

Visualisierungsinstitut der Universität Stuttgart  
University of Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Studienarbeit Nr. 2382

# **Visualisierung Magnetohydrodynamischer Felder**

Ivan Pilipenko

**Course of Study:** Computer Science

**Examiner:** Prof. Dr. Thomas Ertl

**Supervisor:** Dr. Filip Sadlo

**Commenced:** May 09, 2012

**Completed:** November 08, 2012

**CR-Classification:** I.6.6



## Abstract

Magnetohydrodynamics (MHD) describe the dynamics of conducting fluids by means of hydrodynamic and magnetic forces. The solar corona, the subject of research in this thesis, consists of plasma and is hence examined by MHD simulations based on measurements of the heliospheric and magnetic imager (HMI). These imagers obtain magnetic field data from the photosphere, the “surface” of the Sun. There are multiple fields which exist in the context of MHD. This thesis will concentrate on the three main fields: the magnetic, plasma velocity and the current density fields. The thesis first describes the methods needed to parse and precompute the data, the assumptions behind the simulation, as well as the HDF data format. In the main part, several visualization approaches are presented, including a method to analyze the advection property of the magnetic field, an analysis of the fields’ shapes, followed by the interrelations between them on global as well as local scale.



# Contents

1. Introduction	9
2. Related Work	11
3. Data Preparation and Access	13
3.1. MAS Simulation Details	13
3.2. MAS Data Format	14
3.2.1. Loading HDF Files and Conversion to Structured Grid Representation	14
3.2.2. Staggered Grid	15
3.2.3. Data Scaling	16
3.2.4. Conversion between Computational and Physical Space	16
4. Goals	19
4.1. Confirming the Advection Principle	19
4.1.1. First Approach	19
4.1.2. Second Approach	20
4.2. Observations from Field Visualizations	20
4.2.1. General Field Shape	20
4.2.2. Quantification of Current Leaving the Sun	21
4.2.3. Interrelations Between the Fields	21
5. Method	23
5.1. Converting the HDF into a Structured Grid	23
5.2. Confirming the Advection Principle	24
5.2.1. First Approach	24
5.2.2. Second Approach	24
5.2.3. Visualization	24
5.3. Methods Chosen for Field Visualizations	25
5.4. Ridge and Valley Surfaces	25
6. Results	27
6.1. Confirming the Advection Principle	27
6.1.1. First Approach	27
6.1.2. Second Approach	27
6.2. Observations from Field Visualizations	28
6.2.1. General Field Shape	28
6.2.2. Interrelation Between the Fields	30

6.3. Behavior of Current Density Field Swirls . . . . .	33
7. Discussion . . . . .	37
7.1. Data Problems . . . . .	37
7.1.1. Data Format . . . . .	37
7.1.2. Data Granularity . . . . .	37
7.1.3. Output Files . . . . .	37
7.2. Performance Issues . . . . .	38
7.2.1. Memory Usage . . . . .	38
7.2.2. Multi-Threading . . . . .	38
7.3. Documentation . . . . .	39
8. Conclusion . . . . .	41
A. Manual . . . . .	43
A.1. Dependencies and Setup . . . . .	43
A.1.1. Dependencies . . . . .	43
A.1.2. Setting up VTK . . . . .	43
A.1.3. Setting up ReadMAS and HDFExtract . . . . .	43
A.2. Program Usage . . . . .	44
A.2.1. Shell Selection . . . . .	44
A.2.2. Modes . . . . .	44
A.2.3. Hidden Switches . . . . .	45
A.2.4. Rendering Window Navigation . . . . .	46
B. Implementation . . . . .	47
C. Acknowledgemnts . . . . .	51
C.1. MAS Reader . . . . .	51
C.2. HDFExtract . . . . .	51
C.3. Vector and Matrix Classes . . . . .	51
C.4. Misc Code Snippets . . . . .	51
Bibliography . . . . .	53

# List of Figures

---

3.1. The Structure of the staggered MAS grid. Source: [MAS, user guide] . . . . .	15
6.1. Material derivation magnitude mapped to plasma velocity streamlines. Blue stands for low values, red for high values. . . . .	28
6.2. The shape of the magnetic field. . . . .	29
6.3. The shape of the plasma velocity field. . . . .	30
6.4. The shape of the current density field (left) and the isosurface at magnitude zero (right). . . . .	31
6.5. Interrelations between the current density (flame hues), the magnetic (blue hues) and the plasma velocity fields (violet). . . . .	32
6.6. Isosurface of the magnetic field (blue) with swirls (core lines in green) of current density (left) and the valley surface of the magnetic field magnitude (right). . .	33
6.7. Swirls of the current density flowing through a slice of the magnetic field. . . .	34
6.8. Result of overlapping the pillar structure (blue) with the ridge surface of the current density field (orange) (left) and isosurface of current density magnitude at level zero (magenta) (right). . . . .	34
6.9. The two types of swirls with different radial increase rate. Above: tornadoes, Below: spires . . . . .	36

# List of Tables

---

3.1. Table of MHD Variables . . . . .	14
3.2. Table of MAS scaling factors . . . . .	16





# 1. Introduction

While there has been a lot of progress on the simulation and visualization of magnetism and hydrodynamics, fewer works exist regarding the simulation of magnetohydrodynamics (MHD) and almost no techniques are available for the visualization of MHD data. MHD combines concepts from both magnetism and hydrodynamics. Hydrodynamics describes the behavior of fluids, usually using the Navier-Stokes equations or their simplified form, depending on which precision is required from the results. Magnetism describes a property of a matter that causes a reaction when a magnetic field is applied to it. In our case, the magnetic fluid is the Sun's plasma and what we will be looking at are the different aspects of its behavior.

The motivation behind this thesis is to probe what was already done in the related fields and what kind of data is present to allow researching the visualization of MHD data. Further, we would like to assess whether there are any adjustments needed regarding the format of said data, followed by different approaches to analyze the data to advance in the field of MHD visualization. Appropriate visualizations are important for example in the context of analyzing the mechanisms of the Sun.

The goals of this thesis include appropriate visualizations of the data provided by simulations by [MAS] on the basis of data from the Helioseismic and Magnetic Imager (HMI) onboard of the solar dynamics observatory (SDO). The first goal is to successfully convert the staggered grid MAS data to a [VTK]-compatible format. Once we have the data in the VTK format, our goal will be to visualize how magnetic lines move with respect to the flow of the involved plasma, in other words how well the magnetic field is advected with the plasma flow. With this, we hope to get an assessment of the so-called advection principle. Another goal is to investigate various aspects of the different fields, for example their general shape, an analysis of the current leaving the Sun surface, as well as the interrelations between the magnetic, plasma velocity and current density fields. This should improve the general understanding of the inner workings of the Sun.

The visualizations are obtained by visualization software developed in the context of this thesis as well as by exporting derived data to the .vtk format and visualizing them in ParaView.

## 1. Introduction

---

The thesis is structured as follows:

**Chapter 2 – Related Work:** Lists related works from the field of magnetism and hydrodynamics.

**Chapter 3 – Data Preparation and Access:** Explains the assumptions and mathematical methods used in this thesis.

**Chapter 4 – Goals:** Lists the goals of the thesis and explains them in detail.

**Chapter 5 – Goals:** Details the implementation and the reason behind certain choices.

**Chapter 6 – Results:** Covers the obtained results.

**Chapter 7 – Discussion:** Presenting a discussion on problems encountered while working on this thesis as well as possible solutions.

**Chapter 8 – Conclusion:** A short recapitulation of the thesis and possible future work.

**Appendix A – Manual:** Helps with the setup of the environment needed to be able to run the program and provides documentation for the software.

**Appendix C – Acknowledgements:** Contains acknowledgments regarding usage of foreign code.

## 2. Related Work

This chapter summarizes which work related to this thesis exists. Since there are fewer works on MHD compared to magnetism and hydrodynamics, most of the related work comes from these two fields.

Riley et. al [RLM<sup>+</sup>06] conducted a study comparing the results between using Potential Field Source Surface (PFSS) models and magnetohydrodynamic models. According to the study, PFSS has the advantage of simplicity as well as being less demanding with respect to resources. With more and more physics being incorporated into MHD though, this typically results in increasingly diverging results.

Goedbloed et. al released a book, “Principles of Magnetohydrodynamics” [GP08], giving an introduction on the principles of magnetohydrodynamics, followed by a book “Advanced Magnetohydrodynamics” [GKP10] which builds on the material discussed in the first book.

Machado et. al [MSM<sup>+</sup>12] showed concepts for 3D visualization of solar dynamics data, describing coordinate systems and projection techniques needed. They also applied methods from volume rendering and flow visualization, as well as introduced space-time visualization of photospheric data.

Sanderson et. al [SCT<sup>+</sup>10] described a technique that calculates the topology of a toroidal magnetic field using fieldlines, including identifying critical points and other interesting features.

Jähne [Jä12] implemented an adaptive refinement method to locate coronal hole boundaries and visualize the results as vector graphics. This resulted in a more precise than the traditional texture approach which only shows the coarse location of the coronal holes.



## 3. Data Preparation and Access

### 3.1. MAS Simulation Details

The data used in this thesis results from MAS simulation available at [MAS, homepage]. The MAS simulation is based on the data from the Helioseismic and Magnetic Imager (HMI) representing the magnetic field at the photosphere of the Sun and solves the following MHD equations.

$$(3.1) \quad \nabla \times B = \frac{4\pi}{c} J$$

$$(3.2) \quad \nabla \times E = -\frac{1}{c} \frac{\partial B}{\partial t}$$

$$(3.3) \quad E + \frac{1}{c} v \times B = \eta J$$

$$(3.4) \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0$$

$$(3.5) \quad \rho \left( \frac{\partial v}{\partial t} + v \cdot \nabla v \right) = \frac{1}{c} J \times B - \nabla p - \nabla p_w + \rho g + \nabla \cdot (v \rho \nabla v)$$

$$(3.6) \quad \frac{1}{\gamma - 1} \left( \frac{\partial T}{\partial t} + v \cdot \nabla T \right) = -T \nabla \cdot v + S$$

$$(3.7) \quad S = \frac{1}{2kn_e} (-\nabla \cdot q - n_e n_p Q(T) + H + H_d + D)$$

For an explanation of the symbols please see table 3.1. To allow for a simplified solving, the MAS can use the so-called “zero-beta” model. This means it can neglect the pressure  $P$  and gravity  $g$  and assumes a fixed density  $\rho(r)$ . This results in so-called force-free fields.

### 3.2. MAS Data Format

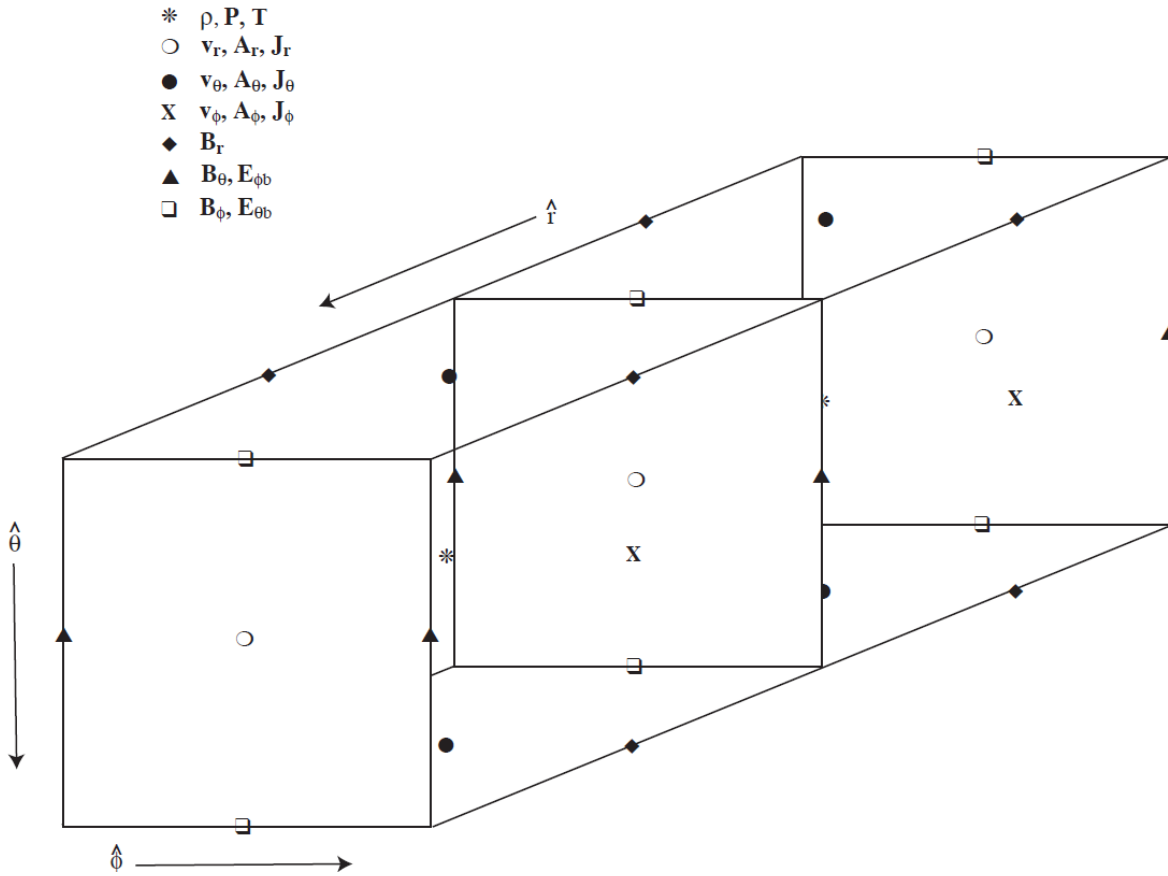
The MAS uses the fourth version of the *hierarchical data file* format (HDF4). Data is stored in a *staggered grid* representation (see section 3.2.2), which is split into separate files based on field type (e.g., magnetic field, velocity field, ...). Each field file is further split into its components (e.g., bp\_xxx.hdf, bt\_xxx.hdf, br\_xxx.hdf for the magnetic field, representing the  $\phi$ ,  $\theta$ , and r components of the magnetic field in spherical coordinates). The data is normalized, with scaling factors written to the respective log file, which requires upscaling the values before they can be used for visualization. The values are stored in *spherical coordinates*, which requires a conversion to *physical coordinates* to obtain the correct visual representation (see section 3.2.4).

#### 3.2.1. Loading HDF Files and Conversion to Structured Grid Representation

Loading is done in two steps: first the HDF file is decoded by the MAS Reader program. The raw data is then passed to a wrapper, HDFExtract. The values are then converted into a structured grid object. Since the original values are given on a staggered grid (see sec. 3.2.2), we interpolate the data on the geometry of a respective structured grid. Data values have to be scaled as described in section 3.2.3 and converted to physical space for rendering.

Variable	Meaning
B	Magnetic Field
J	Current Density
E	Electric Field
$\rho$	Plasma Density
v	Plasma Velocity
p	Plasma Pressure
T	Plasma Temperature
$n_e$	Electron Density
$n_p$	Photon Density
$p_w$	Wave Pressure
g	Gravitational Acceleration
$R_s$	Solar Radius
$\gamma$	Ratio of Specific Heats
$\eta$	Resistivity
$\nu$	Viscosity
A	Vector Potential

**Table 3.1.:** Table of variables used in the MHD equations, as explained in the [MAS, user guide].



**Figure 3.1.:** The Structure of the staggered MAS grid. Source: [MAS, user guide]

### 3.2.2. Staggered Grid

A staggered grid is a structured grid with data values placed not only at cell centers or cell nodes, but also on cell faces and cell edges. This allows for a more compact storage of data but requires interpolation to obtain the original data back, which may introduce noise. For this reason, the data is scaled back *before* the interpolation (so that any introduced noise is not scaled).

The MAS data uses the structure as shown in figure 3.1, variables are defined in table 3.1. The structure shows that only scalar quantities  $\rho$ ,  $P$  and  $T$  have their values at the center of the respective cell. This means that to obtain correct values at the center of the cell for the remaining vector quantities, we have to use different types of interpolation, depending on the position of the data values in the staggered grid. For the velocity  $v$  and current density  $J$  we have to use bilinear interpolation on the faces of the cell for each dimension, while the magnetic field requires linear interpolation on one edge per dimension of the cell.

### 3. Data Preparation and Access

---

Note that neither the flux  $A$  nor the electric field  $E$  are provided on the [MAS, data access] page for download, so they were not used in this thesis.

#### 3.2.3. Data Scaling

As stated in the [MAS, user guide], data values are normalized by the MAS code before they are written to the file, to maximize the numerical precision of the simulation result. This means that values are scaled down to be near 1.0 and require rescaling before they can be used. The scaling factors are slightly different for every simulation run, which is why we have to obtain the respective scaling factors from the corresponding output log. The MAS user guide does provide general scaling factors, which can be seen in table 3.2.

#### 3.2.4. Conversion between Computational and Physical Space

As stated above, the data values are stored in spherical coordinates, separated into their  $\Phi$ ,  $\Theta$ , and radial components. Recombined, they could be represented as a Cartesian grid. We convert the staggered grid representation into a structured spherical grid. The same holds for the data values themselves. In our case, the data values are vectors with 3 components, also stored respective to  $\Phi$ ,  $\Theta$ , and  $r$ .

Parameter	Scaling Factor (MKS)
Length, $l$	$6.96 \times 10^8 \text{ m}$
Time, $t$	$1445.87 \text{ s}$
Velocity, $v$	$481.3711 \text{ km/s}$
Electron density, $n_e$	$10^{14} \text{ m}^{-3}$
Mass density, $\rho$	$1.6724 \times 10^{-13} \text{ kg/m}^3$
Pressure, $p$	$3.875717 \times 10^4 \text{ Pa}$
Temperature, $T$	$2.807067 \times 10^7 \text{ K}$
Magnetic field, $B$	$2.2068908 \times 10^{-4} \text{ T}$
Resistivity, $\eta$	–
Electrical diffusivity, $\eta/\mu_0$	$3.3503427 \times 10^8 \text{ km}^2/\text{s}$

**Table 3.2.:** Table of scaling factors used by MAS, adapted from [MAS, user guide].



The type of conversion needed depends on what we want to convert, e.g., a point, a vector or a matrix. Each type will be explained in the following.

Converting a point:

To convert a point from computational to physical space, we use the following formula:

$$x = r \cdot \sin(\Theta) \cdot \cos(\Phi)$$

$$y = r \cdot \sin(\Theta) \cdot \sin(\Phi)$$

$$z = r \cdot \cos(\Theta)$$

$\Phi$  and  $\Theta$  have to be normalized before inserting them into the formula by dividing them by their respective dimension and multiplying by  $2\pi$  and  $\pi$  respectively. This is done to assert that the values of  $\Phi$  and  $\Theta$  do not exceed  $\pi$  and  $2\pi$  respectively. The reverse conversion is done by applying the following formula:

$$\Phi = \text{atan2}(y, x)$$

$$\Theta = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}}$$

$$r = \frac{z}{\cos(\Theta)}$$

Vector and matrix conversion:

To transform a vector or a matrix from computational to physical space, we have to multiply it by the following Jacobian matrix of the coordinate functions:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -r \cdot \sin(\Theta) \cdot \sin(\Phi) & r \cdot \cos(\Theta) \cdot \cos(\Phi) & \sin(\Theta) \cdot \cos(\Phi) \\ r \cdot \sin(\Theta) \cdot \cos(\Phi) & r \cdot \cos(\Theta) \cdot \sin(\Phi) & \sin(\Theta) \cdot \sin(\Phi) \\ 0 & -r \cdot \sin(\Theta) & \cos(\Theta) \end{pmatrix} \cdot \begin{pmatrix} \Phi_d \\ \Theta_d \\ r_d \end{pmatrix}$$

Note that we have two different sets of  $\Phi$ ,  $\Theta$  and  $r$  in this case: the set without a subscript corresponding to the *point coordinates* the vector is located at, while the set with the subscripts represents the actual *vector components*. The point coordinates have to be normalized as described in the point conversion section above. The reverse transformation is done by multiplying with the inverse matrix.



## 4. Goals

### 4.1. Confirming the Advection Principle

The advection principle describes the phenomenon of charged particles (e.g., electrons) being fixed on magnetic field lines akin to beads on a string, with said magnetic lines and their electrons flowing with the plasma current. The goal here is to visualize to which extent this is true for our MHD data. The motivation behind this goal is to assess whether assuming the magnetic field is frozen with the plasma flow is safe when trying to predict the progression of solar storms and calculating their influence on Earth.

#### 4.1.1. First Approach

Our first approach is to load two sets of HDF data, declaring the first data set as  $t_0 = 0$  and the second one as  $t_1 = 2356585.92s$  (one Carrington rotation) and visualize the material derivative, defined as

$$(4.1) \quad \frac{DB}{Dt} = \frac{B_{t1} - B_{t0}}{t1 - t0} + \nabla(B_{t0}) \cdot v_{t0} = b'$$

where  $B_{t0,t1}$  is the magnetic field and  $v_{t0}$  is the velocity at time step 0 and 1 respectively. The time derivative describes the changes that occurred between the time steps. This part of the equation should be non-zero for some points, reflecting the changes in the magnetic field. Adding the gradient of the magnetic field multiplied by the plasma velocity vector tests whether the plasma flowed in the same direction as the magnetic field or not. Should the final result, the material derivative, be zero, it would mean the advection principle holds.

The visualization is done by creating streamlines of the velocity field, followed by color coding the magnitude of  $b'$  on the resulting streamlines. The streamlines represent the plasma flow, while their color will determine if the advection principle is valid. As an optional setting, a set of arrow glyphs showing the direction of the material derivative can also be displayed.

## 4. Goals

---

### 4.1.2. Second Approach

With the time granularity of the MHD data being rather high (one Carrington rotation), there can be too many changes in the field to be able to see a viable result. We thus follow the alternative of using the MHD equations detailed in section 3.1 to calculate the time derivative. Since there is no scaling factor provided for the current density field, we decided to first verify equation 3.1 before we proceed. Should both sides of the equation be equal, then we assume the scaling of the current density field to be correct.

Once this is verified, we plan to use the equation 3.2 for this purpose. Since the electric field  $E$  is not provided on the data access page, this involves calculating it from equation 3.3. The factor  $\eta$ , which is part of this equation, is not provided neither in the output log nor in the [MAS, User Guide], so we decided to set it to 1.

Once we have the electric field, we can calculate its curl and then proceed by solving the equation 3.2 and obtain our time derivative:

$$(4.2) \quad \frac{\partial B}{\partial t} = (\nabla \times E) \cdot -c$$

Inserting this into our previous material derivative equation 4.1 allows us to load only *one* set of files in contrast to two sets of files (one Carrington rotation apart) from our first approach. The visualization is then again performed by color-coding the magnitude of the material derivation to the streamlines of the velocity field.

## 4.2. Observations from Field Visualizations

To be able to do rapid prototyping of our visualizations, we convert the data from the staggered HDF format to the VTK format. This enables us to load the files in [PV, ParaView], allowing for a simplified inspection. Further visualizations can then be done to improve our understanding of the interrelations between the magnetic, velocity and current density fields.

### 4.2.1. General Field Shape

The goal is to visualize the fields using streamlines to get an idea of how mass-less particles travel inside each field. This should give us an insight on the general structure of the fields. We are especially interested in what shapes the streamlines will take and whether there will be any distinctive regions or features. This should allow us to continue expanding on the knowledge of the interrelations between the magnetic, velocity and the current density fields.

### 4.2.2. Quantification of Current Leaving the Sun

Creating streamlines of the current density field  $J$  showed us that there seems to be little to no current leaving the Sun. To verify this assumption, we calculate an isosurface of  $J$  at isolevel zero. This is done on the radial component of the field by projecting the field in physical space onto a radial unit vector  $(0, 0, r)^T$  so that we can get the magnitude in computational space. Should this yield a closed surface, then there is no current leaving the Sun. Otherwise we should see some structure with holes going down to the photosphere.

### 4.2.3. Interrelations Between the Fields

To continue to improve our knowledge of the inner workings of the Sun, it is important to understand the interrelations between the magnetic, velocity and the current density fields. Our goal here is to look if the fields are influencing each other at all and to attempt to analyze these influences should we find any. We plan to do that by extracting some representative features of the fields and analyzing the interrelations on a local scale, followed by an extension to the global scale.



## 5. Method

This chapter details our methods as well as reasons behind certain design choices. Appendix B provides listings of the corresponding functions in pseudo-code. For a list of existing code used in this thesis, see appendix C.

### 5.1. Converting the HDF into a Structured Grid

We chose to use a curvilinear structured grid because of the spherical geometry information being present unlike in the unstructured case. The GridHolder class allows creating an object using a set of HDF files as input. Since we did not have to load more than three files per set, the class is limited to this value. Calculating the Jacobian matrix as well as the curl of a field is also supported. Since the data has to be converted to physical space for rendering, data is only saved in physical space coordinates to save memory. Calculating the gradient and curl has to be requested on object creation, as we need the data to be in computational space coordinates.

Currently, the class creates an object per grid type (e.g., magnetic field), essentially duplicating the geometry. A rewrite to create one object for all grids loaded is theoretically feasible, but VTK requires a data array to be explicitly defined as “scalars” (with  $n$  number of components), and does not seem to allow flagging another array in the structured grid object as such. Merely flagging a data array as *active* results in an empty output, since VTK does not see any vectors. Another approach would be saving everything in one object and to build a separate one for rendering, containing only the geometry and the required data array flagged as “scalars”. However, this was not attempted because of time constraints.

The loading process itself is quite straightforward. Interpolation is needed in case of a staggered grid because the dimensions of the three grids are different, which is also why we have adjust the dimensions when initializing the grid object parameters. The reason why the calculation of the curl requires the calculation of the gradient matrix is because curl can be easily obtained from it.

### 5.2. Confirming the Advection Principle

#### 5.2.1. First Approach

For the first approach, three grid objects have to be created: two magnetic fields one Carrington rotation apart (*bField\_t0* and *bField\_t1*) and the velocity field (*vField\_t0*). Since the *bField\_t0* object also carries the gradient information, the memory consumption is quite high. This is the reason why objects have to be deleted from memory as soon as possible. The calculation of the material derivative has to be done for each point of the grid, with each calculated value being inserted into the new grid object. This is done in two steps for data access reasons: first the time derivative is calculated, followed by material derivative itself.

#### 5.2.2. Second Approach

For the second approach, we need to create an object for each field present in the MHD equations (see section 3.1). Next step is verifying the equations, failing which will terminate the program as further calculations are not feasible. We deemed a relative rounding error of  $e^{-6}$  to be sufficient to count a verification attempt as positive. Calculating the electric field and its curl, as well as the resulting material derivative is done in multiple steps, for each point and each step, as detailed above.

#### 5.2.3. Visualization

We decided to color-code the material derivative magnitude on the streamlines of the velocity field for both approaches. The reason we chose to generate the streamlines on the velocity field is so that we could follow the plasma flow. Color-coding the magnitude involves finding each point of the streamline object corresponding to a data point in our material derivative object. After this is done, we can then choose further seven neighbor points totaling the eight points needed to perform a trilinear interpolation on the vectors, resulting in a more accurate result vector. This vector is then mapped to the corresponding point of the streamline.

The program also supports generating arrow glyphs of the direction of the material derivative, the direction of the time derivative, as well as a projected gradient vector. This helped us to verify our calculation results were feasible, and were not included in the final visualization image.



### 5.3. Methods Chosen for Field Visualizations

For general field shapes, streamlines seem to be feasible. The advantage over volume rendering or slicing is the ability to specify the density of the seed points. This allows to not only take a look at the structure, but also the flow in the field without having to deal with occlusion issues. Another possible approach is to render ridge or valley surfaces, depending on the type of the field. The different structures revealed by the streamlines can then be further analyzed by rendering a slice at interesting structures. Combining streamlines from one field with slices of another field can show distinctive relations between those fields. Volume rendering might not be suitable because it can be time consuming to get the transfer function right, especially since the amount of data is relatively large, resulting in long refresh times by the rendering plugin.

### 5.4. Ridge and Valley Surfaces

Depending on the form of the field's isosurface, it is feasible to extract ridge or valley surfaces of the field. If the isosurface has a relatively voluminous appearance with many folds, extracting a valley surface can capture the structure of the folds much better. This yields a surface representing the minima of the field. If the isosurface is rather spiky with many leaf-like structures, extracting a ridge surface is feasible. The ridges will correspond to the edges of the "leaves", rendering a representation of the maxima of the scalar field.

Both approaches are calculated in the same way, the only difference is that for the valley surface the scalar field value has to be negated before passing it to the calculation routine, i.e. valleys are ridges of the negated scalar field. The calculation routine itself first calculates the Hessian of the scalar field value. The eigenvectors  $e_i$  corresponding to the largest eigenvalues of the Hessian, which point along the ridge. The eigenvectors  $e_j$  corresponding to the smallest eigenvalues are in turn pointing perpendicular to the ridge. The first criterion for a local maximum is that the directional derivatives in  $e_j$  should be zero. The second criterion required is that the second directional derivative in  $e_j$  is negative. For more information on ridge extraction, refer to [Ebe96].



## 6. Results

### 6.1. Confirming the Advection Principle

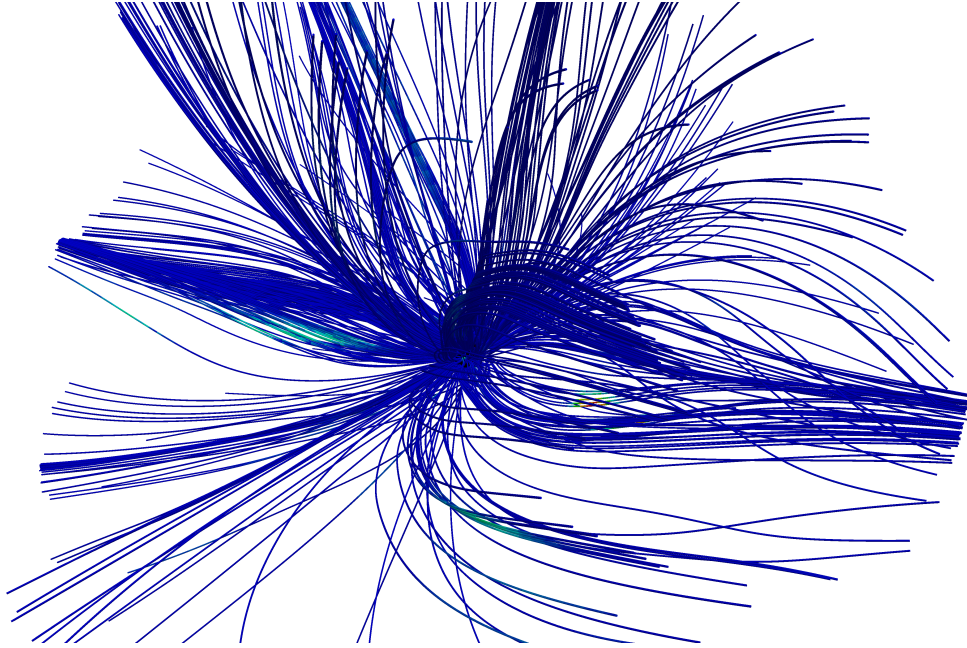
#### 6.1.1. First Approach

Our first approach showed that both the time and material derivatives were non-zero, as visible in figure 6.1. There are several spots with high magnitude, which means that the advection principle is not valid everywhere in the field, but this observation still requires further validation. If this is true, however, then it would mean that making use of the advection principle and only looking at magnetic field changes is not sufficient to predict the progression of a solar storm. Validating this would go beyond the scope of this thesis because the MAS data is not present in an adequate time granularity. The time granularity poses a problem insofar that there could have been too many changes in the fields in the period of time, negatively impacting the precision of the calculations.

#### 6.1.2. Second Approach

The second approach could not be implemented. This had multiple reasons: one of them was because after attempting to verify equation 3.1, it turned out that the relative error was larger than the value itself, for every point value in the field. This means that our scaling of the current density field was wrong, but since it turned out there was no scaling factor for  $J$  stated in the manual or the output log, we assumed no scaling was needed. Apparently, this was not the case. Several attempts at scaling the values manually were done, but none yielded usable results.

Coupled with the fact that the value of  $\eta$  was neither provided in the output file nor in the manual and simply ignoring it is bound to distort the results, this approach was abandoned. Continuing to work on it would require to run a custom MAS simulation explicitly to obtain the value of  $\eta$  as well as the proper scaling factor for  $J$ . Hence, this approach is subject to future work.



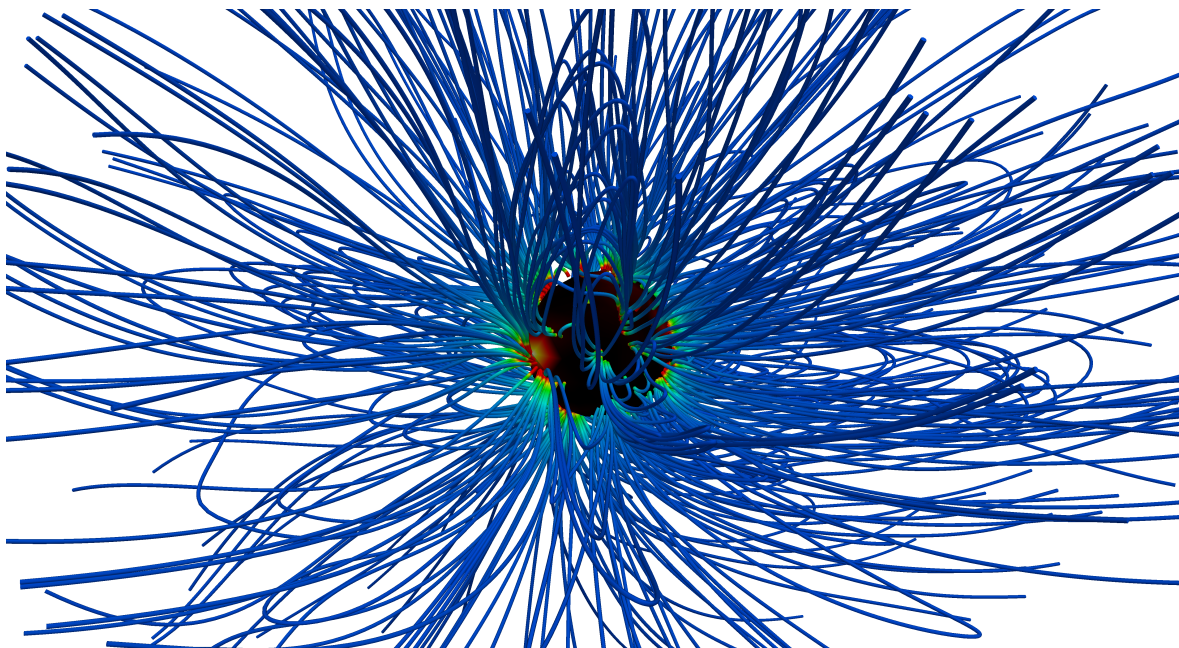
**Figure 6.1.:** Material derivation magnitude mapped to plasma velocity streamlines. Blue stands for low values, red for high values.

## 6.2. Observations from Field Visualizations

### 6.2.1. General Field Shape.

The magnetic field consists of arcs starting somewhere on the Sun's surface (see fig.6.2). They either end somewhere relatively near the starting points back at the photosphere or escape to the outer space. In case of forming an arc, the ending point seems to never be more than approximately a third of the Sun's outline away from the starting point. All the starting points seem to be forming several "hot spots", with few to no lines starting in the areas between those. These are the so-called active regions of the Sun. Another fact worth noting is that the lines are rather smooth, without any sudden changes in direction. The values are high near the Sun surface but rapidly decrease proportionally to the distance from the surface, with weakest values in the outer photosphere. Looking at the magnitude, the difference between the strongest and the weakest magnitude is  $10^6$ .

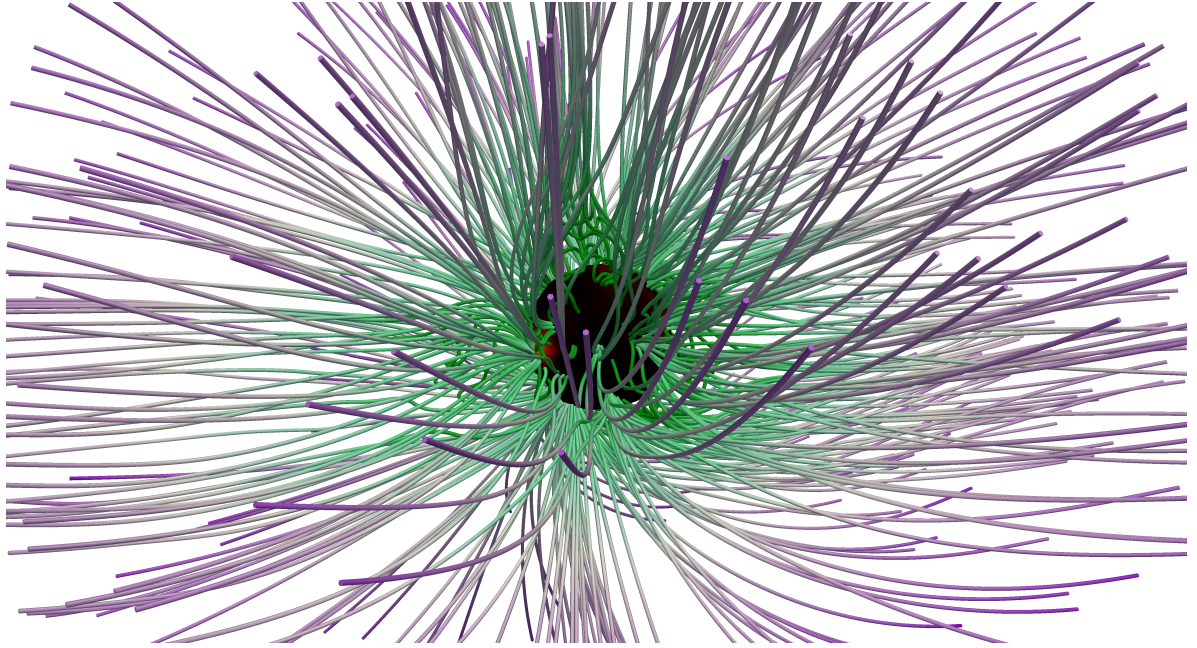
The velocity field seems to be structured similarly to the magnetic field, with the same "hot spots" present in the data (see fig. 6.3). The biggest difference is the ratio of arcs versus escaping lines: for the velocity field, nearly all lines escape towards the outer space. They seem to travel mostly in a smooth arc, however there are some regions that show some kind of displacement taking place. This results in an uneven distribution of streamlines across the coronal sphere even though the seed point distribution was chosen evenly across the Sun surface.



**Figure 6.2.:** The shape of the magnetic field.

We found this displacement to be interesting and looked at what could have caused it in the next section. There are also some lines traveling between the “hotspots”, but those are very few. They seem to have an interesting feature however, namely the many directional changes compared to their smooth magnetic counterparts. Value distribution is inverse proportional to the distance to the Sun’s surface: the further the plasma travels, the higher the speed. The magnitude changes are very large compared to the magnetic field, ending up growing to a factor of  $10^{10}$ . This could be attributed to the fact that the pressure at the Sun’s surface is larger, thus pushing the ascending plasma, resulting in a siphon effect. This could also explain why there are some kind of “hotspots” where more lines start.

The current density field, in contrast to the magnetic and the velocity fields, consists largely of swirls (see left part of fig. 6.4). This is due to the fact that visualizing the particle flow via streamlines is the same as visualizing the rotational flow caused by the Lorentz-force. There seem to be two types of swirls: Ones that diverge from the center (tornadoes) as they rise from the Sun’s surface, and ones that converge into a line (spires). One can also easily see the same “hot spots” as before are present here, as well. The values seem to depend on the distance from the Sun’s surface, with noticeably less current (black) flowing in the upper coronasphere and the inner regions of a swirl. Highest magnitude (white) can be observed nearest to the Sun’s surface and the regions between multiple swirls. The range of the magnitude spans  $10^9$ .



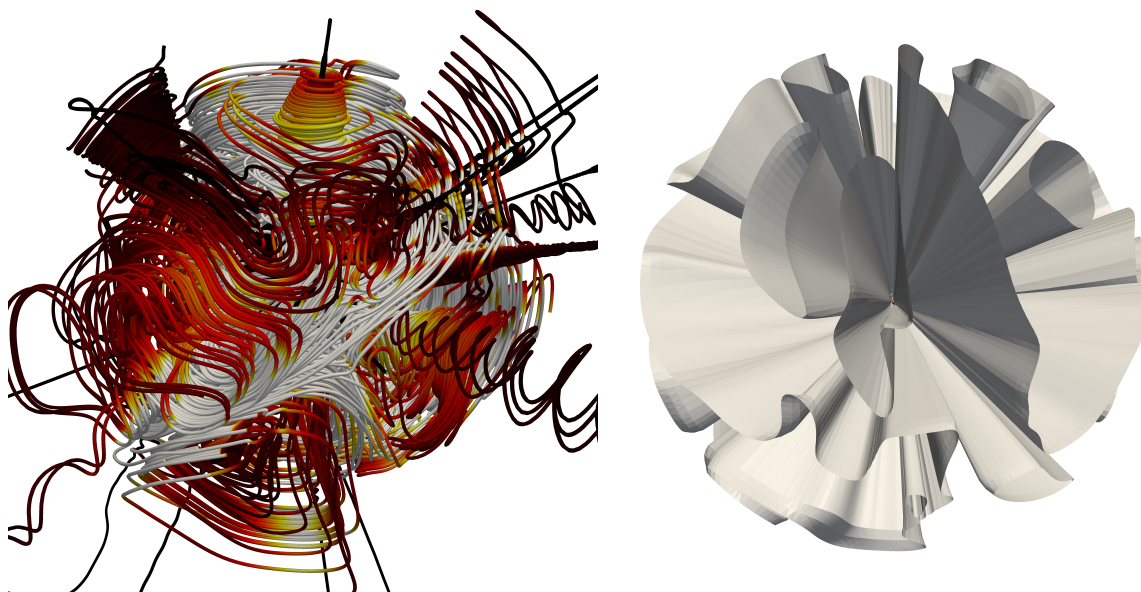
**Figure 6.3.:** The shape of the plasma velocity field.

There seems to be little current leaving the Sun compared to the quantities flowing closer to the Sun's surface. This becomes obvious by looking at the early streamline termination compared to the two other fields - the Sun's photosphere on the left of figure 6.4 is only barely covered by the streamlines and is twice the size compared to figure 6.2 and 6.3. We tried increasing the step count as well as step sizes, but the lines were terminated at the same positions every time.

An isosurface of the radial component of current density field (at level 0) meant to verify this fact can be seen on the right of fig. 6.4. While a certain structure where no current is present can be seen, since it is not a closed surface, our assumption that no current is leaving the Sun was wrong. We assume that the streamlines terminated early because of a large drop in current density triggered the cancellation, thus resulting in shorter streamlines.

### 6.2.2. Interrelation Between the Fields.

Looking at fig. 6.5 shows that the magnetic field (blue) forms arcs between the centers of two swirls of the current density field (flame hues), with only some of the flow moving towards the outer space. Comparing how the plasma velocity (violet) behaves in the same region shows that it roughly follows the flow of the magnetic field, but instead of forming a smooth arc between the two swirl centers, the lines try to escape to the outer space as fast as possible. This results in multiple direction changes in the area between the two swirls.

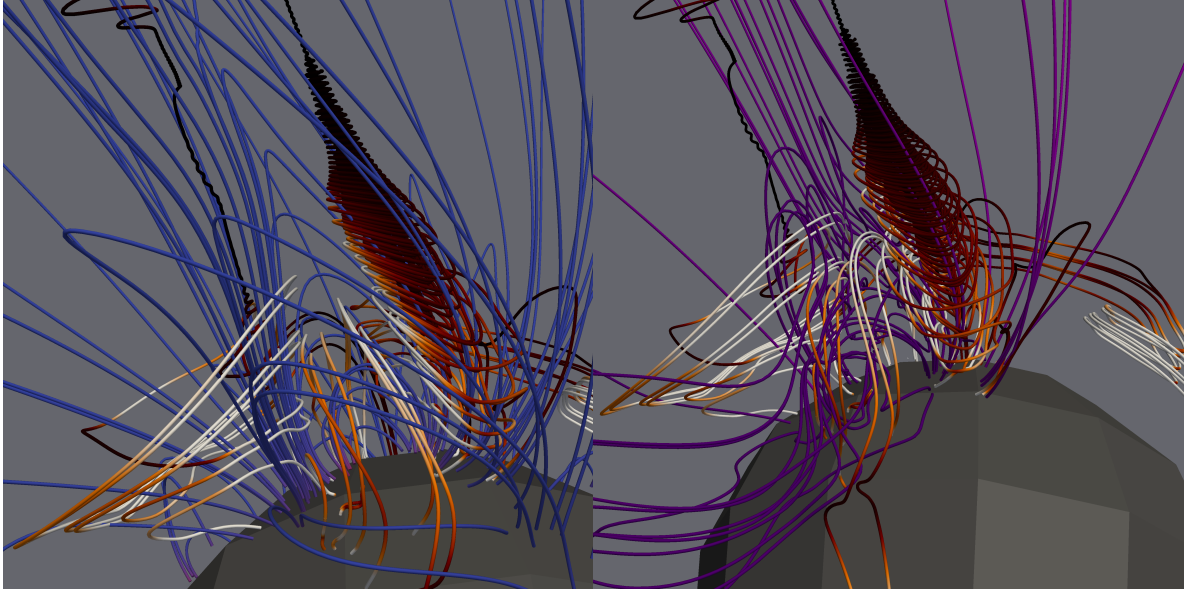


**Figure 6.4.:** The shape of the current density field (left) and the isosurface at magnitude zero (right).

Since there seem to be some kind of “hotspots” or “pockets” where the quantities are much higher than the rest of the field, we attempted to pick out some isolated swirls and performed a volume rendering of the magnetic field at the cross-section of the swirl. The result, however, did not reveal anything significant. For that reason we chose to render more swirls of the current density field, coupled with multiple isosurfaces at middle levels of the magnitude of the magnetic field. To be able to see swirls better, we also added a rendering of vortex core lines.

The result can be seen on the left of figure 6.6. As we suspected, there seem to be areas with stronger magnetic field (see isosurface of magnetic magnitude) inside the regions formed by the swirls of current density (flame hues). The majority of current density streamlines seem to be “flowing” through the valley surfaces in the magnetic field. There are some exceptions, however, where swirls were not following the valleys of the magnetic magnitude. We assume this is because a “pocket” can be composed of more than one swirl. Looking at the vortex core lines (green), there are multiple core lines visible in the bigger “pockets”, which further supports our assumption.

To get a better look at the structure of the magnetic field around the swirls, we put a slice through the volume, as can be seen in figure 6.7. One can see how the swirls go through the weak (dark blue) magnetic field regions and encompass a stronger magnetic field inside (red to cyan). The “pockets” turn out to actually be “pillars” instead. This would mean, the swirls of current density “carry” the magnetic field towards the outer space.



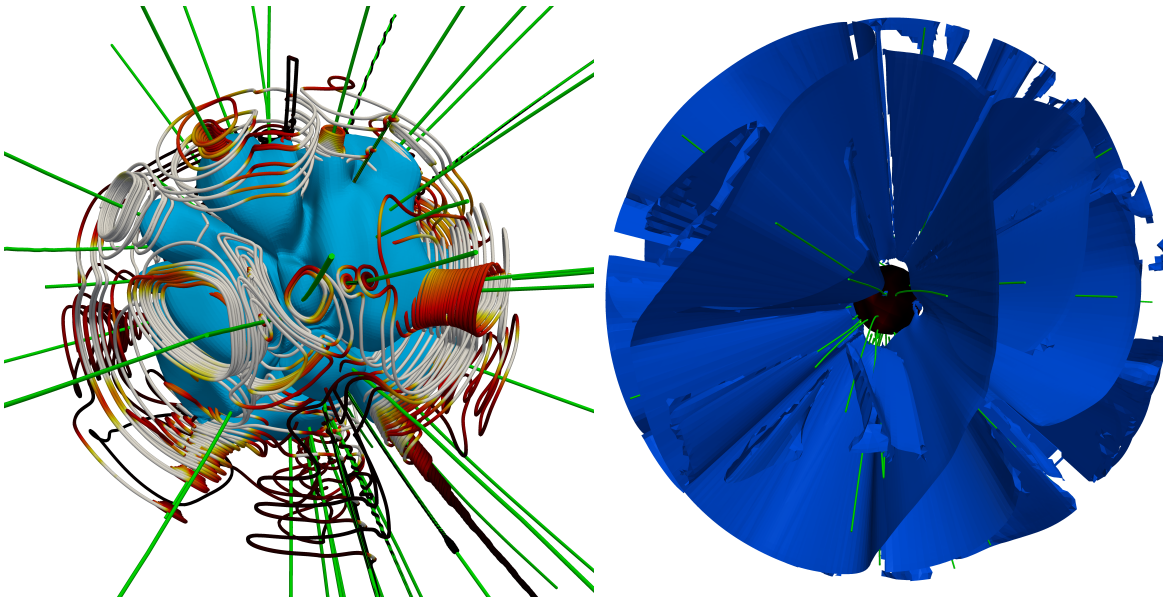
**Figure 6.5.:** Interrelations between the current density (flame hues), the magnetic (blue hues) and the plasma velocity fields (violet).

The “pillars” theory could be proven by rendering a valley surface and verifying that no swirls are cut by the resulting structure. For example, should there be a vortex core line piercing a pillar wall, then the theory would be false. For this purpose, we extracted valley surfaces, limiting the magnitude to a maximum of  $10^{-5}$ . Looking at the resulting image (fig. 6.6) shows that the core lines do *not* pierce any walls. In cases where a vortex is getting too close to a wall, the vortex seems to dissipate. It seems as if those vortex core lines belong to spire-type vortices, which we will be verifying in section 6.3.

To further support our theory, we rendered a ridge surface of the current density field magnitude and overlapped it with the valley surface of the magnetic field, limiting it to a minimum of 500. The result can be seen on the left of figure 6.8. As we expected, the ridge surface representing the peak values of the current density field (orange) is quite similar to the valley surface representing the pillar sides. This not only supports our assumption that the pillars exist, but also shows that some of the pillars really are composed of multiple swirls.

Another interesting fact is that the similarity between the valley surfaces of the magnetic field, the ridge surfaces of the current density field and the isosurface we generated to verify the current leaving the Sun (see fig. 6.8). The isosurface (orange) seems to be positioned at a certain offset relative to the two other surfaces (blue). This can be explained if we assume the current density to be small towards the middle of the swirls. Looking back at the shape of the current density field, this seems to be correct.





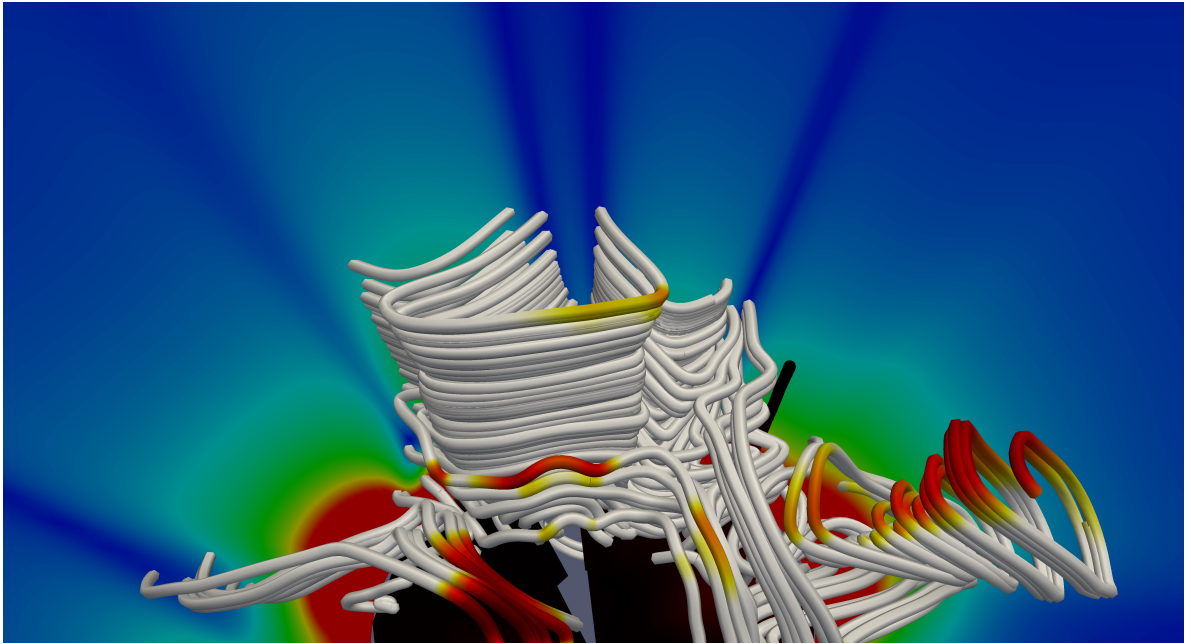
**Figure 6.6.:** Isosurface of the magnetic field (blue) with swirls (core lines in green) of current density (left) and the valley surface of the magnetic field magnitude (right).

There are a few additional planes belonging to the isosurface that are not present in the pillar structure. Those planes cut through some of the swirls, which we assume is because the combination of the different particle flows (for example, swirls rotating in opposite directions) causes them to cancel out. This gives us an additional insight on regions in the field where no current is flowing at all, as they seem to be existing exactly between any neighboring swirls. The exception to that rule can be observed in pillars that are formed by more than one swirl, the reason probably being the fact that multiple swirls forming a pillar have particle flows that travel between swirls, thus not counting as zero.

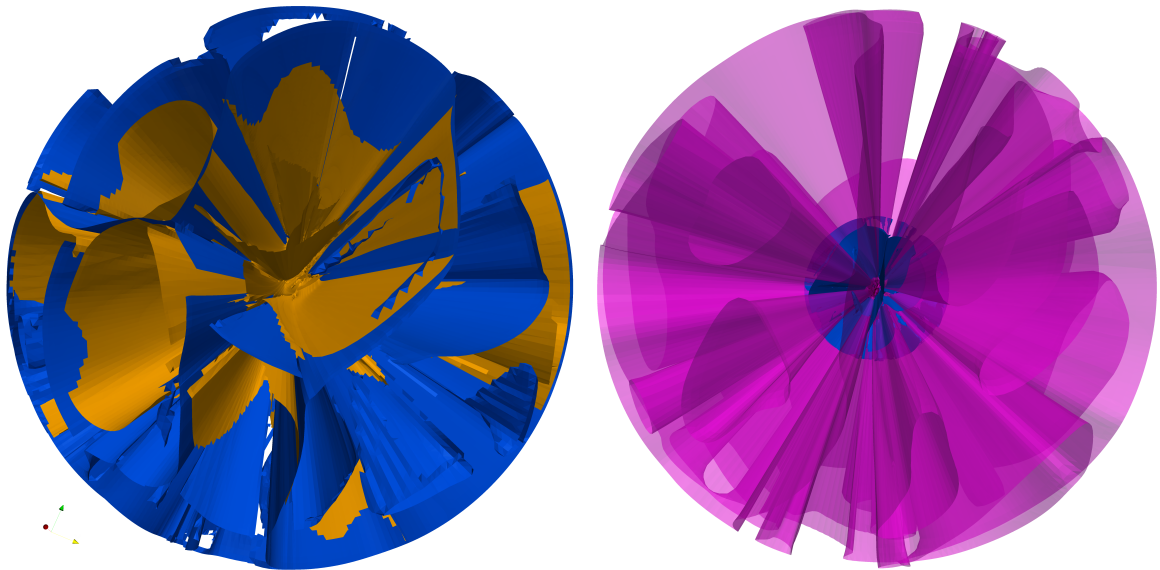
### 6.3. Behavior of Current Density Field Swirls

Now that we established that the current density field ( $j\_field$ ) swirls are in relation with the magnetic field ( $b\_field$ ), we want to look at how different types of swirls correspond to the  $b\_field$ , as well as the plasma velocity ( $v\_field$ ). For this purpose we picked some swirls representative for every type of swirls present in the field.

As previously mentioned in section 6.2.1, there are two types of swirls, “tornadoes” and “spires”. Tornadoes are very common, with sizes varying from relatively small to very large. Spires are rare and do not seem to form their own pillars, usually appearing at the edge of a pillar formed by one or more tornadoes. After generating the pillar structure mentioned above, we saw several vortex core lines ending after getting too close to a pillar wall. At first we assumed those belonged to the spire swirls being pushed aside by their tornado neighbours.



**Figure 6.7.:** Swirls of the current density flowing through a slice of the magnetic field.



**Figure 6.8.:** Result of overlapping the pillar structure (blue) with the ridge surface of the current density field (orange) (left) and isosurface of current density magnitude at level zero (magenta) (right).

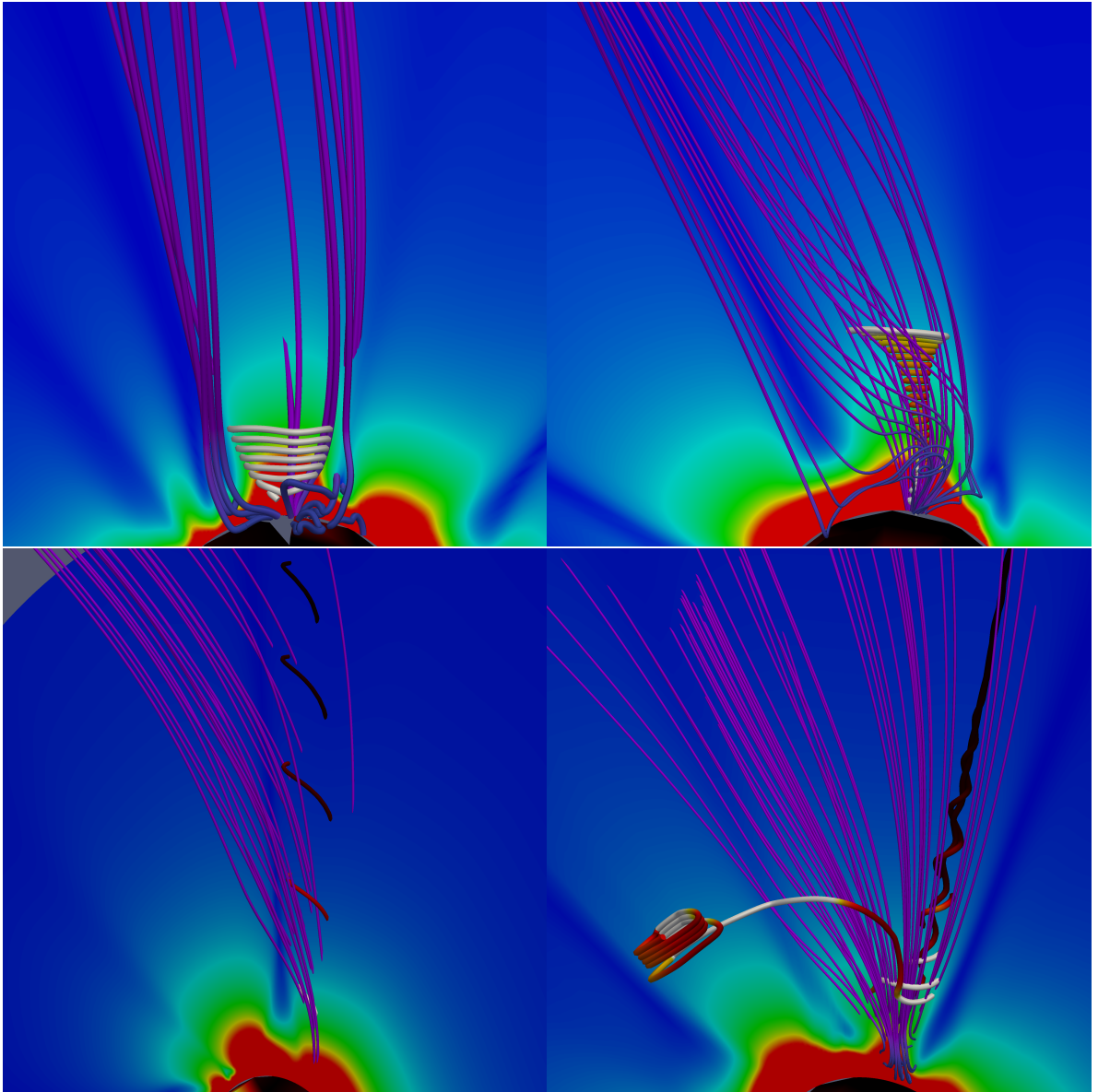
This would explain why the radius of one type of swirls increased while the other one decreased. However, this turned out to not always be the case. Some pillars contained a spire in the middle, with only one more tornado-type swirl around it. At the point where the spire gradually turn into a line, a tornado-type swirl formed from a flow crossing over from a neighboring pillar.

After further analysis of the field, there seem to also be two subtypes for each type, varying in the speed of radial changes as well as starting sizes. The radial change rate did not seem to have affected the  $b_{\text{field}}$  in any noticeable way, so we looked at the  $v_{\text{field}}$  instead.

Looking at the tornado-type swirls with very slow increase in radius and a comparatively large starting radius (see fig. 6.9, upper left), we can see that the plasma flow is rather smooth, flowing towards the outer space. Comparing the plasma flow of a similar swirl with higher radial increase rate (upper right), there suddenly seem to be slight distortions in the plasma flow, with displacement in the direction opposite to the swirl rotation.

Looking at a spire-type swirl with a slow radial decrease rate (lower left), the same displacement can be seen. It is however more prominent, even more so around a spire with fast radial decrease rate which end up converging to a line (lower right). There are far more directional changes of the plasma flow at the base of the spire, and quite a strong displacement as the plasma flows towards the outer space.

Summing up, there seem to be two aspects affecting the flow of the plasma field. The first one is the type of the swirl, with spires showing most prominent influence on the flow. The second aspect, radial in-/decrease rate, has a weaker effect on the plasma flow. The displacement causes the particles to shift outside pillar boundaries, resulting in an uneven distribution of streamlines in the coronal sphere.



**Figure 6.9.:** The two types of swirls with different radial increase rate. Above: tornadoes,  
Below: spires

## 7. Discussion

With this thesis, we prepared the foundations to start visualizing solar MHD. During the process of working on the thesis we encountered several problems. We discuss those in this section.

### 7.1. Data Problems

#### 7.1.1. Data Format

The hdf4 format is outdated. Its new version, hdf5, is not backwards compatible with hdf4, so making use of VTK's hdf5 parser was not possible. Documentation on handling the format is present, but only for fortran and is scattered around the web. This resulted in higher development time needed to implement an extraction tool. It would be best to migrate MAS to hdf5 in the future, especially since hdf4 is size-limited to 2GB.

#### 7.1.2. Data Granularity

Another problem is the time granularity of the data available at the [MAS, data access site], which offers one data set per Carrington rotation. This means we have a  $\Delta t$  of 27.2753 days. For some observations, for example our material derivation calculation, there tend to be too many changes in the fields to allow viable interpretations of the visualization results. We propose increasing the update interval to three days to allow for more precise observations. However, running these MAS simulations was beyond the scope of this thesis.

#### 7.1.3. Output Files

Some fields are named in the MAS documentation and are used in the equations, but are not offered for download on the [MAS, data access page]. One example is the electric field  $E$ .

### 7.2. Performance Issues

There are several performance issues that need solving for a time-efficient MHD visualization environment.

#### 7.2.1. Memory Usage

Even though our program is freeing memory as soon as it is no longer needed, we still need over 4GB of RAM to simply write the complete dataset (one Carrington rotation) to the disk. This can be mitigated by saving the grid geometry only once. This is possible because the grid geometry is the same for all grids, since all the data comes from one simulation run. Of course, if multiple data sets are used, multiple geometries are possible. Whenever a structured grid object is needed, it is simply built on demand. Implementing this functionality was beyond the scope of this thesis because of time constraints.

#### 7.2.2. Multi-Threading

VTK implements a visualization pipeline which is updated on demand. This pipeline is *not thread safe*, severely limiting performance on modern multi core systems. For example, there are functions to request a certain array element in VTK, that return a pointer pointing to this element. This pointer changes once another call is made to that function. While we can deep copy a pointer to avoid this, if two threads happen to call such a function at the same time, there is a high chance of encountering a segmentation fault. In fact, we have written a function that calculated the streamlines on multiple cores and it worked most of the time, crashing with a segmentation fault with same parameters it was running fine in all previous runs. See appendix A for instructions on how to enable this functionality for testing purposes. Another thinkable scenario is one thread requesting an update event, which will update *all of the data*, possibly changing some data another thread uses for its calculations. A solution would be to either simply export the grids to ParaView, or discard VTK completely and use own code instead. In the latter case we suggest keeping the ParaView export functionality. Keeping the rendering functionality is not advisable, however. It may be convenient to use own code, then convert the final object to VTK for rendering, but in practice we found out it is sometimes difficult to assemble correct VTK objects. For example, using an external Marching Cube class to calculate the isosurface for the current density, we encountered a problem that while the surface was calculated correctly and the polydata object was constructed following VTK documentation, the surface could only be properly rendered in its wireframe representation.

## 7.3. Documentation

Some information required to be able to work with all the MHD equations present is missing. For example, there is no information on which value  $\eta$  actually has. Another example is the missing scaling factor of the current density field  $J$ . This holds for both the [MAS, user guide] and the actual simulation output logfile.

On the [VTK] side of things, while every function is properly documented and there are many examples, in some cases the documentation is unclear about how to proceed. For example, even though the example for building a polydata object works, we ran into problems building one that contained a custom set of triangles.





## 8. Conclusion

In this thesis, we presented first approaches to the visualization of solar magnetohydrodynamics. Because of the lack of staggered grid support in VTK, we first provided a technique for appropriate conversion of the [MAS] data to a structured grid representation. Subsequently, several approaches for visualizing the advection property of the magnetic field in the coronal plasma were presented. We also provided visualizations of the current density field with respect to both the magnetic and plasma velocity fields. Interestingly, both provided comparable results.

We then proceeded with several visualizations of the magnetic, plasma velocity and the current density fields, with the goal to understand the interrelations between these fields, thus helping to understand the inner workings of the Sun. We looked at the general shapes, followed by extraction of distinctive features. The analysis of these features resulted in multiple insights. First, we validated the existence of so-called pillars in the magnetic field. These are regions where the magnetic field is considerably stronger than the surrounding regions that rise to the outer coronasphere. Next, we confirmed that the current density field is forming swirls around these pillars, seemingly carrying the magnetic field towards the outer space.

We further found out some of the pillars are formed from multiple swirls, as well as the fact that the current density is weaker towards the center of a swirl. There also seem to be regions where the current is weaker between the swirls. Multiple swirl types were discovered, their effects on the plasma flow depending on their form as well as rate of radial change.

### Future Work

The visualization of MHD data is a upcoming field of research and the presented approaches represent a point to start from. As future work, one could continue analyzing the interrelations between the current density and for example the temperature, or pressure. The derivation of specific MHD features is another desirable result which, however, requires broad investigations by techniques such as those presented in this thesis. Optimization of the software developed in this thesis, as detailed in the discussion chapter (7), as well as more information on variables used by the MAS simulation are other feasible directions for future work.



# A. Manual

## A.1. Dependencies and Setup

This section explains how to set up the environment to be able to compile and run the mVisualizer program.

### A.1.1. Dependencies

The following packages are currently required:

- *libhdf4-dev* for the ReadMAS program
- *libgl1-mesa-dev*, *libxt-dev* and *g++* for VTK
- *cmake* and *ccmake* for compiling VTK and the mVisualizer program

### A.1.2. Setting up VTK

VTK has to be manually compiled for the program to work. The reason behind this is that the precompiled package only includes the libraries, but not the include files. Download the sources from [VTK], extract the package and run *ccmake* in that directory. Press *c* to run the initial configuration. When done, a list of adjustable variables will be displayed.

The only important variable for us is *BUILD\_SHARED\_LIBS*, which we have to set to “ON”. *VTK\_DATA\_ROOT* as well as *VTK\_LARGE\_DATA\_ROOT* not being present is normal and not relevant for the program. Press *c* again, followed by *g* to generate the build files.

Now run *make && make install*.

### A.1.3. Setting up ReadMAS and HDFExtract

Once the *libhdf4-dev* package is installed, navigate to the */ReadMAS/* directory and run *ccmake*. Should it fail to find *hdf4*, edit lines 5 and 6 in *CMakeLists.txt* to match your system configuration. Additionally, lines 28 and 29 may have to be edited, replacing “df” with “hdf”. Generate the build files and run *make*. Repeat the same procedure for the */jaehnesn/HDFExtract* program, editing lines 9, 10 and 35 of the corresponding *CMakeLists.txt*.

### A.2. Program Usage

This section describes every switch the program supports and provides insight on what the program is capable of.

#### A.2.1. Shell Selection

mVisualizer supports slicing with respect to radial “shells” of the data, controlled by the `--min=j` and `--max=k` switches. Make sure  $j \leq k$  is honored, otherwise the program might behave erroneously. When not specified, the whole spherical grid will be loaded.

#### A.2.2. Modes

The mVisualizer program supports multiple operation modes, each requiring a certain number of hdf input files.

The `--sphere` mode renders a sphere at layer specified with the `--max` switch. When called with this switch, the program will expect three hdf files representing the phi, theta and radius components of the magnetic field (bField). If the optional `--shrink` switch is specified, the program will render all layers from `--min` to `--max`.

The `--streamlines` mode renders streamlines using the deepest layer `--min` as seeds. You can control the number of seeds by setting the `--seedratio=x` switch, which chooses every  $x$ th point as a seed. Like the `--sphere` switch above, the program expects three hdf input files of the magnetic field, one for each component of the field.

The `--tderiv` mode will attempt to compute the time derivative  $b'$  using nine input files (three components of bField\_t0, bField\_t1 and vField\_t0 respectively). The output will be rendered as streamlines of the vField\_t0. Seed ratio can be specified with the `--seedratio=x` switch. There is also an additional switch `--showarrows`, which will render arrow glyphs visualizing values on the streamlines.

The `--isosurface` mode will create an isosurface at a certain level passed by the `--level=x` switch. Currently, it expects three component files of the jField as input.

The `--toPV` mode does not render any output, it simply writes the grid files to the disk as \*.vtk files. This allows to load the grids into any program supporting this format, e.g., ParaView for further analysis. The switch requires all twelve hdf files as input and produces following \*.vtk files:

- `bField_t0.vtk`, containing the data values, the gradient and the curl at time step 0.
- `bField_t0_seeds.vtk`, the innermost layer of the magnetic field at time step 0.
- `bField_t1.vtk`, containing the data values of the magnetic field at time step 1.
- `bField_t1_seeds.vtk`, the innermost layer of the magnetic field at time step 1.
- `vField_t0.vtk`, containing the data values of the velocity field.
- `vField_t0_seeds.vtk`, the innermost layer of the velocity field.
- `jField_t0.vtk`, containing the data values of the current density field.
- `jField_t0_seeds.vtk`, the innermost layer of the current density field.
- `tField_t0.vtk`, containing the data values of the temperature field.
- `tField_t0_seeds.vtk`, the innermost layer of the temperature field.
- `dField_t0.vtk`, containing the data values of the density field.
- `dField_t0_seeds.vtk`, the innermost layer of the density field.
- `pField_t0.vtk`, containing the data values of the pressure field.
- `pField_t0_seeds.vtk`, the innermost layer of the pressure field.
- `dBdt.vtk`, containing the time derivative of the magnetic field.
- `gradXB.vtk`, containing the Jacobian of the magnetic field multiplied by the normalized data value at each point of the magnetic field.

### A.2.3. Hidden Switches

Some switches are not mentioned when calling the program with the `--help` parameter. The reason behind this is, they are only partially functional.

There is an optional switch, `--threads=x` which is used to specify how many cores the streamline mode will use. As mentioned in the discussion chapter (7), multithreaded code runs instable due to current VTK version not being thread-safe.

The `--deriv2` mode represents an alternative computation approach of the material derivative  $b'$ , which is only incompletely implemented because the equation could not be verified with the current version of the manual and the output log files.. You can run this mode to see the relative error of the equation, as mentioned in the discussion chapter (7).

### A.2.4. Rendering Window Navigation

If a mode that has renderable output has been chosen, the resulting render window can be navigated with the mouse. Hold the left mouse button (*LMB*) and move the mouse to rotate the view. Hold the middle mouse button (*MMB*) to pan the view. Switching between surface and wireframe representations is possible by pressing *S* and *W* respectively.

## B. Implementation

This appendix contains code listings for important algorithms in pseudo code. The implementation in C++ can be found on the accompanying CD. For a list of existing code used in this thesis, see appendix C.

## B. Implementation

---

---

### Listing B.1 Loading HDF to a *vtkStructuredGrid*

---

```
GridHolder::GridHolder(const vector<HDFData*>& data,
                      std::string name,
                      int startRad,
                      int endRad,
                      bool calcGradient,
                      bool calcCurl) : name(name)

bool staggeredGrid = false;
for each HDFFile
{
    getDataDimensions();
}
if (dimMismatch == true)
{
    staggeredGrid = true;
}
if (staggeredGrid == true)
{
    adjustDimensions();
    saveDimensionsToObject();
    initializeGridObjectValues();
    for(rad = startRad; rad <= endRad; ++rad)
    {
        for(theta = 0; theta < maxTheta; ++theta)
        {
            for(phi = 0; phi < maxPhi; ++phi)
            {
                updateMinMaxValues();
                savePointData(pointInPhysicalCoordinates);
                saveVectorDataInPhysical(vectorInterpolatedByType);
                saveVectorDataInComputational(vectorInterpolatedByType);
                scaleSavedVectorData();
            }
        }
    }
    setGridPoints(points);
    setGridVectors(vectorsInPhysical);
}
else
{
    // as above, but no interpolation and dimension adjustments needed
}
setGridExtent(minVal, maxVal);
if (calcGradient == true || calcCurl == true)
{
    calculateGradient();
}
if (calcCurl == true)
{
    calculateCurl();
}
closeHDFFiles();
```

---



---

**Listing B.2** Calculating the Material Derivation - First Approach (Includes Visualization)

---

```
getGridDimensions(bField_t0);
initializeTimeDerivGridObjectValues();
initializeMatDerivGridObjectValues();
for each vector in bField_t0
{
    insertNewValue((tmpVectorB_t1 - tmpVectorB_t0) / 2356585.92, timeDeriv)
}
for each vector in vField_t0
{
    insertNewValue(timeDeriv + bField_t0_gradient * vFieldVector, matDeriv);
}
dirLines = calculateStreamlines(vGrid_t0);
for each point in dirLines
{
    index = getPointIndex(point);
    interpolateTrilinear(point1-8);
}
magLines = magnitude(dirLines);
render(magLines);
```

---

---

**Listing B.3** Calculating the Material Derivation - Second Approach

---

```
bool result = true;
getGridDimensions();
for each point in grid
{
    if (result == true)
    {
        result = verify(curlB = 4 * PI / LIGHT_SPEED * jField);
    }
    else
    {
        return(EXIT_FAILURE);
    }
}
for each point in grid
{
    calculateEField(1 / LIGHT_SPEED * (vField x bField) = -eField);
    calculateCurl(eField);
    calculateMaterialDerivation();
}
```

---



## C. Acknowledgements

This appendix handles acknowledgements for foreign code used in this thesis.

### C.1. MAS Reader

The MAS Reader is written by Thomas Müller.

### C.2. HDFExtract

The HDFExtract library is written by Sebastian Jähne.

### C.3. Vector and Matrix Classes

Both the vector and the matrix classes are written by Sebastian Jähne.

### C.4. Misc Code Snippets

All code in `/jaehnesn/` folder is written by Sebastian Jähne and is included for the HDFExtract, Vector and Matrix classes to work properly.



# Bibliography

- [Ebe96] D. Eberly. *Ridges in Image and Data Analysis*. Computational Imaging and Vision. Kluwer Academic Publishers, 1996. (Cited on page 25)
- [GKP10] J. P. H. Goedbloed, R. Keppens, S. Poedts. *Advanced Magnetohydrodynamics*. 2010. (Cited on page 11)
- [GP08] J. P. H. Goedbloed, S. Poedts. *Principles of Magnetohydrodynamics*. 2008. (Cited on page 11)
- [Jä12] S. Jähne. Adaptive Visualisierung Coronaler Löcher, 2012. Studienarbeit. (Cited on page 11)
- [MAS] MAS Homepage, <http://www.predsci.com/hmi/home.php>. (Cited on pages 7, 9, 13, 14, 15, 16, 20, 37, 39 and 41)
- [MSM<sup>+</sup>12] G. M. Machado, F. Sadlo, T. Müller, D. Müller, T. Ertl. Visualizing Solar Dynamics Data. *Vision, Modeling, and Visualization*, 2012. (Cited on page 11)
- [PV] ParaView Homepage, <http://www.paraview.org/>. (Cited on page 20)
- [RLM<sup>+</sup>06] P. Riley, J. A. Linker, Z. Mikic, , R. Lionello, S. A. Ledvina, J. G. Luhmann. A Comparison Between Global Solar Magnetohydrodynamic and Potential Field Source Surface Model Results. *The Astrophysical Journal*, 2006. (Cited on page 11)
- [SCT<sup>+</sup>10] A. R. Sanderson, G. Chen, X. Tricoche, D. Pugmire, S. Kruger, J. Breslau. Analysis of Recurrent Patterns in Toroidal Magnetic Fields. *IEEE Transactions on Visualization and Computer Graphics* 16, 2010. (Cited on page 11)
- [VTK] VTK download page, <http://www.vtk.org/VTK/resources/software.html>. (Cited on pages 9, 39 and 43)

All links were last followed on November 07, 2012.



## **Declaration**

All the work contained within this thesis, except where otherwise acknowledged, was solely the effort of the author. At no stage was any collaboration entered into with any other party.

---

(Ivan Pilipenko)