the Sun requires millions of computation hours on current high-performance computing (HPC) hardware. The aim of this work is to describe methods for an efficient parallelisation of boundary conditions and data input/output (IO) strategies that allow for a better scaling towards thousands of processors (CPUs). The *Pencil Code* is tested before and after optimisation to compare the performance and scalability of a coronal MHD model above an AR. We present a novel boundary condition for non-vertical magnetic fields in the photosphere, where we approach the realistic pressure increase below the photosphere. With that, magnetic flux bundles become narrower with depth and the flux density increases accordingly. The scalability is improved by more than one order of magnitude through the HPC-friendly boundary conditions and IO strategies. This work describes also the necessary nudging methods to drive the MHD model with observed magnetic fields from the Sun's photosphere. In addition, we present the upper and lower atmospheric boundary conditions (photospheric and towards the outer corona), including swamp layers to diminish perturbations before they reach the boundaries. Altogether, these methods enable more realistic 3D MHD simulations than previous models regarding the coronal heating problem above an AR – simply because of the ability to use a large amount of CPUs efficiently in parallel.

ARTICLE HISTORY
Received 30 August 2018
Accepted 11 July 2019

KEYWORDS
Magneto-hydrodynamics; astrophysics; Sun; corona; high-performance computing

1. Introduction

Cutting-edge science is often limited only by computational resources. More realistic models come into reach with the continuously increasing computer power. One such fundamental step towards better understanding the famous coronal heating problem and plasma heating mechanisms was achieved with the work of Bourdin *et al.* (2013) and their following publications. This step only became feasible, because the *Pencil Code* got

CONTACT Philippe-A. Bourdin Philippe.Bourdin@oeaw.ac.at
This article has been republished with a minor change. This change does not impact the academic content of the article.

extended with some massive-parallel methods that enabled large-scale models to be run within the time of a typical PhD contract. Otherwise, scaling to less processors, the main simulation run described in Bourdin *et al.* (2013) would have taken about four years using 256 processors instead of about one year it took with 1024 processors.

The coronal heating mechanisms are unclear since many decades (Klimchuk 2006). For a better understanding of the coronal heating, novel models need to be as realistic as possible, so that the most relevant physical processes may be captured and analysed in a data post-processing step. It is currently not possible to use only observations for that task, because some key quantities like the coronal 3D structure and amplitude of the magnetic field still remain inaccessible. Still, long-standing theories need to be verified, like the entangling of magnetic field lines in the corona through shuffling of footpoints in the photosphere (Parker 1972). This could lead to so-called nanoflares that heat the corona through small and short-lived magnetic reconnection events after the field became entangled in the corona (Parker 1988). Magneto-hydrodynamic (MHD) turbulence and Alfvén wave turbulence compete to explain the coronal heat input (see the results of Rappazzo *et al.* 2007, 2008 versus results from van Ballegooijen *et al.* 2011, 2014). More recent models may test these scenarios against a realistic coronal magnetic field configuration (Bourdin *et al.* 2016).

We complement the approach of Rempel (2012) that includes parts of the convection zone below the photosphere to drive the magnetic field. While Rempel (2017) uses more advanced physics, our models are driven by actual photospheric observations and hence are able to match observed structures in the corona directly. This work is a continuation of *Pencil Code* models, like first presented in Bingert *et al.* (2010) that led to a better understanding on why the cross-section of coronal loops appears as roughly constant (Peter and Bingert 2012). Still, we do not know if such loops have a single- or multi-stranded structure in their magnetic fields (Peter *et al.* 2013). Chen *et al.* (2014) use a convection simulation with two emerging sunspots to drive a separate corona model. Peter *et al.* (2015) review what MHD models can tell us on coronal heating mechanisms. For different phenomena, like type-II spicules or a more realistic chromosphere, other groups include radiative transfer or ambipolar diffusion in their equation sets (Hansteen *et al.* 2010, Martínez-Sykora *et al.* 2011, Wedemeyer-Böhm *et al.* 2012).

We note that the Alfvén velocity is at least about one order of magnitude larger in the corona than in the chromosphere, which suggests that any changes in the chromospheric magnetic field would almost instantly and quasi-statically change the coronal field configuration (Bourdin *et al.* 2014, 2015). The field may either reconnect and release magnetic twist into the solar wind, the twisting magnetic perturbation may leave the corona before a substantial twist is build up on "open" field lines that connect to the heliosphere, or the perturbation may cross the whole corona on closed loops and eventually twist the chromospheric field on the other end of the loop, where the Alfvén velocity becomes lower again. To address this topic, we need to know the coronal field more precisely and we may indeed track coronal field lines from 3D MHD model data (Bourdin *et al.* 2018).

In order to obtain realistic coronal magnetic fields, we need to drive the MHD simulation with observations of real solar magnetograms, e.g. of a full active region (AR) that features some coronal EUV-bright loops that are known to be at least 1 MK hot. We cover an AR with some surrounding quiet Sun (about $240 \times 240\,\mathrm{Mm}^2$) with 1024×1024 grid points. The vertical grid resolution varies form about 100 km for the photosphere

and chromosphere to about 800 km in the upper corona. Furthermore, the photospheric horizontal shifting motions from granulation are required to ultimately test the field-line braiding mechanism. To this end, we implemented a photospheric granulation driver in the `solar_corona` module of the *Pencil Code* together with schemes for photospheric and chromospheric nudging that always gently push the model towards the observed state in the photosphere and that provide a lower solar atmosphere that adapts to pressure and temperature stratification changes in the corona. Our chromosphere acts here as a flexible lower boundary condition and as a reservoir for mass and internal energy, which is also the case in reality, because the lower atmosphere hosts significantly more mass than the corona.

With a novel boundary condition for the photospheric magnetic field we allow the granulation driver to push the foot points of field lines. This changes the horizontal component of the magnetic field already in the photosphere and in the grid cells above, which is required to test the braiding mechanism proposed by Parker (1972). Therefore, we need to extrapolate the magnetic field to the interior of the Sun, in order to provide the required ghost zones for the computation of numerical dervatives. Because the pressure increases below the photosphere, magnetic flux tubes shrink in their diameter with depth; see section 2.4 for details.

In the following we describe the photospheric and chromospheric nudging, the granulation driver, a new magnetic-field extrapolation, as well as coronal boundary conditions. Finally, we analyse the parallelisation of the Fourier transform and the data access (IO) routines.

2. Photospheric nudging

The photosphere is the lower boundary condition for our simulation domain. This requires us to set the temperature T, the density ρ, and the vertical components of the vector quantities that are perpendicular to this boundary, the vertical velocity u_z and the magnetic field B_z. In the photosphere, the convective cells from below practically reach the layer, where radiation efficiently cools the plasma. Hence, the advective heat transport by convection ends and the vertical transport of hot plasma breaks into granules, where the plasma motion has a horizontal component of some km/s at about 0.5–0.8 Mm around the granule centre (Ruiz Cobo *et al.* 1996). Once cooled, the now denser plasma gets submerged below the surface due to gravity and will eventually be heated again to complete the convection cycle.

In the same time, we know that plasma beta, the ratio between thermal and magnetic pressure, is near or above unity at the photosphere, in average (Bourdin 2017). As a result, the horizontal motions of breaking granules will advect the magnetic field with them. Many subsequently rising and decaying granules may push the field lines even on spatial scales larger than one single granule and in a random-walk fashion. Finally, magnetic flux bundles are rooted below the photosphere and constantly undergo a smooth global reconfiguration due to dynamo processes, which leads to horizontal motions of these patches on spatial scales much larger than one granule, but typically with a significantly lower speed. To complete the spectrum of photospheric driving motions, we will need to consistently combine both, small-scale granular motions and the large-scale magnetic reconfiguration in the photosphere.

2.1. Magnetic field

For the photospheric magnetic field nudging we use actual observations of the *Hinode* satellite, where the data from *SOT/NFI* (Solar Optical Telescope/Narrowband Filter Imager) give us the line-of-sight (LOS) component of the magnetic field in the photosphere (Kosugi *et al.* 2007, Tsuneta *et al.* 2008); see gray-scale background in figure 1. The resolution of our LOS magnetograms is about 120 km per pixel. Magnetograms do not resemble visible-light intensity observations (cf. Bourdin 2011) and hence we do neither directly see the granules in such magnetograms nor could we derive the horizontal plasma motions. Even intensity maps of the photosphere from the same telescope would not resolve a granule with more than a few pixels. Therefore, we cannot recover the horizontal granular advection from Hinode data and have to find another way to mimic realistic granulation; see section 2.3.

The LOS magnetograms are uncalibrated and we use the smaller field-of-view (FOV) of the *SOT/SP* (Spectro-Polarimetric) instrument (Lites *et al.* 2013) that provides calibrated vector magnetograms and co-align both, the LOS and *SP* data, while we reduce the resolution of the LOS to the *SP* data. A pixel by pixel comparison allows us to find the calibration factor for the full-FOV LOS magnetograms. For consitency, we also interpolate two LOS magnetograms in time to match the exact *SP* observation time. Because both instrument's detectors have a slight rotation against each other, we split the common FOV in about 8×8 subfields and coaling them separately. These steps help to reduce the broadness of the scatter in the distribution of the correlated data that we show in figure 2.

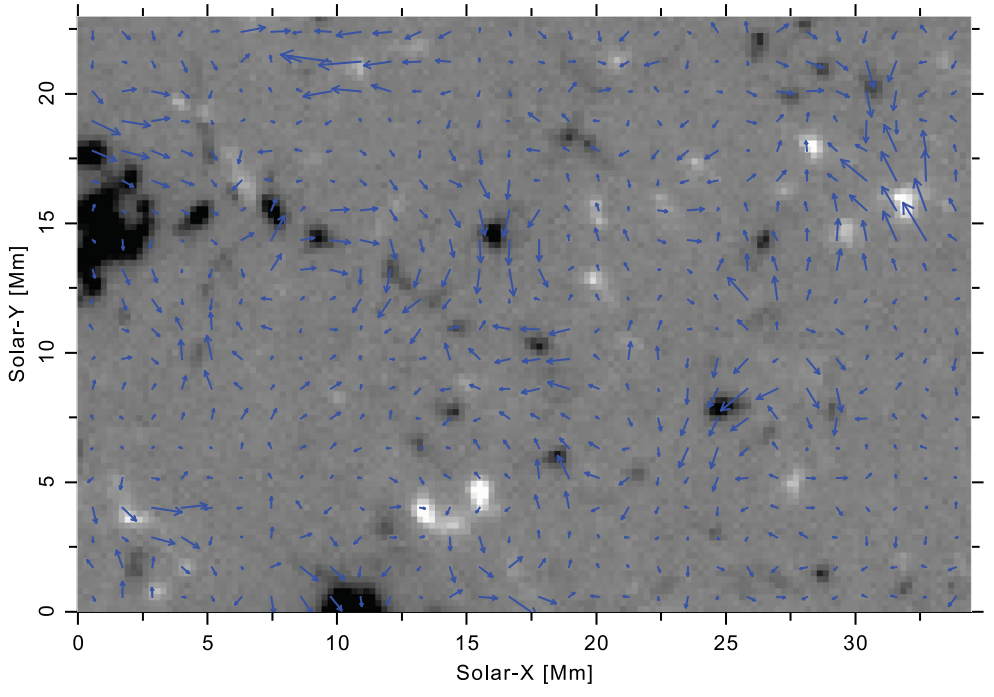

Figure 1. Hinode line-of-sight magnetogram (grayscale, saturated at ± 300 G) with overlaid velocity vectors (blue) obtained with a local correlation tracking. The observation was made near disc centre and is from 2007 November 14 (colour online).

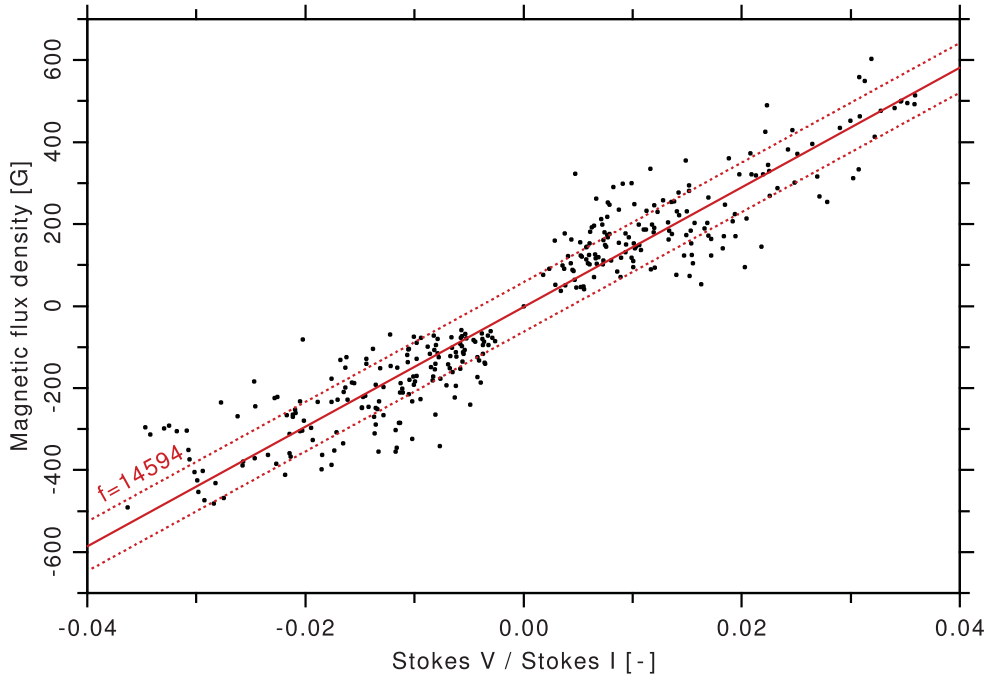

Figure 2. Distribution of co-aligned uncalibrated *SOT/NFI* and calibrated *SOT/SP* data. The red line represents a least-absolute-deviation (LAD) fit with its uncertainty interval marked by dotted lines. Figure taken from Bourdin (2014a); see figure 2.4 on page 31 (colour online).

The horizontal extent of our *NFI* FOV is about one seventh of the solar radius. We correct the LOS magnetograms for the surface curvature within the FOV of *SOT/NFI* by assuming that all strong flux concentrations are mainly vertical, so that we may scale the magnetic field amplitude of both polarities with $1/\cos\theta$, where θ is the angle between the LOS and the vertical direction. We now co-align the time series of magnetograms to the one in middle of the time series so that we correct for spacecraft jitter and the rotating Sun. Finally, we crop all images to their common FOV.

For each of the magnetograms we rescale both polarities separately and with a constant factor to obtain flux-balanced observations. This is required here to avoid an average magnetic monopole that would force all of its flux to leave through the upper boundary, because the simulation domain is periodic in the horizontal directions. Also this avoids unwanted effects from a time-varying total flux that would have to propagate immediately through the whole domain because of our boundary conditions and hence disturb the actual coronal magnetic field in an unnatural way. In any case, the total magnetic flux of the whole Sun is always zero. Natural localised flux imbalances on the real Sun cause magnetic connectivity to remote regions, which we cannot include in our model due to the limited simulation domain size. The connectivity that we miss through our flux balancing is minor and does not significantly change the field geometry in our model.

When we use a non-periodic magnetogram for a periodic simulation domain, we need to overlap the boundaries of the magnetogram with a smooth transition at the borders of the observable FOV. This makes the magnetograms periodic and we do not see artefacts in our model if the overlap region is about 24 pixels or about 6 Mm wide.

If we would now simply impose the obtained flux-balanced time series of LOS magnetograms on our lower boundary for all times, the magnetic field description would on one hand be perfectly consistent with the observation. On the other hand, we would effectively stop the granular driving to operate, because of two reasons: First, the granular motions are not directly recovered with LOS magnetograms. Second, our granulation driver can practically never advect a magnetic field line, because its original state would get restored in the next iteration of the simulation. Therefore, we have to allow the granulation to advect the field lines and in the same time we have to carefully restore the observed magnetic state. Luckily, small-scale granules live much shorter than the large-scale magnetic field needs to change substantially. This allows us to apply two nudging processes on different time scales, where we push the model into the targeted states by an exponential decay with distinct characteristic half times for each process.

2.2. Local correlation tracking

Once we want to allow for time-varying magnetograms at the lower boundary, we need to apply also horizontal plasma motions that are consistent with the displacement of the observed magnetic patches. When plasma beta is around unity, the magnetic field can be shifted from plasma motions or the plasma can be dragged together with moving field lines. Both ways, the magnetic field displacement and the plasma motion should be consistent. Therefore, we deduce the horizontal motions of magnetic patches with a local-correlation-tracking (LCT) method and obtain another lower-amplitude velocity field on spatial scales larger than granules. This LCT velocity field we also need to apply in the photosphere.

Even though we omit here electric fields that are in principle necessary to fulfill the induction equation, the good match of the simulated AR corona with the observed one (Bourdin *et al.* 2013) supports this simplification. Still, the question remains how much the coronal heat input may change if we include those omitted photospheric electric fields.

In the first step to obtain a velocity field out of a magnetogram time series, we rebin the data to the optical resolution, which means we bin a square of 2×2 pixels into one simulation pixel of about 235 km side length. Then we compute the local cross-correlation coefficients of two consecutive frames of the time series, where we shift the frames to each other by one pixel in each direction. Together with the original zero-shift correlation coefficient this gives us five coefficients, where we find the sub-pixel shift vector as the local maximum of a five-point Gaussian fit along two spatial dimensions. We apply a Gaussian-convolution filter to the obtained map of shift vectors to remove small-scale fluctuations that we are not interested in. Finally, we need to interpolate this vector field at the actual pixel positions of our simulation grid.

The resulting velocity field from our LCT method is displayed as blue arrows in figure 1. As we see, some magnetic patches are surrounded by a local velocity field pointing in one direction, where the strongest velocity is near the centre of that patch; see black patch at $x = 25$ Mm and $y = 8$ Mm in figure 1. The correlation decays with distance from the flux concentrations, which is expected because one cannot deduce a horizontal velocity from two consecutive insignificant flux regions (gray background).

2.3. Granulation driver

To mimic the horizontal advecting motions of breaking granules in the photosphere we use a constructed velocity field that resembles granules with a diameter of 1.6 Mm. We modify here a method described by Gudiksen and Nordlund (2005) that is based on a weighted *Voronoi tessellation* driver (Schrijver *et al.* 1997). In contrast to Gudiksen and Nordlund (2002) who use a random pattern for driving their simulation, we generate a velocity field that resembles solar granulation not only in a statistical sense. Around each granule we define an additional radius of 0.24 Mm as a zone where inter-granular lanes may form due to overlapping of the velocity fields from neighbouring granules. The typical lifetime of a granule is $T_{\mathrm{gran}} = 5$ min. The amplitude of the typical velocity in the interior of a granule we define as $v_0 = 1.028$ km/s. The radial velocity within each granule must of course be zero at the centre, increases with distance from the centre, until it decays outside of the granule's radius and within the inter-granular lane, which ranges from 0.8 to 1.04 Mm distance from the centre. An example of a mainly radial velocity field can be seen near $x = 6$ and $y = 8$ Mm in figure 3.

We randomly fill the simulation domain with granules, until there is no more available space for new granules. No new granule is allowed to emerge within the inner radius of 0.64 Mm around any other granule's centre. A granule ceases to exist, when its velocity

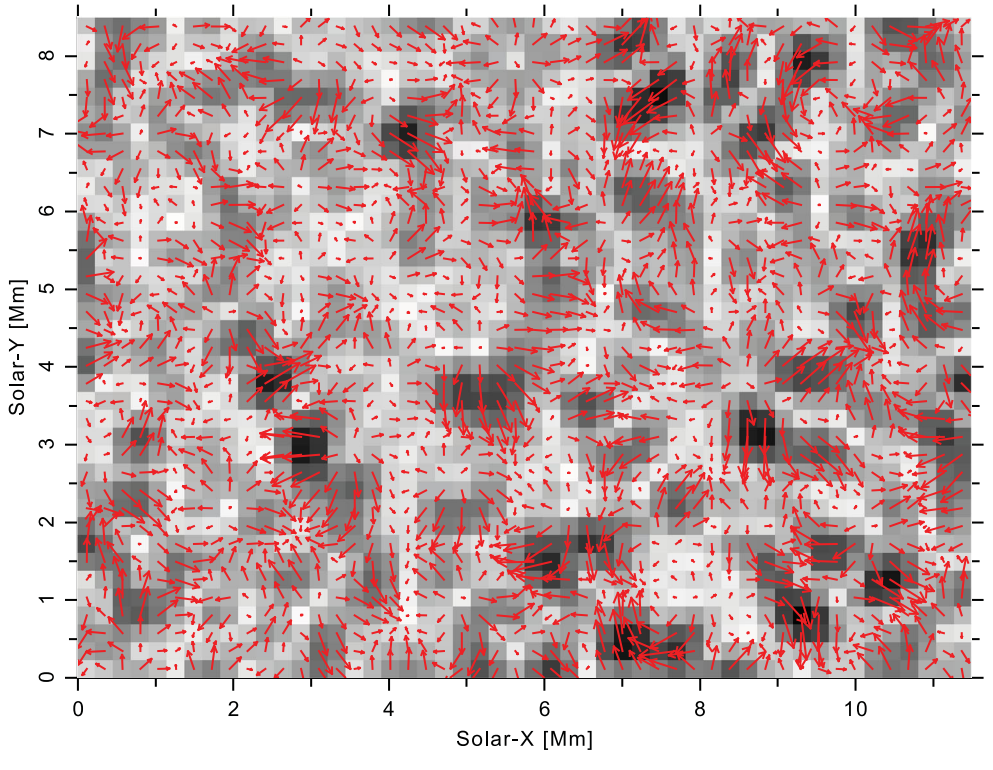

Figure 3. Granulation driver velocity field (grayscale, saturated black is 8 km/s) with overplotted velocity vectors (red). One grayscale square represents one simulation pixel with a side length of 230 km (colour online).

amplitude $v(t)$ drops below the threshold of $v_{\min} = 0.78\, v_0$; see equation (2). The former granule's area then becomes immediately available for new granules.

To avoid that all granules emerge and decay always at the same time, we vary the average granular lifetime with normally distributed random values within ±10% and define this individual lifetime as $T = T_{\text{gran}} \pm 10\%$. We also alter the typical granular velocity amplitude in the same way for each granule as $v_{\max} = v_0 \pm 15\%$. Accordingly, we change the time when the maximum velocity amplitude shall be reached ($t_{\max}$) to be earlier for shorter lifetimes and later for longer lifetimes T as

$$t_{\max} = T\sqrt{|\log(v_{\min}/v_{\max})|}. \tag{1}$$

For a smooth emergence and decay of a granule, we multiply its velocity amplitude with a life-cycle function that depends on the simulation time t and the time when the granule was created t_0:

$$v(t) = v_{\max} \cdot \exp\left[-\left(\frac{t - t_0 - t_{\max}}{T}\right)^2\right]. \tag{2}$$

Now that we have filled our FOV with granules, in particular with overlapping areas that have non-radial velocities, we may amplify the purely stochastic vorticity in our two-dimensional horizontal-velocity field $\boldsymbol{v}_{\text{gran}}(\boldsymbol{r}, t)$ at the position vector $\boldsymbol{r}$. For that, we Fourier-transform the velocity field, extract its rotational part $\hat{\boldsymbol{v}}_{\text{rot}}$, and filter out large wavenumbers with an exponential weighting factor

$$\hat{\boldsymbol{v}}'_{\text{rot}}(\boldsymbol{k}, t) = \hat{\boldsymbol{v}}_{\text{rot}}(\boldsymbol{k}, t) \cdot \exp\left[-(2|\boldsymbol{k}|/k_N)^4\right], \tag{3}$$

where $\boldsymbol{k}$ is the in-plane wave vector and k_N is the Nyquist frequency. We transform $\hat{\boldsymbol{v}}'_{\text{rot}}(\boldsymbol{k}, t)$ back to real space $\boldsymbol{v}'_{\text{rot}}(\boldsymbol{r}, t)$, amplify it with a factor of f_{rot}, and add it back to the initial granular field. We choose $f_{\text{rot}} = 5$ so that the resulting velocity field will have flows along the inter-granular lanes that reach realistic horizontal speeds of about 8 km/s, which is similar to the observed inter-granular velocities; see saturated black colour in figure 3.

Finally, we correct the new velocity field with enhanced vorticity $\boldsymbol{v}_{\text{vor}}$ so that its root-mean-squared velocity becomes identical to one of the initial granulation field:

$$\begin{aligned} \boldsymbol{v}_{\text{vor}} &= \boldsymbol{v}_{\text{gran}} + f_{\text{rot}}\, \boldsymbol{v}'_{\text{rot}}, \\ \boldsymbol{v}^*_{\text{gran}} &= \boldsymbol{v}_{\text{vor}} \cdot \sqrt{\langle \boldsymbol{v}_{\text{gran}}{}^2 \rangle / \langle \boldsymbol{v}_{\text{vor}}{}^2 \rangle}. \end{aligned} \tag{4}$$

We obtain a velocity field with radial outflows that turn into tangential velocities. The tangential flows of neighbouring granules may interfere and form inter-granular lanes with larger velocities than the average speeds inside the granules; see around $x = 3$ and $y = 4$ Mm in figure 3 for an example.

For consistency with external velocity fields, like the LCT velocities derived from a magnetogram time series, all granule centres will be propagated in accordance with the pre-existing velocity vector at each granule's centre. If no other velocity field has been activated

during the generation of the granular driving motions, the granule centres remain where they emerge.

A common problem in massive-parallel simulations are boundary conditions that involve only few processors (see section 5.1 for another example) because the boundary processors then cause inactive delays on the non-boundary processors. The granulation driver code in the *Pencil Code* is not scalable. Therefore, we improve the scalability of our coronal model with a two-step scheme: First, we create the granulation in a 2D simulation of the photosphere and write the generated velocity field to an external file. This step requires only few processors and may run with a substantially larger timestep than the full coronal model because we need only one granulation snapshot about every 10 s and some timestep-critical methods (like the Spitzer heat conduction) are not relevant here. Second, in the 3D coronal model we read the previously generated velocity field, distribute this data to each of the photospheric boundary processors, and then interpolate between the granulation snapshots in time. Because the timestep in the 3D model is substantially lower than 10 s, we save many iterations of the granulation driver code, because the changes therein are minimal. In addition, the time-interpolation between two granulation snapshots can be scaled to all involved boundary processors, because each processor already has all necessary data. Finally, a time-interpolation consists of much less computations than one granulation driver iteration. This scheme reduces the waiting time of non-boundary processors to a minimum and hence makes the coronal model better scalable.[1]

The photospheric velocity driver is of course not switched on immediately. If we would do that, we would cause an instantaneous force acting on an equilibirum state. This is equivalent to a very strong impulse that creates a shock front (see figure 3 of Bourdin 2014b). To avoid this switch-on effect, we smoothly ramp up the targeted velocity field linearly within the first minute of the simulation. In addition, the horizontal bulk velocity $\boldsymbol{u}_{\rm hor}|_j$ at the photospheric boundary and in the three ghost layers below ($j \in \{0, -1, -2, -3\}$) is never directly imposed. We smoothly push the velocity to its target value by an exponential decay that we implement by an additional term in the equation of motion (shortened by "$\cdots$"):

$$\partial \boldsymbol{u}_{\rm hor}/\partial t\big|_j = \cdots - \tau_u \left(\boldsymbol{u}_{\rm hor}\big|_j - \boldsymbol{v}^*_{\rm gran} \right) \tag{5}$$

with the inverse decay half-time $\tau_u = 0.5\,{\rm s}^{-1}$.

Figure 4 shows a direct comparison of LCT and granulation driver velocity histograms, where we see that the LCT velocities have a peak below 0.1 km/s and hence smaller amplitudes than the granulation driver that peaks around 2 km/s. We checked that most of the driving power (Poynting flux) stems from the granulation driver. The granular velocities vary on much smaller spatial scales than the LCT velocities from the horizontal movement of magnetic patches, as also becomes clear from a direct comparison of the axes in figures 1 and 3.

[1] The related parameters in the `solar_corona` module are the logical flag `lgranulation` to activate the granulation driver, `lwrite_driver` to write the generated velocity field to an external file, `tau_inv` as the inverse time scale for velodity-field nudging, `vorticity_factor` as the factor to increase the vorticity, and `dt_gran` as the update interval for the granules. The logical flags `luse_vel_field` and `luse_mag_vel_field` activate the reading of external velocity fields.

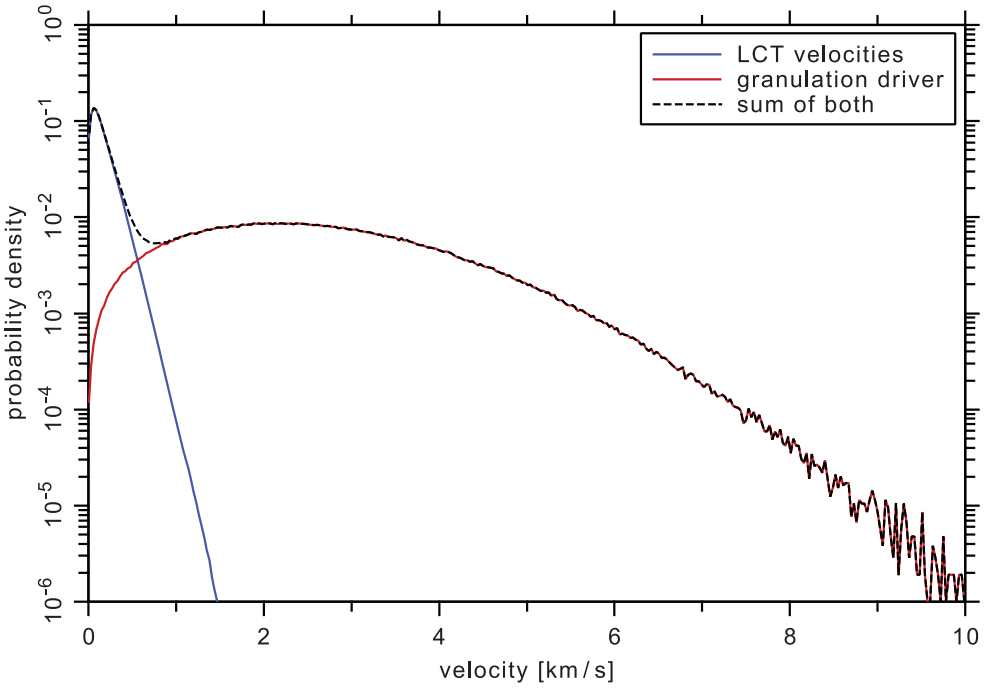

Figure 4. Histograms of LCT (blue) and granulation driver (red) velocities (colour online).

2.4. Magnetic-field extrapolation

We use the vector potential $\boldsymbol{A}$ for our computation and the magnetic field $\boldsymbol{B}$ is then only a derived quantity that is always divergence free and is unique for any given gauge $\boldsymbol{\Phi}_0$:

$$\boldsymbol{B} = \boldsymbol{\nabla} \times (\boldsymbol{A} + \boldsymbol{\Phi}_0). \tag{6}$$

To prescribe the vertical magnetic field B_z in the photosphere, while using the vector potential $\boldsymbol{A}$ within our simulation, we need to set the two components A_x and A_y because

$$B_z = \frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}. \tag{7}$$

A_z may then be set according to a self-chosen gauge, here the Weyl gauge with $\boldsymbol{\Phi}_0 = \boldsymbol{0}$.

In order to mimic the increase in atmospheric and magnetic pressure in the solar interior below the photosphere, we use an inverted potential-field extrapolation that concentrates any magnetic flux bundles with depth, which leads to a reduction of the diameter and an increase of the amplitude of those flux bundles. Typically, the pressure scale height below the photosphere is about 300 km. As we use three ghost layers below the lower boundary with a grid distance of about $\Delta z = 100$ km, we would have to double the contrasts from the surface magnetograms observed at $z(0) = 0$ Mm. This is not possible without introducing artifacts like wiggles and checker-board patterns in the ghost cells. Therefore, we reduce the contrast increase in the lower ghost cells by a constant factor, here to one fifth, as if the pressure scale height would be about 1.5 Mm (see also Bourdin *et al.* 2018, for a more mathematical description). Of course, this reduction introduces a slight error in the horizontal

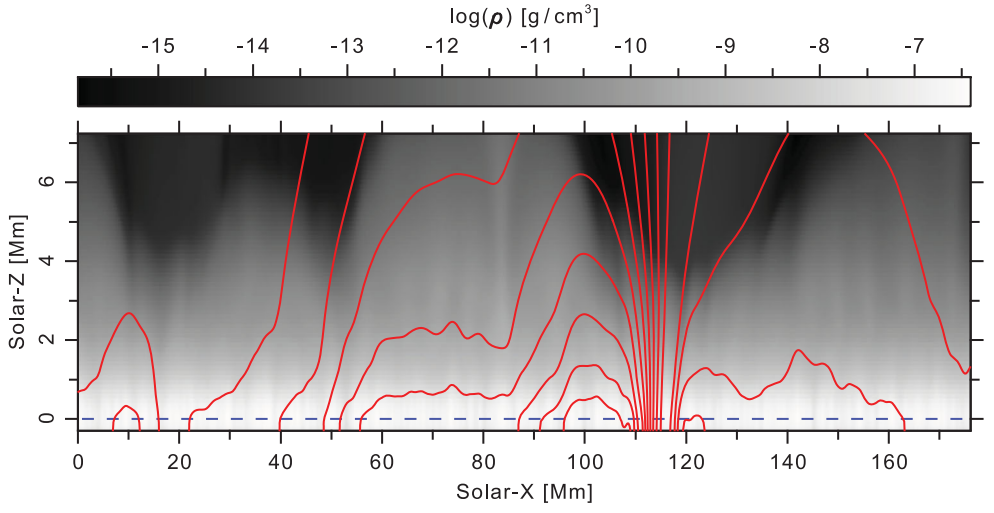

Figure 5. In-plane magnetic field lines (red) in a vertical cut through a strong flux concentration in the lower part of the model. The gray scale indicates the logarithmic density. The blue dashed line is the actual physical boundary of the simulation domain at $z = 0$ Mm (colour online).

field components below the surface. But this error seems acceptable, because significant magnetic structures in the photosphere typically have a horizontal diameter of well above 1 Mm and the field therein is anyway mainly vertical for strong flux concentrations that finally may reach our model corona; see figure 5.

In principle, one could say that this description of the lower magnetic field boundary is similar to a strictly vertical field boundary condition. We like to note this is not the case, because we still allow for any amount of horizontal field at the boundary and hence very small near-surface closed field lines may form; see red lines near $x = 10$ and 120 Mm in figure 5, where at the latter position a field line even gets horizontal below the actual boundary. Such cases are rare, which supports our argument that the error is small when enlarging the pressure scale height as described above, but this shows we do not enforce a vertical field in the photosphere and below.

Different and less complex boundary conditions for the magnetic field may of course exist. For example, we could enforce a vertical field at the lower boundary. But when we do that, many field lines in the physical domain that are not strictly vertical, would have a stronger kink near the boundary as with a more flexible boundary condition. An enforced vertical field at the bottom will generate larger derivatives and hence artificial currents near the photosphere for a multi-polar AR case. In comparison, the vertical-field boundary condition has about twice the current density, twice the maximum magnetic Reynolds number, and about 40 % larger Lorenz forces than we find with the more flexible scheme we use here at the bottom of the domain. This would then require us to double our magnetic diffusivity η in order to keep the magnetic Reynolds number under control. As a consequence, the larger diffusivity leads to more slippage of the field lines when we advect them with our observational driving — which means the driving looses much of its effect. Therefore, currents at the bottom boundary are unwanted and we like to keep the magnetic diffusivity minimal to get the model more realistic.

Previously, the *Pencil Code* had also a regular potential-field extrapolation that relaxes the fields to a force-free state outside the physical domain. This would mean to smear out constrasts in the magnetic field below the photosphere. For the top boundary this is acceptable, because the field is already quite force free, there. But below the photosphere, the pressure increases and magnetic structures are not force free. If we would apply the original potential-field extrapolation, we again generate strong currents because the field will relax to a force-free state. This is a permanent process, because we constantly keep on driving the field away from that state. Therefore, our method with increasing magnetic contrasts in the ghost cells below the bottom boundary is closer to reality than using the regular potential-field extrapolation.

At the top boundary, we may use the regular potential-field extrapolation with contrasts that get smeared out exponentially with height, which is fairly realistic in the outer corona.[2]

2.5. Atmospheric boundary condition

At the lower boundary and below (in the ghost layers) we prescribe the density according to a smoothly combined profile of the stellar interior and the solar atmosphere (as given in Bourdin 2014b). We use a globally constant diffusivity of $D_\rho = 8 \cdot 10^6\,\mathrm{m}^2/\mathrm{s}$ in the mass density ρ, which is needed for the numerical stability in all independent MHD variables. This results in a very small mass transport into the simulation box through the lower boundary because the diffusion acts along the gradient that points upwards. Implicitly, the thermal pressure increases in the upper layers. This pressure increase gets exactly compensated by a hydrodynamic flow equilibrium that requires a downwards bulk plasma motion. Consequently, the photospheric density at the physical boundary layer may vary slightly from the preset value. The closed z-boundary condition sets the vertical bulk velocity $u_z = 0$ at the lower physical boundary $z(0)$. Hence, a mass inflow would then accumulate. To compensate for continuous density changes due to the diffusion, we have to constantly overwrite the density to its initial profile at and below the photosphere. The layers above the photosphere undergo the regular hydrodynamic settlement of the atmospheric stratification due to the gas pressure, mass flows, and gravity.

We set the temperature boundary condition so that a consistent hydrostatic equilibrium is reached at the lower boundary. To achieve this, we recursively set the temperature $T|_j$ in the three lower ghost layers $j \in \{-1, -2, -3\}$ at the grid positions $z(j)$ as:[3]

$$T|_j = T|_{j+1} \frac{\rho|_{j+1}}{\rho|_j} \exp\left[- \Delta z|_j \frac{c_v}{c_p} \frac{g\big|_{j+1}}{T|_{j+1}} \right], \tag{8}$$

where we use the identity $c_v = \gamma/(\gamma - 1)$ for an ideal gas and $\Delta z|_j = z(j+1) - z(j) > 0$ for the bottom boundary. $\rho|_j$ is the density and $g\big|_j$ is the gravity constant at the grid positions $z(j)$.

An alternative, without prescribing the density to its initial value, is to maintain a zero first derivative of the temperature by a symmetric boundary condition and then adapt the

[2] In the `start.in` configuration file we set `bcz` to `'pfe'` for the first magnetic field component. At the same component's position we also set `fbcz_bot` only for the bottom boundary to `0.2` in order to limit the extrapolation below the photosphere to one fifth of the pressure scale height.

[3] In the `start.in` configuration file we set `bcz` to `'fg'` for the density and to `'hse'` for the temperature.

density according to a hydrostatic equilibrium. This has the advantages that, first, there is no heat flow into or out of the simulation domain, and second, that the density may deviate from its initial value and maintain a hydrostatic equilibrium at the boundary; see section 4.1 for details.

3. Chromospheric nudging

Radiative losses act as an energy sink in the solar atmosphere (Cook *et al.* 1989). The Spitzer heat conduction transports energy along the temperature gradient (Spitzer and Härm 1953) and therefore provides an energy source in the chromosphere. Both together may destabilise the chromosphere because once the heat conduction provides less energy, the radiative losses cool the plasma that hence becomes denser. Denser plasma is a stronger source of radiation that cools the plasma further. This may form a run-away effect, because there is no radiative source term in our model that would heat optically thicker plasma through absorption.

Another run-away effect exists for an excess heat input that leads to an adiabatic expansion of some plasma which then looses its ability to radiate an excess of internal energy, because the radiative losses decrease for lower densities. Therefore, we need to stabilise our model chromosphere with a so-called Newtonian cooling method, which resembles the stabilising effects of chromospheric radiative transfer at low computational costs.

3.1. Newtonian cooling

Temperature fluctuations in the lower solar atmosphere get smoothened out through the radiative transfer that efficiently heats denser and optical thicker plasma, while it also cools less dense plasma through reduced absorption and the continuous emission of photons (Spiegel 1957). Both stabilising effects of a proper radiative transfer treatment in the chromosphere can be achieved through a novel combined Newtonian cooling and chromospheric nudging approach.

First, we gently push back the chromospheric temperature to its initial stratification $T_0(z)$ through an exponential decay. Second, we implement a cooling term that slowly pushes the temperature to a target value $T'(\rho(\boldsymbol{r},t))$ at the position $\boldsymbol{r}=(x,y,z)$ that we take from the initial temperature stratification $T_0(z')$ at the height z' where the actual density $\rho(\boldsymbol{r},t)$ is equal to the initial density stratification $\rho_0(z')=\rho(\boldsymbol{r},t)$. We finally obtain the target temperature T^* for the nudging as

$$T^*(\boldsymbol{r},t)=\sqrt{T_0(z)T'(\rho(\boldsymbol{r},t))}. \tag{9}$$

Of course, we like to apply the Newtonian cooling only in the chromosphere, where our model needs it, and not beyond. This we achieve by a cutoff mechanism with four components: (1) we use a density-dependent cutoff that sets in when the density becomes eight orders of natural-logarithmic magnitude smaller than the photospheric value, (2) we smooth out this sharp cutoff boundary with a sine function over the last two orders in this natural-logarithmic magnitude, (3) we enforce another cutoff that sets in above a height of 3 Mm, and (4) we apply another smooth sine transition over the upper 0.3 Mm on this height-dependant cutoff.

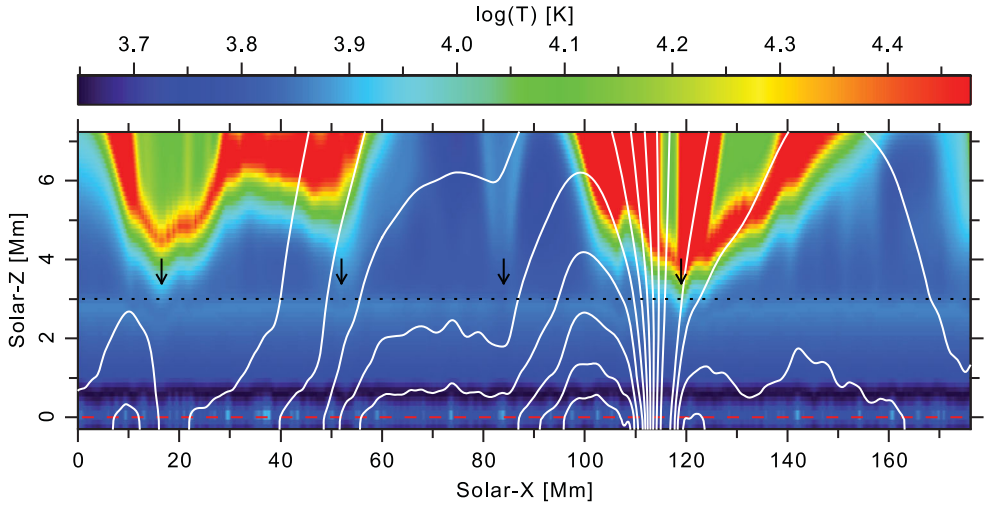

Figure 6. Logarithmic temperature (colour code) in the lower atmosphere for a vertical cut through a strong flux concentration together with the in-plane magnetic field lines (white) at the same location as in figure 5. The red-dashed line indicates the location of the photosphere.

We now add the nudging term to the energy balance (shortened by "··· "):[4]

$$\partial T/\partial t = \cdots + T \exp\left[-\tau_t \left(1 - \frac{T^*}{T}\right) c_\rho c_z\right] \tag{10}$$

with the inverse decay half-time $\tau_t = 0.5\,\mathrm{s}^{-1}$, c_ρ as the smooth density-dependent cutoff function, and c_z as the smooth height-dependent cutoff.

3.2. Compressible atmospheric column

The effect of the Newtonian cooling becomes visible in figure 6, where we see a relatively constant chromospheric temperature is maintained during the simulation run and remains similar to the initial condition below 3 Mm (black dotted). We indicate locations with down arrows, where we find that the atmospheric column gets compressed and hence the temperature follows a similarly compressed stratification. With time, this atmospheric column compression will relax towards the initial state due to our combined Newtonian nudging method; see section 3.1. This allows our model chromosphere to adapt to, e.g. downflows from the corona and eventually relax back to the initial stratification after these downflows end.

At some locations we find the atmospheric column rises quickly to coronal temperatures above 3 Mm; see leftmost and rightmost down arrows in figure 6. In other regions, the lower-coronal plasma remains relatively cool, typically above regions with more horizontal field; see at Solar-X from 60 to 95 Mm.

[4] The related parameters in the `solar_corona` module are the logical flag `lnc_density_depend` to activate the density-dependent Newtonian cooling, `nc_tau` as the inverse time scale necessary for the nudging by an exponential decay, `nc_lnrho_num_magn` as the number of natural-logarithmic magnitudes in density, `nc_lnrho_trans_width` defining the smooth transition of the density-dependent cutoff, `nc_z_max` as the maximum height to apply the Newtonian cooling, and `nc_z_trans_width` to define the smooth transition for the height-dependent cutoff.

In the photosphere, we see that temperatures may be higher and lower, where the density is lower and higher, respectively. This anti-correlating behaviour of temperature and density could be due to some compressional heating from the granular horizontal advecting motions; see the lower boundaries in figures 5 and 6 at $z = 0$ Mm (dashed lines). Also these local changes in temperature will eventually relax to the initial photospheric temperature due to the limited lifetime of granules and the Newtonian cooling.

We find also that the height of the photospheric temperature minimum varies between about 250 to 350 km due to our compressible atmospheric column at the lower boundary of our simulation domain; see wavy black layer in the lower part of figure 6.

4. Upper coronal boundaries

On the upper end of the simulation domain, we prefer not to allow heat or plasma inflows or outflows. The reason is that any inflow would be of undefined velocity, temperature and density. We would anyway not expect heat outflows, because the temperature gradient usually leads only to inflows of heat. At the same time, such inflows are unwanted, because we like to learn about the intrinsic heating of the corona, independent of any boundary condition.

If we allow for plasma outflows, this may lead to an unrealistic mass loss, because in reality this outflowing plasma may well fall back to the Sun, later, due to gravity. Therefore, we prefer to make the simulation domain large enough to capture all relevant plasma flows and to simply close the upper simulation boundary for any plasma and heat flows.

For the magnetic field, though, we like to allow for "open" field lines instead of formulating a boundary condition that either enforces vertical or horizontal fields at the top boundary; as we describe in section 4.2.

4.1. Closed atmospheric boundary

Above the upper boundary at 156 Mm the coronal temperature in standard 1D atmospheric stratifications would still rise (Bourdin 2014b). If we simply prescribe the temperature in the upper ghost layers from such a stratification, we would impose an unwanted thermal energy inflow downwards into the physical box due to heat conduction along the temperature gradient. Therefore, and in contrast to the lower atmospheric boundary in the photosphere, we need to employ a different boundary condition at the upper coronal boundary (in grid cell nz) that assures there is no heat inflow. We achieve this by forcing the temperature gradient to be zero at the upper physical boundary ($z|_{nz} = 156$ Mm) with a symmetric boundary condition for the three upper ghost cells $j \in \{1, 2, 3\}$ at the grid positions $z|_{nz+j}$ as:

$$T|_{nz+j} = T|_{nz-j}\,. \tag{11}$$

Since the temperature near the boundary is almost uniform, also inclined field lines see a symmetric temperature stratification.

We then need to set the density $\rho|_{nz+j}$ again consistent with a hydrostatic equilibrium:

$$\left.\frac{\partial p}{\partial z}\right|_{nz+j} = \rho|_{nz+j}\, g\big|_{nz+j}\,. \tag{12}$$

This leads us to an upper boundary condition for the density that would require a constant temperature, constant gravity, and symmetric grid distances at the top:[5]

$$\implies \quad \rho|_{nz+j} = \rho|_{nz-j} \exp\left[-\frac{1}{\gamma}\left(z|_{nz+j} - z|_{nz-j}\right)\frac{g|_{nz}}{T|_{nz}}\right]. \tag{13}$$

For a non-constant temperature, non-constant gravity, and arbitrary grid distances at the boundary, the density in the ghost layers can be formulated in a recursive way as

$$\implies \quad \rho|_{nz+j} = \rho|_{nz+j-1} \exp\left[-\frac{1}{2\gamma}\left(\Delta' z\big|_{nz+j-1}\frac{g|_{nz+j-1}}{T|_{nz+j-1}} + \Delta' z\big|_{nz+j}\frac{g|_{nz+j}}{T|_{nz+j}}\right)\right], \tag{14}$$

where $\Delta' z\big|_j$ denotes the vertical extent of the grid cell at $z(j)$. If the grid spacing is close to being equidistant near the boundary, we may simplify equation (14) to[6]

$$\implies \quad \rho|_{nz+j} = \rho|_{nz+j-1} \exp\left[-\frac{1}{\gamma}\left(z|_{nz+j} - z|_{nz+j-1}\right)\frac{g|_{nz+j} + g|_{nz+j-1}}{T|_{nz+j} + T|_{nz+j-1}}\right]. \tag{15}$$

4.2. Potential-field extrapolation

When the simulation domain is large enough, the magnetic field near the top boundary should have reached a nearly potential state. We may then use a purely potential field in the three ghost layers above the domain as boundary condition for the magnetic field. Because the field vectors above and below the upper boundary are not perfectly rotation free, we still obtain some currents at the physical boundary, which represent the relaxation from a nearly to a fully potential state. If these currents are strong, they might artificially heat the corona. To counter this effect, we employ an additional magnetic diffusivity below the boundary to smoothen the transition to the potential state and reduce these artificial currents, as we describe in section 4.3.1 below.

For the potential-field extrapolation into the ghost layers, we Fourier-transform the magnetic vector potential $\boldsymbol{A}$, extrapolate it by smoothing out contrasts, and then transform it back:

$$\widehat{\boldsymbol{A}}(k_x, k_y, t)\big|_{nz} = \int \boldsymbol{A}(x, y, t)\big|_{nz} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}}\, \mathrm{d}^2\boldsymbol{r}, \tag{16}$$

where the vector $\boldsymbol{r} = (x, y)$ lies in the horizontal plane and $\boldsymbol{k} = (k_x, k_y)$ denotes the horizontal wave vector. We extrapolate from the location of the physical boundary $z(nz)$ to the coordinates $z(nz + j)$ of the three ghost layers $j \in \{1, 2, 3\}$ with[7]

$$\widehat{\boldsymbol{A}}(k_x, k_y, t)\big|_{nz+j} = \widehat{\boldsymbol{A}}(k_x, k_y, t)\big|_{nz} \exp\left[-|\boldsymbol{k}|\ \Delta'' z\big|_{nz+j}\right]. \tag{17}$$

We denote here the distance between the physical boundary and the extrapolated layer as $\Delta'' z\big|_{nz+j} = z(nz + j) - z(nz) > 0$. The normalised Fourier back transform of $\widehat{\boldsymbol{A}}$ finally

[5] In the `run.in` configuration file we set `bcz` to `'hs'` for the density and to `'s'` for the temperature.
[6] For non-constant gravity in the file `run.in` we set `bcz` to `'hse'` for the density and to `'s'` for the temperature.
[7] In the `run.in` configuration file we set `bcz` to `'pfe'` for the first magnetic field component, which also sets the other two components accordingly.

gives us the vector potential $\boldsymbol{A}$ in the upper three ghost layers. Because $\Delta''z\big|_{nz+j}$ is positive on the upper boundary, we smear out any contrasts in $\boldsymbol{A}$ with increasing height.

4.3. Diffusive swamp region

The potential-field extrapolation relaxes the magnetic field very quickly into a force-free state and hence strong currents may emerge at the upper physical boundary. Also a strong heat transport towards the upper simulation domain may become problematic, because the upper boundary is closed for heat flows and we may accumulate very high temperatures that become numerically demanding in the low-density plasma of the outer corona, where this plasma cannot radiate an excess of internal energy. To circumvent both problems, we use an additional diffusivity for the magnetic field and for the heat conduction below the physical boundary by implementing a so-called swamp region.

The swamp diffusivities act only in the upper part of the simulation domain to reduce the artificial effects on the upper boundary. We achieve a smooth transition by a height-dependent weighting function $w(z)$ that smoothly goes from 0 below to 1 within the swamp region. For this transition we use a cubic step function that starts at height z_s and ends at z_t. The swamp region is fully active above z_t so that $w(z \le z_s) = 0$ and $w(z \ge z_t) = 1$ with zero first derivatives at the edges of the transition function $w(z)$.[8]

4.3.1. Magnetic diffusivity

To diffuse out currents near the upper end of the simulation domain, we implement an additional isotropic magnetic diffusivity within the swamp region. This swamp diffusivity is of course omitted in the energy balance in order not to heat the corona artificially. We multiply the weighting function $w(z)$ to the constant swamp diffusivity η_s and obtain the height-dependent magnetic swamp diffusivity $\eta_s(z) = \eta_s\, w(z)$. Now we add the magnetic swamp diffusion term to the induction equation (shortened by "$\cdots$"):[9]

$$\frac{\partial \boldsymbol{A}}{\partial t} = \cdots + \eta_s(z)\, \Delta \boldsymbol{A} + \boldsymbol{e}_z\, \frac{\partial \eta_s(z)}{\partial z}\, \boldsymbol{\nabla} \cdot \boldsymbol{A}. \tag{18}$$

In the case of Bourdin *et al.* (2013) the simulation domain was large enough, so that currents at the top boundary were not problematic and hence no magnetic swamp diffusivity was used there.

4.3.2. Heat conduction

Similar to the magnetic swamp diffusivity (see above), we implement also a diffusivity that acts on the temperature as a constant, uniform, and isotropic heat conduction. For that, we add another term to the energy balance (shortened by "$\cdots$") and use χ_s as the swamp heat diffusivity constant:[10]

$$\frac{\partial T}{\partial t} = \cdots + \chi_s\, w(z)\, \Delta T. \tag{19}$$

[8] In the `run.in` configuration file the parameters `swamp_fade_start` and `swamp_fade_end` define the height of the smooth transition to the swamp region.

[9] In the `run.in` configuration file one may activate the magnetic swamp diffusivity by setting `swamp_eta` to a value larger than zero and similar to the parameter `eta`.

[10] In the `run.in` configuration file we set the parameter `swamp_chi` larger than zero to activate the heat swamp diffusion.

Again, the height-dependent function $w(z)$ provides a smooth transition between the physical regime and the swamp region.

We like note that χ_s acts either on the temperature T, the natural-logarithmic temperature $\ln T$, or the entropy ϵ, depending on which quantity is active in the simulation setup. Therefore, the physical unit of χ_s is not identical across these cases and may be different from the units of χ used for the regular global isotropic heat conduction $\chi \Delta T$.

4.3.3. Mass diffusion

Similar as for the heat conduction, we implement another swamp region that acts on the density with a diffusive term added to the continuity equation (shortened by "$\cdots$"):[11]

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = \cdots + \chi_\rho \, w(z) \, \Delta\rho. \tag{20}$$

Like for equation (19) we use the same diffusivity parameter χ_ρ with different physical units for either the density or the natural-logarithmic density, depending on what quantity is used for the simulation.

5. Massive-parallel methods

For high-performance computing applications it is crucial to reduce all computations that are serialised or restricted to few processors. The parallelisation of such computations therefore helps to scale an application efficiently to substantially more processors. In the following sections we describe some massive-parallel methods introduced to the *Pencil Code* in the years from 2009 to 2013.

One bottleneck in massive parallelisation for simulation runs like described in Bourdin *et al.* (2013) is the potential-field boundary condition that uses a Fast Fourier Transform (FFT), as well as the massive-parallel file input and output described in section 5.2.

5.1. Fast Fourier transform

Some boundary conditions, like a potential-field extrapolation, need to compute the FFT along both horizontal directions. This requires to collect once all data along x and then once along y. Less parallelised FFT routines collect all data on one processor, perform the FFT, and then distribute the transformed data back to all processors. During the computationally expensive communication and computation, a large number of processors are idle and their potential resources remain unused.

A more efficient method is to collect all data along the one direction which the FFT actually needs and to split the domain along the other direction; see the remapping scheme presented in figure 7. As a result, all processors in one (x, y)-layer may contribute in parallel to the computation of the FFT along one direction. Furthermore, the remapping of the data can be done in a parallel way, so that all communication finishes after only two communication cycles: one send and one receive operation. The trick is to let one half of the processors send first and then receive, while the other half first receive and then send their local data portion. This is actually faster than to collect all data on one processor, because

[11] We activate the density swamp region via the `swamp_diffrho` parameter in the `run.in` configuration file.

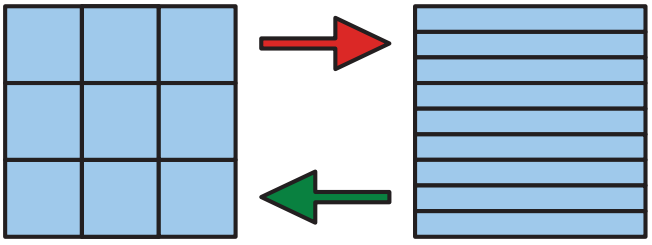

Figure 7. Data remapping strategy for the parallel FFTs in the module ``fourier_fftpack'' (colour online).

this can only be done in a sequential way and requires significantly more communication cycles for a large number of processors. The same holds true for the remapping back to the original subdomains.

The required transpose operation of the remapped data during a two-dimensional FFT needs more communication cycles, namely as many as the number of subdomains along the x-direction. Still, this number is typically less than the number of processors in one (x, y)-layer and the transpose can be parallelised between the processors in the y-direction.

Due to the participation of many processors in the FFT, the whole computation, including the multiple data remappings, is faster than the original scheme of the FFT for only one processor. When the domain is not split into subdomains along x, both schemes are of course similar, but this is usually not the case for large simulation runs with many processors.

5.2. Input/output

One challenge for large-scale simulation runs is the input and output (IO) of data snapshots, as well as their storage requirements. With clever IO strategies one cannot only perform IO operations faster, but also save significant amounts of storage space.

Traditionally, the *Pencil Code* stores data snapshots in a "distributed" manner, where each processor writes one file with its local subdomain portion of the data, like implemented in the ``io_dist'' module. This strategy has the advantage that the writing can be done fully in parallel, but at the cost of writing all inner ghost layers between any neighbouring processors, which contains overlapping and identical data. The smaller the subdomains get and the more processors we like to use in order to boost the computation speed, the larger is the amount of unneeded data that this distributed method requires to store. Also a fully distributed IO method will cause problems on any file system, because these are not made for thousands or more simultaneous IO requests.

In a first step, one might try to collect all data on one processor and write it out into one monolithic snapshot file. This strategy we call "collective" and it is implemented in the ``io_collect'' module. It turns out this method is not optimal regarding the IO speed, because one processor alone can only access a data snapshot in a sequential way. Nonetheless, we may omit to store all inner ghost cells; see the scheme displayed in figure 8.

The next step of improving the IO lies in combining the distributed with the collective strategy. This we implement as the ``io_collect_xy'' module, where all data is firstly collected in along (x, y)-layers by the leading processor in this layer and secondly

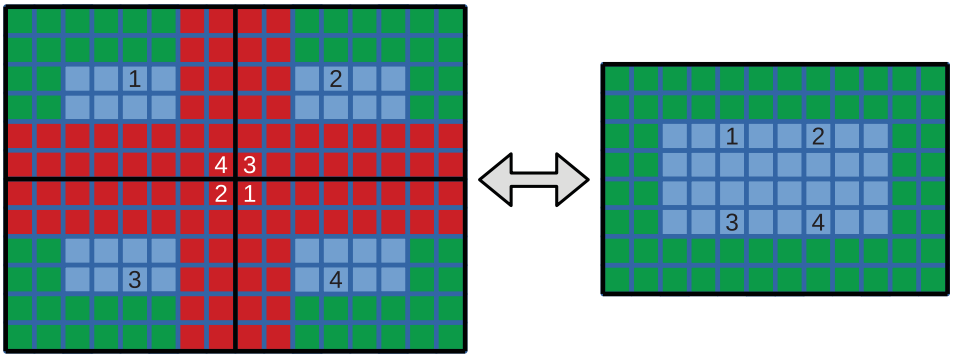

Figure 8. Reduction of required data space for monolithic files in an extreme case, where the needed outer ghosts are coloured in green and the physical domain is blue. The redundant inner ghost cells (see numbers) being saved with the distributed ``io_dist'' IO module in the *Pencil Code* are highlighted in red. Black lines depict the data boundaries for each file. There number of ghost layers is 2 for this example (colour online).

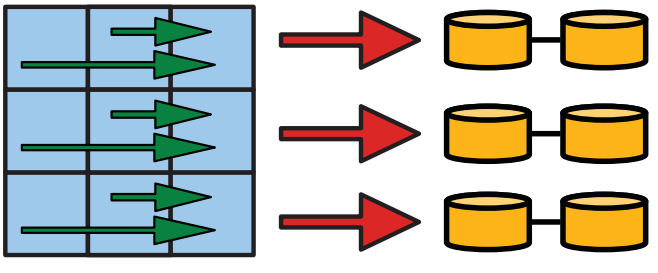

Figure 9. Collective IO strategy of the module ``collect_xy'', where data is collected in subdomain layers along (x, y) and is written to multiple files by the collecting (here rightmost) processors in each (x, y)-plane. The vertical separation of the layers depicts the z-direction in the simulation domain (colour online).

written by all (x, y)-leading processors in parallel; see figure 9. The reading access works in the same way: the (x, y)-leading processors read in parallel and then distribute the data within their layer. For this improvement one has to take the disadvantage of storing the inner ghost cells between all (x, y)-layers, which is still significantly less than storing all inner ghost cells. Still, this advanced method has substantial potential to accelerate the IO by some parallelisation.

Most modern IO methods usually use an intermediate software layers to optimise the number of parallel IO requests and to write out monolithic files that do not need to store any inner ghost cells at all. Such file formats therefore have the potential to save a substantial amount of storage space. In the *Pencil Code* we now provide two IO modules: ``io_mpi2'' that relies on the *MPI-2* standard and ``io_hdf5'' that uses the parallel *HDF5* software library for IO operations. Both use monolithic file formats and hence are optimal regarding the data storage requirements. In the same time, their IO routines are also optimised for scalability and speed; see comparison in table 1, section 6.

Table 1. Properties of the different IO strategies in *Pencil Code* for writing a full data snapshot.

module	time	storage	notes
distributed	2 s	**+31**%	**not scalable above 256 CPUs**
collect	**70** s	min.	one IO node collects globally
collect_xy	10 s	**+16**%	one IO node per (x,y)-layer
mpi2	8 s	min.	**binary IO hidden in MPI2**
hdf5	9 s	min.	portable, extendable structure

5.3. HDF5 file format

The *Pencil Code* is now capable of storing monolithic data snapshots with minimal storage requirements in the *HDF5* format. Snapshot files (that were previously in a raw binary format) are now stored in the self-explanatory *HDF5* format with an extendable data structure. All *HDF5* files carry the filename suffix ``.h5'' instead of the binary ``.dat'' file extension.

Once a user switches to the *HDF5* file format, the build process (`pc_build` or `make`) tries to automatically find the *HDF5* libaray and uses the location where the Fortran compiler wrapper is present.[12] For that, the ``$PATH'' environment variable must point to one of those compiler wrappers (`h5pfc` or `h5fc`). Otherwise, no configuration by the user is required. In a computing centre, this usually requires to load an environment module with the command ``module load ...hdf5...'', where all available modules can be listed with ``module avail''.

The classical snapshot files like `var.h5` do now contain the grid positions, dimensions of the setup, as well as some basic simulation settings in a separate data structure. This additional information can be suppressed to save storage space when a large number of small data snapshots is generated; set `lomit_add_data=T` within the section `run_pars` inside the `run.in` configuration file. The main simulation data is written out in components, like `ux`, `uy`, `uz`, `lnrho` or `rho`, etc. Unused components are of course not written. Inner ghost layers are cut for all quantities that are defined on grid cells. The *HDF5* snapshots are hence always monolithic and are stored in the directory `data/allprocs`.

A typical example of the content of a `var.h5` snapshot can be visualised with the tool ``hdfview''; see figure 10. The datasets (like `ux`) are listed in groups (like `data` or `settings`) that can be opened and closed by a double mouseclick. Some fundamental parameters of the simulation can be contained in snapshot files, e.g. `settings/precison` holds either an ``S'' for single or a ``D'' for double precision. The time of the snapshot is stored as a separate scalar double-precision dataset named `time`. Datasets can also be multi-dimensional arrays and the order of these arrays is in the canonical `Fortran` way, which means the first dimension of the array is along the z-direction. This implies that one has to transpose multi-dimensional data arrays for post-processing with languages like `C` or `Julia`/`Python`, while the datasets are naturally aligned for languages like `Fortran` or `IDL`.

[12] We switch to the *HDF5* format by setting `IO = io_hdf5` and `HDF5_IO = hdf5_io_parallel` in `src/Makefile.local`. On a standard `ubuntu 18.04 LTS` system, one needs to install the package ``libhdf5-openmpi-dev''. The packages ``hdf5-tools'' and ``hdfview'' contain optional tools for inspecting and modifying *HDF5* files.

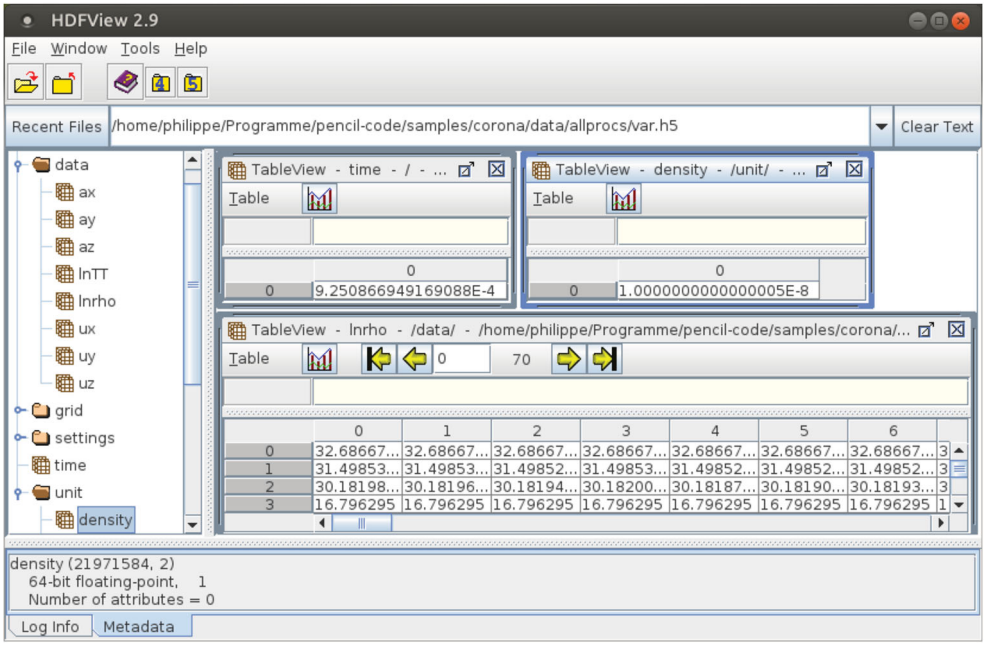

Figure 10. Example content of a ``var.h5'' file, here from the `corona` sample. The conversion factor from code units to physical units is 10^{-8} for the density. This sample stores the logarithmic density `lnrho` as three-dimensional data array, together with the logarithmic temperature `lnTT`, and all components of the vector potential and velocity field. The `time` of the snapshot is given as a scalar. The `settings` and `grid` are optional and are stored in each snapshot by default (colour online).

The so-called persistent data (that is usually associated per processor) is written to global arrays of the size (`nprocx`,`nprocy`,`nprocz`), where `nprocx` means the number of processors along the `x`-direction. Particle data gets collected from all processors and is stored in global arrays together with the information of the mapping of particles to the processors.

In addition to the snapshots, the module ``io_hdf5'' generates also a grid file `data/grid.h5` that contains the global grid data, as well as some fundamental parameters, like the size of the simulation domain, the number of grid points, grid distances, etc.; see figure 11.

For inspecting *HDF5* files from a text-based command line, one can use the ``h5dump'' command. The optional parameter ``-H'' allows to see the file structure (groups and datasets) without printing the actual data.

In table 1 we compare the advantages and disadvantages of each IO method available in the *Pencil Code* for a setup with $1024 \times 1024 \times 256$ grid cells distributed on $8 \times 16 \times 8 = 1024$ processors. One monolithic snapshot has a size of 17 GB and the timings were obtained on the *JuRoPA* supercomputer with the *Lustre* filesystem in a version from 2012. Bold-face entries in table 1 highlight the disadvantages. Please note these timings are hardly comparable with nowadays supercomputers, because the *JuRoPA* hardware and storage system are outdated and had beed decomissioned several years ago.

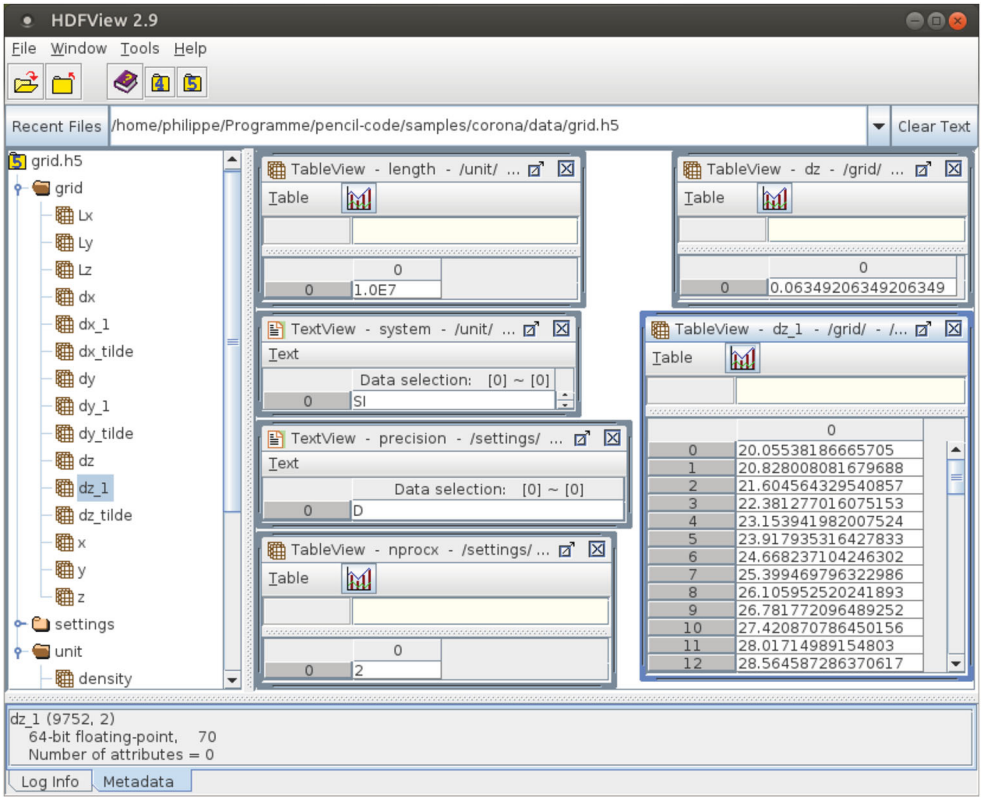

Figure 11. Content of the file ``grid.h5'' from the `corona` sample, where we use a non-equidistant grid in the `z`-direction. The unit length is 10^7 in `SI` units and we use double precision for floating-point numbers. The array `dz_1` contains the inverse grid spacing along the `z`-direction and the scalar `dz` gives the average grid spacing (colour online).

6. Conclusions

We use small-scale granulation and large-scale magnetic patches to drive a corona model by magnetic observations of an active region. The combination of both horizontal-velocity fields extends the spectrum of driving velocities to cover a large range in amplitues and in spatial length scales; see figure 4. An adaptice atmospheric boundary condition together with an adaptive chromospheric Newton cooling allows to formulate a flexible lower-boundary condition for the coronal part of the model. The varying magnetic pressure around strong polarities or the granulation driver may increase the plasma density in the intergranular lanes. This changes the density in the lower atmosphere and requires a matching response in from Newton cooling term. The new Newton cooling scheme implemented in the *Pencil Code* allows for expanding and shrinking atmospheric columns at the lower boundary, so that the temperature stratification is now adaptive.

For the upper boundary we have developed several mechanisms to diffuse away perturbations in the density and temperature. A new swamp-diffusion region can be used to relax non-force-free magnetic fields in order to reduce strong currents at the upper physical boundary, where "open" magnetic fields are desired but are in a force-free state.

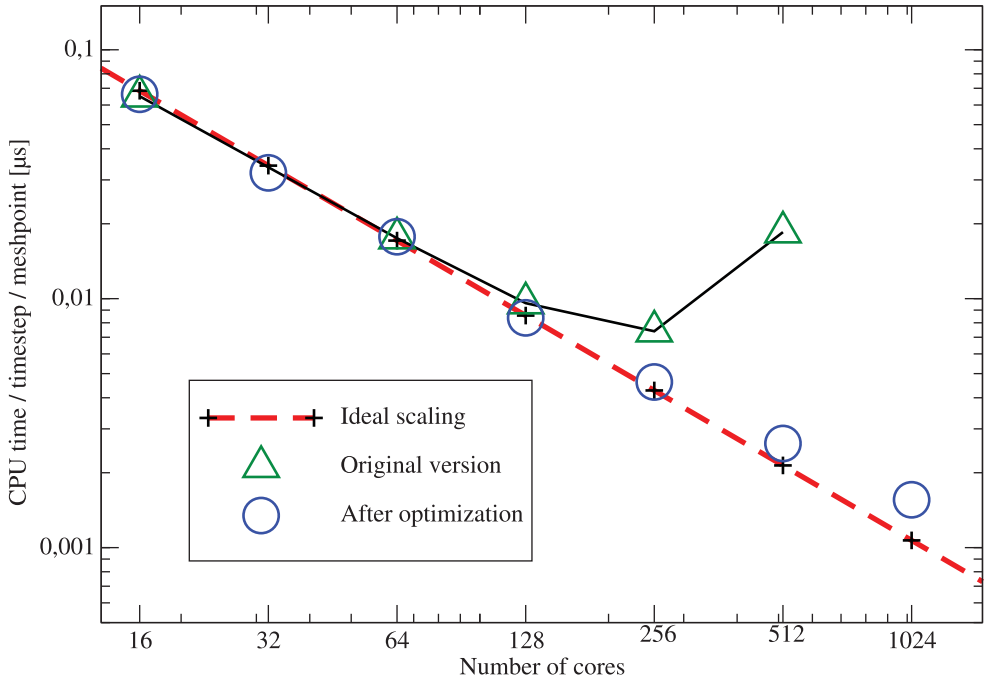

Figure 12. Comparison of the hard scaling before (triangles) and after (circles) the implementation of a massive-parallel FFT in the *Pencil Code* versus the theoretical scaling limit (dashed red line) (colour online).

The IO strategy is crucial for large-scale models in nowadays science where large amounts of data are generated. We find that only the modules ``io_mpi2'' and ``io_hdf5'' are optimal regarding storage requirements and scalability. Of these two, only ``io_hdf5'' offers a transparent, portable, and extendable data structure that allows to change the file formats without loosing backwards compatibility for the main code and for any data analysis scripts. Furthermore, almost all modern data analysis languages provide at least reading routines for the *HDF5* file format.

Boundary conditions that make use of a FFT, like a potential-field extrapolation, benefit from massive-parallel FFT implementation, like the routine ``fft_xy_parallel'' provided in the ``fourier_fftpack'' module. The scalability of such boundary conditions is significantly improved as compared to the original ``fourier_transform'' set of routines.

Altogether, the scalability of the *Pencil Code* could be improved substantially with massive-parallel methods that we implemented for the IO modules and for the FFT used by the potential-field boundary condition 'pfe'. We show a comparison plot for the total runtime in figure 12 that we obtained with a constant global number of grid cells but for an increasing number of processors, which is usually called a "hard scaling" test.

The advantages of the *HDF5* file format become clear from the comparison in table 1. We recommend to concentrate future developments of IO routines on the ``io_hdf5'' module and to fade out support for older IO modules during the next years.

Acknowledgments

The results of this research have been achieved using the PRACE Research Infrastructure resource *Curie* based in France at TGCC, as well as *JuRoPA* hosted by the Jülich Supercomputing Centre in Germany. Hinode is a Japanese mission developed, launched, and operated by ISAS/JAXA, in partnership with NAOJ, NASA, and STFC (UK). Additional operational support is provided by ESA and NSC (Norway).

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Philippe-A. Bourdin http://orcid.org/0000-0002-6793-601X

References

Bingert, S., Zacharias, P., Peter, H. and Gudiksen, B., On the nature of coronal loops above the quiet sun network. *Adv. Space Res.* 2010, **45**, 310–313.

Bourdin, P.A., Denoising observational data. *Contr. Astron. Obs. Skalnate Pleso* 2011, **41**, 149–155.

Bourdin, P.A., *Observationally Driven 3D MHD Model of the Solar Corona Above a Magnetically Active Region*, 2014a (Berlin: uni-edition GmbH.), ISBN 978-3-944072-03-6.

Bourdin, P.A., Standard 1D solar atmosphere as initial condition for MHD simulations and switch-on effects. *Cent. Eur. Astrophys. Bull.* 2014b, **38**, 1–10.

Bourdin, P.A., Plasma beta stratification in the solar atmosphere: A possible explanation for the penumbra formation. *Astrophys. J. Lett.* 2017, **850**, L29 (5pp).

Bourdin, P.A., Bingert, S. and Peter, H., Observationally driven 3D MHD model of the solar corona above an active region. *Astron. Astrophys.* 2013, **555**, A123.

Bourdin, P.A., Bingert, S. and Peter, H., Coronal loops above an active region: Observation versus model. *Publ. Astron. Soc. Jpn.* 2014, **66**, 1–8.

Bourdin, P.A., Bingert, S. and Peter, H., Coronal energy input and dissipation in a solar active region 3D MHD model. *Astron. Astrophys.* 2015, **580**, A72.

Bourdin, P.A., Bingert, S. and Peter, H., Scaling laws of coronal loops compared to a 3D MHD model of an active region. *Astron. Astrophys.* 2016, **589**, A86.

Bourdin, P.A., Singh, N. and Brandenburg, A., Magnetic helicity reversal in the corona at small plasma beta. *Astrophys. J.* 2018, **869**, 2.

Chen, F., Peter, H., Bingert, S. and Cheung, M.C.M., A model for the formation of the active region corona driven by magnetic flux emergence. *Astron. Astrophys.* 2014, **564**, A12.

Cook, J.W., Cheng, C.C., Jacobs, V.L. and Antiochos, S.K., Effect of coronal elemental abundances on the radiative loss function. *Astrophys. J.* 1989, **338**, 1176–1183.

Gudiksen, B. and Nordlund, Å., Bulk heating and slender magnetic loops in the solar corona. *Astrophys. J. Lett.* 2002, **572**, L113.

Gudiksen, B. and Nordlund, Å., An ab initio approach to the solar coronal heating problem. *Astrophys. J.* 2005, **618**, 1020–1030.

Hansteen, V.H., Hara, H., Pontieu, B.D. and Carlsson, M., On redshifts and blueshifts in the transition region and corona. *Astrophys. J.* 2010, **718**, 1070–1078.

Klimchuk, J.A., On solving the coronal heating problem. *Sol. Phys.* 2006, **234**, 41–77.

Kosugi, T., Matsuzaki, K., Sakao, T., Shimizu, T., Sone, Y., Tachikawa, S., Hashimoto, T., Minesugi, K., Ohnishi, A., Yamada, T., Tsuneta, S., Hara, H., Ichimoto, K., Suematsu, Y., Shimojo, M., Watanabe, T., Shimada, S., Davis, J.M., Hill, L.D., Owens, J.K., Title, A.M., Culhane, J.L., Harra, L.K., Doschek, G.A. and Golub, L., The Hinode (Solar-B) mission: An overview. *Sol. Phys.* 2007, **243**, 3–17.

Lites, B.W., Akin, D.L., Card, G., Cruz, T., Duncan, D.W., Edwards, C.G., Elmore, D.F., Hoffmann, C., Katsukawa, Y., Katz, N., Kubo, M., Ichimoto, K., Shimizu, T., Shine, R.A., Streander, K.V., Suematsu, A., Tarbell, T.D., Title, A.M. and Tsuneta, S., The Hinode spectro-polarimeter. *Sol. Phys.* 2013, **283**, 579–599.

Martínez-Sykora, J., Hansteen, V. and Moreno-Insertis, F., On the origin of the type II spicules: Dynamic three-dimensional MHD simulations. *Astrophys. J.* 2011, **736**, 9.

Parker, E.N., Topological dissipation and the small-scale fields in turbulent gases. *Astrophys. J.* 1972, **174**, 499–510.

Parker, E.N., Nanoflares and the solar X-ray corona. *Astrophys. J.* 1988, **330**, 474–479.

Peter, H. and Bingert, S., Constant cross section of loops in the solar corona. *Astron. Astrophys.* 2012, **548**, A1.

Peter, H., Bingert, S., Klimchuk, J.A., de Forest, C., Cirtain, J.W., Golub, L., Winebarger, A.R., Kobayashi, K. and Korreck, K.E., Structure of solar coronal loops: From miniature to large-scale. *Astron. Astrophys.* 2013, **556**, A104.

Peter, H., Warnecke, J., Chitta, L.P. and Cameron, R.H., Limitations of force-free magnetic field extrapolations: Revisiting basic assumptions. *Astron. Astrophys.* 2015, **584**, A68.

Rappazzo, A.F., Velli, M., Einaudi, G. and Dahlburg, R.B., Coronal heating, weak MHD turbulence, and scaling laws. *Astrophys. J. Lett.* 2007, **657**, L47–L51.

Rappazzo, A.F., Velli, M., Einaudi, G. and Dahlburg, R.B., Nonlinear dynamics of the Parker scenario for coronal heating. *Astrophys. J.* 2008, **677**, 1348–1366.

Rempel, M., Numerical sunspot models: Robustness of photospheric velocity and magnetic field structure. *Astrophys. J.* 2012, **750**, 62.

Rempel, M., Extension of the MURaM radiative MHD code for coronal simulations. *Astrophys. J.* 2017, **834**, 10.

Ruiz Cobo, B., del Toro Iniesta, J.C., Rodriguez Hidalgo, I., Collados, M. and Sanchez Almeida, J.. Empirical model of an average solar granule. In *Cool Stars, Stellar Systems, and the Sun*, edited by R. Pallavicini and A.K. Dupree, Vol. 109 of *Astronomical Society of the Pacific Conference Series*, San Francisco, USA: Astronomical Society of the Pacific (ASP), 1996, p. 155.

Schrijver, C.J., Hagenaar, H.J. and Title, A.M., On the patterns of the solar granulation and supergranulation. *Astrophys. J.* 1997, **475**, 328–337.

Spiegel, E.A., The smoothing of temperature fluctuations by radiative transfer. *Astrophys. J.* 1957, **126**, 202.

Spitzer, L. and Härm, R., Transport phenomena in a completely ionized gas. *Phys. Rev.* 1953, **89**, 977–981.

Tsuneta, S., Ichimoto, K., Katsukawa, Y., Nagata, S., Otsubo, M., Shimizu, T., Suematsu, Y., Nakagiri, M., Noguchi, M., Tarbell, T., Title, A., Shine, R., Rosenberg, W., Hoffmann, C., Jurcevich, B., Kushner, G., Levay, M., Lites, B., Elmore, D., Matsushita, T., Kawaguchi, N., Saito, H., Mikami, I., Hill, L.D. and Owens, J.K., The solar optical telescope for the Hinode mission: An overview. *Sol. Phys.* 2008, **249**, 167–196.

van Ballegooijen, A.A., Asgari-Targhi, M. and Berger, M.A., On the relationship between photospheric footpoint motions and coronal heating in solar active regions. *Astrophys. J.* 2014, **787**, 87.

van Ballegooijen, A.A., Asgari-Targhi, M., Cranmer, S.R. and DeLuca, E.E., Heating of the solar chromosphere and corona by Alfvén wave turbulence. *Astrophys. J.* 2011, **736**, 3.

Wedemeyer-Böhm, S., Scullion, E., Steiner, O., Rouppe van der Voort, L., de La Cruz Rodriguez, J., Fedun, V. and Erdélyi, R., Magnetic tornadoes as energy channels into the solar corona. *Nature* 2012, **486**, 505–508.