Statistical Properties of the Solar Active Region Corona:
A Data-Driven Analysis Applied to a Model for Particle Acceleration

A Thesis
Submitted in Fulfillment of the Master’s Degree in Computational Physics at The Aristotle University of Thessaloniki

Anna Toutountzi
March, 2015

Department of Physics, University of Thessaloniki, 54 124, Greece
To my mother,
every time...
Anna Toutountzi
Abstract

The full interpretation of the electromagnetic spectrum of the solar corona has been puzzling the world of Astronomy for over 70 years. One of the basic questions posed by this rich in characteristic features spectrum, relates to the particle acceleration mechanisms taking place in the lower solar atmosphere, and is therefore directly linked to what is known as the “coronal heating problem”, as well as the impulsive particle acceleration responsible for the hard X-ray part of the spectrum. One of the most promising mechanisms suggested, bases its arguments on the dynamically evolving intermittent structure of the magnetic instabilities residing in both “quiescent” and “active” coronal regions, causing the formation of strong, permeating electric fields. In the first part of this study, these magnetic structures are examined by means of their data-derived statistical properties, using, among several data mining techniques, a variety of methods developed for magnetograph data processing. Having obtained information relating to the probability distribution functions followed by certain spatial and dynamical quantities characterizing these magnetic structures, the second part focuses on creating the setting for modeling charged particle dynamics in such magnetic configuration, and studies its effects on the kinetic energy distribution of a large test-particle population. The results indicate that this environment is an impulsive, highly efficient particle accelerator, predominantly dependent on the slope of the distribution function of the electric fields’ strength (a power-law). The temporal variation of this parameter is the subject of ongoing research, especially in terms of the evolution of solar active regions through a sequence of continuously forming and dissipating structures known as nanoflares - microflares - flares, leading to several solar energetic events such as large scale flares and/or coronal mass ejections.
Acknowledgments

The research described in this thesis was supported by the Hellenic National Space Weather Research Network (HNSWRN) via the THALIS Programme.

I sincerely wish to thank Professor Loukas Vlahos [Aristotle University of Thessaloniki], for giving me the opportunity to work alongside him and his team, and for the precious advice and assistance I received during the course of this work from all group members I now consider dear colleagues.

We thank Dr. Manolis K. Georgoulis [RCAAM of the Academy of Athens, Greece] for kindly providing us with the vector magnetograph data this work is based on, Dr. Michaila Dimitropoulou [University of Athens, Department of Physics, Greece] for performing the magnetic field extrapolations as well as a major part of the data analysis during the first phase of this project, and Georgios Chintzoglou, PhD [School of Physics, Astronomy and Computational Sciences, George Mason University, VA, USA] for continuing a part of Dr. Dimitropoulou’s work in extrapolating the magnetic fields with higher resolution during the second phase.

I personally would like to thank Dr. Heinz Isliker [Department of Physics, University of Thessaloniki, Greece] for introducing me to parallel programming and for his assistance on several occasions during the initial stages of this project.

I also personally thank Dr. Anastasios Anastasiadis [National Observatory of Athens, IAASARS], Dr. Manolis K. Georgoulis and Dr. Michaila Dimitropoulou for many helpful discussions during the course of this work.

Anna Toutountzi
# Contents

1 Introduction .................................................. 1

2 Observations and Data Processing ....................... 5
   2.1 Data Pre-Processing: Heliographic Coordinate System – 180° Disambiguation .... 6
   2.2 3D Magnetic Field Reconstruction: Extrapolation Techniques .................. 8
      2.2.1 Potential Field Source Surface ........................................ 8
      2.2.2 Vector Magnetogram Pre-Processing ..................................... 10
      2.2.3 Nonlinear Force-Free Field ............................................ 11
   2.3 Data Reduction Reliability: The Free Magnetic Energy Criterion ............... 13
   2.4 Finalizing the Dataset .................................................. 16

3 Data Analysis .................................................. 18
   3.1 Clustering .......................................................... 18
   3.2 Data Derived Properties .............................................. 20
      3.2.1 Fractal Dimension ................................................... 20
      3.2.2 Spatial Extent ...................................................... 25
      3.2.3 Dynamical Potency .................................................. 25
   3.3 Final Remarks on Data Validity ....................................... 30

4 Model Overview ................................................ 31
   4.1 Basic Model Elements: The Environment .................................... 32
      4.1.1 Free Travel Distances ............................................... 33
      4.1.2 Acceleration Lengths ................................................. 34
      4.1.3 Electric Field Strength .............................................. 35
   4.2 Charged Particle Dynamics ............................................. 40

5 Particle Simulations: Results and Discussion ........ 46
   5.1 A Parametric Study ................................................. 46
   5.2 Particle Energization and Diffusion in Velocity Space ......................... 47

Appendices ......................................................... 52

A Numerical Calculation of the div and curl Operators in 3 Dimensions .... 52
   A.1 Vector Calculus: Divergence and Rotation of a Vector Field ................. 52
   A.2 Finite Differences Using the Method of Undetermined Coefficients ......... 52

B A Spatial Clustering Algorithm .................................. 57
   B.1 Stating the Problem and Classifying the Clustering Process ................. 57
1 Introduction

The full interpretation of the electromagnetic spectrum of the solar corona has been puzzling the world of Astronomy for over 70 years. Even though the emitting corona gives insight into the processes taking place in the solar atmosphere from across the entire electromagnetic spectrum, the most important questions risen in the past decades relate to the energy release processes converting magnetic energy into particle energy, heat, waves and motion, and therefore directly link to observations in EUV, soft and hard X-rays, through which these processes predominantly manifest.

The spectral feature mostly used to infer high temperatures in the solar atmosphere is the EUV lines emitted by highly ionized heavy elements in the plasma. Early discoveries of such lines in the coronal spectrum (Grotrian, 1939; Edlén, 1942), indicated that the coronal gas has a temperature exceeding 1 million degrees Kelvin. That, in conjunction with the relatively low temperatures at the lower chromosphere (tens of thousands degrees Kelvin) eliminating the prospect of heat conduction, gave rise to the problem known as “coronal heating”.

The thermal component of the coronal spectrum lies in the EUV and soft X-ray band (\(\lesssim 10\) keV), with the soft X-rays also presenting a non-thermal, approximate power-law distribution of photons at the high energy tail. The inference of electron energy distributions from photon spectra is based on collisional models for bremsstrahlung emission, known as “thin” (Datlowe and Lin, 1973) and “thick” (Brown, 1971) target, generally applied to thin, hot and dense, cold plasmas respectively, but also depending on the (incident) electron energy.

Hard X-ray (\(\gtrsim 10\) keV) sources in the solar atmosphere are mainly associated with the chromospheric footpoints and loop-tops of coronal loops in flaring active regions. The non-thermal (mostly hard X-ray) part of the coronal spectrum is associated with electrons accelerated up to energies of even tens of MeV. The inversion of a near power-law photon spectrum with an absolute index \(\gamma\) using a thick target model gives a power-law electron energy distribution with and absolute index \(\delta = \gamma - 1\). All observations indicate that \(\delta > 2\) (see e.g. Benz, 2008).

The coupling of the atmospheric layers (chromosphere - corona), suffices to explain the heating of the dense upper chromosphere by means of a coronal process that heats and/or accelerates particle populations which then precipitate into the dense chromosphere tracing the closed magnetic field lines towards the footpoints, where they loose their entire kinetic energy in a cold (thick) target. Both heating and the production of the hard X-ray spectrum from the footpoints is thus explained through thermal conduction and/or free-streaming of energetic particles. In another secondary process called “evaporation”, the hot chromospheric plasma subsequently expands into the corona, thus possibly (re-) heating it.

The sequence of processes described above, gives the general idea behind what is also known as the “standard flare model”. This scenario only applies to solar active regions and, apart from accounting for a series of secondary mechanisms, it also places the primary mechanism of mag-
netic energy release, into the corona. The acceleration of particles therefore, absent in the quiet Sun and unable to heat the generally thin (target) coronal plasma, as well as the heating taking place in both quiet and active regions, require an efficient coronal mechanism, maybe even accountable for both (see e.g. Gontikakis et al., 2013).

The general, widely accepted process by which magnetic energy is built up and subsequently released into the corona, initiates from flux emergence from the convection zone and/or the shuffling of the photospheric footpoints, causing built up discontinuities (shears and stresses) to accumulate free magnetic energy, driving the overall magnetic topology towards an unstable state (Parker, 1972, 1983). This results to an impulsive reconfiguration of the magnetic field via energy release in localized regions (Parker, 1988), where a sudden increase in resistivity dissipates the excess magnetic energy. This is the process usually implied when the term “magnetic reconnection” is used in this context.

Depending on the energy content of the unstable volumes involved in the magnetic reconnection process, and a variety of other properties admitted by the larger scale structure they form as a whole, these regions are often called “flares”, “microflares” or “nanoflares” (Parker, 1988), moving towards smaller and smaller scales in all aspects and also, accordingly addressing the problem of (flare) acceleration and heating (or both), or even “unstable current sheets”, often in terms of the particle acceleration mechanisms proposed in relation to these structures.

Today’s general framework for studying these larger scale structures is flare forecasting, using among a variety of tools, the so-called “flare parameters” (see §3.2.3 for a brief description). In its essence, however, this form of investigation has, in the past, and continues to be conducted in a broader manner, including and combining both theoretical and observational aspects, also extending the study to cover the quiet Sun as well as the flaring active regions.

Referring to the unstable volumes discussed as “events”, and concentrating on the distribution of these events in energy, Hudson (1991) conducted an observationally based investigation regarding the power-law distribution of flare total energies (already reported to have an absolute index $\sim 1.8$), and suggested that a second, “softer” branch, extending to lower energies should exist, describing the energy distribution of those events Parker called “nanoflares” (Parker, 1988). Vlahos et al. (1995); Georgoulis et al. (1995); Georgoulis and Vlahos (1996, 1998) pursued a modeling approach to that same issue. Their “statistical flare” model rendered a double power-law spectrum in flare energies, with the steep and flat parts admitting absolute indices $\gtrsim 1.85$ and $\lesssim 1.62$ respectively. By the time Benz and Krucker (2001) investigated the heating of the quiet solar corona by nanoflares, different studies related to the distribution of flares in energy had not yet reached an agreement concerning the power-law index. Finally, a comparative study between observational data and the statistical flare model performed by Georgoulis et al. (2001), showed that, even though both data and model can be similarly processed/modulated towards coincidence (see §3.2.3), even rendering two different power-law scaling behaviors correspond-
ing to two populations of energetically “weak” and “strong” events, there is still inconclusive
evidence as to whether those two populations, could lead to a double scaling of their composite
energy distribution.

Turning to a particle acceleration mechanism related to the magnetic energy release process
and elements discussed so far, Vlahos et al. (2004) proposed a simple model based on certain
statistical properties admitted by an assumed network of acceleration centers embedded in the
overall magnetic structure above an active region. This complex structure was assumed to have
fractal properties and the acceleration centers comprising it represented the dissipation regions
(or unstable currents sheets) discussed above, by means of a DC electric field. Given three ad
hoc probability distribution functions according to which the distances between those regions, the
strong electric fields they host and their sizes are distributed over the network, Vlahos et al. (2004)
setup a continuous time random walk mapping to model the interaction of charged particles with
this complex structure. A setting in which a particle performs a free flight between these centers,
traveling a distance specified by the first of the above distributions, and then encounters a part of
the fractal, interacts with it via an electric field the strength of which is dictated by the second
distribution, while the particle travels a distance given by the third one, sets up a stochastic
acceleration process. Vlahos et al. (2004) showed that this mechanism is efficient in both heating
and accelerating electrons to up to 1MeV, in less than 1sec.

The present study focuses on applying the particle model described above, this time relying
on an extensive analysis of observational data to provide the forms, as well as most of the param-
eters of the three probability distribution functions incorporated in the model. A major part of
this study is therefore devoted in processing and analyzing the data available, with an intention to
emphasize, at the same time, on the interest this kind of data extraction presents in its own right.

In the next section 2, the set of vector magnetograph data used for this study is introduced,
and the application of a series of standard data pre-processing and processing techniques is de-
scribed. A validation check, based on the free magnetic energy of an active region is then carried
out, in order to finalize the dataset upon which all further analysis is to be applied. In sec-
tion 3, the extrapolated magnetic fields obtained and validated during the previous stage of this
work, are analyzed using a set of tools and methods employed or developed as part of this study.
These tools and methods, along with their product results are thoroughly presented there. Having
extracted the volumes of interest, representing the dissipation regions discussed in this introd-
uction, and after obtaining information regarding several statistical properties of the larger scale
structure they comprise, we proceed in the next section 4 with presenting the particle model.
After discussing several considerations based on which the basic elements of the model (prob-
ability distribution functions) were deduced, we outline the derivation of the mapping scheme
describing the dynamics of charged particles, interacting with the simulated magnetic structure.
The last section 5 begins with a description of a parametric study we performed, by varying the
power-law index of the probability distribution function followed by the electric fields the particles encounter. The results on particle energization and diffusion in velocity space are finally presented and discussed.
2 Observations and Data Processing

We built our model’s basic features upon photospheric vector magnetograms obtained by the Imaging Vector Magnetograph (IVM) of the University of Hawaii’s Mees Solar Observatory. Our data consist of a sequence of 3 photospheric vector magnetograms of the active region NOAA AR 10254 at 21:44, 21:52, and 22:01 UT. The active region was observed by the IVM on 2003 January 13 and its location on the solar disk was S16°, E44° (figure 1). The Stokes spectra for the three magnetograms were obtained as IVM’s Fabry-Pérot interferometer scanned the magnetically sensitive line Fe I 6302.5Å with a spectral resolution of 0.07Å, covering a wavelength range of ∼ 0.5Å. Their field of view (FoV) is 4′.7 × 4′.7, covered using a 512 × 512 pixel CCD array detector with a spatial resolution of 0″.55 × 0″.55 per pixel. (Mickey et al., 1996) Binning the pixels 2 × 2, renders an effective pixel size of 1″.1 × 1″.1, or ∼ (0.78 × 0.78)Mm and an image size of 256 × 256. Since it did not host a higher than a C-class flare, NOAA AR 10254 is considered a non-flaring active region (Tziotziou et al., 2012).

![Figure 1: The measured LoS magnetic field of NOAA AR 10254 (frame 21:44 UT) observed by the IVM. The pixel size of the unbinned IVM image is 0″.551. Tick mark separation is 10 pixels 20″. Figure adapted from Georgoulis (2005b).](image)

The 3D magnetic field above NOAA AR 10254 was extrapolated into the corona using both a potential field source surface (PFSS) model and a force-free field extrapolation (NLFFF) method (§2.2). Prior to both extrapolation techniques the Stokes inverted magnetograph images were disambiguated (§2.1), while the NLFFF extrapolation was also preceded by a pre-processing procedure mostly aiming to improve the consistency of the magnetograph data with the force-free assumption (§2.2.2). This section concludes with one last step of data reduction using free
magnetic energy computations in order to validate the results obtained during the previous stages of data processing.

2.1 Data Pre-Processing:

Heliographic Coordinate System – 180° Disambiguation

The so-called azimuth ambiguity problem relates, in the context of heliophysics, to the symmetric nature of the linearly polarized Zeeman components, causing the transverse magnetic field $B_t$ retrieved from the Stokes inversion method (LaBonte et al., 1999; Li et al., 2009) to not be uniquely defined in terms of its orientation in the image plane, i.e. there are two equally likely solutions separated by 180° in azimuth (see Borrero and Ichimoto, 2011; Wiegelmann and Sakurai, 2012, for recent reviews).

The transverse $B_t$ and longitudinal $B_\ell$ magnetic field components obtained from magnetograph measurements refer to the line-of-sight (LoS) coordinate system. For an AR observed away from disk center, the finite angle between the normal to the solar surface and the LoS produces the so-called projection effects, responsible for various uncertainties arising in relation to the perceived magnetic field (Venkatakrishnan et al., 1988). It has been shown that these effects can be eliminated to a sufficient degree (Hagyard, 1987; Gary and Hagyard, 1990), if the magnetic field is transformed to the heliographic reference frame. Gary and Hagyard (1990) summarized the previous work done in deriving the relevant transformations (Hagyard, 1987; Venkatakrishnan et al., 1988) and after further investigation concluded that, despite all the advantages of the heliographic coordinate system in overcoming the projection effects for off-disk center images, the azimuth ambiguity problem, if not properly addressed, can cause the heliographic projection to produce artificial magnetic field discontinuities and electric currents leading to a false perception of an AR’s overall configuration.

After transforming the magnetograms in our dataset to the heliographic coordinate system following Gary and Hagyard (1990), we pre-processed them for azimuth ambiguity removal using the non-potential magnetic field calculation (NPFC) method initially developed by Georgoulis (2005a) and later refined by Metcalf et al. (2006). Albeit far beyond the essence of this method (rather close to its consistency), it is to note that when applied to magnetograms close to disk center, the NPFC reduces to a certain class of techniques known as acute angle algorithms. In its simplest form, an acute angle method uses the longitudinal LoS magnetic field component $B_\ell$ as a boundary condition in a potential field calculation (see §2.2.1). The method’s purpose is to use the transverse component $B^P_i$ of the potential field as a reference for comparison with the respective observed magnetic field components $B^i_i$, with $i = \phi, \phi + \pi$ denoting the two transverse solutions. The method takes its name from the condition $B^i_i \cdot B^P_i > 0$ it imposes, that the angle between $B^i_i$ and $B^P_i$ be acute (Metcalf et al., 2006). The azimuth is disambiguated by selecting for every point in the 2D magnetographic grid, the transverse solution satisfying the above criterion.
As in the acute angle method, the NPFC algorithm involves applying the acute angle criterion at a certain point during execution. In the NPFC method however, as its name suggests, the reference field $B$ for the application of that criterion is assumed to be non potential. The main issue NPFC addresses, is the computation of this reference field. The method realizes a convergent iterative scheme starting from the ambiguity pair of heliographic solutions $B_1 = (B_{x1}, B_{y1}, B_{z1})$ and $B_2 = (B_{x2}, B_{y2}, B_{z2})$ to form a random distribution $B_z$ over the 2D magnetographic grid using $B_{z1}$ and $B_{z2}$. $B_z$ is then used as a boundary condition to calculate a potential field $B_P$ on the image plane $S$. $B_P$ is considered to be the potential component of the total non potential magnetic field $B$ on $S$, also contributed by a current carrying component $B_C$, such that $B = B_P + B_C$. Provided $B_C$ is divergence-free and lies on $S$, a solution in Fourier space for $B_C$ (Chae, 2001) reads

$$B_C = \mathcal{F}^{-1}\left[ \frac{ik_y}{k_x^2 + k_y^2} \mathcal{F}\left( \frac{4\pi}{c} J_z \right) \hat{x} \right] + \mathcal{F}^{-1}\left[ - \frac{ik_x}{k_x^2 + k_y^2} \mathcal{F}\left( \frac{4\pi}{c} J_z \right) \right] \hat{y},$$

(1)

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the direct and inverse Fourier Transforms of their arguments respectively, $c$ is the speed of light in free space and $J_z = (c/4\pi)(\nabla \times B_C) \cdot \hat{z}$ the current density component in the vertical direction arising through Ampère’s law. Note that, even tough $J_z$ is not known form measurements, it is nonetheless required to fully determine $B_C$.

In estimating $J_z$, Georgoulis made use of the common argument many disambiguation methods base their algorithms on: *Seek the solution which minimizes the estimated current*. While this argument is challenged by the fact that the magnetic field in the photosphere is not necessarily force-free, let alone current-free (a matter discussed further in section 2.2.2), it gains some merit considering that most occurrences of un-physically high estimated values for $J_z$, trace back to the artificial discontinuities in $B_t$ caused by incorrectly resolved ambiguities. Therefore finding an ambiguity-free estimator for $J_z$ seems like a good place to start. Exploiting the fact that, as suggested by the transformations to the heliographic coordinates (Gary and Hagyard, 1990), along with the symmetry implied by the azimuth ambiguity, all components of the sum $B_1 + B_2$ are proportional to $B_t$ (see Georgoulis, 2005a), this ambiguity-free estimator (or proxy) $J_{z,p}$ is defined as

$$J_{z,p} = \frac{c}{4\pi} \left[ \nabla \times \left( \frac{B_1 + B_2}{2} \right) \right] \cdot \hat{z}$$

(2)

where averaging the pair of ambiguity solutions is just an obvious choice.

Once $B$ is computed, the NPFC algorithm proceeds by using its transverse component $B_t$ as a reference for comparison with the respective observed magnetic field components $B_{1,t}$ and $B_{2,t}$. The acute angle criterion $B_{i,t} \cdot B_t > 0, \ i = 1, 2$ (Metcalf et al., 2006) provides a first estimation of the disambiguated distribution $B_z$ (initially set random), which is then used to repeat the whole
iteration scheme until convergence is reached. The convergence criterion leading to a unique solution demands that discontinuities, such as abrupt vector flips (Metcalf et al., 2006) in strong field regions indicating a poor ambiguity removal, cease to accumulate as iteration proceeds (see Georgoulis, 2005a; Metcalf et al., 2006, for further details).

The refinement proposed by Metcalf et al. (2006) includes 1) an improved estimation of $J_{z,p}$ incorporating a term depending on the image distance from disk center, 2) the update of $J_{z,p}$ at each iteration based on the improved estimate of the previous step, 3) padding the boundary data with zeros in order to smooth out the effect of the periodic boundary conditions in the Fast Fourier Transforms employed in the calculation of $B_P$ and $B_C$, and 4) introducing a convergence criterion equivalent to the original one, but more effectively implemented in the algorithm.

2.2 3D Magnetic Field Reconstruction: Extrapolation Techniques

At present, the most viable and widely used approach for reconstructing the coronal magnetic field above an observed AR is a nonlinear force-free extrapolation that uses as a boundary condition the photospheric field obtained from vector magnetograph measurements. This is due to the NLFFF method’s ability to model current carrying magnetic structures, allowing storage of the free magnetic energy considered driving the majority of both small and large scaled phenomena in the solar corona. Contrary to the NLFFF, the potential field source surface models originally proposed to treat that same problem, were based on the assumption that the coronal magnetic field is potential. Among the reasons why this latter approach, even though outdated in terms of its use for the chromosphere and lower corona magnetic field reconstruction, has not become obsolete, is that a PFSS model renders the magnetic field with the lowest energy for a given set of boundary conditions (see, however, beginning of §2.3). This energy can then be used as a reference in estimating the free magnetic energy content of an AR, provided the total magnetic energy has already been obtained by means of e.g. a NLFFF extrapolation (as in Su et al., 2014, for example).

2.2.1 Potential Field Source Surface

The PFSS model employed (Schatten et al., 1969; Altschuler and Newkirk Jr, 1969; Wang and Sheeley Jr, 1992), provides an analytical solution for a potential magnetic field above an observed AR, between the photosphere and an outer “source surface”. The role of this source surface is to model (figure 2) a transition layer across which the solar wind outflow gradually gains dominance over closed field line configurations transporting them out into the interplanetary space, just before its effect is counterbalanced by that of the Sun’s rotation (Schatten et al., 1969). This
potential field is the solution to Laplace’s equation

$$\nabla^2 \Phi = 0,$$

(3)

based on the potential field assumption $\nabla \times \mathbf{B} = 0$ implying that $\mathbf{B} = -\nabla \cdot \Phi$, where $\Phi$ is a scalar potential field, along with the divergence-free condition $\nabla \cdot \mathbf{B} = 0$. All quantities and operators involved are space dependent, expressed in spherical coordinates.

**Figure 2:** Schematic representation of the source surface model. Regions 1, 2, and 3 represent areas below the sun’s photospheric surface, between the photosphere and the source surface, and beyond the source surface into the interplanetary space respectively. Figure adapted from Schatten et al. (1969).

Solving Laplace’s equation requires a set of boundary conditions to appropriately define the solution and an estimation of the source surface’s radial distance from the Sun’s center. Early inquiries regarding the boundary conditions (Schatten et al., 1969; Altschuler and Newkirk Jr, 1969) seemed to settle after Wang and Sheeley Jr (1992) in allowing the photospheric plane to carry surface currents. In accordance to observations, the magnetic field on the photosphere is assumed to have an approximate radial direction. Matching the respective component $B_r^{obs}$ deduced from the LoS measurements, the bottom boundary is restricted by the condition $B_r |_{R_\odot} = B_r^{obs}$. At the outer surface, considered to be the “source” of the interplanetary magnetic field (Schatten et al., 1969), the condition $B_\theta = B_\phi = 0$ is imposed, implying a radial direction in this case as well. The radial distance of the source surface was also subject to several revisions with values ranging from $1.3R_\odot$ to $> 3R_\odot$, leading to a common use of $2.5R_\odot$ (Mackay and Yeates, 2012).
The solution for the three components $B_r$, $B_\theta$ and $B_\phi$ is expressed in terms of spherical harmonics (see Wang and Sheeley Jr, 1992; Mackay and Yeates, 2012). The computations are performed numerically, requiring a number of harmonics inversely proportional to the distance from the center of the Sun and therefore also dependent on the resolution of the photospheric image (Mackay and Yeates, 2012).

### 2.2.2 Vector Magnetogram Pre-Processing

Even with the 180° ambiguity removed, there are still several issues one has to address in order to make a nonlinear force-free extrapolation scheme to consistently reconstruct the coronal magnetic field based on photospheric measurements. Wiegelmann et al. (2006) highlighted the most prominent issues involved in this matter and to address some of them, created a pre-processing procedure to drive the observed magnetograms into a state that is more compliant with certain force-free conditions (see below), yet departing from the original state only within measurement errors. In what follows, a brief account of these issues is given, following the Wiegelmann et al. (2006) report.

- One of the main issues arising in such a treatment is caused by the fact that, while the coronal magnetic field can be consistently considered force-free (low-$\beta$ plasma), when it comes to the photospheric magnetic field this is not necessarily the case. In the photosphere the plasma $\beta$ is closer to unit so the force-free assumption can no longer be as easily justified (also see Gary, 2001). This effectively disrupts the consistent connection between the boundary conditions (magnetogram) and the solution (extrapolated field).

- Another problem, a rather inescapable one unless instrumentation is considerably improved, has to do with the inaccuracy of the measurements of the magnetic field in the photosphere. It is moreover found that the noise levels in the ambiguous transverse component $B_t$ are usually near one order of magnitude greater than those in the longitudinal component $B_\ell$, as a consequence of their deduction using the Stokes vector $(I, Q, U, V)$ (see Wiegelmann et al., 2006, for further details) [Improvements in this matter are, of course, relevant to newly obtained measurements].

- The last issue of interest is the necessity born of computational cost limitations, to smooth the magnetograph data. A first instance of what turns out to be common practice among data extraction and manipulation techniques, was already encountered during the technical specification of our dataset. Binning of the magnetograms by some factor (here 2), is often used to enhance the speed of calculations performed numerically, by reducing the size of the grid spanned. A first level of smoothing is achieved during this process. Smoothing a magnetogram once the grid-size has been determined, is a process that usually refers to
driving the 2D magnetic field closer to a “differentiable state” in order to achieve faster convergence of the numerical methods later applied.

The conditions magnetograph data have to fulfill for the force-free assumption to hold derived by Ali (1989), admit two out of four of the criteria Wieglamann’s routine incorporates, termed as force-free and torque-free conditions. The other two are included to ensure that the resulting pre-processed magnetogram is sufficiently smoothed and in agreement with the original one [note that the noise levels pertinent to photospheric measurements, work in favor of such a procedure by broadening the limits of adjustment (Wiegelmann et al., 2006)]. The iterative smoothing routine seeks to identify the magnetic vector \( \mathbf{B} \) on each point in the 2D magnetographic grid, that minimizes the functional

\[
L_{\text{pre-pro}} = \mu_1 L_1 + \mu_2 L_2 + \mu_3 L_3 + \mu_4 L_4, \tag{4}
\]

where the \( L_n, n = 1, 2, 3, 4 \) functionals correspond to the chi-squared deviations from fulfillment, of each of the criteria mentioned above. The \( \mu_n \) parameters serve as weighting factors that can be adjusted so that all four conditions are met at the same time and possibly give greater weight to selected criteria. They obviously need to be specified before the smoothing scheme is executed (see Wiegelmann et al., 2006).

As mentioned earlier, the procedure described aims at addressing some of the issues pertinent to the matters discussed in this section. There are therefore several conditions one has to make sure are met for the pre-processing to work as expected. These include: 1) that the 180° ambiguity in the observed magnetograms has been resolved, 2) the AR was imaged close to the disk center (these two conditions also ensure that the transformation to the heliographic coordinates effectively prevents projection effects from adding up to the errors in the magnetic field measurements), and 3) that the photospheric flux is more concentrated near the center of the image than towards the boundaries, so that the flux balance condition can be checked using only the bottom boundary of the computational domain.

All three magnetograms comprising our dataset were pre-processed using the method discussed above. A brief discussion relating to the effectiveness of its performance on the particular set of data is, however, postponed until §2.3, where the reliability of the entire data reduction described in this section is put into question.

### 2.2.3 Nonlinear Force-Free Field

The 3D magnetic field above NOAA AR 10254 for all three frames was reconstructed using the nonlinear force-free field optimization method developed by Wiegelmann (2004), as an extension to previous work done by Wheatland et al. (2000) in introducing the optimization principle this method is based on. Wiegelmann’s method implements the computation of both a nonlin-
ear force-free field (NLFFF) and a non force-free field within the same program. We are here
only interested in the former, since treating the non force–free field requires additional data not
incorporated in the current study (see Wiegelmann, 2004, for further details).

The main idea behind Wiegelmann’s method is to iterate for the magnetic field using a vari-
able step-size, forcing each iteration to be repeated until a functional minimization condition is
fulfilled, or the step-size drops below some pre-specified limit. Using a potential field calculation
as a starting point, the estimation of the force-free field is progressively improved by means of
this functional minimization criterion. The functional set up for the NLFFF method satisfies,
upon minimization, the basic force-free equations and is defined as

\[ L = \int_V w(r) \left[ B^{-2} |(\nabla \times B) \times B|^2 + |\nabla \cdot B|^2 \right] \, dr, \]

where \( B \) is the nonlinear force-free magnetic field, \( r = (x, y, z) \) the position vector defined
appropriately on the computational domain, and \( w = w(r) \) a weighting function the role of which
is explained below. Note that the boundary conditions, namely the observed vector magnetograph
data, come in by means of the derivatives involved in both the magnetic field iteration equation
and \( L \)'s computation.

Instead of imposing a rigid form of boundary conditions for the lateral and top boundaries of
the computational domain, where there are no measurements for the magnetic field, Wiegelmann
made use of the weighting function \( w(r) \) already mentioned by Wheatland et al. (2000), but not
implemented. By allowing \( w(r) \) to vary from unity (according to some specified function) across
a boundary layer of finite thickness defined on all but the bottom boundary (where magnetograph
measurements are available), Wiegelmann showed that the magnetic field deduced for the inte-
rior (physical) domain is affected less severely by the arbitrariness of the boundary conditions
imposed on the outer layer surface. This in effect means that, at least for the NLFFF recon-
bstruction, the method strictly requires only the bottom boundary conditions obtained from vector
magnetograms.

In order to emphasize the prerequisites enhancing the performance of the processing tech-
niques described in the latter two paragraphs, we briefly mention that an appropriately disam-
biguated, close to disk center image with a photospheric flux mostly concentrated near its center
rather than the boundaries (see Wiegelmann, 2004; Wiegelmann et al., 2006, for further details),
with as low noise levels as possible, suffices to render a satisfactory reconstruction of the mag-
netic field over an AR.

Both PFSS and NLFFF were constructed using a spatial step-size matching the binned mag-
netograms’ pixel size in all three directions. In the NLFFF extrapolation a boundary layer of
thickness 16 pixels was used, reducing the physical domain to a size of 224 × 224 × 240. Figure 3
shows a visual representation of the magnetic field configuration above NOAA AR 10254 (frame 21:52 UT), as inferred using both methods. The magnetic field lines were generated using a code developed as part of this thesis, implementing the 3D similarity-guided streamline placement technique introduced in Chen (2008).

![Magnetic field configuration above NOAA AR 10254 (frame 21:52 UT), as reconstructed by the NLFFF extrapolation (left) and the PFSS model (right). The bottom plane is a density plot of the vertical magnetic field component (heliographic coordinates) on the boundary magnetogram. The color scheme refers only to the density plots. The pixel size is $1''$, or $\approx 0.78$ Mm for all three dimensions. Tick mark separation is 10 pixels.]

**Figure 3:** Magnetic field configuration above NOAA AR 10254 (frame 21:52 UT), as reconstructed by the NLFFF extrapolation (left) and the PFSS model (right). The bottom plane is a density plot of the vertical magnetic field component (heliographic coordinates) on the boundary magnetogram. The color scheme refers only to the density plots. The pixel size is $1''$, or $\approx 0.78$ Mm for all three dimensions. Tick mark separation is 10 pixels.

### 2.3 Data Reduction Reliability:

**The Free Magnetic Energy Criterion**

The available magnetic energy considered to source both small and large scaled solar events hosted by, or emanating from an AR is stored in excess to some reference energy level corresponding to a quiescent state of the magnetic field above the AR. The free magnetic energy $E_{\text{free}}$ of an AR quantifies the deviation of the coronal magnetic field from this reference state, considered to be either the magnetic energy of the linear force-free field assuming the magnetic helicity of the configuration is conserved (see e.g. Regnier and Priest, 2007), or that of the potential field, in which case $E_{\text{free}}$ provides an upper limit for the energy that is actually available (Su et al., 2014).

Of the methods estimating the free magnetic energy of an AR based on its 3D magnetic field,
the most straightforward one relies on two analytic expressions given in equations (6) below.

\[ E_{\text{free},1} = \frac{1}{8\pi} \int_V B^2 dV - \frac{1}{8\pi} \int_V B_P^2 dV \]  

(6a)

\[ E_{\text{free},2} = \frac{1}{8\pi} \int_V (B - B_P)^2 dV, \]  

(6b)

where \( B \) and \( B_P \) are the NLFFF and PFSS respectively and all volume integrals are computed over the entire computational domain from the photosphere to the corona. The first of these equations defines \( E_{\text{free},1} \) to be the difference between the total magnetic energy of the 3D magnetic field, i.e. the NLFFF (first integral on the right hand side of equation 6a) and the magnetic energy of the potential field (second integral in equation 6a). Equation (6b) defines \( E_{\text{free},2} \) to be the magnetic energy of the “source field” \( B - B_P \) describing the nonpotentiality of the magnetic field (Su et al., 2014, and references therein).

Under ideal circumstances, both these formulas are equivalent ways of expressing the free magnetic energy of an AR. Their use however, entails the use of \( B \) and \( B_P \) as both inferred from imperfect and model dependent methods. In Moraitis et al. (2014) the difference between \( E_{\text{free},1} \) and \( E_{\text{free},2} \) is shown to stem mostly from a NLFFF calculation failing to reconstruct a precise divergence-free magnetic field. Violation of the solenoidal condition in terms of the potential field is considered more easily controlled and therefore thought to be a less important source of errors. In light of these facts \( E_{\text{free}} \) is usually estimated by averaging \( E_{\text{free},1} \) and \( E_{\text{free},2} \) with an uncertainty calculated based on their difference (Su et al., 2014; Moraitis et al., 2014), providing at the same time a measure for the quality of a NLFFF method’s performance on a particular set of observations.

The free magnetic energy is defined accordingly as

\[ E_{\text{free}} = \frac{1}{2}(E_{\text{free},1} + E_{\text{free},2}) \]  

(7)

and for the uncertainty in its estimation we employ the approach of Su et al. (2014) by defining the relative error in \( E_{\text{free}} \) as

\[ r_{E_{\text{free}}} = \frac{\Delta E_{\text{free}}}{E_{\text{free}}}, \]  

(8)

where

\[ \Delta E_{\text{free}} = \frac{1}{4\pi} \int_V |B_C \cdot B_P| dV, \]  

(9)

with \( B_C = B - B_P \) the current carrying field component of the total magnetic field (see §2.1). Note that, depending on the quality of the extrapolation, \( E_{\text{free}} \) (owing to \( E_{\text{free},1} \)) and therefore also \( r_{E_{\text{free}}} \) might actually turn out to be negative.
The discrete analogues of equations (6), (8) and (9) are obtained following e.g. Vlahos and Georgoulis (2004), by replacing $B$ and $B_P$ with the respective vectors $B_i$ and $B_{P,i}$ at location $i = 1, \ldots, N$ in the 3D grid, the integrals with summations over the entire computational domain comprised of $N$ 3D pixels, and the infinitesimal volume element $dV$ with $\delta V = \delta p^2 \delta h$, where $\delta p$ is the pixel size of the binned magnetogram and $\delta h$ the spatial step-size of the extrapolations (in our case $\delta p = \delta h \approx 0.78\text{Mm}$).

In their study on flare productivity of ARs in relation to possible correlations between the free magnetic energy and an AR’s flare index, Su et al. (2014) used 0.3 as a tolerance value for $r_{E_{\text{free}}}$, computed over the entire computational domain. Along with the tolerance value imposed on another index incorporated in their study, measuring the flux imbalance of their extrapolated fields, they validated their analysis sample by dropping over 50% of their initial dataset. Following Su et al. (2014) we performed a similar reliability test on our extrapolated fields.

In each of the three cases studied, the uncertainty in free magnetic energy turned out to be nearly half an order of magnitude greater than $E_{\text{free}}$ itself. The specific values in ergs are $(7.15 \pm 29.00) \times 10^{31}$, $(9.02 \pm 36.00) \times 10^{31}$ and $(8.78 \pm 30.10) \times 10^{31}$ for frames 21:44, 21:52 and 22:01 UT, with relative errors 4.06, 3.99 and 3.43 respectively. It should be noted here, that this test was also performed on the extrapolated fields obtained without pre-processing the corresponding magnetograms and it turned out that in all cases, the free magnetic energy was estimated to be negative with even greater absolute values for the relative errors ($\sim 11$).

These high discrepancies between $E_{\text{free},1}$ and $E_{\text{free},2}$, even after the pre-processing procedure took place, can be attributed to mainly two factors. The first one is the increased noise levels in IVM magnetograph measurements. As Wiegelmann (2004) reported, there is a linear dependence between the quantities driving the NLFFF optimization (minimization) algorithm to convergence and the noise level. Secondly, while the NLFFF optimization method simultaneously pursues the minimization of $|\nabla \cdot B|$ and $|J \times B|$, thus introducing inaccuracies to the results in an intrinsic manner (Valori et al., 2013; Moraitis et al., 2014), the flux balance condition set on magnetograph data is the only “fort” left to “defend” the solenoidal condition. Therefore, if the magnetograms processed present a magnetic flux imbalance, this is expected to significantly affect the extrapolations. Finally, the derivatives involved in both $B$ and $L$’s (equation 5) calculations are expected to raise uncertainties to only a second order degree and poor resolution has been suggested to not significantly affect the results in the free magnetic energy computations (Su et al., 2014).

In order to be able to proceed with our analysis, the next action taken was to extract out of the entire volumes resolved, those fractions comprised of individual 3D pixels with a local relative error in the free magnetic energy less than 0.3. This was based on the reasoning given below.

As suggested by equations (6), any finite volume within our computational domains, including the actual volume elements $\delta V$ involved in all down to resolution scales’ computations, is
entitled to a valid assignment of $E_{free}$. In practice, however, obtaining an accurate value for the free magnetic energy in any such given $\delta V$, has an implication owed to the transition from the continuous formulas to their discrete equivalents. It requires that $B_C \cdot B_P$ be *locally* as small as possible. Demanding that $B_C$ and $B_P$ be nearly perpendicular at any given point in the entire computational domain is not a condition we expect to see fulfilled at first since, lacking any a priori physical support, it has not been incorporated in the extrapolation process to begin with. However, assuming that one is dealing with an ideal extrapolation, equation’s (9) discrete analogue suggests that $\Delta E_{free}$ has turned out to be vanishingly small as compared to $E_{free}$, not due to the presence of terms canceling each other out under summation, but due to the majority of terms taking small and comparable values (taking the absolute value sign inside an integral which would otherwise be simply equal to $|E_{free,1} - E_{free,2}|$ in equation (9), highlights this fact). This means that a test validating an extrapolation in the manner described in this section, silently seeks the condition $B_C \cdot B_P \rightarrow 0$ locally fulfilled throughout the volume of interest. The better the quality of the extrapolation, the greater the fraction of the entire volume properly resolved (i.e. fulfilling the above condition) and, as an immediate consequence, the less bias inflicted to the magnetic field vector of any individual pixel due to its neighbors, through the derivatives involved in the computations.

In order to be fully aware of the implications involved in extracting the structure consisting of pixels rendering a local value for $r_{E_{free,loc}} < 0.3$, we computed the fraction of the whole volume this structure occupies, for each of the extrapolated fields in our dataset. For frames 21:44, 21:52 and 22:01 UT, this fraction of the computational domain turned out to be 0.26%, 0.27% and 0.43% respectively. Another important note relating to the kind of data extraction discussed here is that, as it was necessary for our study, before dropping all 3D pixels failing to pass the above relative error criterion, these had to be used in the calculation of the curl operator of the extrapolated magnetic fields (see §2.4 below). The extent of the inaccuracies introduced in these computations depends on the extent to which the extracted structures are fragmented as opposed to forming “continuous” subvolumes inside their respective computational domains, and is discussed in the next section (§3.3).

### 2.4 Finalizing the Dataset

The current density vector field $J$ over the entire computational domain of the three (pre-processed) extrapolated fields was computed based on Ampère’s law

$$J = \frac{c}{4\pi} \nabla \times B,$$

where $B$ is the NLFFF. The differentiation in the curl operator was performed to second order accuracy with respect to the grid’s cell size using backward, forward and central finite difference
approximations to the first order derivatives involved (see Appendix A for a description of the method).

Beyond this point, any further analysis was performed on the small fractions of the computational domains reported in the previous paragraph, for which the extrapolation has been able to be verified, \textit{ad referendum}. Full knowledge of the processed quantities over the computational domains has only been assumed in order to compare the results obtained, against those this analysis would be based on, had the extrapolated fields not been subjected to any reliability test.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4}
\caption{Left y-axis (red filled triangles): Mean free magnetic energy per unit height of the volume selected via the validation method applied (§2.3 and §2.4). Right y-axis (blue filled squares): The fraction \(N_S/N_{xy}\) of pixels per unit height these volumes contain \((N_{xy} = 224 \times 224)\).}
\end{figure}

In order to gain some insight into the selected volumes, we show in figure 4 their mean free magnetic energy \(\langle E_{\text{free}} \rangle_S = \sum_{i=1}^{N_S} E_{\text{free},i}/N_S\) per unit height, where \(N_S = N_S(z)\) is the number of validated 3D pixels at height \(z\) (in pixel units) and also the fraction of pixels per unit height these volumes contain, i.e. \(N_S/N_{xy}\), where \(N_{xy} = 224 \times 224\). Other than stressing out the fact that this forced data reduction has cost us to lose at least four fifths of the initial volume (an all together loss for \(z \gtrsim 50\)), figure 4 does not present an unreasonable picture. \(\langle E_{\text{free}} \rangle_S\’s\) decrease with height is to be expected as the coronal magnetic field approaches its potential state at higher layers, even though the fashion and rate in which this happens are not known. The fraction of validated pixels dropping in close follow-up of the steeper decrease in \(\langle E_{\text{free}} \rangle_S\) indicates no more than an increasing difficulty to render a total magnetic field vector \(\mathbf{B}\), such that \(\mathbf{B}_C\) remains nearly perpendicular to \(\mathbf{B}_P\) as the former vanishes. This however, reflects more on the general known inefficacy of nonlinear force-free extrapolations to accurately reconstruct magnetic fields in quiet, near potential regions, than of any particular extrapolation performed, or of the validation method used. It does certainly not, however, provide an explanation as to why those two quantities should follow the same pattern and rate of decrease.
3 Data Analysis

The volumes selected via the free magnetic energy validation method described in the previous section were subject to further analysis in an attempt to reveal, within resolution limits and accuracy constraints, small-scale structures embedded in the overall magnetic field configuration shown in figure 3 (left panel), satisfying certain criteria. We identify these “small-scale structures” with current carrying unstable volumes, formed as an excess of magnetic energy accumulates near regions where magnetic discontinuities and/or excessive gradients build-up, just before its release via reconnection. As a second step of the data analysis performed in this study, we seek to investigate the spatial and dynamical correlations between these dissipation regions and recover some of the statistical properties exhibited by the resulting larger-scale structure they comprise. In this section we explore the methods and results of this second-level analysis.

3.1 Clustering

In order to locate those adjacent regions forming spatially disjoint groups (clusters) within the volumes under investigation and identify them with the “unstable volumes” this analysis aims to study, the 3D pixels comprising these clusters need to first be distinguished from the bulk volume and then be acted upon by a clustering procedure. The most commonly used and physically intuitive way to identify such “unstable” pixels is to impose a critical threshold on some scalar field known over the computational domain, considered appropriate to make this kind of distinction. For the purposes of our study, this scalar field can be chosen to be either the free magnetic energy $E_{\text{free}}$, or the magnitude $J$ of the current density. Letting $E_{\text{free}}^{\text{critical}}$ and $J^{\text{critical}}$ denote the thresholds imposed on the respective quantities, those pixels with $E_{\text{free}}$ or $J$ exceeding the above critical values are considered to have passed into an unstable state beyond equilibrium, henceforth susceptible to reconnection (see section 1).

Determining an appropriate value to be set on $E_{\text{free}}^{\text{critical}}$ is a process mostly driven by the range of values admitted by the data, in relation to the total free magnetic energy content of an AR and the degree of localization aimed for, the latter being a more or less speculative and indirect process as well. A similar process, subjected to the same degree of arbitrariness, applies to the determination of $J^{\text{critical}}$ as well as the critical value of any other scalar field chosen to facilitate this kind of pixel selection. The parametric studies investigating the extent to which any such threshold imposition influences the results (e.g. Vlahos and Georgoulis, 2004) arise, in part, from the need to justify the relevant choices made and also relate them to the underlying physical processes studied.

In light of these considerations, had the extrapolated fields in our study been more successful in passing the reliability test (described in §2.3 and §2.4), thus rendering predominantly connected volumes, a reasonable critical value for the free magnetic energy would range between
$10^{19}$ and $10^{21}$ ergs. As it turns out however, the volumes selected through validation already consist of disconnected regions and provide pixels with $E_{\text{free}}$ above that suggested threshold (see e.g. figure 4), so that any threshold imposition seems to serve no purpose at this point in the analysis. In order to avoid allowing the data to completely drive the choices made, but still include those less energetic localized events in an attempt to incorporate a nanoflaring scenario into our study (resolution limitations considered), we imposed a lower cutoff in the free magnetic energy per cluster $E_{\text{free,cutoff}}^{L} \simeq 10^{22}$ ergs.

Another issue relevant to a discussion concerning event inclusion/exclusion is that, taking into account the increasing near-potentiality of the magnetic field at higher coronal layers and giving emphasis on our search for relatively localized events, it becomes obvious that any clustering process performed on our dataset as it is or, in succession to some threshold imposition like the one mentioned above, is liable to rendering huge clusters consisting of low $E_{\text{free}}$ pixels, accumulating as wholes large amounts of free magnetic energy. These clusters were precluded from the final, statistical analysis sample by imposing a maximum (upper) allowed free magnetic energy per cluster, denoted as $E_{\text{free,cutoff}}^{U}$. In order, however, not to dismiss those highly energetic, flaring events, this upper cutoff was set to $\sim 10^{27}$ ergs (following Vlahos and Georgoulis, 2004).

The program developed to perform the clustering needed for this analysis employs a partitional, hard clustering algorithm that uses the Manhattan distance as a dissimilarity measure (Gan, 2011) to group 3D pixels into clusters, without prior specification of the total number of clusters required for completion. In the context of exploratory data mining, the end result of a partitional clustering algorithm is a single partition, as opposed to a multi-level structure of nested partitions produced by hierarchical clustering algorithms. In the same context, hard clustering (versus fuzzy clustering), restricts each data element (3D pixel) to belong to one and only one cluster. (Gan, 2011; Gan et al., 2007) The Manhattan distance, the first member of the family of the Minkowski metrics reads, in the case of 3-dimensional data

$$D(Q^i, Q^j) = \sum_{n=1}^{3} |Q^i_n - Q^j_n|,$$

(11)

where $Q^i$ and $Q^j$ are grid-points representing any two pixels within the 3D grid of the computational domain, with integer defined coordinates $(Q^i_1, Q^i_2, Q^i_3) \equiv (Q^i_x, Q^i_y, Q^i_z)$.

We define a cluster of 3D pixels to be represented by a group of grid-points, each connected to at least one other group member, called its immediate neighbor, via a single edge of the rectangular lattice underlying the 3D grid or, as a limiting case, a single point isolated in the sense just described. Given a reference point $Q^R$ and any other point $Q$ in the grid, these two points will only be immediate neighbors if $D(Q^R, Q) = 1$. The clustering algorithm proceeds using this criterion to identify and systematically record all neighboring points of the volumes under investigation (Appendix B contains a more elaborate description and some introductory
information relating to this data mining procedure).

For the three selected volumes corresponding to frames 21:44, 21:52 and 22:01 UT, the clustering algorithm provided certain cluster schemes with a total number of clusters ranging between $\sim 4000 - 5000$. These were altered insignificantly after selecting the clusters with free magnetic energy in $[E_{\text{free}}^{L,\text{cutoff}}, E_{\text{free}}^{U,\text{cutoff}}]$, as the fraction of discarded clusters was small in all cases.

It is of practical convenience for the subsequent analysis, to define a point representing a cluster. This point was chosen to be the centroid of a cluster, defined here as the point located at the weighted average position of all the grid-points comprising the cluster. This means that this representative point need not, and most probably will not, be a grid-point. Specifically, if a cluster consists of $n_C$ points $Q^i$, $i = 1, \ldots, n_C$, then the coordinates $Q^C_n$, $n = 1, 2, 3$ of its centroid $Q^C$ are defined as

$$Q^C_n = \frac{\sum_{i=1}^{n_C} E_{\text{free}}^i \cdot Q^i_n}{\sum_{i=1}^{n_C} E_{\text{free}}^i}.$$ (12)

### 3.2 Data Derived Properties

The spatial complexity of the structure formed by the reconnection sites present within an AR, as well as the extent of these dissipation regions and their dynamical potency are of great interest in their own right, regardless of any model intending to incorporate its properties in a particular study. The statistical aspect of these properties and the related results, quantified in terms of the complexity index known as fractal dimension and in the form of probability distribution functions (PDFs) followed by certain quantities of interest, is the topic of this paragraph.

#### 3.2.1 Fractal Dimension

The tools and methods of fractal geometry developed during the second half of the 20th century, especially in terms of introducing the concept of a generalized dimension (Balatoni and Rényi, 1956; Rényi, 1959) which is particularly of interest here, were soon formulated in the context of chaotic dynamics (Hentschel and Procaccia, 1983; Grassberger, 1983), also incorporating the method of multifractal (or singularity) analysis later suggested by Halsey et al. (1986). The characteristics of intermittent turbulence on the other hand, in terms of describing the nature of energy dissipation in turbulent fluid flows through regions at various scales, were related to the properties of “fractals” during the 1960s (see e.g. Mandelbrot, 1999, §1.5 and references therein) and continued to be studied through the developing theory of multifractals. The use and applications of the above concepts and methods to space plasmas are given a nice review in Chang et al. (2006), where the authors describe a mechanism thought to be responsible for the onset of intermittent turbulence in astrophysical environments in a generic way, and also demonstrate the use of such techniques using a 2-dimensional MHD simulation for homogeneous turbulence.

The purpose of the short introduction given above is to make clear that the following brief
investigation is by no means an attempt to perform a simplified multifractal analysis. It is a mere attempt to obtain a number which can be called *fractal dimension* of the structure permeating the coronal volumes under study, with a certain degree of confidence and a somewhat limited amount of effort, given the restrictions imposed on the extracted volumes consisting our data. To this end, we will estimate this “fractal dimension” using two methods of numerical computation, namely the *box counting method* (Gagnepain and Roques-Carmes, 1986) and the method based on the (generalized) *correlation sum*, an algorithm proposed in its initial form by Takens (1982); Grassberger and Procaccia (1983b,a), also employing a maximum likelihood estimate for the *correlation dimension* (Grassberger and Procaccia, 1983b,a), the so-called *Takens-Theiler estimator* (Takens, 1985; Theiler, 1988).

Let $C$ denote the set of $N$ centroids $Q^C$ (equation 12) representing the clusters contained in our selected volumes (referred to as just “points” hereafter and denoted as $Q$) and assume the existence of a set $A \supset C$, presumably admitting the fractal topology we are attempting to quantify using $C$. If we partition $A$’s embedding (3D) space into boxes of size $\varepsilon$ and index those $N(\varepsilon)$ boxes covering $C$ denoting them as $B_i(\varepsilon), i = 1, 2, \ldots, N(\varepsilon)$, then by taking increasingly refined such partitions and determining $N(\varepsilon)$ for each $\varepsilon$ down to scales limited by the resolution of our data, we can estimate the *box counting dimension* of $C$ based on the definition

$$D_{BC} = \lim_{\varepsilon \to 0} \frac{\ln (1/N(\varepsilon))}{\ln \varepsilon}. \quad (13)$$

In practice, what one does is attempt to fit a straight line to the small-$\varepsilon$ regime of the log − log plot of $1/N(\varepsilon)$ versus $\varepsilon$ since, reaching the limit $\varepsilon \to 0$ is not feasible due to resolution constraints. This matter is discussed further below.

Now consider defining a ball $B_{Q^i}(r)$ of radius $r = \varepsilon/2$ centered at point $Q^i, i = 1, 2, \ldots, N$ and call $n_i(r)$ the number of points $Q^j$ other than $Q^i$, contained in $B_{Q^i}(r)$. This latter setting leads to the definition of the *generalized correlation sum* (Grassberger and Procaccia, 1983b,a)

$$\hat{C}_q(r) = \left[ \frac{1}{N} \sum_{i=1}^{N} \left( \frac{n_i(r)}{N-1} \right)^{q-1} \right]^{1/(q-1)}, \quad (14)$$

where generally $q \in \mathbb{Z}$ and

$$n_i(r) = \sum_{j=1, j \neq i}^{N} \Theta(r - |Q^i - Q^j|). \quad (15)$$

In the above equation $|\cdot|$ denotes the Euclidean distance between points $Q^i$ and $Q^j$, and $\Theta(Q)$ is the Heaviside function. Note that for $q = 1$ and letting $\hat{C}_1(r) = \lim_{q \to 1} \hat{C}_q(r)$, after applying
L’Hôpital’s rule one has
\[
\hat{C}_1(r) = \exp \left[ \frac{1}{N} \sum_{i=1}^{N} \ln \frac{n_i(r)}{N-1} \right].
\] (16)

The *generalized correlation dimension* is then defined as
\[
D_C^q = \lim_{r \to 0} \frac{\ln \hat{C}_q(r)}{\ln r}.
\] (17)

For \( q = 2 \), the above equations (14), (15) and (17) provide the initial definition of the *correlation dimension* \( D_{CD} \) (Grassberger and Procaccia, 1983b,a).

\( \hat{C}_q(r) \) is an unbiased estimator of the so-called *generalized correlation integral*, through which \( D_C^q \) is associated to the *Hentschel-Procaccia dimensions* (Hentschel and Procaccia, 1983) \( D_{HP}^q \) and/or the *generalized Rényi dimensions* \( D_R^q \) (Rényi, 1959, 1970). Establishing an equivalence relation between the latter two with mathematical rigor is a rather elaborate process, unlikely to be found in physicists literature, as pointed out in Barbaroux et al. (2001) where this issue is being treated in detail. This equivalence as well as the soundness of taking \( D_C^q \) as an approximation to either \( D_{HP}^q \) or \( D_R^q \), both strongly depend on the precise definition of an appropriate measure \( \mu \) on \( \mathcal{A} \) underlying all the relevant definitions, and on the \( q \)-value or regime one is interested in (see also Yakov, 1997, chapter 6).

The importance of establishing such relations can be seen by considering that \( D_R^q \) for \( q = 0 \) reduces to the box counting dimension \( D_{BC} \) (equation (13), see e.g. Barbaroux et al. (2001)), and for \( q = 1 \) it provides the definition of the *information dimension* \( D_{INF} \) (Balatoni and Rényi, 1956) (see also Ott, 1993; Barbaroux et al., 2001). \( D_C^q \) on the other hand, for \( q = 2 \), is easily seen to approximate both \( D_{HP}^2 \) and \( D_R^2 \) which in this case are also equivalent to one another (see Barbaroux et al. (2001) and also Ott (1993) for a Dynamics-oriented approach). It is therefore implied that, by bringing these definitions together, a unified definition of a *generalized “fractal” dimension* could be achieved. This need to provide the term “generalization” with its proper meaning is also complemented by the fact that \( D_C^q \), under certain conditions discussed below, provides a method of estimation very easily implemented in numerical algorithms.

Regardless of the definition of a generalized dimension employed however, this “\( D_q \)” will generally admit a spectrum of values according to its dependence on the parameter \( q \), in a graph known as the *spectrum of \( D_q \) dimensions* (Ott, 1993). \( D_q \) is a non-increasing function of \( q \) (satisfying the inequality \( D_q \geq D_p \), for \( q < p \) (Grassberger, 1983)) and roughly speaking, the “flatness” of the \( D_q \) curve within a specified range of \( q \)-values around zero is one of the features used to distinguish between non-, mono- and multi-fractal sets.

Picking up on the (rather rightfully served) comment regarding the unsubstantiated assumptions made by physicists regarding the equivalence relations discussed above, we make a note on certain facts relating to real data manipulation, which inhibit checking ones data in terms of
compliance with even one of the above definitions, let alone two or more (and then, on top of everything, test their equivalence).

The first thing to note relates to the assumption one is forced to make when handed a single set, like \( C \) in the current study. In order to define a measure \( \mu \) on \( A \), which is the underlying set admitting the properties one seeks to explore, \( C \) must indeed represent \( A \) in terms of its fractal properties, i.e. one must assume that \( C \) was somehow acquired in such a way as to ensure that any such selection would uniquely define \( \mu(C) \), that is assign it a single value independent of \( C \) [This point is also stressed in Hegger et al. (1999) by demanding that the points (in \( C \)) covered by the correlation sum be selected independently, according to an invariant measure on \( A \)].

The second note also relates to the measure \( \mu \), this time pointing to the finite size of \( C \). As equations (14) and (16) imply, for \( q \neq 2 \), \( n_i(r) \) is required to be non-zero for all points \( Q^i \in C \). Even ignoring the fact that this is a fundamental demand stemming from the need to attribute a “dimension” to a set (e.g. the support of a measure) rather than a measure (see Mandelbrot (1999) for a related commentary and also (Farmer, 1982)), so as to relax this condition for \( q = 2 \), for the rest of the \( q \) values we are bound to “stumble” on this “discontinuity” as we approach the small \( r \)-region, due to the finite size of \( C \).

Given the reduction our data was subjected to, leaving us with no means to verify a concise definition of a measure based on \( C \), or to avoid having \( n_i(r) \) vanish for some points even for \( r \sim 16 \), quite larger than the smallest interpoint distance \( r_{min} \sim 1.23 \) (pixel units), we proceeded in a manner displayed graphically in figure 5.

**Box Counting Dimension.**— First the box counting dimension was estimated, based on equation (13). The log – log plot of \( 1/N(\varepsilon) \) versus \( \varepsilon \) is shown in figure 5 in a blue straight line connecting the data points. As Theiler (1990) suggests, the finite size of \( C \) causes an underestimation of \( N(\varepsilon) \) thus increasing the slope of the line fitted on the data. As we approach the lower \( \varepsilon \)-regime the effect of \( C \)’s finite size becomes more profound braking down the scaling exhibited in greater \( \varepsilon \), since \( 1/N(\varepsilon) \) is bounded by the number of points \( N (\sim 4000) \).

**Correlation Dimension.**— The correlation sum \( \hat{C}_2(r) \) was next numerically computed as a function of \( r \), based on equations (14) and (15), for \( q = 2 \). As seen in the same figure 5 (red dotted line), \( \hat{C}_2(r) \) does indeed have twice the dynamic range of \( 1/N(\varepsilon) \) (Theiler, 1990), attempting to show its potential to probe smaller scales limited only by \( r_{min} \), but the scaling brakes down in this case as well, due to the finite size of \( C \).

One last issue worth discussing at this point is the range of \( r \) (or \( \varepsilon \)) over which one is to fit the data. According to (Theiler, 1987), this scaling region \( (r_{S,min}, r_{S,max}) \) should in principle exclude from the fit those regions where the expected scaling brakes down due to the finite size of \( C \) (and/or the finite resolution of the data) – lower end of \( r \), and the finite size of the fractal – upper end of \( r \) (Theiler, 1990). In the two sets of data in figure 5 discussed so far, one does
indeed expect to see $D_{BC} > D_{CD}$, as observed. That is, given an assumed equivalence relation between the generalized dimensions $D^{R}_q$, $D^{HP}_q$ and $D^{C}_q$ under a “unified” $D_q$ implying that since $q = 0 < q = 1 < q = 2$, it is expected that

\[ D_0 = D_{BC} \geq D_1 = D_{1NF} \geq D_2 = D_{CD}. \]

(18)

As to the appropriate range to perform the fitting however, this is not clear at all due to the slight variations in the scaling behavior which can however, affect the results quite substantially.

**Generalized Dimension in a Reduced Dataset.** In an attempt to verify the inequalities (18) and also try to obtain an estimate of the scaling region, we proceeded using equations (14) and (15), for $q = 0$, $q = 1$ and $q = 2$. In order to avoid the points $Q'$ for which $n_i(r) = 0$ however, those points with $n_i(2r_{min}) = 0$ were dropped leaving us with a number of clusters $\sim 3500$. The estimated dimensions for this new setting are denoted as $D_{bc}$, $D_{inf}$ and $D_{cd}$ for the $q$-values given above respectively. As can be seen in figure 5 by comparing the new set of data-points in red (down-triangles), green (squares) and blue (up-triangles) with the connected data previously described, this data reduction has not cost us to loose any of the scaling behavior in the small $r$-regime observed in the previous attempts. It has rather indicated the multifractal structure of the volumes examined, also suggesting an optimal scaling region for which the fittings can render converging values for the fractal dimensions expected to more or less coincide.

![Figure 5](image_url)

*Figure 5:* Estimation of a “fractal dimension”, using (i) the box counting algorithm (blue straight line connecting data) fitted by the dashed-dotted blue line leading to $D_{BC}$, (ii) the Grassberger-Proccacia algorithm for $q = 2$ (red dotted line connecting data) fitted by the dashed-dotted red line leading to $D_{CD}$, and (iii) the generalized correlation dimension approach for $q = 0, 1, 2$, by dropping all points $Q'$ with $n_i(2r_{min}) = 0$ (blue up-triangles, green squares and red down-triangles), fitted by the black dashed lines leading to $D_{bc}$, $D_{inf}$ and $D_{cd}$ respectively. The scaling range over which all fittings were performed is $\sim (1.46, 50)$ in pixel units, as determined by the procedure described (see text).
Takens-Theiler Estimator.-- In order to finalize our results we used the Takens-Theiler estimator, shown to be a maximum likelihood estimator of the correlation dimension $D_C^2$ (Takens, 1985; Theiler, 1988), defined as

$$\hat{D}_C^2 = -\frac{1}{\langle \ln(r/r_{S,max}) \rangle}, \quad (19)$$

where the angle brackets $\langle \cdot \rangle$ denote the average of the enclosed quantity, for all interpoint distances $r < r_{S,max}$ (see also Theiler, 1987, 1990).

The scaling region $(r_{S,min}, r_{S,max})$ was determined as the region over which the applied fittings, as well as the Takens-Theiler estimator using the appropriate $r_{S,max}$, provide the nearest possible values for the corresponding quantities, namely $D_{BC} - D_{hc}$ and $D_{CD} - D_{cd} - \hat{D}_C^2$. The inequalities (18) were satisfied without any extra adjustment. This was achieved for $r_{S,min} = 2r_{min} \sim 2.46$ and $r_{S,max} \sim 50$, as the fittings’ point of intersection roughly suggests (figure 5). For this particular $r_{S,max}$, $\hat{D}_C^2$ estimated $D_{CD}$ to be $\sim 1.7$ and $D_{cd} \sim 1.63$ (compare to the corresponding values given in figure 5). This procedure provided similar results in all three datasets available, so in figure 5 we present only those concerning frame 21:52 UT. As for the “fractal dimension” we set out to estimate, since there is no particular reason as to why any of the quantities acquired should be given more merit than the others, from this point on it is a matter of choice. We will therefore proceed by taking the middle-ground approach of using $D_{inf} \approx 1.72$.

3.2.2 Spatial Extent

Extracting information regarding the spatial extent of the dissipation regions after the data extraction performed in the previous paragraph 3.1, is a straightforward procedure. One can simply bin the number of points $n_C$ comprising each of the clusters recovered into a histogram representing the normalized PDF $P(n_C)$ or, equivalently, denote the volume of any of these clusters as $V_C$ and bin these volumes instead. $P(n_C)$ and the alternative PDF $P(V_C)$ can be seen to have the same form by noting that $V_C = n_C \Delta V$, where $\Delta V (\sim 0.474 \text{Mm}^3)$ is the volume of a single pixel in the 3-dimensional grid. Adopting the latter approach, $P(V_C)$ is shown in figure 6 to form a double power-law with an absolute index $\sim 3.47$ at the lower volume regime, corresponding to clusters with volumes less than $\sim 7 \text{Mm}^3$, and a power-law absolute index $\sim 1.59$ at the right tail of the distribution.

3.2.3 Dynamical Potency

The term “dynamical potency” of the reconnection sites permeating the coronal volumes under study used at the beginning of this paragraph, is meant to describe the energetic content of these regions, in terms of its availability to be dissipated through particle energization. We can therefore quantify this property by means of either the free magnetic energy $E_{free}$ per cluster or, as
Figure 6: Probability distribution function $P(V_C)$ (normalized to 1), of the volumes $V_C$ admitted by the reconnection sites recovered via the clustering process described in §3.1. The parameters of the double power-law (red dashed line) fitted on the data (red dashed line), are displayed on the plot.

it will actually be utilized in the next section, the average current density magnitude $\langle J \rangle$ per cluster. Both these alternatives can be presented in the form of PDFs, denoted as $P(E_{\text{free}})$ and $P(\langle J \rangle)$ respectively. The normalized histograms of $E_{\text{free}}$ and $\langle J \rangle$ are shown in figure 7, fitted by a double power-law and a log-normal distribution respectively (the parameters of the fits are displayed on the plots and are also discussed below). Notice that the right tail of $P(\langle J \rangle)$ (figure 7b) is better fitted by a power-law with an absolute index $\sim 2.95$.

As briefly mentioned in §3.1, the lower cutoff $E_{\text{free}}^{L,\text{cutoff}} \approx 10^{22}$ ergs imposed on the free magnetic energy per cluster was intended to include in our selected volumes, those less energetic events accounting for the nanoflaring hypothesis proposed in Parker (1988). As a baseline for establishing whether this goal was achieved, we can use the distribution of total counts in flares, as obtained from solar X-ray flare observations and/or the total energy distribution of flares simulated by a model based on the concept of Self-Organized Criticality (SOC) (Bak et al., 1987, 1988). A comparison between any of these PDFs with $P(E_{\text{free}})$ is based on an argument made in Vlahos and Georgoulis (2004), suggesting that, if an AR exhibits self-organized critical behavior, thus producing flares as a result of the nonlinear interaction of a large number of unstable volumes embedded in it, then the statistics of the energies released by a sample of flares should not be too different from that of the free energies in a random sample of such unstable volumes.

The two distributions mentioned above represent, from an observational and a theoretical model-wise point of view respectively, one of the characteristic flare parameters expressing the self similar nature of the dynamics underlying the evolution of solar ARs (along with the distributions of the peak count/luminosity and duration of events, they are considered the three most representative flare parameters). The former one, that is, the distribution of total counts in flares,
has been shown to form a power-law with an absolute index varying roughly in (1.39, 1.5) (see e.g. Georgoulis et al., 2001, for a summary of references related to observational evidence), suggesting that ARs exhibit a single scaling behavior. In the case of the simulated total energy distribution of flares however (the related models are discussed below), a double power-law is formed, suggesting a distinction between two populations of events, namely those that are less energetic (nanoflares), admitting an absolute power-law index $\gtrsim 1.85$ and those stronger events (microflares and flares) corresponding to the right tail of the double power-law, with an absolute index $\lesssim 1.62$ (see e.g. Vlahos et al., 1995; Georgoulis et al., 2001).

Models applying the concept of SOC in the context of evolving ARs and solar flares started

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure7.png}
\caption{Probability distribution functions (normalized to 1), of (a) the free magnetic energy $E_{\text{free}}$ (fitted by a Double power-law) and (b) the average current density magnitude $\langle J \rangle$ (fitted by a Log-normal distribution), of each dissipation region identified within the coronal volumes examined (see §3.1). The parameters of the fitted distribution functions (red dashed lines) fitted on the data (blue straight connecting lines), are shown on the respective plots.}
\end{figure}
developing in the early 1990s with Lu and Hamilton (1991), and attributed a double scale invariant behavior to ARs through a refined model of the so-called Statistical Flare proposed in Vlahos et al. (1995) and Georgoulis et al. (1995). In Georgoulis and Vlahos (1996) and Georgoulis and Vlahos (1998), the model-feature driving the system to the SOC state, a nonlinear driver simulating the instabilities triggered by the shuffling of the photospheric footpoints and the flux emergence from the convection zone, was used as a varying parameter in order to study the limits of adjustment admitted by the statistical flare model, in terms of the scaling indices of the flare parameters. Later on, Georgoulis et al. (2001) contributed to the comparison of the results obtained from SOC-based models against those dictated by the data already begun by Lu et al. (1993), both by modulating the statistical flare model towards reproducing the distributions obtained by the WATCH solar burst catalogue (see Georgoulis et al., 2001, for related references), and by dividing the observational (and simulated) data into subsets according to some size criteria, in an attempt to reveal (and compare) multiple underlying self-similarity rules that may compose the resulting single scale invariant behavior exhibited by the observational flare parameters.

These studies concluded that the statistical flare model is amenable to modulation towards reproducing the observed distributions, especially in the case of the total energy of flares, provided the comparison is made between the single index of the observed distribution and the flat index of the simulated one. The investigation relating to sub-grouping the data in an attempt to create two distinct populations of events exhibiting different scaling behaviors showed that, while this can be achieved for both observational and simulated data, several implications render this result inconclusive, at least as far as the form of the composite distribution is concerned (Georgoulis et al., 2001). The safest assertion this latter study lead to, is that, should these two different populations of events truly exist, they both exhibit scale invariant behavior expressed through distinct power-laws, with the nanoflaring events corresponding to steeper parts and the microflaring and flaring events to flatter ones.

Coming to the results presented in the current study so far, comparing the double power-law indices of \( P(E_{\text{free}}) \) given in figure 7a with the approximate ranges of indices based on both observations and simulations relating to the respective total counts and energy distributions given above, it becomes self-evident that, as has already been noted several times in the previous sections, our selected volumes do not effectively probe the lower free energy regime. While the higher-energy branch of the double power-law fitted in \( P(E_{\text{free}}) \) can be considered to reflect the moderate and highly energetic events’ scaling behavior, based on the above discussion it is obvious that the lower-energy branch admits too low an absolute index to correspond to nanoflaring events.

As additional evidence that the flatter part of \( P(E_{\text{free}}) \) should be neither considered credible or incorporated in the subsequent analysis, figure 8 shows \( E_{\text{free}} \) and \( \langle J \rangle \) per cluster, as these
quantities are distributed among clusters of different sizes (volumes \( V_C \)). Focusing on \( E_{\text{free}} \) versus \( V_C \) (red set of data) we can clearly see the upper cutoff set to the free magnetic energy per cluster (~ 10\(^{27}\) ergs), but the lower cutoff is not as evident due to the lack of sufficient data below ~ 10\(^{25}\) ergs (approximately where the scaling changes in figure 7a). Given the form of the PDF \( P(V_C) \) shown in figure 6 which, it should be noted that it is not altered when binning only those clusters with \( E_{\text{free}} \gtrsim 10^{25} \) ergs, we do expect to see a decreasing number of clusters towards higher volumes. This however, does not justify having them completely vanish above ~ 10Mm\(^3\) for free magnetic energies less than ~ 10\(^{25}\) ergs, and directly links to the difficulty caused by the specific set of data to the extrapolation technique, in performing efficiently on the pixels having low free energies (§2.4). Given the inadequacy presented in this set of data, a discussion concerning \( \langle J \rangle \) versus \( V_C \) (blue set of data in figure 8) losses its meaning.

![Graph](image)

**Figure 8:** A graphical illustration of the way \( E_{\text{free}} \) and \( \langle J \rangle \) are distributed over the entire range of clusters sizes, represented by the corresponding volumes \( V_C \) admitted by the clusters comprising the dataset in this study.

Noting that the analysis preceding the clustering process (section 2) was performed on a pixel-to-pixel basis, we expect that \( E_{\text{free, pixel}} \propto |J_{\text{pixel}}|^2 \), even though the force-free parameter \( \alpha(r) \) (stemming from the non-linear force-free condition \( \nabla \times B = \alpha(r)B, J = \left( c/4\pi \right) \nabla \times B \)) may vary from pixel to pixel. We can therefore approximately assume that \( E_{\text{free}} \propto \langle J \rangle^2 \) and, based on the rather well fitted power-law at the right tail of \( P(\langle J \rangle) \) (figure 7b), possibly dismiss the lower current magnitude part of \( P(\langle J \rangle) \) fitting a log-normal distribution (these matters are, however, further discussed in the following sections).
3.3 Final Remarks on Data Validity

As noted in §2.4, the data analysis presented in the current section used the volumes selected through validation of the extrapolations performed, by means of the free magnetic energy criterion of §2.3. Instead of subjecting the data to the validation checks described there, one might have chosen to simply use $E_{\text{free},2}$ (equation 6b) and proceed with the rest of the analysis taking the same steps, as described. In order to check the status of our results, we actually have gone through all these steps assuming full knowledge of the processed quantities over the entire computational domain, using $E_{\text{free},2}$. The (not so) remarkable fact is that almost all the results presented so far, have turned out to be affected insignificantly by this, otherwise negligent, approach.

The only result indicating that there might be something wrong with the extrapolated data, came from a cross-check we performed, regarding the clustering process. In the case making use of all the information available, one is forced to actually impose a threshold on $E_{\text{free}}$ before performing the clustering, as explained in §3.1. The clustering scheme obtained contains 3D pixels admitting a certain range of current density magnitudes $J$. One assumes that if the clustering process is performed once more, this time imposing a threshold (actually, two) on $J$, as specified by the above range of values, the clustering scheme rendered will roughly be the same [this is based on the argument presented right at the end of the last paragraph regarding relation $E_{\text{free, pixel}} \propto |J_{\text{pixel}}|^2$].

As it turns out however, this is not the case. The reason is that this data selection (based on $J$) allows for a much greater number of 3D pixels with $J \lesssim 10^{24} \text{statA.cm}^{-2}$, forming enormous clusters. These clusters can of course later be discarded, on grounds of exceeding the maximum free magnetic energy allowed per cluster $E_{\text{free,cutoff}}$, thus leading to the same $P(\langle J \rangle)$. Unfortunately however, the analysis never reaches this point, as the clustering algorithm is forced to group and record nearly half the computational domain as one single cluster. It therefore takes it virtually “forever” to run [note that this inconsistency, accompanied by the computational inefficacy just discussed, has not presented itself in the kind of analysis chosen to pursue during this study].

One is then forced to repeat the process and apply a more stringent threshold on $E_{\text{free}}$ (e.g. move it towards $10^{24} - 10^{25}$ ergs). This of course eliminates all problems, including those the free magnetic energy criterion was used to treat, albeit in a less systematic way. It is worth noting that having discarded all lower energy pixels right from the beginning leads approximately to the same fractal dimension (§3.2.1), $P(V_C)$ (figure 6) also remains the same, and both $P(E_{\text{free}})$ and $P(\langle J \rangle)$ are rendered as single power laws approximately identical to the right tails of the PDFs shown in figure 7. It is therefore in this sense, that the results were said to be affected insignificantly by this “easier” approach; we end up working on approximately the same dataset, restricted to the higher free magnetic energy (microflare/flare) regime, only, we have no idea why.
4 Model Overview

The part of this study presented so far was exclusively devoted to bringing raw magnetograph data into a form susceptible to a nonlinear force-free extrapolation, in order to reconstruct the 3-dimensional magnetic field occupying the coronal volume immediately above an observed active region. Setting the various validation tests this magnetic field was subjected to aside, the ultimate purpose of this reconstruction was to reveal those regions within the volumes examined, in which a variety of magnetic discontinuities, not quite identified in terms of their cause and origin, are collectively expressed through charged particle energization, as a means to dissipate the excess magnetic energy accumulated within them.

These dissipation regions were subsequently studied in terms of the statistical aspects of certain properties they exhibit. The picture drawn as a result of this analysis, is that of a complex structure displaying certain characteristics commonly attributed to multifractals, comprised of small-scale regions admitting heavy tailed distributions in both their spatial extent and dynamical potency (figures 6 and 7). Figure 9 below graphically illustrates a zoomed-in area of the overall structure deduced and analyzed during the first part of this study, showing both the unstable areas (clusters) and the magnetic field lines permeating them.

A charged particle moving in such an environment is bound to trace the magnetic field lines, and through their complex configuration, occasionally enter one of those sites comprising the permeating structure discussed. The nature of the interaction between a charged particle and any such region it penetrates, as well as the temporal scale of this interaction and the amount of such occurrences within a specified time-frame, will determine, energy wise, the final state of the particle. Turning into a large population of such particles one can then examine this fragmented environment in terms of its diffusive properties, as well as its efficiency as a particle accelerator.

Investigating particle heating and/or acceleration in the kind of intermittent environment the above picture depicts, is the final purpose of this study. The model employed in order to introduce and explore particle dynamics inside such an environment was proposed by Vlahos et al. (2004). The main difference between that study and the work presented here, is the fact that in Vlahos et al. (2004), the basic model elements (discussed below) are either based on MHD-consistent cellular automaton simulations, or simply inferred based on well substantiated arguments, both of which have been pursued with respect to observations. As observational data and data processing techniques became more available since, the same basic model elements in this study are all data driven. A few other differences between the two approaches will be discussed as we explore the specifics of the modeling process in the following paragraphs.
Figure 9: Graphical illustration of the magnetic configuration above an AR, as inferred using the tools described and developed in the first part of this study. The color shaded areas represent the unstable regions produced as nearby magnetic field gradients increase exceeding some threshold. Charged particles penetrate those regions tracing the magnetic field lines (shown in green), giving rise to events during which the excess magnetic energy is dissipated through particle acceleration.

4.1 Basic Model Elements:

The Environment

A charged particle finds itself at some initial energetic state, gyrating about a magnetic field line tracing it towards some of the reconnection sites lying ahead. The processes set in motion once this particle enters that region are collectively represented by a DC electric field acting upon the particle. We therefore treat these regions as “black boxes”, as in Vlahos et al. (2004), regardless of the mechanism(s) influencing the nature of the interaction taking place. At the end of this interaction the particle finds itself in a new energetic state, traveling free again, tracing magnetic field lines until the next encounter.

We wish to turn the information acquired in the previous section 3.2, regarding the fractal dimension of the structure hosting these unstable volumes and the PDFs admitted by the volumes $V_C$ of the clusters recovered and the average magnitude of the current density $\langle J \rangle$ they carry, into three separate PDFs:
(i) One representing the probability of a free-traveling particle to travel a distance $s$ before running into its next encounter,

(ii) one providing the probability to travel some distance $\ell$ while interacting with this unstable volume, and

(iii) another one to provide us with the probability to penetrate a region which will act on it by means of an effective electric field of strength $E$.

With the exception relating to PDF (i) (see below), the other two PDFs are rather straightforwardly derived based on the respective PDFs used to fit the data in section 3.2 (figures 6 and 7). Note however, that PDF’s (iii) complete definition requires several strong assumptions discussed in detail in the respective paragraph 4.1.3.

### 4.1.1 Free Travel Distances

For a particle found at some arbitrary point within a volume permeated by a structure exhibiting fractal properties, moving freely in some random direction, the probability to travel a distance $s$ before meeting some element of the structure was analytically derived in Isliker and Vlahos (2003). They found that the form of the PDF according to which those distances $s$ are distributed is mostly determined by the fractal dimension $D_F$ of the environment structure. For the case in which $D_F < 2$, it has been shown that this PDF is an approximate power-law with an index $D_F - 3$.

Adopting this approach also employed in Vlahos et al. (2004), we define the normalized PDF based on which, each of the particles contained in our sample population gets assigned the free travel distance $s$ following each encounter, as

$$ P(s) = \frac{D_F - 2}{s_{max}^{D_F-2} - s_{min}^{D_F-2}} s^{D_F-3}, \quad s_{min} \leq s \leq s_{max} \tag{20} $$

where $D_F \equiv D_{inf} \simeq 1.72$ (§3.2.1), $s_{min} \simeq 1.58 \times 10^8\text{cm}$ and $s_{max} \simeq 2.27 \times 10^{10}\text{cm}$ are the minimum and maximum interpoint distances admitted by the data respectively, as transformed from pixel units to centimeters, given the pixel linear dimension 0.78Mm (the cluster centroids will continue to be referred to as points throughout). The method for producing numbers randomly distributed according to a power-law is described in Appendix E, §E.4.
4.1.2 Acceleration Lengths

In order to assign some form of linear dimension to the clusters recovered and obtain an estimate of the distances (expected to be) traveled by the particles inside the dissipation regions, we take the most conservative approach in assuming that the \( n_C \) 3D pixels a cluster consists of are distributed spherically around, say, its centroid. Letting \( r \) denote the radius of a cluster, its volume will be \( V_C = (4\pi/3)r^3 \) and its diameter \( \ell \) will be given as

\[
\ell = 2\left(\frac{3V_C}{4\pi}\right)^{1/3}.
\]  

(21)

\( P(V_C) \) of figure 6 is of the form

\[
P(V_C) = \begin{cases} 
AV_C^{\alpha_1}, & V_C^{\min} \leq V_C \leq V_C^{\th} \\
A(V_C^{\th})^{\alpha_2-\alpha_1} V_C^{\alpha_2}, & V_C^{\th} \leq V_C \leq V_C^{\max},
\end{cases}
\]

(22)

where \( A \) is the normalization constant, \( \alpha_1 \simeq 3.47 \) and \( \alpha_2 \simeq 1.59 \) are the absolute double power-law exponents, and \( V_C^{\min} \simeq 0.51, V_C^{\th} \simeq 7.15 \) and \( V_C^{\max} \simeq 24.03 \) in \( \text{Mm}^3 \) determine the ranges of the scaling rules seen in figure 6. Given equation (21), the PDF according to which the cluster diameters \( \ell \) are distributed, interpreted here as the mean acceleration lengths of the particle interactions, will be given as

\[
P(\ell) = \begin{cases} 
B\ell^{-\beta_1}, & \ell_{\min} \leq \ell \leq \ell_{\th} \\
B\ell_{\th}^{\beta_2-\beta_1} \ell^{-\beta_2}, & \ell_{\th} \leq \ell \leq \ell_{\max},
\end{cases}
\]

(23)

where \( B \) is again the normalization constant and \( \beta_1 = 3\alpha_1 - 2, \beta_2 = 3\alpha_2 - 2 \) the absolute double power-law exponents obtained during the derivation of \( P(\ell) \) based on the condition \( P(V_C) dV_C = P(\ell) d\ell \) required by the conservation of probability. Using equation (21) and the values of \( V_C^{\min}, V_C^{\th} \) and \( V_C^{\max} \) given above, we estimate \( \ell_{\min} \simeq 0.99, \ell_{\th} \simeq 2.39 \) and \( \ell_{\max} \simeq 3.58 \), in \( \text{Mm} \). Note that we avoid writing the explicit form of \( B \), as it is a rather cumbersome function of the parameters already given. The method for producing numbers randomly distributed according to a double power-law is described in Appendix E, §E.5.
4.1.3 Electric Field Strength

The form of the distribution $P(E)$ of the effective electric fields encountered by the particles is seemingly the easiest one to derive since, Ohm’s law allows us to define $E$ in terms of the mean magnitude of the current density $\langle J \rangle$ of each dissipation region as

$$E = \eta \langle J \rangle,$$  \hspace{1cm} (24)

where $\eta$ is the plasma resistivity. Assuming $\eta$ is constant and taking into account only the power-law tail of $P(\langle J \rangle)$ shown in figure 7b, $P(E)$ can be written as

$$P(E) = (a - 1) E_{\text{min}}^{a-1} E^{-a}, \quad E_{\text{min}} \leq E,$$  \hspace{1cm} (25)

where $a = 2.95$, taken from the power-law fit of figure 7b. Note that we have only assumed for $E$ the lower bound $E_{\text{min}}$, thus letting it extend to arbitrarily high values. The apparent issue rising from the necessity to actually define a PDF $P(E)$ for the vector $E$ of the effective electric fields, since the particle model built is 3-dimensional, is addressed by first determining $E$’s magnitude according to $P(E)$, and then splitting it into three components $E_x, E_y, E_z$ such that $E$ has a random direction in 3D space. The method implementing this feature is described in Appendix E, §E.3, and in §E.4 the method for creating random numbers distributed according to a power-law is described.

A power-law of the form of equation (25) has only well defined moments of order $m$ when $a - 1 > m$. This means that $P(E)$ can only be normalized ($m = 0$) for $a > 1$, and have a finite mean $\langle E \rangle$ ($m = 1$) and variance $\langle E^2 \rangle$ ($m = 2$) for $a > 2$ and $a > 3$ respectively, the latter two given in accordance by equations

$$\langle E \rangle = \frac{a - 1}{a - 2} E_{\text{min}}, \quad a > 2$$ \hspace{1cm} (26a)

$$\langle E^2 \rangle = \frac{a - 1}{a - 3} E_{\text{min}}, \quad a > 3.$$ \hspace{1cm} (26b)

For $a = 2.95$, the normalized $P(E)$ will have a finite mean, but its variance will diverge. We note here once more, that this complication could have easily been avoided, had we assigned $E$ an upper bound $E_{\text{max}}$ as well.

As far as our choice to dismiss the part of $P(\langle J \rangle)$ fitted by a lognormal distribution (figure 7b) is concerned, this can immediately be justified by recalling the discussion and evidence presented in paragraph 3.2.3, showing the inability of our dataset to effectively probe the regime of those less energetic events that would allow us to officially incorporate a nanoflaring scenario into our study. Our second choice, to only bound $E$ from below, gives rise to a discussion relating to an assumption all models on particle acceleration are forced to make, regarding plasma resistivity.
Plasma Resistivity and the Runaway Effect.— The effects and importance of plasma resistivity in any model on Joule heating and/or particle acceleration can best be described in a complete and simple manner on the basis of Dreicer’s study of the runaway effect (Dreicer, 1959, 1960). Dreicer (1959) assumed a fully ionized gas in the presence of a magnetic field \(B\), acted upon by an electric field \(E\) (both fields can be assumed uniform and stationary without any loss of generality in the notions we are interested to introduce here), and derived the momentum equations of the two-fluid (electron-ion) magnetohydrodynamics (MHD) from the Boltzmann equations for both particle populations, with the collision terms given by the Fokker-Planck equation. The basic features of this set of equations serving the purpose of this brief introduction are best summarized in a few sentences given below:

- Most of the energy given to the system by the electric field is channeled to the electron fluid via drift (electron currents), due to the electron-ion mass ratio.

- The dominant collisional effect is that of electron-ion encounters and is responsible for randomizing the “drift energy” of electrons, thus converting it to “thermal energy”.

- The overall effect of such configuration on the electron fluid is the change in momentum caused by the dynamical interplay between the electric force and the frictional force rising through electron-ion collisions.

Dreicer’s basic assumption was that Joule heating results after like-particle collisions (electrons-electrons) redistribute that “thermal energy”, establishing a Maxwellian distribution for the electrons, displaced by their instantaneous drift velocity relative to the ions (Dreicer also proceeded assuming an ion temperature \(T_i = 0\)). Considering this displaced Maxwellian as the solution to the Boltzmann equation for the electrons, Dreicer was able to then complete the calculation of all the unknown terms [Rosenbluth potentials (Rosenbluth et al., 1957), (see Dreicer, 1959)] entering the respective momentum equation, thus obtaining its final form

\[ m \frac{\partial v_e}{\partial t} = eE - eE_D \Psi(z), \]  

(27)

where \(v_e\) is the electron drift velocity, \(m\) and \(e\) the electron mass and elementary charge respectively, \(E_D\) the so-called Dreicer field (discussed below), and \(\Psi\) a function of a variable \(z\) varying with electron temperature \(T_e\) and drift velocity as \(z \propto T_e^{-1/2} v_e\). The first term on the right hand side of equation (27) represents the electric force acted upon the electron fluid, while the second one gives the frictional force due to electron-ion collisions, thus justifying calling \(\Psi\) the dynamical friction function. Its functional form (figure 10) plays and important role in the theory of runaway particles and is briefly discussed below.

The Dreicer field \(E_D\) defined in Dreicer (1959) and the Spitzer resistivity \(\eta_s\) derived in
(Spitzer and Härm, 1953), are given in SI units, by the following equations

\[ E_D = \frac{ne^3 \ln \Lambda}{4\pi\varepsilon_0^2k_B T_e} \]  

(28)

and

\[ \eta_S = \frac{e^2 m^{1/2} \ln \Lambda}{4\pi\varepsilon_0^2(\kappa_B T_e)^{3/2}} \]  

(29)

respectively, where \( \varepsilon_0 \) is the vacuum permittivity, \( k_B \) the Boltzmann constant, \( n \) the electron number density, and \( \ln \Lambda \) the Coulomb logarithm (dependent on \( n \) and \( T_e \) – ranging for most practical applications between 10–20). From the above equations (28) and (29), the Dreicer field can be written in terms of its dependence on plasma resistivity, in the form

\[ E_D = ne\sqrt{\frac{k_B T_e}{m}} \eta_S. \]  

(30)

Assuming an electric field the existence of which is inferred by the the presence of some current density of magnitude \( J \), as is roughly the case studied through the model employed here, a heuristic form of equation (27), using Ohm’s law and equation (30), is

\[ m \frac{\partial v_e}{\partial t} = e\eta_S J - ne^2 \frac{k_B T_e}{m} \eta_S \Psi(z). \]  

(31)

Without bothering at this point with the balance established between the electric force and the frictional force at every given instance in time, the above equation states that achieving an “anomalously” high plasma resistivity would enhance both forces acting upon the electron fluid and, most importantly, the relevant rates of “drift” energy gain and its subsequent conversion to “thermal” energy. This is the weakest point any model of particle acceleration faces to date; no model predicts the anomalous resistivity necessary to provide a particle accelerator with the efficiency expected in terms of rates of energy transfer.

Turning to the effect of runaway particles, Dreicer (1959) avoided the steady state assumption made in all previous work done regarding this effect. Up to that point, the main assumption was that no matter how strong the electric field acting upon the electron fluid is, the frictional force will increase with electron drift, pushing \( v_e \) towards a terminal value. Proceeding with the analysis briefly outlined above, Dreicer (1959) showed that the dynamical friction function’s dependence to \( v_e \) shown in figure 10, gives rise to an entirely different electron fluid response to the applied electric field.
We now briefly summarize the basic conclusions Dreicer’s study lead to:

- There exists a critical value $E_D \propto n T_e^{-1}$ of the electric field $E$, roughly determining whether particles attaining drift velocities exceeding twice the (bulk) thermal velocity in one collision time on average, will appear or not. Assuming a constant temperature, the two following conditions may apply:
  
  1. $E \frac{E}{E_D} < \Psi_{max}$, in which case $v_e$ reaches a terminal value according to what was generally expected before $E_D$ was introduced\(^1\).
  
  2. $E \frac{E}{E_D} > \Psi_{max}$. This case corresponds to the “runaway” effect; the drift velocity of the particles increases without a bound.

- The Dreicer field’s dependence on the electron temperature $T_e$ renders it a dynamical parameter. Even when the initial condition applies to case 1. above, the increasing temperature will cause $E_D$ to decrease, possibly flipping the balance from condition 1. to condition 2.

- An immediate consequence of $E_D$’s dependence on $v_e$ is that particle populations with velocities belonging to the bulk versus the tail part of an initial Maxwellian distribution can perceive the same electric field $E$ as satisfying above conditions 1. or 2. respectively.

The above discussion has hopefully highlighted certain implications involved in the choices made regarding the inference of $P(E)$ from the data driven $P(\langle J \rangle)$. In order to become more specific, one first thing to note is that the model implemented for this study does not incorporate collisions of any kind. Given this constraint, the safest course of action is to realize a sub-Dreicer model, restricting at least the majority of the effective electric fields assigned to the

\(^1\)Wikipedia does not have an entry on H. Dreicer (!)
encounters the particles experience, to values less than $E_D$. The general assumption around sub-Dreicer models is that runaway does not, on average, even take place, while the Joule heating ceases soon after a moderate rate of occurrence to begin with, since such models imply a plasma resistivity close to $\eta_S$ (note that the Spitzer resistivity $\eta_S$ is almost by definition the resistivity prohibiting any kind of efficient heating to take place since, it was derived under the steady state assumption mentioned above). Whatever the results of a sub-Dreicer model employed are, therefore, collisions are highly unlikely to have an effect on them.

The choice of a sub-Dreicer model is far from being justified, based on an argument saying that our model does not incorporate collisions. Remaining as close as possible to the only model existing on plasma resistivity however, which, renders sub-Dreicer fields for most ranges of current densities admitted by observational data, certainly gives more merit to such a choice.

Even though the mechanisms responsible for the onset of anomalous resistivity are not known, it is generally accepted that anomalous resistivity must occur to a certain extent, thus facilitating the enhancement of energy transfer rates. It is within this frame we allow the power-law tail of $P(E)$ to extend to arbitrarily high $E$-values, thus incorporating the anomalous resistivity scenario to a moderate degree (also adjustable/accessible through a parametric study, see §5.1). Note that this choice minimizes, at the same time, the severeness of the arbitrariness involved in any specification of either one of the interchangeably “free” parameters $\langle E \rangle$ and $E_{\text{min}}$ of $P(E)$ (equations 25 and 26).

Figure 11 indicates within the red rectangle on top of the well known SkyLab figure showing the solar atmosphere’s temperature and density dependence on height above the photosphere, the range of temperature increase we will attempt to simulate using the particle model employed. Taking the electron number density in the solar lower atmosphere and corona to be approximately $n \approx 10^{10} \text{cm}^{-3}$ and starting with an electron temperature of $T_{e,0} = 10^5 \text{K}$, the Coulomb logarithm turns out to be $\ln \Lambda \approx 15$. The Dreicer field according to equation 28 given the above specifications is $E_D \approx 0.459 \text{V.m}^{-1}$, and the Spitzer resistivity (equation 29) $\eta_S \approx 2.326 \times 10^{-4} \text{Ohm.m}$. The parameter chosen to determine the range of the effective electric fields admitted by PDF $P(E)$ is the mean value $\langle E \rangle$, and was set approximately equal to $10^{-4}E_D$, i.e. nearly two orders of magnitude greater than the mean value the Spitzer resistivity would render for $\langle \langle J \rangle \rangle \sim 1000 \text{statA.cm}^{-3}$ using equation $\langle E \rangle = \eta_S \langle \langle J \rangle \rangle$. The lower bound $E_{\text{min}}$ according to equation 26a for $a = 2.95$ is therefore $E_{\text{min}} \approx 5 \times 10^{-5}E_D$. Note that an approximate estimation of the order of magnitude the classical Spitzer resistivity is expected to be enhanced in order to attribute a variety of models with the properties of an efficient accelerator is more than six orders of magnitude (see e.g. Marsch, 2006, for a recent review).
4.2 Charged Particle Dynamics

The previous paragraph was devoted to an inference regarding the three PDFs determining the dynamical properties of the interactions taking place between a test particle population and the unstable regions dissipating their excess magnetic energy via reconnection, based on the analysis and data mining performed in previous sections. In the present paragraph we wish to incorporate those three PDFs $P(s)$, $P(\ell)$ and $P(E)$ given in equations (20), (23) and (25) respectively, into a quantitative study of particle dynamics, i.e. derive the equations of motion dictating it.
The overall spatial setup of our model is graphically represented in figure 12. The coronal loop with two footpoints of opposite polarity based on the photosphere is simulated by a straight cylinder of height \( L \simeq 2.5 \times 10^{10} \text{cm} \) and radius \( R \simeq 1.2 \times 10^{10} \text{cm} \). The magnetic field is assumed to be uniform and stationary, directed upwards along the \( z \)-axis of a coordinate system we let it define. Its magnitude is \( B = 100 \text{G} \). The test-particle population (500,000 electrons) is initially randomly distributed over the entire cylinder, with velocities following a Maxwellian distribution of temperature \( T = 10^5 \text{K} \). The “virtual” dissipation regions also occupy the whole cylinder, in the sense that no restriction is imposed on the three PDFs assigning \( s \), \( \ell \) and \( E \) to an electron, depending on its precise position inside the volume. The only spatial condition imposed is that, should a particle find itself somehow outside the cylinder, its dynamical state gets recorded and iteration stops.

The actual equations of motion one would be forced to use had the magnetic field employed comprised the complex structure shown in figure 9, would either be based on the Lorentz force, or the guiding center approach. Computational considerations relating to the large particle population one must include in order to obtain sufficiently good statistics, especially when studying the diffusive behavior of particles, tend to make the guiding center approximation a more viable choice, due to their far greater tolerance in terms of the time step of the integration. The general form of the relativistic guiding center equations of motion, as derived in (Grebogi and Littlejohn, 1984) and also in (Tao and Chan, 2007), adopting the notation used in (Hamamatsu et al., 2007), are given in cgs units in Appendix D.1.1. What interests us here is the form these equations take, in the particular case studied, in which \( B = B \hat{e}_z \). In this case, they read

\[
\frac{dr}{dt} = \frac{u_\parallel}{\gamma} \frac{B}{B} - \frac{c}{B^2} B \times E, \tag{32a}
\]

\[
\frac{du_\parallel}{dt} = -\frac{e}{mB} B \cdot E, \tag{32b}
\]

\[
\frac{du_\perp}{dt} = 0, \tag{32c}
\]

where \( r \) is the position vector of an electron, \( u_\parallel \) and \( u_\perp \) its parallel and perpendicular to the magnetic field \( B \) velocity components, such that \( u = u_\parallel \hat{e}_\parallel + u_\perp \hat{e}_\perp \), and \( E \) is the electric field. Note that \( u = \gamma \nu \) is the four-velocity vector with \( \nu \) the regular velocity of the particle in the
3-dimensional space, and $\gamma$ the Lorentz factor given by

$$\gamma(u) = \sqrt{1 + \frac{u^2}{c^2}} = \gamma(v) = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}}.$$  \hspace{1cm} (33)

Equations (32) are significantly simpler than the original set (61). They can of course be solved numerically, but their analytical solution can also easily be obtained.

In Appendix D.1.2.1, a kind of discretization of the above equations is discussed, in the context of a certain group of models called Continuous-Time Random Walk (CTRW) models. We briefly describe a typical sequence of occurrences during a particle trajectory, in order to outline the process of discretization through which a CTRW model of the highest possible complexity, yet extremely simple, since it does not involve correlations between its components, is setup.

**FREE TRAVELS.–** Consider a particle starting one of its free travels at an iterative step indexed $i$, with some initial (four-) velocity $u_i$. The PDF $P(s)$ assigns this particle some distance $s_i$ to travel, by the end of which it will encounter an unstable volume carrying an effective electric field of some random direction. Setting $E = 0$ in the above equations (32) and splitting equation (32a) into three component equations, these can immediately be solved by noting that the free-time $\tau_i = s_i/u_i\parallel$, since both $x$ and $y$ remain unaltered. The iterative scheme realizing each particle’s free travels is summarized in the set of recurrent equations (mapping) (34).

**Safety Measure 1: Consistency of Particle Exit.–** The extra precautions required to ensure the consistency of the model relate to the possibility of a free travel leading a particle outside the computational domain, far below (above) the bottom (top) base of the cylinder. Even though a situation like that in the case of the free travel will not affect the energetics of the particles, it can become dangerous in the case of the acceleration travels. Therefore since, it has to be treated for the latter case, it is only a matter of elegance that the treatment be applied in a unified manner, for both free and acceleration travels.

As soon as the position $x_{i+1}, y_{i+1}, z_{i+1}$ of the particle after its free (acceleration) travel is determined, a check is performed to see whether the particle has exited the cylinder. In case it has, its distance from the bases of the cylinder is computed and if it exceeds a pre-specified tolerance value, the iterative step is repeated under the same conditions, but this time the free (acceleration) distance assigned to it is the distance between the particle’s initial position and the cylinder base from which it exits. The respective time interval is computed based on this new travel distance assignment and the particle time is updated accordingly. Given that this treatment introduces a bias in both $P(s)$ and $P(\ell)$ (important in the latter case), it has been checked and verified that this effect is vanishingly small. It is certainly preferable against the two alternatives; to either include particles that have been accelerated on their way to... Mars, or to down-scale all...
assigned distances by one or more orders of magnitude, to generally minimize such occurrences.

**ACCELERATION TRAVELS.**– The \((i + 1)\)-th iterative step of the particle corresponding to an electric field encounter, is determined by \(\mathbf{u}_{i+1} = \mathbf{u}_i\), the distance \(\ell_i\) assigned to its acceleration travel by \(P(\ell)\), and the electric field \(\mathbf{E}_i\) of magnitude \(E_i\) determined by the PDF \(P(E)\), comprised of the random components \(E_{ix}, E_{iy}, E_{iz}\) obtained as described in paragraph 4.1.3 and Appendix E, §E.3. We will from now on denote this PDF as \(P(E)\), implying the procedure described there.

The system of equations (32) with \(E = E_i \neq 0\) can be integrated producing the set of recurrent relations (35), this time involving a serious implication. The guiding center equations of motion are a bit more complicated, and the fact that in this case all spatial coordinates of the particles change (see equations 35c,d,e) forces us to set \(\ell_i^2 = (\Delta x_i)^2 + (\Delta y_i)^2 + (\Delta z_i)^2\) (compare to setting \(s_i = \Delta z_i\) in the case of the free travels). After all necessary substitutions, this equation becomes a quadric in \(\Delta t_i\), the acceleration time (see equation 35g).

**Safety Measure 2: Numerical Instability in the Computation of \(\Delta t\).**– Solving the above forth-order polynomial in \(\Delta t_i\) turns out to be quite a challenge, despite having an analytical solution to work with. As explained in Appendix D, §D.1.2.2, a numerical instability rises when, for example, a highly energetic particle is assigned too small a distance \(\ell_i\) to travel, accelerated by a strong electric field \(E_i\). In such cases, the coefficients of the quadric become asymmetric, in the sense that their values range over entirely different scales, creating as a result, extreme coefficient ratios in the terms involved in the analytic solution. A more intuitive way to describe this situation, is to simply state that the quadric exhibits extreme slopes (derivatives) around its zeros (solutions to \(\Delta t_i\)). This is actually also the solution to this problem, also known as deep/shallow zeros (see e.g. Protopopov, 2007). The acceleration times can be safely and accurately (within a specified tolerance) obtained, by use of a routine implementing a method that does not use derivatives.

**Safety Measure 3: Velocity Reversal.**– Except from the extra precaution taken to ensure that particles exit the computational domain in a consistent manner (described above), in the case of the acceleration travels another safety measure is necessary. This relates to the cases where a particle enters an unstable volume acted upon by an electric field opposing the particle’s velocity (deceleration). In some of these cases, forcing the particle to travel the distance \(\ell_i\) initially assigned might cause it to accelerate towards the direction of the electric field when, in fact, its velocity should at best be reversed with no change in magnitude, if not just decrease significantly. Such cases require special treatment only when the encounters in question result in a flip in velocity direction. If they do, then the simplest approach is taken, under the assumption that the particles traverse the unstable volumes along the \(z\)-axis (a rather safe assumption, considering
that the particles diffuse predominantly in the $z$ direction and almost not at all in the $xy$ plane, compare equations 35c,d,e). If a velocity flip is detected at the end of an encounter, that iterative step is repeated with no change in position (the particle exits the region from the exact point of entrance) and a velocity reversal. The relevant time interval is computed and the particle time is update accordingly.

**An Implication Relating to the CTRW Model: Particle Timing.**—One last issue relating to the special circumstances rising from the CTRW model, gives rise to the notion of particle timing. In a typical numerical integration of the original guiding center equations of motion, the actual time (in seconds) of every event taking place as integration proceeds, is defined in a unified manner for all particles, regardless of whether the routine used to numerically solve the equations of motion uses an adjustable step-size of integration or not. What this means, is that there is a direct correspondence between the actual time and the counter advancing the iteration scheme, applicable to all particles in a unified manner. In the case of the CTRW model described in this paragraph, however, actual time does not correspond in a direct and straightforward manner to the counter of iteration; at least not in a way that can be applied to all particles in a unified manner. In other words, the actual time $t_i$ of a particle at any given iterative step $i$ is entirely different for each particle, as a consequence of the strong dependence of both $\tau_i$ and $\Delta t_i$ on the conditions under which each free travel as well as each acceleration travel take place.

Collecting time series for quantities of interest becomes a challenging task, due to this peculiarity presented by the CTRW model. This issue was addressed by dividing the total time of integration into a number of $N_t$ sub-intervals, such that each interval would represent a given instant in time. Recording the temporal evolution of certain quantities of interest was thus done in a “collective” manner, referring separately to each of the time sub-intervals.

Another problem caused by that “particle timing” issue, was that certain conditions can cause the particles to travel way forth in time, that is, find themselves at an instant $t_i$ that far exceeds the total time of integration $t_{fin}$. This is a similar problem than the one regarding the consistency of particle exit described above, and was therefore treated in a analogous manner.
RELATIVISTIC GUIDING CENTER: The CTRW Mapping

FREE TRAVEL:

\[ u_{i+1}^\parallel = u_i^\parallel, \quad (34a) \]
\[ u_{i+1}^\perp = u_i^\perp, \quad (34b) \]
\[ x_{i+1} = x_i, \quad (34c) \]
\[ y_{i+1} = y_i, \quad (34d) \]
\[ z_{i+1} = z_i + \frac{u_i^\parallel}{|u_i^\parallel|} s_i, \quad (34e) \]
\[ t_{i+1} = t_i + \tau_i, \quad (34f) \]
\[ \tau_i = \frac{s_i}{u_i^\parallel} \quad (34g) \]

ACCELERATION TRAVEL:

\[ u_{i+1}^\parallel = u_i^\parallel + \frac{q}{m} E_i^z \Delta t_i, \quad (35a) \]
\[ u_{i+1}^\perp = u_i^\perp, \quad (35b) \]
\[ x_{i+1} = x_i + \frac{c}{B_0} E_i^y \Delta t_i, \quad (35c) \]
\[ y_{i+1} = y_i - \frac{c}{B_0} E_i^z \Delta t_i, \quad (35d) \]
\[ z_{i+1} = z_i + \frac{c^2}{m E_i^z} (\gamma_{i+1}(\Delta t_i) - \gamma_i), \quad (35e) \]
\[ t_{i+1} = t_i + \Delta t_i, \quad (35f) \]

\[ \ell_i^2 = \left( \frac{c}{B_0} \right)^2 \left[ (E_i^x)^2 + (E_i^y)^2 \right] (\Delta t_i)^2 \]
\[ + \left( \frac{c^2}{m E_i^z} \right)^2 (\gamma_{i+1}(\Delta t_i) - \gamma_i), \quad (35g) \]

where \( \gamma_{i+1}(\Delta t_i) = \sqrt{1 + \frac{(u_i^\parallel)^2 + (u_i^\perp)^2}{c^2}} \quad (35h) \)
5 Particle Simulations: Results and Discussion

Particle simulations were performed using the program (Appendix D) developed to implement the model described in section 4, incorporating into its framework data specified features based on the data processing and analysis reported in sections 2 and 3. Several parameters necessary to fully determine the problem were provided and argued for in the preceding parts of this study. In this section we present the results of a group of three particle simulations, centered around the data derived properties of the environment bringing together several statistical aspects of the complex magnetic configuration above a solar active region.

5.1 A Parametric Study

Exploiting the broad limits of adjustment allowed by certain constraints admitted by the data processed, we performed a short parametric study focusing on the model parameter appearing to strongly affect in a meaningful manner, the way charged particles respond on average to their hosting environment, in terms of diffusion and energization.

The variation of the slope of the probability distribution function $P(E)$ of the electric field strengths admitted by the dissipation regions permeating the coronal volume above an AR around the data specified value $a = -2.95$, appears to not only have a significant qualitative as well as quantitative effect on particle acceleration, but is also the subject of an active research being conducted in relation to flare productivity of solar ARs (see §3.2.3).

(a) By decreasing the power-law index of $P(E)$ from the above value to $a = c_{-} = -3.85$, we attempt to simulate an environment of energetically weaker but more frequent flaring events with an individual energy output < $10^{25}$ ergs, Parker (1988) called nanoflares. According to the nanoflaring scenario, this fragmented environment constitutes an efficient mechanism able to account for the heating of the solar corona.

(b) Given that our data validation process has indicated that we have only probed with sufficient accuracy those events with an excess accumulated magnetic energy $> 10^{25}$ ergs, it is reasonable to assume that the simulation for $a = c_{d} = -2.95$ builds an environment hosting those stronger events also called microflares and flares, commonly attributed with the ability to efficiently accelerate particles, thus considered responsible for the hard X-ray corona.

(c) To further explore the range of effects $P(E)$’s power-law index has on particle acceleration, we proceeded with one last simulation, this time setting $a = c_{+} = -2.05$, i.e. stretching the limits of the PDF in terms of admitting a finite mean. The environment this setting simulates is comprised of extremely energetic localized events that could only be associated to powerful (class M, X) flares.
5.2 Particle Energization and Diffusion in Velocity Space

Figure 13 shows the results of the three simulations described, in terms of the kinetic energy distribution $P(E_{kin})$ of the test-electron population, after a time of 50sec has elapsed. The blue - (empty/filled) square - dotted lines correspond to the initial/final particle population, with “final” meaning the electrons that remained inside the computational domain throughout the entire simulation. The kinetic energy distribution of the particles leaving the cylinder from its top and bottom base are shown in red and green - (upward/downward) triangle - dotted lines.

Case (a), (figure 13(a)), clearly supports the attempted heating scenario, at least for the strongly responsive electrons quickly leaving the computational domain (in $\sim 10$sec). This setting causes an increase in temperature from $\sim 10$keV to $\sim 300$eV, reached in nearly 20sec. Note that a final temperature of $\sim 100$eV ($\sim 1$ million Kelvin) suffices to suggest an efficient heating mechanism is taking place.

Case (b), (figure 13(b)), corresponding to the data deduced PDF for the electric field strengths, also causes heating of the particle population exiting the simulated loop from the footpoints, with similar characteristics than those case (a) exhibits, only less effective. The final temperature in this case is $\sim 200$eV. Besides heating, this setting also accelerates the electrons up to $\lesssim 100$keV, forming a power-law in $P(E_{kin})$’s tail with an absolute index $\approx 3.57$.

Case (c), (figure 13(c)), suggests that an impulsive acceleration mechanism is acting on both particle populations (those remaining inside the computational domain, and those leaving). In nearly 1sec both distributions reach their maximum kinetic energies. The power-law tail of the $P(E_{kin})$ of the particles remaining inside the cylinder has an absolute index of $\approx 2.65$, while the much more extended power-law tail of the distribution of the particles leaving the simulation is $\approx 2.03$. Finally, this latter population is also moderately heated, up to $\sim 25$eV.
Figure 13: Probability distribution function (normalized to 1) of the kinetic energy $E_{\text{kin}}$ of the initial particle population (Initial), the population of particles remaining inside the computational domain during the entire simulation (Inside), and the particles leaving from the bottom and top boundaries (Footpoints 1 and 2). Figures (a), (b) and (c) correspond to the three simulations performed, setting the power-law index of the PDF of the electric field strengths encountered by the electrons to different values. (a) $c_- = -3.85$, (b) $c_{d} = -2.950$ (data deduced), and (c) $c_+ = -2.05$. Cases (b) and (c) correspond to the parametric study performed, and (a) is the data based case (see 4.1.3). The parameters of the fitted distributions (Maxwellians and Power-laws) also shown, are given in the legends of each figure. Total simulation time 50sec.
A common feature presented in all three cases, is the splitting of each of the separate electron populations in two subgroups. For example, in case (a), (figure 13(a)), the kinetic energy of the entire population of particles leaving the cylinder is seen to follow two separate distributions, namely the one roughly corresponding to the initial temperature $\sim 10\text{eV}$, and the second, heated one, reaching $\sim 300\text{eV}$.

This feature directly reflects on the effect of the number of encounters experienced on average by the particles, and heavily depends on the sizes of both characteristic lengths $s$ and $\ell$ admitted by the particle model. The particles exiting the cylinder having encountered a small number of unstable regions are the ones leaving with a slightly altered distribution of energies. The particles having gone through a lot more encounters on the other hand, are the ones setting up the “heated” distribution [or “accelerated” had we chosen to demonstrate this effect using e.g. case (b)].

One first conclusion drawn from this observation is that the effect of the mechanism studied has a unified and global statistical effect on the particles, provided the number of encounters experienced by them on average, is sufficiently large. This is reminiscent of the central limit theorem (CLT), only, in a revised version; one in which the convolution of a sufficiently large number of (independent?) stochastic processes leads not to a normal distribution, but to that distribution this mechanism is capable of providing, given a certain set of parameter values.

Turning to the official, proven CLT, and the condition it posses that the convoluted processes admit probability distribution functions with finite mean and variance, one can note in a continuation of a discussion initiated in §4.1.3, that the power-law indices of $P(E)$ implemented in this parametric study, with absolute values $3.85$, $2.95$ and $2.05$, progressively drive this PDF away from the CLT’s satisfaction. It should, therefore, be expected for these three simulations to result in heating, moderate and enhanced acceleration respectively.

Besides this strictly mathematical argument, however, there is the entire Dreicer theory on the runaway effect briefly discussed in paragraph 4.1.3, explaining the different response of the particles to these three settings. Even though the mean of the electric field strength $\langle E \rangle$ is the same in all simulations, $\sim 10^{-4}E_D$, in the first simulation the number of super-Dreicer fields encountered by the particles is vanishingly small. This case therefore, is expected, as observed, to produce a very limited number of runaway electrons, taken form the high energy tail of the initial Maxwellian. As the number of super-Dreicer encounters increases for the second simulation and even more so for the third one, the number of runaways starts to dominate, thus giving rise to the progressively enhanced acceleration observed, at the expense of heating.

Had the sizes of the lengths admitted by $P(s)$ and $P(\ell)$ (especially that of $s$) been downscaled by one or more orders of magnitude, this would increase the average number of encounters and the unaffected population would cease to exist. This is an important finding, that stresses out the need to observationally access smaller spatial scales. Even though the degree of heating (or acceleration) does not seem to be strongly affected by whether this split in particle population
occurs or not, one cannot help but wonder; if an even worse resolution is able to hinder the proper expression of the heated (accelerated) population, what could a far better resolution possibly reveal?

The temporal characteristics of the mechanisms set in by the increasing power-law indices $c_-, c_d$ and $c_+$ of $P(E)$, as well as their diffusive properties were also studied, and the results are presented in figures 14 and 15, for all three cases.

Cases (a) and (b) do not present significant differences in terms of the quantities studied, namely the mean kinetic energy $\langle E_{kin} \rangle$, and the Fokker-Planck (parallel component) diffusion $D_\parallel$ and drift $A_\parallel$ coefficients of the particles found inside the computational domain at any given instant in time. During the first $\sim 10$ seconds, particle energization proceeds in an enhanced super-diffusive manner, leading a large number of particles outside the computational domain, with kinetic energy distributions nearly like those (final ones) shown in figure 13 (red and green plots). After this initial impulsive phase of energization peaking at around $10$ sec, the particles remaining inside the cylinder are, on average, those less responsive to the electric fields they encounter (see discussion on the Dreicer field, §4.1.3). For a brief time period around peak, diffusion becomes normal and then turns in a much slower rate into a sub-diffusive behavior lasting until the end of all simulations.

Case (c) presents the same qualitative characteristics cases (a) and (b) do, albeit in a significantly reduced degree, with one additional feature strikingly differentiating it from them. The reduced picture of energization expressed by the “continuum” part of the plots in figures 14 and 15 for case (c) (in blue lines), along with those abrupt and shortly lived enhancements (spikes) all quantities exhibit, emphasizes that $\langle E_{kin} \rangle$, $D_\parallel$ and $A_\parallel$ were defined and computed to represent the processes taking place inside the computational domain, and last long enough to have an overall bulk effect in it. Those spikes are traces of the formation of the rightmost part of the flatter power-law tail (absolute index 2.03) of $P(E_{kin})$ in figure 13c (red and green), and simply verify a rather self evident fact: Any extremely enhanced process taking place inside the cylinder quickly drives the particles (its carriers) outside it, thus rendering itself unable to neither affect or be affected by other processes (such as collisions). The “continuum” part of these plots therefore represents the processes involving the bulk population of particles (heating and/or moderate acceleration) which are, indeed, less profound in case (c).
Figure 14: Temporal evolution of the mean kinetic energy of the particle population found inside the computational domain at any given instant in time during an entire simulation lasting 300sec. All three simulations (cases (a), (b) and (c), see text and figure 13) corresponding to the parametric study performed are shown. Case (a): red dashed-dotted line. Case (b): green dashed line. Case (c): blue line.

Figure 15: Temporal evolution of the diffusion and drift coefficients entering the Fokker-Planck equation, computed over the particle population found inside the computational domain at any given instant in time during an entire simulation lasting 300sec. Three simulations and color coding same as figure 14.
Appendices

A Numerical Calculation of the div and curl Operators in 3 Dimensions

A.1 Vector Calculus: Divergence and Rotation of a Vector Field

Let \( \mathbf{B} \) be a \( C^n \) vector field over a set \( A \subset \mathbb{R}^3 \), i.e. \( \mathbf{B} : A \rightarrow \mathbb{R}^3 \), with Cartesian coordinates \( (B_x, B_y, B_z) \). The divergence \( \text{div} \mathbf{B} \) or \( \nabla \cdot \mathbf{B} \) of \( \mathbf{B} \) defines a scalar field over \( A \), given by

\[
\nabla \cdot \mathbf{B} = \frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y} + \frac{\partial B_z}{\partial z}.
\]

(36)

The rotation \( \text{curl} \mathbf{B} \) or \( \nabla \times \mathbf{B} \) of \( \mathbf{B} \), defines a vector field over \( A \), symbolically expressed as the determinant

\[
\nabla \times \mathbf{B} = \begin{vmatrix}
\hat{e}_x & \hat{e}_x & \hat{e}_x \\
\frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\
B_x & B_y & B_z
\end{vmatrix}
\]

\[
= \left( \frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z} \right) \hat{e}_x + \left( \frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x} \right) \hat{e}_y + \left( \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right) \hat{e}_z.
\]

(37)

Now consider a 3-dimensional homogeneous rectangular grid. Let vector field \( \mathbf{B} \) be defined on this grid, that is, assume a triplet \( (B_x, B_y, B_z) \), known for each point in the grid. The aim in this section, is to numerically calculate \( \nabla \cdot \mathbf{B} \) and \( \nabla \times \mathbf{B} \) on that grid. To this end, we will use 2nd order accurate (with respect to the grid’s cell size), finite difference approximations to the first order derivatives involved in definitions (36) and (37), and the method of undetermined coefficients, described in the next paragraph.

A.2 Finite Differences Using the Method of Undetermined Coefficients

Let \( u \) be a real function of \( x \in \mathbb{R} \) and \( x_k, \ k = 1, 2, \ldots \) some equispaced values in a discrete subset \( S \) of \( \mathbb{R} \). Assume that \( x_{k+1} - x_k = \Delta x \in \mathbb{R} \). Now suppose that the values of \( u \)'s derivatives \( (d^m u/dx^m)|_{x_k} \) up to some order \( M \), i.e. \( m = 0, 1, \ldots, M \) are somehow known to us (e.g. obtained from measurements of some kind), even though the actual functional forms of these derivatives may be unknown. We want to numerically calculate the values of the \( n \)-th order derivative \( (d^n u/dx^n)|_{x_k} \) of \( u \) on \( S \), where \( n \geq M \), using some form of finite difference approximation, accurate to some specified order in \( \Delta x \).

The finite difference approximation to the \( n \)-th order derivative \( (d^n u/dx^n)|_{x_k} \) of \( u \) at \( x_k \) can
be written as a linear combination of the known derivatives of $u$, (Tomas, 2013) as:

$$\frac{d^n u}{dx^n} \bigg|_{x_k} \approx \sum_{j=\ell_1}^{\ell_2} \sum_{m=0}^{n} \frac{1}{\Delta x^{n-m}} \cdot \alpha_{j,m} \cdot \frac{d^m u}{dx^m} \bigg|_{x_{k+j}},$$

(38)

for an appropriate choice of $n$, $m$, $\ell_1$, and $\ell_2$, such that $\ell_1 \leq 0$, $\ell_2 \geq 0$, and $(\ell_2 - \ell_1) \geq n$. This choice, according to the finite difference approximation lemma presented below, will determine the accuracy of the approximation.

We present the finite difference approximation lemma as it is given in (Tomas, 2013), with a slight change in notation:

**Lemma.** The $(q+1)$ coefficients $\alpha_{j,m}$ for the finite difference approximation (38) can be obtained by solving the following set of simultaneous equations:

$$\sum_{j=\ell_1}^{\ell_2} \sum_{m=0}^{n} \gamma_{(i-m),j} \cdot \alpha_{j,m} = \delta_{ni}, \quad i = 0, 1, \ldots, q,$$

(39)

where

$$\gamma_{i,j} = \begin{cases} \frac{j!}{i!}, & \text{if } i > 0 \\ 1, & \text{if } i = 0 \\ 0, & \text{if } i < 0 \\ \end{cases}$$

(40)

and $\delta_{ij}$ is the Kronecker delta.

The truncation error is given by

$$\text{Error} = \sum_{r=q+1}^{\infty} \sum_{j=\ell_1}^{\ell_2} \sum_{m=0}^{n} \gamma_{(r-m),j} \cdot \alpha_{j,m} \cdot \frac{d^r u}{dx^r} \bigg|_{x_k} \cdot \Delta x^{r-n}.$$  

(41)

It is now time to turn to the application described at the beginning of this section. This will also make clear the reason why we chose to use the method of undetermined coefficients, instead of approaching the problem in a more familiar way, e.g. by taking explicit Taylor expansions of the function $u = u(x)$ and manipulating them into the formulae needed.

Function $u$ will hereafter be assumed to represent any one of vector $B$’s components $B_x$, $B_y$, and $B_z$. The first thing we notice is that $n = 1$, since we are interested in first order derivatives of $u$. Setting $n = 1$ and writing the terms of the second summation in equation (38) one by one, it becomes

$$\frac{d^n u}{dx^n} \bigg|_{x_k} \approx \sum_{j=\ell_1}^{\ell_2} \left( \frac{1}{\Delta x} \cdot \alpha_{j,0} \cdot u \bigg|_{x_{k+j}} + \alpha_{j,1} \cdot \frac{d^1 u}{dx^1} \bigg|_{x_{k+j}} \right).$$

(42)

We only know the values of $u$ on the grid, and not those of any of its derivatives, so we require that $a_{j,1} = 0$, $\forall j \in [\ell_1, \ell_2]$. At the beginning of this section, it was mentioned that the approximate formulae for the derivatives we are seeking must be 2nd order accurate in $\Delta x$, indicating
an error of order $O(\Delta x^2)$. We know that for a finite difference approximation to a 1st order derivative, the minimum number of neighboring points to be used for a 2nd order accuracy is 2. We will therefore proceed assuming that an accuracy of order 2 requires a minimum number of 3 neighboring points. This means that $(\ell_2 - \ell_1) = 2$ which is greater than $n = 1$ as it should, according to the methods prescriptions.

The next step is to come up with the pairs $(\ell_1, \ell_2)$ differing by 2, that seem suitable for our purposes. Note that for $(\ell_1, \ell_2) = (-1, 1)$ equation (42) will contain $u|_{k-1}$, $u|_k$ and $u|_{k+1}$, suggesting that the resulting formula will be that, known as central finite difference approximation to a first derivative which is indeed second order accurate. By the same reasoning, letting $(\ell_1, \ell_2) = (0, 2)$ implies a forward finite difference approximation, and $(\ell_1, \ell_2) = (-2, 0)$ a backward finite difference approximation to a first derivative, both second order accurate. Since the form of the central finite difference approximation to a 1st order derivative is well known, we will proceed with its derivation in order to exercise our understanding of the method employed, and simply provide the other two formulae at the end of this paragraph.

For $(\ell_1, \ell_2) = (-1, 1)$ and $a_{j,1} = 0$, $j = -1, 0, 1$ equation (42) reads

$$\left. \frac{d^n u}{dx^n} \right|_{x_k} \approx \frac{1}{\Delta x} \cdot \left( \alpha_{-1,0} \cdot u|_{x_{k-1}} + \alpha_{0,0} \cdot u|_{x_k} + \alpha_{1,0} \cdot u|_{x_{k+1}} \right), \quad (43)$$

implying that the number of coefficients $\alpha_{ij}$ we need to calculate is 3. The three simultaneous equations providing these coefficients are given by equation (39), for $q = 2$. We thus have after a little algebra,

$$\gamma_{i,-1} \cdot \alpha_{-1,0} + \gamma_{i,0} \cdot \alpha_{0,0} + \gamma_{i,1} \cdot \alpha_{1,0} = \delta_{1i}, \quad i = 0, 1, 2. \quad (44)$$

Using the definition (40) for $\gamma_{i,j}$ and writing equation (44) as three separate equations, we end up with the system

$$\begin{align*}
\alpha_{-1,0} &+ \alpha_{0,0} + \alpha_{1,0} = 0 \\
-\alpha_{-1,0} &+ 0 \cdot \alpha_{0,0} + \alpha_{1,0} = 1 \\
\frac{1}{2} \alpha_{-1,0} &+ 0 \cdot \alpha_{0,0} + \frac{1}{2} \alpha_{1,0} = 0
\end{align*}$$

or in matrix notation,

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} \alpha_{-1,0} \\ \alpha_{0,0} \\ \alpha_{1,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (45)$$
Now applying Cramer’s rule for solving systems of linear equations, we get the solution

\[
\begin{pmatrix}
\alpha_{-1,0} \\
\alpha_{0,0} \\
\alpha_{1,0}
\end{pmatrix}
= \begin{pmatrix}
-\frac{1}{2} \\
0 \\
\frac{1}{2}
\end{pmatrix}
\] (46)

which we substitute to equation (43) to finally arrive at this well known as **central finite difference approximation to 1st order derivative**:

\[
\left. \frac{du}{dx} \right|_{x_k} = \frac{u|_{x_{k+1}} - u|_{x_{k-1}}}{2 \Delta x} + \mathcal{O}(\Delta x^2).
\] (47)

Note that we wrote this last formula as an equality instead of an approximation, adding the error term \( \mathcal{O}(\Delta x^2) \) at the right hand side since, as can be shown, equation (41) does indeed render an error of second order with respect to \( \Delta x \), given the specified for this case parameter values. We will avoid following these calculations step by step, and simply provide the resulting error formula. In a form indicating that it is of order \( \mathcal{O}(\Delta x^2) \), it reads

\[
\text{Error}_{\text{CFD}} = \frac{1}{6} \cdot \Delta x^2 \sum_{r=4}^{\infty} \left( -\frac{1}{2} \gamma_{r,-1} + \frac{1}{2} \gamma_{r,1} \right) \cdot \left. \frac{d^r u}{dx^r} \right|_{x_k} \Delta x^{r-1},
\] (48)

where the subscript \( \text{CFD} \) is an abbreviation of **Central Finite Difference**.

We now provide the second order accurate formulae for the backward and forward finite difference approximation to the first derivative \( (du/dx)|_{x_k} \), along with their corresponding errors.

The **backward finite difference approximation to 1st order derivative** obtained for \((\ell_1, \ell_2) = (-2, 0)\) (equation (42)), reads

\[
\left. \frac{du}{dx} \right|_{x_k} = \frac{u|_{x_{k-2}} - 4 u|_{x_{k-1}} + 3 u|_{x_k}}{2 \Delta x} + \mathcal{O}(\Delta x^2),
\] (49)

with an error given by

\[
\text{Error}_{\text{BFD}} = -\frac{1}{3} \cdot \Delta x^2 \sum_{r=4}^{\infty} \left( \frac{1}{2} \gamma_{r,-2} - 2 \gamma_{r,-1} + \frac{3}{2} \gamma_{r,0} \right) \cdot \left. \frac{d^r u}{dx^r} \right|_{x_k} \Delta x^{r-1},
\] (50)

where the subscript \( \text{BFD} \) is an abbreviation of **Backward Finite Difference**.

The **forward finite difference approximation to 1st order derivative** obtained for \((\ell_1, \ell_2) = (0, 2)\) (equation (42)), reads

\[
\left. \frac{du}{dx} \right|_{x_k} = \frac{-3 u|_{x_k} + 4 u|_{x_{k+1}} - u|_{x_{k+2}}}{2 \Delta x} + \mathcal{O}(\Delta x^2),
\] (51)
with an error given by

\[
\text{Error}_{\text{FFD}} = -\frac{1}{3} \cdot \left. \frac{d^3 u}{dx^3} \right|_{x_k} \Delta x^2 + \sum_{r=4}^{\infty} \left( -\frac{3}{2} \gamma_{r,0} + 2\gamma_{r,1} - \frac{1}{2} \gamma_{r,2} \right) \cdot \left. \frac{d^r u}{dx^r} \right|_{x_k} \Delta x^{r-1},
\]

(52)

where the subscript \( \text{FFD} \) is an abbreviation of *Forward Finite Difference*.
B       A Spatial Clustering Algorithm

B.1       Stating the Problem and Classifying the Clustering Process

In this section we discuss the algorithm produced for the cluster analysis performed on our data. We begin by stating the problem at hand.

\textit{Given a rectangular and homogeneous spatial grid in 3-dimensions with a vector field defined on it, impose a threshold on the vector field and group the grid-nodes of the resulting partition into disjoint clusters comprised of one or more nodes each.}

As Guojun Gan describes in his monograph (\cite{Gan2011} and references therein), a typical clustering process\(^2\) beginning with the raw data given and ending with the final output delivered in a form appropriate for any further analysis required, consists of five separate steps. We describe these steps here in reference to the procedures successively applied on the data relating to the problem stated above, at the same time introducing some relevant terminology in the context of exploratory data mining.

(a) The raw data provided for the particular problem consist of the coordinates \((Q_x, Q_y, Q_z)\) corresponding to each node \(Q\) in a 3-dimensional grid (discretized space) and the values of some vector field \(J = (J_x, J_y, J_z)\) assigned to each of these grid-points. The \((Q_x, Q_y, Q_z)\) triplet and \(J\) are the so called attributes characterizing each point in the grid. The former is of the \textit{discrete} and \textit{ordinal data type}, with the two terms respectively indicating that \((Q_x, Q_y, Q_z)\) can only take distinct (integer) values which, as triplets, exhibit some natural ordering depending on how the grid is spanned. The latter is an attribute of the \textit{continuous} and \textit{ratio} data type, meaning that the components of \(J\) can take on any value within specified ranges and have a natural zero respectively. Identifying that the grid-points are characterized by these two attributes of the specified data types is a process termed \textit{pattern representation}.

Feature selection\(^3\) and feature extraction are another two processes usually included in this first step. In \textit{feature selection} the user chooses those attributes that seem to be more efficient than others, in terms of enabling the clustering process. In the \textit{feature extraction} process, the original attributes can be transformed into new ones, according to the problem’s needs. In treating the particular problem set, both attributes are needed, since we first need to impose a threshold on the vector field and then identify the disjoint clusters formed by the grid-points of the resulting partition (to be clarified below). During the feature extraction we also deduce

\(^2\)The notion of a cluster depends mainly on the problem posed and the systematic approach selected to treating it. It is therefore difficult to come up with a formal definition of a cluster. We postpone our definition of a cluster pertaining to the particular problem at hand, a definition termed \textit{operational} (Gan, 2011), until all necessary terms and definitions have been introduced.

\(^3\)Feature, as well as variable, are two other words used to term an attribute.
the magnitude $J = \sqrt{J_x^2 + J_y^2 + J_z^2}$ from $J$'s components, in order to facilitate the threshold imposition.

(b) After an appropriate threshold $J_{th}$ has been imposed on the magnitude of $J$ over the grid, and the partition $P$ of interest has been obtained ($P$ is the set of grid-points $Q^i = (Q_{ix}, Q_{iy}, Q_{iz})$ for which $J_i > J_{th}$), the remaining task is to identify the disjoint clusters within $P$. This, however, requires that we at least have a qualitative notion of what a cluster is, in regard to our data. So for the case examined here, we define a cluster as a group of grid-points, each connected to at least one other group member via a single edge of the rectangular lattice underlying the given grid, or, as a limiting case, a single point isolated in the sense just described.

The next step is the quantitative specification of dissimilarity\(^5\) between each pair of grid-points $(Q^i, Q^j)$ in $P$, or elements/objects in the set of data being clustered, in the more general case. The task therefore, in this dissimilarity measure definition step, is to define a dissimilarity or distance measure via a binary function $D$ over the domain of the attribute it is to be applied to. This function can satisfy all four conditions defining a distance function (or metric), or it can violate the triangle inequality, in which case it is called a semimetric. (Gan, 2011) For the particular case examined $D = D(Q^i, Q^j)$.

According to the definition of a cluster given above, it now becomes clear that identifying a set of disjoint clusters distributed over a rectangular homogeneous grid in 3-dimensions, simply requires that all immediate neighbors of each point in $P$ are found and systematically recorded. We can consistently define an immediate neighbor of a reference point $Q^R$ in the grid, to be any other point connected to $Q^R$ via a single edge of the underlying rectangular lattice. The Manhattan distance of the family of the Minkowski metrics

$$L_p(Q^i, Q^j) = \left( \sum_{n=1}^{k} |Q_{in}^i - Q_{in}^j|^p \right)^{\frac{1}{p}},$$

(53)

where $k$ the dimensionality of the data (here $k = 3$) and $(Q_{1x}, Q_{2x}, Q_{3x}) \equiv (Q_{ix}, Q_{iy}, Q_{iz})$, for $p = 1$, i.e.

$$L_1(Q^i, Q^j) = \sum_{n=1}^{3} |Q_{nx}^i - Q_{nx}^j|,$$

(54)

seems fit for our purpose, since given a reference point $Q^R$, it will be $L_1(Q^R, Q) = 1$ if and

---

\(^4\)The determination of an appropriate value for $J_{th}$ is part of an optimization process known as cluster validity, the last step of the clustering process (described below). For the time being we will assume that the clustering process proceeds using a “test” value for $J_{th}$, within a user pre-specified range.

\(^5\)Or similarity, which is defined in an opposite way to dissimilarity, but will not be discussed here, since it is not relevant to the problem under consideration.
only if $Q$ is an immediate neighbor of $Q^R$, and $L_1(Q^R, Q) > 1$ otherwise.\footnote{That is, assuming that successive nodes along each of the grid’s axes are separated by unit intervals.}

(c) Every clustering algorithm, regardless of the nature of the clustering criteria it uses to somehow relate the elements in the data it acts on, must follow a well designed and developed system in applying these criteria on the data, and extracting the information required for an optimal solution to a particular problem. In this step, called clustering, the user must eliminate any vagueness pertaining to the definition of a cluster previously given, and specify the exact structural form of the final output. Lead by these considerations, during this step a method gets selected and implemented, and the clustering algorithm is formally classified.

The variety of clustering problems posed over the last 50 years, gave rise to the development of a large number of clustering algorithms. At the top level of their classification lies the distinction between a hierarchical and a partitional clustering algorithm. The final output of a hierarchical clustering algorithm is a multi-level structure of nested partitions of the original data from which the user may choose the one exhibiting the desired properties (e.g. number of clusters or level of refinement). A partitional clustering algorithm on the other hand, produces a single partition in which the number of clusters can be either fixed beforehand, or remain unknown until the process is complete.

Both these classes of clustering algorithms are subject to further classification, a reference to which, however, does not serve any of the purposes of this short, and oriented towards a specific problem, review (the readers are referred to (Gan, 2011) and (Gan et al., 2007) for further details). The algorithm created to treat the problem under consideration falls into the partitional clustering category and its formal classification simply requires a single additional level of categorization, in order to distinguish between the hard clustering and fuzzy clustering methods. While in hard clustering, each data element belongs to one and only one cluster, in fuzzy clustering, an object can be shared among two or more clusters. Our algorithm can now be formally classified as a partitional, hard clustering algorithm, that groups the original data without any prior knowledge of the number of disjoint clusters in the resulting partition.

(d) A cluster analysis is usually performed in order to uncover some form of structure underlying the data obtained through measurements or observations, possibly revealing a physical law behind the mechanism responsible for the phenomenon under study. The output of a clustering algorithm is therefore usually subject to further analysis, the nature of which dictates the form in which the results should be rendered. In this step of data abstraction, the form of the final output is determined and extracted from the immediate results of the clustering process.
As it so happens, most of the project developed as part of this thesis heavily relies on this cluster analysis. Without delving into topics discussed in other sections, what we need to know about the clusters formed during this analysis, is

- how many clusters arise for a given threshold $J_{th}$,
- the number of grid-points each cluster consists of,
- the coordinates $(x, y, z)$ and the magnitudes $J$ corresponding to each grid-point in a given cluster,
- the position of a point somehow defined to represent each cluster (defined below),
- and the minimum, maximum and average value of the magnitude $J$ corresponding to a given cluster, as computed from the grid-points comprising it.

For this data abstraction step to be completely specified, there remains to define the point representing a cluster, mentioned above. This point was chosen to be the centroid of a cluster, defined here as the point located at the weighted average position of all the grid-points comprising the cluster. This means that this representative point need not, and most probably will not, be a grid-node. Specifically, if a cluster consists of $n_C$ points $Q^i = (Q^i_1, Q^i_2, Q^i_3)$, $i = 1, \ldots, n_C$, then the coordinates $Q^C_n$, $n = 1, 2, 3$ of its centroid are defined as

$$Q^C_n = \frac{\sum_{i=1}^{n_C} J_i \cdot Q^i_n}{\sum_{i=1}^{n_C} J_i}.$$  \hspace{1cm} (55)

Note that we use $J_i$ to weight the contribution of the respective grid-point $Q^i$ in determining the position of each cluster’s centroid, based on the fact that the result of the clustering process performed here, greatly depends on the value given to the threshold $J_{th}$.

(e) The clustering process concludes with this last step called cluster validity. This process assumes that no “programming blunders” jeopardize the algorithm’s general soundness and focuses on assessing the systematic approach of the method used, in relation to the particular problem. The results of a clustering algorithm can be assessed by means of statistical testing, in which case the criteria used by the respective methods are termed either external or internal (Gan, 2011; Halkidi et al., 2002a), or by direct comparison against clustering algorithms based on different clustering methods, or based on the same method but using different parameter values. This latter method is said to be based on relative criteria.

The results of the clustering algorithm constructed for the purposes of this project were assessed and modulated using a method based on relative criteria. This method aims at finding the “best” clustering scheme defined by the algorithm, by determining the optimal values of the parameters associated with it. As already stated above, the number of disjoint

\footnote{“Best” here refers to the clustering scheme that somehow best fits the underlying data.}
clusters in the final partition is not a predefined parameter of the problem. The only parameter definitively leading the clustering process, is therefore the threshold $J_{th}$.

The optimal value of the threshold $J_{th}$ is determined by applying a procedure described in (Halkidi et al., 2002b): The algorithm is ran for a sequence of values $J_{th}$ within a wide range $[J_{min}, J_{max}]$, in order to determine the largest interval $[J_{min}^*, J_{max}^*]$ for which the total number of clusters formed remains constant within a pre-specified tolerance. The optimal value of $J_{th}$ lies in the middle of this interval, i.e.

$$J_{th} = \frac{J_{min}^* + J_{max}^*}{2}.$$  \hfill (56)

### B.2 The Algorithm

The algorithm performing the clustering process described in the previous paragraph, was written in the Fortran programming language. In this paragraph we delve a little deeper into the specifics of the process, following the sequence of operations performed step by step, and giving proper emphasis on some key-points of the routine developed.

This algorithm skips the first step of the cluster analysis called data representation, assuming that another routine preprocessed the original data, providing the magnitude $J$ of the vector field defined on the grid. In the following description we adopt the notation (naming of the variables and procedures) used in the source code developed.

- **Input Acquisition**
  
The program’s input is an unformatted Fortran (*.dat) file. The real array $Qgp$ (Quantity–grid–point) of size $nx \times ny \times nz$ used to read this input file will eventually contain the values of some scalar quantity, at every point in a 3–dimensional grid of the same size.

- **Threshold Imposition**
  
  We set up a variable holding a threshold value for $Qgp$ named threshold and transition to the integer array $I0$ of size $nx \times ny \times nz$ which reads 1 at the grid points where $Qgp > \text{threshold}$ and 0 at the rest of the grid points. The integer variable $\text{num}$ holds the number of grid points for which $Qgp > \text{threshold}$ is satisfied.

- **Grid Point Coordinates after Threshold Imposition**
  
  An integer array named $I1$ is allocated\(^8\) the size $\text{num} \times 3$ and contains the grid – coordinates of the points at which $Qgp > \text{threshold}$ holds. Integer array $I\text{temp}$ is the same

\(^8\)Actually, array $I0$ is defined during the acquisition of the value $\text{num}$ must take, so that $I1$ can be allocated. It is later used in defining $I1$, instead of $Qgp$, simply for reasons of convenience, it being an integer array.
size as $I_1$ and serves as an auxiliary array for the consistent rearrangement of blocks of elements within $I_1$, during the clustering process.

- **Clustering Process: Grouping Immediate Neighbors**
  The clustering process starts with defining the first element in $I_1$ to be the reference point, hereafter denoted RP, with grid-coordinates $(i_{x0}, i_{y0}, i_{z0})$, and with the intention to investigate all subsequent elements of $I_1$ in terms of satisfying the immediate neighbor condition (INC). At this point therefore, any other grid point is a potential immediate neighbor (INBr) of the RP, temporarily holding the three integers $(i_x, i_y, i_z)$ to represent its grid-coordinates.

  A `DO WHILE` loop performs this investigation, and it is nested within an outer `DO` loop through which all unclassified grid points subsequently become reference points.

  **Counters**

  - Integer variable `more` is a switch determining the range $[1, i_{for}]$ of the counter advancing the outer `DO` loop. The values it takes are 0 and 1, where $more = 0$ implies that either the clustering process is at its beginning, or a complete cluster has been formed and got recorded. $more = 1$ on the other hand implies that a cluster is in the process of formation, that is, an INBr of the current RP has been found, and it (the INBr) will now be termed RP and be investigated (this becomes clearer below).

  - `ngps` is an integer variable that is set equal to 1 each time a complete cluster is formed and gets recorded. It is attached to each cluster under formation, counting the number of grid points comprising it.

  - `ntocheck` is a continuously updated integer variable counting the number of unclassified grid points waiting to be investigated, either as RPs or as potential INBr. It therefore determines the range $[1, i_{for}]$ of the outer `DO` loop’s counter, when $more = 0$, and also controls the completion of the inner `DO WHILE` loop. `ntocheck` is initially set equal to `num` and successively decreases as clusters form (by subtracting from it either 1/single-point cluster or `ngps`/multi-point cluster) until it becomes $<1$, an event which signals the termination of the clustering process.

  - `icluster` takes on the value `ngps` has, every time a cluster is in the process of formation ($more = 1$ in that case, which is the only time `icluster` needs to have a consistent value). It therefore determines the range $[1, i_{for}]$ of the outer `DO` loop’s counter whenever ($more = 1$).
– isearch is the counter advancing the DO WHILE loop. Before each initiation of the DO WHILE loop, isearch is set equal to ngps+1 making sure that no pair of grid points is checked in terms of their neighboring relation twice. Within the loop it is used to index the position (within I1) of each potential INBr of the current RP.

– Integer variable index identifies the subsequent clusters formed.

There is a statement label 874 at the beginning of the clustering process, where the range [1,ifor] of the counter advancing the outer DO loop is determined. The events leading to this 874 CONTINUE statement, except from the beginning of the process, are the following:

1. Once a RP is defined, the main job of the inner DO WHILE loop is to check whether subsequent potential INBrs satisfy the INC with respect to that RP. This is accomplished with an IF statement. Accessing that IF statement’s body means that an INBr of the current RP has been found. The following operations are then performed:

   – Integer i j holds the index of the INBr identified.
   – Array Itemp is used to rearrange blocks of elements within array I1 (see figure 16).
   – Array I1 is then updated (I1=Itemp).
   – Counter ngps is increased by one since one additional grid point has been included to the cluster under formation, counter icluster is set equal to ngps, and finally more is set equal to 1.
   – The next line of code that is to be executed leads to the statement label 874.

**Figure 16:** The first element rearrangement within I1 performed during the clustering process. A cluster is in the process of formation.

What happens upon return to the statement label 874 in this case, is probably the trickiest part of this process (the rest of it is pretty straightforward). You may have noticed that the INBr identified is now the first element in the updated I1. The outer DO loop will define this element to be the next RP. This means that the previous RP, now second element in I1 is abandoned so to speak, after having identified only one of its INBrs (there may be others!). This is why icluster is set
equal to ngps, and more=1; and this is why the search for a new INBr will start from the potential INBr indexed (in the updated I1) isearch=ngps+1. The program needs to know that the investigation of the previous RP is not complete (hence more=1⇒ifor=icluster ⇒ icluster=ngps). The outer DO loop is now dedicated to completing the search of the INBrs of grid points belonging to the cluster under formation, which was left unfinished. At the same time the program should know not to investigate the neighboring relation between grid points that have already been determined to be INBrs (hence isearch=ngps+1).

This piling up of “unfinished businesses” stops when a RP (first element of I1) turns out not to have any INBrs (completing the DO WHILE loop). It was however under investigation in the first place, because in the previous iterative step had itself been found to be the INBr of some other RP in the cluster under formation. ifor grid points at the top of I1 must now be checked against all other unclassified grid points, in terms of the INC. This is not necessary however, if the cluster under formation is complete. This is what the next IF ELSE statement is supposed to determine.

2. If ngps=1, then a single–point cluster has been formed. The program records it, moves it at the very last row of I1 (second element rearrangement within I1), so that it will not be investigated again. This is accomplished by updating ntocheck, in this case decreasing it by 1. more is set to 0, the cluster index is updated, and the program returns to the statement label 874 in order to start a whole new search.

If ngps≠1, then there are two possibilities:

3. The cluster under formation is complete. This can only be true however, if i, the outer DO loop’s counter, is equal to ifor, which means that all INBrs, for each grid point already in the cluster, have also been identified. We can now describe things in a more general manner. I1 now consists of two blocks of elements. The upper block contains the grid points of the new complete cluster. At the top of the lower block there are the unclassified grid points (their number should be ntocheck), and its very last elements are the grid points comprising the clusters already recorded. The block rearrangement schematically represented in figure 17 shows that at every definitive iterative step, the unclassified grid points always end up at the top of I1.
In the case where \( i < i_{\text{for}} \), the IF ELSE statement is bypassed with a use of a GO TO statement leading to a second statement label, 875. This simply means that the outer DO loop must be iterated for the cluster under formation to complete.

At every point within the program after a cluster has been recorded there is a check as to whether \( \text{ntocheck} \) has become less than 1, which means that there are no more grid points to investigate. When this condition is met, the clustering process is terminated by breaking the outer DO loop definitely.

- **Output Rendering**

  The output of the algorithm is a formatted Fortran (*.dat) file with 6 columns. The cluster index is given in the 1st column and in the 2nd one the number of grid-points comprising the cluster. Columns 3rd to 5th read the \((x, y, z)\) grid-coordinates of the points belonging to the respective cluster. The 6th column contains the magnitude \( J \) corresponding to each point. It is assumed that the rest of the quantities needed to further process this cluster analysis results (enumerated in the fourth step of the process named data abstraction in the previous paragraph), can be computed from this single output file, using appropriate routines.
C Parallel Random Number Generators in Fortran Using OpenMP

One of the first technical issues a programmer comes across when preparing a parallel program that makes use of one or more pseudo-random number generators (RNGs) for the first time is that, while such a program may run in both sequential and parallel mode producing valid results, these results will most probably (certainly) not coincide. The sequences of pseudo-random numbers assigned to each task during a sequential run will not be the same as those each task acquires when running in parallel mode and moreover, even the sequences obtained during any two subsequent parallel runs will differ quite noticeably. So unless proper precautions are taken, a parallel program that uses random number generators detaches itself from its sequential equivalent, at the same time loosing its reproducibility.

This is an immediate consequence of the fact that, when multiple threads “delve” in random order into a single sequence of numbers generated by an RNG initialized by a unique seed, they end up each picking up a different set of random numbers in every different run. Since controlling the order in which the threads “invade” that unique sequence of random numbers would be a total misconception of parallel programming, the next obvious solution is to force the RNG to generate different sequences of random numbers (by means of different initializing seeds), that somehow correspond not to the thread performing each parallel task, but to the “serial number” of the task itself.

To address this “issue”\(^9\) I have employed a module created by Jason Blevins at the Ohio State University, after applying certain modifications necessary to incorporate the random number generators I was already using in the sequential mode. In this section I describe the functionality of the module adopted and give an account of the modifications I made in order to adjust it to the needs of my parallel program. The original module along with some of the explanations I will be using here, can be found in Jason Blevins’ home page.

I will begin by describing the way the two RNGs\(^10\) I used work, at least in terms of their features relevant to the parallelization process. The RNG providing uniformly distributed numbers in \([0, 1]\) is a function I here call \texttt{rng\_uniform}, and takes a single integer seed as an argument. Let us use this seed’s original name, and call it \texttt{idum}. This seed is actually a variable that changes many times throughout the random number generation process, also passing its various assignments to other variables. Successive calls to \texttt{rng\_uniform} use \texttt{idum}'s constantly updated value, and those intermediate variables mentioned need to preserve their assignments between these calls. This is why, in its initial form, \texttt{rng\_uniform} included them in a \texttt{SAVE} statement. The end product is a random number in \([0, 1]\), assigned to the variable \texttt{rng\_uniform}. The

---

\(^9\) or put differently, this “debugger’s nightmare” since, one cannot rely on the sequential mode to help them identify an error occurring in some of the parallel statements

\(^10\) Both these functions are \textit{Numerical Recipes} routines for the uniform and standard normal distributions respectively.
RNG producing random numbers following the standard normal distribution is a function named \texttt{rng\_gauss} here. It uses \texttt{rng\_uniform} to get a random number in \([0, 1]\) (note, therefore, that some of the calls to \texttt{rng\_uniform} are done within \texttt{rng\_gauss}), and after transforming it to a normally distributed number it passes it onto the \texttt{rng\_gauss} variable. In order to make proper calls to \texttt{rng\_uniform}, \texttt{rng\_gauss} needs to take the updated \texttt{idum} as an argument and in turn, update it accordingly. It therefore also involves the kind of intermediate variables \texttt{rng\_uniform} does.

The point of this coarse description is to by no means underestimate the importance of \texttt{rng\_uniform} and \texttt{rng\_gauss}. These functions are, after all, responsible for producing the random numbers we are interested in. The point is to emphasize that there are certain variables these RNGs involve, each of the parallel tasks needs to know about and manipulate independently, in order to get the “multiple sequence of random numbers” scheme to work.

Fortran 90 introduced the data structure named \texttt{MODULE} to facilitate the need to group and share parameters, variables and subprograms between several different programs. A module can be placed right before the declaration of the main program that is to make use of it, or it can be kept in a separate file provided the main program is properly directed to it. The statement by which a module is made available to the main program is the \texttt{USE} statement, followed by the name of the module. The \texttt{USE} statement must appear right after the declaration of the main program and before the \texttt{IMPLICIT} statement. If the main program is called \texttt{parallel\_tasks} and the module \texttt{rng\_module}, the first way to direct the main program to the module is by use of the following syntax:

\begin{verbatim}
1 MODULE rng_module
2 ! ...
3 END MODULE rng_module
4 PROGRAM parallel_tasks
5 USE rng_module
6 IMPLICIT NONE
7 ! ...
8 END PROGRAM parallel_tasks
\end{verbatim}

\textit{PoC C.1: module inclusion and USE.}

The Fortran 90 feature called \texttt{DERIVED DATA TYPE} is a way of grouping different types of data (integer, real, character, etc.) into an aggregate structure. All derived type’s constituents can be passed as arguments to other programs, which means that they can be initialized and modified as regular variables. In this case the derived type \texttt{rng\_parallel} is defined within \texttt{rng\_module} (see \textit{PoC C.2\textsuperscript{11}}) and its purpose is to collect those RNGs’ variables all tasks need.

\textsuperscript{11}PoC stands for Piece of Code.
(a) know about, but
(b) handle privately.

As long as rng_parallel is specified as a PUBLIC item within rng_module, every program using the module can declare one or more variables to be of type rng_parallel. The following piece of code (PoC C.2) contains the declarations / initializations necessary to the module:

```fortran
MODULE rng_module
IMPLICIT NONE
PRIVATE
INTEGER, PARAMETER :: IK = SELECTED_INT_KIND(9)
INTEGER, PARAMETER :: RDK = SELECTED_REAL_KIND(15)
! PARAMETERS FOR INITIALIZATION OF THOSE VARIABLES IN THE RNGs,
! THAT NEED TO HAVE MULTIPLE INSTANCES.
INTEGER(KIND=IK), PARAMETER :: NTAB = 32
INTEGER(KIND=IK), PARAMETER :: ISD = -19807
INTEGER(KIND=IK), PARAMETER :: NDUM = 123456789
INTEGER(KIND=IK), PARAMETER, DIMENSION(NTAB) :: IZERO = NTAB*0
! THE PROGRAM USING THE MODULE CAN ONLY CALL THESE ROUTINES
PUBLIC :: rng_parallel, rng_seed, rng_uniform, rng_gauss
! EACH (PARALLEL) TASK WILL BE ASSIGNED ITS OWN TYPE rng_parallel
TYPE :: rng_parallel
! the initializations are not necessary, the declarations are!
INTEGER(KIND=IK) :: iseedT = ISD
INTEGER(KIND=IK), DIMENSION(NTAB) :: ivT = IZERO
INTEGER(KIND=IK) :: iyT = 0
INTEGER(KIND=IK) :: idum2T = NDUM
INTEGER(KIND=IK) :: isetT = 0
REAL(KIND=RDK) :: gsetT = 0.0d0
END TYPE rng_parallel
CONTAINS
END MODULE rng_module
```

PoC C.2: module declarations / initializations.

The derived type rng_parallel is defined in lines 15–23 in PoC C.2 and includes iseedT, the variable which will be getting assigned idum’s values during the calls to rng_uniform and rng_gauss, the latter two being the modified RNGs. The rest of the variables in the derived type will be getting assigned the values of those intermediate variables mentioned above, that need to preserve their values between successive calls to the RNGs. It is important to distinguish
between those variables included in `rng_parallel`, such as `iseedT`, and their corresponding variables included in the RNGs, such as `idum`. The former are, through `rng_parallel`, `PUBLIC` variables, while the latter are `PRIVATE` to `rng_module`, as declared in the global statement in line 3 of PoC C.2.

Requirement (a) mentioned previously, that all tasks know about the variables included in `rng_parallel`, is fulfilled by declaring the derived type array variable `rng` inside the main program as shown below (line 6, PoC C.3).

```fortran
1 ! ...  
2 ! NUMBER OF PARALLEL AND SEPARATE TASKS  
3 INTEGER(KIND=IK), PARAMETER :: nt = 4, nst = 10  
4 ! ONLY FOR THE MAIN PROGRAM rng OF TYPE rng_parallel  
5 ! IS AN ARRAY OF DIMENSIONS AS MANY AS THE TASKS  
6 TYPE(rng_parallel), DIMENSION(nt) :: rng  
7 ! ...  
```

PoC C.3: Declaring a derived type array of dimensions as many as the tasks, within the main program.

```fortran
11 !$OMP PARALLEL  
12 ! ...  
13 !$OMP DO  
14 DO i = 1,nt  
15 ! CREATING THE MULTIPLE INSTANCES OF THE RNGs  
16 ! EACH rng(i) IS OF TYPE rng_parallel AND THE DIFFERENT i  
17 ! PROVIDE EACH TASK WITH A DIFFERENT SEED WITHIN rng_seed  
18 CALL rng_seed(rng(i), i)  
19 ! ...  
20 ENDDO  
11 !$OMP END DO  
12 !$OMP END PARALLEL  
```

PoC C.4: Initiating the creation of the RNGs’ multiple instances.

Assuming N threads are devoted to performing nt tasks (note that N is irrelevant to the whole process), each task taken up by a thread will be assigned a “serial number” i through a parallel DO loop, and each will have its own element of the type `rng_parallel rng` array to handle privately. This (b) requirement is fulfilled via the CALL to the subroutine `rng_seed` inside the parallel DO loop (line 8, PoC C.4), thus initiating the creation of what Jason Blevins calls “multiple instances of an RNG”.

What the subroutine `rng_seed` does is shown in PoC C.5. It takes each element `rng(i)` from the main program, and assign and initialize it within `rng_module`, to each task’s private instance `rngV` of the derived type declared (line 10, PoC C.5). Note that, while most variables included in `rng_parallel` are initialized identically (lines 13-17, PoC C.5), `rngV%iseedT`'s
initialization depends on the “serial number” of the task $i$ (mseed within the module), given as second argument to the subroutine. This is the purpose of $\text{rng\_module}$; to initialize each task’s RNG instance in an identifiable and reproducible way.

```fortran
MODULE rng_module

CONTAINS

! EACH TASK CALLS THIS ROUTINE USING ITS OWN rngV and mseed.
! EACH VARIABLE CONTAINED IN rngV GETS INITIALIZED SEPARATELY
! FOR EACH TASK.
! THIS IS HOW THE MULTIPLE INSTANCES OF THE RNGS GET CREATED

SUBROUTINE rng_seed(rngV,mseed)

! ... TYPE(rng_parallel), INTENT(inout) :: rngV
INTEGER(KIND=IK), INTENT(in) :: mseed
rngV%iseedT = ISD - mseed
rngV%ivT(:) = IZERO
rngV%iyT = 0
rngV%idum2T = NDUM
rngV%isetT = 0
rngV%gsetT = 0.0d0
END SUBROUTINE
! ...
END MODULE rng_module
```

**PoC C.5:** Creation of the RNGs’ multiple instances.

The CALL to $\text{rng\_seed}$ (shown again in PoC C.6), and therefore the initialization, is made once for each task. Those $\text{nst}$ successive calls to $\text{rng\_uniform}$ and $\text{rng\_gauss}$ implied by the DO loop contained in lines 3 through 8 in PoC C.6, work on the constantly updated rngV%iseedT constituent of the derived type, corresponding to the main player idum in the way shown in PoC C.7.

```fortran
! ...
CALL rng_seed(rng(i), i)
DO j = 1,nst
  ! SUCCESSIVE CALLS TO THE RNGS ARE MADE BY EACH TASKS
  ! PRIVATE DERIVED TYPE
  uni = rng_uniform(rng(i))
  gas = rng_gauss(rng(i))
ENDDO
! ...
```

**PoC C.6:** Calls to the actual RNGs.
The main program does not have access to thePRIVATE variables idum, idum2, iv and iy manipulated within rng_uniform. The trick in getting the RNG to work as expected, is for each task to keep track of and “safe-keep” the changes those variables are subjected to during its private call to the RNG (lines 12 and 15-17, PoC C.7), and pass them over again at the beginning of the next call (lines 4-7). This way the sequence of random numbers initialized according to each task’s “serial number” remains private to each task, thus creating multiple sequences of random numbers identically reproduced by any run in both parallel and sequential mode.

\begin{verbatim}
FUNCTION rng_uniform(rngV)
  ! ...
  TYPE(rng_parallel), INTENT(inout) :: rngV
  idum = rngV%iseedT
  idum2 = rngV%idum2T
  iv(:) = rngV%ivT(:)
  iy = rngV%iyT
  ! ALGORITHM PRODUCING THE UNIFORMLY DISTRIBUTED RANDOM NUMBERS
  ! ...
  IF (something) THEN
    iv(j)=idum
    rngV%ivT(j) = iv(j)
  ENDIF
  ! ...
  rngV%iyT = iy
  rngV%iseedT = idum
  rngV%idum2T = idum2
  rng_uniform = ! ...
END FUNCTION
\end{verbatim}

**PoC C.7:** Updating the seed and the intermediate variables, before, during and after the random number generation process.

Note that the same module, rng_module, can be used outside the parallel construct of the program. One simply uses a single element of the derived type rng_parallel array rng, by giving i (each task’s identifier in parallel mode) a fixed value, e.g. 1. The use of rng_uniform and rng_gauss is then similar to the one shown in lines 6 and 7 of PoC C.6.
D A Particle Code Realizing Several Modes of Integration, Parallelized Using OpenMP

The source code realizing the model described was built to run parallel using OpenMP as an application programming interface and includes various interwoven channels accessed via the corresponding switches.

D.1 Primary Switches
D.1.1 Equations of Motion

The particle trajectories are obtained by integrating the equations of motion for each particle separately, i.e. by computing the position vector components $x, y, z$ in a 3-dimensional space and also the velocity components, corresponding to a specified set of successive instants in time.

D.1.1.1 Lorentz Force & Guiding Center

A choice can be made, on whether the equations of motion numerically solved will be the ones determined by Newton’s second law with the Lorentz force acting on the particles, or the ones derived by employing the guiding center approximation.

The set of equations that will from now on be referred to as Lorentz Force Equations of Motion is the following:

$$
\frac{dr}{dt} = \mathbf{v},
$$

$$
\frac{d\mathbf{v}}{dt} = \frac{q}{m} \left( \mathbf{E} + \frac{\mathbf{v} \times \mathbf{B}}{c} \right),
$$

where $c$ is the speed of light in vacuum, $m$ and $q$ is the mass and electric charge of the test-particle, $\mathbf{v} = v_x \hat{e}_x + v_y \hat{e}_y + v_z \hat{e}_z$ and $\mathbf{r} = x \hat{e}_x + y \hat{e}_y + z \hat{e}_z$ is the particle velocity and position vectors respectively, $\mathbf{E}$, the electric field and $\mathbf{B}$ the magnetic field of the ambient plasma. Note that cgs units are used throughout.

For the guiding center approximation we have employed the set of equations derived in (Grebogi and Littlejohn, 1984) and also in (Tao and Chan, 2007), adopting the notation used in (Hamamatsu et al., 2007), here called the Guiding Center Equations of Motion:

$$
\frac{dr}{dt} = \frac{1}{B^*} \left[ v_{||} \mathbf{B}^* + c \hat{b} \times \left( \frac{\mu}{q} \nabla \mathbf{B} - \mathbf{E}^* \right) \right],
$$

$$
\frac{dv_{||}}{dt} = -\frac{q}{m B^*} \mathbf{B}^* \left( \frac{\mu}{q} \nabla \mathbf{B} - \mathbf{E}^* \right),
$$

$$
\frac{dv_{\perp}}{dt} = \frac{v_{\perp}}{2 B^*} \nabla \mathbf{B} \left( v_{||} \mathbf{B}^* - c \hat{b} \times \mathbf{E}^* \right),
$$

72
where

\[
B^* = B + \frac{mc}{q} v_\parallel \nabla \times \hat{b},
\]

\[
E^* = E - \frac{m}{q} v_\parallel \frac{\partial \hat{b}}{\partial t},
\]

\[
\hat{b} = \frac{B}{B^*},
\]

\[
B^\parallel = \hat{b} \cdot B^*,
\]

\[
\mu = \frac{m v_\parallel^2}{2 B^*},
\]

and \( v_\parallel \) and \( v_\perp \) are the parallel and perpendicular to the magnetic field \( B \) velocity components, such that \( \mathbf{v} = v_\parallel \hat{e}_\parallel + v_\perp \hat{e}_\perp \) and \( \mu \) is the particle’s magnetic moment.

### D.1.1.2 Non-Relativistic & Relativistic Treatment

The Lorentz Force (57) and Guiding Center (58) equations of motion already given, are appropriate for a non-relativistic treatment. Depending on the electromagnetic field configuration however, the initial energy of the particles, and the total time of integration, some of them may approach the relativistic energy regime. In this case, the appropriate set of equations corresponding to the **Relativistic Lorentz Force Equations of Motion** are:

\[
\frac{dr}{dt} = \frac{u}{\gamma},
\]

\[
\frac{du}{dt} = \frac{q}{m} \left( E + \frac{u \times B}{\gamma c} \right),
\]

where \( u = \gamma \mathbf{v} \) the four-velocity vector (only the three space-like components implied) and \( \gamma \) the Lorentz factor (see below).

For the guiding center approximation, the corresponding **Relativistic Guiding Center Equations of Motion** are:

\[
\frac{dr}{dt} = \frac{1}{B^*_{\parallel}} \left[ \frac{u_{\parallel}}{\gamma} B^* + c \hat{b} \times \left( \frac{\mu}{q \gamma} \nabla B - E^* \right) \right],
\]

\[
\frac{du_{\parallel}}{dt} = -\frac{q}{m B^*_{\parallel}} B^* \left( \frac{\mu}{q \gamma} \nabla B - E^* \right),
\]

\[
\frac{du_{\perp}}{dt} = \frac{u_{\perp}}{2 B^*_{\parallel}} \nabla B \left( \frac{u_{\parallel}}{\gamma} B^* - c \hat{b} \times E^* \right),
\]
where
\[ \gamma(u) = \sqrt{1 + \frac{u^2}{c^2}} = \gamma(v) = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \]  
the Lorentz factor. Equations (59) remain the same for the relativistic case, as long as \( v \) is replaced by \( u \), such that e.g.
\[ \mu = \frac{m u^2}{2 B} = \frac{m \gamma^2 v^2}{2 B}. \]  

D.1.2 Modes of Integration

The model described in section 4 (§4.2), considers a stationary and homogeneous magnetic field \( B = B_0 \hat{e}_z \) inside the flux tube (such that \( \hat{e}_{\parallel} = \hat{e}_z \)). In this case, all sets of equations given so far are greatly simplified, so that it becomes much easier to pursue (most of) the analytical solutions instead of the numerical ones. Moreover, this access to analytical solutions allows for a whole different approach to treating this particular problem, using the Continuous-Time Random Walk Formalism. When referring to the modes of integration therefore, we distinguish between the basic modes of integration and the mixed modes of integration.

D.1.2.1 Basic Modes of Integration

(a) Numerical Integration, which simulates the realistic continuous-time situation by using a suitable numerical integrator with a time-step appropriate to each set of equations of motion, and the

(b) Integration Via a Continuous-Time Random Walk (see e.g. Vlahos et al., 2008). In order for this mode of integration to be understood, we need to set the frame in which it operates.

In a typical random walk problem, a particle at position \( r_i \) at a time \( t_i \) having velocity \( v_i \) is subjected to an abrupt change (an event with zero duration) in position \( \Delta r_i \) and/or velocity \( \Delta v_i \). If we define a sequence of such events for the particle to experience, it will perform a random walk in space and/or velocity space. The way the timing of the particle is defined, that is, the sequence of time intervals \( \{ \Delta t_i \} \) elapsing between successive steps, is crucial because it identifies the type of the random walk performed and therefore determines the model that is to be developed for any subsequent analysis (e.g. a diffusion model). It is also to be noted that the sequence of events describing the steps of the random walk can, and usually are, defined by means of probability distribution functions for the increments \( \{ \Delta r_i \} \), \( \{ \Delta v_i \} \) and \( \{ \Delta t_i \} \), so there lies the ability to introduce models using a variety of dependencies and correlations (or inter-correlations) to the random variables \( r_i = r_0 + i \Delta r_i \), \( v_i = v_0 + i \Delta v_i \) and \( t_i = t_0 + i \Delta t_i \).
In a classical random walk model, one usually defines the step increments \( \{ \Delta r_i \} \) and \( \{ \Delta \nu_i \} \) to be independent, e.g. \( \Delta r_i \) does not depend on any of the previous spatial increments, or any \( \{ \Delta \nu_k \}, \ k = 1, \ldots, i \) (the same applies to \( \Delta \nu_i \)). The time-steps \( \{ \Delta t_i \} \) are all taken to equal some \( \Delta t \), such that time acts as a simple counter. In such a case, a model for normal diffusion can be developed as long as \( \Delta t \) is sufficiently small, so that the modeling process can employ the necessary Taylor expansions.

In a continuous-time random walk, (CTRW) model, one considers the time-steps \( \Delta t_i \) to be a random variable. Depending on the precise way this random variable is defined, various CTRW models arise. According to (Vlahos et al., 2008), the waiting model is the one in which each \( \Delta t_i \) is taken from a certain distribution independent from the rest of the random variables. \( \Delta t_i \) here, is interpreted as the time interval the particle has to wait, before performing the \( i \)-th step of zero duration. In the velocity model I, each step’s spatial increment \( \Delta r_i \) is used to calculate the waiting time before that step is performed, as \( \Delta t_i = |\Delta r_i|/\nu \), where \( \nu = \text{const.} \) is an assumed constant velocity by which all steps are performed. Since we are here considering a random walk in velocity space as well, the complexity of the problem can be taken one step further, to include the velocity dynamics by calculating the waiting times according to \( \Delta t_i = |\Delta r_i|/\nu_i \), where \( \nu_i \) is the speed of the particle before the \( i \)-th step is performed. We call this model the velocity model II. What both velocity models do, is calculate (using a different approach) the duration of events that are otherwise considered to take place in no time at all. It is thus a matter of definition whether one will interpret \( \Delta t_i \) as waiting or traveling time. Note that in the frame of CTRW modeling it is possible, at least for the waiting and the velocity I models, to proceed analytically in the development of a theory that models anomalous diffusion.

In the model we have employed, the particles are considered to travel a distance \( \ell_i \) through which no force acts on them, then enter an energy dissipation region where an electric field \( E_i \) either accelerates or decelerates them, and exit that region after having traveled a distance \( d_i \).

In order to model this problem in a CTRW-like formalism as realistically as possible, we first need to realize that in this case, there are two kinds of events taking place. We define the first kind of event by identifying the particle’s free travel as the “waiting event” of the CTRW models. Unlike in the CTRW models however, this is not an event in which absolutely nothing happens space-wise; a distance \( \ell_i \) is traveled, in a time interval \( \tau_i \), the free-time. The second kind of event is the acceleration travel of the particle, which we identify as the actual “step” defined in the CTRW frame. This is the event during which the particle travels a distance \( d_i \) and is subjected to a velocity increment (or decrement) \( \Delta \nu_i \). Even though we may intend to keep considering this event as having zero duration in the timing of the particles, as in the CTRW models, the point of assigning a specific force acting
on a particle through a specific travel distance is to be able to calculate the exact velocity increment $\Delta \nu_i$ taking place in that region. Any system of the form

\[
\frac{d\nu}{dt} = \frac{F}{m}, \tag{64a}
\]
\[
\frac{dr}{dt} = \nu \tag{64b}
\]

however, for which $F$ and $\Delta r$ are known, forces us upon integration, to first compute $\Delta t_i$ before being able to calculate $\Delta \nu$. So regardless on whether we care to take into account the acceleration-time $\Delta t_i$ in the timing of the particles or not, we need to calculate it, as well as the free-time $\tau_i$, for the purpose of ensuring the consistency of our model.

As already mentioned, when the magnetic field is $B = B_0 \hat{e}_z$, the equations of motion are greatly simplified. For this case, we can derive the recurrent relations (a mapping) that generate the subsequent states of a particle based on its initial conditions and the sequences $\{\ell_i\}$, $\{E_i\}$ and $\{d_i\}$ by writing down the equations of motion for each channel described (Lorentz Force/Guiding Center, Non-Relativistic/Relativistic), in two forms: (i) For $E = 0$ in order to describe the free travels and (ii) For $E \neq 0$ to describe the acceleration travels. The final equations will turn out to have the form:

**FREE TRAVEL:**

\[
\nu_{i+1} = \nu_i, \tag{65a}
\]
\[
r_{i+1} = r_i + \ell_i, \tag{65b}
\]
\[
t_{i+1} = t_i + \tau_i, \tag{65c}
\]
\[
\ell_i^2 = [\Delta r_i(\tau_i)]^2 \Rightarrow f(\tau_i; \nu_i, \ell_i) = 0 \tag{65d}
\]

**ACCELERATION TRAVEL:**

\[
\nu_{i+1} = F(\nu_i, \Delta \nu_i(\Delta t_i)), \tag{66a}
\]
\[
r_{i+1} = r_i + d_i, \tag{66b}
\]
\[
t_{i+1} = t_i + \Delta t_i, \tag{66c}
\]
\[
d_i^2 = [\Delta r_i(\Delta t_i)]^2 \Rightarrow f(\Delta t_i; \nu_i, E_i, d_i) = 0 \tag{66d}
\]

Note that the source code includes all channels but one; the relativistic Lorentz Force CTRW channel was not built due to the high complexity of the equations involved. Also note that equation (66c) updates time, when it was stated that the acceleration-times may not be taken into account in the timing of the particles. The source code includes a special variable in which the timing of the particles is recorded. This way, it is possible to define the particle-timing in a variety of ways, through switches (see secondary switches §D.2). The
recurrent relations used in each of the three channels built in this code, are presented next.

**NON-RELATIVISTIC LORENTZ FORCE:**

**FREE TRAVEL:**

\[
v_{i+1}^x = v_i^x \cos \left( \frac{q B_0}{m c} \tau_i \right) + v_i^y \sin \left( \frac{q B_0}{m c} \tau_i \right),
\]

\[ (67a) \]

\[
v_{i+1}^y = v_i^y \cos \left( \frac{q B_0}{m c} \tau_i \right) - v_i^x \sin \left( \frac{q B_0}{m c} \tau_i \right),
\]

\[ (67b) \]

\[
v_{i+1}^z = v_i^z,
\]

\[ (67c) \]

\[
x_{i+1} = x_i - \frac{\Delta v_{i}^y}{q B_0/m c},
\]

\[ (67d) \]

\[
y_{i+1} = y_i + \frac{\Delta v_{i}^x}{q B_0/m c},
\]

\[ (67e) \]

\[
z_{i+1} = z_i + v_i^x \tau_i,
\]

\[ (67f) \]

\[
t_{i+1} = t_i + \tau_i,
\]

\[ (67g) \]

\[
\ell_i^2 = (v_i^z \tau_i)^2 + \frac{2 [(v_i^x)^2 + (v_i^y)^2]}{(q B_0/m c)^2} \left[ 1 - \cos \left( \frac{q B_0}{m c} \tau_i \right) \right]
\]

\[ (67h) \]
ACCELERATION TRAVEL:

\[
v_{i+1}^x = \frac{1}{q B_0/m c} \left[ \frac{q}{m} E_i^x + \left( \frac{q B_0}{m c} v_i^x - \frac{q}{m} E_i^y \right) \cos \left( \frac{q B_0}{m c} \Delta t_i \right) \right. \\
+ \left( \frac{q B_0}{m c} v_i^y + \frac{q}{m} E_i^x \right) \sin \left( \frac{q B_0}{m c} \Delta t_i \right) \right], \tag{68a}
\]

\[
v_{i+1}^y = \frac{1}{q B_0/m c} \left[ -\frac{q}{m} E_i^x + \left( \frac{q B_0}{m c} v_i^y + \frac{q}{m} E_i^y \right) \cos \left( \frac{q B_0}{m c} \Delta t_i \right) \right. \\
- \left( \frac{q B_0}{m c} v_i^x - \frac{q}{m} E_i^y \right) \sin \left( \frac{q B_0}{m c} \Delta t_i \right) \right], \tag{68b}
\]

\[
v_{i+1}^z = v_i^z + \frac{q}{m} E_i^z \Delta t_i, \tag{68c}
\]

\[
x_{i+1} = x_i + \frac{1}{(q B_0/m c)^2} \left[ \frac{q}{m} E_i^x + \frac{q B_0}{m c} \left( v_i^y + \frac{q}{m} E_i^y \Delta t_i \right) \right. \\
- \left( \frac{q B_0}{m c} v_i^x + \frac{q}{m} E_i^x \right) \cos \left( \frac{q B_0}{m c} \Delta t_i \right) \right. \\
+ \left( \frac{q B_0}{m c} v_i^y - \frac{q}{m} E_i^y \right) \sin \left( \frac{q B_0}{m c} \Delta t_i \right) \right], \tag{68d}
\]

\[
y_{i+1} = y_i + \frac{1}{(q B_0/m c)^2} \left[ -\frac{q}{m} E_i^y + \frac{q B_0}{m c} \left( v_i^x + \frac{q}{m} E_i^x \Delta t_i \right) \right. \\
+ \left( \frac{q B_0}{m c} v_i^y - \frac{q}{m} E_i^y \right) \cos \left( \frac{q B_0}{m c} \Delta t_i \right) \right. \\
+ \left( \frac{q B_0}{m c} v_i^x + \frac{q}{m} E_i^x \right) \sin \left( \frac{q B_0}{m c} \Delta t_i \right) \right], \tag{68e}
\]

\[
z_{i+1} = z_i + v_i^z \Delta t_i + \frac{1}{2} \frac{q}{m} E_i^z (\Delta t_i)^2, \tag{68f}
\]

\[
t_{i+1} = t_i + \Delta t_i, \tag{68g}
\]

\[
d_i^2 = [\Delta x_i(\Delta t_i)]^2 + [\Delta y_i(\Delta t_i)]^2 + [\Delta z_i(\Delta t_i)]^2 \tag{68h}
\]

RELATIVISTIC LORENTZ FORCE:

As already explained, this channel is not built due the (apparent by now) high complexity of the equations involved.
NON-RELATIVISTIC GUIDING CENTER:

FREE TRAVEL:

\[ v_{i+1}^\parallel = v_i^\parallel \]  \hspace{1cm} (69a)
\[ \nu_{i+1}^\perp = \nu_i^\perp \]  \hspace{1cm} (69b)
\[ x_{i+1} = x_i \]  \hspace{1cm} (69c)
\[ y_{i+1} = y_i \]  \hspace{1cm} (69d)
\[ z_{i+1} = z_i + \frac{v_i^\parallel}{|v_i^\parallel|} \ell_i, \]  \hspace{1cm} (69e)
\[ t_{i+1} = t_i + \tau_i, \]  \hspace{1cm} (69f)
\[ \tau_i = \frac{\ell_i}{v_i^\parallel} \]  \hspace{1cm} (69g)

ACCELERATION TRAVEL:

\[ v_{i+1}^\parallel = v_i^\parallel + \frac{q}{m} E_i^z \Delta t_i, \]  \hspace{1cm} (70a)
\[ \nu_{i+1}^\perp = \nu_i^\perp, \]  \hspace{1cm} (70b)
\[ x_{i+1} = x_i + \frac{c}{B_0} E_i^y \Delta t_i, \]  \hspace{1cm} (70c)
\[ y_{i+1} = y_i - \frac{c}{B_0} E_i^x \Delta t_i, \]  \hspace{1cm} (70d)
\[ z_{i+1} = z_i + v_i^\parallel \Delta t_i + \frac{1}{2} \frac{q}{m} E_i^z (\Delta t_i)^2, \]  \hspace{1cm} (70e)
\[ t_{i+1} = t_i + \Delta t_i, \]  \hspace{1cm} (70f)
\[ d_i^2 = \left( \frac{1}{2} \frac{q}{m} E_i^z \right)^2 (\Delta t_i)^4 + \frac{q}{m} E_i^z v_i^\parallel (\Delta t_i)^3 \]  
\[ + \left[ (v_i^\parallel)^2 + \left( \frac{c}{B_0} \right)^2 [(E_i^x)^2 + (E_i^y)^2] \right] (\Delta t_i)^2 \]  \hspace{1cm} (70g)
RELATIVISTIC GUIDING CENTER:

FREE TRAVEL:

\( u_{i+1}^\parallel = u_i^\parallel, \)  
\( u_{i+1}^\perp = u_i^\perp, \)  
\( x_{i+1} = x_i, \)  
\( y_{i+1} = y_i, \)  
\( z_{i+1} = z_i + \frac{u_i^\parallel}{|u_i^\parallel|} \ell_i, \)  
\( t_{i+1} = t_i + \tau_i, \)  
\( \tau_i = \frac{\ell_i}{u_i^\parallel} \)  

ACCELERATION TRAVEL:

\( u_{i+1}^\parallel = u_i^\parallel + \frac{q}{m} E_i^z \Delta t_i, \)  
\( u_{i+1}^\perp = u_i^\perp, \)  
\( x_{i+1} = x_i + \frac{c}{B_0} E_i^y \Delta t_i, \)  
\( y_{i+1} = y_i - \frac{c}{B_0} E_i^x \Delta t_i, \)  
\( z_{i+1} = z_i + \frac{c^2}{m E_i^z} (\gamma_{i+1}(\Delta t_i) - \gamma_i), \)  
\( t_{i+1} = t_i + \Delta t_i, \)  
\( d_i^2 = \left( \frac{c}{B_0} \right)^2 \left[ (E_i^x)^2 + (E_i^y)^2 \right] (\Delta t_i)^2 
+ \left( \frac{c^2}{m E_i^z} \right)^2 (\gamma_{i+1}(\Delta t_i) - \gamma_i), \)  
\( \gamma_{i+1}(\Delta t_i) = \sqrt{1 + \frac{(u_i^\parallel)^2 + (u_i^\perp)^2}{c^2}} \)

Note that in this relativistic case, the evolving velocity is the four-velocity \( u = \gamma \nu, \) so care needs to be taken whenever quantities dependent on the velocity \( \nu \) are calculated.
D.1.2.2 The Deep/Shallow Zeros Numerical Instability

As may have become obvious from equations (67h), (68h), (70g), and (72g), one of the most challenging parts in all the processes concerning the mode of integration via a CTRW, is to find an accurate way of calculating the acceleration-times $\Delta t_i$ (and for the Lorentz channel also the free-times $\tau_i$) as accurately as possible. Even though equation (70g) for example, may seem easy to solve, it is in fact the current velocity of the particle and the choices made for the electric fields and the acceleration lengths that determine whether this otherwise analytical solution will keep returning accurate results as the particle trajectories evolve in time.

The problems rise whenever a particle that has already attained a relatively high velocity, meets an UCS of small size with an electric field of great strength, tending to further accelerate the particle. In these cases, the derivatives of all functions $f(\tau_i$ or $\Delta t_i; \ldots$ parameters) $= 0$ (see equations 65 and 66) with respect to the time interval of interest take extreme values (either large or small) near their roots [a numerical problem called deep (or shallow) zeros (see e.g. Protopopov, 2007), that any root-finding numerical method using derivatives is bound to fail. This is why we were forced to use a method (indeed optimal whenever the functional form of the equation one is trying to solve is not known!) that does not use derivatives.

What may not have become obvious however, is that, for the same reasons, the mode of numerical integration suffers as well. When a particle gets in a situation like the one just described, the time interval of interest becomes too small; much smaller than the constant time-step chosen for the numerical integrator to use. The code then is unable to detect the precise instants in time when the particle either enters or exits an UCS. This means that situations like that need to be predicted as the code runs, so that the time-step can be adjusted to be less than the time interval of interest. In turn, what this means is that these difficult-to-handle equations need to be solved for the mode of numerical integration as well. The problem of accuracy (in entering/exiting the UCSs) can be addressed this way, but the code’s computational cost becomes unaffordable. This is what gave rise to the mode of integration via a CTRW.

D.1.2.3 Mixed Modes of Integration

Unless somehow the physics involved in the problem at hand (such as the need to employ a more realistic magnetic field configuration) demands that we use the numerical mode of integration, this code is better off running in the mode of integration via a CTRW. Having gone into the trouble of building all these channels however, we thought it would be wise
to complete this job by building the extra *mixed modes of integration* which, by the way, required no real extra effort. So the code includes switches through which the following mixed modes of integration are accessed:

(c) *Numerical Integration of the Free Travels and Integration Via a CTRW of the Acceleration Travels.*

(d) *Numerical Integration of the Acceleration Travels and Integration Via a CTRW of the Free Travels.*

### D.1.3 System Boundaries

(a) **Closed Box:** When the *closed box* switch is turned on, the space in which the particles trajectories evolve is finite, but the code is monitoring their positions at all times. When a particle is detected outside the bounded box, its position abruptly changes back to its initial position. This channel was built simply to enable us to realize certain text-book cases, in order to verify the validity of the calculations performed throughout the source code.

(b) **Open Box:** When this switch is turned on, the space in which the particles move is again finite and the code still monitors their positions at all times. But this time, if a particle is detected outside the bounded box, its state (position and velocity-energy) is being recorded, before it gets dropped out of the running integration mode. This way, we are able to observe the behavior of different particle populations, namely the ones that remain inside the configuration box and a number of particle populations that leave the box moving at different directions (generally top, bottom, sides).

### D.2 Secondary Switches

#### D.2.1 Test-Particle Populations

(a) **Test - Particle Species:** The switch controlling the species of the test-particles whose trajectories are simulated, does so through their rest mass and electric charge. Determining the species of the test-particles means that the *entire population* of the test-particles is of that specified kind.

(b) **Background & Injected Test-Particles:** The source code provides the user with the option to split the test-particle population into two different populations, based on their initial conditions. We can have the entire test-particle population randomly distributed inside the configuration box, thus simulating *charged particles of the ambient plasma* (background), or we can assign some to initiate their trajectories at the top base of the configuration box with initial velocities directed downwards, in order to simulate an *injected population of charged particles.*
D.2.2 PDF Models

The basic principle by which the proposed model operates, is one in which three probability
distribution functions (PDFs) $P(\ell)$, $P(E)$ and $P(\Delta x)$ determine the sequence of ‘events’ expe-
rienced by the particles, in terms of the free travel distances $\ell$, the electric field strength $E$ of
the energy dissipation region they encounter, and its size $\Delta x$. Even though it is not extremely
laborious to implement the code with different PDFs in order to perform experiments, the code
includes three PDF models accessed through a switch.

D.2.3 Electric Field Switch

One of the future plans for this code is to implement collisions. In order to test the collision
operator employed however, one needs to ‘turn off’ the electric fields acting on the particles. The
code was built from the beginning in such a way, so that all modes of integration can run in an
electric field-free mode.

D.2.4 Timing of the Particles

As already implied in the description of the CTRW formalism of the model, the timing of the
particles plays a crucial role in enabling us to realize situations for which analytically expected
results can be obtained. It was also stated that the acceleration times $\Delta t_i$ may not be taken into
account in the timing of the particles. Except for this switch (to omit $\Delta t_i$ from the timing), the
code also includes another switch, through which one can only take into account the acceleration
times (and omit the free-times $\tau_i$).

Having discussed the key features of the source code, let us now briefly run through the
process of initialization and execution of a typical simulation.

- Determine the number of particles of the simulation, their species (and maybe split them
  in different populations).
- Choose the PDF model that is to be used (set its parameters).
- Set the desired mode of integration. Also choose the set of equations to be integrated
  (Lorentz Force/Guiding Center, Non-Relativistic/Relativistic) and whether the system bound-
  aries are to be closed or open.
- Determine the total particle-time of the simulation and then, depending on the mode of
  integration, set the total number of integration steps required so that the particles will reach
  the final time chosen (Note that in the mode of integration via a CTRW each particle has
  its own time).
• Adjust certain parameters according to the total number of integration steps, in order to collect an appropriate set of various quantities of interest.

• Make sure that the test-particles are initialized appropriately (the default initial conditions for the particles is for them to be randomly distributed inside the configuration box, having randomly directed velocities with their components following a Maxwellian distribution of some adjustable temperature).

• Execute the program.

One of the features of the code any user would be thrilled to discover, is that soon after the initiation of the execution, the code informs the user of the expected cpu-time required for the simulation to conclude. Taking into account the total number of particles simulated and the number of threads employed for the simulation, the code performs an averaging process using the first 100 particles ran and estimates the length of the simulation in seconds. These calculations are performed even if some of the necessary conditions are not met (such as having the simulation include less than 100 particles) and the user is simply informed that the estimated time may not be accurate.

D.3 Results

We close this description by discussing the results delivered by this code.

• Isolated Test-Particles:
  – Trajectories inside the configuration box.
  – Sequence of the electric fields encountered (magnitudes).
  – Response of the particles through their kinetic energy in time.

• Averaged/Statistical Quantities:
  – Energy:
    * The particles’ average kinetic energy in time.
    * Initial & final kinetic energy distribution of the particles (depending on the system boundaries, there may be more than one final distributions, corresponding to the escaping particle populations).
    * The code can also collect frames of the evolving kinetic energy distribution in time, in order to create a video.
    * Final kinetic energy distributions along various predetermined heights of the configuration box.
– Diffusion:

* Mean square displacement in velocity (parallel, perpendicular components and magnitude) in time.
* Diffusion coefficient in time (parallel component).
* Mean displacement in velocity, in time.
* Drift coefficient in time (parallel component).

• Test - Files: Currently the code can be asked (through a switch) to create various test-files. These files can be used, for example, to test the random number generators used throughout, and also to uncover some not so obvious dependencies or correlations among quantities such as steps in time and velocity etc.
E  Generating Random Numbers
Distributed According to a Specified Manner

E.1 The Inverse Transformation Method

**Proposition:** Let \( Y \) be a scalar random variable with a uniform distribution in \([0, 1]\), denoted \( Y \sim U(0, 1) \) and \( X \) another scalar random variable. If \( F^{-1}_x : (0, 1) \to \mathbb{R} \) is the inverse of a cumulative distribution function (CDF) \( F_x \) with probability density function (PDF) \( f_x \), then setting
\[
Y = F_x(X) \Rightarrow X = F^{-1}_x(Y),
\]
(73)
\( X \) has CDF \( F_x \).

E.2 Uniformly Distributed Random Points Inside a Cylinder

We wish to uniformly distribute a set of points inside a cylindrical volume of radius \( R \) and height \( L \). Our random variables in Cartesian coordinates are \( x, y \) and \( z \) and in cylindrical coordinates \( r, \phi \) and \( z \). The transformation from the cylindrical coordinate system to the Cartesian one is
\[
x = r \cos(\phi) \\
y = r \sin(\phi) \\
z = z,
\]
(74)
where \( x^2 + y^2 \leq R^2 \) and \( 0 \leq z \leq L \) or equivalently \( r \leq R \), \( 0 \leq \phi \leq 2\pi \) and \( 0 \leq z \leq L \).

In order to use cylindrical coordinates and random number generators (RNGs) providing uniformly distributed numbers in \([0, 1]\), we must first find a suitable transformation relating \( r, \phi \) and \( z \) to the random numbers \( \rho, \phi \) and \( zeta \), all three \( \sim U(0, 1) \), produced by the RNGs. For that purpose, we will use the proposition stated above. Before calculating the CDFs required by that proposition, we must first take into account the fact that \( r, \phi \) and \( z \) are a curvilinear system’s coordinates. Assigning to the multivariate random variable \((r, \phi, z)^T\) a PDF \( f(r, \phi, z) \), the normalization condition imposed by probability theory is
\[
\int_V f(r, \phi, z) \, dV = \int_0^R \int_0^{2\pi} \int_0^L f(r, \phi, z) \, r \, dr \, d\phi \, dz = 1.
\]
(75)
and the CDF expressing the probability that \((r', \phi', z')^T \leq (r, \phi, z)^T\) is given by equation
\[
F(r, \phi, z) \equiv P[(r', \phi', z')^T \leq (r, \phi, z)^T] = \int_0^r \int_0^{\phi} \int_0^z f(r', \phi', z') \, r' \, dr' \, d\phi' \, dz'.
\]
(76)
Assuming that \( r, \phi, z \) are statistically independent, we can write \( f(r, \phi, z) \) as the product of
three PDFs $f_r(r)$, $f_\phi(\phi)$ and $f_z(z)$ each one dependent on only one of the scalar random variables $r$, $\phi$ and $z$, i.e.

$$f(r, \phi, z) = f_r(r) \cdot f_\phi(\phi) \cdot f_z(z).$$  \hspace{1cm} (77)

The normalization condition (75) now becomes

$$\int_0^R r f_r(r) \, dr \cdot \int_0^{2\pi} f_\phi(\phi) \, d\phi \cdot \int_0^L f_z(z) \, dz = 1.$$  \hspace{1cm} (78)

The CDF $F(r, \phi, z)$ can also be written as the product of three CDFs each dependent on one of the scalar random variables, namely

$$F(r, \phi, z) = F_r(r) \cdot F_\phi(\phi) \cdot F_z(z),$$  \hspace{1cm} (79)

where

$$F_r(r) \equiv P(r' \leq r) = \int_0^r r' f_r(r') \, dr'$$  \hspace{1cm} (80a)

$$F_\phi(\phi) \equiv P(\phi' \leq \phi) = \int_0^\phi f_\phi(\phi') \, d\phi'$$  \hspace{1cm} (80b)

$$F_z(z) \equiv P(z' \leq z) = \int_0^z f_z(z') \, dz'.$$  \hspace{1cm} (80c)

Note that if we were to directly define $F_r(r)$, $F_\phi(\phi)$ and $F_z(z)$ using the general formula

$$F(s) \equiv P(s' \leq s) = \int_0^s f(s') \, ds'$$  \hspace{1cm} (81)

defining the CDF in the case of a single scalar random variable $S$, we would have missed including the $r'$ in the integrand of equation (80a) originating from the definition $dV = r \, dr \, d\phi \, dz$ of the volume element when a cylindrical coordinate system is considered.

The PDF $f_s$ describing the uniform distribution of a scalar random variable $S$ in $[a, b]$ is

$$f_s(s) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq s \leq b \\ 0 & \text{for } s < a \text{ or } s > b \end{cases}$$  \hspace{1cm} (82)

or restricting the $s$ domain in $[a, b]$, simply

$$f_s(s) = \frac{1}{b-a}, \quad a \leq s \leq b.$$  \hspace{1cm} (83)
Since we want \( r \sim U(0, R), \phi \sim U(0, 2\pi) \) and \( z \sim U(0, L) \), it should be

\[
\begin{align*}
f_r(r) &= \frac{C}{R - 0}, \quad 0 \leq r \leq R \\
f_\phi(\phi) &= \frac{1}{2\pi - 0}, \quad 0 \leq \phi \leq 2\pi \\
f_z(z) &= \frac{1}{L - 0}, \quad 0 \leq z \leq L
\end{align*}
\]

where \( C \) is a normalization constant (it could be considered absorbed in any of the other two CDFs, but as will soon be clear, it is preferable that \( f_r \) absorbs it).

Substituting equations (84) in the normalization condition (78) we find that

\[
C = \frac{2}{R}
\]

and substituting equations (84) again, into the corresponding equations (80), we obtain, after performing the integrations, the three CDFs

\[
\begin{align*}
F_r(r) &= \frac{r^2}{R^2} \\
F_\phi(\phi) &= \frac{\phi}{2\pi} \\
F_z(z) &= \frac{z}{L}
\end{align*}
\]

We are now ready to perform the inversion indicated in (73). Setting \( \rho, \phi \) and \( \zeta \) equal to \( F_r, F_\phi \) and \( F_z \) respectively and solving for \( r, \phi \) and \( z \), we obtain a transformation defined by the following equations

\[
\begin{align*}
r &= R \sqrt{\rho} \\
\phi &= 2\pi \phi \\
z &= L \zeta
\end{align*}
\]

If we furthermore wish to directly relate \( \rho, \phi \) and \( \zeta \) to the Cartesian coordinates \( x, y \) and \( z \), using equations (74) and (87) we obtain

\[
\begin{align*}
x &= R \sqrt{\rho} \cos(2\pi \phi) \\
y &= R \sqrt{\rho} \sin(2\pi \phi) \\
z &= L \zeta
\end{align*}
\]
E.3 Position Vectors Originating at the Center of a Sphere, Uniformly Distributed Inside it.

We wish to uniformly distribute, in both length and direction, a set of position vectors inside a sphere of radius $R$, centered at the origin of a Cartesian coordinate system. Consider the unit vector $\hat{r}$ of a spherical coordinate system. A position vector $\vec{r}$ of length $r$ can be written as $\vec{r} = r \hat{r}$ or, expressing $\hat{r}$ in terms of the Cartesian basis vectors $\hat{e}_x$, $\hat{e}_y$ and $\hat{e}_z$, as

$$\vec{r} = r \sin(\theta) \cos(\phi) \hat{e}_x + r \sin(\theta) \sin(\phi) \hat{e}_y + r \cos(\theta) \hat{e}_z,$$

(89)

where $\theta \in [0, \pi]$ the polar angle and $\phi \in [0, 2\pi]$ the azimuthal angle of a spherical coordinate system.

In order to use the spherical coordinates $r$, $\theta$, $\phi$ and random number generators providing uniformly distributed numbers in $[0, 1]$, we must first find a suitable transformation relating $r$, $\theta$ and $\phi$ to the random numbers $\rho$, $\theta$ and $\phi$, all three $\sim U(0, 1)$, produced by the RNGs. For that purpose, we will use the proposition stated at the beginning of the section. Before calculating the CDFs required by that proposition, we must first take into account the fact that $r$, $\theta$ and $\phi$ are a curvilinear system’s coordinates. Assigning to the multivariate random variable $(r, \theta, \phi)^T$ a PDF $f(r, \theta, \phi)$, the normalization condition imposed by probability theory is

$$\int_V f(r, \theta, \phi) \, dV = \int_0^R \int_0^\pi \int_0^{2\pi} f(r, \theta, \phi) \, r^2 \sin(\theta) \, dr \, d\theta \, d\phi = 1.$$

(90)

and the CDF expressing the probability that $(r', \theta', \phi')^T \leq (r, \theta, \phi)^T$ is given by equation

$$F(r, \theta, \phi) \equiv P[(r', \theta', \phi')^T \leq (r, \theta, \phi)^T] = \int_0^r \int_0^\pi \int_0^{2\pi} f(r', \theta', \phi') \, r'^2 \sin(\theta') \, dr' \, d\theta' \, d\phi'.$$

(91)

Assuming that $r$, $\theta$, $\phi$ are statistically independent, we can write $f(r, \theta, \phi)$ as the product of three PDFs $f_r(r)$, $f_\theta(\theta)$ and $f_\phi(\phi)$ each one dependent on only one of the scalar random variables $r$, $\theta$ and $\phi$, i.e.

$$f(r, \theta, \phi) = f_r(r) \cdot f_\theta(\theta) \cdot f_\phi(\phi).$$

(92)

The normalization condition (90) now becomes

$$\int_0^R r^2 f_r(r) \, dr \cdot \int_0^\pi f_\theta(\theta) \, d\theta \cdot \int_0^{2\pi} f_\phi(\phi) \, d\phi = 1,$$

(93)

The CDF $F(r, \theta, \phi)$ can also be written as the product of three CDFs each dependent on one of the scalar random variables, namely

$$F(r, \theta, \phi) = F_r(r) \cdot F_\theta(\theta) \cdot F_\phi(\phi),$$

(94)
where

\[ F_r(r) \equiv P(r' \leq r) = \int_0^r r'^2 f_r(r') \, dr' \]  \hspace{1cm} (95a)

\[ F_\theta(\theta) \equiv P(\theta' \leq \theta) = \int_0^\theta f_\theta(\theta') \sin(\theta') \, d\theta' \]  \hspace{1cm} (95b)

\[ F_\phi(\phi) \equiv P(\phi' \leq \phi) = \int_0^\phi f_\phi(\phi') \, d\phi'. \]  \hspace{1cm} (95c)

The PDF \( f_s \) describing the uniform distribution of a scalar random variable \( S \) in \([a, b] \), restricting the \( s \) domain in \([a, b] \), is given by equation (83). Since we want \( r \sim U(0, R) \), \( \theta \sim U(0, \pi) \) and \( \phi \sim U(0, 2\pi) \), it should be

\[ f_r(r) = \frac{C_r}{R - 0}, \quad 0 \leq r \leq R \]

\[ f_\theta(\theta) = \frac{C_\theta}{\pi - 0}, \quad 0 \leq \theta \leq \pi \]  \hspace{1cm} (96)

\[ f_\phi(\phi) = \frac{1}{2\pi - 0}, \quad 0 \leq \phi \leq 2\pi \]

Let \( C_r \cdot C_\theta = C \) be a normalization constant (\( C \) could be considered absorbed in any two out of the three CDFs (or even in all three), but as will soon be clear, it is preferable that \( f_r \) and \( f_\theta \) absorb it).

Substituting equations (96) in the normalization condition (93) we find that

\[ C_r = \frac{3}{R^2} \quad \text{and} \quad C_\theta = \frac{\pi}{2} \]  \hspace{1cm} (97)

and substituting equations (96) again, into the corresponding equations (95), we obtain, after performing the integrations, the three CDFs

\[ F_r(r) = \frac{r^3}{R^3} \]  \hspace{1cm} (98a)

\[ F_\theta(\theta) = \frac{1}{2}(1 - \cos\theta) \]  \hspace{1cm} (98b)

\[ F_\phi(\phi) = \frac{\phi}{2\pi}. \]  \hspace{1cm} (98c)

We are now ready to perform the inversion indicated in (73). Setting \( rho \), \( theta \) and \( phi \) equal to \( F_r \), \( F_\theta \) and \( F_\phi \) respectively and solving for \( r \), \( \theta \) and \( \phi \), we obtain a transformation defined by
the following equations

\[ r = R \rho^{1/3} \]  

(99a)

\[ \theta = \arccos(1 - 2 \theta) \]  

(99b)

\[ \phi = 2\pi \phi. \]  

(99c)

### E.4 Power-law Random Variable.

We wish to obtain a set of values \( s \) of a random variable \( S \) with a power-law PDF \( f_s \) of the form

\[ f_s(s) = A s^{-\alpha}, \quad s_l \leq s \leq s_r. \]  

(100)

In order to use a random number generator providing uniformly distributed numbers in \([0, 1]\), we must first find a suitable transformation relating \( s \) to the random number \( ess \sim U(0, 1) \) provided by the RNG. For that purpose, we will use the proposition stated at the beginning of the section.

Assigning to the scalar random variable \( S \) the PDF \( f_s \) of equation (100), the normalization condition imposed by probability theory can provide us with the normalization constant \( A \)

\[
\int_{s_l}^{s_r} f_s(s) \, ds = 1 \Rightarrow \\
\int_{s_l}^{s_r} A s^{-\alpha} \, ds = 1 \Rightarrow \\
A \int_{s_l}^{s_r} [s^{1-\alpha}] = 1 \Rightarrow \\
A = \frac{1 - \alpha}{s_r^{1-\alpha} - s_l^{1-\alpha}}. 
\]  

(101)

The CDF expressing the probability that \( s' \leq s \in [s_l, s_r] \) is given by equation

\[ F_s(s) \equiv P[s' \leq s] = \int_{s_l}^{s} f_s(s') \, ds'. \]  

(102)

Substituting \( f_s \) from (100) and \( A \) from (101) into equation (102), we have

\[
F_s(s) = \frac{1 - \alpha}{s_r^{1-\alpha} - s_l^{1-\alpha}} \int_{s_l}^{s} s'^{-\alpha} \, ds' \\
= \frac{1 - \alpha}{s_r^{1-\alpha} - s_l^{1-\alpha}} \frac{[s'^{1-\alpha}]_{s_l}}{1-\alpha} \\
= \frac{s_r^{1-\alpha} - s_l^{1-\alpha}}{s_r^{1-\alpha} - s_l^{1-\alpha}}. 
\]  

(103)

We are now ready to perform the inversion indicated in (73). Setting \( ess \) equal to \( F_s \) and
solving for \( s \), we obtain a transformation defined by the following equation
\[
s = \left[ (s^1 - \alpha - s^1) e^{s} + s^1 - \alpha \right]^{\frac{1}{1 - \alpha}}. \tag{104}
\]

### E.5 Double Power-law Random Variable.

We wish to obtain a set of values \( s \) of a random variable \( S \) with a double power-law PDF \( f_s \) of the form
\[
f_s(s) = \begin{cases} 
A_1 s^{-\alpha_1}, & s_l \leq s \leq s\text{thres} \\
A_2 s^{\alpha_2 - \alpha_1} s^{-\alpha_2}, & s\text{thres} \leq s \leq s_r 
\end{cases} \tag{105}
\]

In order to use a random number generator providing uniformly distributed numbers in \([0, 1]\), we must first find a suitable transformation relating \( s \) to the random number \( \epsilon_s \sim U(0, 1) \) provided by the RNG. For that purpose, we will use the proposition stated at the beginning of the section.

Assigning to the scalar random variable \( S \) the PDF \( f_s \) of equation (105), the normalization condition imposed by probability theory can provide us with a relation between the two normalization constants \( A_1 \) and \( A_2 
\[
\int_{s_l}^{s_r} f_s(s) \, ds = 1 \Rightarrow \int_{s_l}^{s\text{thres}} A_1 s^{-\alpha_1} \, ds + \int_{s\text{thres}}^{s_r} A_2 s^{\alpha_2 - \alpha_1} s^{-\alpha_2} \, ds = 1 \Rightarrow \\
\frac{A_1}{1 - \alpha_1} [s^{1 - \alpha_1}]_{s_l}^{s\text{thres}} + \frac{A_2}{1 - \alpha_2} s^{\alpha_2 - \alpha_1} s\text{thres}^{1 - \alpha_2} = 1 
\tag{106}
\]

The CDF expressing the probability that \( s' \leq s \in [s_l, s_r] \) is given by an equation of the form
\[
F_s(s) \equiv P[s' \leq s] = \begin{cases} 
\int_{s_l}^{s} f_s(s') \, ds', & s_l \leq s \leq s\text{thres} \\
\int_{s_l}^{s\text{thres}} f_s(s') \, ds' + \int_{s\text{thres}}^{s} f_s(s') \, ds', & s\text{thres} \leq s \leq s_r 
\end{cases} \tag{107}
\]

Substituting \( f_s \) from equation (105) into equation (107) we have
\[
F_s(s) \equiv P[s' \leq s] = \begin{cases} 
\int_{s_l}^{s} \frac{A_1}{1 - \alpha_1} [s^{1 - \alpha_1}]_{s_l}^{s} \, ds', & s_l \leq s \leq s\text{thres} \\
\int_{s_l}^{s\text{thres}} \frac{A_1}{1 - \alpha_1} [s^{1 - \alpha_1}]_{s_l}^{s\text{thres}} + \frac{A_2}{1 - \alpha_2} s^{\alpha_2 - \alpha_1} s\text{thres}^{1 - \alpha_2} \, ds, & s\text{thres} \leq s \leq s_r 
\end{cases} \tag{108}
\]

The normalization constants \( A_1 \) and \( A_2 \) need to be substituted in the above equation and usually, when the normalization constant is only one, this is done using the normalization condition (106). In this case, we must also use the continuity condition, namely the fact that, at \( s\text{thres} \), the two branches of the piece wise function (105) must have the same value. We can then solve the
system of equations (106) and
\[ A_1 s_{\text{thres}}^{-\alpha_1} = A_2 s_{\text{thres}}^{\alpha_2 - \alpha_1} s_{\text{thres}}^{-\alpha_2}, \]  
(109)
in order to obtain \( A_1 \) and \( A_2 \). From equation (109) it follows that \( A_1 \) and \( A_2 \) must be equal, so we set \( A_1 = A_2 = A \). From equation (106), \( A \) turns out to be
\[ A = \frac{1}{\frac{G_1}{1-\alpha_1} + \frac{G_2 s_{\text{thres}}^{\alpha_2 - \alpha_1}}{1-\alpha_2}}, \]  
(110)
or, for later convenience
\[ A_1 = \frac{1}{G_1 + B G_2}, \]  
(111a)
\[ A s_{\text{thres}}^{\alpha_2 - \alpha_1} = \frac{B}{G_1 + B G_2}, \]  
(111b)
where we have set
\[ G_1 = [s_{l}^{1-\alpha_1}]^{s_{\text{thres}}}_{s_{l}} \]  
(112a)
\[ G_2 = [s_{l}^{1-\alpha_2}]^{s_{r}}_{s_{\text{thres}}} \]  
(112b)
\[ B = \frac{1-\alpha_1}{1-\alpha_2} s_{\text{thres}}^{\alpha_2 - \alpha_1} \]  
(112c)
for ease of notation.

The CDF \( F_s(s) \) is now given by
\[ F_s(s) \equiv P[s' \leq s] = \begin{cases} [s_{l}^{1-\alpha_1}]^{s_{\text{thres}}}_{s_{l}} & \text{,} \quad s_{l} \leq s \leq s_{\text{thres}} \\ \frac{G_1}{G_1 + B G_2} [s_{l}^{1-\alpha_2}]^{s_{\text{thres}}}_{s_{\text{thres}}} & \text{,} \quad s_{\text{thres}} \leq s \leq s_{r} \end{cases} \]  
(113)

We are now ready to perform the inversion indicated in equation (73). Setting \( ess \) equal to \( F_s \) and solving for \( s \), we obtain a transformation defined by the following equation
\[ s = \begin{cases} [s_{l}^{1-\alpha_1} + (G_1 + B G_2) \ ess]^{1-\alpha_1}_{s_{l}} & \text{,} \quad s_{l} \leq s \leq s_{\text{thres}} \\ [s_{\text{thres}}^{1-\alpha_2} - \frac{G_1}{B} + \frac{G_1 + B G_2}{B} \ ess]^{1-\alpha_2}_{s_{\text{thres}}} & \text{,} \quad s_{\text{thres}} \leq s \leq s_{r} \end{cases} \]  
(114)
What we notice in the later equation (114), is that the limit values 0 and 1 of the uniform random number \( ess \sim U(0, 1) \) do not restrict the values of \( s \) to the intervals \([s_{l}, s_{\text{thres}}]\) and \([s_{\text{thres}}, s_{r}]\)
corresponding to each of its two branches, unless $ess$ is restricted itself. The value $ess_{\text{thres}}$ of $ess$ for which the above conditions are fulfilled is

$$ess_{\text{thres}} = \frac{G_1}{G_1 + B G_2}. \quad (115)$$

Equation (114) can now be rewritten in the more accurate form

$$s = \begin{cases} 
  \left[ s_1^{1-\alpha_1} + (G_1 + B G_2) \ ess \right]^{\frac{1}{1-\alpha_1}}, & ess \leq ess_{\text{thres}}, \quad s_1 \leq s \leq s_{\text{thres}} \\
  \left[ s_{\text{thres}}^{1-\alpha_2} - \frac{G_1}{B} + \frac{G_1 + B G_2}{B} \ ess \right]^{\frac{1}{1-\alpha_2}}, & ess \geq ess_{\text{thres}}, \quad s_{\text{thres}} \leq s \leq s_r.
\end{cases} \quad (116)$$
References


