

BIRKELAND CURRENTS DERIVED FROM
EISCAT MEASUREMENTS:
TWO CASE STUDIES.

by

Klaus Normann

THESIS

for the degree of

MASTER OF SCIENCE



Faculty of Mathematics and Natural Sciences
University of Oslo

March 2018

”And those who were seen dancing
were thought to be insane
by those who could not
hear the music”

Abstract

Measurements of ion velocities by the Svalbard EISCAT Radar (ESR) has been used to calculate Birkeland current densities. Birkeland currents are field-aligned currents flowing along Earth's magnetic field lines at high latitudes. These calculated densities has been analyzed together with data from the Advanced Composition Explorer satellite (ACE) containing values of parameters in the solar wind and interplanetary magnetic field (IMF). A connection between larger Birkeland current densities and changes in the IMF parameters B_y or B_z is revealed.

Acknowledgements

Most of all I would like to thank two of my friends for helping me through these last stages, you know who you are.

I would also like to thank the people at the Space Physics group for not giving up on me. My co-students for the company, and an extra thanks to Vigdis for solving some programming issues in minutes which probably would have taken me hours.

A big thank you goes to my supervisor Lasse Clausen for showing flexibility during a sometimes bumpy ride.

Finally, a big, big thank you to the student advisors, especially Grete for always holding a door open and see openings where I saw closure.

Thank you all.

List of Figures

1.1	Top panel: Single particle motion with magnetic field out of the paper. Bottom panel: Single particle motion with magnetic field out of the paper and an electric field downwards.	6
1.2	The Sun's interior and exterior [55]	17
1.3	The origin of the fast and slow solar wind, [56]	18
1.4	Schematic figure of the magnetosphere showing the main plasma regions and current systems, [Russell and Luhmann, 1997].	22
1.5	Drawing of the magnetospheric polar region, [57]	23
1.6	"Magnetic field geometry for the SweetParker [reconnection] model. Oppositely directed field lines are brought together and reconnect in a diffusion layer (orange). The plasma is heated by Ohmic dissipation at the diffusion region and accelerated by the pressure gradient and the tension force. The field line diffuses over the half-width of the diffusion layer, δ , which is much smaller than the system size, $2L$." Figure from [31]	24
1.7	Dungey cycle, [33]	25
1.8	Region 1 and region 2 currents, [51].	26
1.9	Sketch of the polar cap current circuit. Birkeland field-aligned currents in color. Red going <i>up</i> , out of the ionosphere. Blue going <i>down</i> , into the ionosphere [7].	26
1.10	The different layers of the ionosphere marked in light gray with typical latitude in [km].	27
1.11	Height dependence of different molecules in the ionosphere, quiet solar conditions <i>somecithere</i>	28
	a Electron density	28
	b Ion density	28
1.12	"Sketches of the northern hemisphere ionospheric flow ordered according to the direction of the IMF. The arrowed solid lines show the plasma streamlines, [...]", figure from [8].	29
1.13	Schematic figure of the ionospheric Dungey cycle plasma flow for southward IMF. Solid lines are the plasma streamline, and the direction of the electric field is given by the short arrows, from [7].	29

1.14	IMF data distribution showing the IMF B_Y and B_Z values to find the B_T , used in figures 1.15 and 1.16, as concentric circles, from [62].	30
1.15	Electrical potentials with convection pattern represented by dotted lines, from [62].	31
1.16	Electrical potentials with convection pattern represented by dotted lines, from [62]. The values of IMF B_Y and B_Z are larger than in figure 1.15.	32
1.17	The ion- and electron gyrofrequency vs the collisional frequencies of both species, picture from [44].	32
1.18	Schematic illustration showing the ion velocity and the associated field-aligned Birkeland current due to a velocity difference. Blue arrow indicates ion velocity is toward radar, red means away. Blue circles with cross means the FAC is going downward, red circle with dot means its going upward, adaptation from [63].	33
1.19	An average configuration of Birkeland currents in the polar ionosphere, times are in local time. Region 1 currents flow out of the ionosphere on dusk side and into the ionosphere on dawn side, while Region 2 currents flow in opposite direction, figure from [28].	33
1.20	Two examples of Birkeland current distribution in the Northern Hemisphere for a northward IMF, figures from [2]. Both events shown in figure (a) and (b) are based on 2 hours of observations derived from Iridium magnetometer data. Times are in MLT	34
	a	34
	b	34
1.21	Two examples of Birkeland current distribution in the Northern Hemisphere for a southward IMF, figures from [2]. Both events shown in figure (a) and (b) are based on 2 hours of observations derived from Iridium magnetometer data. Times are in MLT.	34
	a	34
	b	34
2.1	Power spectrum of received radar signals	37
2.2	Radar beam with radar gates	38
2.3	Different radar pulse codes of ESR	39
2.4	Illustration of the gate intervals, Δx , as a function of beam length, x .	40
2.5	Interpolation a value into the empty cell based on the values of the surrounding cells. Cells close to the empty cell gets a higher weight.	42
2.6	Making a cell around the point in red by making four corners based on half the distances of a , b , c_1 , d_1 , c_2 , d_2 , which lengths are illustrated with dotted arrows.	43

2.7	Parameterisation to numerically calculate the Birkeland current density around point P[b][g], the shaded area.	44
3.1	Solar wind conditions on november 30th, 2014 as measured by ACE. The times on the x-axis are in universal time (UT). The values on the y-axis are in [nT]. The top panel shows component IMF B_y while the bottom panel shows component IMF B_z . The two shaded areas indicate the two cases, case 1 from 07:10 - 08:15 UT, case 2 from 09:10 to 10:15 UT.	46
3.2	The shaded area is the time interval for the 1st case, data from November 30th, 2014 as measured by ACE, time in UT.	48
3.3	Case 1: Illustration of the ESR placement in magnetic coordinates, MLT \approx 10:00. The direction of the radar swipe is an approximation for case 1.	49
3.4	Case 2: Illustration of the ESR placement in magnetic coordinates, MLT \approx 12:00. The direction of the radar fan is an approximation for case 2.	50
3.5	The figures (a) and (b) are for illustration and guidance only, not for further study. (a) Guide to the ion velocity plots. (b) Guide to the current density plots.	51
a	The direction of the ion velocity is indicated by the colors blue or red. As this guide shows, blue color means the ions are moving toward the radar marked as a blue dot, while red means away from the radar	51
b	The direction of the Birkeland current density is indicated by the colors blue and red. Blue means that the current is going downward into the paper, while red means upward out of the paper	51
3.6	Case 1: 07:10 to 08:20. Ion Velocity in [m/s], 30.11.14	52
3.7	Case 1: 07:10 to 08:20. Current density in 10^{-6} [A/m ²], 30.11.14	53
3.8	The shaded area is the time interval for the 2nd case, data from November 30th, 2014 as measured by ACE, time in UT.	54
3.9	Case 2: 09:10 to 10:15. Ion velocities in [m/s], 30.11.14	55
3.10	Case 2: 09:10 to 10:15. Current density in 10^{-6} [A/m ²], 30.11.14	57
4.1	The configuration of an Reversed Flux Transfer Event as proposed by [64] with poleward direction towards the top of the page. Plasma convection is indicated by the large arrow.	62

List of Tables

1.1	Solar wind observed properties near Earth's orbit, table from [33]	18
2.1	Example of median filtering, kernel in light gray and the number to be filtered in dark gray.	42
3.1	November 30th, 2014: Signs of IMF B_y and B_z , based on data from (OMNIweb). Case 1, where B_y goes from positive to negative, between 07:10 and 09:15 is shaded. Case 2, where B_z goes from positive to negative, between 09:10 and 10:15 is shaded.	47

Contents

1	Theoretical Background	3
1.1	Basic plasma physics	3
1.1.1	Single particle motion	3
1.1.2	Fluid description and magnetohydrodynamics	5
1.1.3	Particle velocity	8
1.1.4	Birkeland current density	13
1.2	Geophysical plasma	16
1.2.1	The Solar Wind	16
1.2.2	The Magnetosphere	20
1.2.3	The Ionosphere	26
2	Methods - Instrumentation and Data Processing	35
2.1	Instrumentation	35
2.1.1	EISCAT	36
2.1.2	The Incoherent Scatter Radar Technique	36
2.1.3	The EISCAT Svalbard Radar - ESR	37
2.2	Data processing using Python programming	38
2.2.1	Reading Matlab files into Python and structuring the data	39
2.2.2	Interpolate ion velocity onto the regularised swipe	41
2.2.3	Median filtering of the ion velocity	41
2.2.4	Making a grid	42
2.2.5	Calculate the Birkeland current density and plotting	42
3	Results	45
3.1	Presentation of data	45
3.2	Data selection - identifying two cases	45
3.3	Regarding the positioning of the ESR in MLT	46
3.4	A quick guide for plots	47
3.5	Case 1: 07:10 - 08:15	47
3.6	Case 2: 09:10 - 10:15	50

Contents

4 Discussion	59
4.1 Regarding the height-integrated Pedersen conductivity Σ_P for both cases	59
4.2 Case 1: Northward B_z , change in B_y	60
4.3 Case 2: Negative B_y , change in B_z	63
5 Conclusion	65
6 Outlook	67
Appendix A	i

Introduction

Field-aligned Birkeland currents are a set of currents flowing along geomagnetic field lines. These currents are connecting Earth's magnetosphere to the high latitude ionosphere below. Birkeland currents are an important feature in transferring energy from the upper magnetosphere down into the ionosphere. Their magnitudes are closely linked to the conditions in the solar wind.

The thesis will start with basic principles in the field of plasma physics. The single particle motion will be followed by a fluid description of plasma. The fluid description will build a bridge from single particle motion to the theory of magnetohydrodynamics (MHD) and magnetohydrodynamics will be used to find an analytical expression of the *Birkeland current density*.

As the field-aligned Birkeland currents are a connection between Earth's magnetosphere and the high latitude ionosphere, there will be an introduction to the field of space physics. Starting with the solar wind, the topology of Earth's magnetosphere and ionosphere will be presented to give an explanation on how and where Birkeland current occur.

The two parts of the theory chapter will be supplemented by a method chapter. The method chapter will start by presenting the source of the raw data - the Svalbard EISCAT Radar (ESR) and the technique of incoherent scatter. The second part of the method chapter will tell how the raw data was produced by the ESR and how it will be processed in the thesis.

The results from the data processing are going to be presented as a study of two cases. These cases will be presented as a series of figures. The figures will show the processed values of the ion velocities obtained by the Svalbard EISCAT Radar (ESR) and numerically calculated the values of the Birkeland current densities based on the analytical expression found in the theory chapter. In addition, there will be a written presentation of significant features found in the plots.

These features are going to be discussed and analysed to look for any kinds of patterns or structures presented in the theory chapter.

Based on the acquisition of data sets from the ESR, there is an expectation to have results that won't contradict established theories on Birkeland current distributions. There could however be shed new light on this phenomena as the resolution of the ESR is higher than on previously used instruments to study

Birkeland currents.

Chapter 1

Theoretical Background

1.1 Basic plasma physics

A *plasma* is a gas of charged particles which consists of an equal amount of free positive and negative charges. The number of charges with different signs within a given volume of the plasma are roughly the same. Because of this, a plasma looks electrically neutral from outside the plasma as the randomly distributed electric charges and their electric fields cancel each other out. This means that the plasma behaves *quasineutral* in the stationary state, but will be sensitive to magnetic and electric forces and be able to carry electric currents. This chapter will explain single particle motion and then moving on to a fluid description of plasmas. The fluid description and magnetohydrodynamics will then be used to derive the equation for the *Birkeland current density* which later on is to be numerically calculated based on a method described in chapter 2, methods.

1.1.1 Single particle motion

The single particle motion of a charged particle may provide some physical insight for better understanding the dynamic processes in plasmas. In very low-density plasmas, only single particle motion needs to be considered since collective effects often are unimportant.

Charged particles are affected by electric and magnetic fields which determine their motions. There is a coupling between electric charge and electromagnetic fields, where charged particles at rest are the source of the electrostatic field \mathbf{E} which again gives rise to the *Coulomb force*

$$\mathbf{F}_c = q\mathbf{E} \tag{1.1}$$

Charged particles with velocity \mathbf{v} generate a magnetic field \mathbf{B} which is the origin of the *Lorentz force*

$$\mathbf{F}_L = q(\mathbf{v} \times \mathbf{B}) \quad (1.2)$$

The relation between charged particles and fields is described by *Maxwell's equations*

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} \quad (1.3)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (1.4)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (1.5)$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad (1.6)$$

The motion of charged particles in the presens of an electric and magnetic field is given by the momentum equation.

$$m \frac{d\mathbf{v}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) + \mathbf{F}_g + m \left. \frac{d\mathbf{v}}{dt} \right|_c \quad (1.7)$$

where \mathbf{F}_g are the non-electromagnetic forces and $m \frac{d\mathbf{v}}{dt}$ represent the momentum change due to collisions. These two last terms can often be ignored, leading to

$$m \frac{d\mathbf{v}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (1.8)$$

A couple of important cases of single particle motion is presented below.

Static uniform \mathbf{B} , $\mathbf{E} = 0$

Choosing $\mathbf{B} = B\hat{\mathbf{z}}$ the equation of motion in a static uniform magnetic fiels is given by

$$m \frac{d\mathbf{v}}{dt} = q(\mathbf{v} \times \mathbf{B}) \quad (1.9)$$

which leads to

$$m\dot{v}_x = qBv_y, \quad m\dot{v}_y = -qBv_x, \quad m\dot{v}_z = 0$$

and

$$\begin{aligned}\ddot{v}_x &= \frac{aB}{m} \dot{v}_y = - \left(\frac{qB}{m} \right)^2 v_x \\ \ddot{v}_y &= - \frac{aB}{m} \dot{v}_x = - \left(\frac{qB}{m} \right)^2 v_y\end{aligned}\tag{1.10}$$

Equation (1.10) has the solution called *gyrofrequency* or *cyclotron frequency*:

$$\omega_g = \frac{qB}{m}\tag{1.11}$$

Notice that the signs for electrons and positive ions are opposite, meaning that the particles gyrate in opposite directions. From this the *gyroradius* or *Larmor radius* is defined as

$$r_g = \frac{v_{\perp}}{\omega_g} = \frac{mv_{\perp}}{|q|B}\tag{1.12}$$

This describes the circular orbit of a charged particle when $\mathbf{E} = 0$.

Static uniform \mathbf{B} , finite constant \mathbf{E}

Equation (1.9) is written in non-relativistic form. It is invariant under galilean transformations and this means that the term $(\mathbf{E} + \mathbf{v} \times \mathbf{B})$ does not depend on the frame of reference. Without the collision term or for infinite conductivity and choosing the electric field to be perpendicular to the magnetic field means

$$\mathbf{E} + \mathbf{v} \times \mathbf{B} = 0\tag{1.13}$$

Taking the cross-product of this with \mathbf{B} yields

$$\mathbf{v}_{gc} = \frac{\mathbf{E} \times \mathbf{B}}{B^2}\tag{1.14}$$

This shows that the guiding center drifts in the $\mathbf{E} \times \mathbf{B}$ direction. This drift is independent of charge and mass, so all particles drift with this velocity.

These two motions are shown in figure 1.1.

1.1.2 Fluid description and magnetohydrodynamics

The single particle description of plasma has its limitations. Equation (1.9) contains the time derivative taken at the position of the particle and when looking at a plasma as a whole, this is not a convenient form to use, as tracking the

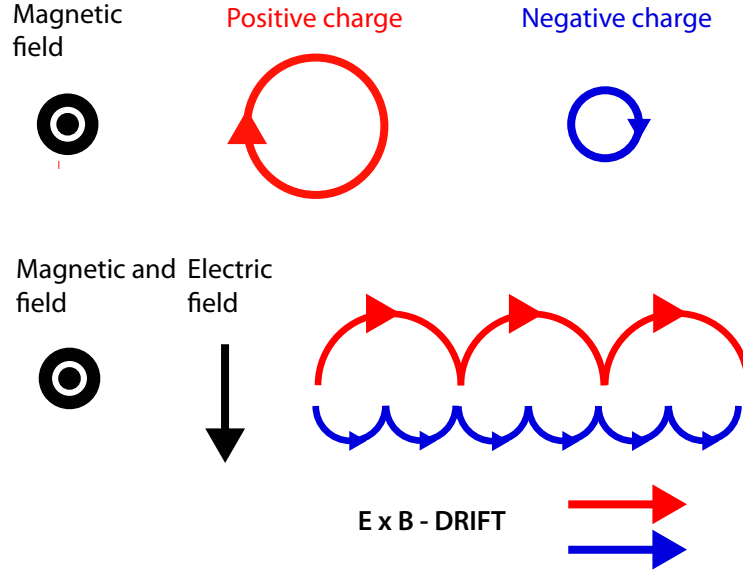


Figure 1.1: Top panel: Single particle motion with magnetic field out of the paper. Bottom panel: Single particle motion with magnetic field out of the paper and an electric field downwards.

movement of each particles quickly becomes unpractical when dealing with lots of particles. Instead, the *fluid approximation* can be used. However, since a plasma consists of at least two species, electrons and ions, one requires two sets of fluid equation to describe their dynamics. The general solutions to a set of two-fluid equations are not easy to solve. However, the two-fluid equations may be reduced to a one-fluid approach.

Equation of continuity

Electrons and the one ion species each need its equation of continuity

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot \rho_i \mathbf{v}_i = 0 \quad (1.15)$$

$$\frac{\partial \rho_e}{\partial t} + \nabla \cdot \rho_e \mathbf{v}_e = 0 \quad (1.16)$$

These two equations can be added to simplify the two fluid description to a one fluid equation with the use of $\rho_m = \rho_i + \rho_e = n_i m_i + n_e m_e$ and the defining the fluid velocity \mathbf{u}

$$\mathbf{u} = \frac{n_i m_i \mathbf{v}_i + n_e m_e \mathbf{v}_e}{n_i m_i + n_e m_e} \quad (1.17)$$

leads to

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot \rho_m \mathbf{u} = 0 \quad (1.18)$$

Equation of motion

The equation of motion looks at a fluid element fixed in space. And so, to change the frame of reference from a particle to a volume element containing many particles, there is a need to introduce a new derivative, called the *convective derivative*. This derivative is the sum of two terms.

$$\frac{d\mathbf{F}(x, t)}{dt} = \frac{\partial \mathbf{F}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} \frac{dx}{dt} \quad (1.19)$$

The reason for this is that the change in the fluid can be caused by two effects. The first term represents the change in the fluid \mathbf{F} at a fixed point in space. The second term represents the change in the fluid due to the observer drifting with the fluid into a region where \mathbf{F} is different. Written more compactly

$$\frac{d\mathbf{F}}{dt} = \frac{\partial \mathbf{F}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{F} = \frac{D\mathbf{F}}{Dt} \quad (1.20)$$

The velocity \mathbf{u} is the *average velocity* of all particles *inside the fluid element* as they move together and has been averaged over a distribution function. Ignoring collisions and other non-electromagnetic forces, and having established the convective derivative to take care of the changes of the fluid element, we find the one-fluid equation from the two-fluid system by adding the equation of motion for each species s , i.e electron and ion.

$$m_s n_s \frac{\partial \mathbf{v}_s}{\partial t} + (\mathbf{v}_s \cdot \nabla) \mathbf{v}_s = q_s n_s (\mathbf{E} + \mathbf{v}_s \times \mathbf{B}) \quad (1.21)$$

using $n_i = n_e = n$, $q_i = -q_e = |q|$ and $\mathbf{J} = n_i q_i \mathbf{v}_i + n_e q_e \mathbf{v}_e$ yields the equation of motion

$$\rho_m \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \mathbf{J} \times \mathbf{B} \quad (1.22)$$

Conservation of mass

Like any ordinary fluid, the MHD fluid is also a continuous medium, meaning that a small volume in the fluid element contains many particles. Conservation of mass means that in a volume V the total mass moving out of the volume,

through the surface S is

$$m = \int_S \rho \mathbf{V} \cdot \mathbf{n} dS \quad (1.23)$$

and the total mass lost in the volume is

$$m = - \int_V \frac{\partial \rho}{\partial t} dV \quad (1.24)$$

and when there are no sinks or sources within the volume, these two expressions are equal, meaning that

$$\int_V \left(\nabla \cdot \rho \mathbf{V} + \frac{\partial \rho}{\partial t} \right) dV = 0 \quad (1.25)$$

$$\Rightarrow \nabla \cdot \rho \mathbf{V} + \frac{\partial \rho}{\partial t} = 0 \quad (1.26)$$

The one fluid ideal *magnetohydrodynamic equations* are frequently used to study plasma phenomena, [41]. These equations are, in addition to the equation of continuity and equation of motion, in order:

$$p = p_0 \left(\frac{n}{n_0} \right)^{5/3} \quad (1.27)$$

$$\nabla \times \mathbf{E} = - \frac{\partial \mathbf{B}}{\partial t} \quad (1.28)$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} \quad (1.29)$$

$$\mathbf{J} = nq\mathbf{U} \quad (1.30)$$

an equation of state, Faraday's law, Ampere's law and Ohms law. The word *ideal* refers to neglecting the viscosity of the medium.

1.1.3 Particle velocity

The fluid description is used to find an expression for the velocity of ions and electrons which again is used to find the *current density*. Current density is defined as a difference in velocity between ions \mathbf{v}_i and electrons \mathbf{v}_e .

$$\mathbf{j} = ne(\mathbf{v}_i - \mathbf{v}_e) \quad (1.31)$$

By starting with the equation of motion, two fluid equation (1.21) the ion velocity and electron velocity is found. This mainly follows the way of [3]

$$n_s m_s \frac{d\mathbf{v}_s}{dt} = n_s q_s (\mathbf{E} + \mathbf{v}_s \times \mathbf{B}) - n_s m_s \nu_{in} (\mathbf{v}_s - \mathbf{u}_n) \quad (1.32)$$

Steady-state solution $\frac{d\mathbf{v}_s}{dt} = 0$ of equation (1.32)

$$n_s m_s \frac{d\mathbf{v}_s}{dt} = 0 = n_s q_s (\mathbf{E} + \mathbf{v}_s \times \mathbf{B}) - n_s m_s \nu_{sn} (\mathbf{v}_s - \mathbf{u}_n)$$

divide by n_s , find \mathbf{v}_s

$$\begin{aligned} m_s \nu_{sn} (\mathbf{v}_s - \mathbf{u}_n) &= q_s (\mathbf{E} + \mathbf{v}_s \times \mathbf{B}) \\ \mathbf{v}_s &= \mathbf{u}_n + \frac{q_s}{m_s \nu_{sn}} (\mathbf{E} + \mathbf{v}_s \times \mathbf{B}) \end{aligned} \quad (1.33)$$

Solving for ions $s = i$, $q_s = e$ and electrons $s = e$, $q_s = -e$

$$\mathbf{v}_i = \mathbf{u}_n + \frac{e}{m_i \nu_{in}} (\mathbf{E} + \mathbf{v}_i \times \mathbf{B}) \quad (1.34)$$

$$\mathbf{v}_e = \mathbf{u}_n - \frac{e}{m_e \nu_{en}} (\mathbf{E} + \mathbf{v}_e \times \mathbf{B}) \quad (1.35)$$

$$\mathbf{v}_s - \mathbf{u}_n = \frac{q_s}{m_s \nu_{in}} (\mathbf{E} + \mathbf{v}_s \times \mathbf{B}) \quad (1.36)$$

Add and subtract $\mathbf{u}_n \times \mathbf{B}$ to the right side of equation (1.36)

$$\mathbf{v}_s - \mathbf{u}_n = \frac{q_s}{m_s \nu_{in}} (\mathbf{E} + \mathbf{u}_n \times \mathbf{B} + \mathbf{v}_s \times \mathbf{B} - \mathbf{u}_n \times \mathbf{B}) \quad (1.37)$$

The cross product is distributive over addition and anticommutative:

$$\begin{aligned} (\mathbf{a} \times \mathbf{b}) + (\mathbf{a} \times \mathbf{c}) &= \mathbf{a} \times (\mathbf{b} + \mathbf{c}) \\ - [(\mathbf{b} \times \mathbf{a}) + (\mathbf{c} \times \mathbf{a})] &= - [\mathbf{a} \times (\mathbf{b} + \mathbf{c})] = (\mathbf{b} + \mathbf{c}) \times \mathbf{a} \end{aligned}$$

With $\mathbf{B} = \mathbf{a}$, $\mathbf{v}_s = \mathbf{b}$, $\mathbf{u}_n = \mathbf{c}$, equation (1.37) reads

$$\mathbf{v}'_s = \mathbf{u}_n + \frac{q_s}{m_s \nu_{in}} (\mathbf{E}' + \mathbf{v}'_s \times \mathbf{B}) \quad (1.38)$$

Introducing variables for easier typing $k_s = \frac{Bq_s}{m_s\nu_{sn}} \rightarrow k_i = \frac{Be}{m_i\nu_{in}}$, $k_e = -\frac{Be}{m_s\nu_{en}}$ ignoring the neutral wind so $\mathbf{u}_n = 0$, and multiply by B

$$\begin{aligned}\mathbf{v}_s &= \frac{B \cdot q_s}{B \cdot m_s \nu_{sn}} (\mathbf{E} + \mathbf{v}_s \times \mathbf{B}) \\ \mathbf{v}_s &= \frac{k_s}{B} \mathbf{E} + \frac{k_s}{B} (\mathbf{v}_s \times \mathbf{B})\end{aligned}\quad (1.39)$$

Equation (1.39) involves the cross product with \mathbf{v}_s , need to find an expression for it by crossing it with itself and solve for $\mathbf{v}_s \times \mathbf{B}$

$$\mathbf{v}_s \times \mathbf{B} = \frac{k_s}{B} \mathbf{E} \times \mathbf{B} + \frac{k_s}{B} (\mathbf{v}_s \times \mathbf{B}) \times \mathbf{B} \quad (1.40)$$

$$\mathbf{v}_s \times \mathbf{B} = \frac{k_s}{B} \mathbf{E} \times \mathbf{B} + \frac{k_s}{B} \left[\underbrace{(\mathbf{v}_s \cdot \mathbf{B})}_{\text{scalar}} \mathbf{B} - B^2 \mathbf{v}_s \right] \quad (1.41)$$

Finding $\mathbf{v}_s \cdot \mathbf{B}$ by dotting equation (1.39) with \mathbf{B}

$$\begin{aligned}\mathbf{v}_s \cdot \mathbf{B} &= \frac{k_s}{B} \mathbf{E} \cdot \mathbf{B} + \frac{k_s}{B} (\mathbf{v}_s \times \mathbf{B}) \cdot \mathbf{B} \\ \mathbf{v}_s \cdot \mathbf{B} &= \frac{k_s}{B} \mathbf{E} \cdot \mathbf{B} + \frac{k_s}{B} \mathbf{v}_s \cdot \underbrace{(\mathbf{B} \times \mathbf{B})}_0 \\ \mathbf{v}_s \cdot \mathbf{B} &= \frac{k_s}{B} \mathbf{E} \cdot \mathbf{B}\end{aligned}$$

Insert this into equation (1.41)

$$\begin{aligned}\mathbf{v}_s \times \mathbf{B} &= \frac{k_s}{B} \mathbf{E} \times \mathbf{B} + \frac{k_s}{B} \left[\left(\frac{k_s}{B} \mathbf{E} \cdot \mathbf{B} \right) \mathbf{B} - B^2 \mathbf{v}_s \right] \\ \mathbf{v}_s \times \mathbf{B} &= \frac{k_s}{B} \mathbf{E} \times \mathbf{B} + \left(\frac{k_s}{B} \right)^2 (\mathbf{E} \cdot \mathbf{B}) \mathbf{B} - k_s B \mathbf{v}_s\end{aligned}$$

Insert this back into (1.39)

$$\begin{aligned}
\mathbf{v}_s &= \frac{k_s}{B} \mathbf{E} + \frac{k_s}{B} \left[\frac{k_s}{B} \mathbf{E} \times \mathbf{B} + \left(\frac{k_s}{B} \right)^2 (\mathbf{E} \cdot \mathbf{B}) \mathbf{B} - k_s B \mathbf{v}_s \right] \\
\mathbf{v}_s &= \frac{k_s}{B} \mathbf{E} + \left(\frac{k_s}{B} \right)^2 \mathbf{E} \times \mathbf{B} + \left(\frac{k_s}{B} \right)^3 (\mathbf{E} \cdot \mathbf{B}) \mathbf{B} - k_s^2 \mathbf{v}_s \\
\mathbf{v}_s &= \frac{1}{1 + k_s^2} \left[\frac{k_s}{B} \mathbf{E} + \left(\frac{k_s}{B} \right)^2 \mathbf{E} \times \mathbf{B} + \left(\frac{k_s}{B} \right)^3 (\mathbf{E} \cdot \mathbf{B}) \mathbf{B} \right] \quad (1.42)
\end{aligned}$$

Inserting for ions and electrons

$$\mathbf{v}_i = \frac{1}{1 + k_i^2} \left[\frac{k_i}{B} \mathbf{E} + \left(\frac{k_i}{B} \right)^2 \mathbf{E} \times \mathbf{B} + \left(\frac{k_i}{B} \right)^3 (\mathbf{E} \cdot \mathbf{B}) \mathbf{B} \right] \quad (1.43)$$

$$\mathbf{v}_e = \frac{1}{1 + k_e^2} \left[-\frac{k_e}{B} \mathbf{E} + \left(\frac{k_e}{B} \right)^2 \mathbf{E} \times \mathbf{B} - \left(\frac{k_e}{B} \right)^3 (\mathbf{E} \cdot \mathbf{B}) \mathbf{B} \right] \quad (1.44)$$

The velocity can be written as a sum $\mathbf{v}_s = \mathbf{v}_{s\parallel} + \mathbf{v}_{s\perp}$ where the components are either parallel or perpendicular to the magnetic field \mathbf{B} . Finding their expressions.

$$\begin{aligned}
\frac{d\mathbf{v}_{s\parallel}}{dt} &= n_s q_s (\mathbf{E}_{\parallel} + \underbrace{\mathbf{v}_{s\parallel} \times \mathbf{B}}_0) - n_s m_s \nu_{sn} \mathbf{v}_{s\parallel} = 0 \\
\mathbf{v}_{s\parallel} &= \frac{B \cdot q_s \mathbf{E}_{\parallel}}{B \cdot m_s \nu_{sn}} = k_s \frac{\mathbf{E}_{\parallel}}{B} \quad (1.45)
\end{aligned}$$

Steady-state, perpendicular component

$$\begin{aligned}
\frac{d\mathbf{v}_{s\perp}}{dt} &= n_s q_s (\mathbf{E}_{\perp} + \mathbf{v}_{s\perp} \times \mathbf{B}) - n_s m_s \nu_{sn} \mathbf{v}_{s\perp} = 0 \\
\mathbf{v}_{s\perp} &= \frac{B \cdot q_s}{B \cdot m_s \nu_{sn}} \mathbf{E}_{\perp} + \frac{B \cdot q_s}{B \cdot m_s \nu_{sn}} \mathbf{v}_{s\perp} \times \mathbf{B} \\
\mathbf{v}_{s\perp} &= \frac{k_s}{B} \mathbf{E}_{\perp} + \frac{k_s}{B} \mathbf{v}_{s\perp} \times \mathbf{B} \quad (1.46)
\end{aligned}$$

Must find $\mathbf{v}_{s\perp} \times \mathbf{B}$, use equation (1.46) and cross it with \mathbf{B}

$$\begin{aligned}
\mathbf{v}_{s\perp} \times \mathbf{B} &= k_s \frac{\mathbf{E}_\perp}{B} \times \mathbf{B} + k_s (\mathbf{v}_{s\perp} \times \mathbf{B}) \times \mathbf{B} \\
\mathbf{v}_{s\perp} \times \mathbf{B} &= \frac{k_s}{B} \mathbf{E}_\perp \times \mathbf{B} + \frac{k_s}{B} \left[\underbrace{(\mathbf{v}_{s\perp} \cdot \mathbf{B}) \mathbf{B}}_0 - B^2 \mathbf{v}_{s\perp} \right] \\
\mathbf{v}_{s\perp} \times \mathbf{B} &= \frac{k_s}{B} \mathbf{E}_\perp \times \mathbf{B} - k_s B \mathbf{v}_{s\perp}
\end{aligned} \tag{1.47}$$

Inserting equation (1.47) into (1.46) and solving for $\mathbf{v}_{s\perp}$

$$\begin{aligned}
\mathbf{v}_{s\perp} &= \frac{k_s}{B} \mathbf{E}_\perp + \frac{k_s}{B} \left(\frac{k_s}{B} \mathbf{E}_\perp \times \mathbf{B} - k_s B \mathbf{v}_{s\perp} \right) \\
\mathbf{v}_{s\perp} &= \frac{k_s}{B} \mathbf{E}_\perp + \left(\frac{k_s}{B} \right)^2 \mathbf{E}_\perp \times \mathbf{B} - k_s^2 \mathbf{v}_{s\perp} \\
\mathbf{v}_{s\perp} &= \frac{1}{1 + k_s^2} \left[\frac{k_s}{B} \mathbf{E}_\perp + \left(\frac{k_s}{B} \right)^2 \mathbf{E}_\perp \times \mathbf{B} \right]
\end{aligned} \tag{1.48}$$

Finding

$$\begin{aligned}
\mathbf{v}_s &= \mathbf{v}_{s\parallel} + \mathbf{v}_{s\perp} \\
\mathbf{v}_s &= k_s \frac{\mathbf{E}_\parallel}{B} + \frac{1}{1 + k_s^2} \left[\frac{k_s}{B} \mathbf{E}_\perp + \left(\frac{k_s}{B} \right)^2 \mathbf{E}_\perp \times \mathbf{B} \right]
\end{aligned}$$

Giving the following expressions for ions and electrons

$$\mathbf{v}_i = k_i \frac{\mathbf{E}_\parallel}{B} + \frac{1}{1 + k_i^2} \left[\frac{k_i}{B} \mathbf{E}_\perp + \left(\frac{k_i}{B} \right)^2 \mathbf{E}_\perp \times \mathbf{B} \right] \tag{1.49}$$

$$\mathbf{v}_e = -k_e \frac{\mathbf{E}_\parallel}{B} + \frac{1}{1 + k_e^2} \left[-\frac{k_e}{B} \mathbf{E}_\perp + \left(\frac{k_e}{B} \right)^2 \mathbf{E}_\perp \times \mathbf{B} \right] \tag{1.50}$$

Defining

$$\mathbf{v}'_s = \mathbf{v}_s - \mathbf{u}_n = \frac{1}{1 + k_s^2} \left[\frac{k_s}{B} \mathbf{E}' + \left(\frac{k_s}{B} \right)^2 \mathbf{E}' \times \mathbf{B} + \left(\frac{k_s}{B} \right)^3 (\mathbf{E}' \cdot \mathbf{B}) \mathbf{B} \right]$$

1.1.4 Birkeland current density

Having assumed charge neutrality $n_e = n_i$ and using the expressions for \mathbf{v}_i equation (1.49) and \mathbf{v}_e equation (1.50), inserted into (1.31)

$$\begin{aligned}\mathbf{v}_i - \mathbf{v}_e &= \left(k_i \frac{\mathbf{E}_{\parallel}}{B} + \frac{1}{1+k_i^2} \left[\frac{k_i}{B} \mathbf{E}_{\perp} + \left(\frac{k_i}{B} \right)^2 \mathbf{E}_{\perp} \times \mathbf{B} \right] \right) \\ &\quad - \left(-k_e \frac{\mathbf{E}_{\parallel}}{B} + \frac{1}{1+k_e^2} \left[-\frac{k_e}{B} \mathbf{E}_{\perp} + \left(\frac{k_e}{B} \right)^2 \mathbf{E}_{\perp} \times \mathbf{B} \right] \right) \\ \mathbf{v}_i - \mathbf{v}_e &= \left(\frac{k_i}{1+k_i^2} - \left[-\frac{k_e}{1+k_e^2} \right] \right) \frac{\mathbf{E}_{\perp}}{B} + \left(\frac{k_i^2}{1+k_i^2} - \frac{k_e^2}{1+k_e^2} \right) \frac{\mathbf{E}_{\perp} \times \mathbf{B}}{B} + (k_i + k_e) \frac{\mathbf{E}_{\parallel} \times \mathbf{B}}{B}\end{aligned}$$

Defining three *conductivities* which will be more explained more in the section regarding the ionosphere: σ_P called the *Pedersen* conductivity, σ_H called the *Hall* conductivity and σ_{\parallel} is the *Birkeland* conductivity.

$$\begin{aligned}\sigma_P &= \frac{n_e e}{B} \left(\frac{k_i}{1+k_i^2} + \frac{k_e}{1+k_e^2} \right) \\ \sigma_H &= \frac{n_e e}{B} \left(\frac{k_i^2}{1+k_i^2} - \frac{k_e^2}{1+k_e^2} \right) \\ \sigma_{\parallel} &= \frac{n_e e}{B} (k_i + k_e)\end{aligned}$$

The current density can now be written as

$$\mathbf{j} = \sigma_P \mathbf{E}_{\perp} - \sigma_H \frac{\mathbf{E}_{\perp} \times \mathbf{B}}{B} + \sigma_{\parallel} \mathbf{E}_{\parallel}$$

Which, since $\mathbf{E}_{\parallel} \times \mathbf{B} = 0$, equals

$$\mathbf{j} = \sigma_P \mathbf{E}_{\perp} - \sigma_H \frac{\mathbf{E} \times \mathbf{B}}{B} + \sigma_{\parallel} \mathbf{E}_{\parallel} \quad (1.51)$$

To find the components of the current density \mathbf{j} we choose a coordinate system where $\mathbf{B} = -B_z \hat{\mathbf{k}}$ and $\mathbf{E} = E_x \hat{\mathbf{i}} + E_y \hat{\mathbf{j}} + E_z \hat{\mathbf{k}}$, $\mathbf{E}_{\perp} = E_x \hat{\mathbf{i}} + E_y \hat{\mathbf{j}}$ and $\mathbf{E}_{\parallel} = E_z \hat{\mathbf{k}}$ and use this on equation (1.51)

$$\begin{aligned}\mathbf{j} &= \sigma_P (E_x \hat{\mathbf{i}} + E_y \hat{\mathbf{j}}) + \sigma_H \frac{E_x \hat{\mathbf{i}} + E_y \hat{\mathbf{j}} + E_z \hat{\mathbf{k}} \times -B \hat{\mathbf{k}}}{B} + \sigma_{\parallel} E_z \hat{\mathbf{k}} \\ \mathbf{j} &= \sigma_P E_x \hat{\mathbf{i}} + \sigma_P E_y \hat{\mathbf{j}} + \sigma_H (E_x \hat{\mathbf{j}} - E_y \hat{\mathbf{i}}) + \sigma_{\parallel} E_z \hat{\mathbf{k}} \\ \mathbf{j} &= (\sigma_P E_x - \sigma_H E_y) \hat{\mathbf{i}} + (\sigma_P E_y + \sigma_H E_x) \hat{\mathbf{j}} + \sigma_{\parallel} E_z \hat{\mathbf{k}}\end{aligned}$$

Which can be written on matrix form and is the generalized Ohm's law:

$$\mathbf{j} = \begin{pmatrix} \sigma_P & -\sigma_H & 0 \\ \sigma_H & \sigma_P & 0 \\ 0 & 0 & \sigma_{\parallel} \end{pmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} \quad (1.52)$$

Looking at the components of the current density $\mathbf{j} = \mathbf{j}_{\perp} + \mathbf{j}_{\parallel}$.

$$\begin{aligned} \mathbf{j}_{\perp} &= j_x \hat{\mathbf{i}} + j_y \hat{\mathbf{j}} \\ \mathbf{j}_{\perp} &= \sigma_P \mathbf{E}_{\perp} - \sigma_H \frac{\mathbf{E}_{\perp} \times \mathbf{B}}{B} \end{aligned}$$

Assume that \mathbf{E}_{\perp} is independent of height dh and use the height integrated conductivity Σ_L , (L for 'letter')

$$\Sigma_L = \int_{h_1}^{h_2} \sigma_L(h) dh$$

where $L = [P, H]$.

$$\mathbf{J}_{\perp} = \Sigma_P \mathbf{E}_{\perp} - \Sigma_H \frac{\mathbf{E}_{\perp} \times \mathbf{B}}{B}$$

Height-integrated current density \mathbf{J}

$$\mathbf{J} = \mathbf{J}_{\perp} + \mathbf{J}_{\parallel} \quad (1.53)$$

Rewrite equation (1.53) with

$$\hat{\mathbf{b}} = \frac{\mathbf{B}}{B}, \quad \mathbf{E}_{\parallel} = \mathbf{E} \cdot \frac{\mathbf{B}}{B} \cdot \left(\frac{\mathbf{B}}{B} \right) = (\mathbf{E} \cdot \hat{\mathbf{b}}) \cdot \hat{\mathbf{b}} = E_z \hat{\mathbf{b}}$$

so

$$\begin{aligned} \mathbf{J} &= \Sigma_P \mathbf{E}_{\perp} - \Sigma_H \frac{\mathbf{E}_{\perp} \times \mathbf{B}}{B} + \Sigma_{\parallel} \mathbf{E}_{\parallel} \\ \mathbf{J} &= \Sigma_P \mathbf{E}_{\perp} - \Sigma_H \mathbf{E}_{\perp} \times \hat{\mathbf{b}} + \Sigma_{\parallel} (\mathbf{E} \cdot \hat{\mathbf{b}}) \cdot \hat{\mathbf{b}} \end{aligned} \quad (1.54)$$

The divergence of this current density must be zero, [3]

$$\begin{aligned}\nabla \cdot \mathbf{J} &= 0 \\ \nabla_{\perp} \cdot \mathbf{J}_{\perp} + \nabla_{\parallel} \cdot \mathbf{J}_{\parallel} &= 0 \\ \nabla_{\perp} \cdot \mathbf{J}_{\perp} &= -\nabla_{\parallel} \cdot \mathbf{J}_{\parallel} \\ \nabla_{\perp} \cdot \mathbf{J}_{\perp} &= -\frac{\partial J_z}{\partial z}\end{aligned}$$

use that $j_z = \frac{\partial J_z}{\partial z}$

$$-\nabla_{\perp} \cdot \mathbf{J}_{\perp} = j_z$$

Insert for \mathbf{J}_{\perp} and calculate

$$\begin{aligned}j_z &= -\nabla_{\perp} \cdot \left[\Sigma_P \mathbf{E}_{\perp} - \Sigma_H (\mathbf{E}_{\perp} \times \hat{\mathbf{b}}) \right] \\ j_z &= -\nabla_{\perp} \cdot \left[\Sigma_P \mathbf{E}_{\perp} + \Sigma_H (\hat{\mathbf{b}} \times \mathbf{E}_{\perp}) \right]\end{aligned}$$

The plasma drift velocity was in section 1.1 found to be $\mathbf{v} = \frac{\mathbf{E} \times \mathbf{B}}{B^2}$, coming from steady-state, meaning $\mathbf{E} = -\mathbf{v} \times \mathbf{B}$. Find $\hat{\mathbf{b}} \times \mathbf{E}_{\perp} = -B\mathbf{v}$

$$\begin{aligned}j_z &= -\nabla_{\perp} \cdot [\Sigma_P \mathbf{E}_{\perp} + \Sigma_H (-B\mathbf{v})] \\ j_z &= -\Sigma_P (\nabla_{\perp} \cdot \mathbf{E}_{\perp}) - (\nabla_{\perp} \cdot \Sigma_P) \mathbf{E}_{\perp} + \Sigma_H (\nabla_{\perp} \cdot B\mathbf{v}) + (\nabla_{\perp} \cdot \Sigma_P) B\mathbf{v} \\ j_z &= -\Sigma_P (\nabla_{\perp} \cdot \mathbf{E}_{\perp}) - \nabla_{\perp} \Sigma_P \cdot \mathbf{E}_{\perp} + \Sigma_H B (\nabla_{\perp} \cdot \mathbf{v}) + (\nabla_{\perp} \Sigma_P) \cdot B\mathbf{v}\end{aligned}$$

Looking at

$$\begin{aligned}\nabla_{\perp} \cdot \mathbf{v} &= \nabla_{\perp} \cdot \left(\frac{\mathbf{E} \times \mathbf{B}}{B^2} \right) = \left(\hat{\mathbf{i}} \frac{\partial}{\partial x} + \hat{\mathbf{j}} \frac{\partial}{\partial y} \right) \cdot \left(-E_x \hat{\mathbf{j}} + E_y \hat{\mathbf{i}} \right) \frac{1}{B^2} \\ \nabla_{\perp} \cdot \mathbf{v} &= 0\end{aligned}\tag{1.55}$$

so

$$j_z = -\Sigma_P (\nabla_{\perp} \cdot \mathbf{E}_{\perp}) - \nabla_{\perp} \Sigma_P \cdot \mathbf{E}_{\perp} + B (\nabla_{\perp} \Sigma_P) \cdot \mathbf{v}\tag{1.56}$$

Using $\mathbf{E} = -\mathbf{v} \times \mathbf{B}$, using formula for the nabla product

$$\begin{aligned}(\nabla_{\perp} \cdot \mathbf{E}_{\perp}) &= \nabla_{\perp} \cdot (-\mathbf{v} \times \mathbf{B}) = \nabla_{\perp} \cdot (\mathbf{B} \times \mathbf{v}) \\ (\nabla_{\perp} \cdot \mathbf{E}_{\perp}) &= \mathbf{v} \cdot (\nabla_{\perp} \times \mathbf{B}) - \mathbf{B} \cdot (\nabla_{\perp} \times \mathbf{v})\end{aligned}$$

The first term on the right hand side is Ampere's law, ignoring the last term in Ampere's law

$$(\nabla_{\perp} \cdot \mathbf{E}_{\perp}) = \mathbf{v} \cdot \mu_0 \mathbf{j} - \mathbf{B} \cdot (\nabla_{\perp} \times \mathbf{v})$$

The first term can be dropped leaving

$$\nabla_{\perp} \cdot \mathbf{E}_{\perp} = -\mathbf{B} \cdot (\nabla_{\perp} \times \mathbf{v})$$

Insert this into equation (1.56)

$$\begin{aligned} j_z &= -\Sigma_P [-\mathbf{B} \cdot (\nabla_{\perp} \times \mathbf{v})] - \nabla_{\perp} \Sigma_P \cdot \mathbf{E}_{\perp} + B(\nabla_{\perp} \Sigma_P) \cdot \mathbf{v} \\ j_z &= \Sigma_P \mathbf{B} \cdot (\nabla_{\perp} \times \mathbf{v}) - \mathbf{E}_{\perp} \cdot \nabla_{\perp} \Sigma_P + B(\mathbf{v} \cdot \nabla_{\perp} \Sigma_P) \end{aligned}$$

rewriting it to

$$J_{\parallel} = \Sigma_P \mathbf{B} \cdot (\nabla \times \mathbf{v}) - \mathbf{E} \cdot \nabla \Sigma_P + B(\mathbf{v} \cdot \nabla \Sigma_P) \quad (1.57)$$

The first term is called the magnetospheric component, the last two terms the ionospheric component, [46]. Assuming homogeneous conductivity the last two terms can be ignored, leading to

$$\frac{J}{\Sigma_P} = \mathbf{B} \cdot (\nabla \times \mathbf{v}) \quad (1.58)$$

1.2 Geophysical plasma

1.2.1 The Solar Wind

Origin

The Sun is an ordinary star of average size and temperature. It is about 4.6 billion years old and is estimated to have a life expectancy of another 5 billion years, typical for a star of size and temperature like the Sun. The Sun is composed mostly of hydrogen, 90 %, and helium, 10 %, and a small amount of other constituents, among others carbon, nitrogen and oxygen, often called CNO. The matter in the Sun is ionized due to nuclear reactions and electrostatics within the Sun which create a very high temperature. This temperature exceeds the binding energies of the atoms and rips the electrons from the protons.

The atmosphere of the Sun is divided into three layers. The photosphere is a layer about 500 km thick and is what we categorize as the surface of the sun. It has a temperature about 6000 °K. Above the photosphere is the chromosphere. It extends roughly 2500 km in height. The final layer of the Sun's atmosphere

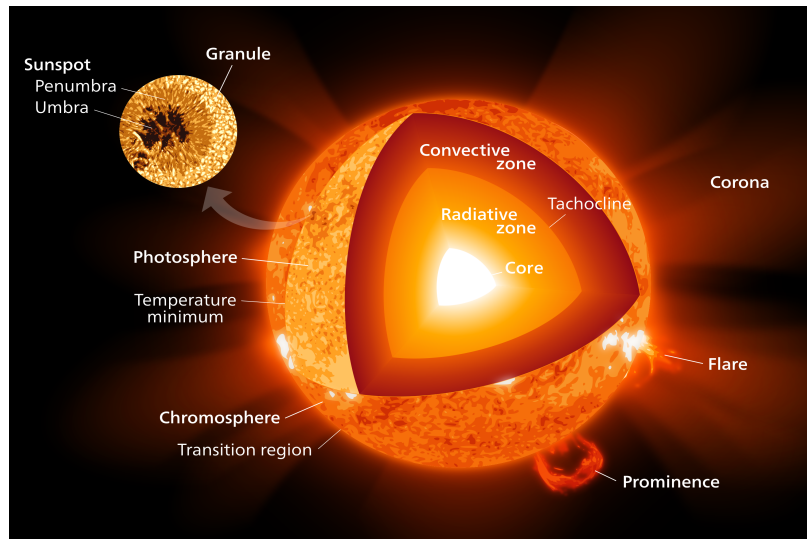


Figure 1.2: The Sun's interior and exterior [55]

is called the corona where the temperature reaches several million °K. As the corona expands into space the temperature decreases. This expansion into space is called the *solar wind* and is happening because the atmosphere of the Sun is not in a static equilibrium.

The interplanetary magnetic field (IMF)

As the Sun is a magnetic star with magnetic fields and the corona is a good electrical conductor, the solar wind carries out a portion of the solar coronal magnetic field. This magnetic field is embedded into the solar wind and is called the *interplanetary magnetic field* (IMF). Because of the IMF, the solar wind may transfer energy, mass and momentum to the magnetosphere and further down into Earth's ionosphere. The activity of the Sun follows a magnetic cycle of 22 years where the poles of the Sun switch places and an 11 year sunspot cycle where the sunspots move closer and closer to the solar equator, figure (??). These changes on the Sun causes effects in space and on Earth. During an active period there is also an increase in high-energy particles ejected from the Sun and phenomena such as coronal mass ejections (CMEs) and solar flares happens more frequently and violently. The solar wind is more disturbed during these years; the velocity is higher, the temperature is higher and the magnetic field is more disturbed.

Parameters

As the solar wind is an expansion of the corona, it consists mostly of the same particles as the corona; protons (ions) and electrons with a small amount of α particles (He^{++}), though the fraction of α particles may increase during disturbed

conditions. The other parameters of the solar wind may fluctuate considerable at Earth's orbit[44]. The velocity of the solar wind has a mean value around

Solar wind parameters near the orbit of the Earth (1 AU)	
Proton density	6.6 cm^{-3}
Electron density	7.1 cm^{-3}
He^{2+} density	0.5 cm^{-3}
Velocity (nearly radial)	450 km/s
Proton temperature	$1.2 \times 10^5 \text{ K}$
Electron Temperature	$1.4 \times 10^5 \text{ K}$
Magnetic field (induction)	$7 \times 10^{-9} \text{ T}$

Table 1.1: Solar wind observed properties near Earth's orbit, table from [33]

450 km/s, but varies from anything between 170 km/s to more than 2000 km/s. Solar wind with a velocity lower than 400 km/s is called *slow* solar wind, while high speed streams with a velocity greater than 600 km/s are called *fast* solar wind. Slow and fast solar winds are associated with different origins on the Sun. The slow solar wind originates from *coronal loops* where material is moving along the magnetic field lines of the Sun. The fast solar wind originates from *coronal holes*. These are open field lines in the Sun's magnetic field which link directly to Earth, resulting in faster plasma flowing along these open field lines.

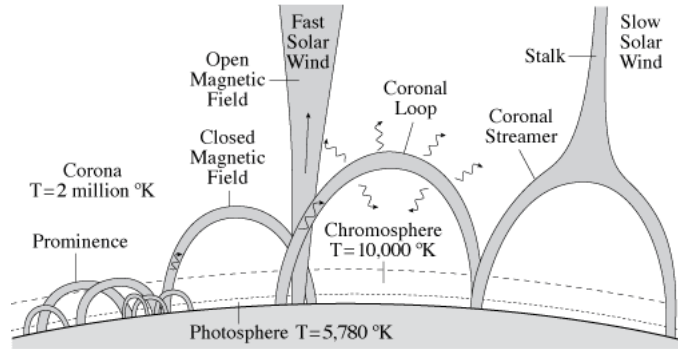


Figure 1.3: The origin of the fast and slow solar wind, [56]

The temperature in the solar wind is, like the density and the velocity, also fluctuating. In fact it may vary up to a factor of more than 100. This has to do with the absence of thermal equilibrium, a random velocity distribution and a temperature coupled to the solar wind bulk velocity. The typical slow velocity, which has higher density than the fast solar wind, has a temperature around 30 000 °K whereas the fast solar wind measures more around 200 000 °K.

The strength of the interplanetary magnetic field is decomposed into three components along a *geosentric solar magnetic* (GSM) coordinate system. In this

system (x, y, z) , x points to the north pole, y to the east and z is the vertical component along Earth's dipole axis. An average strength of the IMF at Earth's position is about 3.5 nT, but it may vary considerably.

The frozen-in concept

To understand the plasma flow in the solar wind and the embedded magnetic field, let's look at equation (1.13) and assume an ideal conducting fluid so that its conductivity σ is infinite. From Maxwell's equation

$$-\frac{\partial \mathbf{B}}{\partial t} = \nabla \times \mathbf{E}$$

using that $\mathbf{E} + \mathbf{U} \times \mathbf{B} = 0$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{U} \times \mathbf{B}) \quad (1.59)$$

Equation (1.59) is identical in form to the vorticity equation for an ordinary homogeneous inviscid fluid [41]. This equation leads to the Kelvin-Helmholtz theorems which states:

1. Flux of vorticity through any closed contour moving with the fluid is constant.
2. Fluid elements that lie on a vortex line continue to lie on the same vortex line.

It seems reasonable to extend the above theorems to magnetohydrodynamics and the flux of a magnetic field, so let's check if that's true.

If C is the contour in the fluid at time t and the fluid moves with velocity $\mathbf{U}(\mathbf{r}, t)$ (figure here) and the displacement of the contour after a time $t + \Delta t$ is indicated by C' , then let S and S' be the surfaces enclosed by C and C' . Let $d\mathbf{l}$ be a line element on C and this element after some time Δt moves a distance $\mathbf{U}\Delta t$, it sweeps out an area of $d\mathbf{l} \times (\mathbf{U}\Delta t)$.

The magnetic flux through the surface S is defined as

$$\Phi = \int_S \mathbf{B} \cdot d\mathbf{S} \quad (1.60)$$

Consider the total magnetic flux that enters and leaves these surfaces, and due to the divergence theorem, the total magnetic flux at any time crossing a closed

surface must vanish, the total flux at time $t + \Delta t$ is:

$$-\int_C \mathbf{B}(t + \Delta t) \cdot \mathbf{n} dS + \int_{C'} \mathbf{B}(t + \Delta t) \cdot \mathbf{n} dS' + \oint \mathbf{B}(t + \Delta t) \cdot d\mathbf{l} \times \mathbf{U} dt = 0 \quad (1.61)$$

The change in the magnetic flux Φ with time is

$$\frac{d\Phi}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Phi_{C'}(t + \Delta t) - \Phi_C(t)}{\Delta t} \quad (1.62)$$

inserting for Φ , equation (1.61)

$$\frac{d\Phi}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\int [\mathbf{B}(\mathbf{t} + \Delta \mathbf{t}) - \mathbf{B}(\mathbf{t})] \cdot \mathbf{n} dS}{\Delta t} - \oint \mathbf{B}(t + \Delta t) \cdot d\mathbf{l} \times \mathbf{U} \quad (1.63)$$

the first term is the time derivative and on the second term use vector identity then Stoke's theorem, so that

$$\frac{d\Phi}{dt} = - \int \nabla \times (\mathbf{E} + \mathbf{U} \times \mathbf{B}) \cdot \mathbf{n} dS \quad (1.64)$$

This integrand vanishes if

$$(\mathbf{E} + \mathbf{U} \times \mathbf{B}) = -\nabla \Psi \quad (1.65)$$

where Ψ is some scalar and flow is irrotational, $\nabla \times \nabla \Psi = 0$. This shows that the magnetic flux through any arbitrary contour moving with the fluid is constant in time - the field is *frozen-in*. This is valid for any magnetic field that moves with velocity $\mathbf{U}(\mathbf{r}, t)$, like the solar wind. The concept of frozen-in flux says that the plasma is following the magnetic field lines that moves.

1.2.2 The Magnetosphere

Earth's magnetic field is generated deep into the core of the planet. The inner core is a solid while the outer core is liquid. The magnetic field is generated by the moving iron liquid in the outer core, driving electric currents giving raise to the magnetic field.

The magnetic field of the Earth is, within a few Earth radii, similar to a dipole field. Further away from Earth, the geomagnetic field acts as an obstacle in space for the solar wind flow and interactions occure between the solar wind with its IMF and the Earth's magnetic field which carves out a separate plasma domain inside Earth's magnetic field. This volume is called the *magnetosphere*. Within the magnetosphere, the geomagnetic field is the dominant factor for the

motion of the electrically charged particles. The particles which are trapped in Earth's magnetosphere originates both from the solar wind and from Earth's upper atmosphere below.

The shape and size of the magnetosphere is determined by the balance between the kinetic pressure of the solar wind plasma and the pressure of Earth's magnetic field:

$$\rho_{SW} u_{SW}^2 = \frac{B_{MS}^2}{2\mu_0} \quad (1.66)$$

where the subscript SW is the solar wind and MS is the magnetosphere.

The kinetic pressure of the solar wind distorts the outer parts of the Earth's magnetic field. At the sunward side of Earth the field is compressed. The distance to the boundary separating the region dominated by the solar wind plasma from the region dominated by magnetospheric plasma is called the *stand-off* distance and is at approximately 10 Earth radii. At the nightside the magnetic field is stretched out into a long magnetotail up to 200 Earth radii, extending to well beyond the Moon.

Structure of the magnetosphere

The structure of the magnetosphere is usually described in the GSM system with noon facing the Sun, dawn is left and dusk is right. The most important regions of the magnetosphere are shown in figure 1.4.

The magnetosphere is made up of regions with different plasma characteristics. Closest to Earth is a region called the *plasmasphere*. This region consists of relatively cold and dense plasma which has escaped from the *ionosphere* below. The plasmasphere corotates with the Earth. This leads to an electric induced field [30]. The ring currents are embedded into the plasmasphere. The upper boundary of the plasmasphere is called the plasmopause. The distance to the plasmopause is variable and depends on the solar wind disturbance, but usually at 4 - 6 Earth radii. The outer boundary of the magnetosphere is called the magnetopause. This is the boundary between the solar wind regime and the Earth dominated plasma. The deflection of the solar wind plasma in the magnetopause results in a finite thickness of the magnetopause and is also responsible for the magnetopause current inside it, called Chapman-Ferraro current. (p.294 Kallenrode). 2-3 Earth radii in front of the magnetopause is the bow shock. Here the supersonic solar wind is slowed down to subsonic speed. The solar wind passes through the bow shock but does not penetrate the magnetopause. The region between the bow shock and the magnetopause is called the magnetosheath. Here the solar wind is slowed down more and deflected.

The polar cusps are two singularities in the dayside magnetosphere. They are the separation regions of geomagnetic field lines that extend to different parts

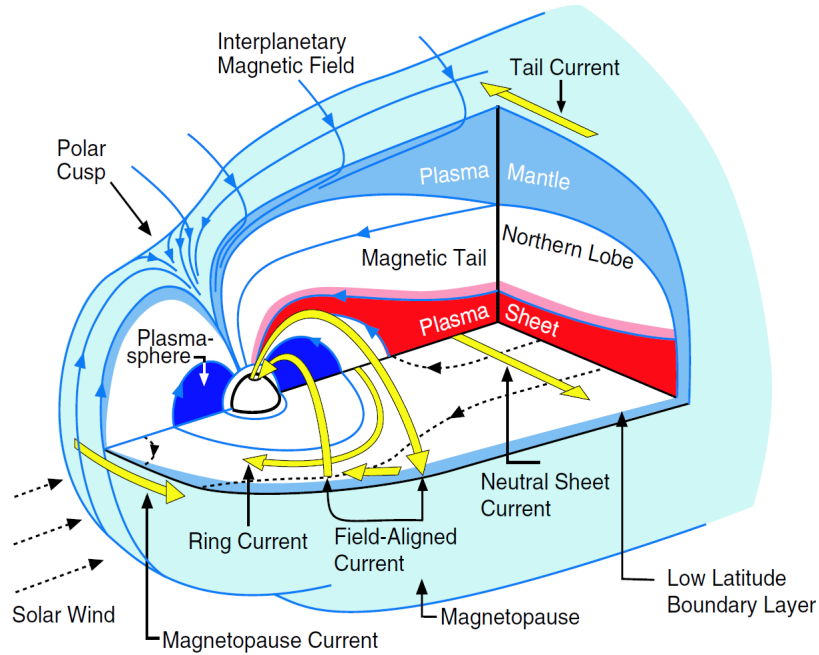


Figure 1.4: Schematic figure of the magnetosphere showing the main plasma regions and current systems, [Russell and Luhmann, 1997].

of the magnetosphere: the closed field lines on the dayside magnetosphere with both anchor points on Earth, and the open field lines on the nightside which have one anchor point on Earth and the other on the Sun. The cusps are filled with plasma from outside the magnetosphere, meaning that plasma from the solar wind can penetrate deep into Earth's atmosphere in these two regions.

Magnetic local time (MLT)

The concept of magnetic local time is used when describing magnetospheric and ionospheric phenomena. 12 MLT is defined as towards the Sun and 24 MLT is midnight. For the ESR the magnetic local time is

$$\text{MLT} = \text{UT} + 2.45 \text{ hours} \quad (1.67)$$

where UT is the universal time measured at the 0° meridian, [6].

Upper polar atmosphere

The magnetosphere in the polar regions has a special kind of topology with three distinguished regions: the *polar cap*, the *polar oval* and *subpolar latitudes*, figure 1.5.

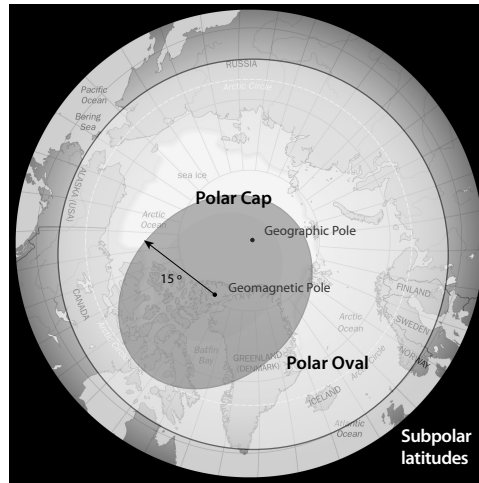


Figure 1.5: Drawing of the magnetospheric polar region, [57]

The polar cap is a circular area with center displaced a few degrees toward the nightside from the *magnetic pole*. The polar cap has a typical radius of 15° . Surrounding the polar cap is the polar oval or auroral oval with a width stretching a few degrees equatorward. The polar oval is more narrow toward the noon sector. The region to the south of the polar oval is called subpolar latitudes. The reason for dividing the polar magnetosphere into these regions is that they are associated with different particle populations in the magnetosphere. The polar cap is connected the magnetotail lobe population via magnetic field lines. The dayside of the polar oval is connected to the polar cusp and the magnetospheric boundary layer, while the nightside of the polar oval is connected to the tail plasma sheet. The subpolar latitudes are connected to the inner magnetosphere.

Since these regions are not symmetric around Earth's geographical pole they rotate together with Earth. For the northern hemisphere this means that different different areas may lay in different zones of the upper polar atmosphere during one rotation.

The width of the polar oval and the radius of the polar cap is not static, but highly influenced by the magnetospheric activity. During active times the cap and oval both expand and during quiet conditions they contract.

Energy transfer from the solar wind to the magnetosphere

The plasma in the solar wind cannot penetrate the geomagnetic field. In order for the solar wind to transfer energy to the magnetosphere there must be an opening in the field lines. This happens during through a phenomena called *magnetic reconnection*. The solar wind plasma, having passed the bow shock and moving into the magnetosheath, will merge with the geomagnetic field lines and the field lines reconnect and moves the plasma in another direction, figure 1.6. For an IMF

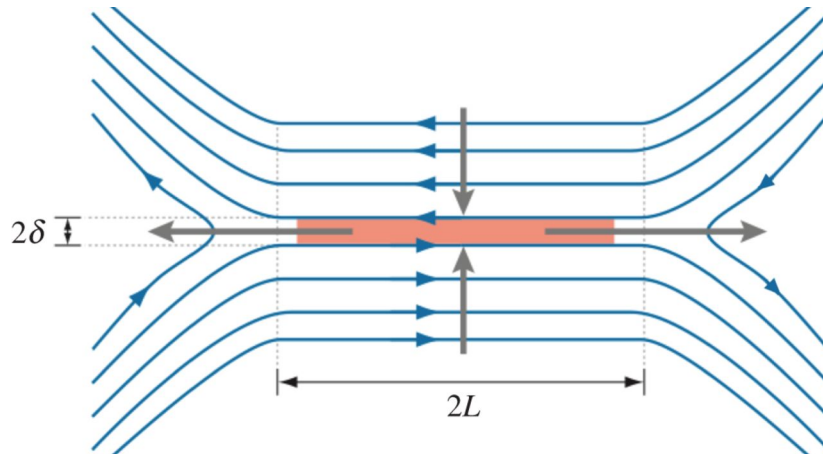


Figure 1.6: "Magnetic field geometry for the Sweet-Parker [reconnection] model. Oppositely directed field lines are brought together and reconnect in a diffusion layer (orange). The plasma is heated by Ohmic dissipation at the diffusion region and accelerated by the pressure gradient and the tension force. The field line diffuses over the half-width of the diffusion layer, δ , which is much smaller than the system size, $2L$." Figure from [31]

with component $B_z < 0$, the reconnection happens at the magnetopause dayside MLT, while an IMF with component $B_z > 0$ gives rise to a *lobe* reconnection, more to the nightside.

After the field lines of the IMF and the geomagnetic field have merged, the merged field lines are dragged over the polar cap by the IMF being carried with the solar wind. These field lines are stretched into a long magnetotail where field lines of opposite direction, from the southern hemisphere and northern hemisphere, generate a tail current system. These field lines will reconnect in the tail and accelerate the plasma back towards Earth. The closed field lines from this nightside reconnection are convected back to the dayside magnetosphere, not across the polar cap, but at lower polar latitudes, numbers (8) and (9), figure 1.7. This description of the plasma convection is called the *Dungey cycle*.

Solar wind-Magnetosphere-Ionosphere coupling

The connection between the solar wind and the magnetosphere can be seen as a dynamo with the ionosphere below acting as a load. This is called the solar wind dynamo. At high latitudes, the terrestrial magnetic field is almost perpendicular to the ionosphere and the outer magnetosphere and the high latitude ionosphere are coupled via electric fields and field-aligned currents, called *Birkeland currents*. Birkeland currents are typical in the polar ionosphere and are divided into two

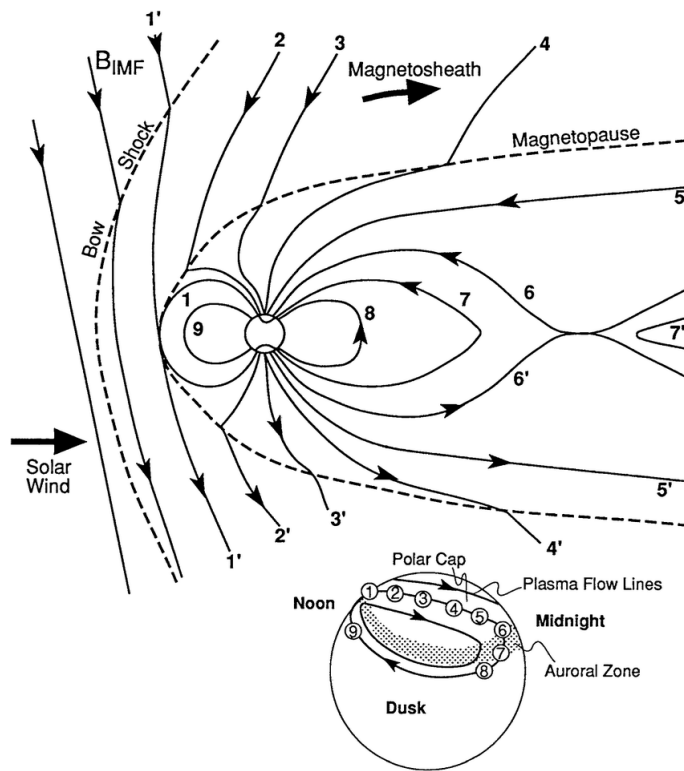


Figure 1.7: Dungey cycle, [33]

regions. Region 1 currents flowing *out* of the ionosphere on the evening side (dusk). Region 2 currents flowing at lower latitudes in an opposite pattern where they flow out of the ionosphere at dawn side and into the ionosphere at dusk side, figure 1.8. The plasma convection in the magnetosphere, which is driven by the solar wind and IMF, is following the $\mathbf{E} \times \mathbf{B}$ direction. This is mirrored down to the ionosphere.

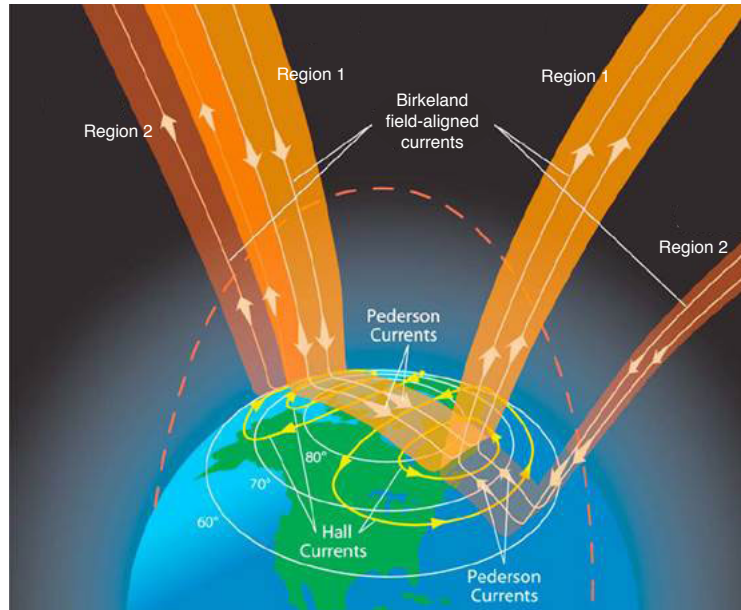


Figure 1.8: Region 1 and region 2 currents, [51].

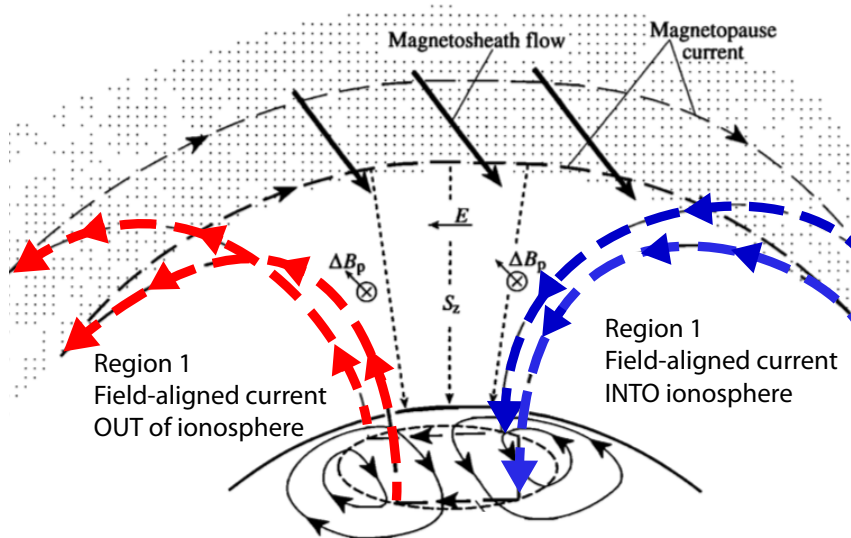


Figure 1.9: Sketch of the polar cap current circuit. Birkeland field-aligned currents in color. Red going *up*, out of the ionosphere. Blue going *down*, into the ionosphere [7].

1.2.3 The Ionosphere

The ionosphere is a *ionized* part of Earth's upper atmosphere. It starts at a height at around 60-70 km above the surface of the Earth and stretches up to

approximately 1000 km where it gradually merges into the plasmasphere. The ionosphere is created by neutral gases absorption of electromagnetic radiation in the UV and EUV range. Although the degree of ionization is in order of 1 %, the

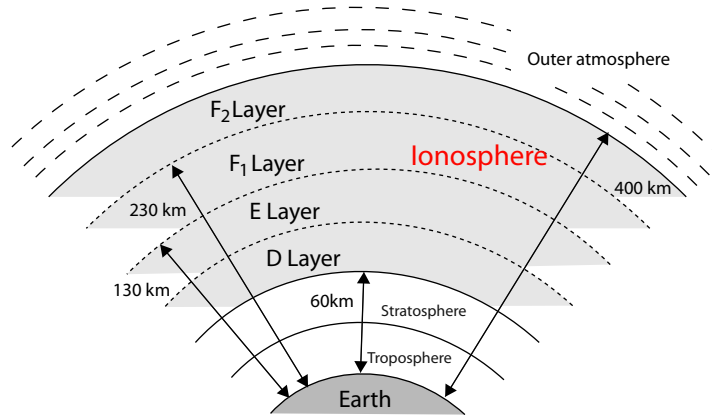


Figure 1.10: The different layers of the ionosphere marked in light gray with typical latitude in [km].

existence of charged particles enables the flow of electric currents. This affects the particle motion and it distinguishes the ionosphere from the magnetosphere, where particle motion is determined by the magnetic field only.

The ionosphere can be described as a variable shell of ionization surrounding the Earth. The dividing of the ionosphere into separate layers is a simplification as the distinction between the layers are not all that clear, except maybe for some events in the E-region, [3]. As seen in figure 1.10 the D region stretches from around 60-90 km, the E region from around 90-130 km and the F-region from around 110-130 km.

Composition

The composition of the ionosphere varies with height as the atmosphere consists of different kinds of molecules. The incoming radiation that is responsible for the ionization covers a broad wavelength band, meaning that different particle species are ionized at typical heights. This is shown in figure 1.11. The dynamics of the ionospheric layers are not only determined by the ionization due to radiation, but also by the losses due to recombination to neutral particles. These loss processes modify the daily variation of the densities.

The lowest part of the ionosphere is called the D region. This is extending from about 60 to 90 km above ground. During quiet conditions, when the solar wind is less disturbed, these regions are present only at daytime. Since it is closer to the neutral atmosphere, the D region has a larger number of atmospheric

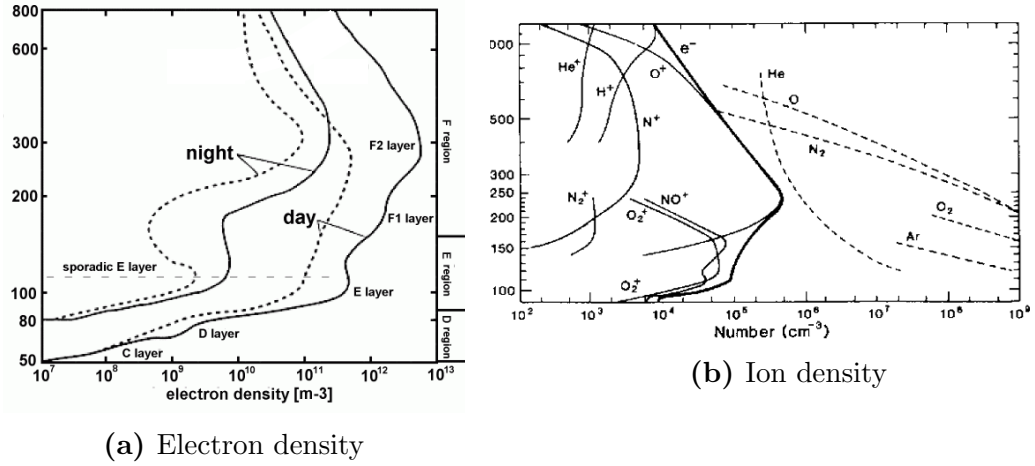


Figure 1.11: Height dependence of different molecules in the ionosphere, quiet solar conditions *somewhere*.

particles and is the most weakly ionized part of the ionosphere. Ions are mostly from O_2^+ and NO^+ .

The E region is extending from about 90 to 120 km above the Earth. The E region has a single maximum electron density at about 120 km, while the most abundant ions are O_2^+ and NO^+ as for the D region. The E region also shows a diurnal variation with a maximum ionization at local noon.

The F region is the top layer of the ionosphere, extending from about 130 to 500 km above the Earth. The region can be split into two separate layers during daytime, the F₁ layer and the F₂ layer, while at nighttime these form a single layer. The F₁ layer exists only during daylight while the F₂ layer is present all day and night but varies in altitude. Ions in the lower parts of the F layer are mainly NO^+ , while in the upper parts O_2^+ and H^+ dominates. The electron density starts to decrease and the upper parts of the ionosphere slowly merges with the magnetosphere.

Convection patterns

As described in the magnetosphere section of 'energy transfer from the solar wind to the magnetosphere', the plasma convection follows a route across the polar cap, figure 1.13. Since the solar wind and the IMF also has a electric field it will create a $\mathbf{E} \times \mathbf{B}$ drift in the ionosphere. This ionospheric convection is depending on the direction of the IMF, which strongly simplified looks like 1.12. The top panel shows a two cell pattern called *twin cells* typical for southward IMF ($B_z < 0$), at with values of B_y on top. The bottom panes shows a configuration for northward IMF ($B_z > 0$), where the two cell pattern has broken up. More complex patterns

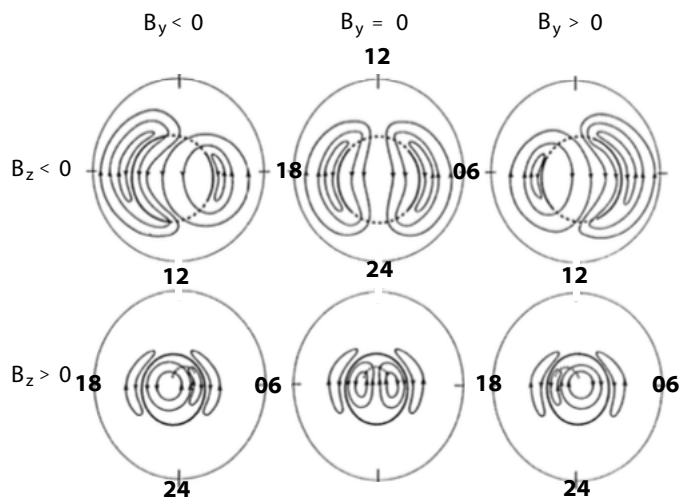


Figure 1.12: "Sketches of the northern hemisphere ionospheric flow ordered according to the direction of the IMF. The arrowed solid lines show the plasma streamlines, [...]", figure from [8].

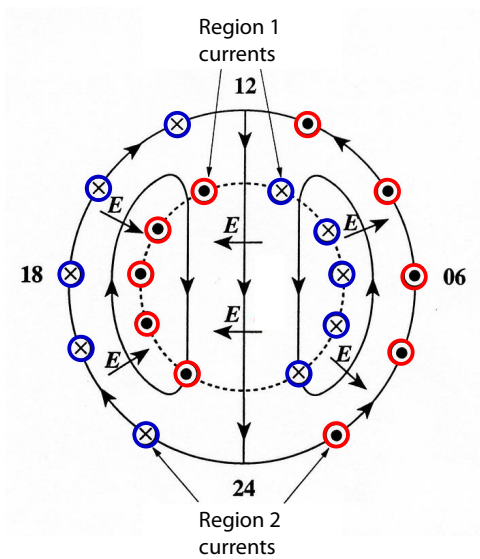


Figure 1.13: Schematic figure of the ionospheric Dungey cycle plasma flow for southward IMF. Solid lines are the plasma streamline, and the direction of the electric field is given by the short arrows, from [7].

are shown in figures 1.15 and 1.16 based on a number of measurements, 1.14.

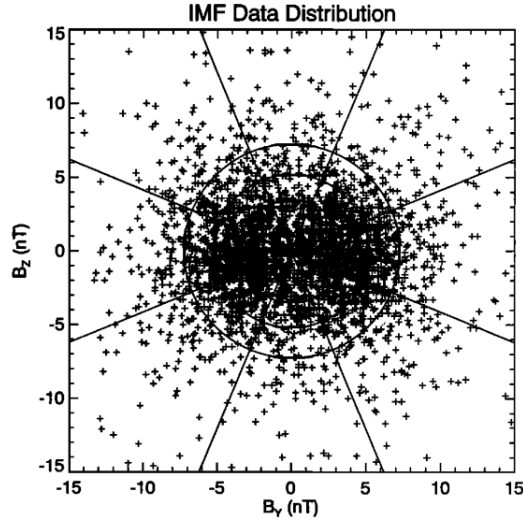


Figure 1.14: IMF data distribution showing the IMF B_y and B_z values to find the B_T , used in figures 1.15 and 1.16, as concentric circles, from [62].

Regarding collisions

The plasma flowing in the ionosphere is a collisional, partially ionized plasma, which roughly follows the Dungey cycle convection, figure 1.13. This flow drags the plasma and heats the neutral gas, connecting the E region of the ionosphere with the magnetosphere [40]. The neutral particles are responsible for the collisions and this affects the movement of the charged particles and how currents are produced [41]. These collisions change the velocities of the individual particles and transfer momentum among the colliding particles. The number of collisions between the charged particles and neutrals depends on the altitude. The ion-electron collisions are neglected since the number of neutrals are so much higher at this altitude. As seen in figure 1.17, collisions affect ions at higher altitude than electrons. When the motion of the ion is interrupted by frequent collisions, this slows down its drift motion in the $\mathbf{E} \times \mathbf{B}$ direction. In the E region, the motion of the ions is dominated by collisions with neutrals while electrons still are moving in an $\mathbf{E} \times \mathbf{B}$ drift. Since the plasma is collision-dominated the frozen-in approximation of the solar wind is no longer valid. The plasma will have a finite conductivity which leads us back to the three ionospheric conductivities defined in section 1.1.4. This again leads to a relative drift between the ions and the electrons and a current density arise from the difference in these velocities, mentioned in section 1.1.3, leading to the Birkeland current density. As seen in figures 1.19, 1.20 and 1.21 the distribution of the Birkeland currents follow a pattern.

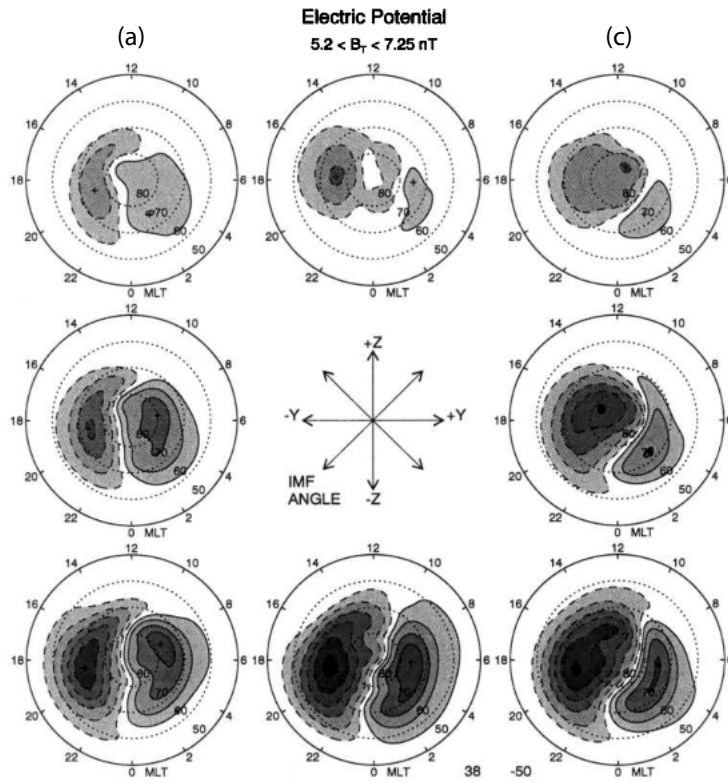


Figure 1.15: Electrical potentials with convection pattern represented by dotted lines, from [62].

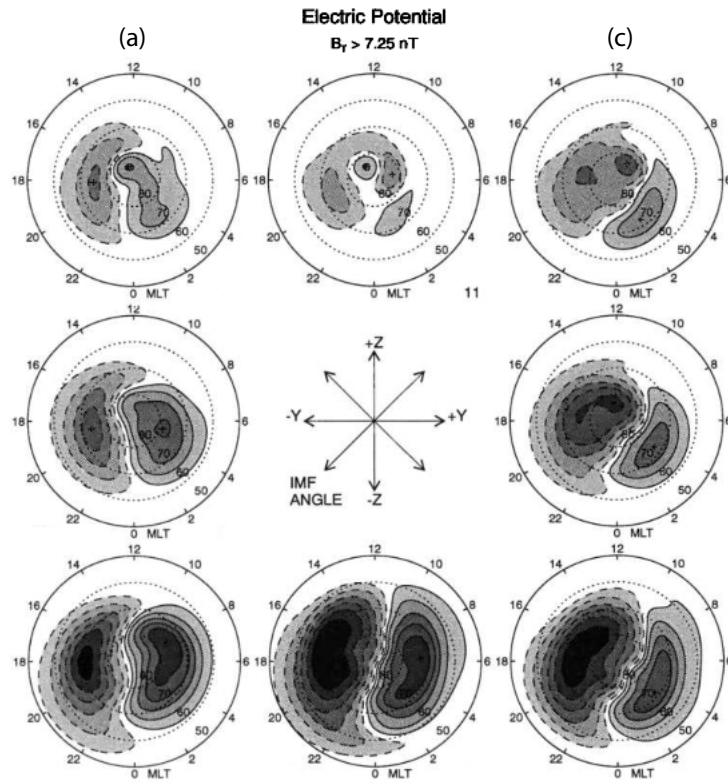


Figure 1.16: Electrical potentials with convection pattern represented by dotted lines, from [62]. The values of IMF B_Y and B_Z are larger than in figure 1.15.

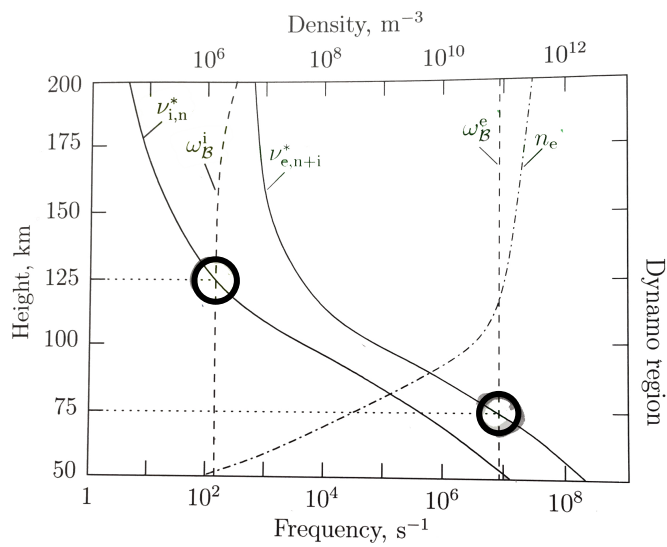


Figure 1.17: The ion- and electron gyrofrequency vs the collisional frequencies of both species, picture from [44].

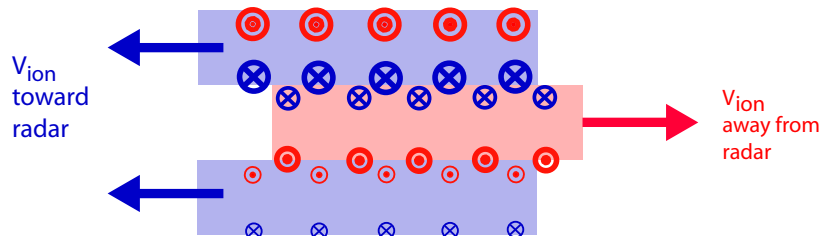


Figure 1.18: Schematic illustration showing the ion velocity and the associated field-aligned Birkeland current due to a velocity difference. Blue arrow indicates ion velocity is toward radar, red means away. Blue circles with cross means the FAC is going downward, red circle with dot means its going upward, adaptation from [63].

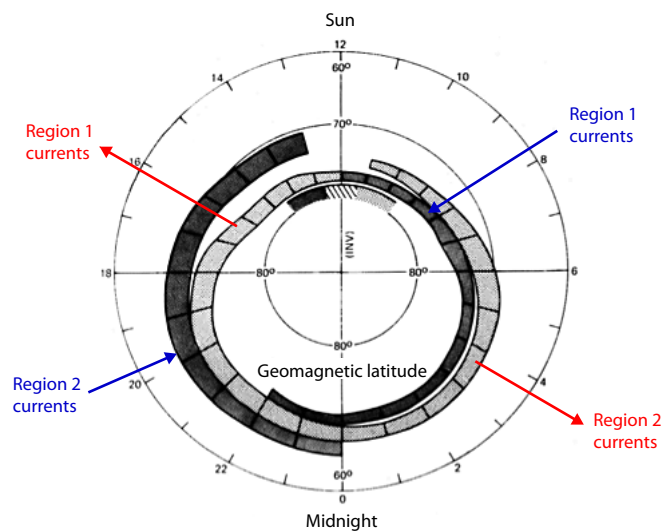


Figure 1.19: An average configuration of Birkeland currents in the polar ionosphere, times are in local time. Region 1 currents flow out of the ionosphere on dusk side and into the ionosphere on dawn side, while Region 2 currents flow in opposite direction, figure from [28].

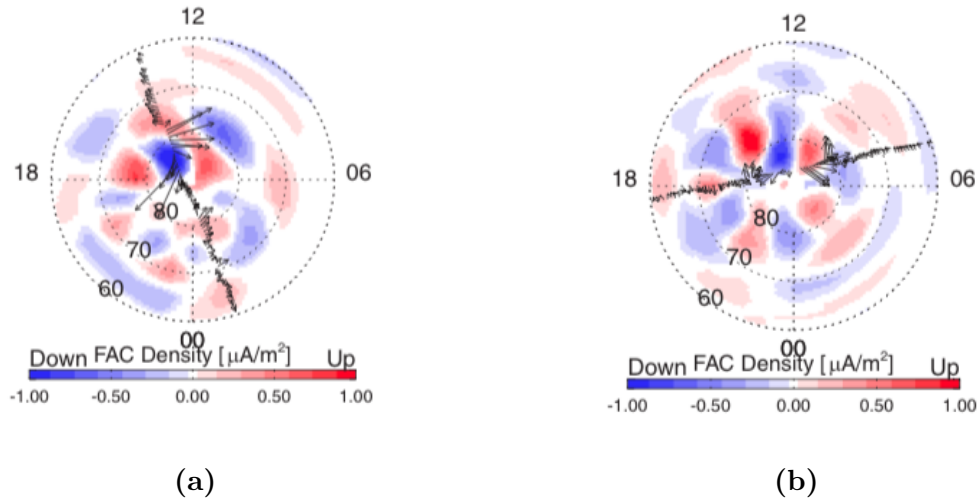


Figure 1.20: Two examples of Birkeland current distribution in the Northern Hemisphere for a **northward** IMF, figures from [2]. Both events shown in figure (a) and (b) are based on 2 hours of observations derived from Iridium magnetometer data. Times are in MLT

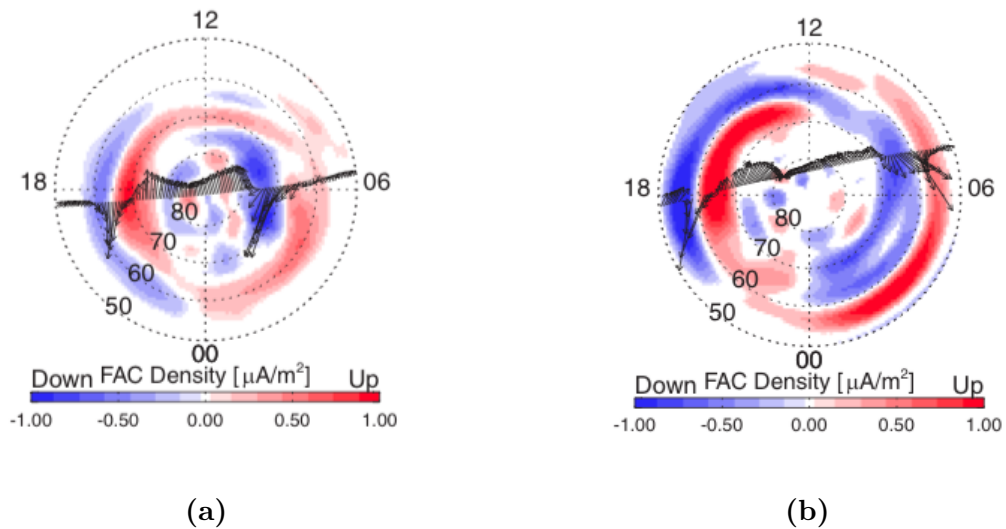


Figure 1.21: Two examples of Birkeland current distribution in the Northern Hemisphere for a **southward** IMF, figures from [2]. Both events shown in figure (a) and (b) are based on 2 hours of observations derived from Iridium magnetometer data. Times are in MLT.

Chapter 2

Methods - Instrumentation and Data Processing

2.1 Instrumentation

RAdio Detection And Ranging (RADAR) is an instrument for detecting and studying remote targets. A radar transmits a radio wave or pulse, called pulse code, in the direction of the target. When the emitted wave comes into contact with an object it is reflected or scattered. The radar signals reflected back towards the transmitter are detected by a receiver. The echo, called backscatter, is then analysed. There are many possible scattering mechanisms. Based on which ranges and properties one is to investigate, the radar system is optimised to fit one particular scattering process. This means that different pulse codes are designed for different experiments, different power output is used and different frequencies and so on.

The data sets from the ESR spawn over two days, 2014-11-30 which contains 2453 Matlab files and 2014-12-01 which contains 1837 Matlab files, each file representing one beam sent out by ESR. In each of these files, variables are listed in columns and need to be extracted.

The data of IMF the components B_y and B_z of the days of study can found from the ACE satellite, [48]. ACE is a satellite orbiting the Lagrangian point 1 between the Sun and Earth. There are several instruments onboard ACE. This thesis uses the data from ACE's magnetometer (MAG) which monitors the interplanetary magnetic field in the solar wind. This data can downloaded as a .txt file, [53]. In the data from the omniweb site the lag from ACE to the magnetopause is already calculated based on the plasma bulk speed measured by ACE.

2.1.1 EISCAT

'EISCAT is an international scientific association with member institutes in several countries. We conduct ionospheric and atmospheric measurements with radars' [21]. The radars are placed in Tromsø, Norway, in Longyearbyen, Svalbard, in Kiruna, Sweden and in Sodankylä, Finland. Out of ten incoherent scatter radars in the world, EISCAT operates three of them. This thesis makes use of the EISCAT incoherent scatter radar, ESR, located in Longyearbyen.

2.1.2 The Incoherent Scatter Radar Technique

When studying the ionosphere with a radar it is important that the radar pulse has a higher frequency than the local plasma frequency, or else the signal is reflected in the ionosphere. The local plasma frequency ω_p , which for an electron in the ionosphere is

$$\omega_{pe} = \sqrt{\frac{n_e \cdot e^2}{2\pi\epsilon_0 m_e}} \quad (2.1)$$

To a good approximation $\omega_{pe} = 9000 \cdot \sqrt{n_e}$ Hz when n_e is in units of cm^{-3} . The peak of electron density in the ionosphere, at the height ~ 250 km is $\sim 10^{12} \text{ m}^{-3}$ ($= 10^6 \text{ cm}^{-3}$) which gives $\omega_{pe} \leq 12$ MHz. An incoherent scatter radar transmits a radar pulse typically from 50 MHz up to 2 GHz. This is well above the electron plasma frequency, equation (2.1). Due to this, the radar can probe the ionosphere above the electron density peaks because the signal is refracted less by the ionization in the ionosphere.

The idea behind incoherent scatter radars is that the electric field in the transmitted radio pulse accelerates the electrons in the ionosphere. The electrons, being very light, respond more to the radar wave than the ions which are much heavier, thus the electrons do most of the reflecting and contribute most to the backscatter. This effect is called Thomson scattering and describes how electromagnetic waves are scattered off charged particles. As the electrons are accelerated due to the transmitted wave, they emit radiation at almost the same frequency as the transmitted pulse. The received signal is composed of millions of these reflections. They are reflected back to the antenna dish and based on this backscatter, a power spectrum that looks like figure (2.1) is obtained from the received signal. Through this power spectrum different plasma parameters are derived.

The received frequencies appears on both sides of the transmitted signal. This happens because particles emitting radiations are moving in all directions. The ones moving away from the radar have a lower frequency than the transmitted signal while the ones moving towards the radar have higher frequency than the transmitted signal, e.g doppler shift.

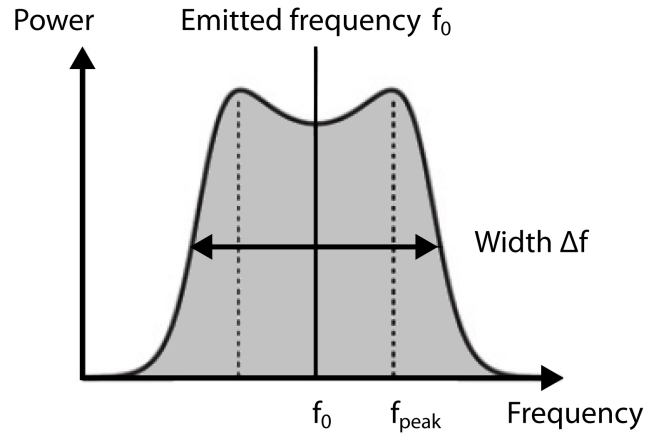


Figure 2.1: Power spectrum of received radar signals

The electron density contributes to the power of the received signal. Higher electron density results in stronger backscatter which is determined by the integral of the area under the graph, figure 2.1.

The two peaks or shoulders in the spectrum figure relates to the *electron temperature* on the form of T_e/T_i . The width from f_0 to the highest peak f_{peak} is related to the *mean ion velocity*. The *ion temperature* is derived from the spectral width Δf since the motion of the electrons follows the ion motion.

The scattering of the transmitted signals lack coherence due to the thermal fluctuations of the electron density, giving name to this specific radar technique which is called *incoherent scattering*.

When the radio pulse is sent out, the received signal depends on the range s from the radar to the echoing region. The range is determined by

$$\begin{aligned} 2s &= v \cdot t \\ s &= \frac{v \cdot t}{2} \end{aligned} \quad (2.2)$$

Each received signal represent a measurement at a different range. These ranges are called a *gates*, the closest gate being the signal first received, and the distance to it calculated by equation (2.2). The number of gates in a signal varies from pulse to pulse, each pulse called a beam.

2.1.3 The EISCAT Svalbard Radar - ESR

The Eiscat Svalbard Radar (ESR) is a part of the European Incoherent Scatter Radar system. This is one of the most powerful tools to study the conditions of

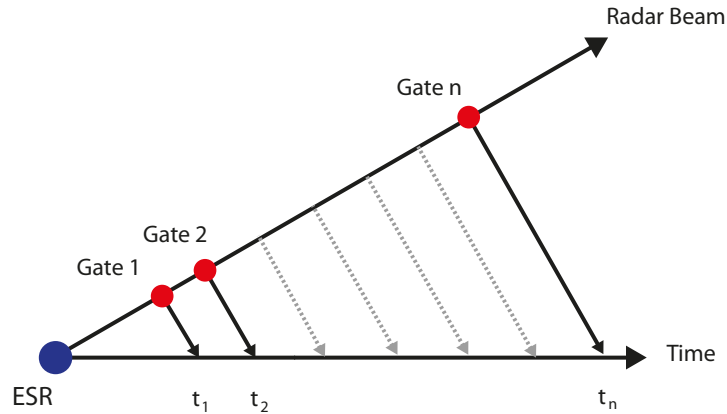


Figure 2.2: Radar beam with radar gates

the ionosphere from ground. The ESR is located in Longyearbyen, Spitsbergen, Svalbard, 78.22°N , 15.65°E . As mentioned in the section above, incoherent scatter radars like the ESR use frequencies well above the local plasma frequency, (2.1).

The ESR is a monostatic radar, meaning that the same antenna transmits and receives the signal. The ESR consists of a 32 m steerable parabolic dish antenna and a 42 m field aligned antenna, operating at a frequency of 500 MHz. The reflected signal is very weak so ESR has a powerful transmitter. The ESR can run different experiments to study the ionosphere. The data in this thesis are from the taro32 experiment. This experiment has a range cover from 46-812 km, with highest radar efficiency from around 200 km to around 800 km. The experiment has a sampling frequency of $25 \mu\text{s}$. It is designed to study the E and F region of the ionosphere with moderate spatial resolution of 3.8 - 7.5 km and time resolution of 6.4 s. For the data used in this thesis, the antenna was in a wind screen wiper mode which means that the antenna was moving azimuthally in a swipe back and fourth, sending out a radar pulse every $25 \mu\text{s}$. These pulses were then integrated to make up the power spectra and derive the different variables.

2.2 Data processing using Python programming

The data processing needed to calculate the Birkeland current density can be split into five main steps:

1. Read the Matlab files and identify the needed variables.
2. Identify each swipe of the radar to make a regularised beam and a regularised swipe.

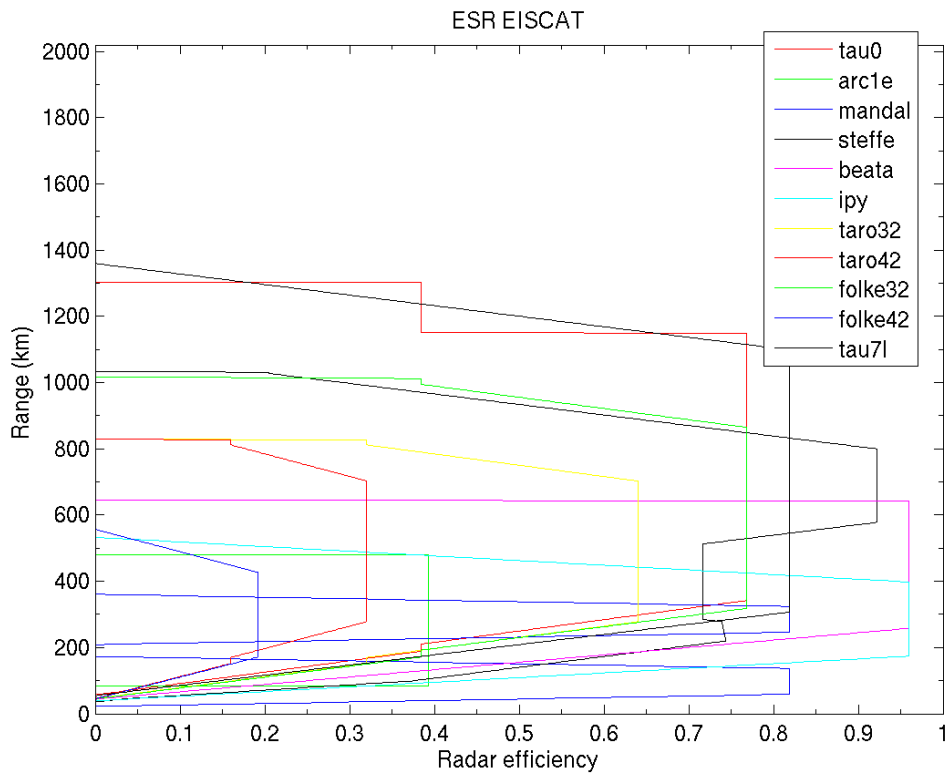


Figure 2.3: Different radar pulse codes of ESR

3. Interpolate the ion velocity onto the regularised swipe and do a median filtering of the ion velocity to remove 'salt and pepper' noise.
4. Calculate the Birkeland current density with the use of Stoke's theorem
5. Make a grid out of the regularised swipe and plot the Birkeland current density on the grid on top of a map of Svalbard.

2.2.1 Reading Matlab files into Python and structuring the data

Matlab files can be read into Python with the use of the Python package *Scipy*, [49]. This will store the Matlab variables in a Python dictionary. The variables can then be extract from the dictionary. The spesific variables needed for this thesis are: the azimuth angle, elevation angle, a nested array called parameters which contains ion velocity, electron density, electron temperature, ion temperature, the distance from gate to the radar and time objects which holds time steps and dates.

The data files from ESR have no set number of measurements, meaning that the number of data points related to each variable may change from file to file. In order to calculate the Birkeland current density, the data must be structured. Every beam within one swipe must be made to have the same length. This means:

1. Identify each swipe of the radar.
2. For each swipe, make a the regularised beam and use the regularised beam to make a regularised swipe.

Identifying a swipe

Each swipe of the radar can be identified by the change in the azimuth angle. If the azimuth angle in one point $[g]$ is smaller than the azimuth angles of the previous point $[g-1]$ and the next point $[g+1]$, or if the azimuth angle in one point $[p]$ is larger than the angles of the previous point $[g-1]$ and the next point $[g+1]$, the radar is ending a swipe.

To make the regularised beam and the regularised swipe

The regularised beam can be made the following way. The distance between gates, dx , in a beam grows with the distance to the radar, x , as illustrated in figure 2.4. By looking at every beam in the same data set a polynomial fitting of

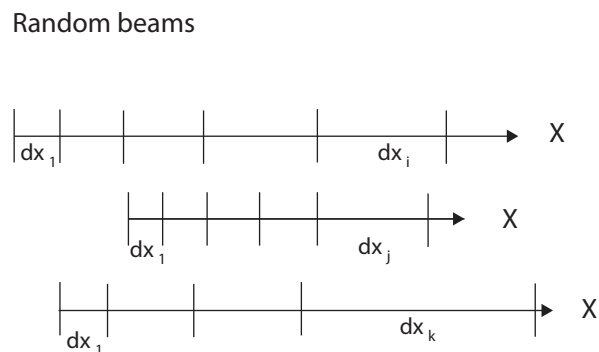


Figure 2.4: Illustration of the gate intervals, dx , as a function of beam length, x .

the gate distances, dx , as a function of radar distance, x , can be done. Keeping in mind that if the gate distances between two gates is very large compared to the rest of the dx 's in the beam, the fit will suffer. A choice is made to set the maximum distance of dx to 100 km and if dx is larger than this, dx is set to *nan*

to not include that interval. This may be in some cases where the radar doesn't receive a strong backscatter, so there is no measuring point where there normally might be one.

For each fitting, the coefficients are stored as well as the distance to the 1st gate and the last gate of that beam. Then the average is taken of all the coefficients for all beams in the data set to get a polynomial fit for the whole swipe. New gates are made, starting from the closest gate going to furthest gate. This is the regularized beam.

A regularized swipe is made by looping through the number of beams in the swipe and adding an identical regularized beam for each of them. The maximum value for the ion velocity is set to 2500 m/s, anything above this is set to *nan* as the measurement is most likely an error, velocities of that magnitude are not present in the ionosphere.

The ion velocity is then put into the array which holds the regularized swipe and placed on the gate which is closes to the actual distance stored in the actual beam.

2.2.2 Interpolate ion velocity onto the regularised swipe

The regularised swipe contains a number of empty gates without values. This is a consequence of making the regularised swipe, since the regularised swipe needs to have more gates than the total number of gates in a swipe to be able to fit all gates of one swipe. These empty gates need to be filled with values in order to be able to calculate the Birkeland current density. This is done by interpolating the velocity between neighbouring gates of the empty gate and put the interpolated value into the missing gate-value.

2.2.3 Median filtering of the ion velocity

The data from ESR is based on radar signals. This kind of data is known to have some noise called *salt and pepper noise*, and is best smoothed by a *median filter*. The Python package *Scipy* has a module for this.

A median filter is a spatial filter which considers the value in one cell and replaces it with the median value of all the neighbouring cells. As an example: A window of size 3x3, called a kernel, here of light gray color2.1, is applied to the data which in this case is of size 5x5. The values within the kernel are sorted by lowest to highest value and the middle value, the median which in this case is 15, is used to replace the center value of 26 in the kernel, ?? . Then the window slides to the next cell, hence the median filter is known as a sliding-window spatial filter. A median filter is good at removing salt and pepper noise since a single cell value or pixel which is very unrepresentative of the window will not affect the median value significantly.

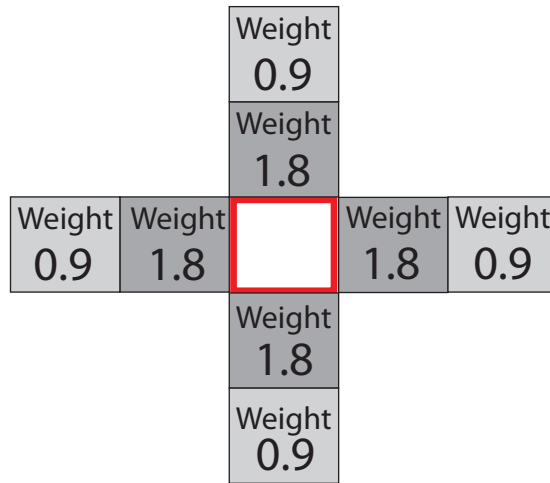


Figure 2.5: Interpolation a value into the empty cell based on the values of the surrounding cells. Cells close to the empty cell gets a higher weight.

10	12	10	20	19
13	12	11	18	18
9	16	26	20	13
13	13	15	16	7
8	11	30	22	17

Table 2.1: Example of median filtering, kernel in light gray and the number to be filtered in dark gray.

2.2.4 Making a grid

A grid is made by adding four corners around the created measuring point representing a gate, as illustrated in figure 2.6. Then each cell is filled with the value of the given point which it encircles and plotted as a color based on that value.

2.2.5 Calculate the Birkeland current density and plotting

To obtain the curl, Stoke's theorem (2.3) was applied to the regularised, gridded ion velocity map. The method is adapted from [46].

$$\int \int_{\sigma} (\nabla \times \mathbf{v}) \cdot \hat{\mathbf{n}} \, ds = \int_L \mathbf{v} \cdot d\mathbf{l} \quad (2.3)$$

Stoke's theorem relates the curl of a vector field in an area σ to the line integral of that vector field around the boundary curve L surrounding the area σ . The

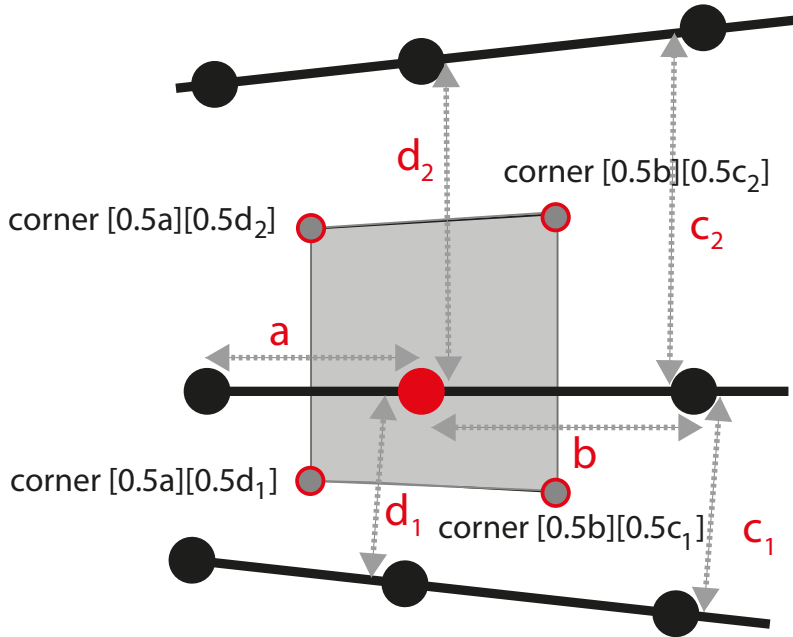


Figure 2.6: Making a cell around the point in red by making four corners based on half the distances of a , b , c_1 , d_1 , c_2 , d_2 , which lengths are illustrated with dotted arrows.

Birkeland current density equation can be rewritten

$$\frac{\mathbf{J}}{\Sigma_p} = \mathbf{B} \cdot (\nabla \times \mathbf{v})$$

which on component form looks like

$$\frac{J_B}{\Sigma_p} = B \sum_i (v_i \cdot dl_i) \quad (2.4)$$

where dl_i is the length of one side, of a total of four, around the area σ , figure 2.7, and the current density is in direction of \mathbf{B} . Each length dl_i must be found. This length can be found using the formula for finding the *bearing*, the straight line between two points on the surface of the Earth following a great circle path [52]. The angle α related to a point can be found according to figure 2.7. $\alpha_i = \theta_{\text{point}} - \theta_{dl_i}$. The green dotted arrow is pointing to the geographical north N.

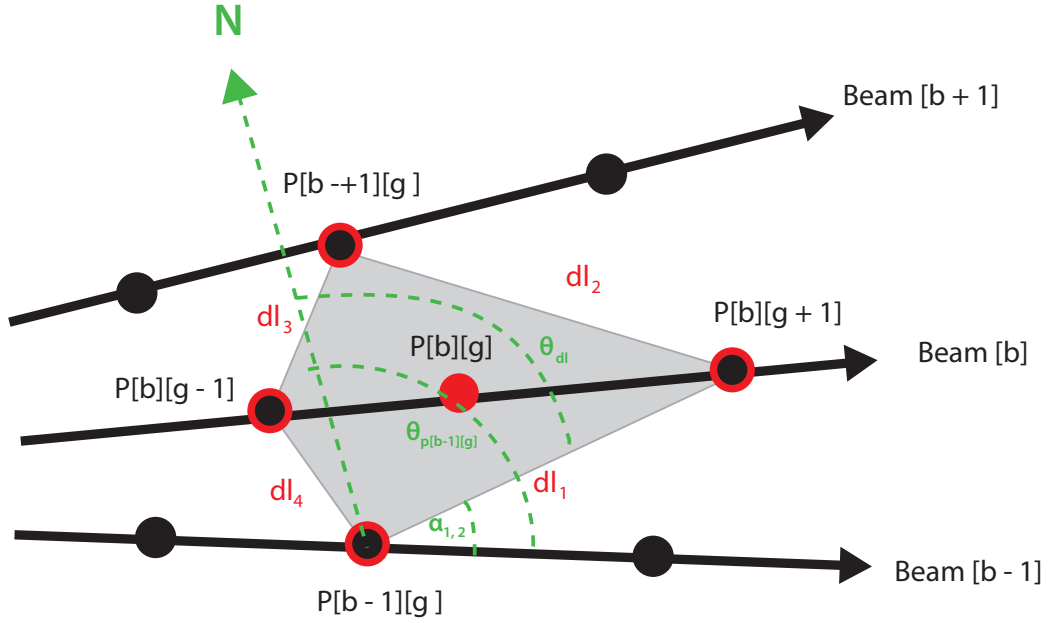


Figure 2.7: Parameterisation to numerically calculate the Birkeland current density around point $P[b][g]$, the shaded area.

$$J_{\text{tot}} = j_1 + j_2 + j_3 + j_4$$

where

$$j_i = \mathbf{v}_i \cdot d\mathbf{l}_i$$

$$j_i = |\mathbf{v}_i| \cdot |d\mathbf{l}_i| \cdot \cos \alpha_i$$

Regarding the plotting

The Python package *Basemap* can be used to draw a map. Basemap uses the latitude and longitude of a point to draw it onto a map. A map projection is chosen and the degree of zoom. To plot the data onto the map, there is a need to convert every plotting point to latitude and longitude, in other words, to find the latitude and longitude of each corner making up each cell of the grid.

Chapter 3

Results

3.1 Presentation of data

The EISCAT Svalbard (ESR) data used in this thesis span around seven hours and thirty minutes over two days, November 30th, 2014 and December 1st, 2014. The raw data consists of among others ion velocity, ion temperature, electron temperature, ion density and electron density.

A grid has been made based on the distance from the radar (ESR) to each measuring point, and missing ion velocity data has been interpolated onto the grid by using the closest existing ion velocity data. Then the ion velocity data has been filtered with a median filter to smooth out salt and pepper noise. The ion velocity has been plotted onto the grid. Each cell in the grid represents one measuring point of data. The filtered ion velocity data has been used to calculate the Birkeland current density which has also been plotted in separate figures, but with using the same grid. Each cell in the grid represents one calculated point of the Birkeland current density. In total, on November 30th, 2014, there are 86 plots of both the ion velocity and the current density, while on December 1st, 2014 there are 59 plots of the same.

3.2 Data selection - identifying two cases

Based on the ACE data, figure 3.1 - summarized in table 3.1, two cases where a component of the IMF is changing sign has been selected. Each case has approximately one hour of data.

I) Case 1 from around 07:10 - 08:15, November 30th, 2014

II) Case 2 from around 09:10 - 10:15, November 30th, 2014

The reason for choosing case 1 is that during this time interval the IMF B_y component is changing sign from positive to negative while the IMF B_z component is

staying positive throughout the interval, figure 3.2. For case 2, the IMF component B_z is changing sign from positive to negative during the time interval while B_y is staying negative throughout the whole interval, figure 3.8 This changing of

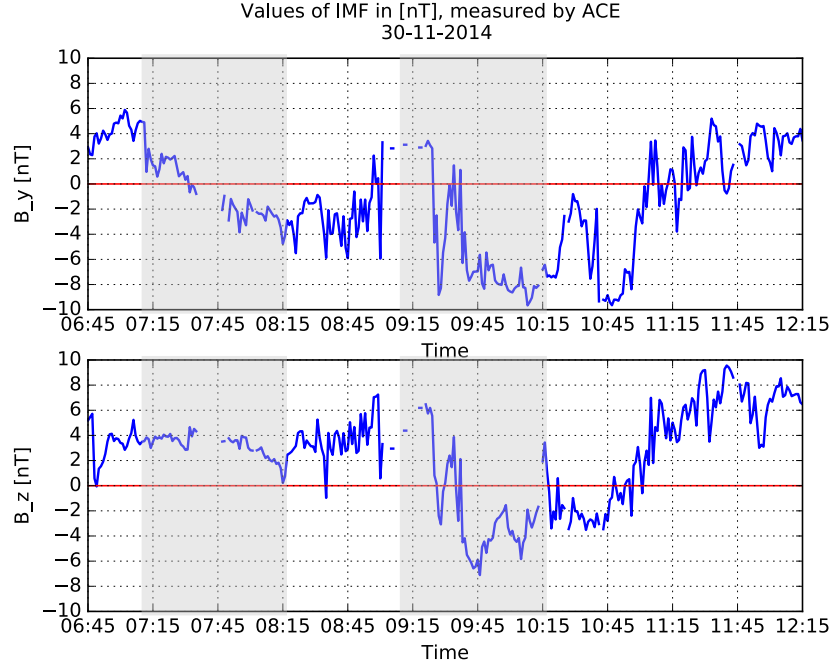


Figure 3.1: Solar wind conditions on november 30th, 2014 as measured by ACE. The times on the x-axis are in universal time (UT). The values on the y-axis are in [nT]. The top panel shows component IMF B_y while the bottom panel shows component IMF B_z . The two shaded areas indicate the two cases, case 1 from 07:10 - 08:15 UT, case 2 from 09:10 to 10:15 UT.

sign in a component of IMF means there is an opportunity for observing a change in the direction of the ion velocities and the Birkeland current density and derive how the plasma convection is.

3.3 Regarding the positioning of the ESR in MLT

The placement of the ESR is converted from UT to MLT by using the definition given in (1.67). For case 1 this means that ESR is positioned at $07:10 + 2:45 \approx 10:00$, figure 3.3. For case 2 the position is $09:10 + 2:45 \approx 12:00$, figure 3.4.

Time (UT)	IMF B_y	IMF B_z
07:10 - 07:30	+	+
07:30 - 09:15	-	+
09:15 - 09:30	+ (uncertain)	+
09:30 - 10:15	-	-
10:15 - 11:00	-	-
11:00 - 12:00	+	+

Table 3.1: November 30th, 2014: Signs of IMF B_y and B_z , based on data from (OMNIweb). Case 1, where B_y goes from positive to negative, between 07:10 and 09:15 is shaded. Case 2, where B_z goes from positive to negative, between 09:10 and 10:15 is shaded.

3.4 A quick guide for plots

The coloring of the ion velocity plots and Birkeland current density plots follow a logic, which may or may not be evident. In order to minimize confusion, a short guide on how to read the coloring is presented in figure 3.5.

3.5 Case 1: 07:10 - 08:15

The ion velocity

Starting at around 07:10 UT the radar is looking eastward. The ion velocity is in a westward direction towards the radar, figure 3.6 (a). In the middle section of the radar swipe the ion velocity is around 500 m/s. Out on the edges of the swipe the velocity is a little lower, with areas of almost no motion with velocities much closer to 0 m/s.

From 07:15, figure 3.6 (b), at latitudes just north of ESR, there is an indication of eastward movement coming from the west. This shows in the most western section of the radar swipe. The area is around 100 km in width and eastern length and will be named Structure 1. Its eastward velocity is around 500 m/s away from the radar.

From 07:21, figure 3.6 (c), Structure 1 has moved more eastward and grown in eastward direction, as the most western section of the radar swipe still shows eastward movement. The velocity away from the radar is still around 500 m/s and the latitude is just north of ESR. There are a few other areas showing movement away from the radar.

As seen in the time interval from around 07:32 to 07:47, the eastward movement of Structure 1 is continuing its journey to the east into an area which previously had only westward ion velocity. It is not growing any more, as seen in figures 3.6 (d) and (e). Its eastward velocity starts to fall in the last parts of

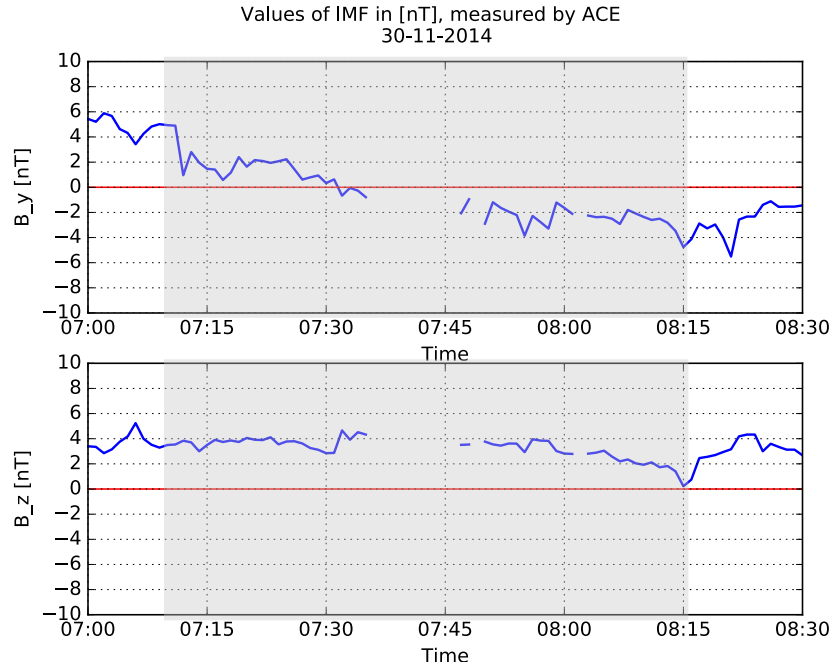


Figure 3.2: The shaded area is the time interval for the 1st case, data from November 30th, 2014 as measured by ACE, time in UT.

the time interval from around 200 m/s and towards 0 m/s.

Figure 3.6 (f) shows a band of velocities close to zero in the path of Structure 1's moving from west to east. This band is stretching in almost the entire radar length view at a geographic latitude just north of 80° . A few minutes past 08:00, in the remnant of Structure 1's path, the radar shows eastward ion velocities 3.6 (g) of around 400 m/s. This band is several hundred km in east-west direction and about 100 km wide.

Shortly after 08:00, figure 3.6 (h), in Structure 1's path, there is a broader area of ion velocities of 500 m/s away from the radar. There is also an area to the southeast view of the radar swipe which shows eastward ion velocities of more than 500 m/s. After 08:10 the structures are starting to blend into each other and the ion velocities are falling in both directions, figure 3.6 (i).

The Birkeland current density

At around 07:10 there are no major structures of higher current densities. figure 3.7 (a).

From 07:15, figure 3.7 (b), east of ESR, there are four structures that can be identified and named. Up 1 to the north and several hundred km long in east-west direction. Down 1, south of Up 1, and much less visible and much shorter in length. Up 2, south of Down 1, and also thin, short and small. Down 2, south of Up 2, which is around 100 km wide north-south and some 100 km

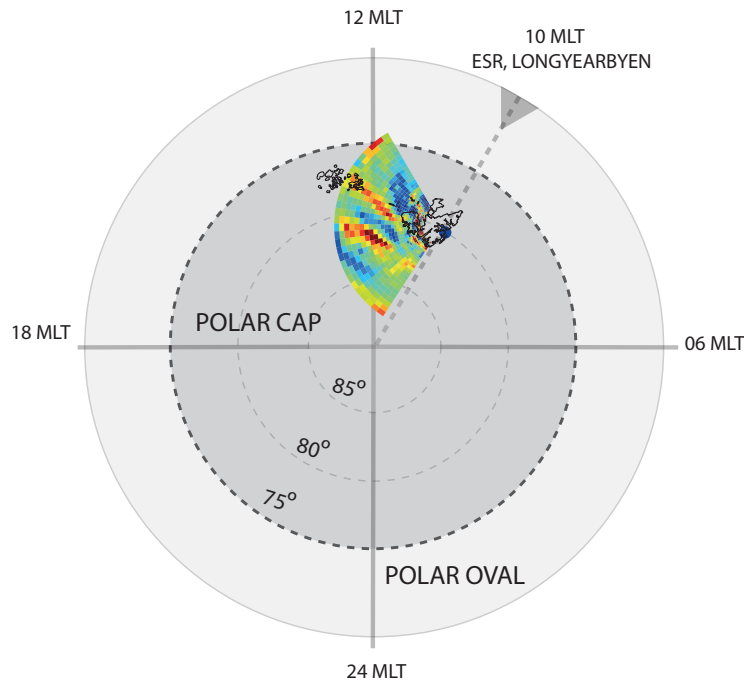


Figure 3.3: Case 1: Illustration of the ESR placement in magnetic coordinates, MLT \approx 10:00. The direction of the radar swipe is an approximation for case 1.

long east-west.

From 07:21, figure 3.7 (c), the same four structures can be identified. Up 1, more north and west than earlier, and much fainter and shorter in east-west direction. Down 1, south of Up 1, more visible and longer, about 200 km east-west. Up 2, south of Down 1, thin, but 400-500 km in east-west direction at around 80° . Down 2, south of Up 2, still broader than the others 00 km wide north-south and some 100s km long east-west.

From around 07:32, figure 3.7 (d), the four named structures are all very visible and following the same pattern as described for the earlier figures. They all have an east-west length of more than 500 km and a south-north width of 40-50 km. Their values are clearly in the 10^{-6} A/m² range. This configuration is still present around 07:45, 3.7 (e), but it seems shifted a bit more to the north.

Figure 3.7 (f) shows four structures of the same ordering as before, but more split up in east-west direction and more disrupted and lower current density values.

At 08:03, figure 3.7 (g) there are now two structures showing clearly, which could be Down 1 and Up 2, while the other identified structures are less visible and fainter.

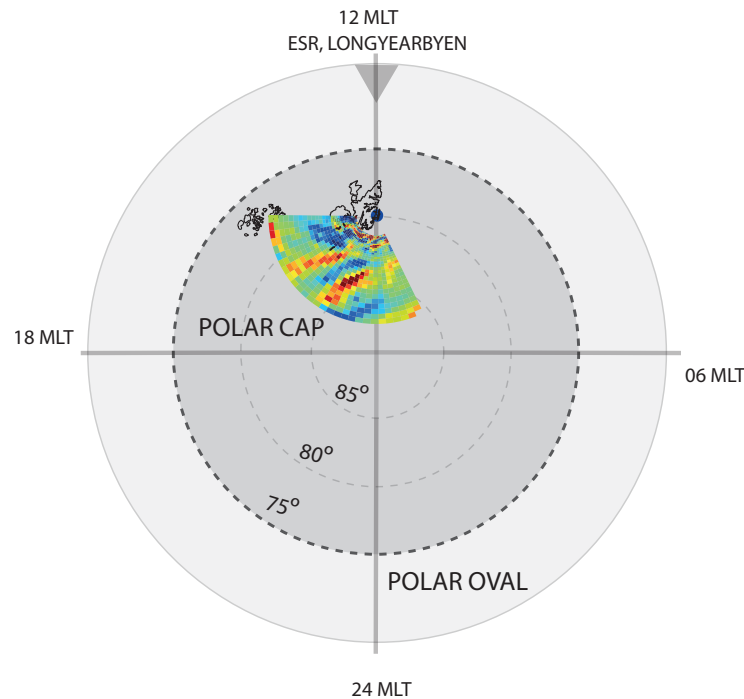


Figure 3.4: Case 2: Illustration of the ESR placement in magnetic coordinates, $\text{MLT} \approx 12:00$. The direction of the radar fan is an approximation for case 2.

Some 7 minutes later, Down 1 is a bit smaller. Up 2 has grown in width and is still several hundred km long, and south of Up 2 there is an area with structures reminding of Down 2, figure 3.7 (h).

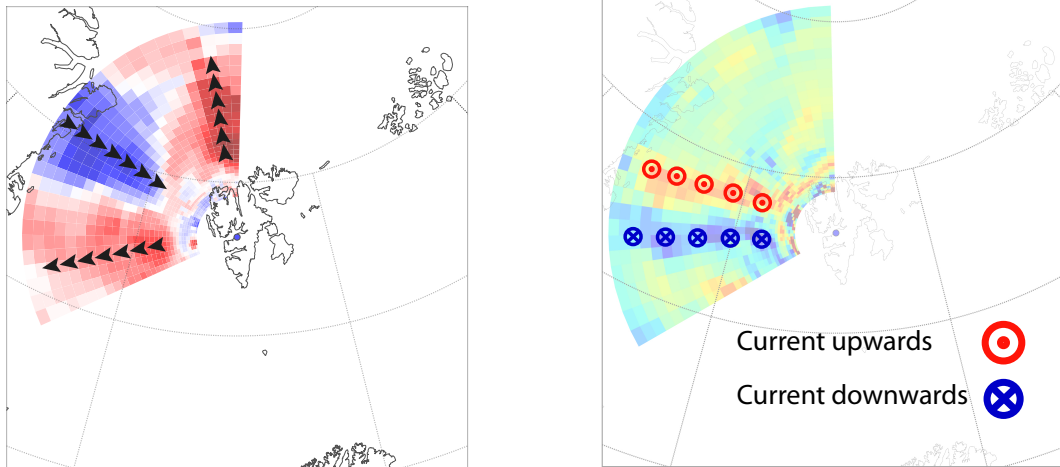
The last plot shows the current density from 08:19 - 08:22 and all the structures has basically vanished.

3.6 Case 2: 09:10 - 10:15

The ion velocity

Case 2 is starting at around 09:10 UT when the radar is looking in a north western direction, more towards the geomagnetic pole. The ion velocity is showing one major area of equatorward motion to the north while south and west there is westward motion 3.9 (a). The equatorward velocity is around 200 m/s while the south-westward motion is higher, around 750-1000 m/s. The pattern of westward moving ions is maintained at around 09:22, figure 3.9 (b), while the equatorward motion at the north reduced and is closer to 0 m/s.

At 09:35 the westward motion is still present in the southern view of the radar



(a) The direction of the ion velocity is indicated by the colors blue or red. As this guide shows, **blue** color means the ions are moving **toward** the radar marked as a blue dot, while **red** means **away** from the radar

(b) The direction of the Birkeland current density is indicated by the colors blue and red. **Blue** means that the current is going **downward** into the paper, while **red** means **upward** out of the paper

Figure 3.5: The figures (a) and (b) are for illustration and guidance only, not for further study. (a) Guide to the ion velocity plots. (b) Guide to the current density plots.

swipe. The equatorward motion in the north is present again, showing speeds of a few hundred m/s. 3.9 (c),

From 09:38 - 09:41 3.9 (d), the northern equatorward motion has expanded more to the west and south. There is westward motion south of this which is enclosed by a eastward movement south of that again.

Around 09:41 the configuration which lasted from around 09:10 to 09:40 is switched around, so that the westward ion motion is now to the north, with a small remain of eastward motion in the middle of it. The westward speed is around 750 m/s. The southern parts of the radar swipe now shows eastward motion only, 3.9 (e). This configuration evolves further into three separable areas in the radar view. In the south, the eastward motion in 3.9 (e) is replaced by ions moving away from the radar in a western direction. North of this there is eastward motion, while north of this eastward motion there is pole-westward motion. 3.9 (f).

Some 5-7 minutes later, closer to 10:00, the configuration is further split up to four separable areas: eastward motion to the south, then some westward motion north of that before another area with eastward motion and then, in the northernmost boundary area of the swipe, there is an area of poleward-westward

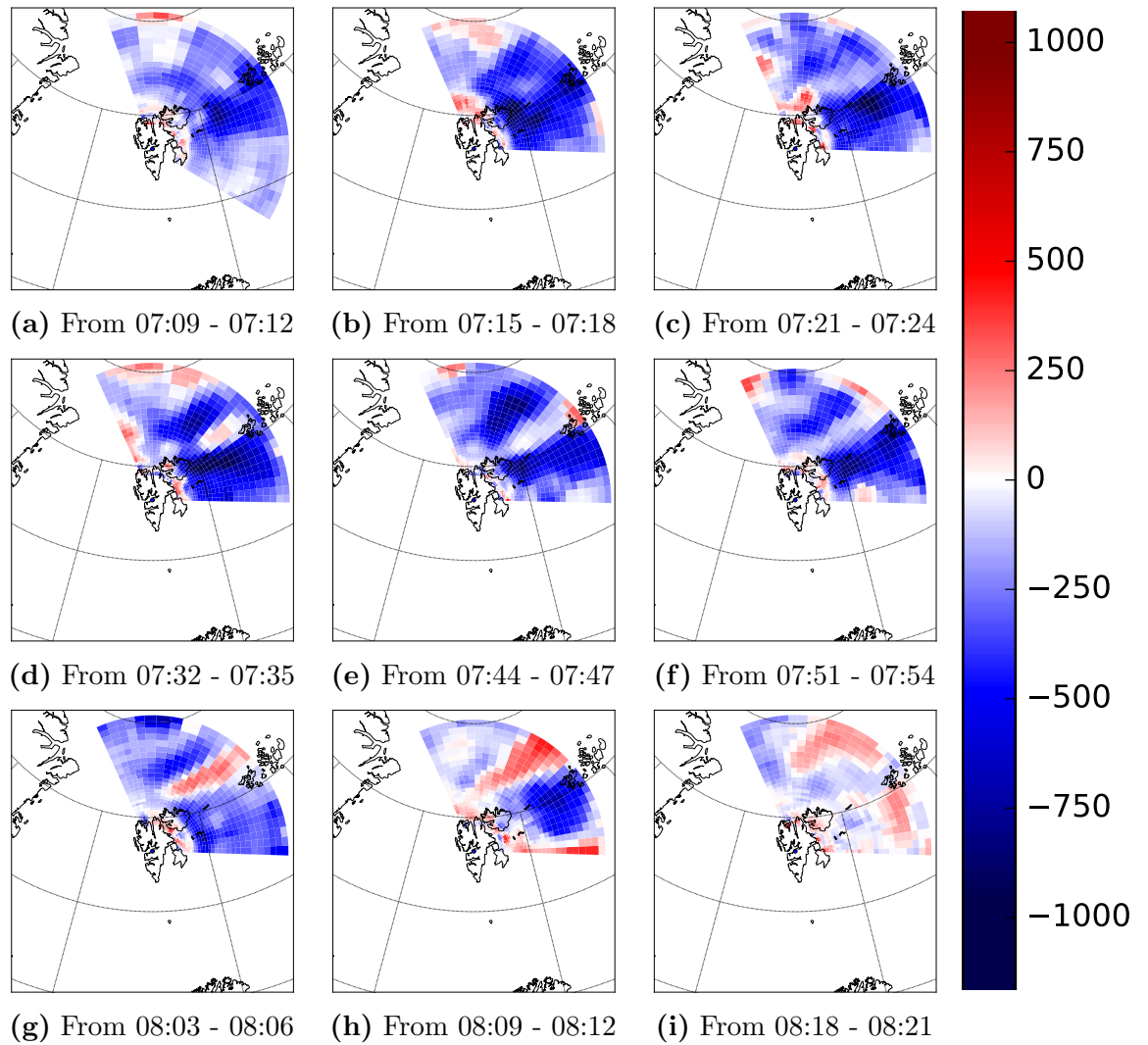


Figure 3.6: Case 1: 07:10 to 08:20. Ion Velocity in [m/s], 30.11.14

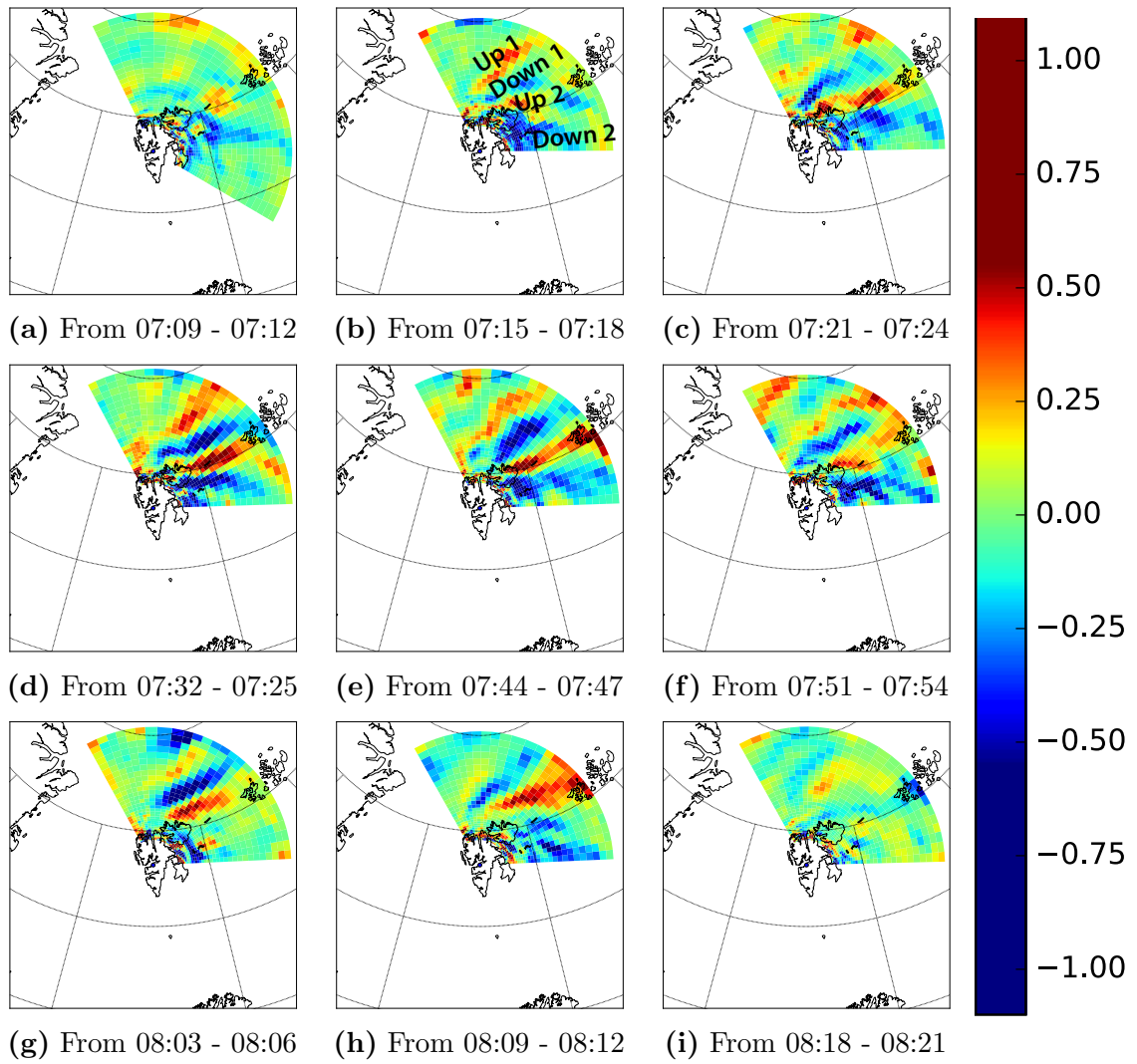


Figure 3.7: Case 1: 07:10 to 08:20. Current density in 10^{-6} [A/m²], 30.11.14

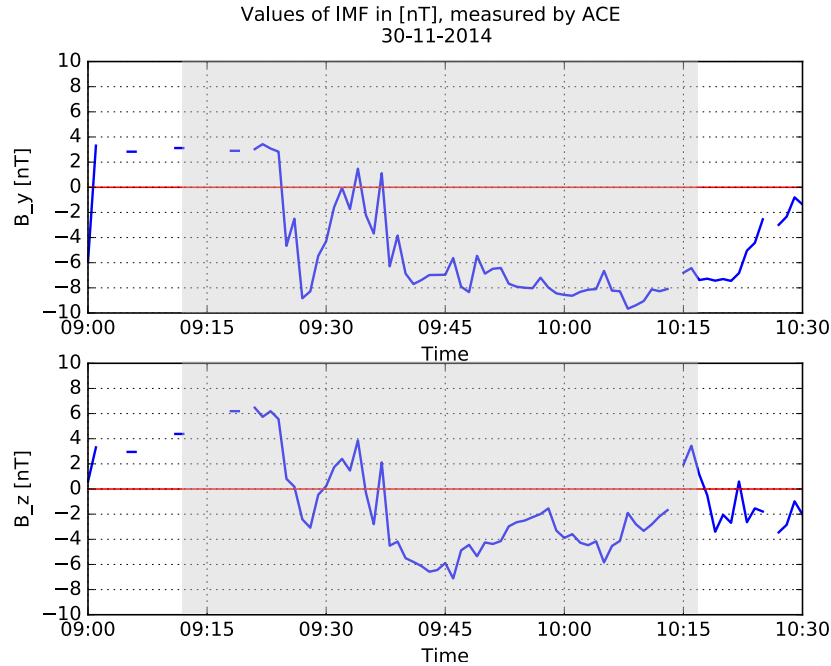


Figure 3.8: The shaded area is the time interval for the 2nd case, data from November 30th, 2014 as measured by ACE, time in UT.

moving ions directed away from the radar, 3.9 (g).

After 10:00 the configuration is a lot more fragmented, with bits and dots of eastward and westward motion scrambled together and no larger structures, 3.9 (h). From around 10:12-10:15 this clears a bit up into a westward motion to the south and eastward motion north of that. 3.9 (i).

The Birkeland current density

Case 2 starts with just some minor structures of currents up and down, most notable a small and narrow band of positive current density, stretching across half the span width of the radar swipe, 3.10 (a). Around 10 minutes later this has fractured into smaller bits, 3.10 (b) where no areas in the radar view show any large structures of significant higher current density. This configuration is still most dominating as the time passes 09:35, figure 3.10 (c).

Closer to 09:40 there is an evolution of two structures in western direction of the radar. The southern one with a current density structure downward and just to the north of this, a structure with upwards pointing current density, figure 3.10 (d). Both of them are hundred kilometers long. Past 09:40 these same two structures are larger and more visible. They are stretching in almost full radar view length, over 500 kilometers, with current densities down and up in the range of 10^{-6} [A/m²], figure 3.10 (e). From 09:50 the configuration becomes more mixed up, with bits and pieces of upward moving current density in the

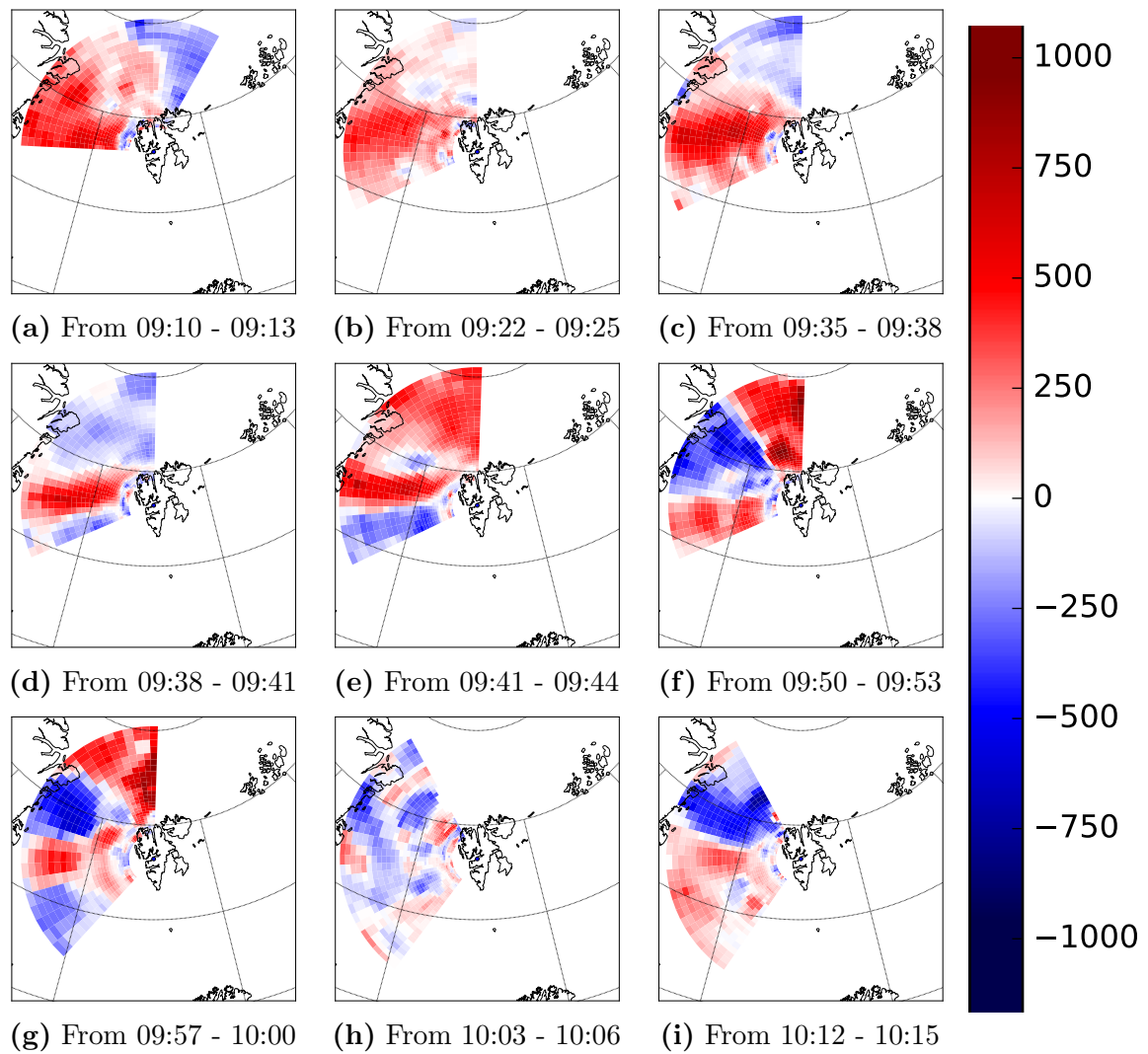


Figure 3.9: Case 2: 09:10 to 10:15. Ion velocities in [m/s], 30.11.14

same area as ten minutes earlier. There is a larger downward current density to the north of this which was not present ten minutes earlier, figure 3.10 (f).

Closer to 10:00 the picture becomes more fragmented, with smaller patches of upward and downward current densities. It seems to be upward current density in the south-middle section of the swipe and more downward current density to the north of this, a configuration which may remind of the one at 09:50, but more fragmented, figure 3.10 (g).

After 10:00 the remnants of any earlier structures are breaking up and there are some smaller patches of upward and downward current density scrambled together towards the middle of the radar swipe, in pole-westward direction. figure 3.10 (h). Closer to 10:15 there is a formation of upward current density in approximately the same area of the swipe as at past 09:40. This formation is some 100 km in length and shows values in the 10^{-6} [A/m²] range, 3.10 (h).

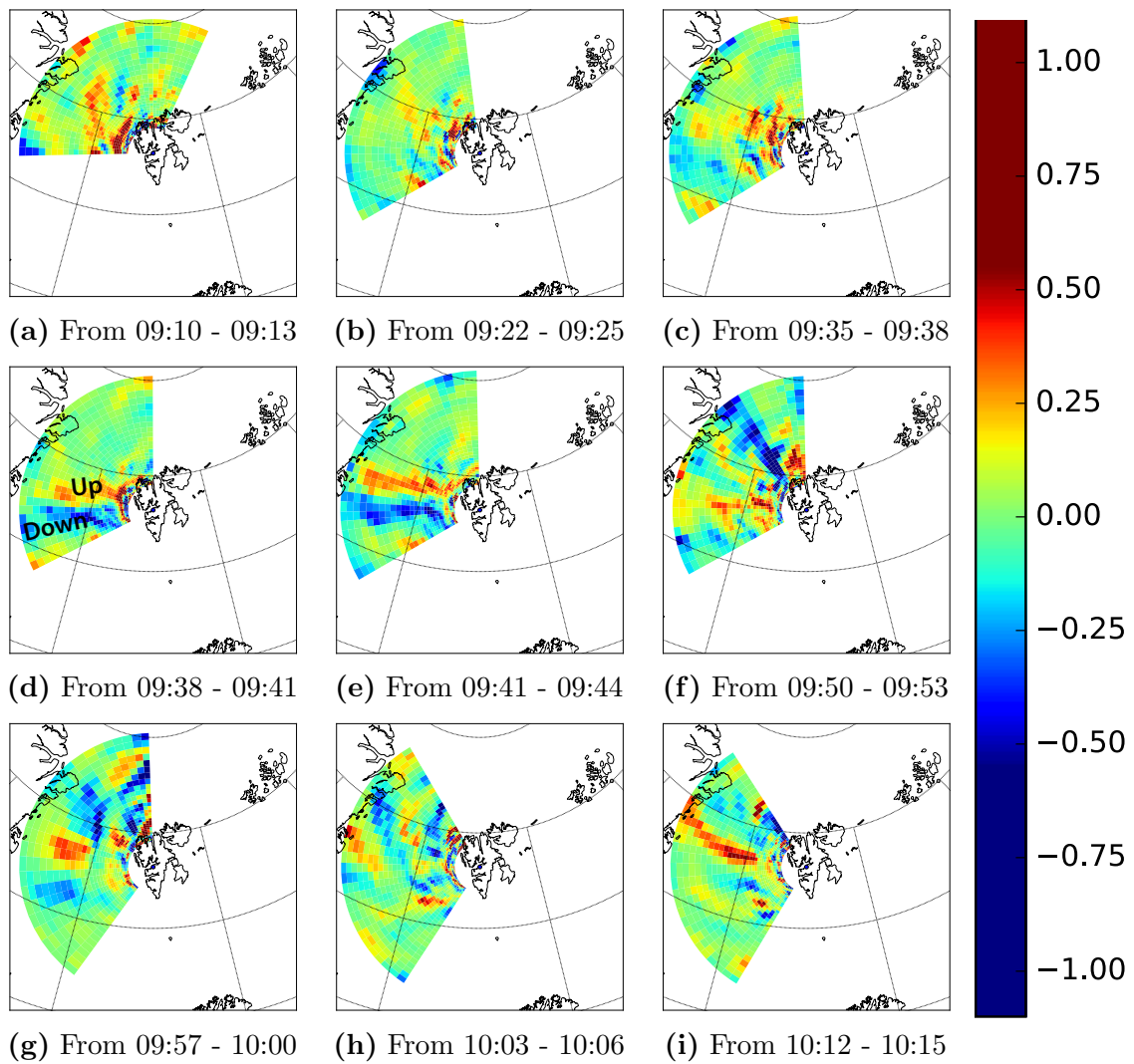


Figure 3.10: Case 2: 09:10 to 10:15. Current density in 10^{-6} [A/m²], 30.11.14

Chapter 4

Discussion

The two cases selected in the previous chapter are two samples from a larger amount of raw data. During the initial analysis done on the IMF data and on every velocity and current density plot, these two cases were considered the most interesting for studying dayside field-aligned Birkeland current density patterns. The two selected cases show structures of upwards and downwards current density of magnitude [$\mu\text{A}/\text{m}^2$]. These FACs are a part of the magnetospheric-ionospheric current system, figures 1.8 and 1.9. A larger current density indicates a change in the reconnection rate or reconnection site and an associated reconfiguration of the plasma convection.

4.1 Regarding the height-integrated Pedersen conductivity Σ_P for both cases

In this thesis there is an assumption that the Pedersen conductivity Σ_P is more or less a constant with value closer to one siemens and that there are no gradients in the conductivities. This meant that the last two terms in (1.57) could be ignored. "On the right, last two terms constitute the 'ionospheric component' of the FAC, while the first term is the 'magnetospheric component'..." and later on "...the FAC magnetospheric component is expected to be substantially greater than the ionospheric component." [46]. The height-integrated Pedersen conductivity depends on the plasma density since

$$\Sigma_P = \int \sigma_P dH \quad (4.1)$$

The main contribution to the value of Σ_P comes from the E region and F region where the plasma density is higher, the lower ionospheric component contributes less to the values of Σ_P . When there is a reconfiguration of the convection cells, there is an associated particle precipitation in the polar cap. This will change

the plasma density and to some degree affect σ_P .

However, the choice of letting the height-integrated Pedersen conductivity be around one siemens does not significantly change the outcome of the calculated Birkeland current densities. The current densities have values from $\pm 10^{-6}$ [A/m²] and smaller. The height-integrated Pedersen conductivity is not of magnitudes large enough to change the exponent so the values for the calculated Birkeland current densities are within right orders of magnitudes. Furthermore, we are interested in the morphology of the currents, not their magnitudes.

4.2 Case 1: Northward B_z , change in B_y

The first fifteen minutes of case 1, from around 07:10 - 07:25, figures 3.6 (a) - 3.6 (c), show a steady westward ion motion with minor movement in the north directed away from the radar. There are no larger structures present and the configuration of ion velocity seems mostly uniform with westward motion.

There are some areas showing lower ion velocities, but these are in the outer edges of the radar swipe. The line of sight at the radar's outer edges is more restrictive than in the middle. This means that the ion motion at the outer edges might be the same as in the center of the swipe, but this will not show as such. The radar detects backscatter directed straight towards it. If the outer edges of the radar swipe has the same ion velocity as the middle areas, the backscatter will not be directed straight towards the radar. Thus, only a component of the velocity will be detected and the radar will record a lower ion velocity. Since there are no significant changes in the IMF parameters during the first quarter of case 1, this could be the situation, figure 3.2.

Given the assumption above, this kind of configuration with mostly westward motion could be a sign of tilted convection cells resembling figure 1.20a. The anti sunward convection is caused by IMF B_z being positive. The tilted reversed cell pattern is influenced by IMF B_y being positive, figure 1.15 (c).

This tilted convection pattern has a lobe cell on the dawn side and a larger, more circular cell on the dusk side. It may include one or two smaller cells between the two dominating cells. Due to the time being around 10:30 MLT, the plasma motion of the larger duskward cell is more likely to be within the view of the ESR. The ESR field of view could be enclosed in the sunward convection motion and not register the lobe- or smaller cells.

The situation with uniform westward ion velocity starts to change from around 07:30, figures 3.6 (e) - 3.6 (g). An area in the middle of the radar swipe is clearly showing eastward ion velocities. The trend continues for the next thirty minutes and coincides with the changing IMF component B_y going from positive to negative at 07:30, figure 3.2.

When IMF B_y is changed from positive to negative the tilted reversed convection cell pattern is reversed. The lobe cell will be on the dusk side while

the larger, more circular cell will be on the dawn side. During this change, the plasma flow will be more turbulent and this may introduce motion in other directions than just westward. The structure appearing as eastward motion could be a response to the change in B_y and reconnection happening more to the dusk side.

As time goes, the eastward moving structure broadens, and in the last figure 3.6 (i), the area of eastward motion is covering almost half the radar swipe. IMF B_y is staying negative after 07:30 so this could be an indication that the newly formed convection pattern has evolved into a more stable cell configuration with positive B_z and negative B_y , figure 1.15 (a).

Current density

During the first ten minutes four structures are identified as areas with higher Birkeland current densities. These structures appear as the IMF B_y component is steadily falling from a maximum of +6 nT towards zero and eventually negative values, figure 3.1. There are no larger changes in the ion velocity motion from 07:15 - 07:30, but the minor changes could be enough to start upwards and downwards structures of current densities.

The emergence of the four Birkeland current density structures seen in 3.7 (b) is likely connected to the falling IMF component B_y . Generally speaking, the current density is larger when a change in IMF is happening. We hypothesize that this is due to a reconfiguration of the plasma convection within the polar cap.

As B_y continues to fall and passes zero, the four structures named Up 1, Down 1, Up 2 and Down 2, grow and show higher current density values of $\pm [\mu\text{A}/\text{m}^2]$ than the surrounding areas. This evolution may be seen as an continuation of the ongoing convection reconfiguration.

When looking from the north to the south, this pattern of upward, downward, upward, downward current densities looks very much alike FACs associated with a phenomena called Reversed Flux Transfer Event (RFE) [64]. This pattern is shown in figure 4.1 which is copy of figure 6 d) [Rinne et al.] just flipped around in a north-south orientation and colored for easier viewing. According to [Rinne et al.] this configuration is directly linked to pulses of reconnection during negative IMF B_z , where a new pulse of reconnection pushes the previous reconnected flux tube poleward. The stretching in length is proposed to happen due to plasma convection and magnetic tension force. However, this phenomena has not been registered during a positive IMF B_z , but it should be mentioned because the current morphology is similar to that proposed by [Rinne et al.]. Could the existence of a changing IMF B_y from positive to values of -6 nT relocate the site of reconnection to a dayside reconnection and not follow the usual lobe reconnection associated with a positive B_z ? If so, could the current density distribution in

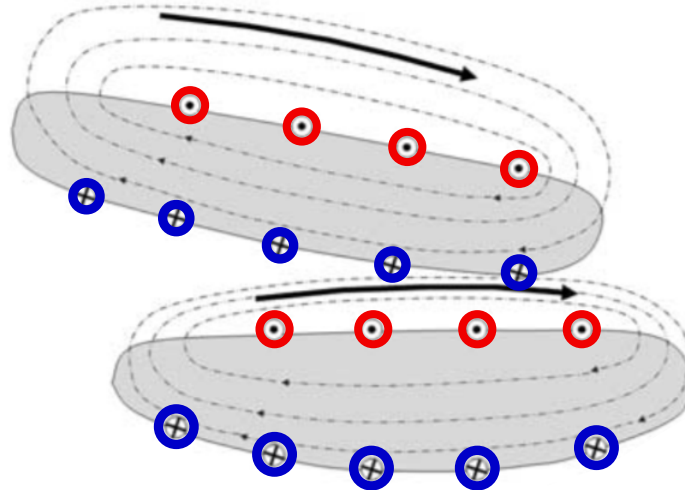


Figure 4.1: The configuration of an Reversed Flux Transfer Event as proposed by [64] with poleward direction towards the top of the page. Plasma convection is indicated by the large arrow.

case 1 be a glimpse of a RFE?

As seen in article [2] (figure 5; upper panel (NE) and (NW)) the statistical distribution of Birkeland currents for a northward IMF is different for IMF $+B_y$ and $-B_y$. The polarity of the northern most FAC in the center of the plots is reversed. It also follows that when moving from one cell configuration to another, the direction of the flow will have to change.

The ESR field of view is in the range of 600 km, while the statistical patterns are derived for on much larger scale, in orders of $20^\circ - 30^\circ$ in latitude. This means that the pattern of the Birkeland current density for is not easily comparable with the statistical model. This does not mean that case 1 is a misfit to the statistical pattern, figure 1.19. It does mean that the resolution of the ESR is much higher, and the numbers of small structures visible to the ESR are much larger, than data collected by satellites or the SuperDarn radar system which are used for the larger statistical Birkeland current patterns.

Passing 08:00 the currents slowly seems to be vanishing. The four structures become three, then two or basically one larger structure, up until times past 08:15 where the configuration resembles that of one hour earlier. This could indicate that the has motion has stabilized and system has adjusted to the changed IMF B_y direction by reconfiguring the convection cells.

Another piece of the puzzle could be a contraction of the polar cap due to continuing positive value of B_z . The current density structure named Up 2, seen in figure 3.7 (h) seems to be moving poleward and out of sight of the radar. This could be interpreted as a contraction of the polar cap.

4.3 Case 2: Negative B_y , change in B_z

Case 2 begins around 09:10 and shows two distinct regions of motion. In north-western direction the ion velocity is westward while the north-eastern sector has ion motion southward. This configuration seems quite stable for around 25 minutes, figures 3.9 (a) - 3.9 (c). The IMF B_y is negative and B_z is positive which means the convection cells are tilted and reversed and the plasma flow is sunward. As seen in figure 1.16 and taking into account the placement of the ESR close to 12 MLT, figure 3.4, this kind of ion velocities seems to fit the description of reversed convection cells.

From 09:38 there are indications of a change in ion motion as the southward motion to the north is expanding into westward direction. However, the major difference is seen a few minutes later at 09:41 when the configuration is totally changed around, figures 3.9 (e) - 3.9 (f). This change is happening because IMF B_z is going from positive to negative around 09:35-09:40. When IMF B_z is in southward direction reconnection happens at the magnetopause. This configuration of negative B_z value will expand the polar cap and enlarge the convection cells in addition to change from sunward drift motion to anti-sunward plasma drift. Looking at figure 3.9 (d) and 3.9 (e), the westward motion could be moving poleward followed by eastward motion south of it, and further to the south of that, another area of westward ion motion as seen in figure 3.9 (f) until close to 10:00, figure 3.9 (g).

After 10:00 the areas of collective motion become smaller and it becomes more difficult to find any particular pattern of ion motion, figure 3.9 (g).

Current density

The first 25 minutes show no larger structures of the current densities. There are some minor signs of upward current density which are mirrored by less visible patches of downward current density, but over all the situation reminds of the central area of figure 1.20b with quiet conditions. There data from ACE during this time interval seems to show a stable IMF, although the measurements from around 09:10 to 09:25 are inadequate.

The two structures of upward and downward current density appearing in figure 3.9 (d) arise in the same time interval as IMF B_z changes from positive to negative while IMF B_y stays negative. As explained for the ion motion this change in IMF B_z will force a reconfiguration of the plasma convection cells

in the polar cap. The onset of a negative IMF B_z around 09:40 will drive an expansion and reversal of the existing convection cells as well as change the site of reconnection on the magnetopause. As long as IMF B_z is southward the expansion will continue as the reconnection site moves at lower latitudes opening up new field lines, e.g 'peeling an onion'.

Figures 3.9 (d) - 3.9 (g) could show a pattern of poleward moving pairs of upward and downward current densities indicating the opening of new field lines and an expansion the polar cap. This would move the Region 1 and Region 2 field-aligned currents going up and down between the magnetosphere and ionosphere, figures 1.9 and 1.8.

After 10:00 the IMF B_z and B_y holds a steadier configuration up until 10:15. The current structures become less visible after 10:00, reflecting the more lower and less uniform ion velocities. As a final note, when getting to 10:15 the ion velocities again show more activity, but from 10:15 the IMF B_z is moving towards zero and possible a change of sign explaining this.

Chapter 5

Conclusion

In this thesis radar data from the Svalbard EISCAT Radar has been processed and analysed to calculate Birkeland current densities. The results have been plotted on a map showing the latitudes and longitudes of the current densities. A selection of two cases was done and these two cases were discussed in light of existing theory and statistical derived models of Birkeland current distributions.

The case studies have revealed a strong relationship between changes in polarity of IMF B_y and B_z respectively and the emergence of higher Birkeland current densities.

The first case when IMF component B_z is northward, but IMF component B_y changes from positive to negative value. A proposal is made that larger Birkeland current density indicates a reconfiguration of the convection patterns due to a change in the reconnection rate or site. As B_y continues to fall, four clearly visible structures of enhanced Birkeland current densities appears. They continue to enlarge and show higher values of current densities in order of [$\mu\text{A}/\text{m}^2$] and this configuration lasts for almost 40 minutes. Their existence fits the view of an ongoing reconfiguration of the plasma convection in the polar cap. Another hypothesis worth mentioning is the phenomena called Reversed Flux Transfer Event (FTE) which shows a very similar configuration of current distribution, but only has been seen during B_z southward. Case 2 shows the same influence regarding change in IMF and emergence of current densities upward and downward. When B_z changes sign there is an response in the plasma drift almost immediately as seen by the ESR, giving rise to the creation of current density structures.

As most of the statistical models of Birkeland current distributions are on a larger scale, the patterns seen in this thesis cannot be compared directly to existing models. However, the current density distributions analyzed do not contradict the large-scale statistical models and it would be of interest to do more work with tools like the ESR which detects motion on much smaller scales.

Chapter 6

Outlook

When the Birkeland current densities were calculated there was an argument for ignoring the 'ionospheric component'. If one were to find the value of σ_P based on measurements by the ESR, the method would be to integrate the right hand side of (4.1). The limits for the integral would be from the lowest radar measurement point to the highest and the area would be the radar swipe, and this would be repeated for every swipe. This fell out of the scope of this thesis and is left as a future activity. It could be coupled with the already existing programming scripts evolved for this thesis and collected in a module for analyzing ESR data and plotting the current densities. This could give more data sets and look for further traces and patterns in the current density configurations.

Bibliography

- [1] H. 177/ch3 4.html. A MEETING WITH THE UNIVERSE.
- [2] B. J. Anderson, H. Korth, C. L. Waters, D. L. Green, and P. Stauning. Statistical Birkeland current distributions from magnetic field observations by the Iridium constellation. *Ann . Geophys*, 26:671–687, 2008.
- [3] A. Brekke. *Physics of the upper polar atmosphere*. Springer-Verlag, 2013.
- [4] L. B. N. Clausen, J. B. H. Baker, J. M. Ruohoniemi, S. E. Milan, and B. J. Anderson. Dynamics of the region 1 Birkeland current oval derived from the Active Magnetosphere and Planetary Electrodynamics Response Experiment (AMPERE). *Journal of Geophysical Research: Space Physics (19782012)*, 117(A6), 2012.
- [5] L. B. N. Clausen, J. B. H. Baker, J. M. Ruohoniemi, S. E. Milan, and B. J. Anderson. Dynamics of the region 1 Birkeland current oval derived from the active magnetosphere and planetary electrodynamics response experiment (AMPERE). *Journal of Geophysical Research: Space Physics*, 117(6):1–11, 2012.
- [6] H. content/uploads/2016/05/High Latitude-IS-Radars.pdf. High Latitude.
- [7] S. W. H. Cowley. Magnetosphere-Ionosphere Interactions’ A Tutorial Review.
- [8] S. W. H. Cowley. Excitation of Flow in the Earth’s Magnetosphere-Ionosphere System: Observations by Incoherent-Scatter Radar. In *Polar Cap Boundary Phenomena*, pages 127–140. Springer Netherlands, Dordrecht, 1998.
- [9] D.J.Southwood. The ionospheric signature of flux transfer events. *Journal of Geophysical Research*, VOL. 92, 1987.
- [10] J. W. Gjerloev, S. Ohtani, T. Iijima, B. Anderson, J. Slavin, and G. Le. Characteristics of the terrestrial field-aligned current system. *Ann. Geophys*, 29:1713–1729, 2011.

- [11] M. Goossens, F. Mittlebach, and A. Samarin. *The Latex Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [12] G. D. Greenwade. The {C}omprehensive {T}ex {A}rchive {N}etwork ({CTAN}). *TUGBoat*, 14(3):342–351, 1993.
- [13] [Http://geomag.nrcan.gc.ca/mag fld/arctics-en.php](http://geomag.nrcan.gc.ca/mag fld/arctics-en.php). Magnetic field in the Arctic regions.
- [14] [Http://roma2.rm.ingv.it/en/research_areas/4/ionosphere](http://roma2.rm.ingv.it/en/research_areas/4/ionosphere). INGV Roma2 — Ionosphere.
- [15] [Http://roma2.rm.ingv.it/en/research_areas/4/ionosphere](http://roma2.rm.ingv.it/en/research_areas/4/ionosphere). Ionosphere electron density.
- [16] [Https://eiscat3d.se/drupal/sites/default/files/ISRSchool2016/19-Enell-GUISDAP_lecture.pdf](https://eiscat3d.se/drupal/sites/default/files/ISRSchool2016/19-Enell-GUISDAP_lecture.pdf). Introduction to GUISDAP.
- [17] [Http://slideplayer.com/slide/9382611/](http://slideplayer.com/slide/9382611/). ESS 200C Lecture 13 The Earth’s Ionosphere - ppt download.
- [18] [Https://omniweb.gsfc.nasa.gov/ow_min.html](https://omniweb.gsfc.nasa.gov/ow_min.html). SPDF - OMNIWeb Service.
- [19] [Https://sdo.gsfc.nasa.gov/](https://sdo.gsfc.nasa.gov/). SDO — Solar Dynamics Observatory.
- [20] [Https://www-spod.gsfc.nasa.gov/Education/bh2_6.html](https://www-spod.gsfc.nasa.gov/Education/bh2_6.html). Brief History: The Polar Aurora.
- [21] [Https://www.eiscat.se](https://www.eiscat.se). EISCAT <https://www.eiscat.se>.
- [22] [Https://www.hindawi.com/journals/ijge/2011/971302/fig1/](https://www.hindawi.com/journals/ijge/2011/971302/fig1/). Electrodynamical Coupling of Earth’s Atmosphere and Ionosphere: An Overview : Figure 1.
- [23] [Http://wdc.kugi.kyoto-u.ac.jp/igrf/gggm/index.html](http://wdc.kugi.kyoto-u.ac.jp/igrf/gggm/index.html). Transformation of Coordinate.
- [24] [Http://www.geomag.bgs.ac.uk/cgi-bin/coord_calc](http://www.geomag.bgs.ac.uk/cgi-bin/coord_calc). Geomagnetic Coordinates.
- [25] [Http://www.haystack.mit.edu/atm/mho/instruments/isr/isTutorial.html](http://www.haystack.mit.edu/atm/mho/instruments/isr/isTutorial.html). Incoherent scatter radar.
- [26] [Http://www.nap.edu/catalog/13060](http://www.nap.edu/catalog/13060). Solar and Space Physics, aug 2013.
- [27] Ieda and Al. Approximate forms of daytime ionospheric conductance. *Journal of Geophysical Research: Space Physics*.

- [28] T. Iijima and T. A. Potemra. Field-aligned currents in the dayside cusp observed by Triad. *Journal of Geophysical Research*, 81(34):5971–5979, dec 1976.
- [29] J.Moen, Y.Rinne, H.C.Carlson, K.Oksavik, R.Fujii, and H.Opgenoorth. On the relationship between thin Birkeland current arcs and reversed flow channels in the winter cusp/cleft ionosphere. *Journal of Geophysical Research*, VOL. 113, 2008.
- [30] M.-B. Kallenrode. *Space Physics : an Introduction to Plasmas and Particles in the Heliosphere and Magnetospheres*. Springer Berlin Heidelberg, 2004.
- [31] Y. Kamide and W. W. Baumjohann. *Magnetosphere-Ionosphere Coupling*. Springer Berlin Heidelberg, 1993.
- [32] C. F. Kennel, L. J. Lanzerotti, and E. N. Parker. *Solar system plasma physics. Volume 2 - Magnetospheres*. 1979.
- [33] M. G. M. G. Kivelson and C. T. C. T. Russell. *Introduction to space physics*. Cambridge University Press, 1995.
- [34] K.Oksavik, J.I.Moen, E.H.Rekaa, H.C.Carlson, and M.Lester. Reversed flow events in the cusp ionosphere detected by S^{uper}D^{ata}R^{ange}N^{et} H^{igh}F^{requency} radars. *Journal of Geophysical Research*, VOL. 116, 2011.
- [35] H. Korth, B. J. Anderson, and C. L. Waters. Statistical analysis of the dependence of large-scale Birkeland currents on solar wind parameters. *Ann. Geophys*, 28:515–530, 2010.
- [36] K. M. Laundal and A. D. Richmond. Magnetic Coordinate Systems.
- [37] B. Lavraud. High-altitude cusp flow dependence on IMF orientation: A 3-year Cluster statistical study. *Journal of Geophysical Research*, 110(A2):A02209, 2005.
- [38] G. Le, J. A. Slavin, and R. J. Strangeway. Space Technology 5 observations of the imbalance of regions 1 and 2 field-aligned currents and its implication to the cross-polar cap Pedersen currents. *Journal of Geophysical Research: Space Physics*, 115(A7), jul 2010.
- [39] H.-S. of-the-polar-cap-at-the-upper-atmosphere-and-ionosphere-of-Earth-which-is_fig1.241060964. a) Schematic of the polar cap at the upper atmosphere and ionosphere of...
- [40] D. Oliveira. Ionosphere-magnetosphere coupling and field-aligned currents. *Revista Brasileira de Ensino de Física*, 36(1):1–5, mar 2014.

- [41] G. Parks and S. (Washington, University. *Physics of space plasmas - An introduction*. Redwood City, CA; Addison-Wesley Publishing Co., jan 1991.
- [42] H. particle drifts.svg. File:Charged-particle-drifts.svg - Wikipedia.
- [43] H. particle drifts.svg. File:Charged-particle-drifts.svg - Wikipedia.
- [44] G. W. Pross. *Physics of the Earth's space environment : an introduction*. Springer, 2004.
- [45] H. science-what-is-a-mole-of air/. Ionosphere ion density.
- [46] G. J. Sofko, R. Greenwald, and W. Bristow. Direct determination of largescale magnetospheric fieldaligned currents with SuperDARN. *Geophysical Research Letters*, 22(15):2041–2044, 1995.
- [47] H. svalbard radar/. EISCAT Svalbard Radar EISCAT.
- [48] Webpage. ACE satellite. <https://science.nasa.gov/missions/ace>.
- [49] Webpage. ACE satellite. <https://docs.scipy.org/doc/scipy/reference/tutorial/io.html>.
- [50] Webpage. Basemap tutorial. <https://basemaptutorial.readthedocs.io/en/latest/index.html>.
- [51] Webpage. Birkeland currents. https://en.wikipedia.org/wiki/Birkeland_current.
- [52] Webpage. Calculate bearing. <http://www.movable-type.co.uk/scripts/latlong.html>.
- [53] Webpage. Data from ACE. https://omniweb.gsfc.nasa.gov/form/omni_min.html.
- [54] Webpage. Magnetosphere. https://www.researchgate.net/figure/Schematic-figure-of-the-magnetosphere-showing-the-main-plasma-regions-and-current-system-fig2_233742440.
- [55] Webpage. Natural Frequency - the Sun. <http://naturalfrequency.com/wiki/sun?mobile=true>.
- [56] Webpage. Wikimedia Commons. <https://www.quora.com/What-are-some-insane-facts-about-the-Sun>.
- [57] Webpage. Wikimedia Commons. <http://velamentous.blogspot.no/2010/11/land-of-aurora-borealis.html>.

- [58] Webpage. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Sunspot_butterfly_graph.gif.
- [59] Webpage. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Sun_poster.svg.
- [60] Webpage. Wikimedia Commons. <http://gpsm.space-science.ro/esteem/>.
- [61] Webpage. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Schematic_of_combined_Field-Aligned_Currents_and_ionospheric_current_system.svg.
- [62] D. R. Weimer. Models of highlatitude electric potentials derived with a least error fit of spherical harmonic coefficients. *Journal of Geophysical Research: Space Physics (1978-2012)*, 100(A10):19595–19607, 1995.
- [63] Y.Rinne, J.Moen, H.C.Carlson, and M.R.Hairston. Stratification of east-west plasma flow channels observed in the ionospheric cusp in response to IMF By polarity changes. *Geophysical Research Letters*, VOL. 37, 2010.
- [64] Y.Rinne, J.Moen, K.Oksavik, and H.C.Carlson. Reversed flow events in the winter cusp ionosphere observed by the European Incoherent Scatter (EISCAT) Svalbard radar. *Journal of Geophysical Research*, VOL. 112, 2007.
- [65] E. G. Zweibel and M. Yamada. Perspectives on magnetic reconnection. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 472(2196):20160479, dec 2016.

Appendix A

Python programs

plot_IMF.py

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime
import scipy.io
import os
import glob
'''
Version 16.05.17
'''

'''
Function for reading a .txt file containing IMF data from a path.
The .txt file has 8 columns with values, starting at line 11 and ordered the following way:
Year = 1st, Day = 2nd, Hour = 3rd, Minute = 4th, Bx = 5th, By = 6th, Bz = 7th,
    Speed = 8th
Setting starttime and endtime for plot on form [hour, minute], date on form [month, day]
'''

path = '/Users/klausno/Documents/UiO/master/eiscat_data/IMF_data/'
starttime = [9, 00]
endtime = [10, 30]
date = [30, 11]
ival = 15

def IMF_data(path):
    filenames = []
    hours = []
    minutes = []
    B_xs = []
    B_ys = []
    B_zs = []
    speeds = []

    for filename in os.listdir(path):
        filenames.append(filename)
```

```

if filename.endswith('.txt'):
    infile = open(os.path.join(path, filename), 'r')
    lines = infile.readlines()
    infile.close()
    y = []
    d = []
    hour = []
    minute = []
    B_x = []
    B_y = []
    B_z = []
    speed = []

    for line in lines[10:]:
        values = np.array(line.split())
        y.append(float(values[0]))
        d.append(float(values[1]))
        hours.append(float(values[2]))
        minute.append(float(values[3]))
        B_xs.append(float(values[4]))
        B_ys.append(float(values[5]))
        B_zs.append(float(values[6]))
        speeds.append(float(values[7]))

    year = int(y[0])
    #d = day nb 334... :-P, not on form month, day

return np.array(filenamees), year, np.array(hours), np.array(minutes), np.array(B_xs),
        np.array(B_ys), np.array(B_zs), np.array(speeds)

def IMF_plot(start, end):
    """
    Removing any data in B_x, B_y, B_z and speed that is nonsense, setting them to
    NaN to not plot them.
    time_step dt = 1 minute
    """
    filenamees, year, hours, minutes, B_x, B_y, B_z, speed = IMF_data(path)
    n = len(filenamees)

    for i in range(n):
        t = datetime.datetime(year, date[1], date[0], 0, 0)
        dt = datetime.timedelta(minutes=1)
        times = []

        B_x[B_x==9999.99] = np.nan
        B_y[B_y==9999.99] = np.nan
        B_z[B_z==9999.99] = np.nan
        speed[speed==99999.9] = np.nan

```



```

for j in range(len(speed)):
    times.append(t)
    t += dt

starttime = datetime.datetime(year, date[1], date[0], start[0], start[1])
endtime = datetime.datetime(year, date[1], date[0], end[0], end[1])

for k in range(len(times)):
    if times[k] == starttime:
        a = k
        continue
    if times[k] == endtime:
        b = k+1
        break

times = times[a:b]
B_x = B_x[a:b]
B_y = B_y[a:b]
B_z = B_z[a:b]
speed = speed[a:b]

#fig = plt.figure(figsize=(12, 6))
fig = plt.figure()
c = 1.5
d = 1.0
fig.suptitle('Values of IMF in [nT], measured by ACE \n %s' %
             (starttime.strftime("%d-%m-%Y")))

ax2 = plt.subplot(211)
plt.plot(times, B_y, linewidth=c)
plt.yticks(np.arange(-10.0, 11.0, 2))
plt.ylim(-10.0, 10.0)
#plt.plot(times, B_y, linewidth=c)
plt.xlabel('Time')
#plt.ylabel('B_y [nT]', rotation=0, labelpad=25)
plt.ylabel('B_y [nT]')
plt.grid(True)
ax2.xaxis.set_major_locator(mdates.MinuteLocator(interval=ival))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))

ax3 = plt.subplot(212)
#ax3.plot(times, B_z, linewidth=c, sharey=ax2)
plt.plot(times, B_z, linewidth=c)
#plt.xticks(np.arange(-10.0, 11.0, 2))
plt.yticks(np.arange(-10.0, 11.0, 2))
plt.ylim(-10.0, 10.0)
plt.xlabel('Time')
plt.ylabel('B_z [nT]')
#plt.ylabel('B_z [nT]', rotation=0, labelpad=25)

```

```

plt.grid(True)
#plt.gca().xaxis.set_major_locator(mdates.HourLocator())
ax3.xaxis.set_major_locator(mdates.MinuteLocator(interval=ival))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))

ax2.axhline(y=0, c='red',linewidth=d)
ax3.axhline(y=0, c='red',linewidth=d)

'''
fig2 = plt.figure(1)
fig.suptitle('Solar wind speed')
plt.plot(times, speed, linewidth=d)
plt.xlabel('Time')
plt.ylabel('km/s')
plt.grid(True)
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
plt.show()
'''

interval = str(starttime.strftime("%d_%m_%Y_%H_%M-")) +
            str(endtime.strftime("%H_%M"))
plt.savefig('IMF_plots/IMF_%s.pdf'.format([i] % interval)
            #plt.show()

return

IMF_plot(starttime, endtime)

```

main_program.py

```

import scipy.io
from numpy import *

#path = '/Users/klausno/Documents/UiO/0_programming_master/120_matlabfiler'
#path =
    '/Users/klausno/Documents/UiO/0_programming_master/310_matlabfiler_30_11_14'
path =
    '/Users/klausno/Documents/UiO/0_programming_master/2453_matlabfiler_30_11_14'
#path =
    '/Users/klausno/Documents/UiO/0_programming_master/310_matlabfiler_12_01_14'
#path =
    '/Users/klausno/Documents/UiO/0_programming_master/1837_matlabfiler_12_01_14'

'''
Step 1, read the matlab files and make parameters.
'''

from read_matlab import *
filenames, path = read_matlab(path)
azimuth_angle, elevation, distance, parameters, timeinterval = store_data(filenames,
    path)

'''

```

```

Step 2, find times and date
'''
from azimuth import *
t, dt, date = azimuth_time(azimuth_angle, timeinterval)

'''
#Step 3, make grid, plot data on grid and map.
'''

from make_regular_grid import *
make_regular_grid(distance, parameters, t, dt, date, azimuth_angle)

```

read_matlab.py

```

import scipy.io
import os
import glob
import matplotlib.pyplot as plt
from pylab import *
import datetime

'''
Read a matlab file .mat into python
Read all files in a given folder
and store it as a list of strings, called filenames
'''

def read_matlab(path):
    filenames = []

    for filename in os.listdir(path):
        if filename.endswith('.mat'):
            filenames.append(filename)

    return sorted(filenames), path

'''
Function for storing the data from all the filenames as arrays.
It returns arrays of the variables.
'''

def store_data(filenames, path):
    azimuth = []
    elevation = []
    distance = []
    parameters = []
    timeinterval = []

    '''
    Looping over every file in the given path, assuming they are all .mat files.
    Should change this to pick out ONLY .mat files?
    '''

    for file in filenames:

```

```

'''
Adding a string to a string can be done this way:
a = path + '/filename' results in path/filename

Create the new path so it can be read and then make the parameters from the
read file.
x is a dictionary with the parameters as keys: x = {key: values, key: values...}
'''

newpath = path + '/' + file
data = scipy.io.loadmat(newpath)

azimuth.append(float(data['r_az']))
elevation.append(float(data['r_el']))
distance.append(data['r_range'])
parameters.append(data['r_param'])
timeinterval.append(data['r_time'])

return array(azimuth), array(elevation), array(distance), array(parameters),
        array(timeinterval)

```

azimuth.py

```

import scipy.io
#from scitools.std import *
import os
import glob
import matplotlib.pyplot as plt
from pylab import *
import datetime
import numpy as np

'''
A function to find the swipes of the radar from one end
to the other, where the azimuth angle changes from falling to growing.
'''

def azimuth_time(angle, timeinterval):
    n = len(angle)
    year = []
    month = []
    day = []
    hour = []
    minute = []
    second = []

    for time in timeinterval:
        year.append(time[0][0])
        month.append(time[0][1])
        day.append(time[0][2])
        hour.append(time[0][3])
        minute.append(time[0][4])
        second.append(time[0][5])

```

```

'''
Reading the start time called time, the timestep called dt and the date.
'''
time = datetime.datetime(int(year[0]), int(month[0]), int(day[0]), int(hour[0]),
                          int(minute[0]), int(second[0]))
dt = datetime.timedelta(seconds = (second[1] - second[0]))
times = []
date = datetime.date(int(year[0]), int(month[0]), int(day[0]))

for j in range(n):
    times.append(time)
    time += dt

#print times[3]

return np.array(times), dt, date

```

make_regular_grid.py

```

import scipy.io
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from pylab import *
from scipy.interpolate import griddata
from scipy import interpolate
import numpy as np
from matplotlib.pyplot import figure, show
from matplotlib.image import NonUniformImage
from matplotlib import cm
from mpl_toolkits.basemap import Basemap

from my_polynomial import *
from regularise_array import *
from plot_radar_data import *
from current_density import *
from new_beams_n_azimuths import *
from lat_n_lon import *
from my_interpolate import *
from median_filter import *

radar_lat = 78.22
radar_lon = 15.65

def make_regular_grid(distance, parameters, t, dt, date, azimuth_angle):
    '''
    Remember to change the sign of velocity v_i so it matches the (raw data) convention
    '''
    n_e = []
    T_i = []
    T_e = []

```

```

v_i = []
distance_list = []

for value in distance:
    distance_list.append((value))

for lists in parameters:
    for value in lists:
        n_e.append(value[0])
        a = value[1]
        T_i.append(a)
        T_e.append(value[2]*a)
        v_i.append(value[4]*(-1.0))

beams = []
n_list = []
for beam in distance_list:
    radar_range = []
    for value in beam:
        radar_range.append(float(value))

    radar_range = np.array(radar_range)
    beams.append(radar_range)
    n_list.append(len(radar_range))

'''
Sort every variable as a nested list with
the same number of elements in all the nested lists
'''

v_list = []
e_density_list = []
ion_temp_list = []
electron_temp_list = []
#timelist = []
for n in n_list:
    v_list.append(np.array(v_i[:n]))
    e_density_list.append(n_e[:n])
    ion_temp_list.append(np.array(T_i[:n]))
    electron_temp_list.append(T_e[:n])
    #timelist.append(t[:n])
    #del t[:n]
    del v_i[:n]
    del n_e[:n]
    del T_i[:n]
    del T_e[:n]

v_list = np.array(v_list)
beams = np.array(beams)
electron_density = np.array(e_density_list)

```

```

ion_temp = np.array(ion_temp_list)
p_list = []

'''
Find the minimum angle and maximum angle to locate start and stop
of each swipe.
'''
for p in range(len(azimuth_angle)):
    if p == 0:
        continue
    if p == (len(azimuth_angle) - 1):
        p_list.append(p)
        continue
    if azimuth_angle[p] < azimuth_angle[p-1] and azimuth_angle[p] <
        azimuth_angle[p+1]:
        p_list.append(p)
    if azimuth_angle[p] > azimuth_angle[p-1] and azimuth_angle[p] >
        azimuth_angle[p+1]:
        p_list.append(p)

'''
Loop over each swipe
Empty lists to store and save the data in a file.
q and r are counters
'''

Vlist = []
nelist = []
Jlist = []
swipelist = []
angelist = []
q = 1
r = 0
for p in p_list:
    radar_swipe = np.array(beams[r:p])
    radar_angles = np.array(azimuth_angle[r:p])
    ion_velocity = np.array(v_list[r:p])
    n_e = np.array(electron_density[r:p])
    t_i = np.array(ion_temp[r:p])
    starttime = t[r]
    stoptime = t[p]
    '''
    If the ion velocity is larger than 2500 m/s the value is set to NaN,
    because it is probably a wrong or not existing radar measurement.
    '''
    for z in ion_velocity:
        for k in range(len(z)):
            if abs(z[k]) > 2500:
                z[k] = nan

    coeffs = []
    '''

```

```

Loop for finding grid beam by polyfitting the distance dx as a function of x
for each beam, then take the average of the coefficients.
Must ignore very big steps in dx, dx > 100 km, or else the fit becomes very bad.
Need a start and stop range guess for my grid beam.
Make this into a separate function/script
'''

start = 1000
stop = 0
for beam in radar_swipe:
    dx_length = []
    n = len(beam)
    for i in range(n-1):
        dx = beam[i+1] - beam[i]
        if dx > 100:
            dx = nan

        dx_length.append(dx)

    c = my_polynomial(beam[1:], dx_length)
    coeffs.append(c)

    for value in beam:
        if value < start:
            start = value
        if value > stop:
            stop = value

m = len(coeffs)
a, b, c, d = 0, 0, 0, 0
for i in range(len(coeffs)):
    a += coeffs[i][3]
    b += coeffs[i][2]
    c += coeffs[i][1]
    d += coeffs[i][0]

a, b, c, d = a/m, b/m, c/m, d/m
'''

Setting N to some value just as a counter
'''

x = []
x.append(start)
N = 200

dx_list = []
for i in range(N-1):
    dx = a + b*x[i] + c*x[i]**2 + d*x[i]**3
    dx_list.append(dx)

    #x[i+1] = x[i] + dx
    xnew = x[i] + dx
    x.append(xnew)

```



```

        if xnew > stop:
            break

    """
    Make my gridded beam, gridded swipe and gridded v_data.
    Masking NaNs so they plot as empty polygons.
    """

    gridded_beam = np.array(x[:])
    #gridded_beam = np.array(x)
    gridded_swipe = np.array([gridded_beam for i in range(p-r)])

    v_data = regularise_array(radar_swipe, gridded_beam, ion_velocity)
    n_e_data = regularise_array(radar_swipe, gridded_beam, n_e)
    t_i_data = regularise_array(radar_swipe, gridded_beam, t_i)

    """
    When calculating current_density:
    Need to check the number of beams in each swipe.
    If there are only two beams I cannot calculate the current density,
    so I must skip that swipe and go to the next.
    q += 1 is updating the swipe number
    r = p is updating the time.

    About else:
    must swap the order of the arrays as the radar moves
    back and forth, not forth and forth.
    """

    if len(radar_angles[1:-1]) == 0:
        q += 1
        r = p
        continue

    else:
        if radar_angles[0] < radar_angles[-1]:
            radar_angles[:] = radar_angles[::-1]
            v_data[:] = v_data[::-1]
            n_e_data[:] = n_e_data[::-1]
            t_i_data[:] = t_i_data[::-1]

        else:
            radar_angles[:] = radar_angles
            v_data[:] = v_data
            n_e_data[:] = n_e_data
            t_i_data[:] = t_i_data

    starttime = t[r]
    stoptime = t[p]

```

```

v_ion = interpolate_myway(v_data)
T_ion = interpolate_myway(t_i_data)
ne = interpolate_myway(n_e_data)
#v_masked_unfiltered = np.ma.masked_equal(v_data, 0.0, copy=False)
#v_masked_interpol = np.ma.masked_equal(v_ion, 0.0, copy=False)
v_filtered = median_filter(v_ion)
v_masked_data = np.ma.masked_equal(v_filtered, 0.0, copy=False)
J_data = current_density(gridded_swipe, radar_angles, v_filtered)
#J_filtered_v = J_data
#J_masked_data = np.ma.masked_equal(J_filtered_v, 0.0, copy=False)
#J_data = current_density(gridded_swipe, radar_angles, v_ion)
#J_filtered = median_filter(J_data)
#J_masked_data = np.ma.masked_equal(J_filtered, 0.0, copy=False)
#n_e_data = regularise_array(radar_swipe, gridded_beam, n_e)

n_e_filtered = median_filter(ne)
n_e_masked_data = np.ma.masked_equal(n_e_filtered, 0.0, copy=False)
Ti_filtered = median_filter(T_ion)
Ti_masked_data = np.ma.masked_equal(Ti_filtered, 0.0, copy=False)

'''
Starting to plot
Make the colorbar smaller. Like shrink=0.7
Figsize in inches on form figsize(width, height).
For single plots use figsize(10, 6), dual and triple figsize(12, 6)
'''

'''
Experiment with pcolormesh for faster plotting
'''

#my_pcolormesh(gridded_swipe, radar_angles, J_filtered_v)

subtitles = ['Ion velocity', 'Current density', 'Electron density', 'Ion
             temperature']
fs1 = 14.0
fs = 15.0
fs3 = 13.5
sz1 = 0.9
sz2 = 0.7
sz3 = 0.45

'''
For single plots:
'''
fig = plt.figure(figsize=(12, 8))
fig.suptitle('Swipe no %g \n %s-%s' % (q, starttime.strftime('%d.%m.%Y
             %H:%M:%S'), stoptime.strftime('%H:%M:%S')),
             fontsize=15, fontweight='bold' )

ax = fig.add_subplot(111)

```

```

#ax.set_title( subtitles[0] + ' in ' + r'$ \left [ \frac{m}{s} \right ]$',
              fontsize=fs, position=(0.5, 1.033))
fig.subplots_adjust(top=0.88)
plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles, v_masked_data, -
               1200, 1200, r'$\frac{m}{s}$', fig, ax, sz=sz1)
#plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles, v_filtered, -1200,
               1200, r'$\frac{m}{s}$', fig, ax, sz=sz1)
plt.savefig('plots/301114/v_filtered/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/301114/v_filtered_masked/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/301114/v_interpol_myway/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/011214/v_interpol_myway/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/011214/v_filtered/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/301114/v_raw_data/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/011214/v_raw_data/swipe_no_{}.pdf'.format(q))
plt.close('all')

'''

#current density
ax = fig.add_subplot(111)
#ax.set_title( subtitles[1] + ' in ' + r'$10^{-6} \left [ \frac{A}{m^2} \right ]$',
              fontsize=fs1, position=(0.5, 1.033))
#ax.set_title( subtitles[1], fontsize=fs1, position=(0.5, 1.033))
plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles[1:-1],
               J_masked_data*1E6, -1.1, 1.1, r'$10^{-6} \frac{A}{m^2}$', fig,
               ax, sz=sz1)
#plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles[1:-1], J_data*1E6, -
               1.1, 1.1, r'$10^{-6} \frac{A}{m^2}$', fig, ax, sz=sz1)
#fig.subplots_adjust(top=0.88)
#plt.savefig('plots/301114/j_filtered/swipe_no_{}.pdf'.format(q))
plt.savefig('plots/301114/j_filtered_v/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/011214/j_value_myway/swipe_no_{}.pdf'.format(q))
#plt.close('all')

'''

'''

ax = fig.add_subplot(111)
ax.set_title( subtitles[2] + ' in ' + r'$\left [ \log_{10} \frac{1}{m^3} \right ]$',
              fontsize=fs, position=(0.5, 1.033))
#plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles,
               log10(n_e_masked_data), 9, 13, r'$\log_{10} \frac{1}{m^3}$', fig,
               ax, sz=sz1)
plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles[1:-1],
               log10(n_e_filtered), 9, 13, r'$\log_{10} \frac{1}{m^3}$', fig, ax,
               sz=sz1)
fig.subplots_adjust(top=0.83)
plt.savefig('plots/301114/n_e_filtered/swipe_no_{}.pdf'.format(q))
#plt.savefig('plots/011214/ne_value/swipe_no_{}.pdf'.format(q))
plt.close('all')

'''

'''

ax = fig.add_subplot(111)

```

```

ax.set_title( subtitles[3] + ' in ' + r'$\left [ K \ \right ]$', fontsize=fs3,
              position=(0.5, 1.033))
#plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles, Ti_masked_data, 0,
                 5000, 'K', fig, ax, sz=sz1)
#plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles[1:-1],
                 Ti_masked_data, 0, 5000, 'K', fig, ax, sz=sz3)
plot_radar_data(gridded_swipe[1:-1, :-2], radar_angles[1:-1], Ti_filtered, 0,
                 2550, 'K', fig, ax, sz=sz1)
fig.subplots_adjust(top=0.83)
plt.savefig('plots/301114/Ti_filtered/swipe_no_{0}.pdf'.format(q))
plt.close('all')
'''

r = p
q += 1

return

```

bearing.py

```

from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib.tri as tri
from pylab import *
from scipy.interpolate import griddata
import math
import matplotlib as mpl

'''
Function for calculating the bearing of the initial point, given latitude
and longitude of start point and end point.
'''

def bearing(end_lon, start_lon, end_lat, start_lat):

    theta_bearing = ( arctan2(sin(end_lon - start_lon)*cos(end_lat),
                             cos(start_lat)*sin(end_lat) -
                             sin(start_lat)*cos(end_lat)*cos(end_lon - start_lon)) )

    return theta_bearing

```

corners.py

```

import numpy as np

'''
Sub function for making the corners of a polygon
'''

def corners(lon1, lat1, lon2, lat2):

```

```

vertices = []

for i, j in zip(range(len(lon1)-1), range(len(lon2)-1)):
    p1 = [lon1[i], lat1[j]]
    p2 = [lon1[i+1], lat1[j+1]]
    p3 = [lon2[i+1], lat2[j+1]]
    p4 = [lon2[i], lat2[j]]

    p = np.array([p1, p2, p3, p4, p1])
    vertices.append(p)

vertices = np.array(vertices)
return vertices

```

current_density.py

```

from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib.tri as tri
from pylab import *
from scipy.interpolate import griddata
import math
import matplotlib as mpl

from lat_n_lon import *
from bearing import *
from length_of_dl import *
#from interpolate_velocity import *

'''
Function for calculating the current density
gridded_swipe is a nested array with all the beams of one swipe
radar_angles is an array with all angles of one swipe
v is a nested array, the gridded velocity data
radar_lat is the latitude of the radar
radar_lon is the longitude of the radar
lyr_lat = 78.22
lyr_lon = 15.65
'''

def current_density(gridded_swipe, radar_angles, v, radar_lat=78.22, radar_lon=15.65):
    convert = pi/180.
    B = -60E-6
    lats, lons = lat_n_lon(gridded_swipe, radar_angles, radar_lat, radar_lon)
    lats = lats*convert
    lons = lons*convert

    '''
    Ignore the first beam b and first gate g since I cannot calculate the curl there

```

```

End the loop before the last beam b and gate g since I cannot calculate the curl
there.
J is a list to store the calculated current densities
'''
J = []
for b in range(len(lats)-1):
    if b == 0:
        continue

    j_list = []
    for g in range(len(lats[b])-1):
        if g == 0:
            continue

        #Find bearings
        theta_p_1 = bearing(lons[b-1][g+1], lons[b-1][g], lats[b-1][g+1],
                           lats[b-1][g])
        theta_p_2 = bearing(lons[b][g], lons[b][g+1], lats[b][g], lats[b][g+1])
        theta_p_3 = bearing(lons[b+1][g-1], lons[b+1][g], lats[b+1][g-1],
                           lats[b+1][g])
        theta_p_4 = bearing(lons[b][g], lons[b][g-1], lats[b][g], lats[b][g-1])

        theta_dl_1 = bearing(lons[b][g+1], lons[b-1][g], lats[b][g+1], lats[b-1][g])
        theta_dl_2 = bearing(lons[b+1][g], lons[b][g+1], lats[b+1][g], lats[b][g+1])
        theta_dl_3 = bearing(lons[b][g-1], lons[b+1][g], lats[b][g-1], lats[b+1][g])
        theta_dl_4 = bearing(lons[b-1][g], lons[b][g-1], lats[b-1][g], lats[b][g-1])
        theta_dl_area = bearing(lons[b][g-1], lons[b-1][g], lats[b][g-1], lats[b-1][g])
        theta_dl_area_2 = bearing(lons[b][g+1], lons[b+1][g], lats[b][g+1],
                                  lats[b+1][g])

        #Find the angle alpha
        alpha_1 = theta_p_1 - theta_dl_1
        alpha_2 = theta_p_2 - theta_dl_2
        alpha_3 = theta_p_3 - theta_dl_3
        alpha_4 = theta_p_4 - theta_dl_4

        #Find dl's
        dl_1 = length_of_dl(lons[b][g+1], lons[b-1][g], lats[b][g+1], lats[b-1][g])
        dl_2 = length_of_dl(lons[b+1][g], lons[b][g+1], lats[b+1][g], lats[b][g+1])
        dl_3 = length_of_dl(lons[b][g-1], lons[b+1][g], lats[b][g-1], lats[b+1][g])
        dl_4 = length_of_dl(lons[b-1][g], lons[b][g-1], lats[b-1][g], lats[b][g-1])

        alpha_dl_1_dl_4 = theta_dl_1 - theta_dl_area
        alpha_dl_2_dl_3 = theta_dl_3 - theta_dl_area_2

        area1 = abs(dl_1 * dl_4 * sin(alpha_dl_1_dl_4))/2.0
        area2 = abs(dl_2 * dl_3 * sin(alpha_dl_2_dl_3))/2.0
        total_area = area1 + area2

```

```

'''
Find the current density
'''
j1 = (v[b-1][g]) * dl_1 * cos(alpha_1)
j2 = (v[b][g+1]) * dl_2 * cos(pi -alpha_2)
j3 = (v[b+1][g]) * dl_3 * cos(pi + alpha_3)
j4 = (v[b][g-1]) * dl_4 * cos(2*pi -alpha_4)
'''

Find the curl around the gate by summing up the line parts of the integral
'''

j_tot = (j1 + j2 + j3 + j4)*(B/total_area)
j_list.append(j_tot)

j_list = np.array(j_list)
J.append(j_list)

return np.array(J)

```

lat_n_lon.py

```

import numpy as np
from pylab import *

'''
Function for calculating the latitude and longitude of the gates in radar beam.
Function takes:
radar position; latitude and longitude in radians
all the beams of one swipe,
the azimuth angle of those beams, in radians
then calculated the latitude and longitude of the gates in the beams
Formula:
lat = arcsin(sin(radar_lat)*cos(delta) + cos(radar_lat)*sin(delta)*cos(theta))
lon = radar_lon + arctan2(sin(theta)*sin(delta)*cos(radar_lat), cos(delta) -
    sin(radar_lat)*sin(lat))
with D = radar_swipe[j][i], delta = D/RE
'''

def lat_n_lon(radar_swipe, radar_angles, radar_lat=78.22, radar_lon=15.65):
    convert = pi/180.
    RE = 6.371E3
    lats = []
    lons = []
    radar_lat0 = radar_lat*convert
    radar_lon0 = radar_lon*convert

    for j in range(len(radar_swipe)):
        lat_list = []
        lon_list = []

        for i in range(len(radar_swipe[j])):

```

```

lat = arcsin(sin(radar_lat0)*cos(radar_swipe[j][i]/RE) +
             cos(radar_lat0)*sin(radar_swipe[j][i]/RE)*cos(radar_angles[j]*convert))
lon = radar_lon0 +
      arctan2(sin(radar_angles[j]*convert)*sin(radar_swipe[j][i]/RE)*cos(radar_lat0),
             cos(radar_swipe[j][i]/RE) - sin(radar_lat0)*sin(lat))

lat_list.append(lat/convert)
lon_list.append(lon/convert)

lats.append(np.array(lat_list))
lons.append(np.array(lon_list))

lats = np.array(lats)
lons = np.array(lons)

return lats, lons

```

length_of_dl.py

```

from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib.tri as tri
from pylab import *
from scipy.interpolate import griddata
#import math
import matplotlib as mpl

'''
Function for calculating the distance between two points on the surface
of the Earth, given latitude and longitude of starting point and end point.
RE is the radius of the Earth
'''
def length_of_dl(end_lon, start_lon, end_lat, start_lat):
    RE = 6.371E6

    kl = np.sin((end_lat - start_lat)/2.) * np.sin((end_lat - start_lat)/2.)
    la = np.cos(start_lat)*np.cos(end_lat)*np.sin((end_lon -
        start_lon)/2.)*np.sin((end_lon - start_lon)/2.)

    '''
    a = np.sin((end_lat - start_lat)/2.) * np.sin((end_lat - start_lat)/2.) +
        np.cos(start_lat)*np.cos(end_lat)*np.sin((end_lon -
        start_lon)/2.)*np.sin((end_lon - start_lon)/2.)
    '''

    a = kl + la

    c = 2.0*np.arctan2(sqrt(a), sqrt(1.0 - a))

```



```
#print kl, la
dl = RE*c
return dl
```

median_filter.py

```
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
from scipy import signal
import scipy.ndimage

'''
Function to do the median filtering to smooth out salt and pepper
noise.
'''

def median_filter(z):
    z_filtered = sp.ndimage.median_filter(z, 3, mode='nearest')
    #z_filtered = sp.ndimage.median_filter(z, 3, mode='reflect')

    return z_filtered
```

my_interpolate.py

```
import numpy as np
from pylab import *
from scipy.interpolate import griddata
from scipy import interpolate

'''
Version 16.05.2017.
'''

def interpolate_myway(data, gridded_beam=None, radar_angles=None, threshold=10,
    weight=1.8):
    new_data = data.copy()
    for i in range(len(data)):
        for j in range(len(data[i])):
            '''
            If v[i][j] is approx zero, do interpolation with its neighbours and find v[i][j]
            Else, v[i][j] is not zero, smooth v[i][j] with its neighbours
            '''
            if abs(data[i][j]) < threshold:
                sum_n = 0
                no_n = 0
                #for n_i in [i-1, i+1]:
                for n_i in [i-2, i-1, i+1, i+2]:
                    if n_i > -1 and n_i < len(data):
                        if not abs(data[n_i][j]) < 1E-6:
                            sum_n += data[n_i][j]
```

```

        no_n += 1
    for n_j in [j-2, j-1, j+1, j+2]:
        if n_j > -1 and n_j < len(data[i]):
            if not abs(data[i][n_j]) < 1E-6:
                sum_n += data[i][n_j]
                no_n += 1
    if no_n != 0:
        new_data[i][j] = sum_n/float(no_n)

    else:
        no_n = weight
        sum_n = weight*data[i][j]
        for n_i in [i-2, i-1, i+1, i+2]:
            if n_i > -1 and n_i < len(data):
                sum_n += data[n_i][j]
                no_n += 1
        for n_j in [j-2, j-1, j+1, j+2]:
            if n_j > -1 and n_j < len(data[i]):
                sum_n += data[i][n_j]
                no_n += 1
        if no_n != 0:
            new_data[i][j] = sum_n/float(no_n)

'''
Might want to return the weight to use it in plot text?
'''

return new_data

def interpolate_griddata(data, gridded_beam, radar_angles, threshold=10):
    new_data = data[abs(data - 0.0) > threshold].copy()
    points = []

    for T in range(len(radar_angles)):
        for r in range(len(gridded_beam)):
            if abs(data[T][r] - 0.0) > threshold:
                point = np.array((gridded_beam[r], radar_angles[T]))
                points.append(point)

    points = np.array(points)
    values = list(np.array(new_data).flat)

    X, Y = np.meshgrid(gridded_beam, radar_angles)
    new_data = interpolate.griddata(points, values, (X, Y), fill_value=0,
        method='cubic')
    return new_data

def interpolate_linear(data, gridded_beam, radar_angles, threshold=10):

```

```

new_data = data[abs(data -0.0) > threshold ].copy()
points = []

for T in range(len(radar_angles)):
    for r in range(len(grided_beam)):
        if abs(data[T][r] -0.0) > threshold:
            point = np.array((grided_beam[r], radar_angles[T]))
            points.append(point)

points = np.array(points)
values = list(np.array(new_data).flat)

f = interpolate.LinearNDInterpolator(points, values, fill_value=0)
new_data = f(X, Y)
return new_data

def interpolate_clough(data, grided_beam, radar_angles, threshold=10):
    new_data = data[abs(data -0.0) > threshold ].copy()
    points = []

    for T in range(len(radar_angles)):
        for r in range(len(grided_beam)):
            if abs(data[T][r] -0.0) > threshold:
                point = np.array((grided_beam[r], radar_angles[T]))
                points.append(point)

    points = np.array(points)
    values = list(np.array(new_data).flat)

    f3 = scipy.interpolate.CloughTocher2DInterpolator(points, values, fill_value=0,
        tol=1e-6)
    new_data = f(X, Y)
    return new_data

```

my_polynomial.py

```

import matplotlib.pyplot as plt
from numpy import *
from numpy.polynomial import Polynomial

'''
Function for interpolation.
Interpolate is the distance dx between points in the beam,
think of dx as a function of x and beam[i]; f(x, beam[i]) = dx.
Returns the coefficients of the interpolated polynom on form:
a + bx + cx**2 + dx*3, changed to polyfit(x, y, 3)
Polyfit returns an array.
Setting NaN-values to zero.
'''

```

```
def my_polynomial(x, dx):
    x = x
    f = dx
    p = polyfit(x, f, 3)
    find_nans = isnan(p)
    p[find_nans] = 0.0

    return p
```

new_beams_n_azimuths.py

```
from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib.tri as tri
from pylab import *
from scipy.interpolate import griddata
import math
import matplotlib as mpl

'''
Function for making additional beams and angles,
centered around the actual beams of the radar.
Need these extra beams so I can draw a polygon around the actual gate in the
beams.
radar_swipe is radar_swipe
angle is radar_angles
'''

def new_beams_n_azimuths(radar_swipe, radar_angles):
    '''
    radar_swipe is a nested array with all the beams of one swipe.
    The change in angle, d_angle, values are in degrees.
    The change in distance, dx, between gates in one radar_beam, values are in [km].
    '''
    d_angle = [radar_angles[i] - radar_angles[i+1] for i in range(len(radar_angles)-1)]
    d_angle.append(d_angle[-1])
    d_angle = np.array(d_angle)/2.
    dx = (radar_swipe[:,1:] - radar_swipe[:, :-1])/2.

    new_radar_swipe = []
    for i, j in zip(range(len(radar_swipe)), range(len(dx))):
        radar_swipe_list = []

        for k, m in zip(range(len(radar_swipe[i])), range(len(dx[j]))):
            #adding an extra point in the beginning of the beam
            if k == 0:
                radar_swipe_list.append(radar_swipe[i][k] - dx[j][m])
                radar_swipe_list.append(radar_swipe[i][k+1] - dx[j][m])
```

```

        else:
            radar_swipe_list.append(radar_swipe[i][k+1] -dx[j][m])

            radar_swipe_list.append(radar_swipe[i][k+1] + dx[j][m])
            new_radar_swipe.append(np.array(radar_swipe_list))

#adding an extra beam on the end
new_radar_swipe.append(new_radar_swipe[-1])
new_beams = np.array(new_radar_swipe)

new_angle = []
for k, m in zip(range(len(radar_angles)-1), range(len(d_angle))):
    if k == 0:
        new_angle.append(radar_angles[k] + d_angle[m])
        new_angle.append(radar_angles[k+1] + d_angle[m])

    else:
        new_angle.append(radar_angles[k+1] + d_angle[m])

new_angle.append(radar_angles[k+1] -d_angle[m])
new_angles = np.array(new_angle)

return new_beams, new_angles

```

plot_radar_data.py

```

from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
from pylab import *
import matplotlib as mpl

from lat_n_lon import *
from new_beams_n_azimuths import *
from corners import *
from extremal_z import *

'''
Function for plotting radar data.
Basemap draws the map, setting it to cover Svalbard and the area around.
The loop makes the polycollections of the corners and plots them on the map
as a polygon with color based on values in radar_data.
'''

def plot_radar_data(radar_swipe, radar_angles, radar_data, zmin, zmax, variable, fig, ax,
                    sz, cmap=plt.cm.seismic, radar_lat=78.22, radar_lon=15.65):
    my_map = Basemap(projection='stere', lon_0=radar_lon, lat_0=radar_lat,
                     resolution='i', height=1670000, width=1670000)
    my_map.drawcoastlines(linewidth=0.2)
    my_map.drawcountries()
    my_map.drawmeridians(np.arange(0, 360, 30))

```

```

my_map.drawparallels(np.arange(0, 90, 5))

grid_beams, grid_angles = new_beams_n_azimuths(radar_swipe, radar_angles)
gridlats, gridlons = lat_n_lon(grid_beams, grid_angles, radar_lat, radar_lon)
norm = mpl.colors.Normalize(clip=False, vmin=zmin, vmax=zmax)

for i, j in zip(range(len(gridlons)-1), range(len(gridlats)-1)):
    x, y = my_map(gridlons[i], gridlats[j])
    x2, y2 = my_map(gridlons[i+1], gridlats[j+1])
    verts = corners(x, y, x2, y2)
    zcolor = radar_data[i]
    coll = mpl.collections.PolyCollection(verts, array=zcolor, edgecolor="none",
        cmap=cmap, linewidth=0.25, norm=norm)
    coll.set_clim(vmin=zmin, vmax=zmax)
    ax.add_collection(coll)
    ax.autoscale_view()

'''
NOTE TO SELF:
THE FORMAT OF THE COLORBAR NUMBERS, WANT SCI NUMBERING 1E-6
etc
Colorbar clim changes the axis on the colorbar, but not the colors themselves
'''

subtext_1 = r'$\log_{10} \ \ \frac{1}{m^3}$'
subtext_2 = 'K'
subtext_3 = r'$10^{-6} \ \ \frac{A}{m^2}$'

c1 = 1.0
c2 = 0.8
c3 = 0.5

if subtext_1 in variable:
    #this is for n_e
    cbar = fig.colorbar(coll, ax=ax, orientation='vertical', shrink=sz)
    cbar.set_clim(zmin+c1, zmax-c1)
if subtext_2 in variable:
    #this is for T_i
    cbar = fig.colorbar(coll, ax=ax, orientation='vertical', shrink=sz)
    cbar.set_clim(zmin*c2, zmax*c2)
if subtext_3 in variable:
    #this is for J
    cbar = fig.colorbar(coll, ax=ax, orientation='vertical', shrink=sz)
    cbar.set_clim(zmin*c3, zmax*c3)

else:
    #this is for v
    cbar = fig.colorbar(coll, ax=ax, orientation='vertical', shrink=sz)
    #cbar.set_clim(zmin*c3, zmax*c3)
    cbar.set_clim(zmin*c2, zmax*c2)

```

```

# cbar.ax.text(1.7, 1.08, r'$ 10^{-6} \left[ \frac{m}{s} \right]$',
               horizontalalignment='center', verticalalignment='top',
               #fontsize=15)

'''
Plotting the radar coordinates as a blue dot (X, Y): Longyearbyen
'''

X, Y = my_map(radar_lon, radar_lat)
my_map.plot(X, Y, 'o', color='blue', markersize=6)

return

```

regularise_array.py

```

import matplotlib.pyplot as plt
import numpy as np
from pylab import *
import mpl_toolkits.axes_grid1 as axes_grid1

'''
Function for making a regularised array,
Version 16.05.2017
'''

def regularise_array(radar_swipe, gridded_beam, z_data):
    n = len(gridded_beam)
    reg_array_zeros = np.zeros(shape=(radar_swipe.size, n))

    for i in np.arange(radar_swipe.size):
        c = np.zeros(n)
        for j in range(len(radar_swipe[i])):
            for k in range(n):
                if gridded_beam[k] >= radar_swipe[i][j]:
                    c[k] = z_data[i][j]
                    break

        c = np.array(c)
        c[c==0] = np.nan
        c_zeros = np.nan_to_num(c)
        reg_array_zeros[i] = c_zeros

    '''
    To not remove nans, comment out this:
    c_zeros = np.nan_to_num(c)
    reg_array_zeros[i] = c_zeros
    and comment in:
    reg_array_zeros[i] = c
    '''

    #reg_array_zeros[i] = c

return reg_array_zeros

```