

MHSXtraPy - A Python code for the extrapolation of magnetohydrostatic fields on the Sun using analytical solutions

L. Nadol,¹★ T. Neukirch,¹

¹*School of Mathematics and Statistics, University of St Andrews, St Andrews, KY16 9SS, United Kingdom*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

We present a Python code for calculating and displaying magnetic field extrapolations from given two-dimensional boundary conditions, specifically from solar surface magnetograms. The code implements analytical magnetohydrostatic models that incorporate the transition from non-force-free to force-free magnetic fields in the solar atmosphere. It allows for different parameterisations of this transition and includes functions to compute magnetic fields, plasma pressure, and density. Fast Fourier methods ensure efficient computation, and the output includes three-dimensional visualisations of field lines and plasma structures. The implementation is optimised for accessibility and speed, making it suitable for both research and educational purposes. The only prerequisite for running the code is a Python compiler. All source code, examples, input files, solutions, and instructions are available for download from GitHub.

Key words: Software – Numerical methods – Sun: MHS – Sun: magnetic field – Sun: extrapolation

1 INTRODUCTION

Magnetic field extrapolation methods based on photospheric observations as boundary conditions are an essential tool for investigations of the structure of the solar corona and the phenomena occurring within it.

Solar magnetic fields are categorised as force-free or as non-force-free depending on the ratio of plasma to magnetic pressure, also referred to as the plasma beta, β_P . As a consequence the magnetic fields in the lower layers of the solar atmosphere are regarded as non-force-free, while the coronal magnetic field is considered force-free. A transition between the two states takes place with increasing height; this change occurs mostly localised around a height of approximately 2 Mm above the photosphere (Gary 2001), a region referred to as the transition region.

However, force-free models are frequently used for the whole solar atmosphere as they are easy to apply and rely on a reduced set of equations. These models are categorised into potential, linear, and non-linear fields depending on the amplitude and spatial variability of the current density. While potential and linear models are commonly used as quick-look approaches, non-linear force-free (NLFF) methods can capture certain aspects of the physics occurring on the Sun more accurately, e.g. localised twist and shear in field lines (Wiegelmann et al. 2017; Wiegelmann & Sakurai 2021).

Both linear and non-linear force-free models neglect plasma forces and assume that the Lorentz force vanishes only allowing currents parallel to the magnetic field. Therefore, a disadvantage of force-free methods remains the inconsistency between the assumption of a force-free photosphere in these model and the fact that the observed photospheric boundary conditions are not force-free (based on the definition of β_P). An alternative to force-free models is given by

magnetohydrostatic (MHS) methods which additionally include perpendicular currents and as a consequence can capture the effects of pressure gradient and gravitational forces (Zhu et al. 2022).

By making additional assumptions, the MHS equations can lead to linear equations for which solutions can be obtained analytically without the need for numerical solvers. While such assumptions may limit the physical accuracy of the model, especially when it comes to representing twist in magnetic field lines, linear equations can be solved at much lower computational cost and therefore can be used as a complementary method to computationally expensive non-linear models. This is an advantage as recent improvements of observations and data availability lead to the need for magnetic field extrapolation models of ever higher resolution and run-time efficiency.

Analytical MHS models have been around for decades (e.g. Low 1991, 1992) and have been continuously advanced to include more flexibility when modelling currents perpendicular to the magnetic field and to increase computational performance (e.g. Neukirch & Wiegelmann 2019; Nadol & Neukirch 2025).

This paper presents the **MagnetoHydroStatic eXtrapolation Python** tool (short: MHSXtraPy), a code that was written specifically for the application of these analytical models. It provides an accessible, open-source tool that can be used for time-efficient investigations and visualisations of MHS fields from photosphere to corona. The language chosen is Python as it provides an ecosystem of libraries useful for our purposes.

In the following we first introduce the theory behind the tool focusing on the output quantities of the code; second, we demonstrate its functionality using a SHARP magnetogram (Bobra et al. 2014); and last, briefly discuss technical aspects of the implementation and conclusions.

★ E-mail: lmn6@st-andrews.ac.uk

2 MHSXTRAPY

This section gives a brief overview of the background theory used for MHSXtraPy. How this connects to and is represented in the application of the code is shown in figure 1.

2.1 Formulation

In the MHS case the system of differential equations

$$\mathbf{j} \times \mathbf{B} - \nabla p - \rho g \hat{\mathbf{z}} = 0, \quad (1)$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j}, \quad (2)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (3)$$

where $\mu_0 = 4\pi \times 10^{-7} \text{ Hm}^{-1}$ the permeability of a vacuum and g the constant gravitational acceleration ($g = 272.2 \text{ ms}^{-2}$ on the Sun), needs to be solved for the magnetic field vector $\mathbf{B}$, the current density $\mathbf{j}$ as well as the plasma pressure p and plasma density ρ in combination with the ideal gas law

$$p = \rho RT / \bar{\mu},$$

where $R = 8 \times 10^3 \text{ JK}^{-1}\text{kg}^{-1}$ the gas constant, T the temperature and $\bar{\mu} = 1.67262 \times 10^{-7} \text{ kg}$ the mean molecular weight. These are the four quantities MHSXtraPy determines.

2.1.1 Current Density

In order for an analytical solution of Equations (1), (2) and (3) to exist, a specific form of the current density is used,

$$\mu_0 \mathbf{j} = \alpha \mathbf{B} + f(z) \nabla B_z \times \hat{\mathbf{z}}, \quad (4)$$

where the function $f(z)$ governs the amplitude of the perpendicular currents in the system and α is the ratio of the toroidal to the poloidal components of $\mathbf{B}$ (see equation (7)). This function is therefore responsible for describing the transition from non-force-free to force-free magnetic fields in the equations of the model.

Originally an exponentially decaying amplitude of the form

$$f_L(z) = a_L \exp(-\kappa z), \quad (5)$$

was suggested by Low (1991, 1992), where a_L is an amplitude parameter for the degree of non-force-freeness on the photosphere and κ determines how fast the perpendicular currents drop off in the upper solar atmosphere, as $f_L(z) \rightarrow 0$ for $z \rightarrow \infty$. This solution has been used for extrapolation in Wiegmann et al. (2015, 2017).

Alternatively, Neukirch & Wiegmann (2019) suggested

$$f_{N+W}(z) = a_{N+W} \left[1 - b \tanh \left(\frac{z - z_0}{\Delta z} \right) \right] \quad (6)$$

to control the amplitude of the perpendicular currents with height z . This choice of $f(z)$ in equation (4) increases the parameter space, and as a consequence the flexibility one has in modelling the transition from non-force-free to force-free. Similar to a_L in equation (5) the amplitude of f_{N+W} is controlled by a_{N+W} . The additional parameters z_0 and Δz determine the height at which that transition occurs and the width over which it takes place, respectively. The parameter b can be interpreted as a "switch-off"-parameter as $b = 1$ corresponds to a fully force-free state being reached above z_0 ; $b \neq 1$, on the other hand, allows for a certain degree of non-force-freeness being maintained in the upper atmospheric layers (for details see Neukirch & Wiegmann 2019; Nadol & Neukirch 2025).

Both of the above options lead to analytical expressions for the magnetic field $\mathbf{B}$. Once $\mathbf{B}$ is calculated, $\mathbf{j}$ can be obtained using equation (4).

2.1.2 Magnetic Field

Using the poloidal-toroidal representation (e.g. Neukirch & Rastätter 1999)

$$\mathbf{B} = \nabla \times [\nabla \times (\Phi \hat{\mathbf{z}})] + \nabla \times (\alpha \Phi \hat{\mathbf{z}}) \quad (7)$$

of the magnetic field in combination with a current density of form (4) one can obtain a solution defined by

$$\Phi(x, y, z) = \iint_{-\infty}^{\infty} \bar{\Phi}(z; k_x, k_y) \exp[i(k_x x + k_y y)] dk_x dk_y, \quad (8)$$

where

$$\frac{d^2 \bar{\Phi}}{dz^2} + [\alpha^2 - k^2 + k^2 f(z)] \bar{\Phi} = 0 \quad (9)$$

with $k^2 = k_x^2 + k_y^2$, k_x and k_y the wave numbers in x - and y -directions. If equation (5) is used, $\bar{\Phi}$ is expressed in terms of Bessel functions, and if equation (6) is used $\bar{\Phi}$ is given by hypergeometric functions.

Nadol & Neukirch (2025) used an asymptotic form of (6) in order to avoid the computationally expensive hypergeometric solutions. In this case the solutions for $\bar{\Phi}$ are defined by combinations of exponential functions. This asymptotic solution can be used when certain assumptions about Δz are made. This solution requires Δz small, such that $\omega = (z - z_0)/\Delta z$ becomes either large and positive, or large and negative, depending on the sign of $z - z_0$. If Δz is much larger than z_0 , the absolute value of ω does not become large for moderate values of z . Therefore, Δz must be smaller than z_0 . In this case the asymptotic solution can be obtained under the assumption that equation (6) is approximated by a step function (Nadol & Neukirch 2025).

The three MHS solutions described above are implemented in MHSXtraPy. Potential and linear force-free (LFF) extrapolations can also be calculated by making certain parameter choices: for potential fields one has to choose $\alpha = a = 0$, while $\alpha \neq 0$ in combination with $a = 0$ leads to the LFF case, where a refers to either a_L in equation (5) or a_{N+W} in equation (6).

The extrapolation is based on a photospheric magnetogram as boundary condition at $z = 0$ and assumes $\mathbf{B} \rightarrow 0$ for $z \rightarrow \infty$. For handling the boundary condition the code makes use of two assumptions: (1) The photospheric magnetogram used as boundary condition at $z = 0$ is flux-balanced. In cases of a bottom boundary condition for which the magnetogram is not flux-balanced the method suggested by Seehafer (1978) is used to generate a flux-balanced lower boundary condition; (2) The lateral boundary conditions (x - and y -directions) are assumed to be periodic. Under these assumptions the boundary condition $B_{z,Obs}(x, y, 0)$ can be represented by a Fourier series of the form

$$B_{z,Obs}(x, y, 0) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h_{nm} \exp(ik_n x + ik_m y) \quad (10)$$

with coefficients $h_{nm} \in \mathbb{C}$, $k_n = 2\pi n/L_x$ and $k_m = 2\pi m/L_y$, where L_x and L_y are the normalising length scales in the x - and y -directions. From this the magnetic field components are obtained from equation (7) (for details see e.g. Nadol 2025).

Generally, using the method described above the magnetic field $\mathbf{B}$ can be calculated analytically for all arbitrary (x, y, z) . However, the boundary condition will usually be given on a discrete grid. In case of an observational boundary condition $B_{z,Obs}$ this grid is pre-determined and has a finite resolution: If the resolution of this "data grid" is given by N in x -direction and M in y -direction, the maximal number of Fourier modes in the x - and y -directions is restricted by N and M , respectively. Therefore, the sums in equation (10), and

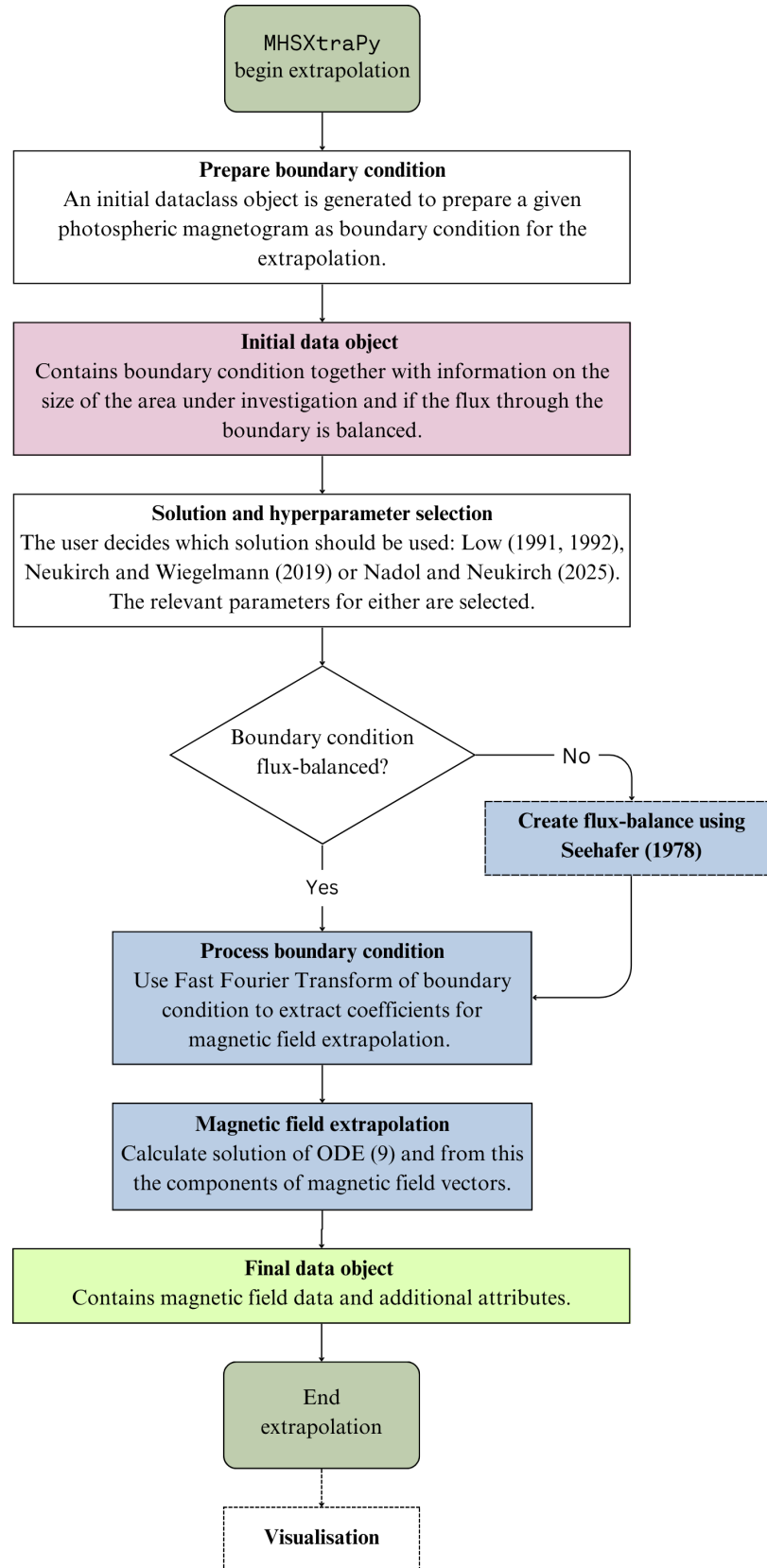


Figure 1. Flow chart representing the application of MHSXtraPy.

consequently for the magnetic field components, will be truncated. In the code the maximally possible number of Fourier modes is set as default, but a restricted number of modes $N_{F,x} \leq N$ in x -direction and $M_{F,y} \leq M$ in y -direction can also be selected. MHSXtraPy calculates the coefficients h_{nm} for $1 \leq n \leq N_{F,x}$ and $1 \leq m \leq M_{F,y}$ from the output of a standard numerical Fast Fourier Transform (FFT) calculation. These coefficients can be used together with the solution of equation (9) to calculate B_x , B_y and B_z . Restricting $N_{F,x}$ and $M_{F,y}$ as described above could prove useful for future runtime optimisations of the code, as in general, most FFT implementations perform best on certain list sizes as they utilise the algorithm by Cooley & Tukey (1965) which is optimised for input lengths that are powers of two.

Truncating the Fourier sums does still allow the calculation of the magnetic field and other quantities at arbitrary points (x, y, z) . However, to increase the numerical efficiency of the code it only calculates and stores these quantities on a discrete grid of points ("graphics grid"). Interpolation between these grid points is, for example, used to plot magnetic field lines. Although the size of the graphics grid and the size of the data grid (provided by the boundary magnetogram) are independent of each other they are set to have the same resolution in the x - and y directions in the current version of the code. The resolution in the z -direction of the graphics grid is set to be the maximum height of the computational domain divided by the pixel size of the magnetogram.

2.1.3 Plasma Pressure and Density

In the MHS solution plasma pressure and density are given by

$$p(x, y, z) = p_b(z) - \frac{f(z)}{2\mu_0} B_z^2, \quad (11)$$

$$\rho(x, y, z) = \frac{1}{g} \left(-\frac{dp_b(z)}{dz} + \frac{df(z)}{dz} \frac{B_z^2}{2\mu_0} + \frac{f(z)}{\mu_0} \mathbf{B} \cdot \nabla B_z \right), \quad (12)$$

where p_b is an integration "constant" (in x and y , therefore allowed to vary with z) with $p_b(z) > 0$ for all z . Methods to choose p_b include both a priori (as done here) and posteriori options (as for example done in Wiegmann et al. 2015). Both methods are prone to different difficulties, e.g. negative pressure and density values potentially appearing somewhere in the domain (primarily in the a priori case), or unrealistically large values of pressure and density being generated somewhere in the domain (primarily in the a posteriori case). Therefore, MHSXtraPy is primarily used to obtain the deviations in pressure and density from the background pressure p_b and background density $\rho_b = -\frac{dp_b(z)}{g dz}$. These deviations from hydrostatic balance are explicitly given by

$$\Delta p = -\frac{f(z)}{2\mu_0} B_z^2, \quad (13)$$

$$\Delta \rho = \frac{1}{g} \left(\frac{df(z)}{dz} \frac{B_z^2}{2\mu_0} + \frac{f(z)}{\mu_0} \mathbf{B} \cdot \nabla B_z \right). \quad (14)$$

Additionally, MHSXtraPy provides two different background atmosphere models, but when using these one needs to ensure that the full pressure and density stay positive and do not become unrealistically large.

The first one is based on a background temperature profile represented by a hyperbolic tangent connecting a photosphere of nominal value 5600 K and a corona of nominal value 2.0×10^6 K. The second one uses linear interpolation between given temperatures at given heights. In both cases, pressure and density are obtained using the ideal gas law together with the Boltzmann constant

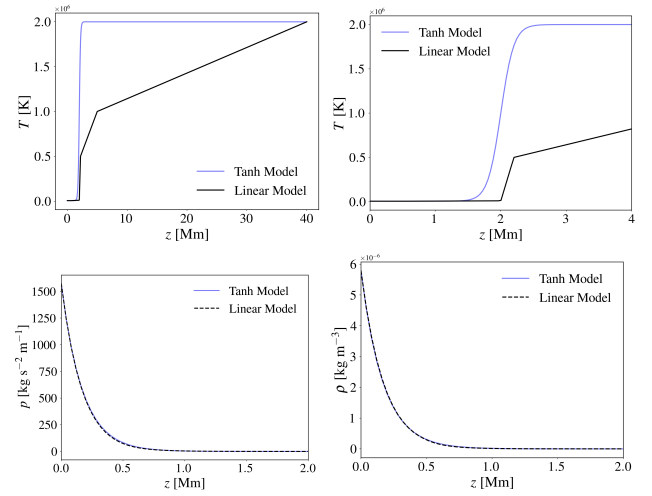


Figure 2. Examples for the two background atmospheres implemented in MHSXtraPy. The two figures in the top row show the temperature curves both for the full extrapolation height 40 Mm (on the left) and for a region around the transition region (up to 4 Mm above the photosphere). The hyperbolic tangential temperature profile (blue) is calculated with photospheric temperature 5600 K and coronal temperature 2×10^6 K. The linear temperature profile (black) is interpolated between 5600 K, 5200 K, 10^4 K, 5×10^5 K, 10^6 K and 2×10^6 K at heights 0 Mm, 0.5 Mm, 2.0 Mm, 2.2 Mm, 5.0 Mm and 40.0 Mm above the photosphere. The two figures in the bottom row show the resulting pressure and density curves, respectively, for the two temperature profiles.

$k_B = 1.380649 \times 10^{-23}$ J K⁻¹, mean molecular weight $\bar{\mu}$ and solar gravitational acceleration g .

Examples for both background atmosphere models can be seen in Figure 2. The first model, which uses a hyperbolic tangential function for the temperature profile, is shown in blue. While this temperature profile is not fully realistic, it allows for modelling a sharp increase in temperature at the height of the transition region, here at $z = 2$. The linear interpolation model, here shown in black, is an empirical choice motivated by observed values of temperature in the solar atmosphere and letting the user introduce as many given steps as desired. It was included to provide an alternative to the purely analytical model.

Generally, any background model requires dp_b/dz strictly negative in the whole domain; this excludes for example the model by (Vernazza et al. 1981).

2.2 Example

In this section we demonstrate the functionality of the code by presenting an example application using an SDO/HMI observation of active region NOAA 11158 (SHARP 377) as boundary condition. This region emerged on 2011-02-12 and was observed over 5 days by SDO/HMI. The snapshot used as boundary condition here was observed at midnight TAI¹ on 2011-02-14; both the HMI magnetogram as well as a matched AIA 171 observation are shown in figure 3.

2.2.1 Data preparation

SDO/HMI observes full disk magnetograms, such that the obtained image of the line-of-sight magnetic field component needs to be cut

¹ International Atomic Time, see e.g. https://gssc.esa.int/navipedia/index.php/Atomic_Time.

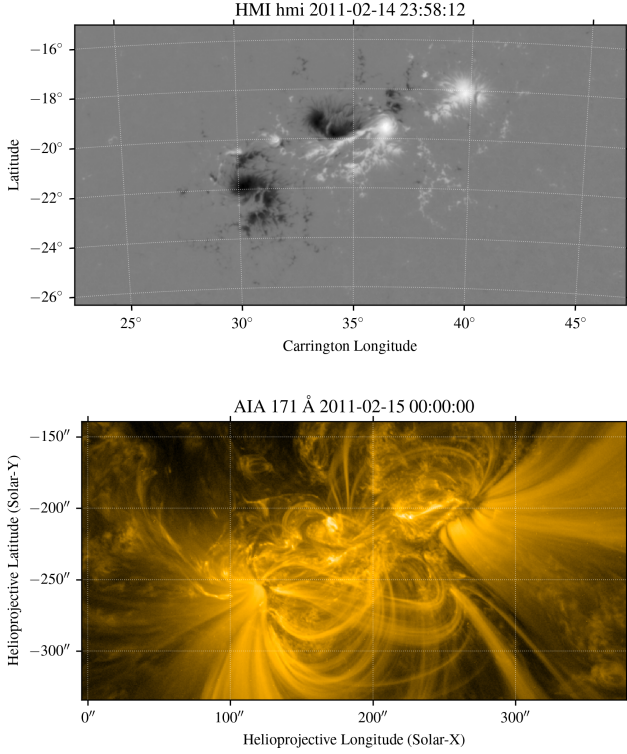


Figure 3. Magnetic and coronal observations of the investigated active region on the Sun. Top panel: Radial SHARP magnetogram as observed by SDO/HMI on 2011-02-14 at 23:58:12 TAI. Bottom panel: Extreme ultraviolet image from the Atmospheric Imaging Assembly (AIA) at 171 Å, taken shortly after on 2011-02-15 at 00:00:00 TAI, displaying the coronal loops associated with the active region.

to an area around the (active) region, that one aims to investigate. As MHSXtraPy uses Cartesian geometry, it is important that the region of interest is close to the disk centre and the cutout small enough that neglecting the curvature of the solar surface is appropriate and the utilisation of such geometry justified. Additionally, while cutouts of smaller size are favourable for shorter runtimes, they may also lead to interesting features and information getting lost. This trade-off between increased numerical efficiency and minimising the effects of the boundaries needs to be considered when using observational data as input.

Specifically, it is advantageous if the active region under investigation is reasonably well isolated. Users of the code could consider to modify the boundary condition at $z = 0$ provided by an observed magnetogram by adding a zone in the x - and y - directions, a few pixels in width, with $B_z = 0$. While this increases the grid size the method is fast enough to allow for this adjustment. However, this essentially forces the field lines to be horizontal above the added regions.

For the example shown here we have used a preprocessed SHARP region that fulfils the above criteria. The used cutout extends approximately 271 Mm in x -direction (longitudinal) and 137 Mm in y -direction (latitudinal). Its magnetic field strength ranges from -2565.05 to 2617.67 Gauss.

2.2.2 Solution selection and parameters

As mentioned above the code allows for the application of three different solutions: (1) the solution in which the transition from non-force-free to force-free is controlled by equation (5) (Low 1991); (2) the solution in which this transition is controlled by equation (6) (Neukirch & Wiegmann 2019); and (3) the solution in which this transition is controlled by the step function approximation of equation (6) (Nadol & Neukirch 2025).

For the example here, we have chosen solution (3). In this case the parameters α , a_{N+W} , z_0 and Δz need to be specified.² Theoretically, the parameter α can be chosen freely and a_{N+W} can be chosen depending on the value of α as it is bounded by

$$a_{N+W,max} = \frac{k_{min}^2 - \alpha^2}{(1+b)k_{min}^2},$$

where $k_{min}^2 = \min \{k^2 = k_n^2 + k_m^2 \mid \forall n, m\}$. While α is responsible for how much the resulting field lines twist, a_{N+W} specifies the amplitude of the current perpendicular to the magnetic field on the photosphere. As both these effects are frequently observed on the Sun, we want to choose α and a_{N+W} in practice in a way, that the effect of both are balanced and both resulting phenomena are included in the model to a meaningful extent.

If one wanted to prioritise either of the effects, the respective value must be chosen larger (and the other value smaller). While this is possible, it might be advantageous to use a non-linear force-free model in the case of strongly twisted magnetic field, but as a consequence no perpendicular currents are included in the model. Ideally, a non-linear MHS solution would be used, but this requires significantly larger computational resources, and usually longer runtimes.

General methods on how to determine the parameter α have been proposed for example by Leka & Skumanich (1999) or Hagino & Sakurai (2004). The latter one has been implemented in the code. These methods can be applied when measurements of all three vector components are available on the photosphere. If only line-of-sight observations are available one must use other approaches to finding a good value of α : e.g. one can match the magnetic field geometry to other EUV observations to find a value of α for which the extrapolated field lines match best with visible structures (see e.g. Wiegmann & Madjarska 2023). Generally, applying the MHS model in cases where the "optimal" value of α leads to very small values of a , makes the use of the model redundant compared to linear force-free models.

In this example, we choose $\alpha = 0.01$ for illustrative purposes as this allows for relatively large values of a_{N+W} as $a_{N+W,max} = 0.4069$ (for $b = 1$). We have chosen $a_{N+W} = 0.4$. The other two parameters z_0 and Δz are chosen such that they agree with commonly observed values. The centre of the region over which the transition from non-force-free to force-free occurs is set to $z_0 = 2$ Mm, approximately the upper chromosphere, and the width over which it takes place is set to $\Delta z = 0.2$ Mm, approximately the width of the transition region.

2.2.3 Results

The results of our extrapolation calculation from the chosen boundary condition are shown in Figures 4, 5, 6 and 7.

² The approximation of equation (6) by a step function eliminates Δz from the expression for $f(z)$ in the calculation of $\mathbf{B}$. Nevertheless, Δz is still needed for the calculation of $\mathbf{j}$, p and ρ . Here $f(z)$ is directly used and therefore, not asymptotically approximated.

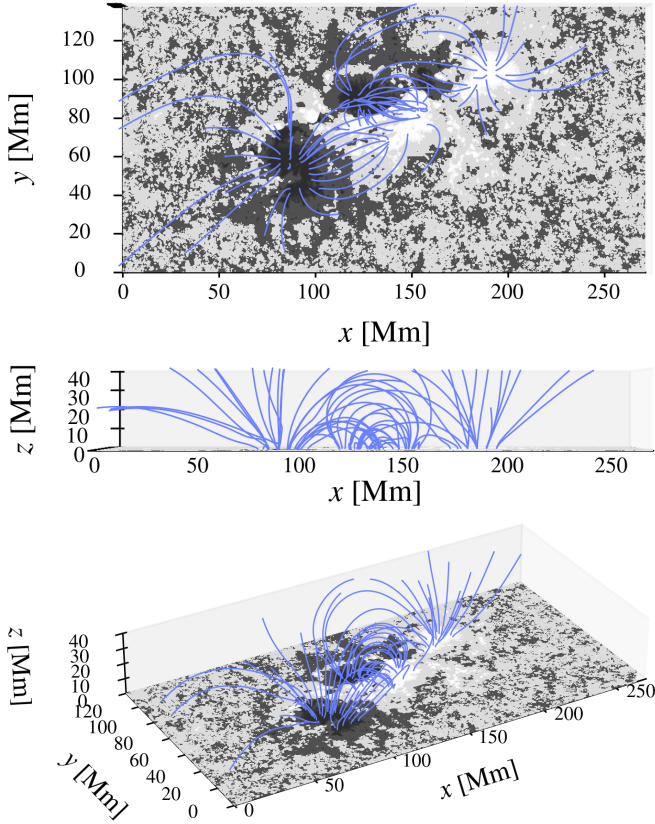


Figure 4. Field lines visualised by MHSXtraPy resulting from the magnetic field extrapolation above active region NOAA 11158.

Figure 4 shows extrapolated field lines above the region of interest from above, the side and from with an angular view onto the photosphere to emphasise the 3D nature of the model.

The model calculates field lines leaving the box³ above the regions of stronger absolute magnetic field as well as field lines closed inside the computational box between areas of opposite polarity. While the standard extrapolation height in the code is kept at 20 Mm such that the photosphere, chromosphere and the lower corona are included, we have chosen an upper boundary of $z_{max} = 40$ Mm in this example to showcase the possibilities the code has to offer. A higher upper boundary can lead to display field lines closed inside the box that extend higher into the atmosphere connecting negative and positive polarities further apart than just in the centre of the plane as seen in the middle panel of figure 4. Larger extrapolation heights might provide valuable insight into the structure of coronal loops, flares and CMEs. However, an even higher upper boundary suggests an intent to investigate the magnetic field further away from the Sun rather than closer to its surface rendering a NLFF model probably more suitable, as the effects of the non-force-free regions around $z_0 = 2$ Mm become more and more negligible as the extrapolation height z_{max} becomes larger.

Figure 5 shows the variation in plasma pressure and density resulting from the model at (x, y) where B_z is maximal on the photosphere

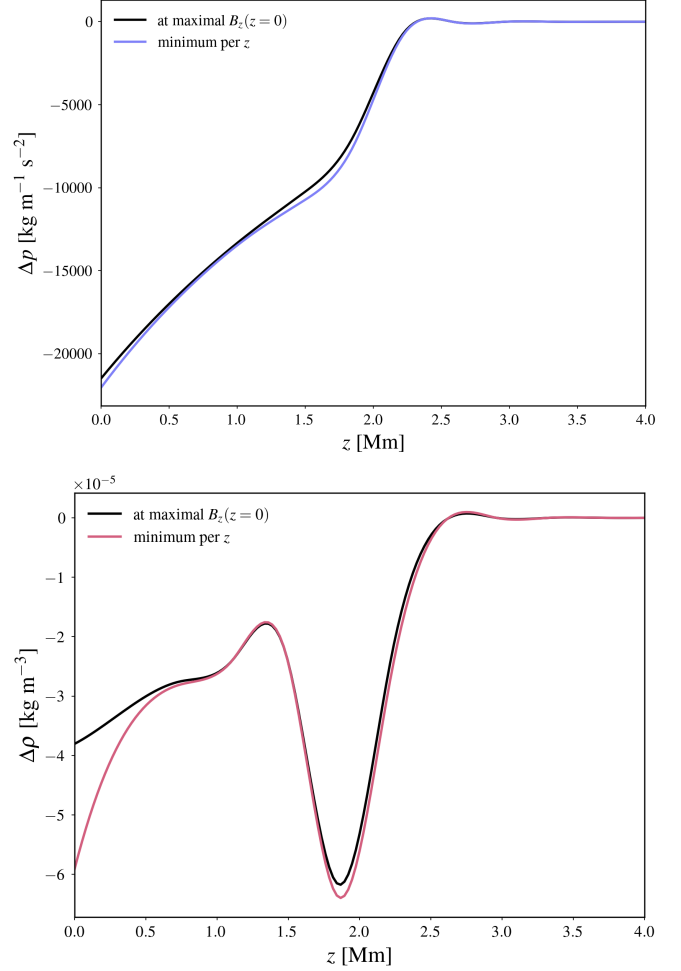


Figure 5. Variations in plasma pressure and density with height z above the photosphere, between $z = 0$ Mm and $z = 4$ Mm. The change from non-force-free to force-free occurs between approximately $z = 1.8$ and $z = 2.2$ Mm, which is particularly obvious in the density variation.

as well as the minimal variation at each z . The variations of both quantities with height z are shown between the photosphere and 4 Mm above it. Therefore, the centre of the plot coincides with the centre of the region over which the transition from non-force-free to force-free takes place. It is obvious that above $z = 2$ Mm the model approaches a force-free state as both pressure and density variation quickly tend towards zero here.

Figures 6 and 7 similarly show the pressure and density deviation from hydrostatic balance. Each panel in the figures corresponds to a different height above the photosphere (top to bottom: photosphere, transition region, lower corona) and shows how the plasma and pressure varies in the horizontal direction. The largest deviations appear around the positions of positive and negative polarities on the bottom boundary at $z = 0$. Consistent with figure 5 the pressure drops off with height and the density has a minimum around $z = 2$ Mm.

3 PERFORMANCE

The relevant quantities for the model (the magnetic field $\mathbf{B}$, the current density $\mathbf{j}$, as well as plasma pressure p and plasma density ρ) are

³ This refers to field lines seen as open in our computational domain. For this example the extrapolation domain is defined by $(x, y, z) \in [0, 271.13] \times [0, 137.39] \times [0, 40]$ Mm with resolution $744 \times 377 \times 109$.

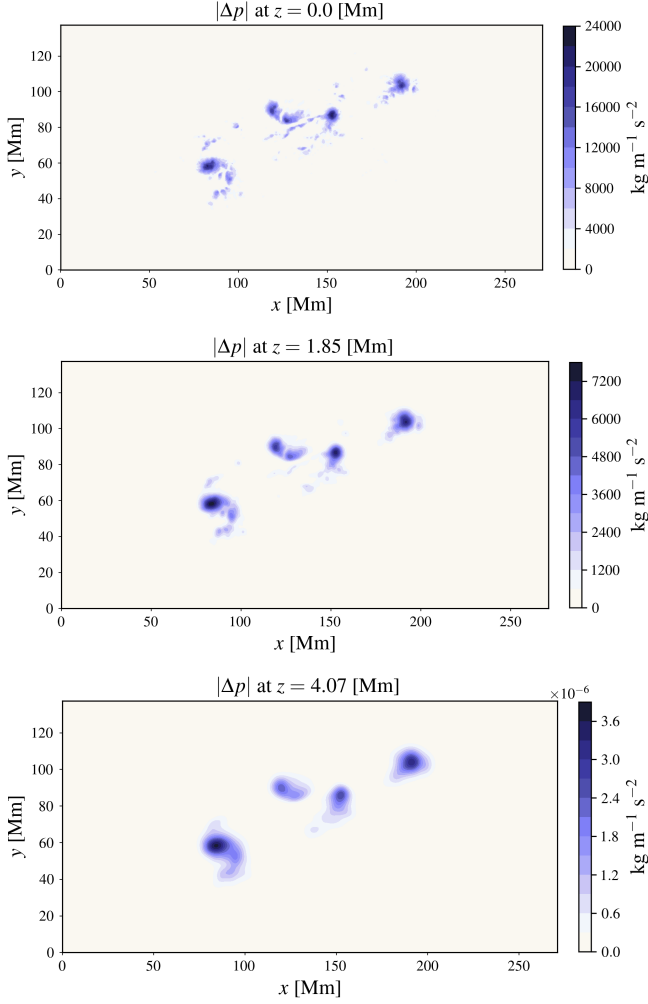


Figure 6. Variation in plasma pressure at different heights across the whole horizontal domain. The observed heights are $z = 0$ (the photosphere), $z = 1.85$ (approximately the height of the transition region), and $z = 4.07$ (in the lower corona). In the lower corona observations and values of β_p suggest that an almost force-free state should be reached. The model represents this well as seen in the scale being multiplied by 10^{-6} in the bottom panel.

multi-dimensional arrays. Such arrays are particularly well handled by the NumPy library (Harris et al. 2020), which features a rich interface for this data structure, making Python the natural choice for the development of MHSXtraPy. NumPy provides the class `ndarray`, which supports a large variety of operations that are essential for the time-efficient calculation of our analytical solution. Despite Python being a high-level, scripting language, `ndarray` operations are implemented natively and provide good performance due to easy vectorisation.

Next to NumPy, Python contains other scientific libraries advantageous for our purpose, including SciPy (Jones et al. 2001; Virtanen et al. 2020), Matplotlib (Hunter 2007), SunPy (The SunPy Community et al. 2020, 2023) and Astropy (The Astropy Collaboration et al. 2013, 2018, 2022). Additionally, Numba (Lam et al. 2015), an open source just-in-time (JIT) compiler that translates a subset of Python and NumPy code into fast machine code, is used. By using the industry-standard LLVM compiler library, Numba enables compiled numerical algorithms in Python to approach the speeds of C

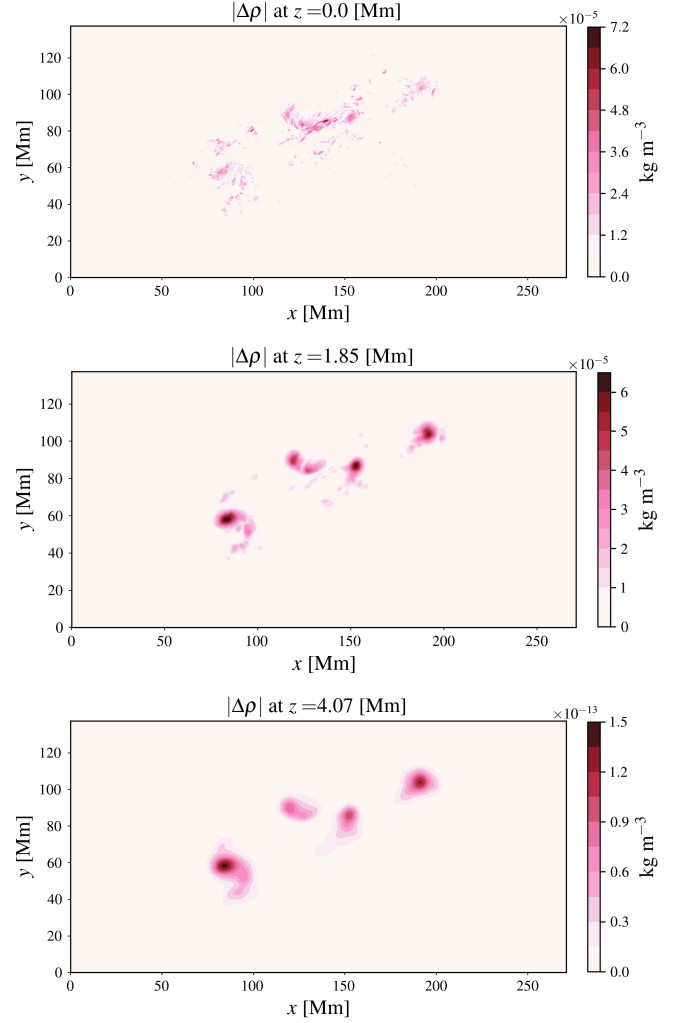


Figure 7. Variation in plasma density at different heights across the whole horizontal domain. The observed heights are $z = 0$ (the photosphere), $z = 1.85$ (approximately the height of the transition region), and $z = 4.07$ (in the lower corona). In the lower corona observations and values of β_p suggest that an almost force-free state should be reached.

or Fortran. This can be achieved by the user through a simple application of decorators rendering Numba an essential tool for runtime optimisation.

While vectorisation and parallelisation have been applied partly and have already lead to improvements in numerical efficiency, the potential for computational optimisation of the code is not yet exhausted. Especially the utilisation of parallelisation could be extended significantly: MHSXtraPy has so far only been executed using the Central Processing Unit (CPU) on a standard laptop computer as the JIT compiler used for the parallelisation of the code only optimises CPU performance, but does not utilise the Graphics Processing Unit (GPU). Therefore, further improvements could be expected through GPU use. This in combination with an additional increase in storage efficiency might allow for even more time-efficient computation with larger data sets.

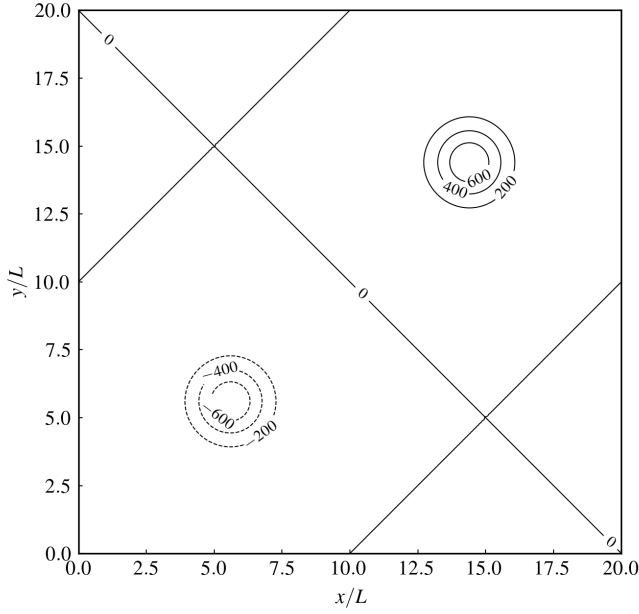


Figure 8. Artificial photospheric magnetic field $B_z(x, y, 0)$ in Gauss, as described in equation (15) for a normalising magnetic field strength $B_0 = 500$ G. Solid lines for positive values (sources), dashed lines for negative values (sinks).

3.1 Runtime test

To give an idea of the scale of domain that is practical to use with MHSXtraPy we time the execution of the calculation of the magnetic field using the IPython magic shell `%timeit`. The following test has been carried out on a MacBook Air (2020) M1 processor with 16 GB RAM.

We used `%timeit` to estimate the runtime of the function `b3d`, which can be found in the MHSXtraPy module of the same name. This function calculates all magnetic vector field components, B_x , B_y , B_z , as well as the partial derivatives of B_z , which are necessary to calculate the plasma pressure (12). This test was carried out using a simple, artificial initial condition of size n^2 given by

$$\frac{B_z(x, y, 0)}{B_0} = \frac{\exp(\lambda \cos(\tilde{x} - \mu))}{2\pi I_0(\lambda)} \frac{\exp(\lambda \cos(\tilde{y} - \mu))}{2\pi I_0(\lambda)} - \frac{\exp(\lambda \cos(\tilde{x} + \mu))}{2\pi I_0(\lambda)} \frac{\exp(\lambda \cos(\tilde{y} + \mu))}{2\pi I_0(\lambda)} \quad (15)$$

where $\tilde{x} = \pi(x/10 - 1)$ and $\tilde{y} = \pi(y/10 - 1)$, $\mu = 1.2/\pi + 1$ and $\lambda = 10$, for $x, y \in [0, 20]$, and I_0 is a modified Bessel functions of the first kind. In Figure 8 we show the contour lines of this boundary condition calculated for a normalising magnetic field strength $B_0 = 500$ G.

The discretisation n of the square, two-dimensional bottom boundary is increased during the test, such that the time complexity of the algorithm can be estimated. We have used n from 50 to 500 with a step size of 50. At the same time the vertical resolution $n_z = 200$ is kept fixed, such that the overall number of volume cells is $200n^2$ in each step. The runtime is averaged over 5 repeats of 5 loops each, resulting in overall 25 executions of `b3d` per configuration.

Figure 9 shows the result of this test comparing two variants of using MHSXtraPy: one for the case in which the boundary condition is not flux-balanced ("With Seehafer") and one for cases with naturally flux-balanced boundary condition ("Without Seehafer"). As the data points are approximately on a straight line in the log-log-scaled lower

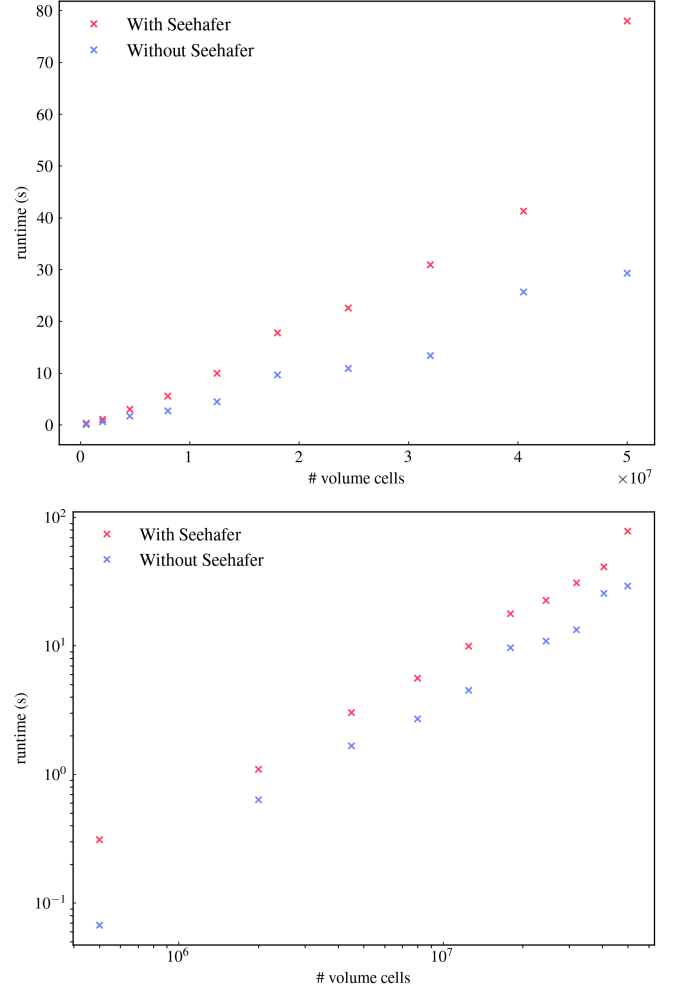


Figure 9. Cumulative runtime of the calculation of $\mathbf{B}$ and ∇B_z using the solution by Nadol & Neukirch (2025) (with and without using Seehafer 1978) by number of volume cells.

panel of Figure 9 it is implied that `b3d` is approximately a polynomial time complexity algorithm in all both cases.

Due to the limitations of computational power of the laptop computer this test has been carried out on, the maximal possible discretisation n was limited making this only an approximate classification of how the algorithm's runtime grows as the input size grows. While the runtime test above gives insight into the general time complexity of the algorithm and how the two variants compare to one another, its meaningfulness for the practical application to state-of-the-art line-of-sight magnetograms is limited as these typically vary significantly in size and complexity. To put this time complexity into perspective, we note that the extrapolation presented in Section 2.2 takes on average around 12.5 seconds.

4 CONCLUSIONS

MHSXtraPy is a newly developed code for solar magnetic field extrapolation which (i) is easy to apply due to its thorough documentation and in-code commentary, (ii) offers multiple opportunities for future development of the model and optimisation of the code, and (iii) pro-

vides an alternative to popular (NL)FF models taking into account other aspects of the physics on the Sun.

MHSXtraPy is a new and efficient tool for MHS extrapolation based on analytical solutions, offering a practical alternative to other methods. The code includes three analytical MHS solutions (plus options for potential and LFF extrapolation) and supports the inclusion of a non-force-free lower solar atmosphere, improving consistency with observed β_p values. Like all analytical models, it has limitations imposed by the assumptions needed to obtain such solutions and the parameter restrictions resulting from these assumptions.

Future developments will focus on optimising the code further and exploring more models. Further validation, through comparisons with observations and results from (NL)FF or other MHS extrapolations, will be essential to assess the model's reliability and guide improvements.

ACKNOWLEDGEMENTS

LN acknowledges financial support by the School of Mathematics and Statistics, University of St Andrews, throughout her PhD. TN acknowledges financial support by the UK's Science and Technology Facilities Council (STFC) via Consolidated Grants ST/S000402/1 and ST/W001195/1. LN thanks Jack Smith for helpful discussions and suggestions on the technical aspect of this work.

Software. The code and this work made extensive use of the following Python libraries/modules: NumPy (Harris et al. 2020), SciPy (Jones et al. 2001; Virtanen et al. 2020), Matplotlib (Hunter 2007), SunPy (The SunPy Community et al. 2020, 2023), Astropy (The Astropy Collaboration et al. 2013, 2018, 2022), and Numba (Lam et al. 2015).

DATA AVAILABILITY

The MHSXtraPy code is publicly accessible on GitHub <https://github.com/LMNadol/MHSXtraPy>.

All solar data used in this work were obtained from JSOC and are publicly available. The Jupyter notebook for the example presented in this paper can be found at <https://github.com/LMNadol/MHSXtraPy/blob/main/notebooks/paper.ipynb>.

Bugs and feature requests may be reported through the standard GitHub procedures, or via e-mail with the corresponding author.

CONFLICT OF INTEREST

Authors declare no conflict of interest.

REFERENCES

- Bobra, M. G., Sun, X., Hoeksema, J. T., Turmon, M., Liu, Y., Hayashi, K., Barnes, G., & Leka, K. D., 2014. The Helioseismic and Magnetic Imager (HMI) Vector Magnetic Field Pipeline: SHARPs - Space-Weather HMI Active Region Patches, *Sol. Phys.*, **289**(9), 3549–3578.
- Cooley, J. W. & Tukey, J. W., 1965. An algorithm for the machine calculation of complex fourier series, *Mathematics of Computation*, **19**(90), 297–301.
- Gary, G. A., 2001. Plasma beta above a solar active region: Rethinking the paradigm, *Solar Physics*, **203**(1), 71–86, doi: 10.1023/A:1012722021820.
- Hagino, M. & Sakurai, T., 2004. Latitude Variation of Helicity in Solar Active Regions, *Publications of the ASJ*, **56**(5), 831–843, doi: 10.1093/pasj/56.5.831.
- Harris, C. R., Millman, K. J., Walt, S. J. v. d., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. v., Brett, M., Haldane, A., Río, J. F. d., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., & Oliphant, T. E., 2020. Array programming with NumPy, *Nature*, **585**(7825), 357–362, doi: 10.1038/s41586-020-2649-2.
- Hunter, J. D., 2007. Matplotlib: A 2D Graphics Environment, *Computing in Science and Engineering*, **9**(3), 90–95, doi: 10.1109/mcse.2007.55.
- Jones, E., Oliphant, T., Peterson, P., et al., 2001. SciPy: Open source scientific tools for Python, <http://www.scipy.org/>.
- Lam, S. K., Pitrou, A., & Seibert, S., 2015. Numba: a LLVM-based Python JIT compiler, *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6, doi: 10.1145/2833157.2833162.
- Leka, K. & Skumanich, A., 1999. On the value of ' α AR' from vector magnetograph data, *Solar Physics*, **188**(1), 3–19, doi: 10.1023/a:1005108632671.
- Low, B., 1991. Three-dimensional structures of magnetostatic atmospheres. iii—a general formulation, *The Astrophysical Journal*, **370**, 427–434, doi: 10.1086/169829.
- Low, B., 1992. Three-dimensional structures of magnetostatic atmospheres. iv-magnetic structures over a solar active region, *The Astrophysical Journal*, **399**, 300–312, doi: 10.1086/171925.
- Nadol, L., 2025. *Three-dimensional solar magnetic field extrapolation using analytical magnetohydrostatic equilibrium solutions*, Ph.D. thesis, University of St Andrews.
- Nadol, L. & Neukirch, T., 2025. An efficient method for magnetic field extrapolation based on a family of analytical three-dimensional magnetohydrostatic equilibria.
- Neukirch, T. & Rastätter, L., 1999. A new method for calculating a special class of self-consistent three-dimensional magnetohydrostatic equilibria, *Astronomy and Astrophysics*, **348**, 1000–1004.
- Neukirch, T. & Wiegmann, T., 2019. Analytical three-dimensional magnetohydrostatic equilibrium solutions for magnetic field extrapolation allowing a transition from non-force-free to force-free magnetic fields, *Solar Physics*, **294**(12), 1–18, doi: 10.1007/s11207-019-1561-0.
- Seehafer, N., 1978. Determination of constant alpha force-free solar magnetic fields from magnetograph data, *Solar Physics*, **58**(2), 215–223, doi: 10.1007/BF00157267.
- The Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., Unther, H. M., Deil, C., Woillez, J., Conseil, S., Kramer, R., Turner, J. E. H., Singer, L., Fox, R., Weaver, B. A., Zabalza, V., Edwards, Z. I., Azalee Bostroem, K., Burke, D. J., Casey, A. R., Crawford, S. M., Dencheva, N., Ely, J., Jenness, T., Labrie, K., Lim, P. L., Pierfederici, F., Pontzen, A., Ptak, A., Refsdal, B., Servillat, M., & Streicher, O., 2013. Astropy: A community Python package for astronomy, *Astronomy and Astrophysics*, **558**, A33, doi: 10.1051/0004-6361/201322068.
- The Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Primary Paper Contributors, Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., Astropy Coordination Committee, Ardelean, C., Babej, T., Bach, Y. P., Bachetti, M., Bakanov, A. V., Bamford, S. P., Barentsen, G., Barmby, P., Baumbach, A., Berry, K. L., Biscani, F., Boquien, M., Bostroem, K. A., Bouma, L. G., Brammer, G. B., Bray, E. M., Breytenbach, H., Buddelmeijer, H., Burke, D. J., Calderone, G., Rodríguez, J. L. C., Cara, M., Cardoso, J. V. M., Cheedella, S., Copin, Y., Corrales, L., Crichton, D., D'Ávella, D., Deil, C., Depagne, E., Dietrich, J. P., Donath, A., Droettboom, M., Earl, N., Erben, T., Fabbro, S., Ferreira, L. A., Finethy, T., Fox, R. T., Garrison, L. H., Gibbons, S. L. J., Goldstein, D. A., Gommers, R., Greco, J. P., Greenfield, P., Groener, A. M., Grollier, F., Hagen, A., Hirst, P., Homeier, D., Horton, A. J., Hosseinzadeh, G., Hu, L., Hunkeler, J. S., Ivezić, Z., Jain, A., Jenness, T., Kanarek, G., Kendrew, S., Kern, N. S., Kerzendorf, W. E., Khvalko, A., King, J., Kirkby, D., Kulkarni, A. M., Kumar, A., Lee, A.,

- Lenz, D., Littlefair, S. P., Ma, Z., Macleod, D. M., Mastropietro, M., McCully, C., Montagnac, S., Morris, B. M., Mueller, M., Mumford, S. J., Muna, D., Murphy, N. A., Nelson, S., Nguyen, G. H., Ninan, J. P., Nöthe, M., Ogaz, S., Oh, S., Parejko, J. K., Parley, N., Pascual, S., Patil, R., Patil, A. A., Plunkett, A. L., Prochaska, J. X., Rastogi, T., Janga, V. R., Sabater, J., Sakurikar, P., Seifert, M., Sherbert, L. E., Sherwood-Taylor, H., Shih, A. Y., Sick, J., Silbiger, M. T., Singanamalla, S., Singer, L. P., Sladen, P. H., Sooley, K. A., Sornarajah, S., Streicher, O., Teuben, P., Thomas, S. W., Tremblay, G. R., Turner, J. E. H., Terrón, V., van Kerkwijk, M. H., de la Vega, A., Watkins, L. L., Weaver, B. A., Whitmore, J. B., Woillez, J., Zabalza, V., & Contributors), A., 2018. The astropy project: Building an open-science project and status of the v2.0 core package*, *The Astronomical Journal*, **156**(3), 123, doi: 10.3847/1538-3881/aabc4f.
- The Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., van Kerkwijk, M. H., Robitaille, T. P., Merry, B., Bachetti, M., Günther, H. M., Authors, P., Aldcroft, T. L., Alvarado-Montes, J. A., Archibald, A. M., Bódi, A., Bapat, S., Barentsen, G., Bazán, J., Biswas, M., Boquien, M., Burke, D. J., Cara, D., Cara, M., Conroy, K. E., Conseil, S., Craig, M. W., Cross, R. M., Cruz, K. L., D'Eugenio, F., Dencheva, N., Devillepoix, H. A. R., Dietrich, J. P., Eigenbrot, A. D., Erben, T., Ferreira, L., Foreman-Mackey, D., Fox, R., Freij, N., Garg, S., Geda, R., Glattly, L., Gondhalekar, Y., Gordon, K. D., Grant, D., Greenfield, P., Groener, A. M., Guest, S., Gurovich, S., Handberg, R., Hart, A., Hatfield-Dodds, Z., Homeier, D., Hosseinzadeh, G., Jenness, T., Jones, C. K., Joseph, P., Kalmbach, J. B., Karamehm-toglu, E., Kaluszyński, M., Kelley, M. S. P., Kern, N., Kerzendorf, W. E., Koch, E. W., Kulumani, S., Lee, A., Ly, C., Ma, Z., MacBride, C., Maljaars, J. M., Muna, D., Murphy, N. A., Norman, H., O'Steen, R., Oman, K. A., Pacifici, C., Pascual, S., Pascual-Granado, J., Patil, R. R., Perren, G. I., Pickering, T. E., Rastogi, T., Roulston, B. R., Ryan, D. F., Rykoff, E. S., Sabater, J., Sakurikar, P., Salgado, J., Sanghi, A., Saunders, N., Savchenko, V., Schwardt, L., Seifert-Eckert, M., Shih, A. Y., Jain, A. S., Shukla, G., Sick, J., Simpson, C., Singanamalla, S., Singer, L. P., Singhal, J., Sinha, M., Sipöcz, B. M., Spitler, L. R., Stansby, D., Streicher, O., Sumak, J., Swinbank, J. D., Taranu, D. S., Tewary, N., Tremblay, G. R., de Val-Borro, M., Kooten, S. J. V., Vasović, Z., Verma, S., de Miranda Cardoso, J. V., Williams, P. K. G., Wilson, T. J., Winkler, B., Wood-Vasey, W. M., Xue, R., Yoachim, P., Zhang, C., Zonca, A., & Contributors, A. P., 2022. The astropy project: Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package*, *The Astrophysical Journal*, **935**(2), 167, doi: 10.3847/1538-4357/ac7c74.
- The SunPy Community, Barnes, W. T., Bobra, M. G., Christe, S. D., Freij, N., Hayes, L. A., Ireland, J., Mumford, S., Perez-Suarez, D., Ryan, D. F., Shih, A. Y., Chanda, P., Glogowski, K., Hewett, R., Hughitt, V. K., Hill, A., Hiware, K., Inglis, A., Kirk, M. S. F., Konge, S., Mason, J. P., Maloney, S. A., Murray, S. A., Panda, A., Park, J., Pereira, T. M. D., Reardon, K., Savage, S., Sipöcz, B. M., Stansby, D., Jain, Y., Taylor, G., Yadav, T., Rajul, & Dang, T. K., 2020. The SunPy Project: Open Source Development and Status of the Version 1.0 Core Package, *The Astrophysical Journal*, **890**, 68–, doi: 10.3847/1538-4357/ab4f7a.
- The SunPy Community, Barnes, W., Christe, S., Freij, N., Hayes, L., Stansby, D., Ireland, J., Mumford, S., Ryan, D., & Shih, A., 2023. The SunPy Project: An Interoperable Ecosystem for Solar Data Analysis, *arXiv*, doi: 10.48550/arxiv.2304.09794.
- Vernazza, J. E., Avrett, E. H., & Loeser, R., 1981. Structure of the solar chromosphere. iii. models of the euV brightness components of the quiet sun, *The Astrophysical Journal Supplement Series*, (45), 635–725, doi: 10.1086/190731.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. v. d., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., Mulbregt, P. v., SciPy 1.0 Contributors, Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G.-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavic, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., Cardoso, J. V. d. M., Reimer, J., Harrington, J., Rodríguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tarte, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O., & Vázquez-Baeza, Y., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nature Methods*, **17**(3), 261–272, doi: 10.1038/s41592-019-0686-2.
- Wiegmann, T. & Madjarska, M. S., 2023. Automatic Computation of Linear Magneto-Hydro-Static Equilibria, *Solar Physics*, **298**(1), 3, doi: 10.1007/s11207-022-02094-2.
- Wiegmann, T. & Sakurai, T., 2021. Solar force-free magnetic fields, *Living Reviews in Solar Physics*, **18**(1), 1, doi: 10.1007/s41116-020-00027-4.
- Wiegmann, T., Neukirch, T., Nickeler, D. H., Solanki, S. K., Martínez Pillet, V., & Borrero, J. M., 2015. Magneto-static Modeling of the Mixed Plasma Beta Solar Atmosphere Based on Sunrise/IMaX Data, *The Astrophysical Journal*, **815**(1), 10, doi: 10.1088/0004-637X/815/1/10.
- Wiegmann, T., Neukirch, T., Nickeler, D. H., Solanki, S. K., Barthol, P., Gandorfer, A., Gizon, L., Hirzberger, J., Riethmüller, T. L., van Noort, M., Blanco Rodríguez, J., Del Toro Iniesta, J. C., Orozco Suárez, D., Schmidt, W., Martínez Pillet, V., & Knölker, M., 2017. Magneto-static Modeling from Sunrise/IMaX: Application to an Active Region Observed with Sunrise II, *The Astrophysical Journal*, **229**(1), 18, doi: 10.3847/1538-4365/aa582f.
- Wiegmann, T., Petrie, G. J., & Riley, P., 2017. Coronal magnetic field models, *Space Science Reviews*, **210**(1), 249–274, doi: 10.1007/s11214-015-0178-3.
- Zhu, X., Neukirch, T., & Wiegmann, T., 2022. Magnetohydrostatic modeling of the solar atmosphere, doi: 10.48550/arXiv.2203.15356.

This paper has been typeset from a $\text{\LaTeX}$ file prepared by the author.