Photoemission current derived from multiple RPC-LAP sweeps

by

Sigve Smedsrud Harang

Thesis

for the degree of

Master of Science

Faculty of Mathematics and Natural Sciences
University of Oslo

October 9, 2017

Chapter 1

Dedication

To the loved ones in my life.
To the ones that left us too soon.

Chapter 2

Acknowledgments

I give my sincerest gratitude to my supervisors and advisors: Wojciech Jacek Miloch, Cyril Simon Wedlund and Joakim J. P. Paulsson, you were always available and incredibly helpful at all times. I would also like to thank in particular Joakim J.P. Paulsson for co-supervising my thesis, it has been an absolute joy working with you.

I thank the 4DSpace team for creating an open, inspiring and professional work environment. This team is a good example of what a multidisciplinary science team can do. I would especially like to thanks the ever helpful Bjørn Lybekk, and Lei Yang that have been very supportive. I want to thank the emeriti Arne Pedersen, Alv Egeland and Karl Måseid for being a constant source of inspiration to the group

I want to thank the RPC team, and the especially the LAP team at the Swedish Institute of Space Physics in Uppsala for sharing with me an interesting topic of research. Especially Anders Eriksson, Fredrik Johansson and Elias Odelstad, it has been a pleasure working with you, on the thesis and with research.

Many thanks to friendly lecturers, Hans Pècseli, Lasse Clausen, and Per Even Sandholt.

I want to give my thanks to the other students at the Space Physics group for interesting discussions and great company during travels. Thank you so much, Birgitte, Victoria, Shafa, Gullik, Vigdis, Lukas, Henrik, Steffen and David for creating a warm and welcoming student atmosphere at the group.

Many thanks to my family and friends.

Contents

Nomenclature

Physical Constants

μ_0	Vacuum permeability , page 13	$4\pi \cdot 10^{-7}\mathrm{Hm}^{-1}$
ε_0	Vacuum permittivity , page 20	$8.85 \cdot 10^{-12}\ \mathrm{Fm}^{-1}$
k_B	The Boltzmann constant , page 13	$1.38 \cdot 10^{-23}\mathrm{m}^2\mathrm{kgs}^{-2}\mathrm{K}^{-1}$
q	Elementary charge , page 20	$1.6 \cdot 10^{-19}\mathrm{C}$
AU	Astronomical unit , page 6	$149\ 597\ 871 \cdot 10^3\mathrm{m}$
e	Electron charge , page 21	$-1.6 \cdot 10^{-19}\mathrm{C}$

Physical Variables

β	Plasma beta , page 13	None
β_i	Regression coefficient , page 35	None
λ	Wavelength , page 15	m
λ_D	Debye length , page 20	m
Φ	Electric potential, page 20	V
B	Magnetic field , page 13	T
E_e	Irradiance , page 6	Wm^{-2}
f	Frequency , page 15	Hz
I_p	The probe current , page 18	A
I_{SE}	Secondary emission current , page 92	A
n	Number density , page 13	m^{-3}
N_s	Number of sweeps , page 40	None

r_c	Spacecraft to comet radial distance , page 60	m
r_s	Radial spacecraft to the Sun distance , page 6	m
T	Temperature , page 13	K
V_b	The bias potential between probe and spacecraft , page 18	V
V_p	The probe potential with respect to space, page 18	V
V_{sc}	The spacecraft potential with respect to undisturbed plasma , page 18	V
z	General charge carrier , page 20	C
Å	Ångstrøm , page 90	10^{-10}m
j	Current density , page 23	Am^{-2}
m	Mass , page 21	kg
p	Impact parameter , page 26	m

Chapter 3

Introduction

The main topic of this thesis is the development and application of new method for detecting and analyzing the solar UV-flux using the current-voltage characteristics (IV-characteristics) from the Langmuir probes aboard the Rosetta spacecraft. First, I will present the background for the problem and the particle collection theory in order to explain the underlying equations of the method. Then, I will present an implementation of the method, which is inspected using synthetic data. Finally, I will apply the method to the intended Rosetta dataset and extract the photoemission current to the probe when the probe is sunlit. Studying the photoemission current due to the ultraviolet light is of scientific interest because it is driven by the same radiation, which is a direct source of ionization of neutrals in the cometary environment as well as changing the spacecraft potential through the photoelectric effect. (Vigren and Galand, 2013; Bodewits et al., 2016; Galand et al., 2016).

Introducing the Rosetta Mission

The European Space Agency's landmark Rosetta mission began in 1993, as a part of the Horizons 2000 science program. The Rosetta mission was an ambitious project which main purpose was the first ever long term, detailed and up close study of a comet (Schulz, 2009). After initially targeting the comet 46P/Wirtanen, the choice fell on 67P/Churyumov-Gerasimenko. The name of the comet has been given in honor of the two scientists attributed with its discovery, Churyumov and Gerasimenko (Glassmeier et al., 2007a). To accomplish the rendezvous and escort of a comet, scientists from around the world came together to build Rosetta and the landing module called Philae (Glassmeier et al., 2007a). The name of the orbiter is taken from the famous Rosetta stone, a priestly decree in three different ancient languages that was instrumental in decoding the Egyptian hieroglyphs. Philae is named after an obelisk of the same name, which was used in conjunction with the Rosetta stone in the deciphering process.

The launch of Rosetta was initially scheduled for January 2003, but was postponed to March 2004, due to an unexpected failure of the Ariane 5 rocket in late 2002. Rosetta begun its massive ten year trek from Kourou in French-Guiana. Reaching the comet at the right time and position was a serious challenge for the science team because of the multiple gravitational assist maneuvers needed to achieve a velocity comparable to that of the comet (Vasile and Pascale, 2006). During its journey the spacecraft was mostly dormant but it woke up for two fast flybys of asteroids Lutetia and Steins, before the rendez-vous with 67P/C-G. On the 20th of March 2014 the Rosetta team made its first narrow angle photograph of the comet, at that time it was seen as a little dot, about five million kilometers in the distance, see figure 3.1.

Figure 3.1: The first narrow angle image taken of 67P/C-G by Rosetta. The picture was taken at 5 million kilometers distance, the comet is found inside the small circle indicating its location. Time of capture:21 March 2014, 11:37. By ESA/MPS for OSIRIS-Team MPS/UPD/LAM/IAA/SSO/INTA/UPM/-DASP/IDA [1]

Comet 67P is a short period Jupiter-family comet, with an elliptical orbital period of about 6 years. The comet nucleus measures roughly $(4.3 \times 2.6 \times 2.1)$km,

[1]The picture was first presented on ESA's Flickr account. (ESA, 2014) http://bit.ly/2xJ8F36

and it has two lobes connected by a "neck" (Jorda et al., 2016). Fred L. Whipple wrote an influential paper on the composition of comets in 1950, which modeled comets as mostly ice with some dust (Whipple, 1950). Most of the features of his model have been confirmed, however in recent studies after obtaining more detailed mass density measurements the perception of comets as "dirty ice balls", has been changed in favor of comets as "icy dust balls" instead (Fulle et al., 2016). What this means is that going forward we must seek to understand the interplay between dust and volatiles in order to get the full picture of comet activity. The new density measurements also suggest that the comet nucleus contains more minerals than previously thought.

Comets are believed to be remnants of the earliest days of the solar system, preserved in freezing temperatures at $-220°$ Celsius in the Kuiper Belt and the Oort Cloud. Presumably there is a lot to be gained from investigating comets, because we have in them an opportunity to study bodies virtually unchanged since the beginning of the solar system (Glassmeier et al., 2007a). Comets have not gone through the same chemical progression as planets. If the chemical and physical processes which create comets can be interpreted correctly, they can reveal to us details of the early solar system nebulae (Festou et al., 2004). Essentially Rosetta aims to begin answering crucial questions about the birth of the solar system and early development of the planets, such as the forming of oceans and the genesis of organic compounds. Rosetta's mission is an all-inclusive study of 67P/C-G, intended to gather as much useful data as possible to be analyzed over time. Among the most prominent scientific topics are studies of organic compounds, efforts to probe the surface and internal structure of the comet, as well as mapping the plasma environment of the coma.

A common misconception held by many, even by the scientific community until the 1950's, is that space is essentially a perfect vacuum. The interplanetary medium, which contains all particles and fields that inhabit the region between large solar system objects like planets, the Sun or comets, is not vacuum. The space between planets is in fact inhabited by plasma. A plasma is a partially or fully ionized gas containing free electrically charged particles. It has roughly equal numbers of positive and negative charge thus being quasi-neutral at large scales. In the stationary case, a plasma looks neutral to an outside observer since the particle charge fields mutually cancel, despite being made up of charged particles. Since the particles are required to be free, they must have higher kinetic energy than potential energy due to its nearest charged neighbor. This usually means that the particles must be hot, or else they would not have the required kinetic to stay "free". Typical sources of plasma on Earth are lightning strikes and flames, which are quite rare compared to all other states of matter. It is however the most common state of matter in the universe, and more than 99%

of all known matter is in a plasma state, thanks in no small part that stars are entirely made up of plasma (Baumjohann and Treumann, 1997). Most of the plasma particles in the interplanetary medium are particles that are emitted from the Sun, and this plasma is called the solar wind (Prölss, 2012).

Approaching perihelion

As 67P is approaching the Sun, on the comet's way towards its perihelion, at about $3-4$AU the constantly increasing radiation causes the comet to emit gas and dust from the surface. This process of heating the comet body sublimates the ice on the surface of the comet, which picks up dust, and creates an outflowing atmosphere, called the coma (Combi et al., 2004). As the comet, chased by the spacecraft approach the Sun, the solar irradiance is expected to increase roughly with the square of the inversed distance:

$$E_e \propto \frac{1}{4\pi r_s^2} \tag{3.1}$$

Where E_e is the solar irradiance, and r_s is the radial distance to the Sun.
Inactive comets at large distances from the Sun, more than roughly $3-4$AU are similar to the Moon in that they are essentially big rocks absorbing the solar wind. However when activity levels increase as the comet approaches its perihelion at 1.25AU, an atmosphere akin to that of Mars or Venus emerges, complete with mass loading and boundary formations (Wedlund et al., 2016). Mass loading is when momentum and energy from the solar wind is transferred to the ions of the coma. The solar wind "pick up" ions of the coma and is subsequently decelerated and deflected due to conservation of energy (Behar et al., 2016). Volatile materials from within the comet begin to vaporize and stream into space, pulling dust with it, this mixed cloud of molecules and dust creates a long tail.The tail will face radially away from the Sun under pressure from the solar wind, however the the charged ions in the tail couple to the magnetic field as well as the flow of the solar wind and therefore has a slight angle to the dusty portion of the tail, causing the characteristic tail split (Alfvén, 1957)

The cometary atmosphere is incredibly tenuous compared to that of the Earth, it is however quite dense compared to the solar wind, normaly two orders of magnitude large. In contrast to atmospheres around strongly magnetized bodies, the coma is unshielded from ionizing radiation, the chemicals in the coma are therefore rapidly ionized, creating an ionosphere on the scale of $10^3 - 10^4$km (Combi et al., 2004). This mix of ionized molecules, ions, electrons in addition to outgassing dust makes for a very dynamic and complex plasma environment (Hansen et al., 2007).

When compared to other comets, 67P/C-G is not a large or particularly active comet. It is for example dwarfed in size and activity by the great comet Hale-Bopp[2]. Hale-Bopp has the H_2O outgassing rate of about 10^{30} molecules per second, which is roughly 5 orders of magnitude more than that of 67P at 4AU (Weaver et al., 1997), (Taylor et al., 2015). Understandably the smaller comets have a more manageable environment for sensitive scientific instruments, which is one of the reasons 67P/C-G was chosen as the target comet. nevertheless outgassing rates are subject to seasonal variations, at times forcing Rosetta to make evasive maneuvers moving to a safe distance from the nucleus.

Figure 3.2: The great comet Hale-Bopp discovered by Alan Hale and Thomas Bopp independently, here exhibiting a double tail. In 1997 it was visible from Earth for a total of 18 months. Image courtesy of NASA.

Since the beginning of the space age there have been roughly twenty space missions that targeted comets[3]. Most of these missions (12) were flybys, in which a spacecraft makes remote measurements during a brief pass of the comeet, often through the tail. A few especially notable missions have tried for something more, such as the NASA's Stardust mission of 1999, which did a flyby as well as gathering a sample which was later returned to Earth in 2006 (Brownlee et al., 2003). There was also the NASA mission of 2005, named Deep Impact which had an impact module designed to collide with the comet nucleus, forming a crater and

[2]A comet is called great if it becomes visible to the naked eye at any time during its orbit.
[3]A list of space missions related to comets is available at http://bit.ly/2xHlXdl

blowing up dust facilitating interesting studies (A'Hearn et al., 2005). Building on previous experience, the Rosetta mission went even further with Philae's soft landing on the comet surface. Furthermore the mission duration was sufficient to capture a full passage of the comet near to the the Sun, allowing for a detailed study lasting two years. The mission duration made it possible to study the time evolution of the comets active phase at the inbound and outbound trajectory, as well as perihelion, the time of highest cometary activity.

The Rosetta Plasma Consortium (RPC)

The Rosetta mission is an international partnership between approximately fifty scientific groups. Collaborations like these allow the scientific community to undertake missions which are far too challenging and costly for any single team. The orbiter scientific payload was created by science teams from both Europe and the United States, while the lander was provided by a European team led by the German Aerospace Research Institute (DLR) (Bibring et al., 2007).

One of the mission objectives of Rosetta was to fully investigate and form a more complete picture of the plasma environment around comet 67P/C-G. Although the comet is embedded into the solar wind, it does have a plasma environment of its own. It is this environment and its coupling to the solar wind that scientists want to investigate further. In order to study the atmosphere of the comet in detail, Rosetta has been equipped with a set of plasma instruments specifically tailored to this task which is called the Rosetta Plasma Consortium (RPC) (Carr et al., 2007). In this thesis I work solely with data from the RPC-LAP instrument, as seen in figure 3.3, the dual Langmuir probe system sit at the end of two long booms. The probes are installed as far as possible from the main body of the spacecraft in order to reduce interference from the spacecraft (Carr et al., 2007).

The following list contains instruments that make up the Rosetta Plasma Consortium, with reference to the papers describing their objectives and operations.

- Ion Composition Analyzer (ICA), (Nilsson et al., 2007)
- Ion and Electron Sensor (IES), (Burch et al., 2007)
- Langmuir Probe (LAP), (Eriksson et al., 2007)
- Flux gate Magnetometer (MAG), (Glassmeier et al., 2007b)
- Mutual Impedance Probe (MIP), (Trotignon et al., 2007)

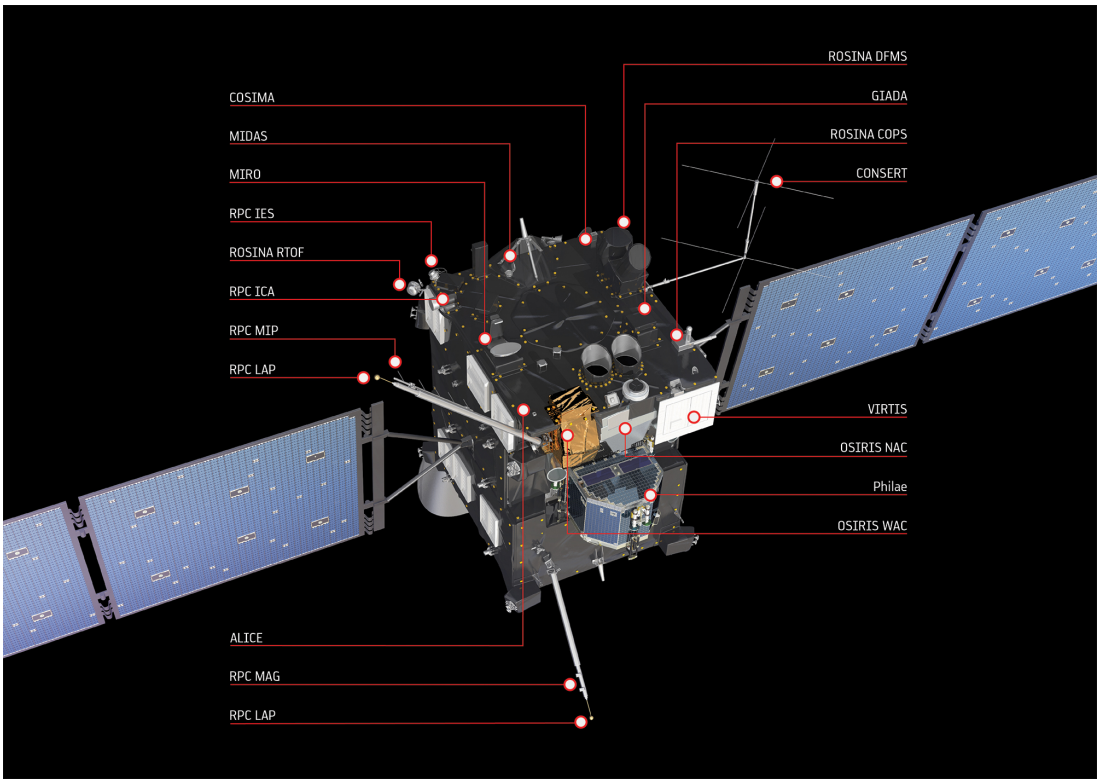


Figure 3.3: The Rosetta Spacecrafts instruments. Image credit: The European Space Agency.

These five instruments together form a comprehensive in-situ plasma laboratory with the purpose of investigating among others these areas of interest:

1. Physical properties of the cometary nucleus and its surface: By physical properties we mean remnant magnetization of the surface, electrical properties such as surface charging, as well as any surface perturbation due to solar wind interactions.

2. The inner coma structure, dynamics and aeronomy: The coma is a sort of atmosphere that forms around a comet as it approaches the Sun and is gradually heated, the coma is generally made up of evaporated ice and dust. Aeronomy in this context is the study of ionization and dissociation of an atmosphere, phenomena that are present in the coma.

3. The development of cometary activity: The comet was undergoing major changes during its lap around the Sun, Rosetta begun tracking the comet at about 3AU, and followed 67P/C-G to its perihelion at 1AU and back out.

This thesis is connected to point three from the list. Unfortunately there is no instrument which is dedicated to the study of solar radiation mounted on Rosetta. The team has however found a way to utilize the Langmuir Probes as an ad-hoc proxy for the solar flux. This means that we can use data analysis techniques to obtain a crude estimate of the solar flux, even though the instrument was not specifically designed for this purpose. This method of extracting additional data from the Langmuir probe is the crux of this thesis, and the name of the method is the multiple sweeps method. This method utilizes the current-voltage characteristics provided by the LAP instrument in the Langmuir probe mode. A thorough description of the method will be given in chapter 5. The focus is on successive analysis with the current-voltage characteristics. By optimizing the way we analyze these characteristics, we mainly hope to improve the method's cadence, which will possibly open up new avenues of investigation. We also give insight into the effectiveness and robustness of the method by testing the method using synthetic data.

The Langmuir Probe

The central instrument in this thesis is the RPC Langmuir Probe sensor (RPC-LAP), mounted on the Rosetta spacecraft (Eriksson et al., 2007). The Langmuir probe is in theory and practice a rather simple device. One can make a crude adaptation of the probe with an insulated wire, a DC power supply and an oscilloscope (Huddlestone, 1965). Even though the sensor itself is uncomplicated in design, the analysis however is deceptively complex, as a multitude of challenges appear in the probe theory and in the interpretation of measurements. The Langmuir probe will usually be operated with an applied voltage, drawing a current from the plasma. Precautions must be taken when using Langmuir probe because there is a possibility that unless certain conditions are met, the perturbation of the plasma introduced by the probe itself, is non-localized (Huddlestone, 1965).

The Langmuir probe is named after the Nobel Laureate Irving Langmuir (1881-1957), who is famous for his work in developing lightbulb technology, as well as his research on plasma and atomic theory (Taylor, 1958). He was among the very first to actually coin the term plasma, and he introduced the concept of electron temperature. In 1924 Langmuir invented the method of measuring plasma density and temperature using electric probes, the method is now a mainstay of plasma laboratories around the world, and is named in his honor. A photograph of a probe that is identical to the probes that fly aboard Rosetta is pictured in figure 3.4.

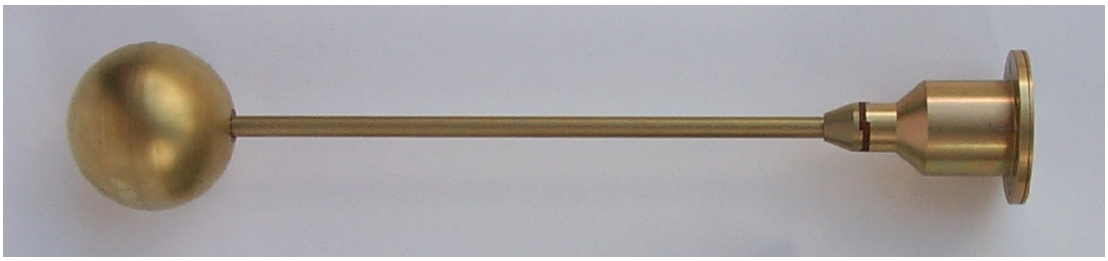


Figure 3.4: A Langmuir spherical probe identical to the ones on the Rosetta spacecraft. These probes were made for ESA by the Swedish Institute of Space Physics in Uppsala. It is a titanium crafted probe with titanium nitride coating, the diameter of the crown is 5 centimeters. Photo by AndersIE (Own work) [Public domain], via Wikimedia Commons

The Sun and Solar Wind

The foremost driver of interplanetary plasma events in the solar system is the Sun. The interplanetary medium is sustained by the Sun, which is constantly spewing out plasma and radiation. Comets embedded into the interplanetary medium display very interesting behavior in contact with the solar wind. As they approach the Sun, under increased radiation, an atmosphere is formed complete with boundary layers and shock formations (Mendis and Houpis, 1982). Exploring the solar wind - comet interactions is one important goal of the Rosetta mission, one of the central problems between the two is the ionization of cometary neutrals by solar radiation. Thus, the method presented in this thesis may be a helpful tool for studies relating to the long or short term solar radiation levels at 67P/C-G.

The Sun is constantly producing vast amounts of energy by nuclear fusion processes in the core, see figure 3.5. The process of transporting this energy to the surface is one that takes of the order of 10.000 years, mostly due to the radiative transport phase from the core to the convective zone. In the convective zone the energy transport mechanisms create incredibly strong magnetic fields close to the surface of the Sun. These magnetic fields may occasionally break into the photosphere, which is essentially the surface of the Sun. There the magnetic fields manifest as sunspots of colder plasma relative to the environment. It is within these sunspots that we most often observe coronal mass ejections and flares. Above the surface we find the chromosphere which is about $3 - 4000$km thick with spicules reaching even further out up to 10.000km (Prölss, 2012). The density and temperature fall with distance from the surface in the chromosphere only to rapidly increase in the above lying transition region between the chromosphere and the outermost layer which is the corona. In the transition region and corona the temperature rise from just a few thousand Kelvin to about 1

million Kelvin over a very short distance, why this happens is still an active area of research.

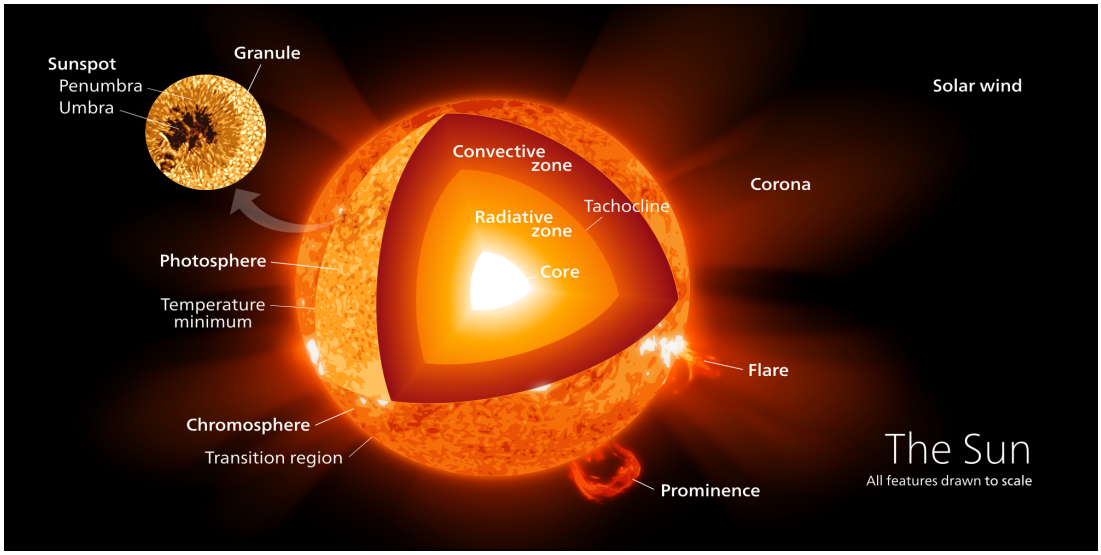


Figure 3.5: The different layers of the Sun. A massive amount of energy $E = 4{*}10^{26}$W is produced in the core by fusion processes and then transported to the surface in a process that takes about 10.000 years. The convective zone is dominated by plasma flow processes that create intense magnetic fields that further fuels interplanetary processes, through. By Kelvinsong (Own work) [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

The activity level of the Sun is more or less constant, in the visible radiation and EU radiation spectrum it varies by approximately 0.3%, meaning that there is little to no change in its power output on average (Prölss, 2012). However the activity level of the Sun is subject to a 11 year cycle, of which we are in recorded cycle number 24, since reliable data was first gathered in 1755 (Kane, 2002). These cycles do have a significant impact on the observed activity level, as it turns out solar cycle 24 is among the lowest activity cycles in the recorded data with only minimal sunspot activity (Wang and Colaninno, 2014). Even during years of low solar activity there might be several strong solar flares, in high activity years there can typically be roughly 70 major flare events (Prölss, 2012). In the case of geoeffective events, such as Coronal Mass Ejections (CME's), some are carried with the solar wind directly towards the Earth. The most severe of these space weather events can cause damage to satellites and in some cases even incapacitate important infrastructure on Earth, such as power grids and GPS services (Hapgood, 2011). The solar wind and its dynamics does not fall within the scope of this thesis, but we need some background information on the Sun

and solar transient events, as some of these phenomena may influence operations of the RPC-LAP instrument. By solar wind transient we understand, short term events typically one day or less, such as CME's, and solar flares (Schwenn, 1996).

Coronal mass ejections and prominences

Looking at the Sun through a coronograph, which is a telescopic lens that blocks out the main body of the Sun so that the surroundings are more easily discernible, we can sometimes see large rings of magnetic fields and plasma extending from the solar surface called prominences when viewed in profile, and filaments when viewed from above (Pesnell, 2015). These prominences are relatively cool plasma rings at approximately eight thousand Kelvin that is extending from the chromosphere into the much hotter corona which is about one million Kelvin. (Tandberg-Hanssen, 1974). Prominences are found in both active and inactive regions of the Sun, normally with a footprint in two regions of the opposite magnetic polarity. The ratio of plasma pressure to magnetic pressure is called the plasma beta $\beta = \frac{nk_BT}{B^2/(2\mu_0)}$. Where n is the particle number density, k_B is Boltzmann's constant, B is the magnetic field strength, T is the temperature, μ_0 is vacuum permeability. In the lower parts of the corona the plasma beta small, $\beta < 1$, which in turn means that the plasma is magnetic pressure dominated. Prominences are in effect supported against gravity by their magnetic fields, which can become unstable leading to the prominence eruption and Coronal Mass Ejections (Munro et al., 1979). These eruptions of stored magnetic energy are in fact thought to be the driver for Coronal Mass Ejections as well as solar Flares (Ronald L Moore, 1988). Prominences and eruptions are very much an active area of research and the underlying physics is not yet fully understood.

Coronal Mass Ejections are thought to occur when magnetic field over time gathers energy up to a point where magnetic pressure overcomes the magnetic tension and abruptly erupts. The eruption is supported by magnetic reconnection, which is when the topology of the magnetic field lines is rearranged so that magnetic energy is converted to kinetic energy, particle acceleration and increased temperature of the plasma. The magnetic reconnection effect is qualitatively an effect that splices magnetic field lines of two different domains, this effect is thought to power the outward expansion and subsequent escape into space of CME's, (J Lin and T G Forbes, 2000). These transient events will likely cause significant perturbations of the background solar flux intensity measured at Rosetta drastically increased EU and EUV-radiation over a certain period of time.

Class	Peak Flux $[W/m^2]$ in $(1-800\text{pm})$ range
A	$< 10^{-7}$
B	$10^{-7} - 10^{-6}$
C	$10^{-6} - 10^{-5}$
M	$10^{-5} - 10^{-4}$
X	$> 10^{-4}$

Table 3.1: The different flare categories as observed in the picometer range by the GOES spacecraft at Earth, all values in watts per square meter $[W/m^2]$. Each class is a decade, meaning that B is ten times stronger than A, while C is 100 times stronger than A. Each class is also subdivided into a linear scale. So an C2 is twice as powerful as C1 and four times as powerful as A B5.

Solar flares

Solar flares release electromagnetic radiation by all wavelengths λ, from ultraviolet to infrared. However in this thesis I am most interested in the UVB, or "near-UV", and higher frequency spectrum of radiation. The RPC-LAP material and coating have a work function, that requires radiation stronger than UV to free photoelectrons, (Westlinder et al., 2004). Thus the lower limit of radiation energy that the probe is sensitive to is the UV.

Solar flares are thought to be closely linked with Coronal Mass Ejections, as they are often observed in conjunction, although their exact relationship remains unclear (Zhang et al., 2001; Harrison, 1995). We say that flares are released, since the energy of flares most often come from the stored energy in magnetic loops connecting solar sunspots (Schmieder and Aulanier, 2012). A flare event is when the magnetic free energy, that is the energy stored in the magnetic field which is above the magnetic potential in the same field, is released from the solar atmosphere into the interplanetary medium. A major event may release as much as $E = 10^{25}$J of energy into space, the released energy is concentrated in radiation and plasma traveling outwards into space.

There are many ways to classify flares, as they have different properties in different spectra and compositions. The most widely spread classification today is the GOES peak flux range scale, which sorts flares into five different categories depending on the peak flux radiation in the $100 - 800$ picometer range as measured by the GOES spacecraft. The classes are A, B, C, M and X see table 3.1 for the respective peak flux of each class. Class A flares can be considered slightly more intensive than the background radiation, while X class are among the most luminous phenomenons in the Solar System.

It should be noted that the GOES flare categorization can be slightly misleading in terms of categorizing flare power output and irradiance. The GOES classification emerges from the peak flux in a certain wavelength range, however any given flare has an unique distribution function depending on wavelength. This means that the total integrated flux over all wavelengths is not necessarily correlated with the strongest peak flux in a specific frequency domain (Chamberlin et al., 2012). Flares are normally associated with a sharp and impulsive release of energy in the X-ray spectrum, but it is easy to forget that flares release energy across all wavelengths. Normally the time evolution of flares can be divided into an impulsive and a gradual phase. A typical flare signature in the high energy spectrum has a sudden peak, where the flux dramatically rises in the matter of a few minutes, followed by a quick decrease during the gradual phase before returning to background levels. It has therefore been suggested that the energy release in the gradual phase across many lower frequencies is as important as the short-lived flare peak. In a statistical study of flares concerning the specific issue of energy release in flares, (Chamberlin et al., 2012) found that GOES may underestimate the total energy released by as much as 300%, because it does not consider atypical energy distributions.

The LAP instrument is sensitive to electromagnetic radiation energies above 4.2eV which roughly corresponds to radiation above $f = 10^{15}$Hz or equivalently below $\lambda = 295$nm. It follows that any flare signals therefore are the integrated light curves of UVB and higher frequencies. A flare is therefore expected to be recognizable as a sudden instrument response on the order of minutes followed by a gradual decrease over the duration of tens of minutes to hours, where the relative strength of the highest peak corresponds to the GOES classification. In practice, the peak will not be reliably detectable using the multiple sweeps method, as we lack the cadence to reliably locate the peak, furthermore it may prove difficult to relate Rosetta and the comets position to the path of a flare with respect to the GOES observations made in vicinity of Earth. The comparison to the GOES system is nevertheless an interesting one to make as a general guideline, especially if we can reliably compare flare events detected at Rosetta with those registered by satellites or ground instruments that study flares.

Chapter 4

Theoretical Basis

The probe characteristics

The basic operational mode of the probe is to immerse it into plasma, apply voltage which gradually sweeps from negative to positive potential or vice versa. We can then analyze the total collected current (I) versus voltage (V), called the IV-characteristics of the probe. In short the purpose of analyzing the characteristic is that we can use the characteristic to extract basic plasma parameters such as density and temperature. More involved methods with multiple probes can also measure plasma oscillations and flow. (Huddlestone and Chen, 1965) and (Nagaoka et al., 2001). In this chapter I present basics of the probe theory and IV-characteristics which is the backbone of the method presented later in this thesis.

The most important property of the probe is that it is capable of making local measurements of plasma at relatively high spatial and temporal resolution. As opposed to other techniques like microwave propagation and spectroscopy, which measures averages over large volumes of plasma(Huddlestone, 1965). The benefit is obvious, with higher resolution we can do more in depth analysis of space plasma.

However, the theory is based upon the assumption that the velocity distribution of particles in the plasma is Maxwellian. This assumption can be problematic, because we might not always know what sort of velocity distribution we are dealing with in the experiment. Furthermore the placement of a conducting probe in a plasma environment is bound to perturb the plasma itself. Invariably we see the development of a plasma sheath as a boundary region between the probe and the unperturbed plasma. The sheath region is such an important concept which was discussed, by Mott-Smith and Langmuir (Mott-Smith and Langmuir, 1926), who developed the original probe theory. The sheath region is important because it raises significant issues with relating the measured total

probe current to actual plasma parameters.

General probe characteristics

A typical idealized probe characteristic or IV-characteristics, is shown in figure 4.1, and may be divided into three distinct regions; the ion saturation region, transition region and electron saturation region. The probe measures a current while stepping through a series of bias voltage steps which are with respect to the ground of the spacecraft. The spacecraft has a potential with respect to space, which means that the absolute potential between the probe and the potential of the plasma at infinity is:

$$V_p = V_b + V_{sc} \tag{4.1}$$

Where V_p is the probe potential, V_b is the applied bias voltage and V_{sc} is the spacecraft potential. When the probe potential is negative with respect to space, we are in the ion saturation region, (region A in figure 4.1). It is called the ion saturation region because as the negative bias is increased, eventually all ions in the plasma will be collected and all electrons repelled. Increasing the negative bias potential will naturally cause a build-up of charge from the collected ions. If we try to increase the negative voltage even further, the ion current will eventually saturate. The saturation occurs because the area of the charge region, depends only weakly on voltage, and therefore does not increase to infinity (Huddlestone and Chen, 1965).

As the voltage moves towards a positive bias we enter the transition region denoted by a B in figure 4.1. With the bias voltage gradually increasing, becoming more positive, electrons are eventually able to overcome the potential barrier and reach the probe. At this point the current current regions are in a transition between two states of negative and positive saturation. At one point on the curve we notice that the probe current I_p, this point is called the floating point. At the floating point the negative and positive current cancel out and the net current to the probe is therefore zero.

The transition region can provide information about the electron distribution. It can be shown that if the transition region is purely exponential after discarding the ion current portion, then we are dealing with a Maxwellian electron distribution (Huddlestone, 1965). Discarding the ion current can be done by using a model for the ion current, and subtracting it from the total probe current. The intercept between the two linear fits marked with dotted blue lines in figure 4.1, marks the space potential. The space potential is the potential at which there is no potential difference between the probe and space. Which means that no

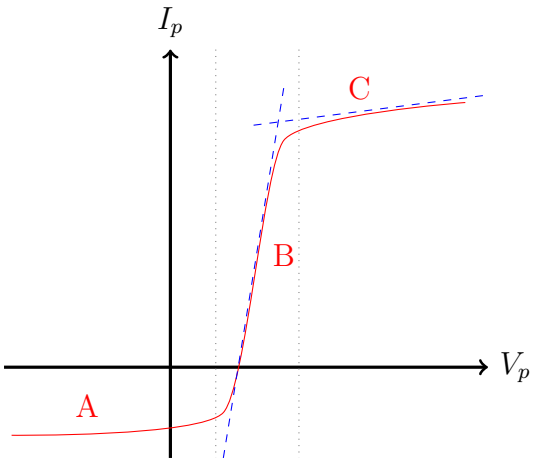


Figure 4.1: Idealized IV-characteristic for a single Langmuir probe. The three distinct regions of the characteristic is marked by vertical dotted lines, (A) the ion saturation region. (B) The transition region, and (C) the electron saturation region. By convention positive probe current is defined as current from the probe to the plasma, i.e electrons collected by the probe is considered positive current. See text for description of the different regions.

particles are attracted or repelled. A typical property of space plasmas is that the electron population has a higher thermal energy than the ion population. This means that when there is no the probe is at the space potential, the current measured by the instrument is dominated by the electron current. We therefore locate the space potential in between the transition region and the electron saturation region. Increasing the bias voltage even further beyond the transition region and the space potential, we eventually reach the electron saturation current region marked with C in figure 4.1. This region has a similar shape to that of the ion saturation region. However, the absolute value is normally much higher for electrons because of the mass disparity between mobile electrons and immobile ions in space plasmas. As with the ion saturation region the current has a relatively weak dependence on the voltage, so the ion current eventually saturates.

Assumptions

In the first approach we consider an idealized plasma with the following properties:

- Maxwellian energy/velocity distribution
- Unmagnetized: No magnetic fields acting on the particles.

- Collisionless: By collisionless we mean that the mean free path of a particle is much larger than all of the characteristic lengths in the system, such as the probe diameter (r_p) and the Debye shielding length (λ_D).
- Velocity: In all space plasmas the electrons are much faster than the ions, because ions have a much larger inertia than electrons. This is important in relation to the formation of boundary layers.

The Debye shielding length

A fundamental question when operating Langmuir probes is, how far does the probe perturbation reach into the space plasma? Suppose the reach in a dielectric medium is set up by the Debye shielding effect. The Debye length, which is a characteristic length of the electrostatic effect and subsequent perturbation reach, is given as (Hutchinson, 2002):

$$\lambda_D = \sqrt{\frac{\varepsilon_0 k_B / q_e^2}{n_e/T_e + \sum_j z_j^2 n_j/T_j}} \tag{4.2}$$

Where ε_0 is the vacuum permittivity, k_B is the Boltzmann constant, q is an elementary charge. Units denoted e is for electron, while denoted j means arbitrary particle species. n is the particle number density, z is a general charge carrier, T is the temperature. This equation takes into account different charge carrier species, but in space physics applications there is often room for simplification. Assuming that ion mobility is negligible to the time scales of the interaction in question, one can use this alternative formula (Prölss, 2012):

$$\lambda_D = \sqrt{\frac{\varepsilon_0 k_B T_e}{n_e^2}} \tag{4.3}$$

Considering typical plasma parameters observed by the Rosetta team, we find that the electron temperature is at roughly 5eV (Odelstad et al., 2016), while the typical densities observed are in the region of 150cm^{-3} (Edberg et al., 2015). Inserting these values into the Debye length formula yields a characteristic length scale of the perturbations in the region of $\lambda_D \approx 1.4$m, which means we are definitively inside the sheath when probe 1 and probe 2, are mounted at the tip of booms that are 2.2m and 1.6m from the spacecraft. The electric potential associated with the Debye length can be given as:

$$\Phi = \Phi(r) \exp\left(\frac{-r}{\lambda_D}\right) = \frac{e}{4\pi\varepsilon_0 r} \exp\left(\frac{-r}{\lambda_D}\right) \tag{4.4}$$

For every λ_D, the electric potential Φ will diminish by a factor of $1/e$. Physically the Debye effect in plasmas arises when we disturb the particle density of a quasi-

neutral plasma and set up an electrostatic potential, which is then attenuated with distance by the Debye shielding effect. We follow the derivation by (Pécseli, 2012), and take the Poisson equation for potential field in space:

$$\nabla^2\Phi(\boldsymbol{r}) = \frac{e}{\varepsilon_0}\left[n_i(\boldsymbol{r}) + n_e(\boldsymbol{r})\right] \tag{4.5}$$

Where n_i and n_e are the ion and electron densities. $\Phi(\boldsymbol{r})$ is the potential of a positive point charge. Let us assume that the plasma density is equal to the ion density, $n = n_i$, as well as an electron velocity distribution on the form:

$$f_e(\boldsymbol{u}, \boldsymbol{r}) = n_0 \left(\frac{m}{2\pi k_B T_e}\right)^{3/2} \exp\left(-\frac{1/2mu^2 - q_e\Phi(\boldsymbol{r})}{k_B T_e}\right) \tag{4.6}$$

Where m is for mass, and the plasma is isothermal, u is the velocity. Let us integrate with respect to $\boldsymbol{u}$:

$$\int_{-\infty}^{\infty} f_e(\boldsymbol{u}, \boldsymbol{r}) d\boldsymbol{u} \equiv n_e(\boldsymbol{r}) = n_0 \exp\left(\frac{e\Phi(\boldsymbol{r})}{k_B T_e}\right), \tag{4.7}$$

which is the Boltzmann isothermal velocity distribution. Now this potential is a bit problematic, it is a second-order, nonlinear differential equation which in general is not easily solvable. We can however simplify things if we assume that the energy of the particles in the potential field is much smaller than their thermal energy $eV \ll k_B T_e$. Keeping only the first terms of the Taylor expansion we arrive at:

$$n_e(\boldsymbol{r}) = n_0 \left(1 + \frac{e\Phi(\boldsymbol{r})}{k_B T_e}\right) \tag{4.8}$$

Inserting this result back into equation 4.5, we get:

$$\nabla^2\Phi(\boldsymbol{r}) = \frac{e}{\varepsilon_0}\left[n_0\left(1 + \frac{e\Phi(\boldsymbol{r})}{k_B T_e}\right) - n_0\right] = \frac{e}{\varepsilon}\left(\frac{e\Phi(\boldsymbol{r})}{k_B T_e}\right) \tag{4.9}$$

We can solve this version of the Poisson equation in different dimensional spaces. In this derivation we choose the simplest, which is the one dimensional solution, by linearization:

$$\Phi_{1d}(r) = a \exp\left(-\sqrt{\frac{rnq_e^2}{\varepsilon_0 k_B T_e}}\right) = a \exp\left(-\frac{|r|}{\lambda_D}\right) \tag{4.10}$$

Where the scaling factor of λ_D attenuating the potential with distance is interpreted as the Debye length, a is an arbitrary constant. This result can also be quickly obtained by dimensional analysis, since the only combination of temperature, density, Boltzmann's constant, vacuum permeability and electric charge that gives a non-trivial constant of unit length is precisely the Debye length (Pécseli, 2012).

The plasma sheath

The plasma sheath, also referred to as "the Debye sheath"is a layer in plasma of disturbed charge density. Usually a sheath is formed in response to a region of increased charge density, sustaining net charge neutrality. In the general case of some object with an acquired surface charge, the plasma response in the surrounding plasma is typically several Debye lengths. A natural trait of plasma is to try preserving net charge neutrality at all times (Merlino, 2007).

Suppose we insert an isolated body into a space plasma, then the question is: What happens to the plasma in the vicinity of the body? This question is obviously very sensitive to the nature of the plasma in question. But for normal space plasma conditions we can make a few assumptions. To begin with, we can model the background plasma as charge neutral and fully ionized with the electron temperature being approximately equal to the ion temperature $T_e \approx T_i$. Which in other words means that the electrons are much more mobile than the massive immobile ions (Baumjohann and Treumann, 1997). The surface of an object immersed in space plasma will then immediately start to absorb the fast moving negative electrons gathering a negative surface charge (Lochte-Holtgreven and Richter, 1968). Because of quasi-neutrality, the surface charge must be balanced by a positive space charge, we identify this space charge region as the sheath. If we consider a steady state situation for the purpose of explaining some characteristics of the plasma sheath, then naturally, an equilibrium state must be achieved after some characteristic time. In the steady state picture, we have an isolated body with a negative surface charge, outside of which there exists a positive space charge and possibly a boundary region towards the undisturbed plasma.

The Bohm criterion

Let us assume that we have immersed an objects wall into a plasma which will immediately begin absorbing electrons and gaining negative charge. Increasing the negative potential of the wall to the plasma creates a potential barrier for the plasma electrons. After a while a steady-state system appears which has no net current and the negative wall charge is balanced by the space plasma. Consider an ion coming from infinity with initial velocity u_0, that enters the sheath at $r = 0$. If we want to calculate the potential drop towards the the wall as a function of r. We assume that there are no collisions on the path of the ion to the wall, then conservation of energy requires:

$$\frac{1}{2}m_i u = \frac{1}{2}m_i u_0 - e\phi(r) \tag{4.11}$$

The continuity equation requires that $n_0 u_0 = n(r)u(r)$, which combined with equation 4.11 gives:

$$n_i(r) = n_0 \left(1 - \frac{2e\phi}{m_i u_0^2}\right)^{1/2} \tag{4.12}$$

To further investigate such an electrostatic problem we need the Poisson equation 4.5 to describe the potential field. Furthermore we need an expression for the electron density n_e, a reasonable assumption is to take the Boltzmann relation for electrons (Chung and Hutchinson, 1988):

$$n_e = n_{e,0} \exp\left(\frac{e\phi}{k_B T_e}\right) \tag{4.13}$$

Inserting equation 4.12 and 4.13 in to the Poisson equation 4.5:

$$\nabla^2 \phi = \frac{e n_0}{\varepsilon_0} \left[\exp\left(\frac{e\phi}{k_B T_e}\right) - \left(1 - \frac{2e\phi}{m_i u_0^2}\right)^{1/2}\right] \tag{4.14}$$

We already assumed that the potential drop as we approach the wall repels electrons, in other words we require that potential is decreasing with increasing r. From the Poisson equation 4.5 it means that $n_i(r) > n_e(r)$, and thus that:

$$\exp\left(\frac{e\phi}{k_B T_e}\right) < \left(1 - \frac{2e\phi}{m_i u_0^2}\right)^{1/2} \tag{4.15}$$

Now using the Taylor expansion for small $|\phi|$:

$$1 + \frac{e\phi}{k_B T_e} < 1 + \frac{e\phi}{m_i u_0^2} \tag{4.16}$$

We can now solve for the ion velocity u_0 at the sheath edge, assuming negative potential $\phi < 0$, we arrive at the Bohm criterion Riemann (1991):

$$u_0 > \sqrt{\frac{k_B T_e}{m_i}} \tag{4.17}$$

This shows that ions must be accelerated before reaching the sheath to match the Bohm velocity $u_B = \sqrt{(k_B T_e)/m_i}$. The acceleration is provided by relatively weak electric fields in a region prior to the sheath, which is typically wider than the sheath, called the pre-sheath (Lochte-Holtgreven and Richter, 1968).

Child-Langmuir law

In preparation for the probe specific theory of collecting charged particles in a plasma using biased probes, it is of interest to inspect a more general case. The Child-Langmuir Law law gives the maximum of space-charge-limited current j

in a planar diode. By combining the Bohm-Criterion (4.16) with the Child-Langmuir Law, we want to arrive at a general probe current density (Child, 1911).

Let us consider two infinite plates A and B separated by a distance d. Plate A is positively charged and will force ions to move from A to B by Coulomb's law, i.e the electric force. The potential at A is zero while at B it is ϕ, the potential difference is then ϕ.

From the Poisson equation 4.5 with boundary conditions, $\phi_A(x = 0) = 0$, $\phi_B(x = d) = \phi$ and $\frac{\phi_A(x=0)}{dx} = 0$.

$$\nabla^2\phi = -\frac{\rho}{\varepsilon_0} = -\frac{j}{u\varepsilon_0} \tag{4.18}$$

Where $j = \rho u_i$. Let us take $-e\phi = 1/2mu^2$ solve for u and insert into equation 4.18:

$$\nabla^2\phi = \frac{d}{dx}\left(\frac{d\phi}{dx}\right) = -\frac{j}{\varepsilon_0 v}\sqrt{\frac{m}{-2q}}\phi^{-1/2}$$

Multiplying by $d\phi$, separating variables and integrating on both sides gives:

$$\frac{1}{2}\left(\frac{d\phi}{dx}\right)^2 = 2\frac{4}{9}\sqrt{\frac{-2q}{m}}\phi^{1/2} + C = 2K\phi^{1/2} + C$$

Where $K = \frac{4}{9}\sqrt{-2q/m}$. The integration constant is $C = \frac{1}{2}\left(\frac{d\phi}{dx}\right)^2$ at A. From initial conditions, for increasing currents $\frac{d\phi}{dx} \to 0$ and for maximum current it is zero $\frac{d\phi}{dx} = 0$. The maximum current is then:

$$\frac{d\phi}{dx} = 2\sqrt{D}\phi^{1/4} \tag{4.19}$$

Where $D = -\frac{j}{\varepsilon_0 u}\sqrt{\frac{m}{-2q}}$. Integrating again gives:

$$\frac{2}{3}\phi^{3/4} = D^{1/2}x$$

Putting the D back in:

$$\frac{2}{3}\phi^{3/4} = \left(-\frac{j}{\varepsilon_0 u}\sqrt{\frac{m}{-2q}}\right)^{1/2} x$$

Solving for j and inserting the distance d between the plates for x:

$$j = \frac{4}{9}\phi^{3/2}\varepsilon_0 v\sqrt{\frac{-2q}{m}}\frac{1}{d^2} = -K\phi^{3/2}\frac{1}{d^2} \tag{4.20}$$

This is the largest current that positive ions can carry due to potential difference between to infinite and parallel plane diodes separated by a distance d.

Particle Collection Theory

We need a charged particle model which explains the current collection for the LAP instrument. Since it is a charged, spherical probe with radius $r_p = 2.5$cm, and it is immersed in plasma with Debye length $\lambda_D \approx 1.4$m, we are in the thick sheath limit. Which means that not all particles that enter the sheath of the probe will be collected. They can be modeled using the orbital motion limited current theory (OML theory).

Orbital motion limit collection

Orbital motion limit (OML) theory, deals with the collection of charge carriers in plasma by biased probes or surfaces that are much smaller than the Debye length. The problem of charge collection in plasma is in some ways reminiscent of Keplerian dynamics in terms of orbits and Rutherford scattering in the use of impact parameters. The theory is based on a few important assumptions. The first is that the object placed in the plasma is smaller than the Debye shielding length λ_D, and the second is that the plasma is stationary. We also assume that the particle stream is collisionless and that the plasma is unmagnetized. The last assumption is that the potential from infinity to the probe is smoothly varying. (Lochte-Holtgreven and Richter, 1968).

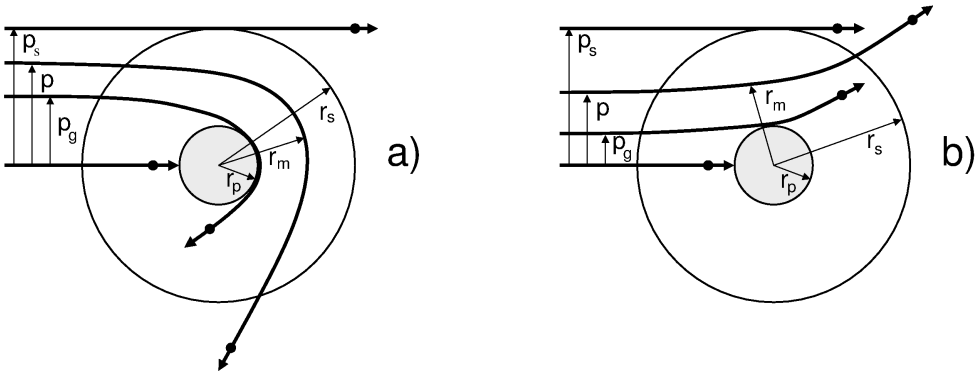


Figure 4.2: Spherical probe orbits for attraction potential a), and reflecting potential for b). r_p is the probe radius, r_m is the trajectory radius, while r_s is the sheath radius, equivalently for the impact parameters p. Appropriated with author permission from (Pécseli, 2012).

Let us, consider a particle at infinite distance from the probe. We set the potential at infinity to be $\phi_\infty = 0$, and the particles, we choose in this case to

work with ions, have an initial velocity of u_0. The probe is negatively biased and since we assumed that the potential is smoothly varying, the attracted particles always feel acceleration towards the probe from infinity.

The parameter governing the collection of particles is the impact parameter p. When particles approach the probe they are forced into an orbital trajectory because of the potential drop around the probe with characteristic orbits as in figure 4.2. The impact parameter is simply the vertical distance from the trajectory of the particle to the center of the potential field.

The name of the theory implies that the collected particles are limited by the orbital motion given by the equations for energy and angular momentum. Our aim is to derive an expression for the probe current I_p as a function of probe potential ϕ_p, by dividing the particle current into infinitesimal packets:

$$dI_p = A_p qudn = \pi p^2 qudn \tag{4.21}$$

Where $A_p = \pi r_p^2$ is the cross-sectional area of the probe. The grazing impact parameter p_g is the impact parameter that causes the particle trajectory to be a tangent to the probe surface, i.e the largest impact parameter that is collected by the probe. Any particles with impact parameter $p > p_g$ will miss the surface, while particles with $p \leq p_g$ will hit the surface and therefore contribute to the probe current. Trajectories of particles with impact parameter p larger than the sheath radius r_s, will be assumed to continue undisturbed[1].

The OML electron Current

Roughly following a derivation of the electron saturation current by Melzer, Andre (2017) and in (Lochte-Holtgreven and Richter, 1968), I will first derive the electron saturation current for a negative probe potential. In the negative potential field regime of $e\phi_p < 0$, ions are attracted and electrons repelled. Starting at infinity an electron with critical impact parameter p_g will have the angular momentum:

$$L = |\boldsymbol{r} \times m\boldsymbol{u}| = |\boldsymbol{r} \times \boldsymbol{p}| = m_e u_\infty p_g \tag{4.22}$$

Where L is the angular momentum, $\boldsymbol{r}$ is the position vector, m, $\boldsymbol{u}$, and p is the mass, velocity and momentum of the particle. Collecting all electrons that are hitting the probe with tangential incident, the angular momentum L at the surface is then given as:

[1] Theoretically another solution exist for particles with imaginary impact parameter. These particles can have closed orbits within the sheath and thus never be collected in the collisionless case (Medicus, 1961).

$$L = m_e u_e r_p \tag{4.23}$$

We can set up an energy budget from infinity to the probe surface, using conservation of angular momentum:

$$\frac{1}{2} m_e u_{e,\infty}^2 = \frac{1}{2} m_e u_e^2 - e\phi_p \tag{4.24}$$

Included in the energy budget is the kinetic and potential energy of the electron in the probe potential field. Assuming a non-collisional particle beam, the angular momentum of the electron must be conserved and we get:

$$\frac{1}{2} m_e u_{e,\infty}^2 = \frac{1}{2} m_e u_{e,\infty}^2 \left(\frac{u_e^2}{u_{e,\infty}^2} - \frac{e\phi_p}{1/2 m_e u_{e,\infty}^2} \right) = \frac{1}{2} m_e u_{e,\infty}^2 \left(\frac{p_g^2}{r_p^2} - \frac{e\phi_p}{1/2 m_e u_{e,\infty}^2} \right) \tag{4.25}$$

Solving for the critical impact parameter p_g:

$$p_g^2 = r_p^2 \left(1 + \frac{2e\phi_p}{m_e u_{e,\infty}^2} \right) \tag{4.26}$$

And the electron collection cross section is then:

$$\sigma_e \equiv \pi p_g^2 = \pi r_p^2 \left(1 + \frac{2e\phi_p}{m_e u_{e,\infty}} \right) \tag{4.27}$$

Note that the cross section is always strictly smaller than the ram surface area of the probe for negative probe potential and particles streaming towards the probe. Naturally electrons are screened from the probe for negative potentials. Coming back to equation (4.21), we have to replace the ram-surface of the probe with a cross-section. And we will introduce in the place of a single velocity an isotropic velocity distribution:

$$dI_e = A_p q u dn = -\sigma_e j dj = -ne\sigma_{e,\infty} f(u_e) du_e \tag{4.28}$$

The electron current can now be integrated through a velocity distribution $f(u_e)$, where we choose to model the plasma with an isotropic Maxwellian distribution of electrons, given as:

$$f(u_e) = 4\pi u_e^2 \left(\frac{m_e}{2\pi k_B T_e} \right)^{3/2} \exp\left(-\frac{1/2 m_e u_e^2}{k_B T_e} \right) \tag{4.29}$$

The isotropic Maxwellian distribution is a reasonable choice for a model, however it should be noted that space plasmas often have electron distributions which are non-thermal. Distributions that instead of generating Maxwellian distributions, rather display power-law distributions (Leubner, 2004). By assuming the

Maxwellian distribution the probe current can be found by integrating (4.29), over du_e :

$$I_e = -4\pi^2 r_p^2 n_e e \left(\frac{m_e}{2\pi k_B T_e}\right)^{3/2} \int_l^\infty \left(1 + \frac{2e\phi_p}{m_e u_e^2}\right) u_e^3 \exp\left(-\frac{1/2 m_e u_e^2}{k_B T_e}\right) du_e \tag{4.30}$$

The lower velocity bound of the integral is limited by the electrostatic potential of the probe $-e\phi_p$. Only electrons with kinetic energy $m_e u_e^2/2 > -e\phi$ are collected by the probe. The lower bound for the velocity distribution integral is then given by $l = \sqrt{-2e\phi_p/m_e}$. We split the integral in two parts, and solve them using standard integral identities:

$$\begin{aligned} I_e = -4\pi^2 r_p^2 n_e e \left(\frac{m_e}{2\pi k_B T_e}\right)^{3/2} \Bigg[\int_l^\infty u_e^3 \exp\left(-\frac{1/2 m_e v_e^2}{k_B T_e}\right) du_e + \\ \int_l^\infty \frac{2e\phi_p}{m_e} u_e \exp\left(-\frac{1/2 m_e u_e^2}{k_B T_e}\right) du_e \Bigg] \end{aligned} \tag{4.31}$$

This integral can be evaluated using these identities $\int_l^\infty u e^{-AU^2} du = 1/(2a) e^{al^2}$, and $\int_l^\infty u^3 \exp(-AU^2) du = 1/2a^2(1 - al^2) e^{-al^2}$ found in standard mathematics formula collections such as (Rottmann, 1960):

$$I_e = -\pi r_p^2 n_e e \sqrt{\frac{8 k_B T_e}{\pi m_e}} \exp\left(\frac{e\phi_p}{k T_e}\right) \tag{4.32}$$

In order to clean up the notation we collect the purely thermal motion current:

$$I_{e0} = \frac{1}{4} A n e \sqrt{K_B T_e/(2\pi m_e)}, \tag{4.33}$$

where $A = 4\pi r_p^2$ is the surface area of the spherical probe. The electron contribution to the IV-characteristic of the probe is then given as a function of the probe potential:

$$I_e = I_{e0} \exp\left(\frac{e\phi_p}{k_B T_e}\right), \qquad \text{for } e\phi_p < \phi_s \tag{4.34}$$

In the negative potential regime the corresponding electron current is weak compared to the positive regime, this is because the thermal electron flux to the probe decreases by a Boltzmann factor $\exp(e\phi_p/k_B T_e)$. Through similar calculations L. Schott (Lochte-Holtgreven and Richter, 1968), attains the probe electron current for positive potentials:

$$I_e = \begin{cases} -I_{e0} \exp\left(\frac{e\phi_p}{k_B T_e}\right), & \text{for } e\phi_p < \phi_s \\ -I_{e0} \left(1 + \frac{e\phi_p}{k_B T_e}\right), & \text{for } e\phi_p > \phi_s \end{cases} \tag{4.35}$$

Note that by using the same approach for ions, and inserting the proper charge and potential, we can quickly attain the ion currents as well.:

$$I_i = \begin{cases} I_{i0} \exp\left(-\frac{e\phi_p}{k_B T_i}\right), & \text{for } e\phi_p > \phi_s \\ I_{i0}\left(1 - \frac{e\phi_p}{k_B T_i}\right), & \text{for } e\phi_p < \phi_s \end{cases} \tag{4.36}$$

Where $I_i = \frac{1}{4} A n e \sqrt{K_B T_e/(2\pi m_e)}$, however we shall adapt a streaming ion model for the plasma model used in this thesis.

The streaming ion current

For the ion current we can adapt a model in which we consider ions to be "cold", by which we mean that the ion drift velocity is much larger than the ion thermal velocity

$$u_i \gg u_{i,th}$$

. Which is the same model that is used in (Yang et al., 2016), and in (Eriksson et al., 2017).

The drift velocity is u_i, and $u_{i,th}$ is the thermal velocity. The outline for the derivation of the ion current I_i is similar to the previously derived electron current of equation (4.28), and yields:

$$dI_i = -\pi r_p^2 n_i e v_{i,th} f(u) dv = -A_i n_i e v_{i,th} f(u) du, \tag{4.37}$$

where $A_i = \pi r_p^2$ is the projected ram surface of the spherical probe to the streaming ions. Adapting a cold ion drifting model means that the assumption of isotropic velocity distribution utilized in the derivation of the electron saturation current does not hold. Instead, we may use a drift velocity function for high streaming velocities (Whipple, 1981):

$$f_s(u) = \sqrt{\frac{\pi}{4}} u \left[\left(1 + \frac{1}{2u^2} - \frac{e\phi_p}{k_B T_i u^2}\right) \text{erf}(u) + \frac{1}{\sqrt{\pi} u} e^{-u^2} \right] \tag{4.38}$$

Where erf(u) is the error function with $u = \frac{u_i}{\sqrt{k_B T_i/m_i}}$. In the cold ion, high streaming velocity limit, the ion current is then:

$$I_i = \begin{cases} -I_{i0}\left(1 - \frac{e\phi_p}{E_i}\right) & \text{for } e\phi_p < E_i \\ 0, & \text{for } e\phi_p > E_i \end{cases} \tag{4.39}$$

Where E_i is the kinetic energy of the cold drifting ions. At positive potentials larger than the kinetic energy of the monoenergetic ions there can be no probe current, because the particle stream is not energetic enough to overcome the potential barrier set up by the probe.

Photoemission

In 1905 Albert Einstein wrote four revolutionary articles, one of these postulated the photoelectric effect, which contributed to his Nobel prize awarded in 1921 (Einstein, 1905). The photoelectric effect is closely related to what we study in this thesis, it is the direct cause of photoemission current. When the sunlight shines upon the probe, electrons are dislodged from the surface. Einsteins great insight was realizing that the energy of photons was carried in discrete quantized packets, and thus would only create photoelectrons when these packets was above a critical frequency. Because the energy of photons are connected to frequency through the Planck relation $E = hf$, where h is the Planck constant and f is the frequency. The energy needed to dislodge an electron is described by the work function $W = -e\phi - E_f$, where $-e\phi$ is the electron charge times the electrostatic potential just above the surface material, and E_f is the Fermi level or the chemical potential for electrons. It is a chemical property of the surface material, specifically it is the thermodynamic work required to add one electron to the object (Kittel, 1967).

If the probe is negatively biased, the photoelectrons will be accelerated away from the probe and constitute a net flux of charge away from the probe, in other words a photoelectric current. On the other hand, when the probe is positively biased the photoelectrons must overcome that potential barrier to escape, if not the electrons will be recollected, subsequently there is no photoelectric current. The photoelectric sheath was studied by Réjean J.L. Grard using laboratory data in (Grard, 1973). He made solid characterizations of probes in different environments. We base our model of the photoemission current on a modern interpretation of Grard:

$$I_{ph} = \begin{cases} -I_{ph0} & \text{for } e\Phi_p < 0 \\ -I_{ph0}\left(1 + \frac{e\phi_p}{k0_B T_{ph}}\right)^{\mu} \exp\left(-\frac{e\phi_p}{k0_B T_{ph}}\right) & \text{for } e\Phi_p > 0 \end{cases} \tag{4.40}$$

Where $I_{ph0} = A_{ph} j_{pho}$ is the photoemission saturation current. A_{ph} is the projected surface of the probe facing the sun. And j_{ph0} is the photoemission current density to the probe, which depends on the solar EUV spectrum as well as the surface coating of the probe. The model assumes a Boltzmann energy distribution, T_{ph} is the temperature of emitted photoelectrons. In equation (4.40) the μ exponent is a factor that depends on how we interpret the angular distribution of the emitted photoelectrons. For purely radial distribution $\mu = 1$ and for purely isotropic $\mu = 0$.

A Short Summary of Theory

In this chapter, we have presented a theoretical vantage point relevant to the application of the RPC-LAP instrument on Rosetta. We started by looking at the probe characteristic in a general view, we discussed the different regions of the characteristics and some basics of the importance of saturation currents. We briefly touched upon the significance of the floating potential and how we could interpret the junction between the transition region and the electron saturation region of the current-voltage characteristic. In preparation of investigating probe current densities we looked closer at a general problem of immersing an object into space plasma. We looked at the Debye shielding length and plasma sheaths, before using the Bohm criterion and the Child-Langmuir law to develop a general formula for the probe current density. We then applied our experience to the integration of OML-theory into our model, at this point we have a solid foundation for examining the sweep currents given to us by the RPC-LAP instrument. However, in this thesis the main focus is on the photoemission currents associated with the characteristics, this is explored in section 4.3. In the end we tie up a few loose ends with a short review of the photoelectric sheath and secondary emission. This theoretical introduction can serve as a foundation for the data analysis and method development presented in chapter 5.

Note that we have not considered other probe currents such as secondary electron and ion emission currents. We will discuss other sources of probe current in the coming chapters where relevant.

Chapter 5

Multiple Sweeps Method & its Analysis

Any Langmuir probe is sensitive to illumination. Photons hitting the probe excite electrons due to the photoelectric effect, in the ion saturation region of the current-voltage characteristic none of these electrons are reabsorbed. Instead they are accelerated away from the probe to induce a photoemission current. Electrons moving away from the probe constitute a current which is registered by the probe and can be seen as a fixed negative shift of the current-voltage sweep, see figure 5.1. This method for measuring photoemission current can be used locally aboard a spacecraft.

In this chapter I will present the multiple sweeps method for extracting the photoemission current. This multiple sweeps method builds on the method that we previously introduced by Johansson et al. (2017).

At the end of this chapter I will highlight the main difference with respect to the old method.

The photoemission current is typically of the order [nA] for RPC-LAP in the solar wind (Johansson et al., 2017). Given certain conditions the photoemission current can be of the order, or even greater than the ion current at negative bias in the ion saturation region of the IV-characteristic. Surely it is of interest to investigate how this current behaves with respect to probe performance. In this chapter I outline how this photoemission current can be extracted and handled in a suitable way, or studied as a proxy to the solar EUV-flux.

Analysis Method

Let's take a closer look at the main contributions to the IV-characteristics. From the theory section we have derived equations for probe currents I_i in equation

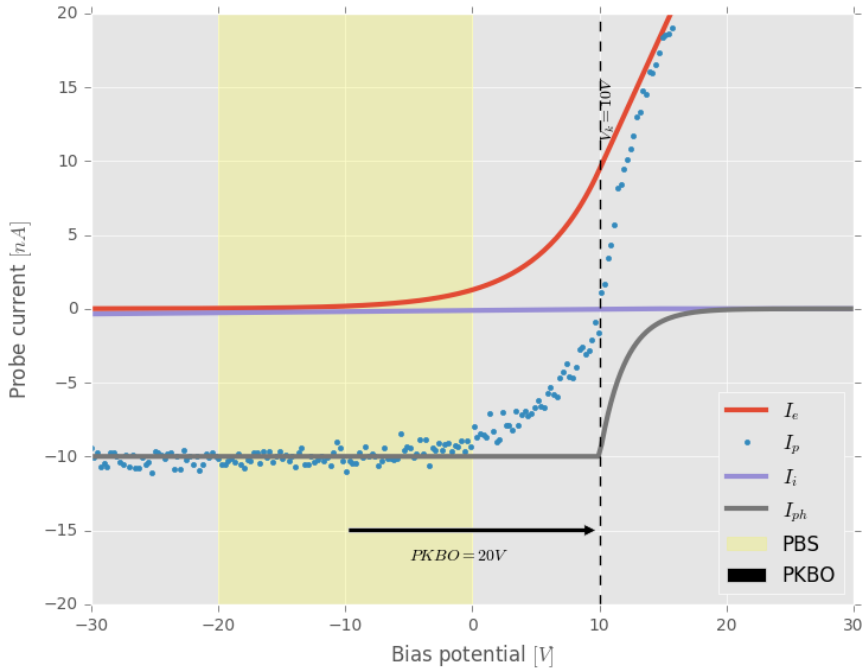


Figure 5.1: A synthetic probe IV-characteristic. The blue dots are a synthetic probe current, and in violet, red and grey lines are the different model currents. The potential bias span (PBS) is marked with a yellow shade, and the potential knee bias offset is marked with a black arrow. The sweep fitting step of the method applies a linear fit to the PBS portion of the characteristic.

4.39, I_e in equation 4.35 and I_{ph} in equation 4.40, which are the ion saturation, electron saturation and photoemission currents respectively. Since we are working in the ion saturation region of the IV-characteristics with negative bias, the total probe currents reduce to the following equations. For the ion current we have:

$$I_i = -I_{i0}\left(1 - \frac{eV_P}{E_i}\right) \tag{5.1}$$

Where $I_{i0} = nueA_i$, and V_p is the probe potential. For the electron response we have:

$$I_e = I_{e0}exp\left(\frac{eV_P}{KT_e}\right), \tag{5.2}$$

where $I_{e0} = A_e ne\sqrt{k_B T_e 2\pi m_e}$, notice the exponential decrease with increasingly negative potential bias.

The last simplified equation is the photoemission current:

$$I_{ph} = -I_{ph0} \approx -10nA \tag{5.3}$$

These equations lay the foundation for the estimation of photoemission current using the multiple sweeps method.

Theoretical approach

In this section I will outline a theoretical approach to the ideal performance of the multiple sweeps method. Let us begin by analyzing the total of the simplified currents in idealized space plasma, valid for the ion saturation region (see figure 4.1) of the IV-characteristics. This is the total probe probe current I_p, accumulated by the Langmuir probe instrument:

$$I_p = I_i + I_e + I_{ph} \tag{5.4}$$

Assuming that there is no secondary emission current and that the plasma is comprised of one species, we can further simplify the total current to only contain the ion saturation current and the photoemission current. This is achieved by analyzing the total probe current in a part of the ion saturation region that has sufficiently negative probe potential so that $eV_p/k_BT_e \ll -1$, which means that the electron probe current is negligible. A negligible electron probe current is ensured by the potential knee bias offset (PKBO) parameter from figure 5.1. Presently the probe current reduces to:

$$I_p = I_i + I_{ph} \tag{5.5}$$

Assuming that the photoemission current is stationary with respect to variations in the ion saturation current I_i during a sweep, we can further reduce the number of variables, and in effect isolate the remaining two variables. We apply a linear fit to the potential bias span region from figure 5.1, to get the slope gradient of the ion saturation region which is dominated by the total probe current from equation 5.5. We already assumed that the photoemission current is stationary on the timescale of a sweep, the slope gradient is therefore detached from the photoemission current.

The next step of the method relies on the assumption that when there equation 5.5 is valid, and $I_i = 0$, then $I_p = I_{ph}$, however the ion current can only ever be zero for zero plasma density. The solution is to use multiple sweeps to extrapolate the ion saturation current to zero. We interpret any residual probe current for zero ion saturation current as the photoemission current.:

$$dI_p(n = 0)/dV_B = \beta_1 I_p + \beta_0 = \beta_1 I_{ph} + \beta_0 = 0 \tag{5.6}$$

Where β_0 and β_1 are regression coefficients. Solving for $I_i = I_{ph}$ we get an expression for the photoemission current:

$$I_{ph} = -\frac{\beta_1}{\beta_0} \tag{5.7}$$

Given β_1 and β_0 from a slope linear extrapolation to zero, we can extract the photoemission current. This method is critically dependent on the slope extrapolation, in those cases where the extrapolation can not be reliably done, the method may break down.

Limitations

The method is inherently linked to the equations derived in sections 4.2.2, 4.2.3 and 4.3, and therefore has the same theoretical constraints. In this section, I will outline a few additional limitations from implementing the method. Having an idea of the limitations of the method is important when trying to optimize the performance of the analysis.

Ideally the sweep should be constant throughout the sweep for $V_p < V_k$, if it changes drastically it will challenge the validity of the model used for the ion saturation current. Now let's take a step back and investigate the derivative of the probe current of how the total probe current behaves. Inserting equation 4.39 for negative potential, and equation 5.7 into 5.6, we have:

$$\frac{dI_p}{dV_B} = \beta_1(I_i + I_{ph}) + \beta_0 = \beta_1 \left[-eA_s nu \left(1 - \frac{qV_P}{E_i} \right) - \frac{\beta_0}{\beta_1} \right] + \beta_0 \tag{5.8}$$

Solving for β_1, which is the slope gradient in the ion saturation region, for $V_p < V_k$:

$$\beta_1 = \frac{e}{eV_p - E_i} \tag{5.9}$$

The method is sensitive to changes in β_1, presenting a restriction on our method even in the ideal case. It is clear that the method then is restricted by the variations in plasma density and ion velocity during sweeps:

$$\frac{dI_p}{dV_B} = \beta_1(I_p - I_{ph}) \propto \frac{n}{u_i} \tag{5.10}$$

High variation in plasma density is actually beneficial as it improves the spread in the data, resulting in a more accurate extrapolation. A high spread in the ion velocity on the other hand appears to be disadvantageous. The ion velocity restriction is explored by synthetic data in section 5.2.1. The accuracy

of the method is limited through the ion velocity of the energy term in equation 5.9, which means that we expect optimal performance when the fluctuations in the ion velocity of the plasma is as small as possible. At the same time we need multiple data points with some degree of variation to facilitate a precise extrapolation of $dI_p(n = 0)/dV_B$ to extract I_{Ph0}. A satisfactory spread of data points is generally achieved in space plasmas because the plasma density n tends to vary a lot on small timescales, while the velocity fluctuations does not. Density fluctuations are observed at 67P on the timescale of minutes to seconds (Henri et al., 2016). While the velocity has been reported to not vary extensively, it is believed that ion velocity fluctuations is a primary source of error (Vigren et al., 2016).

Inner workings of the analysis method

The method can be roughly divided into two parts. The first step is to apply a linear fit to the ion saturation region of the IV-characteristic of the sweep, I call this sweep fitting, see fig 5.1. The fitting region can be adjusted, but needs to be sufficiently negative to have negligible I_e contamination. The position as well as the length of the fitting region can be adjusted. A longer region is more statistically robust, but run the risk of picking up on the electron current contamination of the ion saturation region if it is too long.

Step two in the analysis is to collect multiple sweeps, and extrapolate to $dI_p(n = 0)d/V_b = 0$, this step is called slope extrapolation. The key concept is that when the slope is zero, the ion current contribution should also be zero. Since the electron current contribution is considered negligible in the ion saturation region of the IV-characteristic, then by the elimination method; any remaining current to the probe in this model must be photoemission current.

Sweep fitting

The analysis routine is given in the appendix, in section 12.1 and in section 12.2, and is written in the programming language Python. It is initialized by specifying a set of method specific parameters. The user must specify a potential knee bias offset (PKBO), and a potential bias span (PBS). These values are in Volt and specify the bias voltage offset from the knee into the ion saturation region, and the range of the fit centered at the specified bias voltage offset. Ideally one would use a very high PKBO so as to negate the electron current completely, but this is not be viable from a pragmatic perspective. A high PKBO will limit the available potential to use as the fitting range, since it is centered around the PKBO. We will have to characterize the method in order to obtain the optimal set of values to use in this method. The optimal settings will likely vary somewhat depending on the plasma environment, therefore I will try to ascertain a

pair of values which are suitable for the plasma conditions generally observed near and far from the comet, by using synthetic data.

Depending on the level of noise and the plasma parameters in play we may sometimes get negative slope values from the sweep fitting. Negative slopes is not a part of the probe current model and, will be discarded as noise artifacts in an effort to refine the analysis. This is partly because the model is restricted to positive slopes, however it also act as a filtering for the IV-characteristic response to secondary particle emission. Negative slopes may also be the result of a noise when applying the linear fit.

The slope extrapolation

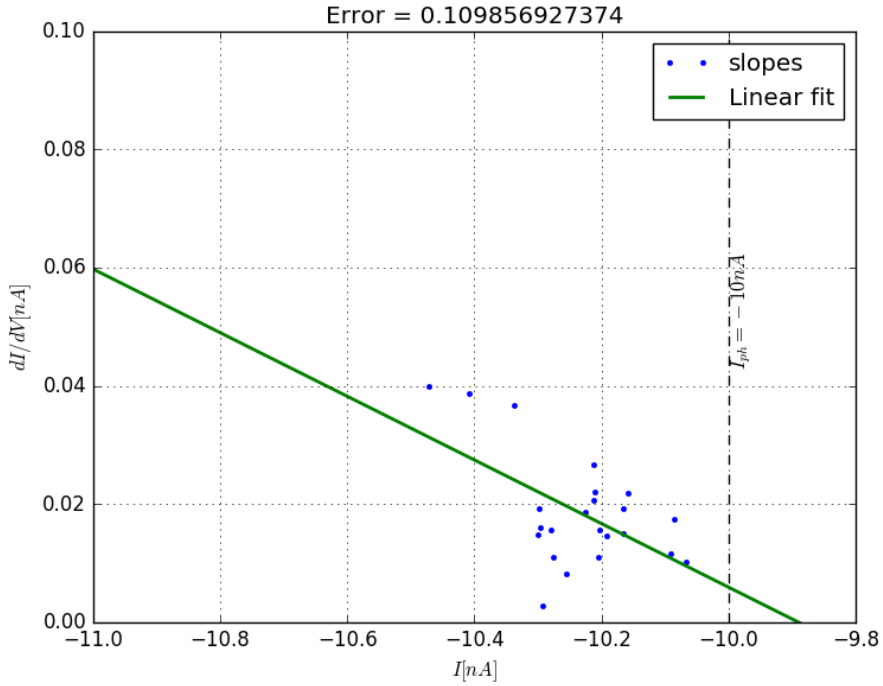


Figure 5.2: A slope extrapolation using the multiple sweeps method on synthetic data. The photoemission from the synthetic data is marked in a dotted line, while the extrapolation points are in blue dots. The extrapolation in green is found to be relatively close to the model photoemission.

The second part of the method is to extrapolate the sweep slopes to zero, see figure 5.2, which shows the method applied to synthetic data. We collect from multiple sweeps a suitable number of slopes. These slopes serves as extrapolation points for which we extrapolate to zero slope. The extrapolation points must

have some variance in order to provide some spread so that the extrapolation is possible. The plasma is constantly varying to some degree and naturally provide enough spread to facilitate the extrapolation. For the zero slope current, we reverse the process, and extract the probe current at zero slope. Our plasma model explain that for zero slope the only stationary probe current is the photoemission current.

Synthetic Data

With synthetic data, that is data which is generated using the previously derived model for the total probe current, we gain complete control of parameters, and can inspect the methods performance in a controlled manner. The synthetic data is formed with the full current model outlined in the theory section with equations, 4.35, 4.39 and 4.40. The instrument has some inherent noise which is of the order $0.5nA$. The comparison is done with the ideal theoretical limit of the model in mind, so there is no added noise in the approach. The current is modeled for $T_e = 5\text{eV}$, a potential bias $V_B \in [-30, 30]\text{V}$, while varying the ion velocity and plasma density.

From equation 5.4, we generate a synthetic probe current with ion, electron and photoemission components. The probe current model is then fed with the plasma parameters characterizing the particular plasma environment that we want to study, as an example we use the two regimes in table 5.1 as a basis for comparing performance in different plasma environments. After choosing a plasma parameter space the synthetic probe current is promptly modified with noise from a Gaussian distribution to simulate realistic conditions. These probe current are then treated as "real data" and fed through the same analysis routine, so that we can inspect the performance of all aspects of the method. The synthetic sweep in figure 5.1 is generated using this approach.

Parameters	Solar Wind	Comet Perihelion
n_i	$2cm^{-3}$	$500cm^{-3}$
Δn_i	$3cm^{-3}/N_s$	$2000cm^{-3}/N_s$
u_i	$350km/s$	$1km/s$
Δu_i	$50(km/s)/N_s$	$0.1(km/s)N_s$
T_e	$10eV$	$1eV$
PKBO	$20V$	$20V$
$Noise$	$0.5nA$	$0.5nA$
$Mean$	30	30

Table 5.1: Parameters used to produce figures 5.5 and 5.6. The parameter N_s is the number of sweeps.

The plasma is usually many times denser in the atmosphere of the comet as opposed to the solar wind. For this reason, probe performance with respect to density is therefore an interesting property to investigate. Another property worth investigating is the impact of electron temperature. In the ideal case the electron current is negligible for negative potential, we will therefore investigate the impact of the electron current when it is not strictly zero. The photoemission knee which approximately corresponds to the floating potential of the probe, is set to 10V for this analysis. The bias voltage is the same as the most common Langmuir operating modes, which is $[-30, 30]$V. The ion mass and velocity are also interesting properties, since they couple directly to ion energy; which from equation 5.9, clearly is paramount to the method.

When a suitable parameter space has been decided, we can go ahead and construct the model probe current using equations 4.35, 4.39 and 4.40. After creating a set of data of currents with the desired plasma parameters, we can work with the idealized case, either with or without electron current interference, or with different levels of noise. We estimate the noise levels to be approximately $0.5nA$ for the real data set.

When applying the method to synthetic data, the theoretical limitation of equation 5.9 is clear to see. In figure 5.3, I have traced the theory with solid lines and synthetic data with markers. The colors separate different ion velocities. While the markers are spread because of a linear variation in density for each sweep. Figure 5.3 showcases three ideal cases of fixed ion velocity while varying plasma density. We need a set of measurements under varying plasma density conditions to accurately extrapolate the photoemission current. Each of the three lines with fixed velocity give very good estimates of the photoemission, however, should we try to make one linear extrapolation within the same density span, but across the three different velocities, we would get very large errors. This is a confirmation of what we learned from equation 5.10.

Synthetic data compared to theory

At first glance the model behaves as expected in the ideal limit, we see can recapture the photoemission level that was put into the model with a high level of precision. The essential difference between the synthetic data and the theory in figure 5.3 is that only the synthetic data contains the electron contribution. As we can see there are some slight deviations because of this discrepancy especially noticeable for high densities. This electron current influence may lead to overestimation of the slope, and subsequently impact the I_{ph} estimation.

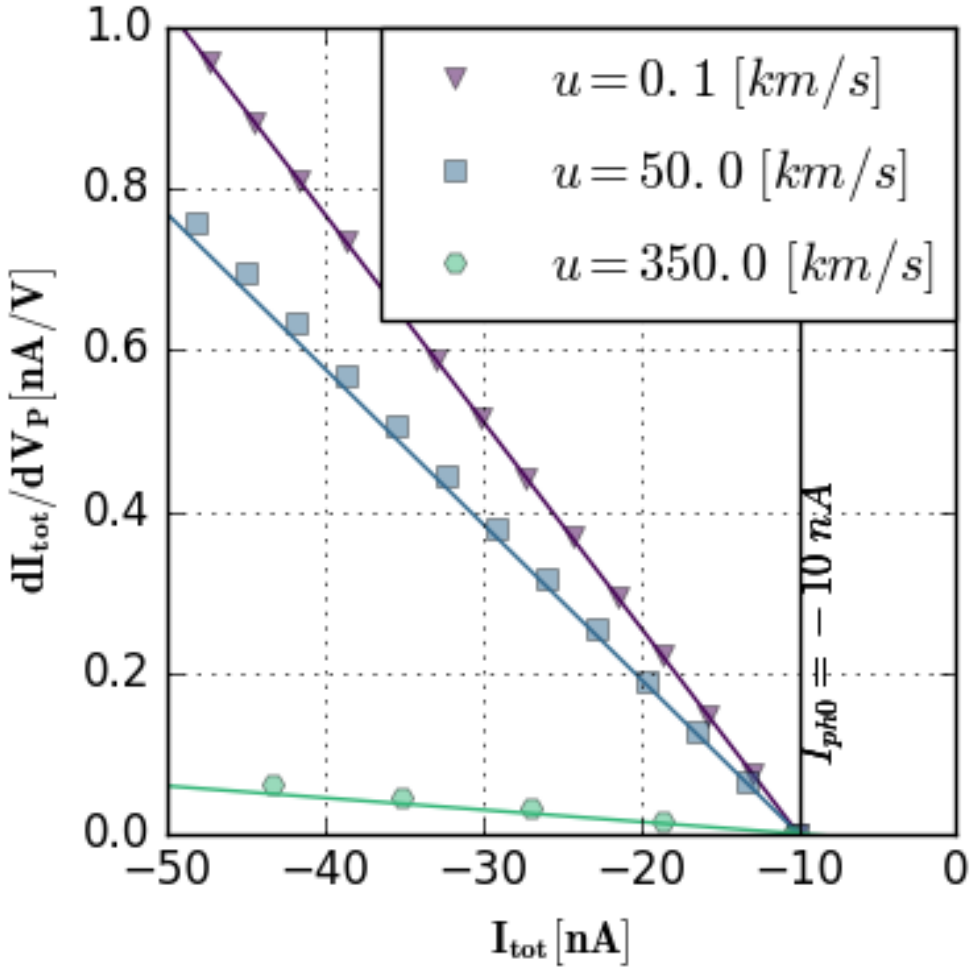


Figure 5.3: Slope dI_{tot}/dV_p versus I_{tot}. Triangle, square, and circle points are taken from the synthetic data sweeps, each with varying n but constant u. The solid lines are calculated from Equation 5.9 with u corresponding to the synthetic data. There is a good agreement between the synthetic data and Equation 5.9. Small deviations, especially for $u = 50$ km/s, are related to high plasma densities, where I_e becomes more significant.

Choosing Method Input Parameters

Before applying this method to real data, a few parameters must be determined. These parameters decide the resolution of the method, as well as statistical robustness, so there must be a trade off.

There are in essence three parameters that the user must decide, the potential knee bias offset (PKBO), the potential bias span (PBS) and the number of sweeps. Let us repeat what these parameters controls. The PKBO control which part of the IV-characteristics we use as the basis of the method. We want to choose a value, which guarantees that we are firmly within the ion saturation region, in effect that we are far enough from the potential knee to achieve satisfactory electron shielding. Ideally we would like to set the PKBO at infinite negative bias and subsequently have perfect electron current shielding. Completely exhaustive electron shielding is not feasible, we must therefore limit ourselves to at most PKBO = 30V. For most cases something like PKBO = 20V should be sufficient, see figure 5.4.

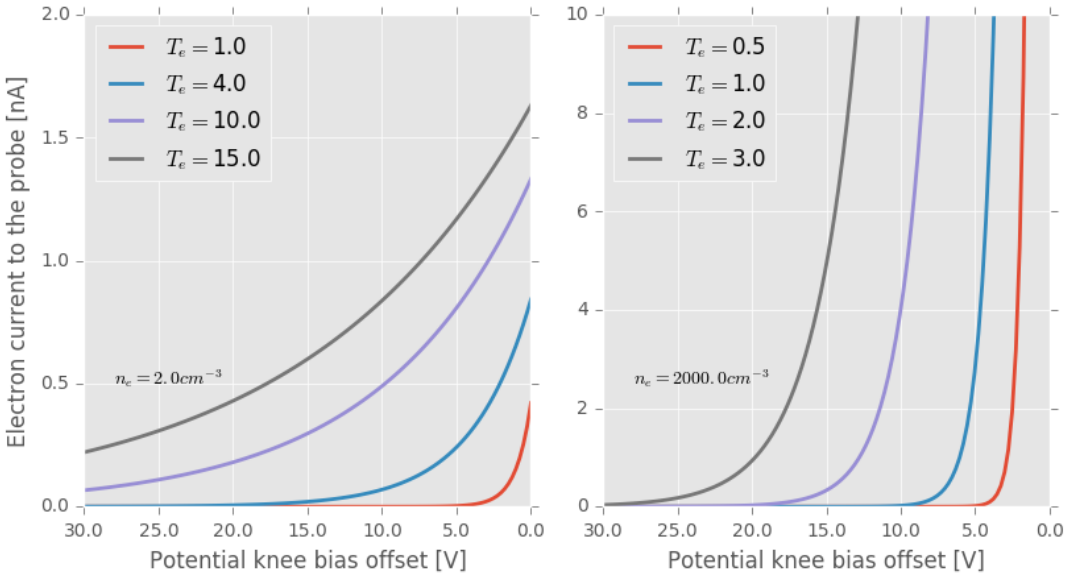


Figure 5.4: The electron response for the two regimes, at different temperature levels

The second parameter, which is the potential bias span (PBS) decides the range, in volts, of the linear fit applied to the probe current in the saturation region of the sweep, see figure 5.1. The potential bias span decides how many datapoints we take from the sweep fitting step to the slope extrapolation. For a typical operating mode the range is from -30 to 30 Volt with 240 samples. At 240/40V = 4 samples per volt. A larger range means more sample points for the ion saturation, but also increased contamination from the electron current response. This suggests that there is a trade off between the theoretically strongest performance and statistical robustness.

Parameters	Solar Wind	Cometary environment
n_i	$[1-5]cm^{-3}$	$[10-4000]cm^{-3}$
u_i	$350km/s$	$[1-5]km/s$
T_e	$10eV$	$[1-2]eV$

Table 5.2: A table of some typical plasma parameter in the two slow solar wind and the cometary atmosphere.

The last factor is the number of sweeps to consider for each extrapolation, this parameter essentially controls the resolution of the method. Each sweep is about 4 seconds, but with varying sweep cadence between sweeps, from 32s in burst mode, up to at most 288s. The cadence information is part of the mode settings, an added benefit of lower cadence is that it could be easier to discard outlier I_{ph} estimates and with lesser impact on the final result. I have implemented three resolution for photoemission analysis using the multiple sweeps method. The cadence and implementation of the resolution of the method will be further elaborated in chapter 6.1.1.

General characterization of the multiple sweeps method for two space plasma regimes

Characterization can be done separately for plasma conditions which are vastly different, such as the solar wind and the cometary environment. In this sectioncI will characterize the method as extensively as possible for two regimes, once for the solar wind and again for the dense cometary atmosphere of perihelion. The most important approximate plasma parameters for the two regions, are given in table 5.2. I have used similar values as in (Eriksson et al., 2017).

I will try to find a pair of input parameters PKBO and PBS that is a suitable working set of parameters that work reasonably well for most if not all of the mission. The biggest technical issue is operating modes that are otherwise suitable, but have a short bias span. One such as operating mode or "macro", is macro 506 with a range of [-12V, 12V]. This macro will have to be handled separately, at a later stage. At this point the goal is to find a set of input parameters that is suitable for the vast majority of the mission and for the preferred science macros.

A short PBS reduce error due electron current contamination of the total probe current, but will also reduce the data for the slope linear fit. It is my belief that a stronger fit is of higher importance, due to the methods sensitivity to the fit. An exception can be at times where the plasma provide a strong electron current response.

I have taken representative values for the solar wind and cometary environment, listed in table 5.1. I will use the models and techniques previously derived, to create a synthetic dataset, which we use to test the method for different input parameters. The model electron temperature and density influence on the total probe current as a function of PKBO is given in figure 5.4. A large PKBO give small electron current contamination to the total electron current, remember that the method assume negligible electron current. As a starting point for the characterization, let us use PKBO = 20V, which gives very good electron shielding for the solar wind and reasonable shielding for the cometary environment, while simultaneously leaving room for varying the PBS.

With the synthetic data set using values from table 5.1, and a bootstrap PKBO as the starting point for an investigation we can test effect of varying PBS range on error. I have tested increments of 1V of the PBS range from 5V to 30V, for each increment 9 and 25 sweeps has been created. The multiple sweeps method has been applied to the collection of sweeps, from the method and I_{ph} estimate has been made. The I_{ph} estimate has been compared to the theoretical I_{ph} input to the synthetic data and the difference is taken as the error of the method. For each PBS increment, 30 I_{ph} measurements have been made and an average error is computed. These data are presented in figure 5.5 for the solar wind parameters, and in figure 5.6 for the cometary environment. Using these figures as a guideline we can choose a suitable PBS parameter for the method. In the following application to the real dataset I have used PKBO = 20 and PBS= 10 as the standard model parameter input.

In figure 5.5 and figure 5.6, I present averaged result of running the analysis method on the two regimes, with different combinations of sweeps and PBS. The solar wind regime is apparently more well behaved than the cometary environment. For low PBS, we can observe that the method suffer from a poorly conditioned extrapolation due to too few samples. For large PBS however we see that the method has some slight increase in error due to increased electron current response influence. If we were to increase the PBS even further to say 50, or 60 Volt, the method would be expected to break down completely. These results suggest that in the case where all data is available, it is possible to achieve at most a time resolution 15-20 minutes for the solar wind case, and 20-30 minutes for the cometary environment.

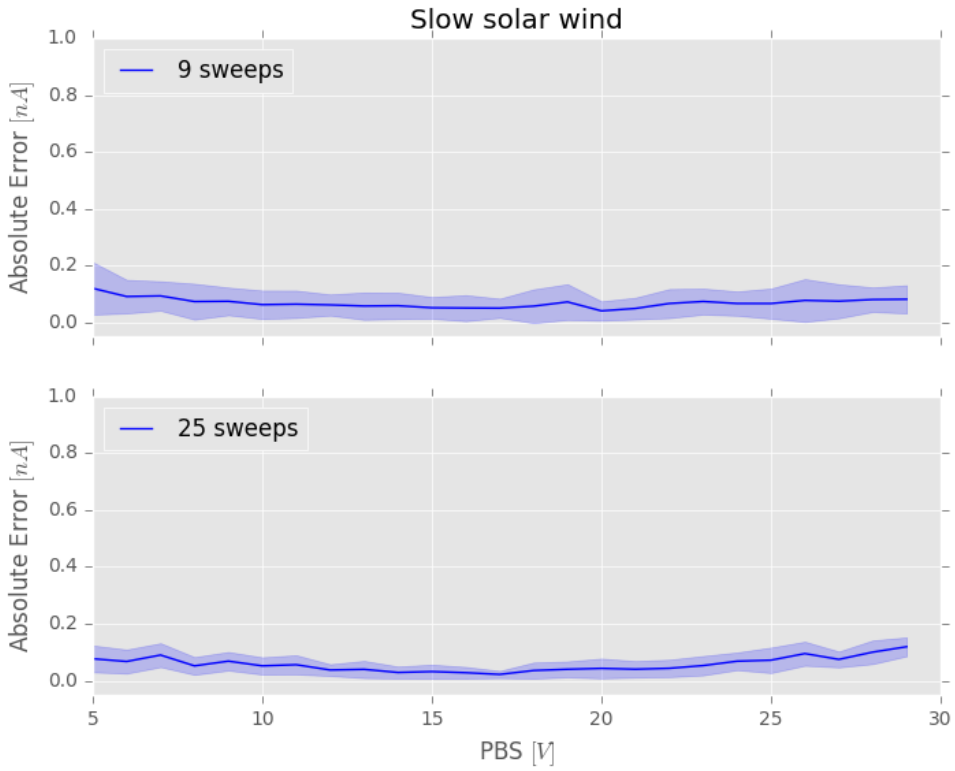


Figure 5.5: Estimated average error versus potential bias span (PBS) and number of sweeps, averaged over 25 runs of the variables in table 5.1 for the slow cometary environment. This figure provides a means to choose a reasonable PBS value, for general cometary environment conditions. We also see the statistical effect of taking more sweeps.

The Multiple Sweeps Method Error Analysis for Two Space Plasma Regimes

Taking variables from observations done by RPCLAP and applying them to the model used so far, we can properly test the method throughout a reasonable parameter space. The parameter space used in this analysis is spanned by the two governing variables n and u_i, we use the values from table 5.2. We already came to the understanding that the change in ion velocity during sweeps is important from equation 5.9, we will also run this test for two different ion velocity steps between individual sweeps Δu_i, 116m/s and 38m/s. The electron temperature is set at $T = 5$eV throughout.

The density range $n \in [0, 4000]\text{cm}^{-3}$ is divided into 100 equispaced grids, while the ion velocity is divided into 25 and 5 equispaced intervals. The different

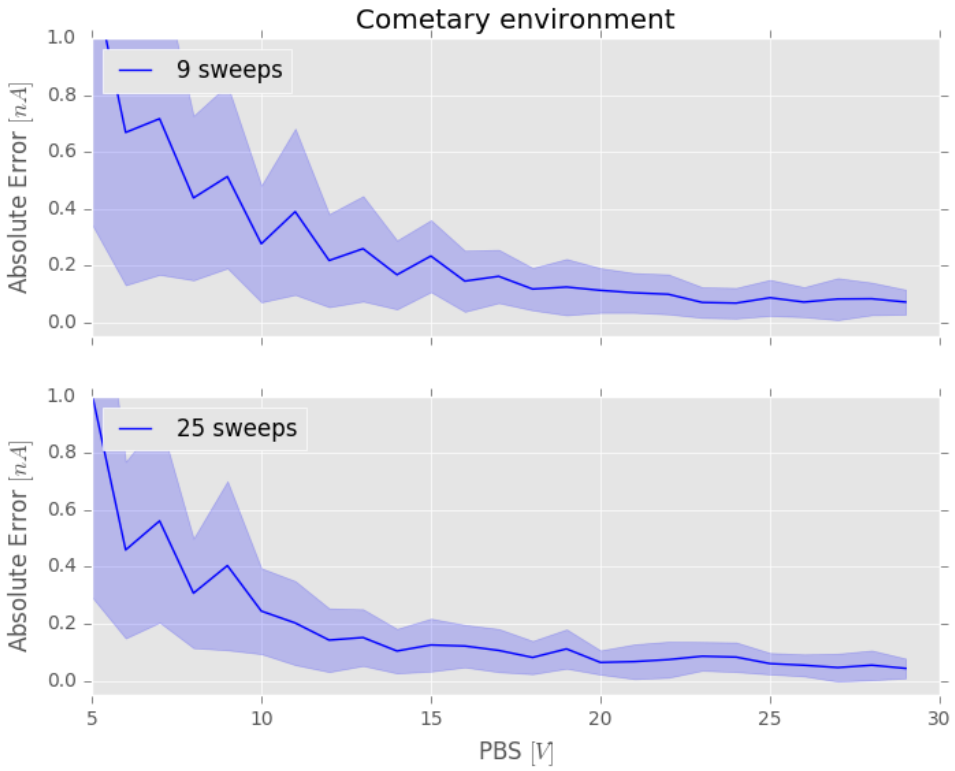


Figure 5.6: Estimated error versus potential bias span (PBS) and number of sweeps, averaged over 25 runs of the variables in table 5.1. This figure provides a means to choose a reasonable PBS value, for general solar wind conditions. We also see the statistical effect of taking more sweeps. A single sweep is roughly four seconds long, but a couple of minutes apart meaning that the resolution with 30 sweeps is in the region of 1 hour.

sized intervals of the velocity ensure that we have two different velocity steps between sweeps. From figure 5.7, we see that the larger ion velocity step, i.e the 5×100 grid, produces the largest error. The fine grid 3×3 has equispaced values of density and velocity in the square and give nine current-voltage characteristics, one for each combination of values. These nine values are then fed to the multiple sweeps method and the resulting I_{ph} estimate is compared with the input value of $I_{ph,input} = -10\text{nA}$.

The main result from the error analysis of the cometary environment can be found in figure 5.7, at first glance we see that the erroneous data are concentrated in the high density, low velocity regime. The first row, has moderately changing ion velocity, $\Delta u_i = 117\text{m/s}$, while the second row has a smaller $\Delta u_i = 38\text{m/s}$, between individual sweeps. It is reassuring that the row with the lowest Δu_i also

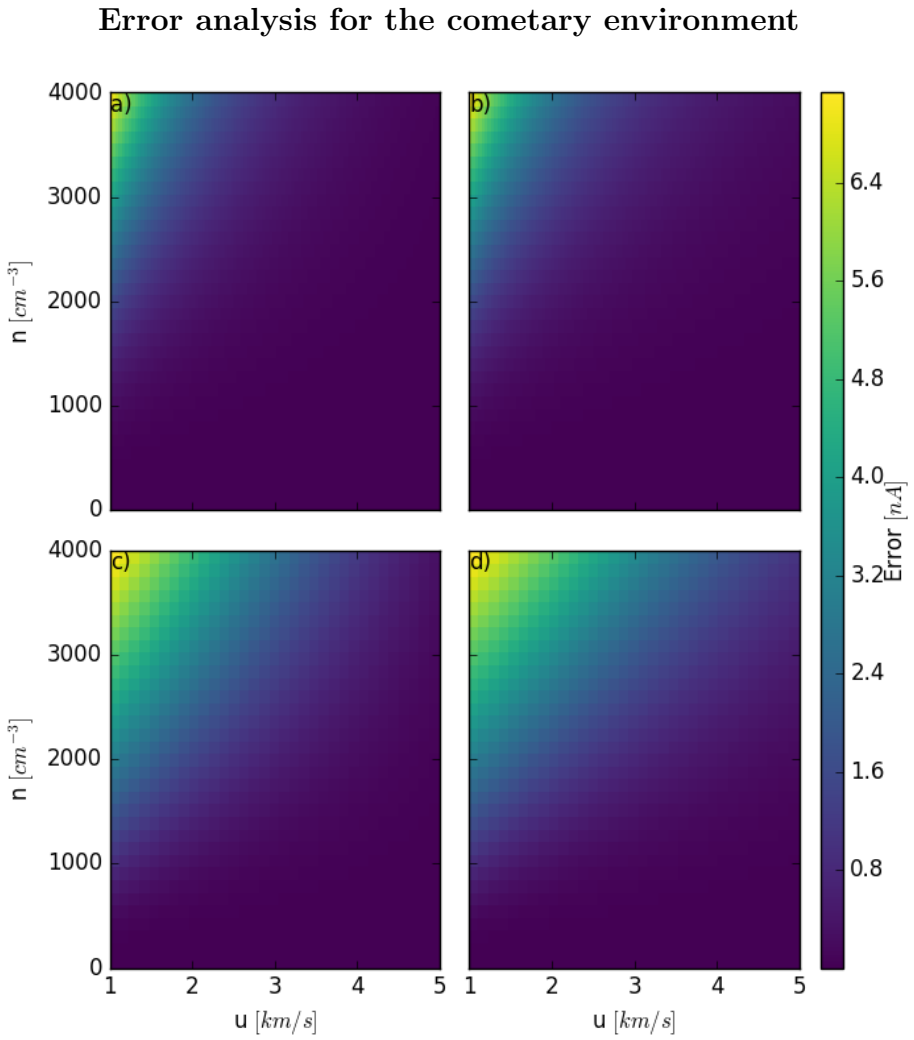


Figure 5.7: The error in I_{ph0} for two different levels of velocity variation Δu_i and two different ranges potential knee bias offsets. The PKBO is varied for each column so that panels a) and c) have PKBO= 10 V. Panels b) and d) have PKBO= 15 V. Panels a) and b) has a velocity variation of $\Delta u \simeq$ 117 m/s between the sweeps used for each grid square. For panels c) and d) $\Delta u \simeq 38$ m/s. Modeled at spacecraft potential $V_{sc} = 10$V, for the cometary environment.

exhibits the smallest errors. We know that I_e decay exponentially for negative biases, it is the influence from this current that we observe in the upper left corners of each plot, which is for high density and low ion velocity. The electron

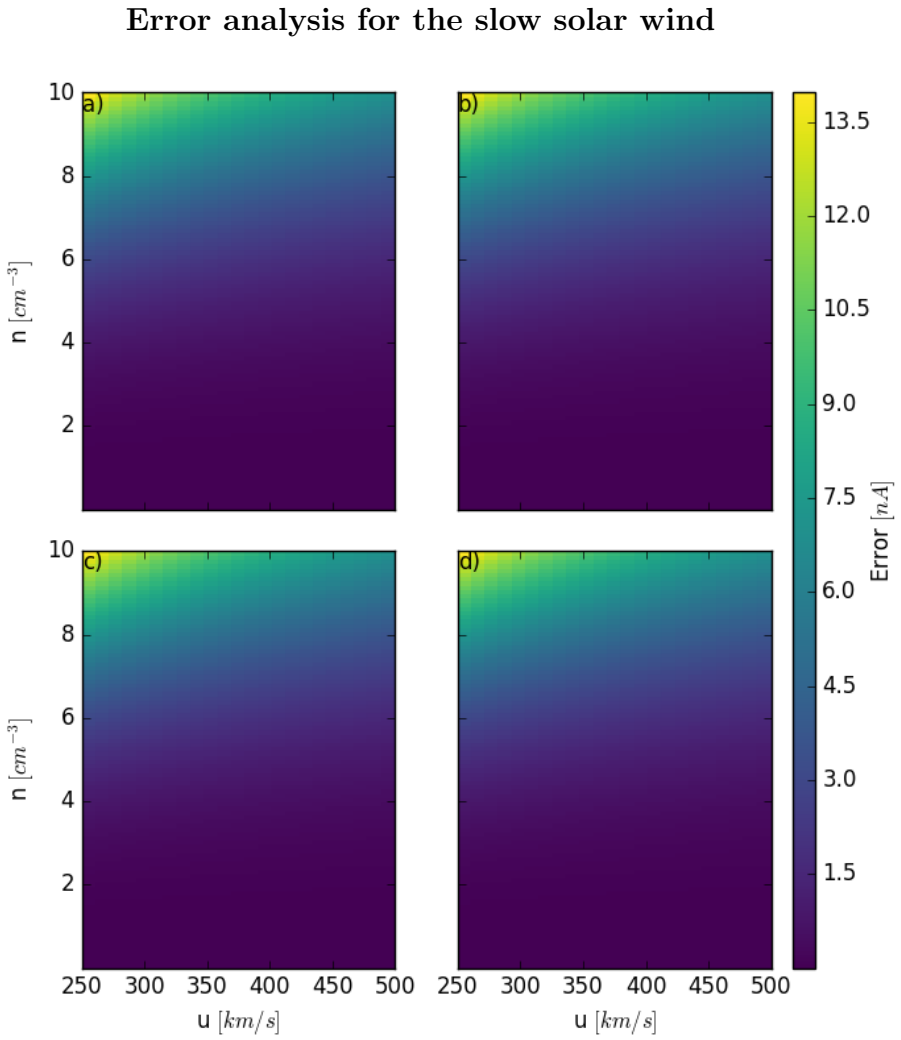


Figure 5.8: The error in I_{ph0} for two different levels of velocity variation Δu_i and two different potential knee bias offsets. The PKBO is varied for each column so that panels a) and c) have PKBO= 20 V. Panels b) and d) has PKBO= 10 V. Panels a) and b) has a velocity variation of $\Delta u \simeq$ 10 km/s between the sweeps used for each grid square. For panels c) and d) $\Delta u \simeq 50$ km/s. Modeled at the spacecraft potential $V_{sc} = 10$V, with plasma parameters similar to that of the slow solar wind.

current influence is expected, considering that the ion current contribution to the sweep is low for low ion velocities, which means that the probe current is more sensitive to the electron current response for low u_i. All in all, the method seems

to be performing well in the ideal limit. There seems to be a level of robustness to the method, which has proven that it can operate across a large plasma parameter space with synthetic data. Density values approaching 4000cm^{-3} are not common, but can happen in bursts in a cometary environment, and may affect the method significantly.

The other input parameter that connects to statistic robustness is the number of sweeps gathered for the slope extrapolation step. Again we need a reasonable number of sweeps to reliably extrapolate a photoemission current from the sweeps. A single sweep is taken in a matter of seconds, but there are a few minutes between each sweep, in general one sweep is taken every other minute. This means that for a reliable result we will need to determine how well the method performs given these input parameters. I have explored different input parameters in the two different regimes (table 5.2), to investigate any apparent differences between the pair. From figures 5.5, and figure 5.6 we see the difference between taking 9 and 25 sweeps for each I_{ph}, it does not seem to make a great difference. It might be because the method is robust, or that the synthetic sweeps are very well behaved. When we apply the multiple sweeps method to the RPC dataset we will test with different number of sweeps.

Chapter 6

Technical Details

In this chapter I present specific properties of the RPC-LAP instrumentation and operation that is central to the method in this thesis. The RPC instrument has already been briefly introduced in section 3.2, now let us focus on the RPC-LAP. In this section we will look at the different science modes and operational modes available. We will also discuss some instrumentation issues that will make the presentation of the method straightforward and more coherent.

Macros

The RPC-LAP instrument operational modes or "macros", control all aspects of the instrument. These are command sequences which can order the instrument to change bias, sweeps, or even initiate on-board filtering of the data (Eriksson et al., 2007). The macros are useful keys that switch the operating mode of the instrument without the need for large up-link volumes, which would take up to much of the telemetry capacity. In table 6.1, some of the core characteristics of the preferred science macros for the multiple sweeps method. We want the bias span to be as large as possible, so that we can access all of the current-voltage characteristics regions. We want the step to be small, so that the sweep sampling rate is as high as possible. We want the cadence to be low to record as many sweeps as possible in as short time as possible. Ultimately the cadence between each sweep puts an upper limit to the resolution of the method, because the cadence determines the time delay between each sweep.

Macro cadence

Generally, one sweep takes 1-5 seconds to complete, with a cadence of about 30 seconds to 5 minutes meaning that the resolution for an I_{ph} estimate using 10 sweeps is in the region of 5-50 minutes. In practice we must take care of any missing data periods. The way that we have solved this problem in the script is

Table 6.1: The preferred science macros for the RPC-LAP multiple sweeps method. They have a large bias span, a high bias voltage sampling and low cadence.

Macro	Bias [V]	Step [V]	Cadence [s]
412	-30, 30	0.25	64
416	-30, 30	0.25	160
417	-30, 30	0.25	160
517	-30, 30	0.5	160
612	-30, 30	0.5	160
615	-30, 30	0.25	160
616	-30, 30	0.25	160
617	-30, 30	0.25	64
624	-30, 30	0.25	160
710	-28, 28	0.5	160
914	-28, 28	0.25	160

to look at a collection of sweeps which are equivalent to a chosen time resolution, and imposing a minimum amount of sweeps necessary to give a trustworthy I_{ph} estimate in this time period, see appendix for script details. For example, we may set a time resolution of 1 hour, which is roughly equivalent to 50 sweeps depending on macro settings, and impose a minimum number of valid sweeps. If this requirement of say, 30 sweeps in one hour, is not met, then the estimation is discarded, and we move on to the next time step. One could conceivably do it the other way around, and set a minimum amount of sweeps then collect until there is enough to meet the required statistical number of sweeps, and then apply the method. This method however, would not have equal time steps, and would sometimes collect sweeps over a very long time period. The resolutions that we have utilized in this thesis are given in table

Table 6.2: The different resolution levels utilized in the analysis.

Resolution	Min numb. of sweeps	Max numb. of sweeps
High	10	15
Medium	20	40
Low	30	50

Science Modes of the RPC-LAP

The instrument can operate in different science modes, which are tailored for specific tasks. Unfortunately, the two science modes can not operate at the same time. In addition, because the RPC is a suit of instruments designed to complement each other, sometimes the RPC-LAP lends one of its probes to RPC-MIP Trotignon et al. (2007). Operating in conjunction with RPC-MIP has been shown to disturb the Langmuir probe sweeps, consequently the macros that are used for operation with RPC-MIP are not usable for the multiple sweeps method.

Electric field mode

An important operating mode is the electric field mode, which measures the electric field between LAP1, and LAP2. In electric field mode the instrument alternates the bias to the probes between on and off, while the potentials between each probe is measured. The electric field is calculated by using $E = (V_{p1} - V_{p2})/d$ where, V_{p1} and V_{p2} are the respective probes and d is the distance between them. The instrument can operate in two modes, one long term continuous low frequency mode and a quick scan high frequency mode. The RPC-LAP instrument can, when operated in electric field mode, give estimates on the variation of the spacecraft potential V_{sc}. The electric field mode is not heavily utilized in this thesis, although effects of the spacecraft potential on the multiple sweeps method are discussed in section 8.1.1. A timeseries of the estimated spacecraft potential by Odelstad et al. (2016), are presented in figure 6.1.

Sweep mode

The main operating mode in this thesis is the sweep mode, which is the mode that produces IV-characteristics similar to that of figure 4.1. The multiple sweeps method for extracting the photoemission current is certainly subject to limitations, the most obvious being that it simply does not work for current-voltage sweeps that have specific shapes. Certain plasma regimes may have shapes that look nothing like the ideal shape presented in figure 4.1, the shape of the curve is related to previously stated limitations to the plasma parameters. Compare the ideal case of figure 4.1, with figure 6.2 that presents the actual measurements by RPC-LAP. The probe current measurements are the represented by green dots, while a red line has been drawn through the points used for the slope linear fit. These are two sets of sweeps that are recorded within the same day, using the same macro 624. Notice that there seem to be two distinct paths that the sweeps take. Within the same set of sweeps, that are in the next step used to extrapolate the photoemission current.

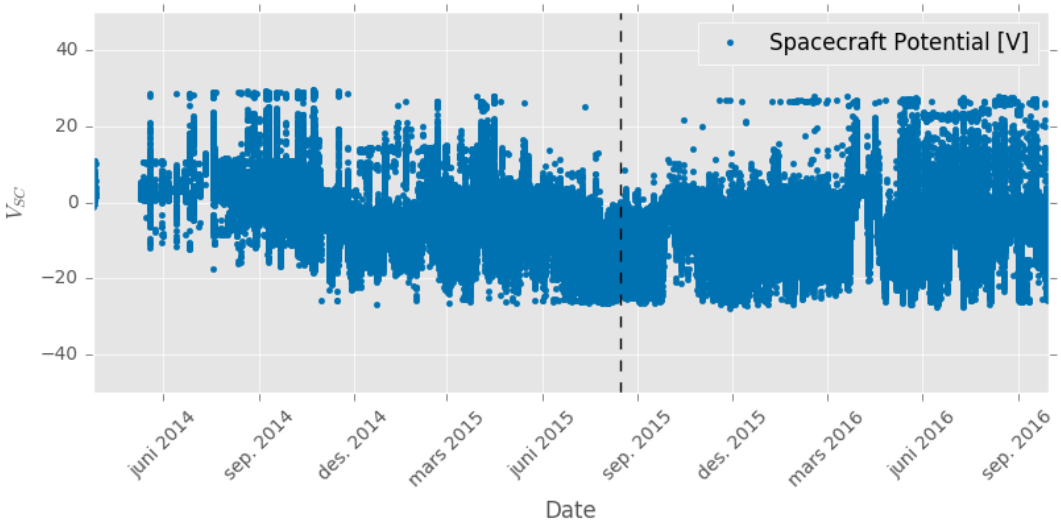


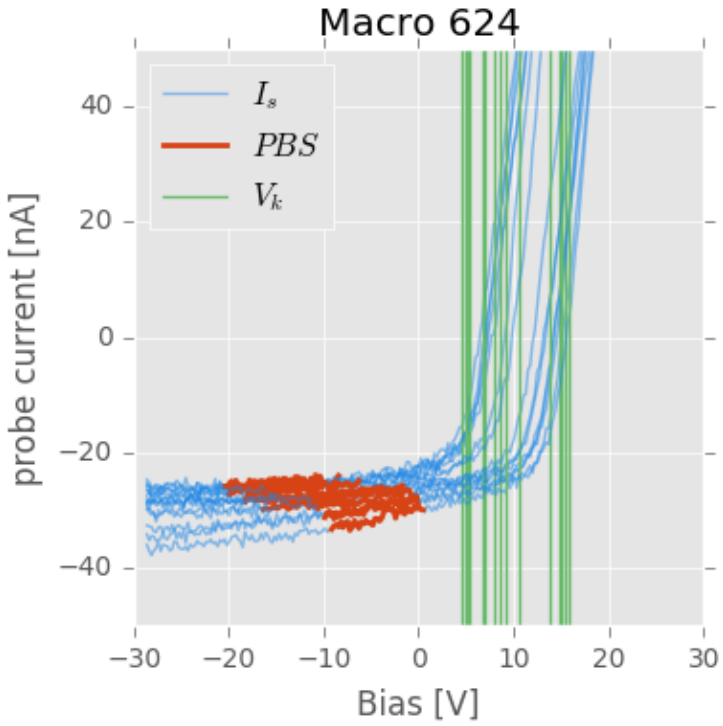
Figure 6.1: An estimation by the RPC-LAP team of the spacecraft potential during the mission. The estimated spacecraft potential is given in blue dots, with time along the x-axis and potential on the y-axis. Surprisingly the spacecraft charging seems to be very dynamic as well as stronger than first anticipated. The fact that the spacecraft charging is predominantly negative is positive for this method, negative V_{sc} provides additional electron current response shielding otherwise not accessible. Spacecraft potential data provided by the Swedish Institute of Space Physics, Uppsala, Sweden https://www.space.irfu.se/rosetta/.

If the electron temperature were to change by some process that introduce a plasma population with warm electrons for example, the electron response I_e will from equation 4.35, increase the slope of the ion saturation region current as well as increase the probe current I_S. Take this scenario to the extreme and the shape of the electron current response will absolutely dominate the other probe currents, which leads to a current-voltage characteristic that is very different from the shape of the ideal IV-characteristic. In this scenario the analysis method breaks down and we must use some other method to reliably investigate solar flux.

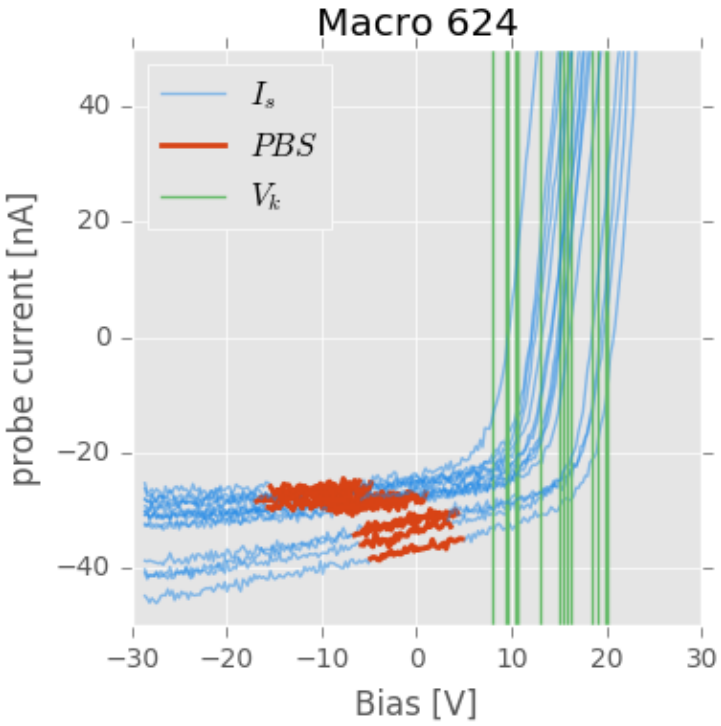
Data Collection

We have to make a few considerations when collecting data for analysis. It is crucial that we ensure that the probe is in fact sunlit during our analysis, and ensure that all the information central to the analysis is present in each datafile. All of the data files and the information contained in them has been provided

Two sets of sweeps taken from the 5th of July 2015.



(a) A set of real sweeps from 00:45



(b) Another set of sweeps some at 3:45

Figure 6.2: Two different set of sweeps produced using the multiple sweeps method at high resoultion, which means using 10-15 sweeps for the analyis.

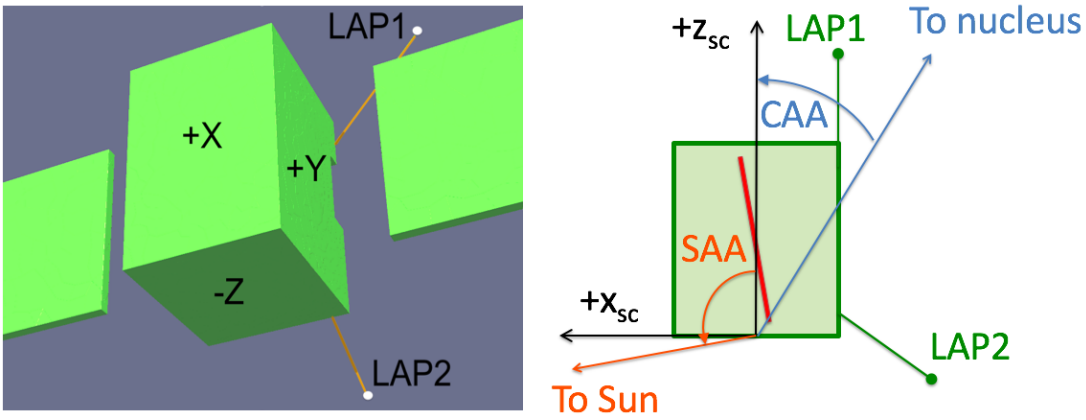


Figure 6.3: The Rosetta Spacecrafts shape with respect to solar panels and the Langmuir probes. X, Y and Z are coordinates centered on the spacecraft. LAP 1 and LAP2 are the two Langmuir probes, extended on booms to reduce the effects of spacecraft charging. SAA is the solar aspect angle, the angle which the sun rays hit the spacecraft, and notably the spacecraft solar panel array. CAA is the cometary aspect angle, here shown in relation to the spacecraft and the radial axis towards the sun. Image Credit: Fredrik Johanson drawing & Anders Eriksson sketch (Johansson et al., 2017).

by the LAP team at Swedish Institute of Space Physics (IRFU), Uppsala, Sweden.

If the probe is in the shadow of say, a solar panel or the high gain antenna for communications to Earth, then presumably there should be no photoemission current to detect. It has been shown in (Johlander, 2012), that an effective filtering of the solar irradiated probes is to filter with respect to the solar aspect angle. The coordinate system of the spacecraft with respect to its surroundings is given in figure 6.3. For example LAP 1 is sunlit when $SAA \in [131, 181]$, any sweeps outside of this range is naturally omitted from this analysis routine. The Uppsala team has implemented a simple filtering tag which is based on the solar aspect angle, so that all data files for the probes are marked as either sunlit or obscured. Uppsala is the principal investigator for RPC-LAP:https://www.space.irfu.se/rosetta/.

Another crucial parameter that is contained withing the data files, is the photoemission knee. This value serves as a reference for the multiple sweeps method to ensure that the same part of the characteristic is investigated between across multiple sweeps. If this parameter is not available from the collected datafile, then the datafile must be discarded, and the method breaks down. The photoemission knee is found by inspecting each sweep, for reference see figure 4.1, looking for the exact point where the characteristic changes from the ion satu-

ration region to the transition region. This point has been defined as the point of maximum inflection of the curve. The photoemission knee has been estimated by the data providers. They analyzed individual sweeps using a smoothing algorithm, and then calculating the second derivative of the curve to find the point of maximum curvature to locate the photoemission knee. Some sweeps may have shapes that do not conform to the predicted shape, or may have multiple inflection points, which complicate the localization of the photoemission knee. For some IV-characteristics, the knee potential falls somewhere in between easily identifiable and obscured, then it becomes a question of how well defined does V_k need to be for us to make use of it to extrapolate the photoemission currents. I have chosen to leave that decision to the data providers. We do not look further into the photoemission knee detection method in this thesis.

Chapter 7

Results

This chapter presents results from applying the new multiple sweeps method to the RPC-LAP sweep mode dataset. I look at the timeseries and normalized slope distributions for the whole mission, using preferred science macros from table 6.1. I also present some results from using an expanded macro list in an effort to provide better data coverage. Implications of some these mission overview timeseries have also been covered in (Johansson et al., 2017). However the main focus in the present study is to evaluate the possible use of this method to analyze short timescale ($<$ 1day) photoemission current signatures. I have decided to look for signatures of known flare events, which if found can be used as a benchmark, for further study.

A Mission Overview of the Photoemission Current

The RPC-LAP dataset spans a total of 31 months, the results of the multiple sweeps method I_{ph} for the duration of the mission are given in figure 7.1. In the top panel LAP 1 and LAP 2 with a moving window standard deviation are marked in red and blue dots, respectively. The moving window standard deviation of 30 measurements for the two probes, is given in shades of red and blue respectively. For reference, a simple model for the solar irradiance, which is scaled to the sun-shadow transition (SST) method markers in Mars 2015 and April 2016, is traced with a yellow line. The SST method is explained in some detail in the appendix, 11.3. It is a very reliable method for extracting the photoemission current from the RPC-LAP, but it does lack extensive data coverage. In this thesis I utilized the SST method as a calibration for the multiple sweeps method. The top panel has been limited to $I_{ph} \in [0, 50]$nA in order to focus on the shape of the photoemission timeseries curve.

In the lower panel we find the spacecraft positional data, the comet-sun radial distance r_s is traced in yellow, while the spacecraft to comet radial distance r_c is given in red. First notice the data gap from the beginning of the mission to December 2014, in this period none of the preferred science macros were in operation. Apart from the data gap in the beginning of the mission there is also a small data gap for the two first weeks of June 2015 as well as the last two weeks of June 2016. Note in the positional data the many maneuvers that were performed in Mars 2015, as well as the night-side excursion of September 2015[1]. The night-side excursion took the spacecraft away from the nucleus to about $r_c = 1500$km.

In the lower panel we trace the radial distance from the spacecraft to the Sun in astronomical units, throughout the mission. Also in the bottom panel, is the spacecraft to comet nucleus radial distance in kilometers, given in red. We start tracing the distance as the spacecraft approached the comet in August 2014. The photoemission current estimates from the multiple sweep method start in December 2014, and display similar time evolution as the intermittent sun shadow transition (SST) method. After March 2015 as the spacecraft begins maneuvering in relation to the comet there is a period of high standard deviation in the timeseries at the beginning of April. A quick inspection of that cluster of LAP 1 measurements is given in the appendix 11.5. Around perihelion, probe 2 does have period of high standard deviation, as well as in march standard deviation in March 2016 for probe 1.

We see that the general shape of the timeseries is that of a parabola increasing as the comet and spacecraft approach perihelion. The parabola shape is to be expected; as the comet moves closer to the sun the photoemission current is predicted to increase with the solar flux $I_{ph} \propto r_s^{-2}$, as explained in the introduction 3.1.1. It is apparent when comparing the timeseries to the solar irradiance model that there is a discrepancy of roughly 20nA between the two at perihelion. This "missing" photoemission has been investigated in detail by Johansson et al. (2017), suggesting a possible explanation of a cometary dust cloud between the Sun and the spacecraft. The multiple sweeps method timeseries show good agreement with the SST method, at all phases of the mission.

Inspecting the photoemission timeseries and comparing probe 1 with probe 2, we can already infer that the probes respond individually to the plasma. In the second half of the mission after perihelion probe 2 has a slight persistent negative offset in comparison to probe 1. In the first half of the mission probe 1 has a

[1]The Night-side refers to the nucleus ground track of the surface instruments. The spacecraft remains sunlit during this excursion.

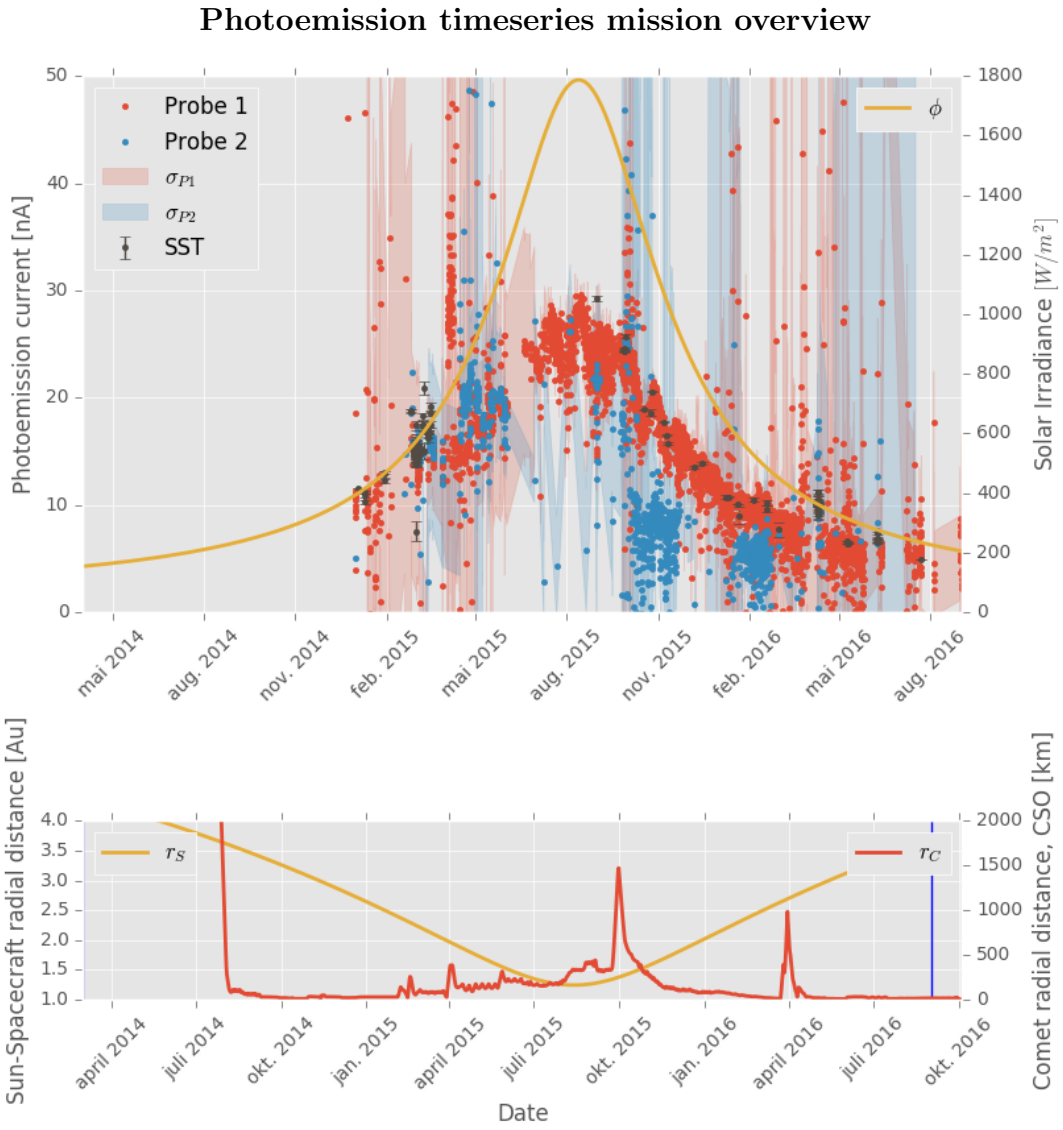


Figure 7.1: A photoemission current timeseries from Rosetta's escort of the comet 67P/C-G. In panel one the multiple sweeps method of probe 1 and probe 2 as well as the Sun-Shadow Transition method. A simple solar irradiance model, denoted with ϕ, is added for reference. In the lower panel, the spacecraft radial distance with respect to the comet r_c and the Sun is traced with red and yellow lines.

large cluster of data which lies above probe 1 in April 2015, before it falls in line and is of the same order as probe 1 in May 2015.

Investigating the Sweep Fits Using Distributions

Let us look back at figure 5.3, imagine the three lines as not just different velocities but different, plasma populations. Suppose we gather all the slope extrapolation points, divide by the probe current to normalize, and plot the distribution. We then get a tool that can help us validate the results we see from the timeseries. We can check with the normalized slope distribution if the extrapolation data points are well conditioned for sweep fitting. We have assumed that the plasma consists of a single population, if we also assume that the sweep fitting is mainly influenced by random error, due to fluctuations in plasma parameters, we expect a Gaussian distribution. The random error assumption is reasonable, because there are no obvious reasons why the sweep fit which is influence by instrument noise should not have randomized errors. Should the distribution shape be non-Gaussian, the emerging patterns may reveal new information about the dataset. If we observe are highly variable plasma conditions we can get a strongly dispersive distribution, because of large fluctuations in the sweep fit extrapolation points.

The normalized slope distributions are the distribution of the linear fit to the slopes of individual sweeps, divided by the model current for that same sweep. To create a distribution of statistical significance for a given period of time, we collect the slopes from step one of the multiple sweeps model, see again figure 5.1.4. We take the collected slopes and divide them by their respective linear model current, and subsequently plot them in a histogram. This distribution is then the collective statistics of the data used in step two of the method, the extrapolations step from figure 5.2. This type of distribution is best utilized on short timescales, because of the assumptions that we made in justifying the distributions. For example if we are looking at the normalized slope distribution for the whole mission, figure 7.2, then we can already claim with confidence that the one plasma population assumption is broken. Even still the distribution looks reasonable. It is by no means strictly Gaussian, but it is possible to make a reasonable normal fit if we adjust it to the peak and limit the data input to surround the peak.

Now that we have inspected the timeseries of the photoemission current, let us also consider the normalized slope distribution of the overview timeseries. The total distribution for the whole mission timeseries is given in figure 7.2, the the number of normalized slope extrapolation points is given in red bars. First of all both probe distributions are bell-shaped, already suggesting that random errors dominated some of the noise. Interestingly the distribution peak of probe 2 seems to be about $0.006\mathrm{V}^{-1}$ negatively shifted with respect to probe 1. There is a small hump in both of these probes distribution suggesting that there, might

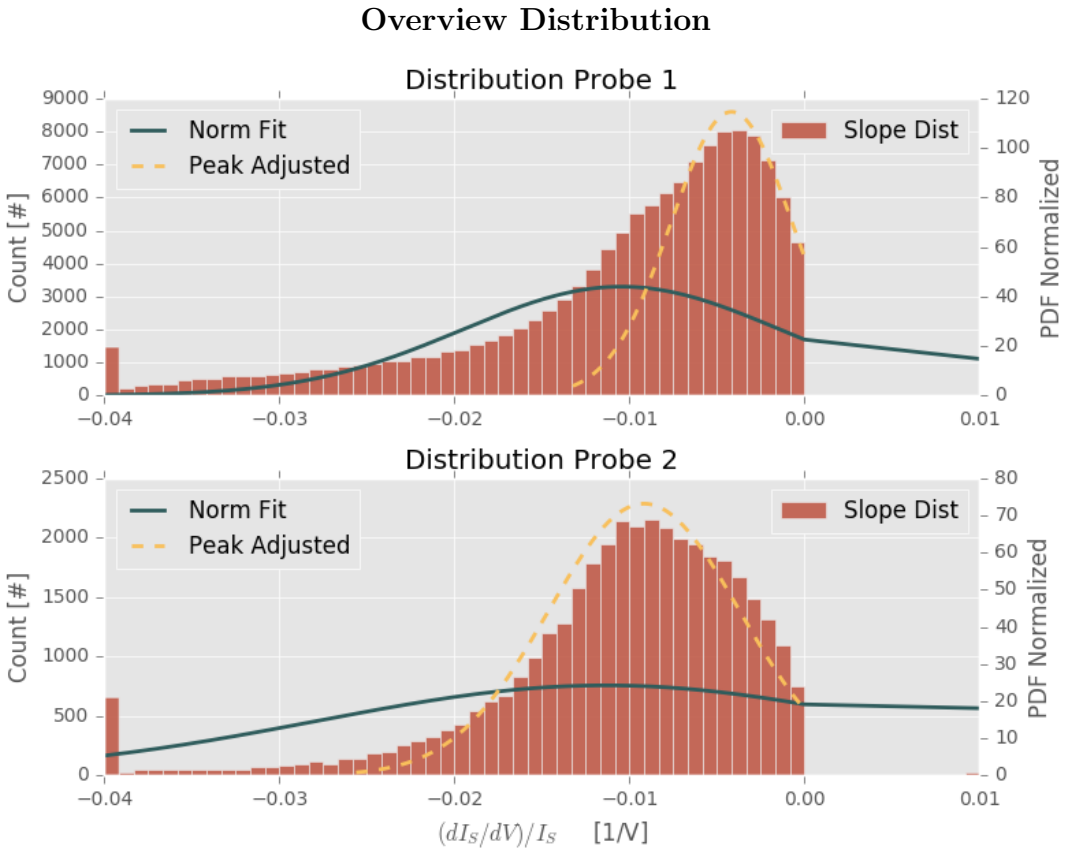


Figure 7.2: The main bulk of the normalized slope distribution of the whole mission duration. Along the Y-axis we have the counts for the distribution, as well as the Probability Density Function (PDF) On the X-axis we find the linear sweep slope fit over probe current. In red bars is the normalized slope distribution, in the top panel probe 1, in the bottom panel probe 2. The blue line is the a best fit normal distribution. The yellow line is a peak adjusted normal distribution best fit.

be two dominant plasma populations. It could also mean that there is a bias due to macro issues, or a bias emerging from the different sample sizes of the two probes. The distributions are clipped, meaning that all the distribution values above the most positive and below the most negative bin value is added to the end bins. It is readily apparent from the negative side of the distributions, both probes have some outliers outside the figure window. In blue and yellow lines, I have traced two best fit normal distribution solutions. The blue line is the best fit to the distribution. The long tail-end of the distributions means that the normal fit does not closely match the distribution.

In order to produce a nice visual guide for the main peak and main bulk of

the distributions I adjusted the normal fit slightly, the new fit I call the peak adjusted normal fit and it is given by a yellow dashed line. For the peak fit, the potential distribution function is shifted from the mean of the set to the peak of the distribution. In order to filter most of the outliers for the peak fit it is capped at 3 standard deviations of the standard normal distribution, from the peak. The resulting yellow trace serves as a visual reference for the peak of the main bulk, as well as providing statistical information focused on that peak of the distribution.

Some statistics of the distributions from this chapter are given in table 7.1. If we set aside the peak fit in this first inspection, we find that the mean is slightly more negative for probe 2 than for probe 1. The shift towards more negative values in probe 2 might be because the probes measure at different times, see figure 7.1, and is therefore biased to their respective measurement periods. When it comes to extracting physical information from these distributions we are mostly interested in the behaviors of the main bulk, the outliers are expected to be largely erroneous measurements. This is a reason why the peak adjusted normal fit is useful, since it targets the peak and the main bulk of the distribution. However, extracting a physical interpretation from the overview distribution is not easy, it is more useful for detecting instrumental anomalies and inspecting the outliers of the whole mission. The standard deviation for probe 1 is higher than the standard deviation of probe 2, suggesting that the distribution of probe 1 is more disperse than the distribution of probe 2.

The skewness is negative for both probes which is not surprising considering how the method is implemented. The skewness simply means that the tail of the distributions are shifted to the negative side over the positive side, this is a direct result of the negative slope filtering mechanism in the multiple sweeps method. The filtering process is explained in section 5.1.4. Only positive ion saturation region slopes are kept, but when the slopes are normalized to the negative probe current, the emerging normalized slope distribution is negative.

The kurtosis[2] value is normalized to that of a theoretical normal distribution. The kurtosis is comparing the tail-end of the distribution to that of a normal distribution. Since the kurtosis of a normal curve is three this means that the tail-end of probe 1 is quite pronounced, while for probe 2 it is comparable to that of the normal curve.

[2]All kurtosis values are relative to the univariate normal distribution in this thesis.

Table 7.1: Statistics for the normalized slope distributions in this chapter. The first and second elements of the value pairs are probe 1 and probe 2 respectively.

Distribution	**Mean** (10^{-3})		**Std.** (10^{-3})		**Skewness**		**Kurtosis**	
Overview	-10.5	-11.3	9.1	16.4	-1.70	-9.69	4.02	431.52
Inbound	-6.68	-4.31	4.38	3.13	-0.73	-1.45	-0.06	2.61
Outbound	-11.18	-13.07	9.55	13.74	-1.61	-7.48	3.80	201.50
Overview Peak	-6.0	-9.6	3.5	5.4	-0.23	-0.52	-0.98	-0.17
Inbound Peak	-4.19	-3.22	2.20	1.62	-0.05	-0.07	-1.00	-0.97
Outbound Peak	-6.52	-11.06	3.82	4.77	-0.26	-0.47	-0.95	-0.09

Comparing the Inbound and the Outbound Passage

From the cometary physics perspective it is interesting to study the comets evolution as it is gradually heated on the inbound leg of journey to its peak at perihelion, before eventually cooling off again during the outbound. It is for this reason I now take a closer look at similarities and differences of the approach and departure. In the following normalized slope distributions I have split the overview timeseries in two halves at perihelion, figure7.3 and 7.4 for the inbound, and figure 7.5 and 7.6 for the outbound. Again we are not looking for physical interpretations right away, rather investigating what the method does with the normalized slope data on the scale of the mission duration.

The shape and other visual aspects of the timeseries has already been covered but the figures are repeated here for clarity. Splitting the timeseries in two halves emphasizes the disparity in data coverage between the inbound and the outbound. From this first statistical inspection we note that the mean shifts towards more negative values from the inbound to the outbound. We also note that the probe with the most negative mean value switches from the inbound to the outbound, from probe 1 being the most negative on the inbound to probe 2 on the outbound. The mean is slightly more negative on the outbound compared to the inbound for both probes, which means that the gradients of the slopes are steeper. On the inbound, probe 1 has a quite pronounced bump, which is not visible on the outbound. This could mean that the probe pick up two different plasma populations, which manifests as distinct slope gradients through for example high and low electron temperature.

The standard deviation on the inbound is lower than for the outbound, this might be because of the smaller dataset available on the inbound with respect

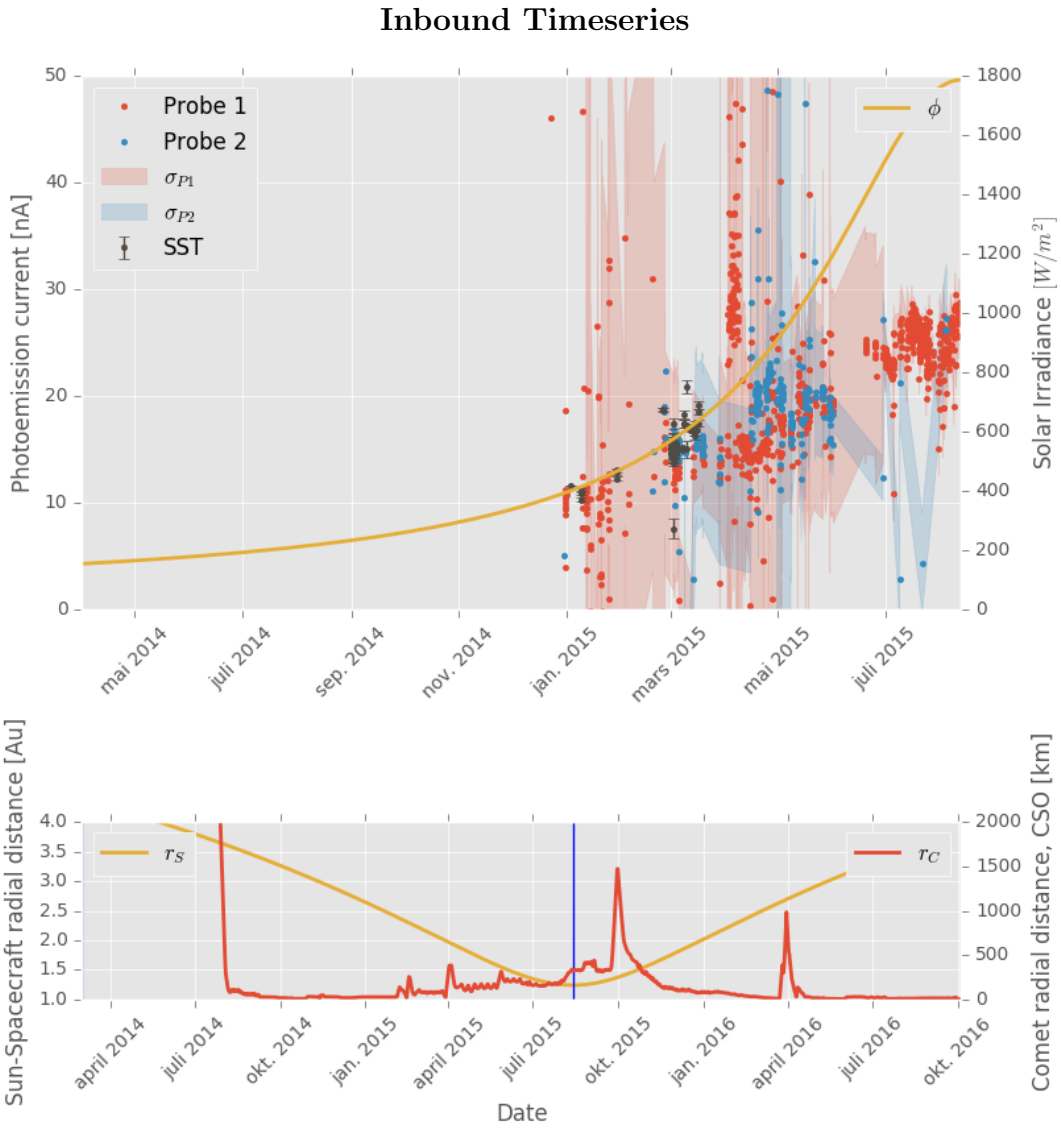


Figure 7.3: Timeseries of the photoemission current derived by the multiple sweeps method on the inbound of the Rosetta mission. This timeseries is made using only trusted macros. Probe 1 are given in red and probe 2 in blue, a rolling window standard deviation in shades of red, and blue for the respective probes. For reference, the sun-shadow transition method in dark gray markers, and a simple solar irradiance model are given in yellow. There is spacecraft positional data in the lower panel, where the blue line marks the separation between the inbound and the outbound in the timeseries.

to the outbound. probe 2 has a more negative skewness than probe 1, for both the inbound and the outbound, suggesting that the negative tail is longer, and thicker than for probe 1. However the skewness difference between the two probes

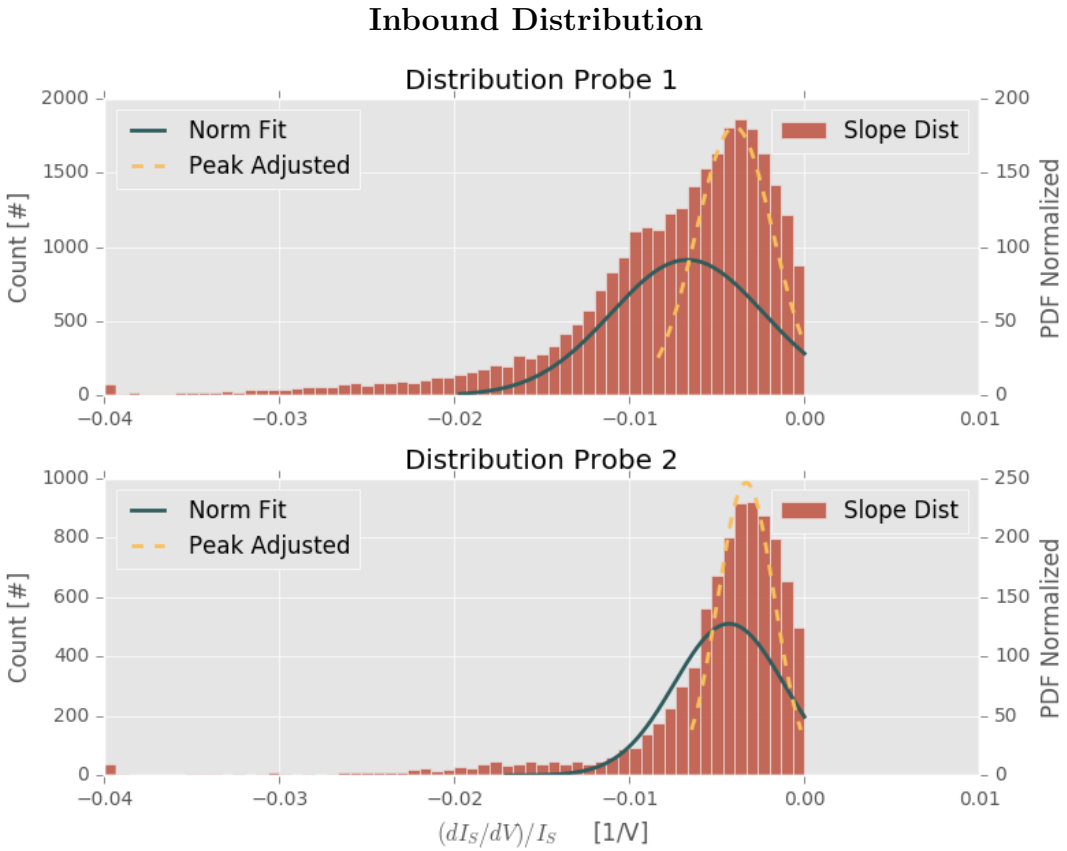


Figure 7.4: A normalized slope distribution for the inbound leg of the Rosetta mission. The probability density function in histogram bars, with a standard normal fit in blue and an peak adjusted normal fit in yellow staples. In the top panel, probe 1, in the lower panel, probe 2.

is larger on the inbound rather than the outbound. The relative kurtosis of both probes on the outbound is drastically reduced from the inbound, meaning that the outbound leg is characterized by drastically fewer and less extreme outliers. Especially the kurtosis of probe 2 is drastically reduced from the inbound to the outbound.

Normalized slope distributions for short timescales

The mission overview results are interesting in themselves, however another use of these distributions is much more interesting from the physical point of view, rather than instrumental. That is to apply them to the short timescale investigation of the timeseries photoemission data. I use them as a tool to indicate whether or not a short timescale signal is trustworthy. On short timescales where our assumption are more often justified, we can also use the distributions to ex-

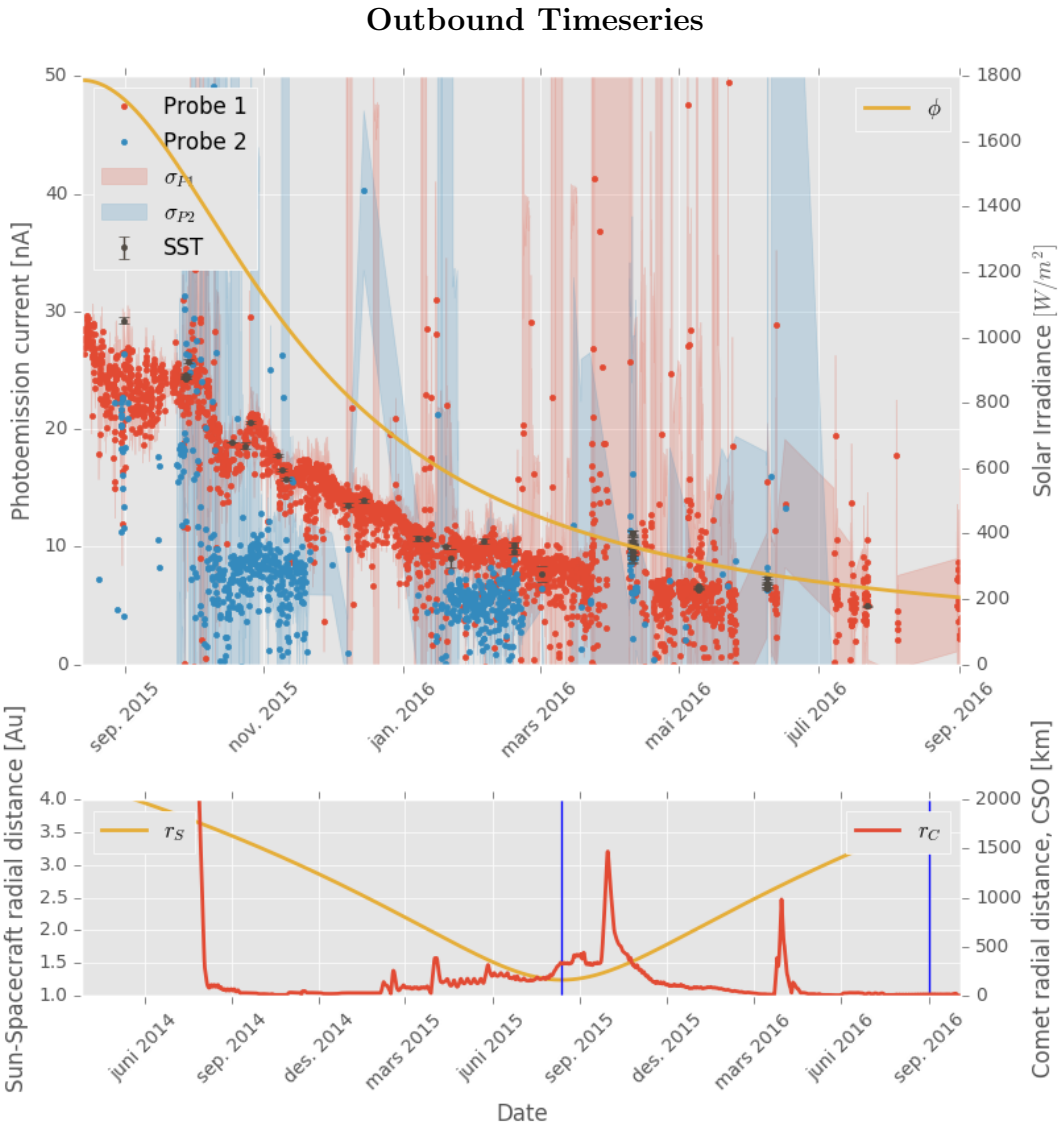


Figure 7.5: Timeseries of the photoemission current derived by the multiple sweeps method on the outbound of the Rosetta mission. This timeseries is made using only trusted macros. Probe 1 is given in red and probe 2 in blue markers, a rolling window standard deviation is painted in shades of red, and blue for the respective probes. For reference, the sun-shadow transition method is added with dark gray markers. A simple solar irradiance model is traced with a yellow line. There is spacecraft positional data in the bottom panel, the blue lines mark the duration of the outbound timeseries in the top panel.

tract additional information that the timeseries does not provide. It is for this reason that the distribution is to be considered additional information to the

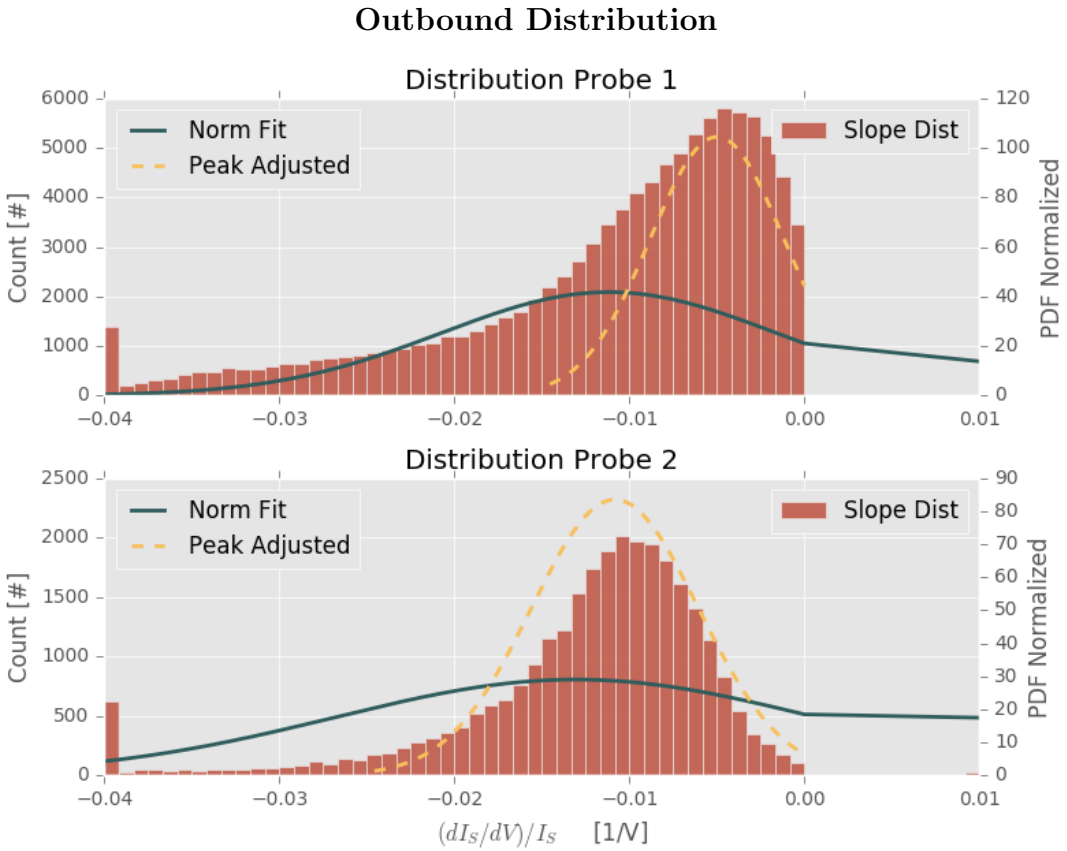


Figure 7.6: A normalized slope distribution for the outbound leg of the Rosetta mission. The probability density function in histogram bars, with a standard normal fit in blue and an peak adjusted normal fit in yellow staples. In the top panel, probe 1, in the lower panel, probe 2.

multiple sweeps method, and can be used in conjunction with other methods, like the SST method or models.

Comparing the Dataset to Transient Events

The multiple sweeps method presented in this thesis is able to extract a detailed timeseries from the RPC-LAP dataset. We have already shown that the multiple sweeps method is seemingly able to trace the photoemission current response, and the solar flux by proxy at large mission timescales. Using a similar multiple sweeps method Johansson et al. (2017) have already obtained the evolution of the solar flux throughout the mission duration, as well as identified perturbations to the solar flux such as the solar sidereal rotation ($\approx$ 24.5 days).

With the method in presented in chapter 5, the improved resolution possibly facilitates an event study into solar transient events on the timescale of hours to days. To coordinate different investigations into solar transient events the Rosetta team has compiled a cross instrument list of know solar transients at Rosetta, including flares, CME's, CIR's and SEP's. Out of these, only flares and CME's are directly observable as intensification in the solar EUV flux. CIR's and SEP's may change the plasma parameters indirectly affect all measurements done with the Langmuir probe. A CME or SEP event might alter the plasma population in a way that is so distinct that it is visible with the distribution inspection McKenna-Lawlor et al. (2016); Edberg et al. (2016). It might also be possible to infer events perturbing plasma conditions such as CIR's and SEP's by proxy, although this is not the main focus here.

Flares reported at Rosetta by the RPC team.

Table 7.2: All subset of transient events reported at Rosetta by the RPC team. The list is a work in progress by the Rosetta RPC community.

Date and time	**Flare Class**	**Duration** [min]	**Coverage**
2014/06/10 - 12:00	X1.8	10, 35	None
2014/08/24 - 12:20	M6.0	15	Expanded
2014/09/10 - 18:20	X1.6	90	None
2014/09/28 - 03:20	M5.0	45	Expanded
2014/10/02 - 19:20	M7.0	20	Expanded
2014/10/22 - 14:00	X1.7	90	None
2014/10/22 - 18:00	M1.7	-	None
2015/11/20 -	M7.0	-	Trusted
2016/02/18 -	-	-	Trusted
2016/04/18 - 00:45	M7.0	30	None

First I will present results from investigating the photoemission timeseries using the multiple sweeps method on data using the most reliable macros for this type of study, see 6.1 for more information on the macros. There are two reported events that coincide with the trusted macros data coverage. These are two GOES category M7 flare events. These events were reported in November 2015, and February 2016, see all potential flare events in table 7.2 (C. Simon Wedlund, personal communication). Three other flare events has been covered using an expanded macro list 11.1, with macros not specifically tuned for the multiple sweep method. Another three flares has been reported by other instruments, however in periods where there is no data cover with the new method.

2015.11.20 M7 flare event

The first flare which I want to investigate is the M7 event reported on the 20th of November 2015. There was not specified a time of impact, I have therefore marked the whole day in a yellow shade in figure 7.7. However there does not seem to be any significant increase in the photoemission signature, that would follow with a strong increase in solar flux at any time during the 20th of November 2015. This event was one of the flares reported by Johansson et al. (2017).

Let us look closer at the statistics of the slope distributions for this period. In figure 7.8 we see the normalized slope distribution for the period 15th to the 25th of November 2015. probe 1 registers a distinct double peaked curve in this period, which may suggest that the timeseries was influenced by a considerable second population of plasma in addition to the main bulk.

Table 7.3: Statistics for the normalized slope distributions in the period 15th to the 25th of November 2015. Value pairs are probe 1 and probe 2 respectively.

Distribution	**Mean** (10^{-3})		**Std.** (10^{-3})		**Skewness**		**Kurtosis**	
Normal	-12.39	-11.89	7.71	4.56	-0.75	-0.42	-0.22	0.18
Peak	-10.37	-11.57	5.69	3.87	-0.51	-0.18	-0.54	-0.42

2016.02.18 flare event

In the report by Johansson et al. (2017) there is a reported possible flare at the 18th of February 2016. The timeseries data from the multiple sweeps method for the 17th to the 25th of February 2016 is given in figure 7.9. The multiple sweeps photoemission current response in red and blue lines with the rolling window standard deviation in shades of red and blue. There are two data points of the SST method which are marked with dark gray markers. The spacecraft positional data is given in the lower panel as before. The blue lines in the lower panel, represents the time period of panel one in relation to the whole mission. February 2016 was a period of little to no positional maneuvers with respect to the comet. This event was recorded on the outbound leg of the journey, at roughly $r_S = 3.3$AU. There is a sharp peak of roughly 405nA in the photoemission current timeseries at 2016/02/22 13:30. The previous and following measurements are at background levels $\approx$ 10nA at 11:20 and 15:45 the same day. The data coverage is generally good in this period, all of the data is collected from macro 914.

Inspecting the slope distribution for the corresponding time and resolution in figure 7.10, we see that the shape of the two distributions are very similar to that

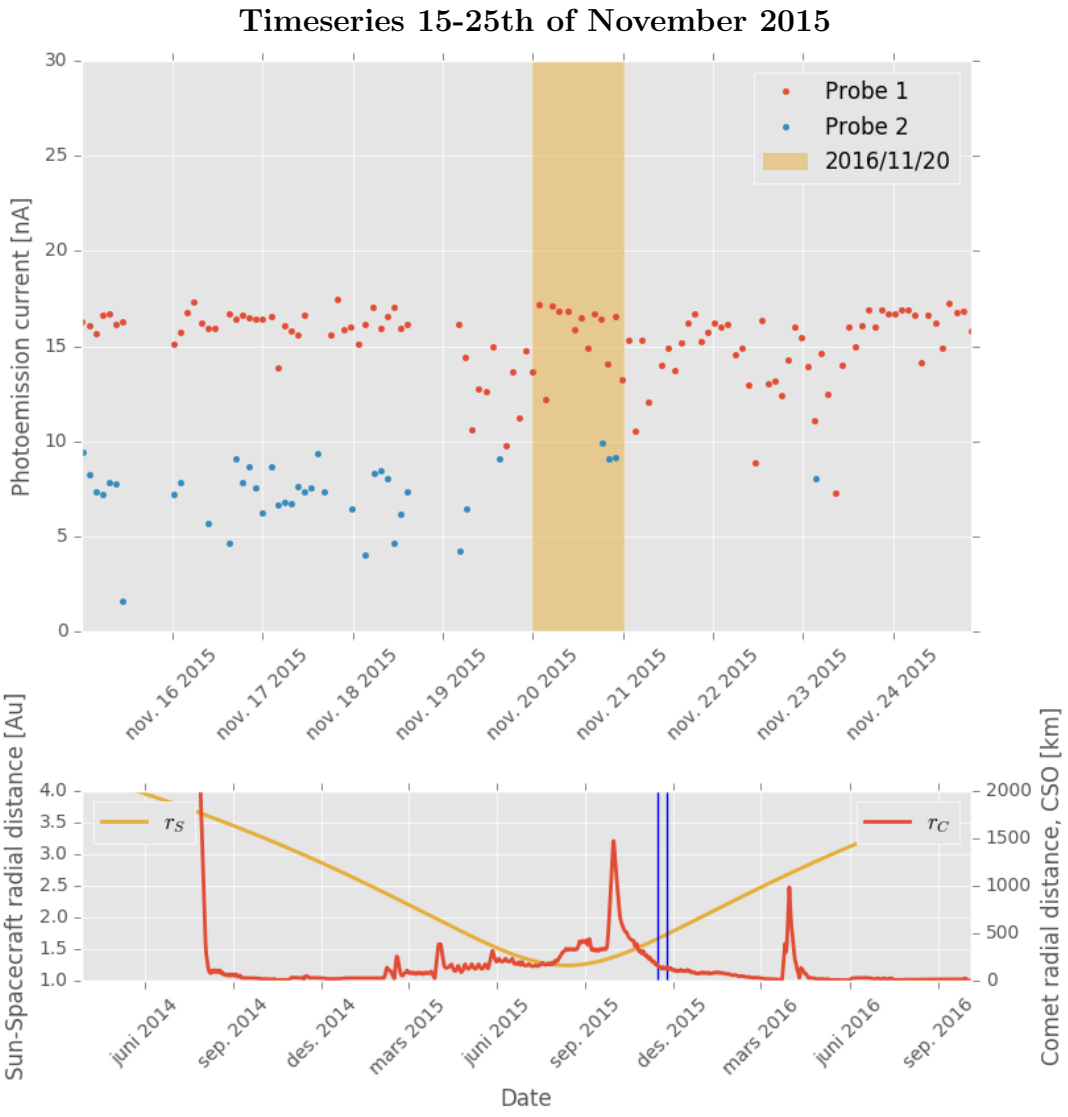


Figure 7.7: Top panel: A timeseries capturing the days leading up to, including and after a reported M7 Flare in November 2015. In red and blue the photoemission currents derived from the multiple sweeps method probe 1 and probe 2 respectively. Spacecraft positional data in the bottom panel, blue lines mark the top panel view

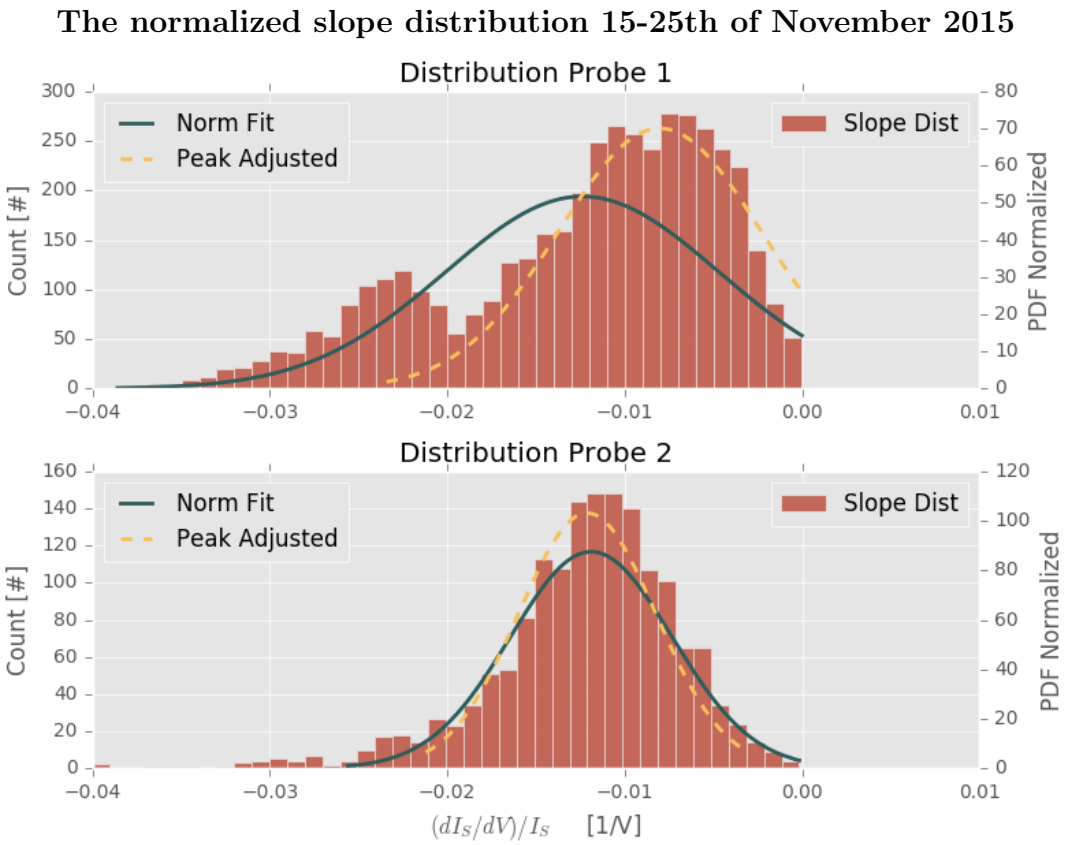


Figure 7.8: The normalized slope distribution for the span of days investigated in figure 7.7. probe 1 in the top panel, and probe 2 in the bottom panel. A normal fit, and a peak adjusted normal fit has been added for reference in blue and yellow.

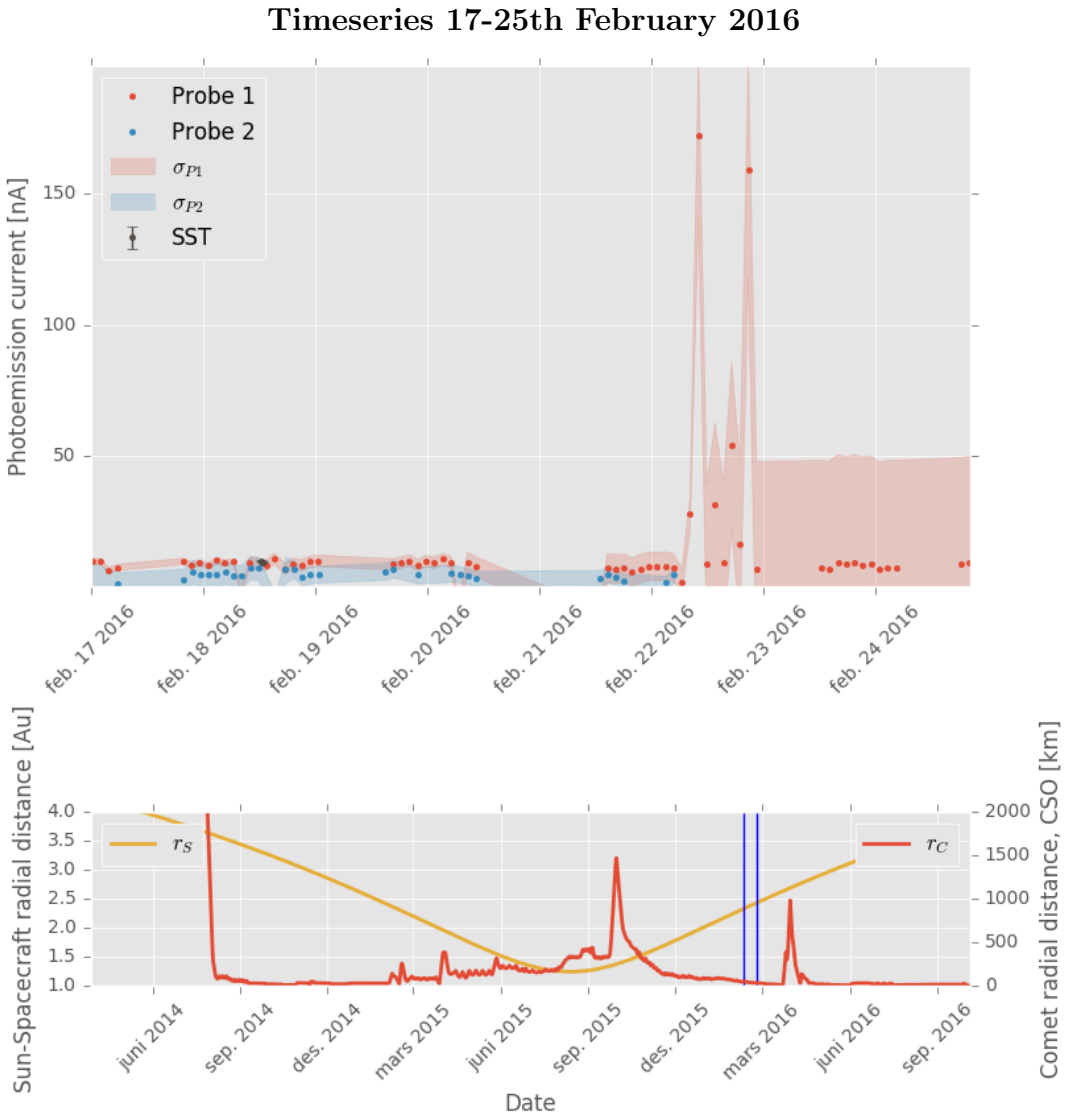


Figure 7.9: A low resolution timeseries for the period February 17-20, 2016. The multiple sweeps photoemission current is given in red for macro 914 probe 1 and in blue for macro 914, probe 2. The SST method is given with dark gray markers, positional data for the spacecraft in the lower panel.

of the outbound on the whole. Details of the statistics for this period is given in table 7.4. The normalized slope distribution does seem to have a heavy negative tail-end suggesting that a population of outliers are potentially disrupting the extrapolation used in the photoemission estimates that establish a timeseries.

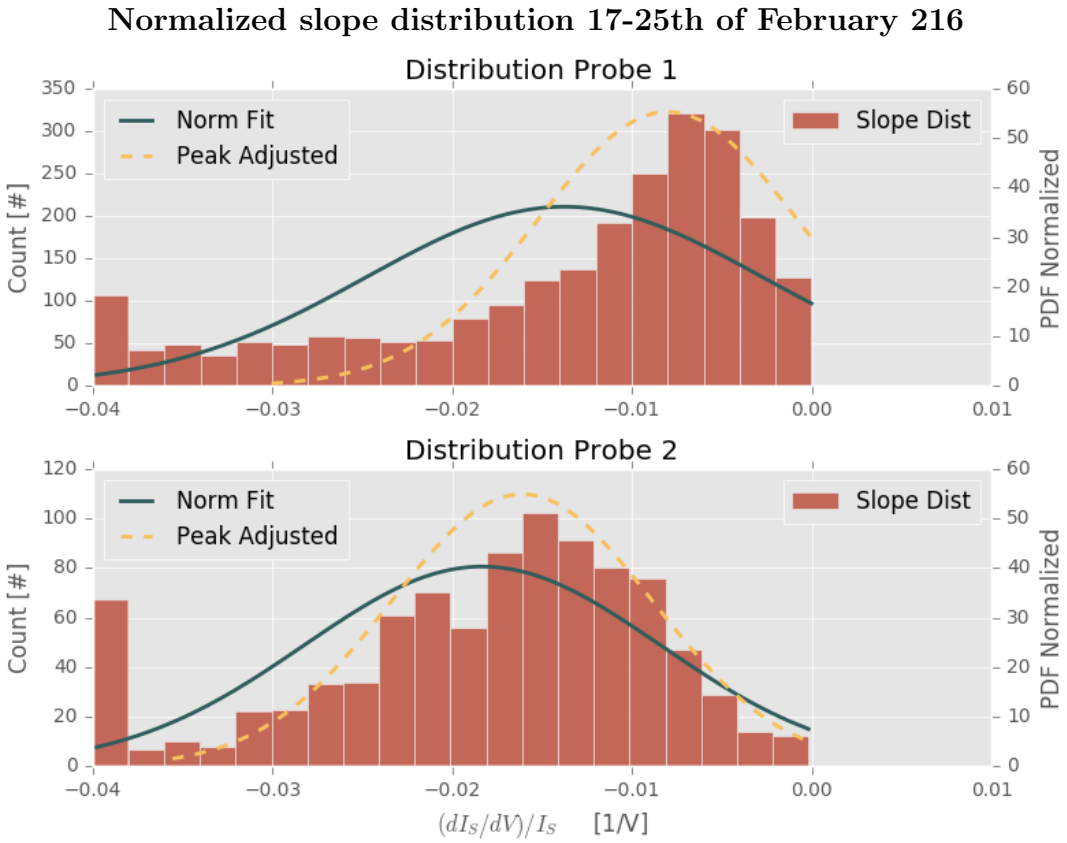


Figure 7.10: Medium resolution distribution for the period February 17-25th, 2016. The count is given by red bars, top panel for probe 1, bottom panel for probe 2.

Table 7.4: Statistics for the normalized slope distributions in the period 17th to the 25th of February 2016. Value pairs are probe 1 and 2 respectively.

Distribution	**Mean** (10^{-3})		**Std.** (10^{-3})		**Skewness**		**Kurtosis**	
Normal	-13.75	-18.32	11.03	9.90	-1.20	-1.05	0.75	1.05
Peak	-10.68	-16.38	7.21	7.27	-0.89	-0.33	0.02	-0.37

Expanded Macro List

In order to investigate the early mission we need to expand the macro list to also include macros not specifically recommended for the multiple sweeps method. There are a few macros in particular that contain a lot of data during the early mission. I want to try and utilize as much data as possible, in order to gain a more complete picture of the method. This is a way of testing the limits of the method, while simultaneously investigating the early mission flares from table 3.1. The expanded macro list can be found in the appendix 11, in table 11.1. An overview of the photoemission estimation results from analyzing the expanded macro list is given in figure 7.11. It is readily apparent that a lot more data is available through this list, albeit with an seemingly higher data spread.

2014.08.24 M6 flare event

There was reported a flare of GOES category M6.0 at the 24th of August 2014 approximately 12:20 at Rosetta. The duration was reported to be around 15 minutes, the flare duration is marked in yellow shade in the figure 7.12. The photoemission current is marked by red and blue markers in the top panel as before. There does not seem to be any increase in the photoemission current response in the timeseries at the time of flare impact. However there is a single point of increased photoemission in probe 1, immediately preceding the flare.

The normalized slope distributions for probe 1 and probe 2 int the time period 24-25th of August 204 is given in figure 7.13. There is a distinct difference between the distributions of the two probes in this period. On the one hand, probe 1 has a relatively sharp peak and a slightly lingering negative tail-end. On the other hand probe 2 is dispersed and display less of a tail than probe 1, both probes provide similar photoemission results.

2014.09.28 M5 flare event

Another flare event was recorded at 28th of September 2014 in the multiple sweeps expanded macros dataset. The impact has been approximated to 03:20, and the duration was reported to be 45 minutes. A timeseries for the flare is found in figure 7.14, the background while being quite tranquil, is absolutely dwarfed by a strong signal of almost 800nA recorded one hour after the calculated flare impact on Rosetta.

Looking closer at the normalized slope distribution the slope distribution of probe 1 is perfectly Gaussian, apart from the cut off at zero, in the period before, during and right after the reported flare. Suggesting that there are no major

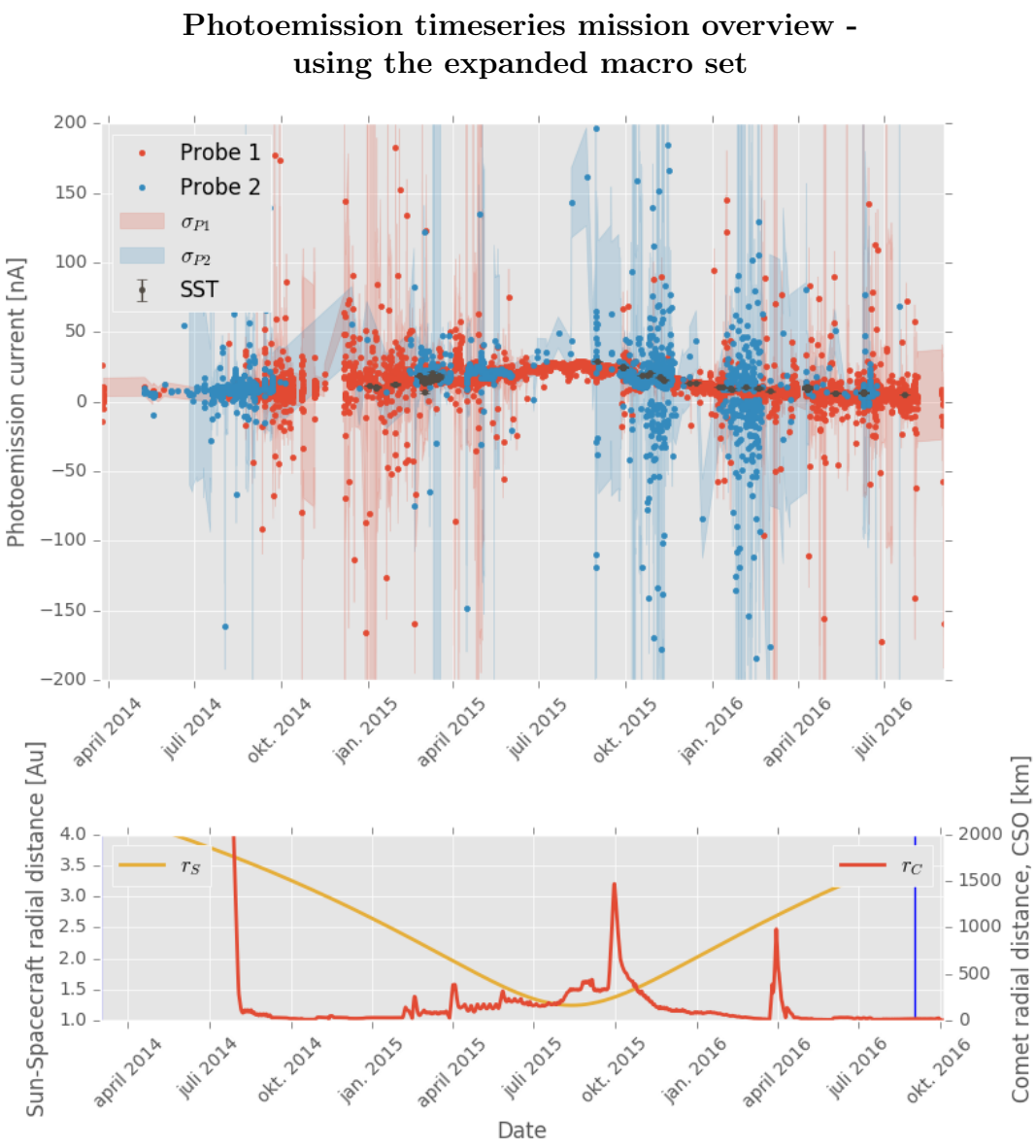


Figure 7.11: A photoemission current timeseries from when Rosetta was escorting the comet 67P/C-G. Shaded areas are the standard deviation for each probe in their respective colors. In panel two, spacecraft positional information: radial distance to the comet in red, radial distance to the sun in yellow. The dataset is noisier than the for the corresponding results derived from the trusted macros, but it covers the early mission.

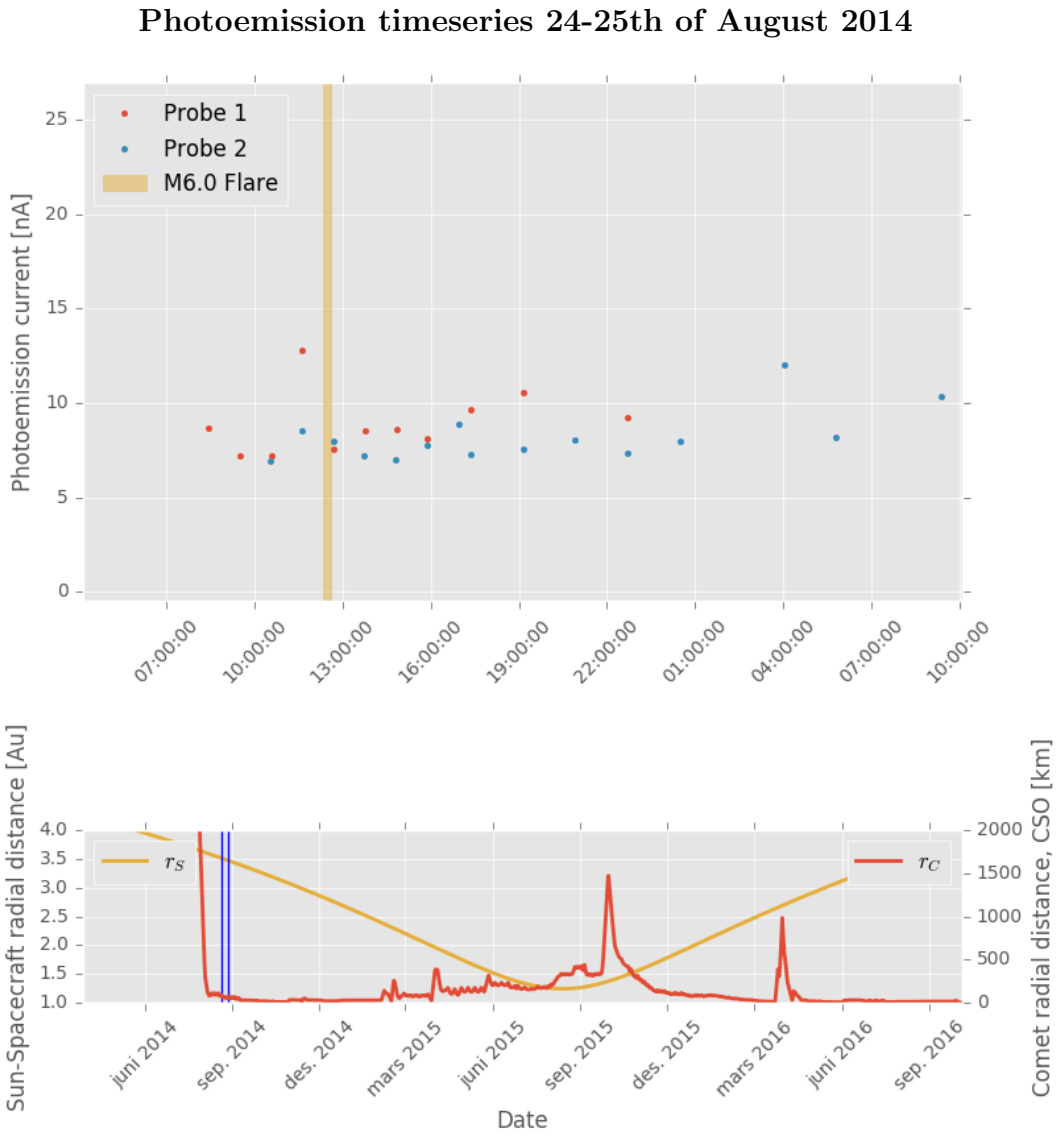


Figure 7.12: A timeseries capturing two days surrounding a reported M6.0 Flare in August 2014. In red and blue the photoemission currents derived from the multiple sweeps method probe 1 and probe 2 respectively. Spacecraft positional data in the bottom panel, blue lines mark the top panel view

Slope extrapolation distribution 24-25th of August 2014

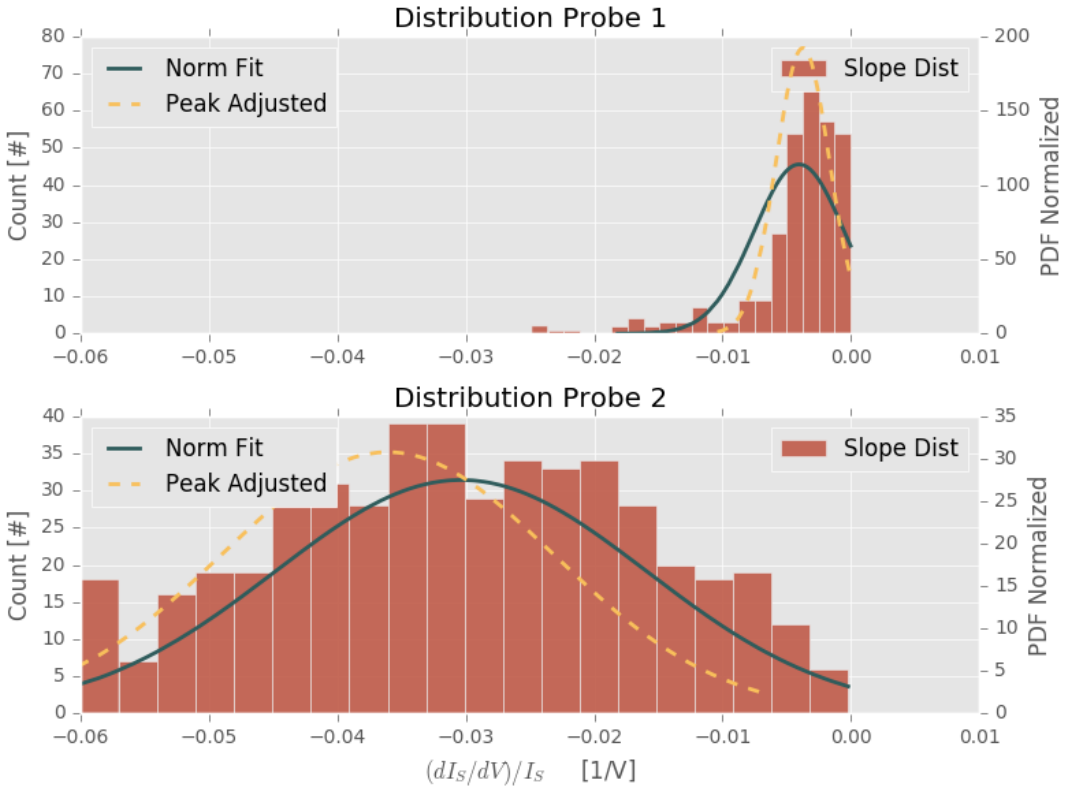


Figure 7.13: A normalized slope distribution for the period 24-25th of August 2014. The distribution data for probe 1 is given in the top panel, the distribution for probe 2 is given in the lower panel.

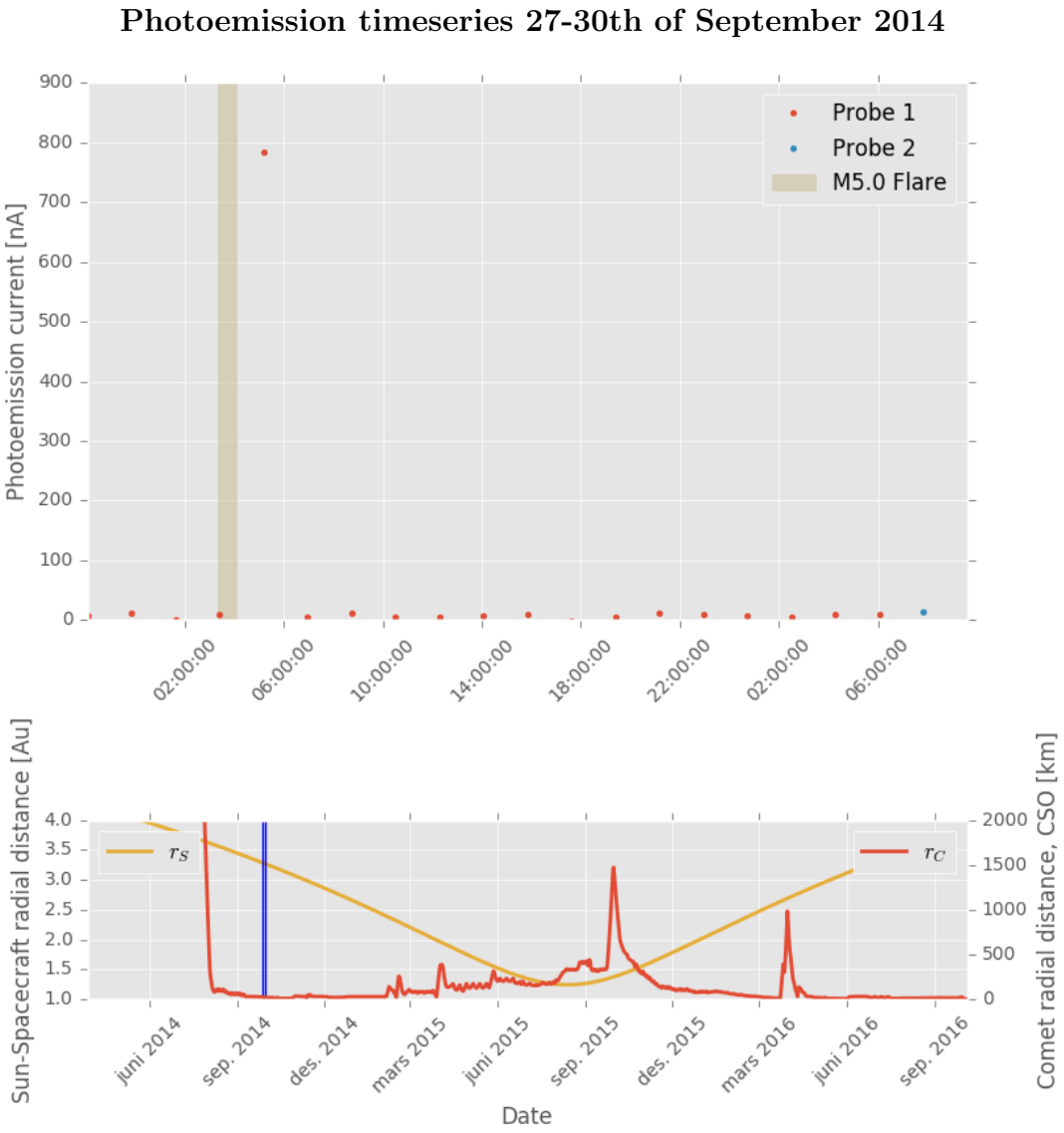


Figure 7.14: A three day photoemission timeseries for the M5 flare at the 28th of September 2014. As usual the multiple sweeps method estimations are marked in red and blue dots for probe 1 and probe 2 respectively. The spacecraft positional data is given in the lower panel. Notice that the photoemission current scale is particularly large for this event.

error sources other than random noise in this period.

Slope extrapolation distribution 27-30th of September 2014

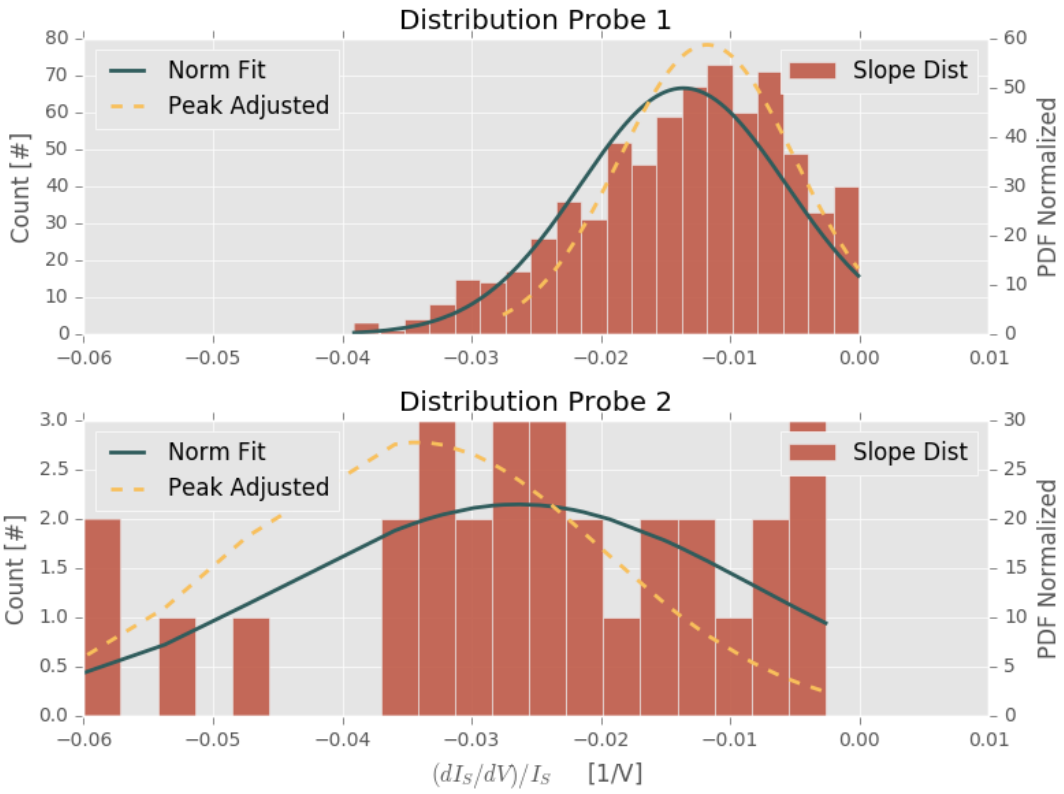


Figure 7.15: A normalized slope distribution for the period 27-30th of September 2014. probe 1 in the top panel is the most relevant for this particular event as the data for probe 2 is not statistically significant in this period.

2014.10.02 M7 flare event

The last event I want to present is an M7 event with impact at Rosetta at 19:20 October 2nd 2014. In figure 7.16, we only have red markers of probe 1 in this period. The flare duration is marked by yellow shading. There is no discernible photoemission response to the flare activity. The slope distribution is Gaussian indicating that the measurements error is dominated by random noise. The single data point caught in the middle of the flare duration is not statistically significant to draw a conclusion on the flare response by itself. It should be noted that, even though the single point is right in the middle of the event, it is essentially an average between the two neighboring data markers. In this instance the method resolution is definitively to low draw any conclusions.

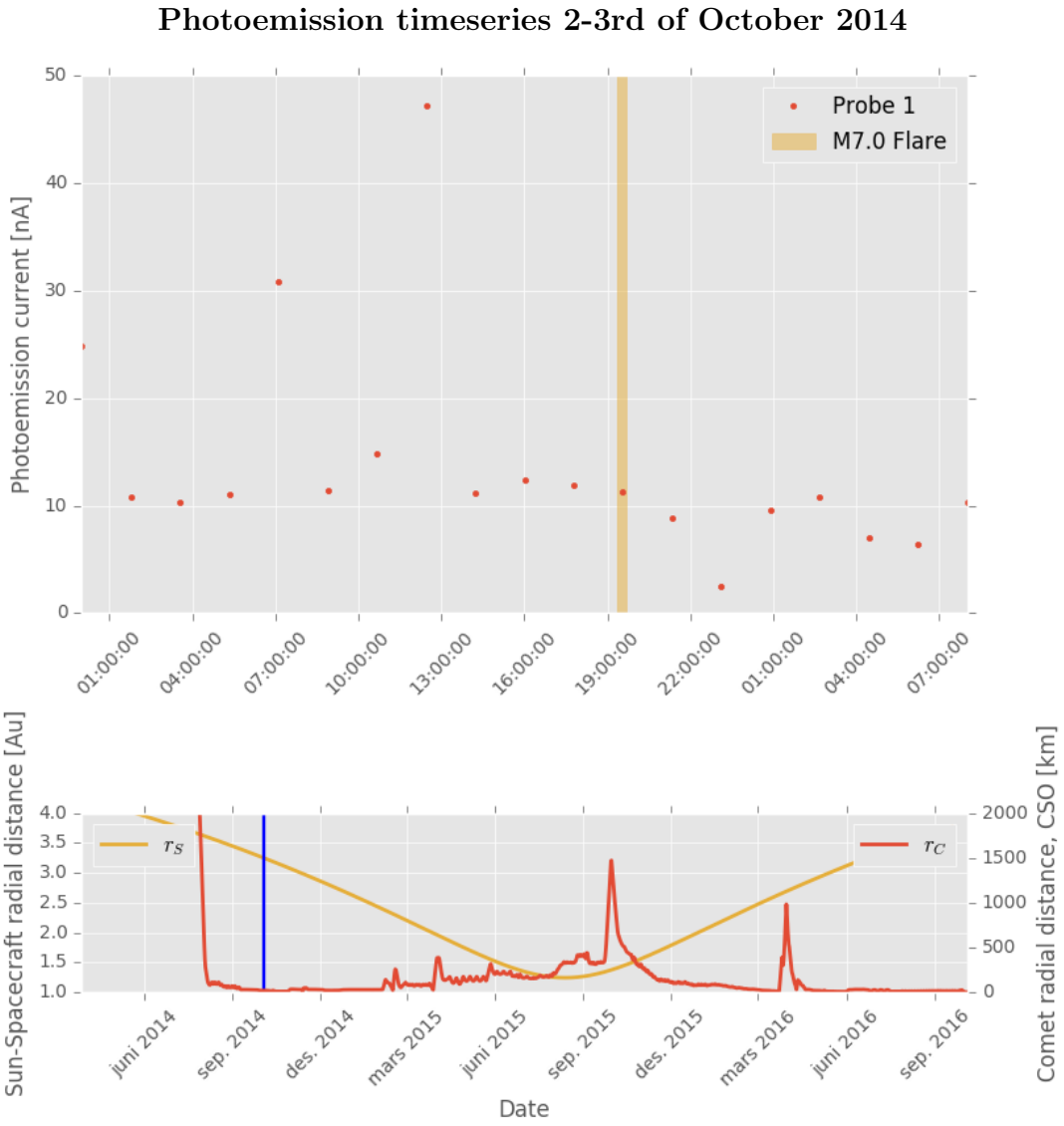


Figure 7.16: A two day photoemission timeseries for a M7.0 flare at the 2nd of September 2014. As usual the multiple sweeps method estimations are marked in red and blue dots for probe 1 and probe 2. The spacecraft positional data is given in the lower panel. There is no discernible reaction in the photoemission timeseries for this event.

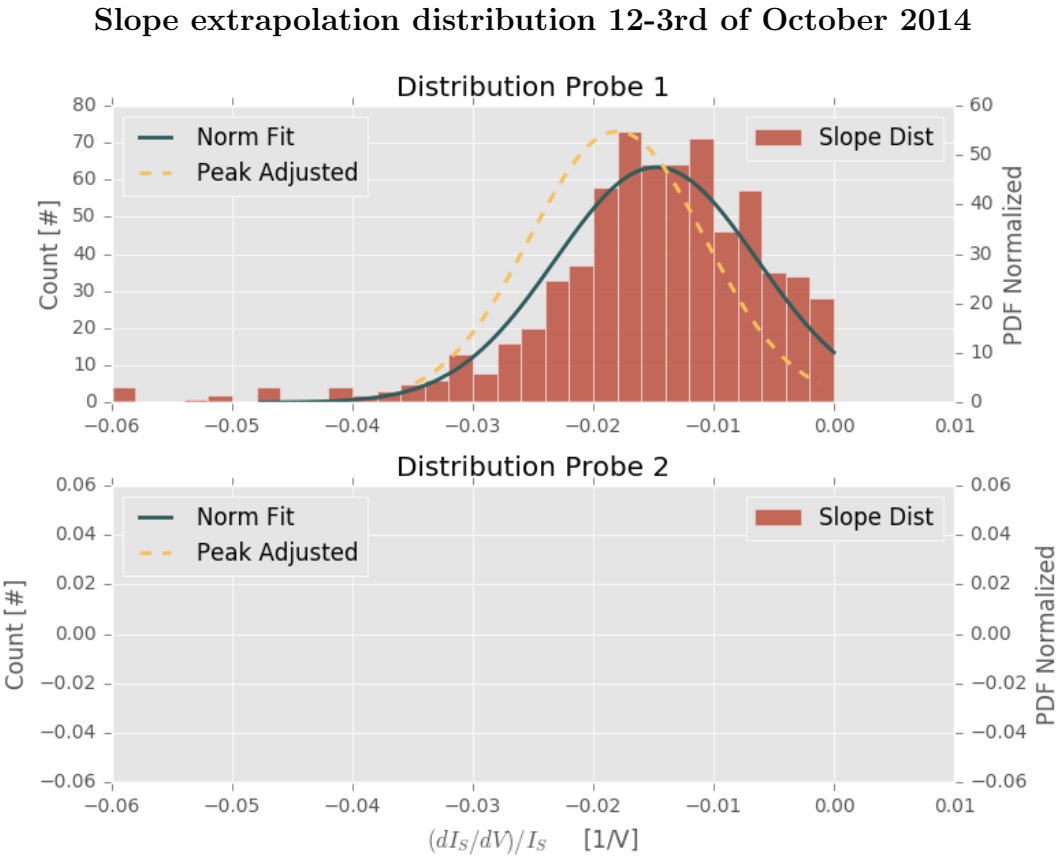


Figure 7.17: A normalized slope distribution for the period 2-3rd of October 2014. probe 1 in the top panel is the most relevant for this particular event as there is the data for probe 2 in this period.

Chapter 8

Discussion

In this chapter I will discuss the prominent aspects of the multiple sweeps method. I will talk about assumptions, strengths and weaknesses. I will emphasize that this method has room for improvement, and that that it merits further development.

Comments on the Multiple Sweeps Method

In this section general remarks concerning the performance and reliability of the multiple sweeps method are presented by topic.

Spacecraft charging

There has been a concern regarding the spacecraft charging of Rosetta. The charging of the spacecraft due to space plasma can create a spacecraft potential field, that will disturb the sensitive instruments on it. We know the spacecraft potential field definitively does affect the measurements to some degree. A simple Debye calculation $\lambda_D \approx 1.4$m can prove that the probes, even though they are placed on booms $1 - 2$m away from the body of the spacecraft are partially within the potential sheath of the spacecraft. The subject of spacecraft charging at Rosetta has been studied by Odelstad et al. (2016), where it was shown that the spacecraft for the most part is negatively charged, and that it can vary between positive and negative. Highly positive spacecraft charging is a concern, because it may invalidate our assumption of negligible electron current response in the probe. A negative spacecraft charge however, is beneficial to the method, because it further isolates the ion saturation current from energetic electron current contributions to the probe current. There is no down side to a highly negative spacecraft charge for the multiple sweeps method.

Knee potential

There are some issues with the localization of the knee potential. Remember that V_k is the bias potential for which the IV-characteristics change from the ion-saturation region to the transition region, see figure 4.1. The knee is very important for the multiple sweeps method because it serves as a fixed reference point between multiple sweeps. This means that we can gather a set of sweeps and be confident that we evaluate the same part of the characteristic, even if the current-voltage characteristic is shifted between sweeps. The characteristic will shift if for example the spacecraft itself is charged differently by some process (Odelstad et al., 2016). Issues arise for sweeps where the knee potential is hard to identify, those sweeps must be discarded as explained in section 6.3. Therefore a possible improvement to the method may be to find a solution to this problem. Perhaps it may be possible to alleviate the problem through better modeling, by for example removing a component of the sweep that muddles the knee potential, increasing the chance of identifying the knee, thus increasing the data retained. The potential knee and reference problem is tricky, it works well for the ideal sweeps, in practice however, space plasma display a diverse set of current-voltage shapes. For the truly abnormal IV-characteristics, where a sweep fit is not reliably collectible, the method breaks down, it does not in this case suffer for lack of photoemission knee.

Negative slopes

In the first step of the method, the sweep fitting step, we take a least squares linear fit to a region specified by the PKBO and PBS method input parameters. PKBO and PBS are the potential knee bias offset and potential bias span parameters. When I applied the method to the space plasma I found that contrary to what the model accounts for sometimes the slope gained from the linear fit would be negative. A negative slope is "forbidden" and cannot readily be explained by anything other than instrumental issues, noise or extensive secondary emission currents. I choose to discard negative slopes outright. However a case can be made for keeping them, the negative slopes may contribute to achieving statistical robustness. This is apparent when we are looking at the normalized slope distributions for example. There are two policies we can adapt for keeping the negative slopes:

1. Do nothing, keep all negative slopes.

2. Discard only highly negative slopes, i.e a negative slopes threshold filter.

I do not think that keeping all negative slopes is a good solution, because some of these highly negative slopes may severely interrupt the statistical distribution of step two, the slope extrapolation to zero current. The highly negative outliers

may by extension disrupt the photoemission current estimates, and lead to large errors. Albeit we already observe large spikes in the dataset, some may argue that the negative slopes therefore should be kept for transparency and completeness. The other approach is to set a low threshold and allow for some negative slopes, as long as they are very close to zero. This approach is reasonable in the sense that for some plasma regimes the slope will naturally be very close to zero, then allowing for some slightly negative slopes, may yield additional extrapolation points to increase the statistical robustness of the extrapolation step. The reason why I did not choose this path is that it is in my view most transparent to simply remove the negative slopes all together.

Discussion on the Results From Synthetic Data

As a general analysis tool, the synthetic data approach has been very good to investigate in a controlled manner different aspects of the multiple sweeps method. The most crucial aspect of the synthetic data was that it allowed us to verify the robustness of the method as well as providing a set of method input parameters that we had confidence in. In section 5.3.1 we inspected the theoretical electron current contribution to the probe current, in order to bootstrap the characterization. A similar study statistical study could be done by varying both PKBO, and PBS at the same time. I decided against this, because I think that there is merit to inspecting them one by one in the first approach to better understand their respective roles for the total error. The multiple sweeps method has from my perspective proven that it is quite flexible, take for example macro 506 from the expanded data set 11.1. The bias span of the macro is only $V_p \in [-12, 12]$V, which means that it should not be suitable for the multiple sweeps method. However it was one of the most frequently used macros from the early mission, see figure 8.1. For the results in section 7.5, we decided to include macro 506 anyway with the caveat that any results from it should be carefully scrutinized. The rest of the expanded macro set ran with PKBO = 20 and PBS = 10 the same as for the previously used, preferred science macro set. An exception was made for Macro 506 and 525, which tried to facilitate their inclusion by using PKBO = 4, and PBS = 6. An example of a set of sweeps collected under the 506 macro is given in figure 8.2

My impression of the input parameters after analyzing them using the synthetic data approach is that the method is sensitive to the chosen parameters, although not crucially so in the sense that there is a large span of input parameters that can work. This is exemplified by macro 506. This suggest that the multiple sweeps method works very well for what it is aiming for, under the assumptions that has been made. Noise, and other issues mainly arise due to assumption violations and instrumental issues. In order to make the method

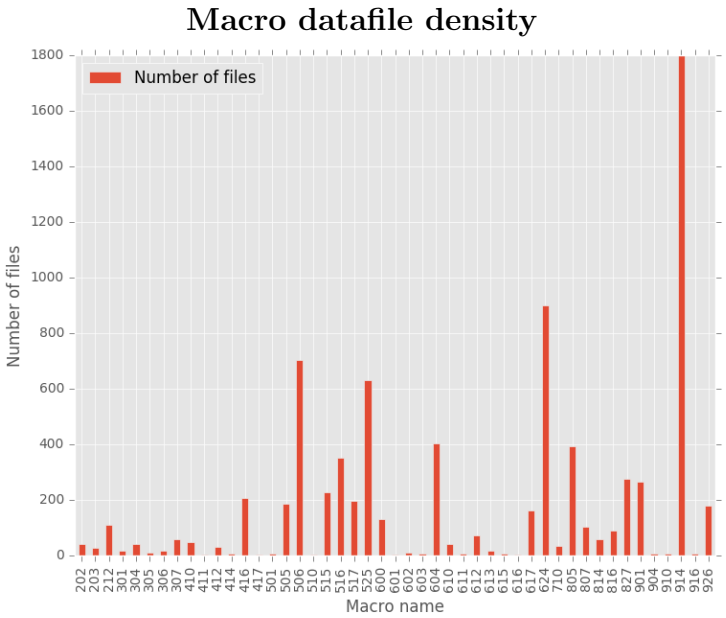


Figure 8.1: A histogram visualizing the data density for each macro. The bars represent the number of files for each macro. The different macros have different lengths, but this overview roughly translates to the total amount of data for each macro. This is not an exhaustive list of macros.

more reliable one should include a filtering mechanism directly on the individual sweeps, that check if the sweep is suitable for analysis or not. One would have to find some criterion's that lead to poorly defined sweep characteristics, a suggestion could be to filter sweeps that have abnormal shape. A filtering criterion will have to be established. One suggestion is to use a statistical analysis on every single sweep, for example the distribution approach used in this thesis, to weed out the IV-characteristics that break the model assumptions. If we can remove the model breaking sweeps, we know that the model performs well because of the work that has been done with synthetic data.

Error analysis

We were able to verify the theoretical predictions made in the first approach to the method that specifically the ion velocity and electron temperature at high densities will be consequential for the performance of the method. We investigated the accuracy of the multiple sweeps method by using synthetic data to create a data set that spanned a wide range of plasma parameters, and with this dataset testing the method. From figures 5.7, and 5.8 it is evident that the method works reasonably well in principle. The intrinsic error of the method is contained to 7nA for the cometary environment parameter space, and within 14nA for the solar wind. The electron current contamination is especially prominent in the

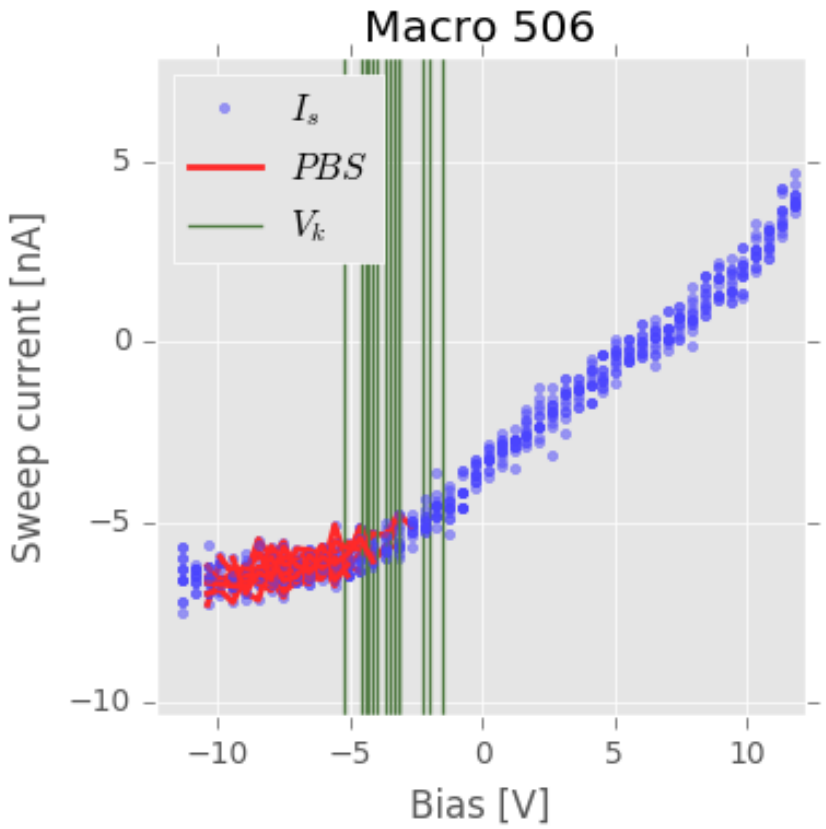


Figure 8.2: A collection of sweeps collected using the 506 macro. A large portion of the data files is using this macro.

solar wind, not because of the density but rather the high electron temperature relative to the cometary environment. This suggest that the method will struggle if electron temperatures comparable to that of the solar wind $T_e \approx 10$eV are observed at any point in the high density cometary environment.

The method show an intrinsic robustness in the parameter space, from the error analysis figures it is apparent that for the intrinsic error the choice of PKBO= 20 or PKBO= 10 does not make a big difference. We observe similar levels of noise regardless of the parameter space. This suggest that within the regime of the assumptions that has been made on density, velocity and temperature, there is room to adjust the PKBO in order to suit individual macros or regimes.

Comments on the Timeseries

It is presently not feasible to do an in depth investigation of transient events using the multiple sweeps method, for that the cadence is too low. The strength of the improved method lies in its cadence, which has been optimized to reduce the time between measurements while conserving precision.

In the photoemission article by Johansson et al. (2017) we essentially use the RPC-LAP instrument as an independent solar flux monitor in the sub 2500Årange. It has been tried before in the Pioneer mission to Venus and was found to successfully track the solar EUV-flux (Hoegy et al., 1993). Brace Hoegy and Theis found that 51% of the photoemission can be contributed to $Lyman_\alpha$ spectral line at (1216Å), while 46% was shown to come from wavelengths in the region of (550-1100 Å) the last 3% were contributed to wavelengths above $Lyman_\alpha$, meaning that the Langmuir probe covered the total EUV flux (Brace et al., 1988). The Pioneer data was shown to fluctuate in unison with variations in relation to the solar cycle and rotation as well as the the 7.2 months major harmonic to the 11 year cycle. We expected to find similar results when looking into the large scale timeseries fluctuations, and were indeed able to reproduce the periodicities observed with other instruments in the article. With the new method we tried to look at flare events on timescales down to $T \approx 1$h.

The timeseries in the article has been compared to measurements made by NASA's Mars Atmosphere and Volatile EvolutioN (MAVEN) (Eparvier et al., 2015) and Thermosphere Ionosphere Mesosphere Energetics Dynamics (TIMED) (Woods et al., 2005), which both carry instruments specifically designed to study the EUV spectrum. The different techniques complement each other and paint a picture which is hard to disprove, that there is something occulting the solar EUV-flux. Since a thorough analysis of the solar EUV-flux.

The results are based on three different analysis methods with different methodology, albeit from the same instrument. We employ the original multiple sweeps method which was developed by the Uppsala team, as well as the Sun-Shadow Transition method discussed in 11.3. The third method is a single sweep model fitting, where each sweep is compared to the model introduced in section 4.2.1. This model operates by substracting a model containing all of the non-photoemission related currents from the IV-characteristic, we can extract the photoemission. It should be noted that this method is susceptible to erroneous measurements from the sweeps, but has been binned to ascertain statistical robustness. More details on the automatic single sweep analysis routine can be found in (Odelstad et al., 2016). These three methods put together has demonstrated that the RPC-LAP instrument can measure photoemission reliably, and is not heavily influenced by

individual error sources.

In the original dataset we found large amplitude fluctuations at two different dates, coinciding with known flare activity at Rosetta, table 3.1. The new method tries to resolve these flares by attacking the problem from another angle. Instead of relying on correlating a time limited high variance period of the timeseries to the flare event, we tried to also increase the resolution and precision of the method to actually trace the time evolution of the flare in the EUV-spectrum.

During the outbound leg of the journey, the method in general slightly underestimates the photoemission current, when compared to the SST method and to the irradiance model. The most striking feature we found is that there are good correlation between to MAVEN and TIMED far away from perihelion, but when we move closer the discrepancy gradually rise to approximately 50% at perihelion. Some of these discrepancies may be attributed to surface contamination of the probes. However, the main shape and missing photoemission current can not be easily explained by any other means than the conclusions made by Johansson et al. (2017), which is that the solar flux is occulted by a dust cloud of cometary origin. Much the same shape of the timeseries is recaptured using the multiple sweeps method presented in this thesis.

Comments on the transient events

After investigating the timeseries in search of flares in section 7.4, there is not much to report as no clear flare signature has been observed. From the outset it seemed plausible that the method should be able to detect flares. The reasoning was simple, the method had initially showed promising signs, picking up photoemission currents which could be connected to the solar flux. The method showed the capacity to follow solar fluctuations on the timescales of days such as the sidereal solar rotation, and the resolution of the method using multiple sweeps promised resolutions downwards of one hour, and even lower for the burst mode macros with low cadence. So the logical next step in my mind was to push the method towards smaller timescales. For this reason we went looking for a phenomenon that would create clear signatures that was on a lower timescale than previously observed. Flares was the natural target, they are intense and have defined timescales, which provide a signature that we can look for in the photoemission timeseries.

The most obvious problems are data gaps, resolution and signal to noise ratio. Logically we cannot detect flares in the periods where there is no photoemission estimates, the lack of data naturally limits the number of flares we can compare

the timeseries with. The other large problem is a combination of the resolution and signal to noise ratio. In the multiple sweeps timeseries the signal to noise ratio is rather poor at times. We frequently see some extreme outliers, both positive and negative. When we inspect the timeseries on the scale of days, one can sometimes see that certain macros fluctuate with amplitude swings in the region of $10 - 15$nA. These may be disregarded as erratic measurements due to instrument issues, or cross instrument contamination. However, even the well behaved macros display sudden peaks of varying size. The overall erratic timeseries means that to confidently claim that a flare was observed, we need more than a few isolated data points to base the observation on. This is where the resolution issue becomes apparent. The resolution is generally quite good, all things considered, it is just not enough to observe flares on the scale $< 5 - 6$ hours. Some of the macros do have increased resolution, these are called burst mode macros, but they not used frequently. No reported flares have been reported to coincide with macros operating in burst mode.

In addition to the aforementioned reasons, another problem that may be equally important but somewhat harder to quantify is that the flare itself disturb the plasma conditions around the spacecraft. The flare may produce highly variable plasma conditions, and introduce an additional plasma population that disturbs the measurements. After all, the signatures that was reported by Johansson et al. (2017), were based on periods of strongly increased variance in the single sweep, and multiple sweeps datasets, which coincided with known flare activity.

Secondary Emission

Generally speaking the secondary emission current is due to impacting high energy particles with energies above the ionization energy of the probe surface. Such high energy impacting particles will induce emission of secondary particles, in this instance electrons from the probe surface. The impact induce multiple free electrons depending on the yield function of the probe. When the probe is at negative bias these particles are readily accelerated and constitute a current. Should the probe be positively biased at the time, the newly free particles must have overcome the potential barrier or else they will be recaptured. It is worth noting that the secondary emission is proportional to the ion and electron currents to the probe, such that when there is zero ion and electron current $I_e = I_i = 0$, the secondary emission current must also be zero $I_s = 0$. A case can be made for ignoring the secondary emission for the most part, because generally a the high energy population required for a strong secondary emission generally is not present in the cold plasma we expect to observe at 67P. However, there

may very well be times of perturbed plasma conditions in which the local plasma around the Rosetta spacecraft is for example modified by some solar transient event which could inject a high energy plasma population that will affect the probe measurements (Garnier et al., 2013). In such a case we would expect to see higher variations in the dataset as well as an apparently stronger photoemission as the method does not differentiate between secondary and photoemission currents.

Electrons are filtered by the sheath and negative potential set up by the probe bias, therefore only a small part of the electron population cause secondary emission currents. This means that the secondary electron emission current weaker than the electron current to the probe for negative bias. It is considered unlikely in a cometary environment that these electrons are numerous enough to severely affect the method. However it begs caution that electrons from such sources will decrease the slope while being decoupled from the ion current. on the other hand the effect would be to decrease the total probe current which mitigates the error introduced by the electron probe current.

Solar Energetic Particles

Solar Energetic Particles are particles which originate from solar flares, or particles that have been accelerated by transient events in the high altitude coronal atmosphere. They are analogous in their source and energy to cosmic rays of galactic origin, which are believed to be accelerated from shocks as well, albeit from shocks in relation to supernovas not a common star. It was proven by Parker in 1957 that only the solar magnetic field have the required strength to accelerate protons to the energies and in the quantities observed inside the solar system (Parker, 1957). The reason for bringing up Solar Energetic Particles in this thesis is that they may have pretty consequential impacts on comets, it has been suggested that given the right conditions SEP's may create a macroscopic positive electric double-layer on the surface of comets similar to 67P/C-G (Ibadov, 2012). It is expected that a SEP event registered at Rosetta will cause an increased photoemission current signature in RPC-LAP for extended periods of 1-10 hours. There is a SEP event reported by the RPC team at Rosetta from the 1st to the 9th of September 2014, coinciding with a CME and flare activity McKenna-Lawlor et al. (2016).

Chapter 9

Future Work

Inspecting Spacecraft Potential

A consistent concern is the effect of spacecraft charging on the RPC-LAP instrument. In our case we are especially interested in verifying that the multiple sweeps method results are reliable. It would be an interesting study from an instrumental, and method development perspective to simply correlate periods of highly positive spacecraft charging with periods of increased noise from the multiple sweeps method. An improvement to the method may be to discard data outright during times of highly positive spacecraft potential to improve reliability and perhaps reduce outliers.

A Statistics Investigation

A closer look at the timeseries distributions using statistics might be fruitful with respect to uncovering hidden problems with method. I am thinking that it could be interesting to look closer at the heteroscedascticity of the distributions. Space plasma can sometimes produce spurius events and the measurements may be influenced by different particle populations. It is not farfetched to think that the variance is not exactly proportional to the mean of the given dataset, in that case a weighted least-squares method might be more suitable.

Furthermore the normalized slope distributions have shown promising signs. The distribution technique has not been fully taken advantage of yet, as it was just recently decided to work with them. We are thinking about more ways to utilize them as of writing.

Transient Events for Further Study

Naturally there are many transient events during a two year mission, as the method matures at some point there might be merit to investigating transient events in more detail. The multiple sweeps method may be used to track solar transient events in the future, perhaps providing real time monitoring and warning for other systems. The method in itself does not have the required cadence to properly resolve solar flares at the present time, but I am confident that it could be possible in the future.

Chapter 10

Conclusions

Last remarks

When we work with developing a novel method for analyzing a complex system such as a space plasma using a very sensitive instrument such as the RPC-LAP, we must be very careful in interpreting the data. It is important emphasize that the statistical methods employed in this thesis is well grounded and handled with integrity. It is my opinion that the results we present in this thesis are accurate and that the method work the way it is expected to. However I want to specify that any conclusions drawn from working with the multiple sweeps method must be scrutinized, it is very easy to make errors of judgment when interpreting results to draw physical conclusions.

We found that the multiple sweeps method developed in this thesis is in good agreement with other similar studies, i.e the study performed by Johansson et al. (2017). The method has been characterized, we can now choose input parameters with confidence, for both the solar wind and the cometary plasma conditions, as well as adjust the method to accommodate different macro settings. The methods intrinsic error has been investigated in different plasma conditions has been investigated using synthetic data, the analysis suggest that the method is robust, as well as precise.

We have applied the multiple sweeps method to a real data set within a complex plasma environment across varying plasma regimes. At times the assumptions which the method rests upon may be broken, from which we learned that care must be taken to minimize risk of inaccurate interpretation of results. We were not able to definitively observe flares using the method at this stage in its development. Although it may be possible to do so in the future. I suggest further work on the use of normalized slope distributions to further the development of the method and its capabilities.

Bibliography

A'Hearn, M. F., Belton, M. J. S., Delamere, W. A., Kissel, J., Klaasen, K. P., McFadden, L. A., Meech, K. J., Melosh, H. J., Schultz, P. H., Sunshine, J. M., Thomas, P. C., Veverka, J., Yeomans, D. K., Baca, M. W., Busko, I., Crockett, C. J., Collins, S. M., Desnoyer, M., Eberhardy, C. A., Ernst, C. M., Farnham, T. L., Feaga, L., Groussin, O., Hampton, D., Ipatov, S. I., Li, J.-Y., Lindler, D., Lisse, C. M., Mastrodemos, N., Owen, W. M., Richardson, J. E., Wellnitz, D. D., and White, R. L. (2005). Deep Impact: Excavating Comet Tempel 1. *Science*, 310(5746):258–264.

Alfvén, H. (1957). On the Theory of Comet Tails. *Tellus*, 9(1):92–96.

Baumjohann, W. and Treumann, R. A. (1997). *Basic Space Plasma Physics*. World Scientific. Google-Books-ID: e4yupcOzJxkC.

Behar, E., Lindkvist, J., Nilsson, H., Holmstrm, M., Stenberg-Wieser, G., Ramstad, R., and Gtz, C. (2016). Mass-loading of the solar wind at 67p/Churyumov-Gerasimenko - Observations and modelling. *A&A*, 596:A42.

Bibring, J.-P., Rosenbauer, H., Boehnhardt, H., Ulamec, S., Biele, J., Espinasse, S., Feuerbacher, B., Gaudon, P., Hemmerich, P., Kletzkine, P., Moura, D., Mugnuolo, R., Nietner, G., Patz, B., Roll, R., Scheuerle, H., Szego, K., and Wittmann, K. (2007). The rosetta lander (Philae) investigations. *Space science reviews*, 128(1-4):205–220.

Bodewits, D., Lara, L. M., AHearn, M. F., Forgia, F. L., Gicquel, A., Kovacs, G., Knollenberg, J., M. Lazzarin, (), Z.-Y. L., Shi, X., Snodgrass, C., Tubiana, C., Sierks, H., Barbieri, C., Lamy, P. L., R. Rodrigo, Koschny, D., Rickman, H., Keller, H. U., Barucci, M. A., Bertaux, J.-L., Bertini, I., Boudreault, S., G. Cremonese, Deppo, V. D., Davidsson, B., Debei, S., Cecco, M. D., Fornasier, S., Fulle, M., Groussin, O., Gutirrez, P. J., Gttler, C., Hviid, S. F., Ip, W.-H., Jorda, L., Kramm, J.-R., Khrt, E., Kppers, M., Lpez-Moreno, J. J., Marzari, F., Naletto, G., Oklay, N., Thomas, N., Toth, I., and Vincent, J.-B. (2016). Changes in the Physical Environment of the Inner Coma of 67p/ChuryumovGerasimenko with Decreasing Heliocentric Distance. *AJ*, 152(5):130.

Brace, L. H., Hoegy, W. R., and Theis, R. F. (1988). Solar EUV measurements at Venus based on photoelectron emission from the Pioneer Venus Langmuir Probe. *J. Geophys. Res.*, 93(A7):7282–7296.

Brownlee, D. E., Tsou, P., Anderson, J. D., Hanner, M. S., Newburn, R. L., Sekanina, Z., Clark, B. C., Hrz, F., Zolensky, M. E., Kissel, J., McDonnell, J. a. M., Sandford, S. A., and Tuzzolino, A. J. (2003). Stardust: Comet and interstellar dust sample return mission. *J. Geophys. Res.*, 108(E10):8111.

Burch, J. L., Goldstein, R., Cravens, T. E., Gibson, W. C., Lundin, R. N., Pollock, C. J., Winningham, J. D., and Young, D. T. (2007). RPC-IES: The Ion and Electron Sensor of the Rosetta Plasma Consortium. *Space Sci Rev*, 128(1-4):697–712.

Carr, C., Cupido, E., Lee, C. G. Y., Balogh, A., Beek, T., Burch, J. L., Dunford, C. N., Eriksson, A. I., Gill, R., Glassmeier, K. H., Goldstein, R., Lagoutte, D., Lundin, R., Lundin, K., Lybekk, B., Michau, J. L., Musmann, G., Nilsson, H., Pollock, C., Richter, I., and Trotignon, J. G. (2007). RPC: The Rosetta Plasma Consortium. *Space Sci Rev*, 128(1-4):629–647.

Chamberlin, P. C., Milligan, R. O., and Woods, T. N. (2012). Thermal Evolution and Radiative Output of Solar Flares Observed by the EUV Variability Experiment (EVE). *Sol Phys*, 279(1):23–42.

Child, C. D. (1911). Discharge From Hot CaO. *Phys. Rev. (Series I)*, 32(5):492–511.

Chung, K.-S. and Hutchinson, I. H. (1988). Kinetic theory of ion collection by probing objects in flowing strongly magnetized plasmas. *Phys. Rev. A*, 38(9):4721–4731.

Combi, M. R., Harris, W. M., and Smyth, W. H. (2004). Gas dynamics and kinetics in the cometary coma: Theory and observations. In *In Comets II*.

Edberg, N. J. T., Andrews, D. J., Burch, J. L., Carr, C. M., Cupido, E., Eriksson, A. I., Glassmeier, K.-H., Goldstein, R., Henri, P., Koenders, C., Mandt, K., Nilsson, H., Odelstad, E., Stenberg Wieser, G., and Vigren, E. (2016). CME impact on comet 67p. volume 18, page 7343.

Edberg, N. J. T., Eriksson, A. I., Odelstad, E., Henri, P., Lebreton, J.-P., Gasc, S., Rubin, M., Andr, M., Gill, R., Johansson, E. P. G., Johansson, F., Vigren, E., Wahlund, J. E., Carr, C. M., Cupido, E., Glassmeier, K.-H., Goldstein, R., Koenders, C., Mandt, K., Nemeth, Z., Nilsson, H., Richter, I., Wieser, G. S., Szego, K., and Volwerk, M. (2015). Spatial distribution of low-energy plasma around comet 67p/CG from Rosetta measurements. *Geophys. Res. Lett.*, 42(11):2015GL064233.

Einstein, A. (1905). ber einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. *Ann. Phys.*, 322(6):132–148.

Eparvier, F. G., Chamberlin, P. C., Woods, T. N., and Thiemann, E. M. B. (2015). The Solar Extreme Ultraviolet Monitor for MAVEN. *Space Sci Rev*, 195(1-4):293–301.

Eriksson, A. I., Bostrm, R., Gill, R., hln, L., Jansson, S.-E., Wahlund, J.-E., Andr, M., Mlkki, A., Holtet, J. A., Lybekk, B., Pedersen, A., Blomberg, L. G., and Team, T. L. (2007). RPC-LAP: The Rosetta Langmuir Probe Instrument. *Space Sci Rev*, 128(1-4):729–744.

Eriksson, A. I., Engelhardt, I. a. D., Andr, M., Bostrm, R., Edberg, N. J. T., Johansson, F. L., Odelstad, E., Vigren, E., Wahlund, J.-E., Henri, P., Lebreton, J.-P., Miloch, W. J., Paulsson, J. J. P., Wedlund, C. S., Yang, L., Karlsson, T., Jarvinen, R., Broiles, T., Mandt, K., Carr, C. M., Galand, M., Nilsson, H., and Norberg, C. (2017). Cold and warm electrons at comet 67p/Churyumov-Gerasimenko. *A&A*, 605:A15.

ESA (2014). Rosetta's first sighting of its target in 2014 narrow angle view.

Festou, M., Keller, H. U., and Weaver, H. A. (2004). *Comets II*. University of Arizona Press. Google-Books-ID: AHF9ZraafV8C.

Fulle, M., Altobelli, N., Buratti, B., Choukroun, M., Fulchignoni, M., Grn, E., Taylor, M. G. G. T., and Weissman, P. (2016). Unexpected and significant findings in comet 67p/ChuryumovGerasimenko: an interdisciplinary view. *MNRAS*, 462(Suppl 1):S2–S8.

Galand, M., Hritier, K. L., Odelstad, E., Henri, P., Broiles, T. W., Allen, A. J., Altwegg, K., Beth, A., Burch, J. L., Carr, C. M., Cupido, E., Eriksson, A. I., Glassmeier, K.-H., Johansson, F. L., Lebreton, J.-P., Mandt, K. E., Nilsson, H., Richter, I., Rubin, M., Sagnires, L. B. M., Schwartz, S. J., Smon, T., Tzou, C.-Y., Vallires, X., Vigren, E., and Wurz, P. (2016). Ionospheric plasma of comet 67p probed by Rosetta at 3 au from the Sun. *Mon Not R Astron Soc*, 462(Suppl_1):S331–S351.

Garnier, P., Holmberg, M. K. G., Wahlund, J.-E., Lewis, G. R., Grimald, S. R., Thomsen, M. F., Gurnett, D. A., Coates, A. J., Crary, F. J., and Dandouras, I. (2013). The influence of the secondary electrons induced by energetic electrons impacting the Cassini Langmuir probe at Saturn. *J. Geophys. Res. Space Physics*, 118(11):7054–7073.

Glassmeier, K.-H., Boehnhardt, H., Koschny, D., Khrt, E., and Richter, I. (2007a). The Rosetta Mission: Flying Towards the Origin of the Solar System. *Space Sci Rev*, 128(1-4):1–21.

Glassmeier, K.-H., Richter, I., Diedrich, A., Musmann, G., Auster, U., Motschmann, U., Balogh, A., Carr, C., Cupido, E., Coates, A., Rother, M., Schwingenschuh, K., Szeg, K., and Tsurutani, B. (2007b). RPC-MAG The Fluxgate Magnetometer in the ROSETTA Plasma Consortium. *Space Sci Rev*, 128(1-4):649–670.

Grard, R. J. L. (1973). Properties of the satellite photoelectron sheath derived from photoemission laboratory measurements. *J. Geophys. Res.*, 78(16):2885–2906.

Hansen, K. C., Bagdonat, T., Motschmann, U., Alexander, C., Combi, M. R., Cravens, T. E., Gombosi, T. I., Jia, Y.-D., and Robertson, I. P. (2007). The Plasma Environment of Comet 67p/Churyumov-Gerasimenko Throughout the Rosetta Main Mission. *Space Sci Rev*, 128(1-4):133–166.

Hapgood, M. A. (2011). Towards a scientific understanding of the risk from extreme space weather. *Advances in Space Research*, 47(12):2059–2072.

Harrison, R. (1995). The nature of solar flares associated with coronal mass ejections. *Astronomy and Astrophysics*, 304:585–594.

Henri, P., Broiles, T., Eriksson, A., Bghin, C., Lebreton, J.-P., Vallieres, X., More, J., Wattieaux, G., Engelhardt, I. A. D., Edberg, N., Odelstad, E., Vigren, E., Glassmeier, K.-H., Goetz, C., Koenders, C., Richter, I., Volwerk, M., Burch, J. L., Goldstein, R., and Mandt, K. (2016). Localised plasma density enhancements around comet CG/67p. volume 18, page 16587.

Hoegy, W. R., Pesnell, W. D., Woods, T. N., and Rottman, G. J. (1993). How active was solar cycle 22? *Geophys. Res. Lett.*, 20(13):1335–1338.

Huddlestone, R. H. (1965). *Plasma diagnostic techniques*. Academic Press.

Huddlestone, R. H. and Chen, F. F. (1965). *Plasma diagnostic techniques*. Academic Press, chapter 4 edition.

Hutchinson, I. H. (2002). Principles of Plasma Diagnostics: Second Edition. *Plasma Phys. Control. Fusion*, 44(12):2603.

Ibadov, S. (2012). Space observations of comets during solar flares: A possible explanation for comet brightness outbursts. *Advances in Space Research*, 49(3):467–470.

J Lin and T G Forbes (2000). Effects of reconnection on the coronal mass ejection process. *Journal of geophysical research*, 105(A2):2375–2392.

Johansson, F. L., Odelstad, E., Paulsson, J. J. P., Harang, S. S., Eriksson, A. I., Mannel, T., Vigren, E., Edberg, N. J. T., Miloch, W. J., Wedlund, C. S., Thiemann, E., Eparvier, F., and Andersson, L. (2017). Rosetta photoelectron emission and solar ultraviolet flux at comet 67p. *arXiv:1709.03874 [astro-ph, physics:physics]*. arXiv: 1709.03874.

Johlander, A. (2012). *Photoemission on the Rosetta spacecraft*.

Jorda, L., Gaskell, R., Capanna, C., Hviid, S., Lamy, P., urech, J., Faury, G., Groussin, O., Gutirrez, P., Jackman, C., Keihm, S. J., Keller, H. U., Knollenberg, J., Khrt, E., Marchi, S., Mottola, S., Palmer, E., Schloerb, F. P., Sierks, H., Vincent, J. B., AHearn, M. F., Barbieri, C., Rodrigo, R., Koschny, D., Rickman, H., Barucci, M. A., Bertaux, J. L., Bertini, I., Cremonese, G., Da Deppo, V., Davidsson, B., Debei, S., De Cecco, M., Fornasier, S., Fulle, M., Gttler, C., Ip, W. H., Kramm, J. R., Kppers, M., Lara, L. M., Lazzarin, M., Lopez Moreno, J. J., Marzari, F., Naletto, G., Oklay, N., Thomas, N., Tubiana, C., and Wenzel, K. P. (2016). The global shape, density and rotation of Comet 67p/Churyumov-Gerasimenko from preperihelion Rosetta/OSIRIS observations. *Icarus*, 277(Supplement C):257–278.

Kane, R. P. (2002). Some Implications Using the Group Sunspot Number Reconstruction. *Solar Physics*, 205(2):383–401.

Kittel, C. (1967). Introduction to Solid State Physics. *American Journal of Physics*, 35(6):547–548.

Leubner, M. P. (2004). Fundamental issues on kappa-distributions in space plasmas and interplanetary proton distributions. *Physics of Plasmas*, 11(4):1308–1316.

Lochte-Holtgreven, W. and Richter, J. (1968). *Plasma diagnostics.* North-Holland Pub. Co., Amsterdam. OCLC: 439496.

McKenna-Lawlor, S., Ip, W., Jackson, B., Odstrcil, D., Nieminen, P., Evans, H., Burch, J., Mandt, K., Goldstein, R., Richter, I., and Dryer, M. (2016). Space Weather at Comet 67p/ChuryumovGerasimenko Before its Perihelion. *Earth Moon Planets*, 117(1):1–22.

Medicus, G. (1961). Theory of Electron Collection of Spherical Probes. *Journal of Applied Physics*, 32(12):2512–2520.

Melzer, Andre (2017). Introduction to Colloidal (Dusty) Plasmas.

Mendis, D. A. and Houpis, H. L. F. (1982). The cometary atmosphere and its interaction with the solar wind. *Rev. Geophys.*, 20(4):885–928.

Merlino, R. L. (2007). Understanding Langmuir probe current-voltage characteristics. *American Journal of Physics*, 75(12):1078.

Mott-Smith, H. M. and Langmuir, I. (1926). The Theory of Collectors in Gaseous Discharges. *Phys. Rev.*, 28(4):727–763.

Munro, R. H., Gosling, J. T., Hildner, E., MacQueen, R. M., Poland, A. I., and Ross, C. L. (1979). The association of coronal mass ejection transients with other forms of solar activity. *Sol Phys*, 61(1):201–215.

Nagaoka, K., Okamoto, A., Yoshimura, S., and Tanaka, M. Y. (2001). Plasma Flow Measurement Using Directional Langmuir Probe Under Weakly Ion-Magnetized Conditions. *Journal of the Physical Society of Japan*, 70(1):131–137.

Nilsson, H., Lundin, R., Lundin, K., Barabash, S., Borg, H., Norberg, O., Fedorov, A., Sauvaud, J.-A., Koskinen, H., Kallio, E., Riihel, P., and Burch, J. L. (2007). RPC-ICA: The Ion Composition Analyzer of the Rosetta Plasma Consortium. *Space Sci Rev*, 128(1-4):671–695.

Odelstad, E., Stenberg-Wieser, G., Wieser, M., Eriksson, A., Nilsson, H., and Johansson, F. (2016). Measurements of the electrostatic potential of Rosetta at comet 67p. In *14th Spacecraft Charging Technology Conference, ESA/ESTEC, Noordwijk, NL, 04-08 APRIL 2016*, pages Abstract–123. ESA Publications Division, European Space Agency.

Parker, E. N. (1957). Acceleration of Cosmic Rays in Solar Flares. *Phys. Rev.*, 107(3):830–836.

Pécseli, H. L. (2012). *Waves and Oscillations in Plasmas*. Series in Plasma Physics. Taylor & Francis. DOI: 10.1201/b12702 DOI: 10.1201/b12702.

Pesnell, W. D. (2015). Solar Dynamics Observatory (SDO). In Pelton, J. N. and Allahdadi, F., editors, *Handbook of Cosmic Hazards and Planetary Defense*, pages 179–196. Springer International Publishing. DOI: 10.1007/978-3-319-03952-7_16.

Prölss, G. (2012). *Physics of the Earths Space Environment: An Introduction*. Springer Science & Business Media. Google-Books-ID: HvD9CAAAQBAJ.

Riemann, K.-U. (1991). The Bohm criterion and sheath formation. *J. Phys. D: Appl. Phys.*, 24(4):493.

Ronald L Moore (1988). Evidence that magnetic energy shedding in solar filament eruptions is the drive in accompanying flares and coronal mass ejections. *The Astrophysical journal*, 324:1132.

Rottmann, K. (1960). *Mathematische Formelsammlung*.

Schmieder, B. and Aulanier, G. (2012). What are the physical mechanisms of eruptions and CMEs? *Advances in Space Research*, 49(11):1598–1606.

Schulz, R. (2009). Rosettaone comet rendezvous and two asteroid fly-bys. *Sol Syst Res*, 43(4):343–352.

Schwenn, R. (1996). An Essay on Terminology, Myths, - and Known Facts: Solar Transient - Flare - CME - Driver Gas - Piston - BDE - Magnetic Cloud - Shock Wave - Geomagnetic Storm. *International Astronomical Union Colloquium*, 154:187–193.

Tandberg-Hanssen, E. (1974). *Formation of Prominences - Springer*, volume 12 of *Geophysics and Astrophysics Monographs*.

Taylor, H. (1958). Irving Langmuir, 1881-1957. *Biographical Memoirs of Fellows of the Royal Society*, 4:167–184.

Taylor, M. G. G. T., Alexander, C., Altobelli, N., Fulle, M., Fulchignoni, M., Grn, E., and Weissman, P. (2015). Rosetta begins its Comet Tale. *Science*, 347(6220):387–387.

Trotignon, J. G., Michau, J. L., Lagoutte, D., Chabassire, M., Chalumeau, G., Colin, F., Dcrau, P. M. E., Geiswiller, J., Gille, P., Grard, R., Hachemi, T., Hamelin, M., Eriksson, A., Laakso, H., Lebreton, J. P., Mazelle, C., Randriamboarison, O., Schmidt, W., Smit, A., Telljohann, U., and Zamora, P. (2007). RPC-MIP: the Mutual Impedance Probe of the Rosetta Plasma Consortium. *Space Sci Rev*, 128(1-4):713–728.

Vasile, M. and Pascale, P. D. (2006). Preliminary Design of Multiple Gravity-Assist Trajectories. *Journal of Spacecraft and Rockets*, 43(4):794–805.

Vigren, E., Altwegg, K., Edberg, N. J. T., Eriksson, A. I., Galand, M., Henri, P., Johansson, F., Odelstad, E., Tzou, C.-Y., and Vallires, X. (2016). Model-Observation Comparisons of Electron Number Densities in the Coma of 67p/Churyumov-Gerasimenko during January 2015. *The Astronomical Journal*, 152:59.

Vigren, E. and Galand, M. (2013). Predictions of Ion Production Rates and Ion Number Densities within the Diamagnetic Cavity of Comet 67p/Churyumov-Gerasimenko at Perihelion. *The Astrophysical Journal*, 772:33.

Wang, Y.-M. and Colaninno, R. (2014). Is Solar Cycle 24 Producing More Coronal Mass Ejections Than Cycle 23? *ApJL*, 784(2):L27.

Weaver, H. A., Feldman, P. D., A'Hearn, M. F., Arpigny, C., Brandt, J. C., Festou, M. C., Haken, M., McPhate, J. B., Stern, S. A., and Tozzi, G. P. (1997). The Activity and Size of the Nucleus of Comet Hale-Bopp (C/1995 O1). *Science*, 275(5308):1900–1904.

Wedlund, C. S., Kallio, E., Alho, M., Nilsson, H., Wieser, G. S., Gunell, H., Behar, E., Pusa, J., and Gronoff, G. (2016). The atmosphere of comet 67p/Churyumov-Gerasimenko diagnosed by charge-exchanged solar wind alpha particles. *A&A*, 587:A154.

Westlinder, J., Sjblom, G., and Olsson, J. (2004). Variable work function in MOS capacitors utilizing nitrogen-controlled TiNx gate electrodes. *Microelectronic Engineering*, 75(4):389–396.

Whipple, E. C. (1981). Potentials of surfaces in space. *Rep. Prog. Phys.*, 44(11):1197.

Whipple, F. L. (1950). A comet model. I. The acceleration of Comet Encke. *The Astrophysical Journal*, 111:375–394.

Woods, T. N., Eparvier, F. G., Bailey, S. M., Chamberlin, P. C., Lean, J., Rottman, G. J., Solomon, S. C., Tobiska, W. K., and Woodraska, D. L. (2005). Solar EUV Experiment (SEE): Mission overview and first results. *J. Geophys. Res.*, 110(A1):A01312.

Yang, L., Paulsson, J. J. P., Wedlund, C. S., Odelstad, E., Edberg, N. J. T., Koenders, C., Eriksson, A. I., and Miloch, W. J. (2016). Observations of high-plasma density region in the inner coma of 67p/ChuryumovGerasimenko during early activity. *MNRAS*, 462(Suppl 1):S33–S44.

Zhang, J., Dere, K. P., Howard, R. A., Kundu, M. R., and White, S. M. (2001). On the Temporal Relationship between Coronal Mass Ejections and Flares. *ApJ*, 559(1):452.

Chapter 11

Appendix A

Supplementary Overview Figures

In figure 11.1, I have really pushed the method to the limit of what can be considered reasonable in terms of reliability and statistical significance. When doing event analysis I found that there is merit to adjusting the resolution of both the method itself, as well as any fitting routines that is applied to the event in question. Looking at high level stuff, I found that the low resolution method is well suited, it has the fewest but also the most reliable data points. Since small scale variation are picked up with this method, you must at some point introduce a smoothing in order to give the data a clear cut and meaningful presentation. It is preferable with this method to introduce the first level of smoothing directly into to the method by increasing the number of sweep considered for each I_{Ph} estimation, subsequently decreasing the resolution but increasing the precision. The resolution values are given in table 6.2, and discussed in section 6.1.1.

Macro Information

In table 11.1, we have listed all of the macros used to increase the coverage for the early mission in result section 7.5. These are all the macros from the preferred science macros set, with some additional macros which should have reasonable performance as well. I have removed some of the preferred science macros that did not contain enough data to produce any I_{ph} estimates. Notable inclusions are macros 506 and 525, with varying success. Figure 11.3 from the biweekly distributions review, suggest that macro 525 mostly introduced noise to the dataset. It was probably a mistake to include it in the first place.

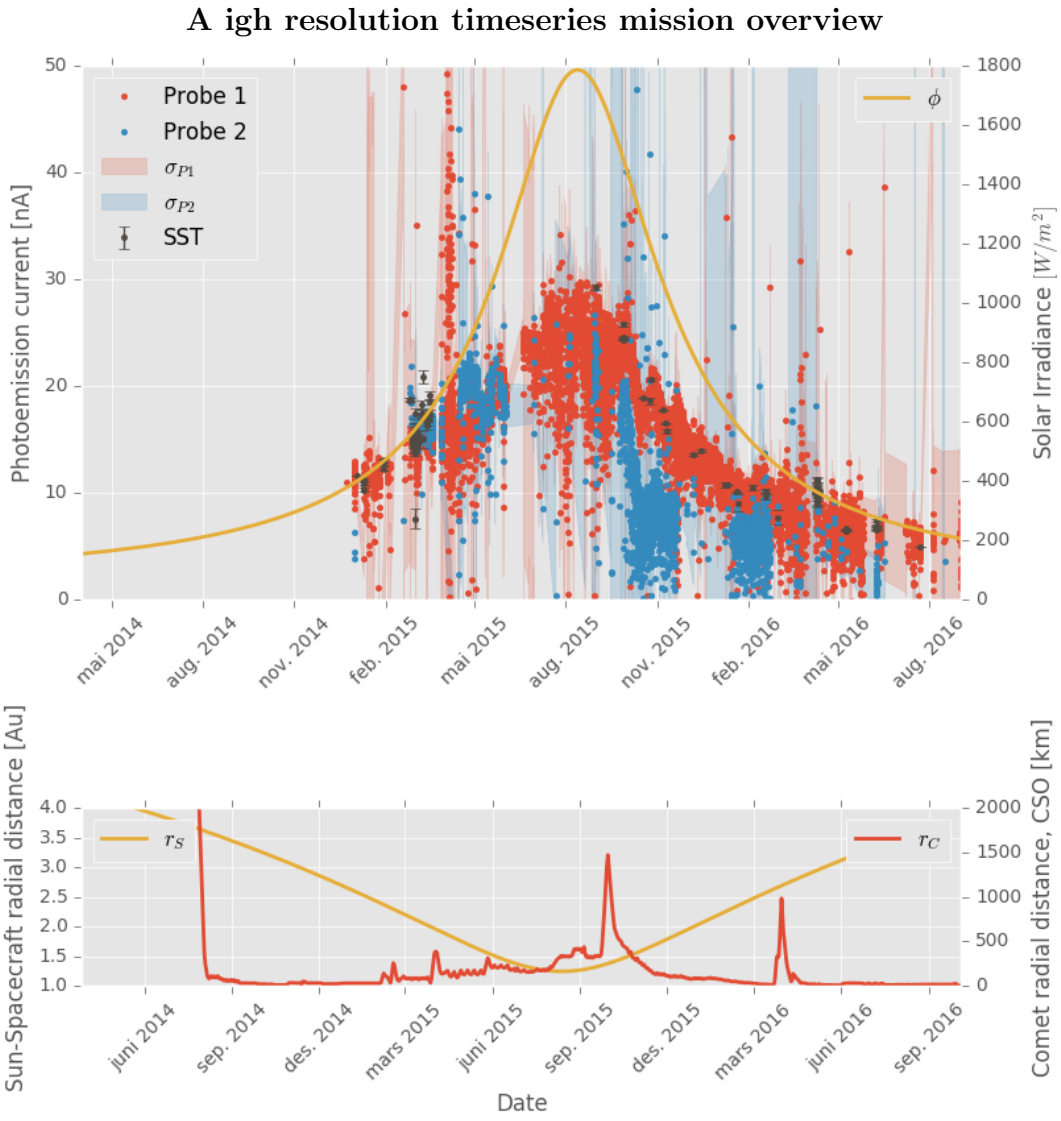


Figure 11.1: A photoemission current timeseries from Rosetta's escort of the comet 67P/C-G. In panel one the multiple sweeps method of probe 1 and probe 2 as well as the Sun-Shadow Transition method. A simple solar irradiance model, denoted with ϕ, is added for reference. In the lower panel, the spacecraft radial distance with respect to the comet r_c and the Sun is traced with red and yellow lines.

Introduction to the Sun-Shadow Transition Method

The current-step or Sun-Shadow transition method developed by Elias Odelstad of the Department of Physics and Astronomy, Uppsala University, Sweden has been used to validate the multiple sweeps method. The method is considered

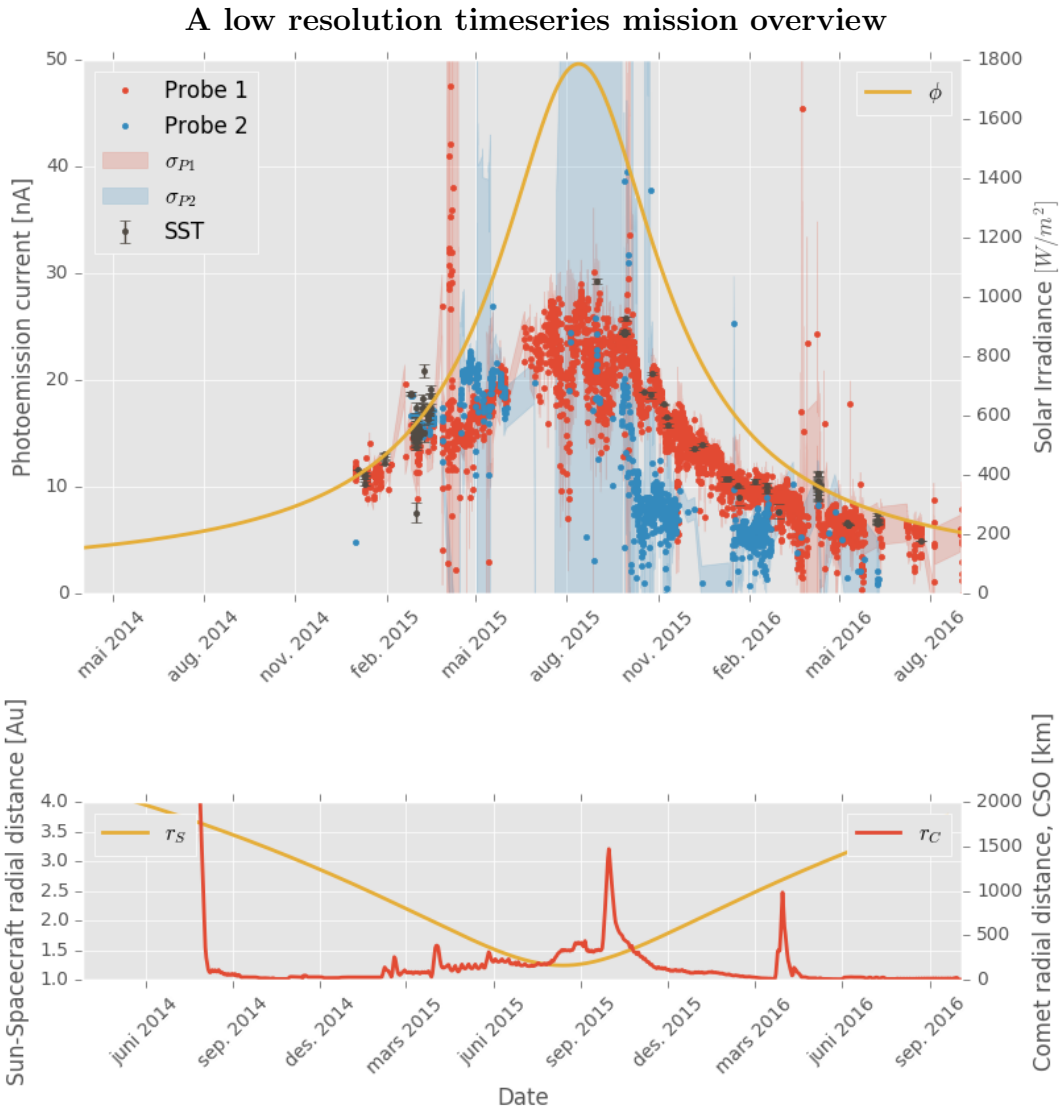


Figure 11.2: A photoemission current timeseries from Rosetta's escort of the comet 67P/C-G. In panel one the multiple sweeps method of probe 1 and probe 2 as well as the Sun-Shadow Transition method. A simple solar irradiance model, denoted with ϕ, is added for reference. In the lower panel, the spacecraft radial distance with respect to the comet r_c and the Sun is traced with red and yellow lines.

very robust and reliable, unfortunately it has poor coverage, making it unreliable for data analysis, but useful as a reference tool. The reason for the poor coverage is obvious when you consider the working principle of the method. As is implied by the name Sun-Shadow Transition, the method relies on the Langmuir probe to Transition out of the light into shadows cast by the satellite solar panels. Given sufficiently negative probe potential, we should be able to detect an current shift

Table 11.1: Expanded macro list, most notable inclusions are 506, and 525 which contain a lot of data, but must be treated with special care.

Macro	Bias [V]	Step [V]	Cadence [s]	Trusted
506	12, -12	0.5	160	No
516	-30, 30	0.5	160	No
517	-30, 30	0.5	160	Yes
525	31, -17	0.5	160	No
604	-30, 20	0.25	96	No
610	-30, 30	0.5	160	No
612	-30, 30	0.5	160	Yes
613	-30, 30	0.5	160	No
615	-30, 30	0.25	160	Yes
624	-30, 30	0.25	160	Yes
901	-30, 30	0.25	160	No
910	-28, 28	0.5	160	No
914	-28, 28	0.25	160	Yes

when moving from the sunlit photoemission regime, into the shadows where there can be no photoemission. The method is subject to a few conditions that must be met. First of all LAP must be as previously mentioned at negative potential with respect to the spacecraft, this is in order to avoid any shielding of photoemission current, which in effect set a bias constraint on the analysis. The samples must be taken at times where there are no violent changes in the plasma, or if there are, they must be short lived so they can be easily filtered. The method is limited transitions which are from fully sunlit to completely shadowed in less than two minutes as well. For the current shift estimate a two minutes average is computed before and after the transition.

One example of a transition is given in figure 11.4, notice the two minute data sampling before and after the transition which in this case took a little over a minute.

Inspecting the Timeseries at the beginning of April 2015.

In figure 11.5, the photoemission series at the beginning of April has been plotted using the preferred science macros. We see that for probe 1 the timeseries photoemission experience a large offset from the expected photoemission level, as well as discrepancy with respect to the timeseries immediately preceding and af-

Normalized slope distribution sorted by macro for May 2016

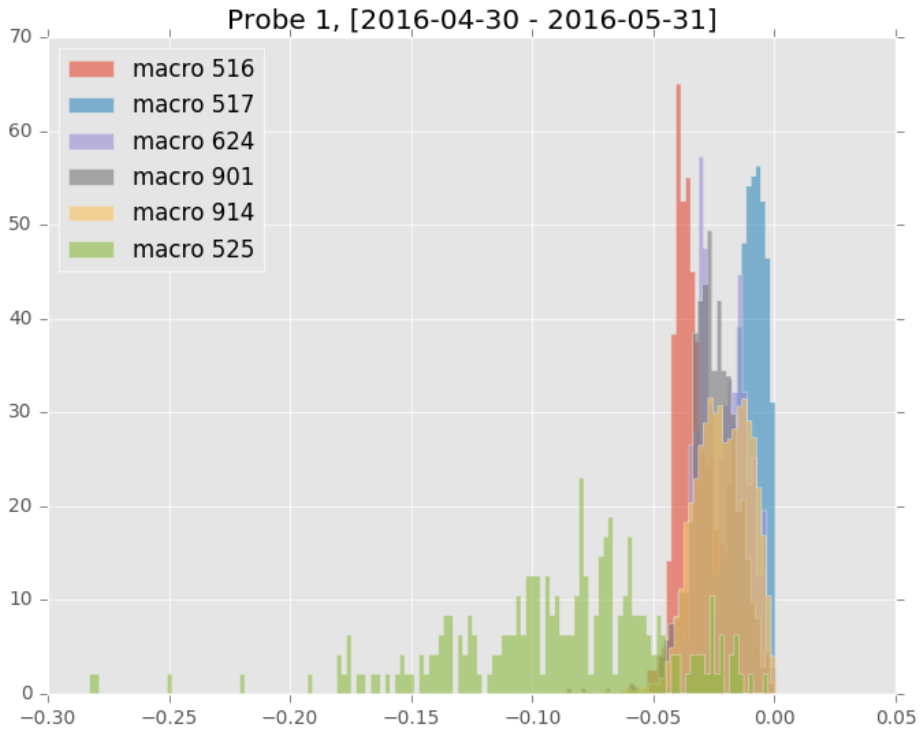


Figure 11.3: A normalized slope distribution plot for probe 1, the month of May 2016. This distribution is sorted by macro, one of the macros that has been showing an erratic timeseries response is the macro 525.

ter the period 2-7th of April. All of the timeseries data during this month is from the macro 624, listed in table 6.1. Comparing figure 11.5, with the normalized slope distribution for probe 1. The distribution in figure 11.6, suggest that there was two different distinct normalized slope distributions present in the month of April, suggesting either two different plasma distributions or interference from other instruments or noise. There are no reports of transient at Rosetta during this period.

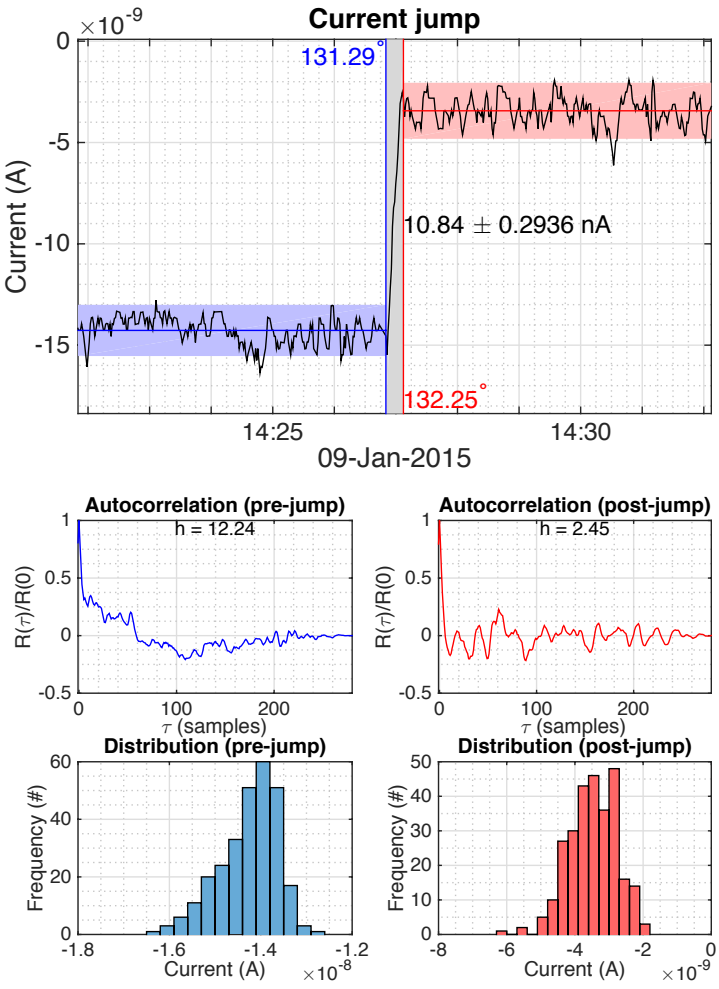


Figure 11.4: An example of Sun-Shadow Transition, the current level shift is calculated to $10.84nA$ with higher precision than any of the other methods discussed. The blue and red zone markers are the current in the periods just before and right after the transition. The blue region period right before the transition start is marked with 131.29 degrees, which is the solar aspect angle, corresponding to the probe geometry in figure 6.3, the red region is marked at 132.25, which is right after transition. Auto-correlation and the distribution is similar before and after the transition. These transitions are the best tool available for cross-calibrating the I_{Ph} multiple sweeps method. (Figure courtesy of Elias Odelstad, Swedish Institute of Space Physics).

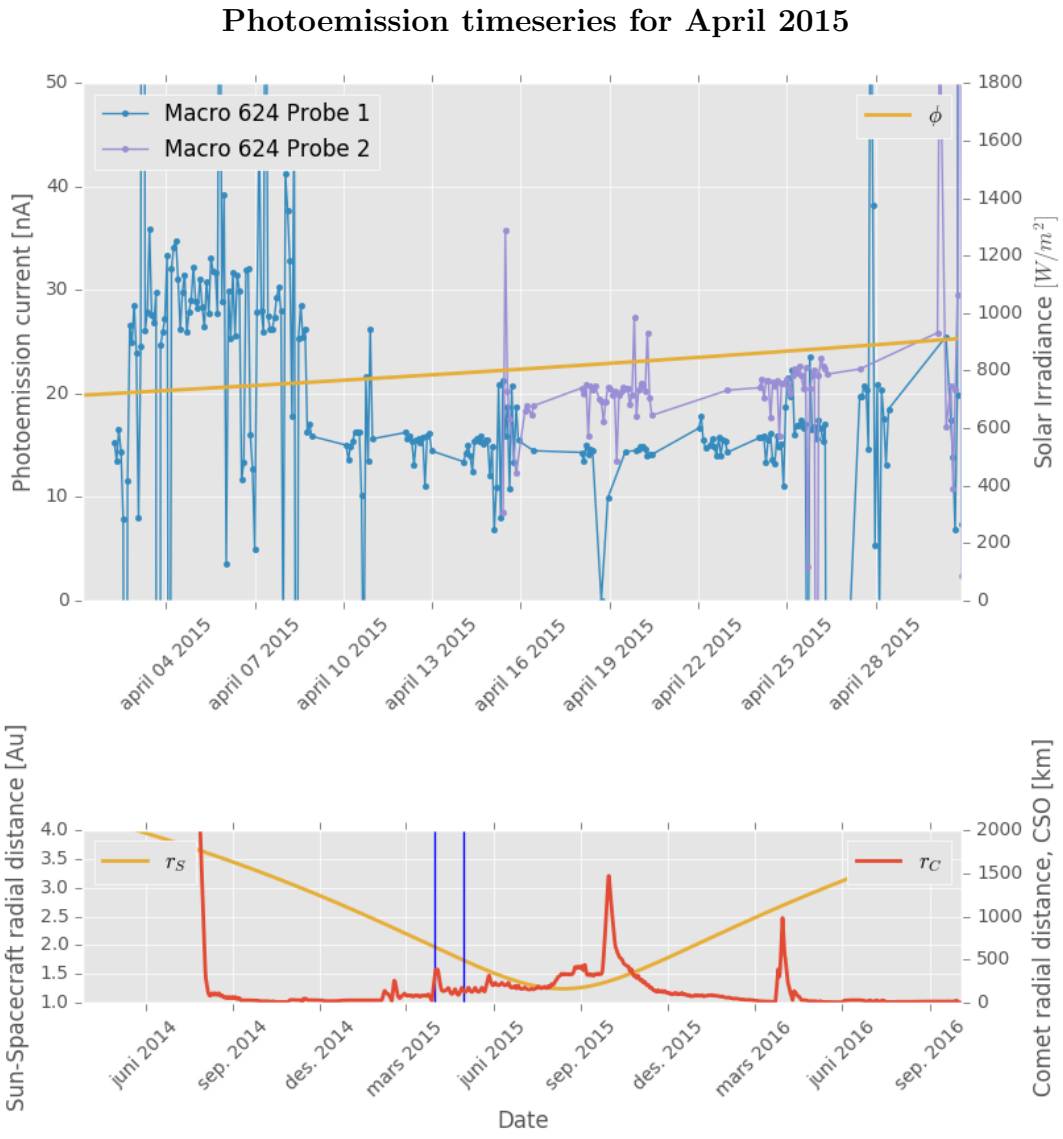


Figure 11.5: A photoemission current timeseries from Rosetta's escort of the comet 67P/C-G in April 2015. In panel one the multiple sweeps method of probe 1 and two as well as the Sun-Shadow Transition method. A simple solar irradiance model is added for reference. In the lower panel, the spacecraft radial distance with respect to the comet and the sun is traced with red and yellow lines.

Overview Distribution for probe 1 in April 2015

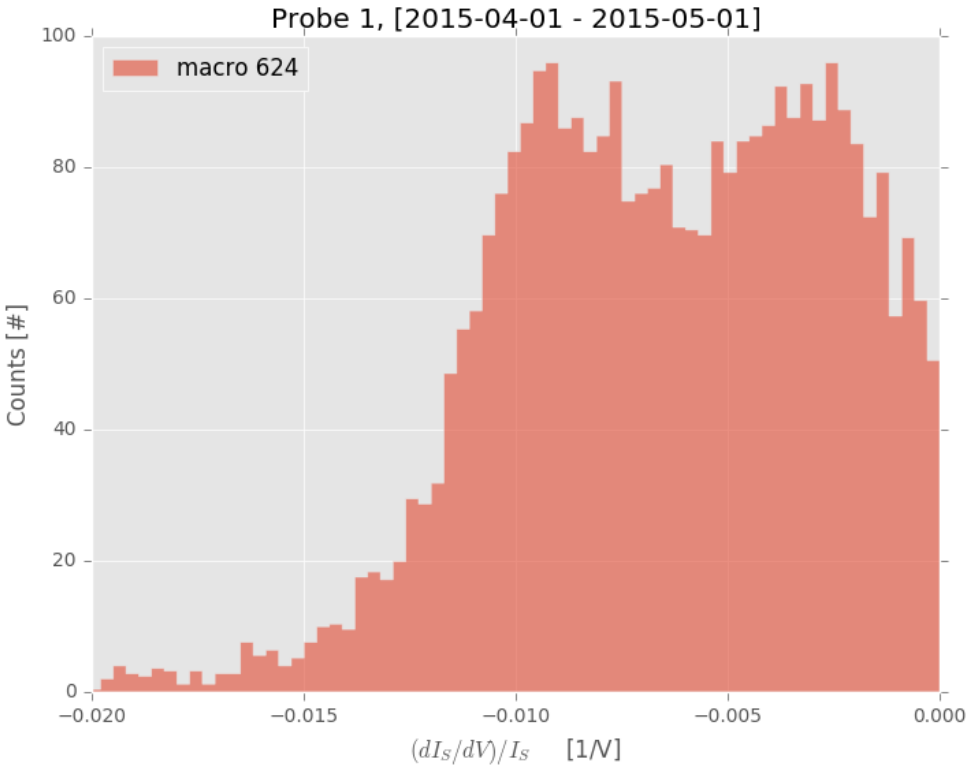


Figure 11.6: The main bulk of the normalized slope distribution for probe 1 in the month of April 2015. Along the Y-axis we have the counts for the distribution. On the X-axis we find the linear sweep slope fit over probe current. In red bars is the slope extrapolation distribution.

Chapter 12

Appendix B

A Script For the Synthetic Data Analysis

```
import csv
import numpy as np
import pandas as pd
from matplotlib import cm
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from mpl_toolkits.mplot3d import Axes3D
from scipy import interpolate
from decimal import *


class synth_data():
  def __init__(self):
    pass

  def electronResponse(self,ne,te,Vk,vb):
    """Return the electron response for given parameters"""
    self.ne = np.float_(ne)
    self.te = np.float_(te)
    self.Vk = np.float_(Vk)
    self.vb = np.float_(vb)
    temp    = np.zeros(len(vb))

    kB  = 1.381e-23      # Boltzmanns constant q   =
       1.60217646e-19
    Electron charge rp  = 2.5*1e-2          # Probe radius
    4.0*np.pi*(rp**2) # Probe area me  = 9.10938188e-31    #
       Electron mass


    Ie0     = As*ne*q*np.sqrt(kB*te/(2*np.pi*me))
       #Thermal current
```

```
    for v in vb:
      idx     = np.argmin(np.abs(vb-v))               #Find
          index in array
      if v <= Vk:                       #If bias lower than
          plasma potential
        temp[idx]   = Ie0*np.exp(q*(v-Vk)/(kB*te))  #Retarded
            current
      else:                                           #Else
          OML current
        temp[idx]   = Ie0*(1+q*(v-Vk)/(kB*te))
            #Attracted current
    return temp

  def ionResponse(self,ni,ui,mi,Vk,vb):
    """Return the ion response for given parameters"""
    self.ni = np.float_(ni)
    self.ui = np.float_(ui)
    self.mi = np.float_(mi)
    self.Vk = np.float_(Vk)
    self.vb = np.float_(vb)

    temp    = np.zeros(len(vb))                       #
        Create array
    q       = 1.60217646e-19                          #
        Electron charge
    rp      = 2.5*1e-2                                #
        Probe radius
    Ai      = np.pi*(rp**2)                  # Cross section
        area of probe
    Ii0     = ni*ui*q*Ai                              # Ram
        ion current
    Ei      = mi*(ui**2)/2                            # ion
        energy


    for v in vb:
      idx     = np.argmin(np.abs(vb-v))          # Find index
          in array
      if v-Vk < Ei/q:        # If ions have enought energy to
          reach probe
        temp[idx]   = -Ii0*(1-q*(v-Vk)/Ei)    # Cold drifting
            ion approx
    return temp


  def photoEmResponse(self,Vk, vb, Iph0):
    """Return the photoemission response for given
        parameters"""
    self.Vk     = np.float_(Vk)
    self.vb     = np.float_(vb)
    self.Iph0   = np.float_(Iph0)
```

```
    temp        = np.zeros(len(vb))

    q   = 1.60217646e-19    # Electron charge
    kB  = 1.381e-23         # Boltzmanns constant
    eV  = 11604.505         # Electronvolt to kelvin

    Tph     = 2*eV          # Temperature of photoionized
       electrons

    for v in vb:
      idx     = np.argmin(np.abs(vb-v))              #Find
         index in arr
      if v <= Vk:                     #If bias lower than
         plasma potential
        temp[idx]   = -Iph0
          #Photoemission
      else:
        temp[idx]   = -Iph0*np.exp(-q*(v-Vk)/(kB*Tph))
    return temp

  def seeRespons(self, the, tsee, nhe, sey, Vk, vb):
    #Return a secondaery electron response, (never used)
    self.the    = the
    self.tsee   = tsee
    self.nhe    = nhe
    self.sey    = sey
    self.Vk     = Vk
    self.vb     = vb

    temp        = np.zeros(len(vb))
    q           = 1.60217646e-19    # Electron charge
    kB          = 1.381e-23         # Boltzmanns constant
    rp          = 2.5*1e-2                                 #
       Probe radius
    As          = 4.0*np.pi*(rp**2) # Probe area
    me          = 9.10938188e-31    # Electron mass

    Ie0         = sey*As*nhe*q*np.sqrt(kB*the/(2*np.pi*me))



    for v in vb:
      idx     = np.argmin(np.abs(vb-v))

      if (v-Vk < 0) and (v-Vk > -(kB*the)/q):
        temp[idx]   = -Ie0*np.exp(q*(v-Vk)/(kB*the))

      elif v-Vk > 0:
        temp[idx]   = -Ie0*np.exp(-q*(v-Vk)/(kB*tsee))

    return temp
```

```
def addNoise(self, arr, noise):
  """Add Gaussian noise to an array"""
  self.arr    = arr
  self.noise  = noise

  if noise == 0:
    return arr
  else:
    temp_noise  = np.random.normal(0,noise,len(arr))
    return arr+temp_noise


def get_I(self, mi, ni, ne, ui, te, Vk, vb, Iph0, noise,
   F_grid, the, tsee, nhe, sey):
  """Return the probe current for the fine grid
  from given parameters. Essentially gather the
  wanted current response with fine grid parameters.
  """
  self.mi         = mi
  self.ni         = ni
  self.ne         = ne
  self.ui         = ui
  self.te         = te
  self.Vk         = Vk
  self.vb         = vb
  self.Iph0       = Iph0
  self.noise      = noise
  self.F_grid     = F_grid
  self.the        = the
  self.tsee       = tsee
  self.nhe        = nhe
  self.sey        = sey


  # Creating values for the fine grid
  ui      = np.linspace(ui[0], ui[1], F_grid)
  ne      = ni = np.linspace(ni[0], ni[1], F_grid)
  I = []
  # Running through the fine grid points.
  for i in range(len(ni)):
    for j in range(len(ui)):
      I_e         = self.electronResponse(ne[i], te, Vk, vb)
      I_i         = self.ionResponse(ni[i], ui[j], mi, Vk,
          vb)
      I_ph        = self.photoEmResponse(Vk, vb, Iph0)
      # I_see       = self.seeRespons(the,tsee, nhe, sey,
          Vk, vb)
      I_temp      = I_i + I_ph + I_e
      # I_temp        = self.addNoise(I_e + I_i + I_ph,
          noise)
```

```
            I.append(I_temp)

        return  I

    def get_dIdV(self, pbs, pkbo, I, Vk, vb):
        """Given a knee
        """

        self.pbs    = pbs
        self.pkbo   = pkbo
        self.I      = I
        self.Vk     = Vk
        self.vb     = vb

        slope     = []
        intercept   = []

        I_pbs     = []
        bias_pbs  = []

        neg_slope   = []
        I_pbs_neg   = []

        for i in range(len(I)):
            pbs_index = np.arange((Vk - pkbo - pbs/2)*4 + 120, (Vk -
                pkbo + pbs/2)*4 + 120, 1)
            temp_I = I[i]
            bias_pbs_temp   = vb[pbs_index[0]     : pbs_index[-1]]
            I_pbs_temp      = temp_I[pbs_index[0] : pbs_index[-1]]

            temp_slope, temp_intercept = np.polyfit(bias_pbs_temp,
                I_pbs_temp, deg=1)

            if temp_slope >= 0:
                I_pbs.append((vb[0] + pbs/2)*temp_slope +
                    temp_intercept)
                slope.append(temp_slope)
                bias_pbs.append(bias_pbs_temp)
                intercept.append(temp_intercept)

            else:
                neg_slope.append(temp_slope)
                I_pbs_neg.append((vb[0] + pbs/2)*temp_slope +
                    temp_intercept)



        parameters_dict = {'slope':slope ,'intercept':intercept,
            'I_pbs':I_pbs,
            'bias_pbs':bias_pbs, 'neg_slope':neg_slope,
                'I_pbs_neg':I_pbs_neg}
```

```
    return parameters_dict

  def error_calc(self, pbs, pkbo, I, Vk, vb, mi, U_grid,
      N_grid, noise,ij,
                 F_grid, te):
    self.pbs        = pbs
    self.pkbo      = pkbo
    self.I          = I
    self.Vk         = Vk
    self.vb         = vb
    self.mi         = mi
    self.U_grid     = U_grid
    self.N_grid     = N_grid
    self.noise      = noise
    self.ij         = ij
    self.F_grid     = F_grid
    self.te         = te

    eV              = 11604.505         # Electronvolt to
        kelvin
    I_ph_theory     = 10e-9
    Amu             = 1.6605389*1e-27
    q               = 1.60217646e-19    # Electron charge


    didv            = self.get_dIdV(pbs, pkbo, I, Vk, vb)

    slope           = didv['slope']
    intercept       = didv['intercept']
    I_pbs           = didv['I_pbs']

    neg_slope       = didv['neg_slope']
    I_pbs_neg       = didv['I_pbs_neg']

    A               = np.vstack([I_pbs,
        np.ones(len(I_pbs))]).T

    beta, alpha     = np.linalg.lstsq(A, slope)[0]

    I_ph            = -alpha/beta
    error           = np.abs(I_ph + I_ph_theory)

    ###############
    # Calc plots
    ###############

    # plt.figure()
    # plt.subplot(3,1,1)
    # plt.title("pbs=%s, te=%s, err=%.2f " % (pbs , te/eV,
        error))
```

```
    # plt.xlabel('$I_S$')
    # plt.ylabel('Region A slope $(dI_S/dV_B )$')
    # plt.plot(I_pbs, slope,  'o', label='Slopes')

    # plt.plot(I_pbs, np.array(I_pbs)*beta + alpha , 'k',
      label='Least squares fit')
    # plt.plot(I_pbs_neg, neg_slope, 'or', label='Negative
      slopes' )
    # plt.legend()
    # plt.xlim([-7e-8,10e-9])
    # plt.ylim([-0.1e-9,2.5e-9])
    # plt.grid()
    # plt.subplot(3,1,2)
    # plt.xlabel('$I_S$')
    # plt.ylabel('Region A slope $(dI_S/dV_B )$')

    # plt.plot(I_pbs, slope,  'o', label='Slopes')
    # plt.plot(I_pbs, np.array(I_pbs)*beta + alpha , 'k',
      label='Least squares fit')
    # plt.plot(I_pbs_neg, neg_slope, 'or', label='Negative
      slopes' )
    # plt.grid()

    # plt.subplot(3,1,3)
    # plt.plot(I_pbs, slope,  'o', label='Slopes')



    #
        plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/"
       +
    #"Python/Rosetta_ph_study/Synthetic_data/Plots/Ideal/Calc/"
       +
    #
        "Iv_curves_U_grid=%s_F_grid=%s_pbs=%s_mi=%s_std=%snA_#%s.png"
    #  % (U_grid*U_grid, F_grid*F_grid, pbs, mi/Amu,
        noise/1e-9, ij))
    ## plt.show()
    # plt.grid()
    # plt.close()

    return error


  def iv_fit(self, pbs, I, Vk, vb, mi, U_grid, noise,
    ij, F_grid, error_temp, ui, ni):

    self.pbs        = pbs
    self.I          = I
    self.Vk         = Vk
    self.vb         = vb
```

```
        self.mi          = mi
        self.U_grid      = U_grid
        self.noise   = noise
        self.ij          = ij
        self.F_grid      = F_grid
        self.error_temp = error_temp
        self.ui          = ui
        self.ni          = ni


        q            = 1.60217646e-19    # Electron charge
        rp           = 2.5*1e-2          # Probe radius
        As           = 4.0*np.pi*(rp**2) # Probe area
        kms          = 1e3          # kilometer
        n            = 1e6          # density cm e-3
        nA           = 1e-9         # nanoampere

        ui           = np.mean(ui)
        ni           = np.mean(ni)
        I0           = ni*ui*q*As
        ui           = ui/kms
        ni           = ni/n

        Amu          = 1.6605389*1e-27

        didv         = self.get_dIdV(pbs, I, Vk, vb)

        slope        = didv['slope']
        intercept    = didv['intercept']
        bias_pbs     = didv['bias_pbs']


        plt.figure()
        plt.title('err=%s n=%.2f u=%.2f I0=%.2e'
                  % (error_temp/nA, ni, ui, I0))
        plt.xlabel('Bias [V]')
        plt.ylim([-0.5e-7,0.5e-7])
        plt.ylabel('probe current [A]')
        plt.grid()
        # plt.legend(loc=2)

        for i in range(len(bias_pbs)):
          plt.plot(vb, I[i], label='I sweep')
          plt.axvline(pbs, label='potential knee = %s' % (pbs))
          plt.plot(bias_pbs[i], slope[i]*bias_pbs[i] +
              intercept[i],
           lw=4.0,  label='Knee offset region fit')


        #
            plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/Python/"
```

```
        +
    # "Rosetta_ph_study/Synthetic_data/Plots/Ideal/IV/" +
    #
        "Iv_fit_C_grid=%s_F_grid=%s_pbs=%s_mi=%s_std=%snA_#%s.png"
    # % (U_grid, F_grid, pbs, mi/Amu, noise/1e-9, ij))

    plt.show()
    plt.close()


  def error_plt2(self, error_matrix, ui, ni, U_grid, N_grid,
      mi,
                 noise, F_grid, pbs, te):

    self.error_matrix   = error_matrix
    self.ui             = ui
    self.ni             = ni
    self.U_grid         = U_grid
    self.N_grid         = N_grid
    self.mi             = mi
    self.noise          = noise
    self.F_grid         = F_grid
    self.pbs            = pbs
    self.te             = te

    from mpl_toolkits.axes_grid1 import ImageGrid

    params = {'text.latex.preamble' :
        [r'\usepackage{siunitx}',
     r'\usepackage{amsmath}']}

    plt.rcParams.update(params)

    kms                 = 1e3
    n                   = 1e6               # ne = ni
    Amu                 = 1.6605389*1e-27
    eV                  = 11604.505         # Electron volt
        to kelvin
    kinetic_eV          = 1.60217653e-19    # kinetic to eV


    ui                  = ui/kms
    ni                  = ni/n

    fig, ((ax1, ax2),(ax3, ax4)) = plt.subplots(2, 2,
                                                figsize=(2*4.15,2*4.15))


    plt.title("Error estimate for Cometary Environment")

    ui_mesh             = np.linspace(np.min(ui), np.max(ui),
```

```
        np.shape(error_matrix[0])[1])
ni_mesh             = np.linspace(np.min(ni), np.max(ni),
        np.shape(error_matrix[0])[0])

ui_mesh2            = np.linspace(np.min(ui), np.max(ui),
        np.shape(error_matrix[1])[1])
ni_mesh2            = np.linspace(np.min(ni), np.max(ni),
        np.shape(error_matrix[1])[0])

X, Y    = np.meshgrid(ui_mesh,  ni_mesh)
X2, Y2  = np.meshgrid(ui_mesh2, ni_mesh2)

vmin = ''
vmax = ''

im      = ax1.pcolor(X,Y, np.array(error_matrix[0])/1e-9,
cmap=plt.get_cmap('viridis')) #vmin=-20, vmax= 20)

ax1.set_xticks(ax1.get_xticks()[0:])
ax1.set_yticks(ax1.get_yticks()[1:])
ax1.set_xticklabels([])
ax1.text(ax1.get_xticks()[0], ax1.get_yticks()[-1], 'a)',
         verticalalignment = 'top', color='black')

ax2.pcolor(X,Y, np.array(error_matrix[1])/1e-9,
cmap=plt.get_cmap('viridis')) #vmin=-20, vmax= 20)

ax2.set_xticks(ax2.get_xticks()[0:])
ax2.set_yticks(ax2.get_yticks()[1:])
ax2.set_xticklabels([])
ax2.set_yticklabels([])

ax2.text(ax1.get_xticks()[0], ax1.get_yticks()[-1],
         'b)', verticalalignment = 'top', color='black')

ax3.pcolor(X2,Y2, np.array(error_matrix[2])/1e-9,
cmap=plt.get_cmap('viridis')) #vmin=-20, vmax= 20)

ax3.set_xticks(ax3.get_xticks()[0:])
ax3.set_yticks(ax3.get_yticks()[1:])

ax3.text(ax1.get_xticks()[0], ax1.get_yticks()[-1],
         'c)', verticalalignment = 'top', color='black')

ax4.pcolor(X2,Y2, np.array(error_matrix[3])/1e-9,
cmap=plt.get_cmap('viridis')) #vmin=-20, vmax= 20)

ax4.set_yticklabels([])
ax4.set_xticks(ax4.get_xticks()[0:])
ax4.set_yticks(ax4.get_yticks()[1:])
```

```
    ax4.text(ax1.get_xticks()[0], ax1.get_yticks()[-1], 'd)',
             verticalalignment = 'top', color='black')


    cbar_ax     = fig.add_axes([0.84, 0.15, 0.02, 0.8])

    cb = fig.colorbar(im, cax=cbar_ax)
    # cb.set_ticks([-10,-8,-6, -4, -2, 0, 2 , 4, 6, 8, 10])
    cb.set_label('Error $[nA]$', labelpad=-0.9)


    plt.subplots_adjust(bottom=0.15,top=0.95, wspace=0.2,
                        hspace=0.2,left=0.19, right=0.82)

    ax1.set_ylabel('n $[cm^{-3}]$')
    ax3.set_ylabel('n $[cm^{-3}]$')

    ax3.set_xlabel('u $[km/s]$')
    ax4.set_xlabel('u $[km/s]$')

    plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/Python/"
       +
        "Rosetta_ph_study/Synthetic_data/Plots/Ideal/" +
        "error_plot_U_N_grid=%s_%s_F_grid=%s_Te=%s_YY.png"
        % (U_grid,N_grid, F_grid, te/eV))

    plt.show()


def kplot(self, mi, ui, ni, pbs, vb, I, Vk, ij, F_grid,
    error, te):
    self.mi         = mi
    self.ui         = ui
    self.ni         = ni
    self.pbs        = pbs
    self.vb         = vb
    self.I          = I
    self.Vk         = Vk
    self.ij         = ij
    self.F_grid     = F_grid
    self.error      = error
    self.te         = te


    q           = 1.60217646e-19    # Electron charge
    kms         = 1e3
    Vp          = Vk - (vb[0] + float(pbs/2))
    Amu         = 1.6605389*1e-27
    rp          = 2.5*1e-2          # Probe radius
    As          = 4.0*np.pi*(rp**2) # Probe area
    eV          = 11604.505         # Electronvolt to kelvin
```

```
ui          = np.linspace(ui[0], ui[1], F_grid)
ne          = ni = np.linspace(ni[0], ni[1], F_grid)

didv        = self.get_dIdV(pbs, I, Vk, vb)

slope       = didv['slope']
intercept   = didv['intercept']
I_pbs       = didv['I_pbs']
neg_slope   = didv['neg_slope']
I_pbs_neg   = didv['I_pbs_neg']

A           = np.vstack([I_pbs, np.ones(len(I_pbs))]).T

beta, alpha = np.linalg.lstsq(A, slope)[0]
I_ph        = alpha/beta
I_ph_theory = -10e-9

fig         = plt.figure()
ax          = plt.subplot(2,1,1)

fig.set_figheight(30)
fig.set_figwidth(15)

ax.get_xaxis().set_major_formatter(
  plt.matplotlib.ticker.FuncFormatter(
    lambda x, p: format(x, ',')))

plt.title('didv k plot m=%s err=%.2f' % (mi/Amu, error))
plt.xlabel('$I_S$ [A]')
plt.ylabel('Slopes  $dI_S$/$dV_B$ [A/V] ')
plt.grid()


for i in range(len(ui)):
  E_i             = 0.5*mi*ui[i]**2
  k               = -q/(E_i - q*Vp)
  m               = I_ph_theory*k

  I_linspace      = np.linspace(np.min(I_pbs), 0, 10)
  Theoretic_slope = np.array(I_linspace)*k + m

  plt.plot(I_linspace, Theoretic_slope, label='Theory')

plt.plot(I_pbs, slope, 'o', label='Synthetic data')
plt.plot(I_linspace, np.array(I_linspace)*beta + alpha,
    'k',
    label='Least squares sweep fit')

#############
# Subplot 2
```

```
    ############

    ax2 = plt.subplot(2,1,2)
    plt.xlabel('I_S [A]')
    plt.ylabel('Slopes  dI_S/dV_B [A/V] ')
    # plt.legend()


    for i in range(len(ui)):
      E_i             = 0.5*mi*ui[i]**2
      k               = -q/(E_i - q*Vp)
      m               = -I_ph_theory*k

      I_linspace      = np.linspace(np.min(I_pbs), 0, 10)
      Theoretic_slope = np.array(I_linspace)*k + m

      plt.plot(I_linspace, Theoretic_slope, label='Theory')

    plt.plot(I_pbs, slope, 'o', label='Synthetic data')
    plt.plot(I_linspace, np.array(I_linspace)*beta + alpha,
             'k', label='Least squares sweep fit')

    plt.axis([-1.5e-8, -0.5e-8, -0.5e-9, 0.5e-9])
    plt.grid()

    plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/"
       +
        "Python/Rosetta_ph_study/Synthetic_data/Plots/Ideal/kplot/"
           +
        "didv_kplot_pbs=%s_mi=%s_#%s_te=%s.png"
        % (pbs, mi/Amu, ij, te/eV))

    # plt.show()
    plt.close()


  def fixed_u_plot(self, mi, ni, te, Vk, vb, pbs, Iph0,
                   the, tsee, nhe, sey):

    """ Takes ui and n values and create analysis plots
    synthetic data is compared with theory.
    """

    self.mi          = mi
    self.ni          = ni
    self.te          = te
    self.Vk          = Vk
    self.vb          = vb
    self.pbs         = pbs
    self.Iph0        = Iph0
    self.the         = the
```

```
        self.tsee       = tsee
        self.nhe        = nhe
        self.sey        = sey

        q           = 1.60217646e-19    # Electron charge
        kinetic_eV  = 1.60217653e-19    # kinetic to eV
        Amu         = 1.6605389*1e-27   # mu to kg
        kms         = 1e3
        eV          = 11604.505         # Electronvolt to kelvin
        Vp          = Vk - (vb[0] + float(pbs/2))
        ne          = ni
        n           = 1e6

        ui_temp     = np.array([0.1, 50, 350])*kms
        ni          = np.linspace(0,100, 1000)*n

        plt.figure(figsize=(8.30, 8.30))
        params = {'text.latex.preamble' : [
            r'\usepackage{newtxtext}',
        r'\usepackage{newtxmath}']}
        plt.rcParams.update(params)


        plt.axvline(-Iph0/1e-9, color='k')

        # plt.title('yet to be determined')
        cmap = plt.get_cmap('viridis')

        symbols = ['v', 's', 'H']
        markerskip = [6,5,1]

        for i in range(len(ui_temp)):
          I               = []
          # Isee          = []
          ni_temp         = ni*(ui_temp[i]/ui_temp[0])
          for j in range(len(ni_temp)):
            I_e         = self.electronResponse(ni_temp[j], te,
                Vk, vb)
            I_i         = self.ionResponse(ni_temp[j],
                ui_temp[i], mi, Vk, vb)
            I_ph        = self.photoEmResponse(Vk, vb, Iph0)
            # I_see       = self.seeRespons(the, tsee, nhe, sey,
                Vk, vb)
            I_temp      = I_i + I_ph + I_e
            # I_temp_see  = I_i + I_ph + I_e + I_see
            # i_temp        = self.addNoise(I_temp, noise)
            I.append(I_temp)
            # Isee.append(I_temp_see)

          didv        = self.get_dIdV(pbs, I, Vk, vb)
          slope       = np.array(didv['slope'])
```

```
        I_pbs       = np.array(didv['I_pbs'])

        beta, alpha = np.polyfit(I_pbs, slope, deg=1)
        I_ph_data   = -alpha/beta

        ## see

        # didv_see  = self.get_dIdV(pbs, Isee, Vk, vb)
        # slope_see     = np.array(didv_see['slope'])
        # I_pbs_see     = np.array(didv_see['I_pbs'])

        # beta_see, alpha_see = np.polyfit(I_pbs_see,
            slope_see, deg=1)
        # I_ph_data_see  = -alpha_see/beta_see

        #########
        #K theory
        #########
        E_i             = 0.5*mi*ui_temp[i]**2
        k               = -q/(q*Vp + E_i)
        m               = Iph0*k
        I_theory        = np.linspace(np.min(I_pbs), 0, 10)
        slope_theory    = I_theory*k + m

        color = cmap(float(i)/len(ui_temp))

        plt.plot(np.array(I_pbs[0::markerskip[i]])/1e-9,
            np.array(slope[0::markerskip[i]])/1e-9,
            marker=symbols[i], alpha=0.5, color=color,
            label='$u=%s \ [km/s]$' % (ui_temp[i]/kms))

        # plt.plot(I_pbs_see/1e-9, slope_see/1e-9, marker='*',
            color=color, label='Isee')

        plt.plot(np.array(I_theory)/1e-9,
            np.array(slope_theory)/1e-9,color=color,
            label='Theory')

        plt.grid()
        plt.tight_layout()
        plt.axis([-50,0,0,1])
        plt.text(-Iph0/1e-9, 0.4, "$I_{ph0}= -10 \ nA$",
            rotation='vertical')
        plt.xlabel('$\mathbf{I_{tot} [nA]}$')
        plt.ylabel('$\mathbf{dI_{tot}/dV_P [nA/V]}$')

        plt.legend(loc=1, numpoints=1, borderaxespad=0.)
        plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/Python/"
            +
            "Rosetta_ph_study/Synthetic_data/Plots/Ideal/Fixed_u_n/"
                +
```

```
            "fixed_u_pbs=%s_mi=%s_ui=%s_te=%s_the=%s.png"
            % (pbs, mi/Amu, ui_temp/kms, te/eV, the/eV))
    plt.show()
    # plt.close()

  def fixed_n_plot(self, mi, ni, te, ui, Vk, vb, pbs, Iph0):
    self.mi     = mi
    self.ni     = ni
    self.te     = te
    self.ui     = ui
    self.Vk     = Vk
    self.vb     = vb
    self.pbs    = pbs
    self.Iph0   = Iph0

    Amu         = 1.6605389*1e-27   # amu to kg
    eV          = 11604.505         # Electronvolt to kelvin
    n           = 1e6

    I           = []
    for i in range(len(ui)):
      # i_e       = self.electronResponse(ne[i], te, Vk, vb)
      i_i       = self.ionResponse(ni, ui[i], mi, Vk, vb)
      i_ph      = self.photoEmResponse(Vk, vb, Iph0)
      # i_temp    = self.addNoise(i_e + i_i + i_ph, noise)
      i_temp    = i_i + i_ph
      I.append(i_temp)

    didv        = self.get_dIdV(pbs, I, Vk, vb)
    slope       = didv['slope']
    I_pbs       = didv['I_pbs']

    beta, alpha = np.polyfit(I_pbs, slope, deg=1)
    I_ph_data   = -alpha/beta

    I_linear    = np.linspace(np.min(I_pbs), 1e-10, 10)

    linfit      = I_linear*beta + alpha

    plt.figure()
    plt.title('n=%s [n/cm3] I_ph_data=%s' % (ni/1e6,
       I_ph_data/1e-9))
    plt.plot(np.array(I_pbs)/1e-9,np.array(slope)/1e-9, 'o',
      label = 'Synthetic data')

    plt.plot(I_linear/1e-9, linfit/1e-9, label = 'Linfit')
    plt.xlabel('$I_S$ [nA]')
    plt.ylabel('$dI_S/dV_B$ [nA]')
    plt.grid()
    plt.legend()
```

```
    plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/Python/"
        "Rosetta_ph_study/Synthetic_data/Plots/Ideal/Fixed_u_n/"
            +
        "fixed_n_pbs=%s_mi=%s_ni=%s_te=%s.png"
        % (pbs, mi/Amu, ni/n, te/eV))

    # plt.show()
    plt.close()

  def iv_characteristic_plot(self, vb, Vk, the, tsee, nhe,
      sey, pbs, mi):

    self.vb     = vb
    self.the    = the
    self.tsee   = tsee
    self.nhe    = nhe
    self.sey    = sey
    self.pbs    = pbs

    plt.style.use('ggplot')

    eV          = 11604.505             # Electronvolt to
        kelvin
    Amu         = 1.6605389*1e-27       # amu to kg
    n           = 1e6                   # ne = ni = plasma
        density [ni/cm^3]
    kms         = 1e3                   # kilometer per second
    nA      = 1e-9
    Iph0        = 10e-9

    ui = np.array([5])*kms
    ni = ne = np.array([20])*n
    te = 5*eV
    noise =  0.5*nA
    pkbo = 20


    fig = plt.figure(figsize = (2*4.15, 1.5*4.15))
    ax = fig.add_subplot(111)
    I  = []
    I_ph        = self.photoEmResponse(Vk, vb, Iph0)

    for i in range(len(ni)):
      I_e         = self.electronResponse(ne[i], te, Vk, vb)

      plt.plot(vb, I_e/nA         , label = '$I_e$', lw=3  )
      for j in range(len(ui)):
        I_i         = self.ionResponse(ni[i], ui[j], mi, Vk,
            vb)
```

```
        # I_see      = self.seeRespons(the, tsee, nhe, sey,
            Vk, vb)

        I_temp       = self.addNoise(I_e + I_i + I_ph, noise)
        I.append(I_temp)

        plt.plot(vb, I_temp/nA,'.', label = '$I_{p}$' )
      plt.plot(vb, I_i/nA, label = '$I_i$', lw =3)


    plt.plot(vb, I_ph/nA,  label = '$I_{ph}$', lw=3)
    plt.axvline(Vk, ls="dashed", color='black')
    plt.text(10, 15, "$V_k = 10V$", rotation='vertical')
    ax.add_patch(patches.Rectangle((Vk-pkbo-pbs[0]/2, -25),
        pbs[0], 100,
                  alpha=0.2, color='yellow', label='PBS'))

    ax.add_patch(patches.FancyArrowPatch((Vk-pkbo, -15),
      (Vk, -15), arrowstyle='simple', mutation_scale=20,
          facecolor="black",
      label="PKBO"))
    plt.text(-5, -17, "$PKBO = %sV$" % pkbo,
        rotation='horizontal')
    plt.grid()
    plt.xlabel("Bias potential $[V]$")
    plt.ylabel("Probe current $[nA]$")
    plt.ylim([-20,20])
    plt.legend(loc=4, numpoints=1)
    # plt.show()

    didv = self.get_dIdV(pbs[0], pkbo, I, Vk, vb)

    # slope           = didv['slope']
    # intercept       = didv['intercept']
    # I_pbs           = didv['I_pbs']

    # A               = np.vstack([I_pbs,
        np.ones(len(I_pbs))]).T

    # beta, alpha     = np.linalg.lstsq(A, slope)[0]

    # I_ph            = -alpha/beta
    # I_ph_theory   = 10*nA
    # error           = I_ph + I_ph_theory
    # I2 = np.linspace(-11*nA,-9*nA,100)

    # fig2 = plt.figure(figsize = (2*4.15, 1.5*4.15))
    # plt.title('Error = %s' % str(error/nA))
    # plt.plot(np.array(I_pbs)/nA, np.array(slope)/nA, '.',
        label='slopes')
    # plt.axvline(-10, ls="dashed", color='black')
```

```
    # plt.text(-10, 0.06, "$I_{ph} = -10nA$",
        rotation='vertical')

    # plt.ylim(0,0.1)
    # plt.xlim(-11,-9.8)
    # plt.xlabel('$I [nA]$')
    # plt.ylabel('$dI/dV [nA]$')
    # plt.plot(I2/nA, (np.array(I2)*beta + alpha)/nA,
        label='Linear fit', lw=2)
    # plt.legend()
    plt.grid()

    plt.show()


  def error_noise_analysis(self, Vk, vb, mi, Iph0):
    self.Vk             = Vk
    self.vb             = vb
    self.mi             = mi
    self.Iph0           = Iph0

    plt.style.use('ggplot')

    nA                  = 1e-9
    kms                 = 1e3
    ij                  = None
    the                 = None
    tsee                = None
    nhe                 = None
    sey                 = None
    I_ph_theory         = 10*nA
    eV                  = 11604.505
    n                   = 1e6


    # ni                = np.linspace(0,4000,35)*n
    # ni                = np.array([1000,2000,3000,4000])*n
    # ui                = np.linspace(1, 5, 35)*kms

    pkbo       = 20
    noise        = np.array([0.5])*nA

    # solar wind cond
    ui       = 350*kms
    te         = 10*eV
    dui      = 50*kms

    ni       = 2*n
    dni      = 3 *n

    # # cometary environment
```

```
    # ui     = 1*kms
    # dui      = 0.1*kms
    # te     = 1*eV

    # ni     = 500*n
    # dni      = 2000*n



    pbs      = range(5,30)
    sweeps     = [3, 5]
    mean_range  = 30
    f, axarr = plt.subplots(2, sharex=True)
    axarr[0].set_title('Slow solar wind')
    # axarr[0].set_title('Cometary environment')

    for l in range(len(sweeps)):
      error_lst      = []
      std_lst      = []
      ui_lst       = np.linspace(ui, ui + dui, sweeps[l])
      ni_lst       = np.linspace(ni, ni + dni, sweeps[l])
      for k in range(len(pbs)):
        error = []
        for m in range(mean_range):
          I        = []

          for i in range(len(ui_lst)):

            for j in range(len(ni_lst)):
              I_e          = self.electronResponse(
                    ni_lst[j], te, Vk, vb)

              I_i          = self.ionResponse(
                    ni_lst[j], ui_lst[i], mi, Vk, vb)

              I_ph         = self.photoEmResponse(
                   Vk, vb, Iph0)

              I_temp       = I_i + I_ph + I_e

              I_temp         = self.addNoise(
                  I_e + I_i + I_ph, noise)
              I.append(I_temp)

          print pbs[k], "pbs"
          print pkbo, "pkbo"
          didv         = self.get_dIdV(pbs[k], pkbo, I, Vk, vb)
          slope        = np.array(didv['slope'])
          I_pbs        = np.array(didv['I_pbs'])
          beta, alpha = np.polyfit(I_pbs, slope, deg=1)
          I_ph_data    = -alpha/beta
```

```
            error_temp  = np.abs(I_ph_data + I_ph_theory)
            error.append(error_temp)

          std       = np.std(error , axis=0)
          error       = np.mean(error, axis=0)
          std_lst.append(std)
          error_lst.append(error)

        axarr[l].plot(pbs, np.array(error_lst)/nA,color='blue',
              label="%s sweeps" % str(sweeps[l]*sweeps[l]))

        axarr[l].fill_between(pbs,
            np.array(error_lst)/nA-np.array(std_lst)/nA,
             np.array(error_lst)/nA+np.array(std_lst)/nA,
             color='blue', alpha=0.2, antialiased=True)

        axarr[l].set_ylim(np.array([-0.05,1]))
        axarr[l].grid()
        axarr[1].set_xlabel('PBS $[V]$')
        axarr[l].set_ylabel('Absolute Error $[nA]$')
        axarr[l].legend(loc=2)
        axarr[l].grid()

      #
          plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/"+
      # Python/Rosetta_ph_study/Synthetic_data/Plots/Ideal/" +
      # "PBS_SWP_analyis_m=%s_te=%s_PKBO=%s_ui=%s_ni=%s_dui= +
      # "%s_dni=%s_avgmean=%s.png" %
      # (m,te/eV,pkbo, ui/kms, ni/n, dui/kms, ni/n, mean_range))

      plt.show()


    def electron_influence(self):
      # plot electron influence

      n = 1e6
      eV= 11604.505
      nA = 1e-9
      plt.style.use('ggplot')

      ne = 2*n
      te = np.array([1,4,10,15])*eV
      Vk = 0
      vb = np.linspace(0,-30,120)
      fig, (ax1, ax2) = plt.subplots(1, 2,
          figsize=(2*4.15,1.1*4.15))
      ax1.set_xticklabels(np.linspace(30,0,7))


      for i in range(len(te)):
```

```
        I_e = self.electronResponse(ne,te[i],Vk,vb)
        ax1.plot(vb, I_e/nA, lw=2, label='$T_e =$%s' %
            str(te[i]/eV))
        ax1.set_ylim([0,2])

      ax1.set_ylabel('Electron current to the probe [nA]')
      ax1.set_xlabel('Potential knee bias offset [V]')
      ax1.legend(loc=2)
      ax1.text(-28, 0.5, "$n_e =%scm^{-3}$" % str(ne/n))
      ne2 = 2000*n
      te2 = np.array([0.5, 1, 2, 3])*eV

      for j  in range(len(te2)):
        I_e = self.electronResponse(ne2,te2[j],Vk,vb)
        ax2.plot(vb, I_e/nA,lw=2, label='$T_e =$%s' %
            str(te2[j]/eV))
        plt.ylim([0,10])
      ax2.text(-28, 2.5, "$n_e =%scm^{-3}$" % str(ne2/n))
      ax2.set_xlabel('Potential knee bias offset [V]')
      ax2.set_xticklabels(np.linspace(30,0,7))

      ax2.legend(loc=2)
      plt.tight_layout()

      #
          plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/"
      # Python/Rosetta_ph_study/Synthetic_data/Plots/Ideal/" +
      # "ectron_influence.png")
      plt.show()

    def run(self):
      """ Initialize synthetic data analysis routine. """

      #  --- Physical constants
      q           = 1.60217646e-19        # Electron charge
      me          = 9.10938188e-31        # Electron mass
      kB          = 1.381e-23             # Boltzmanns constant
      rp          = 2.5*1e-2              # Probe radius
      As          = 4.0*np.pi*(rp**2)     # Probe area
      eV          = 11604.505             # Electronvolt to
          kelvin
      Amu         = 1.6605389*1e-27       # amu to kg
      n           = 1e6                   # ne = ni = plasma
          density [ni/cm^3]
      kms         = 1e3                   # kilometer per second
      nA          = 1e-9

      #   --- Definition of variables
      #   ne      - electron density
      # must be in m^-3
```

```
#   te      - electron temperature
# must be in kelvin

#   Vk      - plasma potential
# alternative - local potential at the probe



#   vb      - probe bias potential
# relative to zero

#   Iph0    - photoemission
# emission when vb < Vk, needs to be in Ampere

#   Tph     - temp. of emitted elec.
# temperature of photoemitted electrons from the probe

#   ni      - ion density
# assumed to be equal to bulk electron density

#   ui      - ion flow speed
# must in m/s

#   mi      - effective ion mass
# must in kg


the         = 50*eV    # 10-350 eV?
tsee        = 2*eV
nhe         = 0.1*n
sey         = 2


Vk          = 10
# Knee bias potential

vb          = np.linspace(-30,30,240)
# The bias potential range.

pbs         = [20, 10]
# Bias potential span: pbs = 10 -> [-30V,-20V] range

Iph0        = 10e-9
# Photoemission current

noise       = np.array([0.5])*nA
# Noise level in nano Amperes.

mi          = [30*Amu]
# Ion mass [kg]
```

```
te          = np.array([5])*eV
# Electron temperature [kelvin]

pkbo    = 20



N_grid      = 100
U_grid      = [25,5]
# Coarse grid size, with odd numbers only, 5x5 and up.

F_grid      = 3
# Fine grid size.



ne = ni     = np.linspace(1, 10*n, N_grid)
# Plasma density analysis

self.fixed_u_plot(mi[0], ni, te[0], Vk, vb, pbs[0],
Iph0, the, tsee, nhe, sey)
# Running through functions for fixed ui, n analysis.


for u in range(len(ni)):
  self.fixed_n_plot(mi[0], ni[u], te[0], ui, Vk, vb,
     pbs[0], Iph0)

self.iv_characteristic_plot(vb, Vk, the, tsee, nhe, sey,
   pbs, mi[0])

self.electron_influence()
noise_level_error = self.error_noise_analysis(Vk, vb,
   mi[0], Iph0)

error_matrix_lst  = []
# values should be row by row, top to bottom for imshow

for l in range(len(U_grid)):
  ui          = np.linspace(250*kms, 500*kms, U_grid[l])
  # Ion velocity analysis span
  for p in range(len(te)):
  # running through parameter values for analysis
    for m in range(len(mi)):
      for k in range(len(pbs)):
        error_matrix = []
        for o in range(len(noise)):
          for i in range(len(ni)-1):
            error_row   = []
            error_matrix.append(error_row)  # x axis
            ni_temp = ne_temp = [ni[i], ni[i+1]]
```

```
                for j in range(len(ui)-1):
                  ui_temp     = [ui[j], ui[j+1]]
                  ij          = "%s%s" % (i,j)
                  I_temp  = self.get_I(mi[m],
                    ni_temp, ne_temp, ui_temp,
                    te[p], Vk, vb, Iph0,
                    noise[o], F_grid, the, tsee,
                    nhe, sey)

                  error_temp  = self.error_calc(
                  pbs[k], pkbo, I_temp, Vk, vb,
                  mi[m], U_grid[l], N_grid,
                  noise[o], ij, F_grid, te[p])

                  # self.iv_fit(pbs[k], I_temp, Vk,
                  #vb, mi[m], U_grid, noise[o], ij,
                  #F_grid,
                  error_row.append(error_temp)

                  # self.kplot(mi[m], ui_temp,
                  #ni_temp, pbs[k], vb, I_temp,
                  #Vk, ij, F_grid, error_temp, te[p])
                error_matrix_lst.append(error_matrix)

    self.error_plt2(error_matrix_lst, ui, ni, U_grid[l],
      N_grid, mi[m], noise[o], F_grid, pbs, te[p])


if __name__ == '__main__':
  test = synth_data()
  test.run()
```

A Script For Analyzing the RPC-LAP Dataset

```
import os
import glob
import numpy         as np
import numpy.polynomial.polynomial as poly

import pandas          as pd
import matplotlib.dates    as mdates
import matplotlib.pyplot   as plt
import matplotlib.patches    as patches
import matplotlib.dates    as mdates
```

```
from datetime import datetime
from scipy import signal
from scipy import interpolate
import scipy.stats as stats
from collections import Counter

from time import strftime as strf
import scipy.io



class Photo():
  def __init__(self):
    pass

  def get_data(self, directory, macro, start_datetime,
     stop_datetime, probe_id):
    self.directory    = directory
    self.macro        = macro
    self.start_datetime = start_datetime
    self.stop_datetime  = stop_datetime
    self.probe_id     = probe_id


    path         =r'%s' % directory

    A1S_files        = glob.glob(os.path.join(path,
       "*_%s_A%sS.csv") % (macro, probe_id))
    B1S_files        = glob.glob(os.path.join(path,
       "*_%s_B%sS.csv") % (macro, probe_id))
    I1S_files        = glob.glob(os.path.join(path,
       "*_%s_I%sS.csv") % (macro, probe_id))

    # A = pd.read_csv(A1S_files[0])
    # print A.keys()



    A_cols         = [u'START_TIME(UTC)', u' STOP_TIME(UTC)',
       u' Illumination', u' Vph_knee' ]
    df_A_temp        = (pd.read_csv(f, usecols=A_cols) for f
       in A1S_files)

    # Concatenate object along a particular axis

    df_A             = pd.concat(df_A_temp, ignore_index=True)

    df_A[u'START_TIME(UTC)']  =
       pd.to_datetime(df_A[u'START_TIME(UTC)'],
       format='%Y-%m-%dT%H:%M:%S.%f')
```

```
    df_A[u' STOP_TIME(UTC)']  = pd.to_datetime(df_A[u'
        STOP_TIME(UTC)'], format='%Y-%m-%dT%H:%M:%S.%f')
    df_A              = df_A.sort_values(by=u'START_TIME(UTC)')
    df_A              = df_A.set_index([u'START_TIME(UTC)'])

    # B Frame

    B           = pd.read_csv(B1S_files[0],
        names=['BIAS_TIME', 'BIAS'])
    swp_cols       =['SWP_START', 'SWP_STOP', 'LOCAL_START',
        'LOCAL_STOP', 'QF']

    swp_cols.extend(map(str,np.arange(0,len(B))))

    df_I_temp         = (pd.read_csv(f, header=None) for f in
        I1S_files)
    df_I          = pd.concat(df_I_temp, ignore_index=True)
    df_I.columns      = swp_cols

    df_I['SWP_START']    = pd.to_datetime(df_I['SWP_START'],
        format='%Y-%m-%dT%H:%M:%S.%f')
    df_I['SWP_STOP']  = pd.to_datetime(df_I['SWP_STOP'],
      format='%Y-%m-%dT%H:%M:%S.%f')
    df_I          = df_I.sort_values(by='SWP_START')
    df_I          = df_I.set_index(['SWP_START'])

    A             = df_A.loc[start_datetime:stop_datetime]
    I             = df_I.loc[start_datetime:stop_datetime]

    A.to_pickle("/home/sigvesh/Master/Thesis-Git-Repository/"+
                "Python/pickles/A.pkl")
    B.to_pickle("/home/sigvesh/Master/Thesis-Git-Repository/"
        +
                "Python/pickles/B.pkl")
    I.to_pickle("/home/sigvesh/Master/Thesis-Git-Repository/"
        +
                "Python/pickles/I.pkl")

    return A, B, I



  def get_Swpfit(self, pbs, pkbo, V_B, Start_times,
      Stop_times, SAA, Vk, I, ill, step, macro):

    slope      = []
    intercept    = []

    I_pbs      = []
    bias_pbs   = []
```

```
neg_slope   = []
I_pbs_neg   = []
for i in range(len(Start_times)):

  I_Swp = I.loc[Start_times[i], '0':]
  if SAA[i] == ill:
    if "NaN" in str(Vk[Start_times[i]]):
      print "passed"
    else:
      V_pbs_center  = -float(Vk[Start_times[i]]) - pkbo
      V_pbs_min     = (V_pbs_center - pbs/2)
      V_pbs_max     = (V_pbs_center + pbs/2)


      if macro in ["212", "506", "212", "505", "600",
         "604", "807", "817"]:
        pbs_index     =
           np.arange((np.abs(V_B-V_pbs_max)).argmin(),
           (np.abs(V_B- V_pbs_min)).argmin(), 1)
      else:
        pbs_index     =
           np.arange((np.abs(V_B-V_pbs_min)).argmin(),
           (np.abs(V_B-V_pbs_max )).argmin(), 1)

      # print len(pbs_index)
      # print V_pbs_center,
        (np.abs(V_B-V_pbs_min)).argmin(),
        (np.abs(V_B-V_pbs_max )).argmin()

      # plt.plot(V_B, I_Swp)
      if len(pbs_index) >= 10:
        bias_pbs_temp   = np.array(V_B[pbs_index[0]    :
           pbs_index[-1]]).astype(np.float)
        I_pbs_temp      = np.array(I_Swp[pbs_index[0] :
           pbs_index[-1]]).astype(np.float)
        # print len(bias_pbs_temp), len(I_pbs_temp)

        # plt.plot(V_B, I_Swp/1e-9, ".",  alpha=0.8,
           color = "#a9c6af", label = "$I_S$" if i == 0
           else "")
        # plt.plot(bias_pbs_temp, I_pbs_temp/1e-9,
           alpha=1, color = "#be523f", label = "$PBS$" if
           i == 0 else "", lw = 2)
        # plt.xlabel("Bias [V]")
        # plt.ylabel("probe current [nA]")

        temp_slope, temp_intercept  =
           np.polyfit(bias_pbs_temp, I_pbs_temp, deg=1)
        # plt.plot(bias_pbs_temp,
           bias_pbs_temp*temp_slope + temp_intercept,
```

```
                alpha=1, color = "blue", lw=1)

            if temp_slope > 0:
              I_pbs.append((V_pbs_center)*temp_slope +
                  temp_intercept)
              bias_pbs.append(bias_pbs_temp)

              intercept.append(temp_intercept)
              slope.append(temp_slope)

            elif np.isnan(temp_slope) == True:
              pass
            else:
              neg_slope.append(temp_slope)
              I_pbs_neg.append((V_pbs_center +
                  pbs/2)*temp_slope + temp_intercept)
      else:
        pass
    # plt.title("Macro %s" % macro)
    # plt.legend(loc= 2)
    # plt.show()


    param_dict = {'slope':slope ,'intercept':intercept,
        'I_pbs':I_pbs, 'bias_pbs':bias_pbs,
        'neg_slope':neg_slope,
              'I_pbs_neg':I_pbs_neg, 'Vk':Vk}
    return param_dict


  def macro_analysis(self, macro,step, pbs, pkbo, swp_res, A,
      B, I, min_pass, ill):

    I_ph          = []
    t0            = []
    V_SC          = []

    i = 0
    while (i+1)*swp_res < len(A):

      V_B          = B['BIAS']
      Start_times      =
          A.index.values[i*swp_res:(i+1)*swp_res]
      Stop_times       = A.loc[Start_times, u' STOP_TIME(UTC)']
      SAA          = A.loc[Start_times, u' Illumination']
      Vk           = A.loc[Start_times,  u' Vph_knee']

      Swpfit     = self.get_Swpfit(pbs[0], pkbo[0], V_B,
          Start_times, Stop_times, SAA, Vk, I, ill, step,
          macro)
```

```
        x       = np.array(Swpfit['I_pbs'])
        y       = np.array(Swpfit['slope'])

        if len(x) < min_pass:
          pass
        else:

          beta, alpha = np.polyfit(x, y, deg=1)

          # plt.plot(x,y, '.')
          # plt.plot(x, x*beta + alpha)
          # plt.show()

          I_ph_temp   = -alpha/beta
          I_ph.append(I_ph_temp)

          t0.append(Start_times[0])
          V_SC.append(Vk)


        i += 1
      return I_ph, t0, V_SC



    def V_SC_plot(self, directory, macro, start_datetime,
          stop_datetime, macro_only_p1, macro_only_p2,
          pbs, pkbo, swp_res, probe_id, min_pass):

      fig       = plt.figure(figsize = (2*4.15, 4.15))
      ax        = fig.add_subplot(111)

      Vk        = []
      Vk_ts       = []
      for j in range(len(probe_id)):
        for i in range(len(macro)):
          print macro[i], probe_id[j]
          if probe_id[j] == "2" and macro[i] in macro_only_p1:
            pass
          elif probe_id[j] == "1" and macro[i] in macro_only_p2:
            pass
          else:
            A, B, I       = self.get_data(directory, macro[i],
                start_datetime, stop_datetime, probe_id[j])
            Start_times     = A.index.values[:]
            Vk_temp       = A.loc[Start_times, u'
                Vph_knee'].dropna()

            plt.plot(Vk_temp, '.', color="#0072B2",
                label="$V_{s/c} $" if i == 0 and j ==0 else "")
```

```
            Vk.extend(Vk_temp.values)
            Vk_ts.extend(Vk_temp.index)

    Vk_df       = {"Ts":Vk_ts , "Vk":Vk}
    Vk_df       = pd.DataFrame(temp_df)

    # Vk_medfilt      = signal.medfilt(Vk, 21) # window
        length 21
    # plt.plot(Vk_ts, Vk_medfilt, label = "$V_{s/c} median
        filtered$"  if len(Vk_medfilt) != 0 else "")

    plt.ylim([-35,35])
    plt.axvline(np.datetime64('2015-08-15T02:50:00.000000000'),
        ls="dashed", color='black')
    plt.xticks(rotation=45)
    plt.xlabel("Date")
    plt.ylabel("$V_{SC}$")
    plt.tight_layout()
    plt.legend(numpoints=1)
    plt.show()
    plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/"
        +
           "Python/Vsc/Rosetta_Vsc_%s_R%s.png"
           % (pklnamestring, swp_res))



def run_pkl(self, directory, macro, step, macro_only_p1,
            macro_only_p2, all_p1_macros, all_p2_macros,
            start_datetime, stop_datetime, probe_id, pbs,
            pkbo, swp_res, min_pass, ill, pklnamestring):
  nA          = 1e-9    # [A]
  Ps          = 3.9e26  # [W] Sun power output
  AU          = 1.5e11  # [m]

  matlab_data      = scipy.io.loadmat("/home/sigvesh/Data/" +
             "csv_datafiles/sun_shadow_Iph0.mat")

  SST_Iph        = pd.DataFrame(matlab_data["Iph0"])
  SST_Iph_std      = pd.DataFrame(matlab_data["Iph0_std"])
  SST_UTC        = pd.DataFrame(matlab_data["UTC"])
  pkl_df         = pd.concat([SST_UTC, SST_Iph, SST_Iph_std],
            axis=1, join="outer")
  pkl_df.columns     = ["UTC", "Iph0", "Iph0_std"]

  SST_UTC_datetime  = []
  for i in range(len(pkl_df)):
    temp = pd.to_datetime(pkl_df["UTC"][i][0],
        format='%Y-%m-%dT%H:%M:%S.%f')
    SST_UTC_datetime.append(temp)
```

```
    pkl_df["UTC_Timestamp"] = SST_UTC_datetime
    pkl_df          = pkl_df.set_index(pkl_df["UTC_Timestamp"])

    rs              = pd.read_csv("/home/sigvesh/Data/" +
          "csv_datafiles/SUN_CG_R.csv",
          delim_whitespace=True, header=None)

    rco             = pd.read_csv("/home/sigvesh/Data/" +
          "csv_datafiles/ROS_R_CSO.csv",
          delim_whitespace=True, header=None)

    rs.columns      = ["Time", "rs"]
    rco.columns     = ["Time", "rco"]
    rs              = rs.iloc[::60*12]              # Reduce
       dataframe to reasonable size
    rco             = rco.iloc[::60*12]

    rs['Time']      = pd.to_datetime(rs['Time'],
                    format='%Y-%m-%dT%H:%M:%S.%f')
    Pc              = Ps/(4*np.pi*(np.array(rs["rs"])*AU)**2)
            # Irradiance at Rosetta
    rs              = rs.set_index(rs["Time"])
    rs["Pc"]        = Pc
    pkl_df          = pd.concat([pkl_df, rs["Pc"], rs["rs"]],
              axis=1, join="outer")


    rco['Time']     = pd.to_datetime(rco['Time'],
                    format='%Y-%m-%dT%H:%M:%S.%f')
    rco             = rco.set_index(rco["Time"])
    pkl_df          = pd.concat([pkl_df, rco["rco"]],
              axis=1, join="outer")

    for j in range(len(probe_id)):
      probe_t0_temp    = []
      probe_lph_temp     = []

      for i in range(len(macro)):
        print macro[i], probe_id[j]
        if probe_id[j] == "2" and macro[i] in macro_only_p1:
          pass
        elif probe_id[j] == "1" and macro[i] in macro_only_p2:
          pass
        else:
          A, B, I      = self.get_data(directory, macro[i],
              start_datetime, stop_datetime, probe_id[j])

          # A    = pd.read_pickle("/home/sigvesh/Master/" +
          #"Thesis-Git-Repository/Python/pickles/A.pkl")
          # B    = pd.read_pickle("/home/sigvesh/Master/" +
          #"Thesis-Git-Repository/Python/pickles/B.pkl")
```

```
            # I    = pd.read_pickle("/home/sigvesh/Master/" +
            #"Thesis-Git-Repository/Python/pickles/I.pkl")
            if macro[i] in ["506", "525"]:
              pkbo = [4]
              pbs  = [6]
              I_ph, t0, V_SC       =
                  self.macro_analysis(macro[i], step[i], pbs,
                  pkbo, swp_res, A, B, I, min_pass, ill)
            else:

              I_ph, t0, V_SC       =
                  self.macro_analysis(macro[i], step[i], pbs,
                  pkbo, swp_res, A, B, I, min_pass, ill)

            I_ph           = np.array(I_ph)/(-1*nA)

            if len(I_ph) != 0:
              probe_t0_temp.extend(t0)
              probe_Iph_temp.extend(I_ph)

            temp_df          = {"time_temp":t0  , "Iph_M%sP%s" %
                (macro[i], probe_id[j]):I_ph}
            temp_df          = pd.DataFrame(temp_df)

            temp_df['time_temp']  =
                pd.to_datetime(temp_df['time_temp'],
                format='%Y-%m-%dT%H:%M:%S.%f')
            temp_df          = temp_df.set_index(['time_temp'])
            pkl_df           = pd.concat([pkl_df,temp_df],
                axis=1, join='outer')

            # pkl_df["all_macros_p%s" % probe_id[j]]  =
                pd.(pkl_df["all_macros_p%s" % probe_id[j],
                temp_df], join="outer")

        # temp_df_probe   = {a:b for a,b in
            temp_df_probe.items() if b}

        if len(probe_Iph_temp) != 0:


          # temp_df_probe              = dict(zip(probe_t0_temp,
              probe_Iph_temp))
          temp_df_probe       =
              {"time_temp_probe":probe_t0_temp, "Probe-%s" %
              probe_id[j]:probe_Iph_temp}
          temp_df_probe      = pd.DataFrame(temp_df_probe)
```

```
        temp_df_probe["time_temp_probe"]  =
            pd.to_datetime(temp_df_probe["time_temp_probe"],
            format='%Y-%m-%dT%H:%M:%S.%f')
        temp_df_probe             =
            temp_df_probe.set_index(["time_temp_probe"])

        pkl_df                =
            pd.concat([pkl_df,temp_df_probe], axis=1,
            join='outer')

    pkl_df.to_pickle("/home/sigvesh/Master/Thesis-Git-Repository/"+
    "Python/pickles/%s_date-%s_min=%s_res=%s.pkl"
    % (pklnamestring, str(start_datetime)[0:10], min_pass,
        swp_res))



def overview_plot(self, start_datetime, stop_datetime,
    min_pass, swp_res, probe_id, macro, macro_only_p1,
    macro_only_p2, pklnamestring):

    pkl_df = pd.read_pickle("/home/sigvesh/Master/" +
    "Thesis-Git-Repository/Python/pickles/"+
    "%s_date-%s_min=%s_res=%s.pkl"

    % (pklnamestring, str(start_datetime)[0:10], min_pass,
        swp_res))

    f, (ax1, ax2) = plt.subplots(2,1, gridspec_kw =
        {'height_ratios':[3, 1]}, figsize=[2*4.15,2*4.15])

    dt64_dropna       = pkl_df.index.values #Get datetime64
        values
    ts            = (dt64_dropna -
        np.datetime64('1970-01-01T00:00:00Z')) /
        np.timedelta64(1, 's') # datetime64 -> timestamp

    ########## subplot 1
    for j in range(len(probe_id)):
      for i in range(len(macro)):
        print macro[i], probe_id[j]
        if probe_id[j] == "2" and macro[i] in macro_only_p1:
          pass
        elif probe_id[j] == "1" and macro[i] in macro_only_p2:
          pass
        else:
          # print rs["Iph_M%sP%s" % (macro[i],
              probe_id[j])].dropna()

          # temp_df = rs["Iph_M%sP%s" % (macro[i],
              probe_id[j])].dropna()
```

```
        temp_I          = pkl_df["Iph_M%sP%s" % (macro[i],
            probe_id[j])].dropna()
        ax1.plot(temp_I, ".", alpha=1.0,
        label = "$I_{Ph}$ M%s-P%s"
        % (macro[i], probe_id[j]) if temp_I.count() != 0
            else "")
        plt.setp(ax1.get_xticklabels(), rotation=45)

    # print pkl_df.keys()
    # ax1.plot(pkl_df[u'Probe-1'].dropna(), ".", label=
        "Probe 1")# if pkl_df[u'Probe-1'].count() != 0 else "")
    # ax1.plot(pkl_df[u'Probe-2'].dropna(), ".", label=
        "Probe 2")
    # ax1.plot(pkl_df["all_macros_p2"].dropna(), "-", label=
        "Probe 2" if pkl_df["all_macros_p2"].count() != 0 else
        "")

    ax1.legend(numpoints=1, loc=2)
    ax1.set_ylim([0,50])
    # ax1.set_xticklabels([])
    ax1.set_ylabel("Photoemission current [nA]")
    ax3 = ax1.twinx()

    ax3.plot(pkl_df["Pc"][start_datetime:stop_datetime].dropna(axis=0,
        how='any')*2,lw=2.0, color="#e5ae38", label="Solar
        Irradiance")
    ax3.set_ylabel("Solar Irradiance $[W/m^2]$")
    ax3.set_ylim([0,1800])
    ax3.grid()
    ax3.legend(numpoints=1)

    ############### Subplot 2 ####################

    ax2.plot(pkl_df["rs"][start_datetime:stop_datetime].dropna(axis=0,
        how='any'),lw=2.0, label="$r_S$", color="#e5ae38")
    ax2.set_xlabel("Date")
    # ax2.set_yticks(np.arange(min(pkl_df["rs"]),
        max(pkl_df["rs"])+1, 2.0))
    ax2.set_ylim([1,4])
    # ax2.axvline(start_datetime)
    # ax2.axvline(stop_datetime)
    ax2.set_ylabel("Sun radial distance $r_S$ [AU]")
    ax2.legend(numpoints=1, loc=2)

    plt.setp(ax2.get_xticklabels(), rotation=45)

    ax4 = ax2.twinx()

    ax4.plot(pkl_df["rco"][start_datetime:stop_datetime].dropna(axis=0,
        how='any'),lw=2.0, label="CSO")
```

```
    ax4.set_ylabel("Comet radial distance CSO [km]")
    ax4.legend(numpoints=1)
    ax4.set_ylim([0, 2000])
    ax4.grid()
    plt.tight_layout()
    #
        plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/Python/timeserie
        %
    #       (pklnamestring, stop_datetime,probe_id[j],
        swp_res))

    plt.show()

  def overview_plot_with_filter(self, start_datetime,
      stop_datetime, min_pass, swp_res, probe_id, macro,
      macro_only_p1, macro_only_p2, pklnamestring):
    pkl_df = pd.read_pickle("/home/sigvesh/Master/" +
        "Thesis-Git-Repository/Python/pickles/" +
        "%s_date-%s_min=%s_res=%s.pkl"
        % (pklnamestring, str(start_datetime)[0:10],
            min_pass, swp_res))

    f, (ax1, ax2) = plt.subplots(2,1, gridspec_kw =
        {'height_ratios':[3, 1]}, figsize=[2*4.15,2*4.15])

    dt64_dropna       = pkl_df.index.values #Get datetime64
        values
    ts            = (dt64_dropna -
        np.datetime64('1970-01-01T00:00:00Z')) /
        np.timedelta64(1, 's') # datetime64 -> timestamp

    for j in range(len(probe_id)):
      all_px = pd.DataFrame()


      for i in range(len(macro)):
        if probe_id[j] == "2" and macro[i] in macro_only_p1:
          pass
        elif probe_id[j] == "1" and macro[i] in macro_only_p2:
          pass
        else:
          # print rs["Iph_M%sP%s" % (macro[i],
              probe_id[j])].dropna()
          # temp_df = rs["Iph_M%sP%s" % (macro[i],
              probe_did[j])].dropna()

          print macro[i], probe_id[j]

          temp_I        = pkl_df["Iph_M%sP%s" % (macro[i],
              probe_id[j])].dropna()
```

```
            if probe_id[j] == 1:
              ax1.plot(temp_I, ".-", alpha=1.0, label = "Macro
                 %s Probe %s" % (macro[i], probe_id[j]) if
                  temp_I.count() != 0 else "")
              # plt.setp(ax1.get_xticklabels(), rotation=45)
              all_px  = pd.concat([all_px, temp_I])
            else: #, color=color[colorcycler]
              ax1.plot(temp_I, ".-", alpha=1.0, label = "Macro
                 %s Probe %s" % (macro[i], probe_id[j]) if
                  temp_I.count() != 0 else "")
              # if len(temp_I) != 0:
              # plt.setp(ax1.get_xticklabels(), rotation=45)
              all_px  = pd.concat([all_px, temp_I])

          all_px.sort_index(inplace=True)

          # all_px = all_px[all_px[0] > 0]
          if len(all_px[0].values) != 0:
            window = 30

            I_filt        = all_px[0]
            # I_filt      = signal.medfilt(all_px[0], window)
            # I_filt      =
                signal.savgol_filter(all_px[0].values, window,
                polyorder=2)
            I_filt_ts       = all_px[0].index

            variance      = all_px[0:].var(0)
            # print variance, "p%s" % probe_id[j]
            # print "probe%s, variance:" % probe_id[j], variance

            std         = pd.rolling_std(I_filt, window)# ,
                min_periods=None, freq=None, center=False,
                how=None)

            # plot = ax1.plot(I_filt_ts, I_filt, '.', label =
                "Probe %s" % (probe_id[j]) if len(I_filt) != 0
                else "")

            # ax1.fill_between(I_filt_ts, I_filt - std, I_filt +
                std, color=plot[0].get_color(),  alpha=0.2,
                antialiased=True, label="$\sigma_{P%s}$" %
                probe_id[j])#,  color=color)
            # ax1.plot(temp_I, ".", alpha=1.0, label = "$I_{Ph}$
                M%s-P%s" % (macro[i], probe_id[j]) if
                temp_I.count() != 0 else "")

          else:
            pass
```

```
    # bad data SST screening
    nA = 1e-9
    pkl_df_sst  = pkl_df[pkl_df.Iph0_std/nA < 1]
    # SST method

    # ax1.set_xticklabels([])
    ax1.set_ylabel("Photoemission current [nA]")

    ax1.errorbar(pkl_df_sst["UTC_Timestamp"][start_datetime:stop_datetime],
            np.array(pkl_df_sst["Iph0"][start_datetime:stop_datetime]/nA),
            np.array(pkl_df_sst["Iph0_std"][start_datetime:stop_datetime]/nA),
            marker=".", linestyle="", label="SST" if
                len(np.array(pkl_df_sst["Iph0"][start_datetime:stop_datetime]/nA
                != 0 else "", color="#514b44")
    plt.setp(ax1.get_xticklabels(), rotation=45)
    # ax1.axvspan("2014-08-24T12:20:00.000000",
        "2014-08-24T12:35:00.000000", color="#e5ae38",
        alpha=0.5, label="M6.0 Flare")
    # ax1.axvspan("2014-09-10T18:20:00.000000",
        "2014-09-10T19:50:00.000000", color="#e5ae38",
        alpha=0.5, label="M1.6 Flare")
    # ax1.axvspan("2014-10-22T14:00:00.000000",
        "2014-10-22T14:30:00.000000", color="#e5ae38",
        alpha=0.5, label="X1.7")
    # ax1.axvspan("2014-10-22T18:00:00.000000",
        "2014-10-22T18:30:00.000000", color="#855160",
        alpha=0.5, label="M1.7")

    # ax1.axvspan("2014-09-28T03:20:00.000000",
        "2014-09-28T04:05:00.000000", color="#e5ae38",
        alpha=0.5, label="M5.0 Flare")
    # ax1.axvspan("2014-10-02T19:20:00.000000",
        "2014-10-02T19:40:00.000000", color="#e5ae38",
        alpha=0.5, label="M7.0 Flare")
    # ax1.axvspan("2015-11-20T00:00:00.000000",
        "2015-11-21T00:00:00.000000", color="#e5ae38",
        alpha=0.5, label="2016/11/20")
    # ax1.axvspan("2016-04-18T00:45:00.000000",
        "2016-04-18T01:15:00.000000", color="#e5ae38",
        alpha=0.5, label="M7.0 Flare")

    ax1.set_ylim([0,50])




     ##Irradiance
    ax3 = ax1.twinx()
    ax3.plot(pkl_df["Pc"][start_datetime:stop_datetime].dropna(axis=0,
        how='any')*2,lw=2.0, color="#e5ae38", label="$\phi$")
    ax3.set_ylabel("Solar Irradiance $[W/m^2]$")
```

```
    ax3.set_ylim([0,1800])
    ax3.grid()
    ax1.legend(numpoints=1, loc=2)
    ax3.legend(numpoints=1)

    ################################################
    ############### Subplot 2 ######################
    ################################################
    ax2.plot(pkl_df["rs"].dropna(axis=0, how='any'),lw=2.0,
        label="$r_S$", color="#e5ae38")

    ax2.axvline(start_datetime)
    ax2.axvline(stop_datetime)

    ax2.legend(numpoints=1, loc=2)
    ax2.set_xlabel("Date")
    ax2.set_ylim([1,4])
    ax2.set_ylabel("Sun-Spacecraft radial distance [AU]")
    plt.setp(ax2.get_xticklabels(), rotation=45)

    ax4 = ax2.twinx()
    ax4.plot(pkl_df["rco"].dropna(axis=0, how='any'),lw=2.0,
        label="$r_C$")
    ax4.set_ylabel("Comet radial distance, CSO [km]")
    ax4.legend(numpoints=1)
    ax4.set_ylim([0, 2000])
    ax4.grid()
    plt.tight_layout()

    # print "Distance, start_datetime [aU]: ",
        pkl_df["rs"][start_datetime:stop_datetime]
    # print "flux ratio to goes",
        1/(pkl_df["rs"][start_datetime])**2

    #
        plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/"
        +
    "Python/timeseries/Timeseries_%s_%s_P%s_Res%s.png" %
    #       (pklnamestring, stop_datetime,probe_id, swp_res))
    plt.show()
    # # plt.close()


  def slope_distribution_analysis(self, directory, probe_id,
      macro, step, macro_only_p1, macro_only_p2,
      start_datetime,
                  stop_datetime, pkbo, pbs, ill, min_pass,
                      swp_res, pklnamestring):

    for j in range(len(probe_id)):
```

```
        swp_fits       = [0,0]
        slopes_over_I      = []
        neg_slopes_over_I    = []

        for i in range(len(macro)):
          print macro[i], probe_id[j], step[i]
          temp_step = step[i]
          if probe_id[j] == "2" and macro[i] in macro_only_p1:
            pass
          elif probe_id[j] == "1" and macro[i] in macro_only_p2:
            pass
          else:
            A, B, I      = self.get_data(directory, macro[i],
                start_datetime, stop_datetime, probe_id[j])

            k = 0
            while (k+1)*swp_res < len(A):
              V_B          = B['BIAS']
              Start_times      =
                 A.index.values[k*swp_res:(k+1)*swp_res]
              Stop_times       = A.loc[Start_times, u'
                 STOP_TIME(UTC)']
              SAA          = A.loc[Start_times, u' Illumination']
              Vk           = A.loc[Start_times,  u' Vph_knee']


              if macro[i] in ["506", "525"]:
                pkbo = [4]
                pbs  = [6]

                Swpfit     = self.get_Swpfit(pbs[0], pkbo[0],
                   V_B, Start_times, Stop_times, SAA, Vk, I,
                   ill, temp_step, macro[i])
              else:
                Swpfit     = self.get_Swpfit(pbs[0], pkbo[0],
                   V_B, Start_times, Stop_times, SAA, Vk, I,
                   ill, temp_step, macro[i])
              x        = np.array(Swpfit['I_pbs'])
              y        = np.array(Swpfit["slope"])
              print x,y

              if len(x) < min_pass:
                pass
              else:
                slopes_over_I.extend(y/x)

                swp_fits[j]  += 1
              k += 1

        np.save("/home/sigvesh/Master/Thesis-Git-Repository/" +
          "Python/slope_dist/%s_slopedist_date_%s_P-%s_R-%s.npy"
```

```
        % (pklnamestring, str(start_datetime)[0:10],
            probe_id[j], swp_res), slopes_over_I)
        np.save("/home/sigvesh/Master/Thesis-Git-Repository/" +
          "Python/slope_dist/%s_swpfits_date_%s_P-%s_R-%s.npy"
          % (pklnamestring, str(start_datetime)[0:10],
            probe_id[j], swp_res), swp_fits)

        slopes_over_I     = np.load("/home/sigvesh/Master/" +
              "Thesis-Git-Repository/Python/slope_dist/" +
              "%s_slopedist_date_%s_P-%s_R-%s.npy"
              % (pklnamestring, str(start_datetime)[0:10],
                  probe_id[j], swp_res))


        swp_fits      = np.load("/home/sigvesh/Master/" +
          "Thesis-Git-Repository/Python/slope_dist/" +
          "%s_swpfits_date_%s_P-%s_R-%s.npy"
          % (pklnamestring, str(start_datetime)[0:10],
            probe_id[j], swp_res))

        mean        = np.mean(slopes_over_I)
        std         = np.std(slopes_over_I)
        skew        = stats.skew(slopes_over_I)
        kurt        = stats.kurtosis(slopes_over_I)

        ax1 = plt.subplot(2,1,j+1)
        # plt.subplot(1,1,1)

        max_treshold  = 5*std
        min_treshold  = -5*std

        # plt.ylim([0,50])

        xmax = 0.01
        xmin = -0.1

        slopes_over_I.sort()
        slopes_over_I_clip = np.clip(slopes_over_I, xmin, xmax)

        slopes_over_I   = slopes_over_I [slopes_over_I  <
            max_treshold]
        slopes_over_I   = slopes_over_I [slopes_over_I  >
            min_treshold]

        mean        = np.mean(slopes_over_I)
        std         = np.std(slopes_over_I)
        skew        = stats.skew(slopes_over_I)
        kurt        = stats.kurtosis(slopes_over_I)

        outlier_filtered_mean    = np.mean(slopes_over_I)
        outlier_filtered_std  = np.std(slopes_over_I)
```

```
        outlier_filtered_skew    = stats.skew(slopes_over_I)
        outlier_filtered_kurt = stats.kurtosis(slopes_over_I)



        plt.title("Distribution Probe %s" % (probe_id[j]))
            #(len(slopes_over_I), str(start_datetime)[0:10],
            pklnamestring, swp_res, len(slopes_over_I)))
        n, bins, patches = ax1.hist(slopes_over_I_clip,
            bins=40, alpha=0.85, label="Slope Dist",
            color="#be523f")#,histtype="stepfilled")#, normed=1,
            histtype='stepfilled', alpha=0.75, label="P%s" %
            probe_id)
        ax1.set_ylabel("Count [#]")
        ax1.legend(loc = 1)

        # plt.setp(ax1.get_xticklabels(), rotation=45)


        ax2 = ax1.twinx()

        ax2.grid("off")

        normfit = stats.norm.pdf(slopes_over_I, mean, std)
        ax2.plot(slopes_over_I, normfit, label="Norm Fit",
            lw=2.0, color="#2f5c5c")

        peak = np.where(n == n.max())
        peakbin = bins[peak][0]

        peak_treshold     = 2

        slopes_over_I_peak  = slopes_over_I

        slopes_over_I_peak  =
            slopes_over_I_peak[slopes_over_I_peak < peakbin +
            peak_treshold*std]
        slopes_over_I_peak  =
            slopes_over_I_peak[slopes_over_I_peak > peakbin -
            peak_treshold*std]

        mean_peak         = np.mean(slopes_over_I_peak)
        std_peak        = np.std(slopes_over_I_peak)
        skew_peak         = stats.skew(slopes_over_I_peak)
        kurt_peak       = stats.kurtosis(slopes_over_I_peak)
        peak_normfit      = stats.norm.pdf(slopes_over_I_peak,
            peakbin, std_peak)

        ax1.set_xlim([xmin, xmax])
        ax2.plot(slopes_over_I_peak, peak_normfit, "--",
            label="Peak Adjusted", lw=2.0, color="#f9c05c")
```

```
      ax2.legend(loc=2)
      print "probe %s" % probe_id[j]
      print " &%2f & %2f  & %4f & %4f" %
         (outlier_filtered_mean,  outlier_filtered_std,
         outlier_filtered_skew,  outlier_filtered_kurt)
      print "& %2f  &%2f &  %4f & %4f" % (mean_peak,
         std_peak,  skew_peak, kurt_peak)
      print
      if j == 1:
        ax1.set_xlabel("$(dI_S/dV)/I_S$      [1/V]")#  if
           probe_id[j] != "1" else "")
      else:
        pass

      ax2.set_ylabel("PDF Normalized")
      plt.tight_layout()
    if len(slopes_over_I) > 25:
      plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/"
         +
        "Python/slope_dist/figures/%s_date_%s_P%s_R%s"
        % (pklnamestring, str(start_datetime)[0:10],
           probe_id[j], swp_res))

    else:
      pass
    plt.show()
    plt.close()


  def plot_step_two(self, directory, macro, step,
     macro_only_p1,
    macro_only_p2, all_p1_macros, all_p2_macros,
       start_datetime,
    stop_datetime,probe_id,pbs, pkbo, swp_res, min_pass, ill,
    pklnamestring):

    nA = 1e-9
    for j in range(len(probe_id)):


      for i in range(len(macro)):
        print macro[i], probe_id[j]
        if probe_id[j] == "2" and macro[i] in macro_only_p1:
          pass
        elif probe_id[j] == "1" and macro[i] in macro_only_p2:
          pass
        else:
```

```
            A, B, I      = self.get_data(directory, macro[i],
                start_datetime, stop_datetime, probe_id[j])

            I_ph         = []
            t0           = []
            V_SC         = []

            i = 0
            while (i+1)*swp_res < len(A):

              V_B          = B['BIAS']
              Start_times      =
                 A.index.values[i*swp_res:(i+1)*swp_res]
              Stop_times       = A.loc[Start_times, u'
                 STOP_TIME(UTC)']
              SAA          = A.loc[Start_times, u' Illumination']
              Vk           = A.loc[Start_times,  u' Vph_knee']
              Swpfit     = self.get_Swpfit(pbs[0], pkbo[0], V_B,
                 Start_times, Stop_times, SAA, Vk, I, ill,
                 step[i])

              x        = np.array(Swpfit['I_pbs'])
              y        = np.array(Swpfit['slope'])


              plt.plot(x/nA,y/nA, '.')

              if len(x) < min_pass:
                pass
              else:
                beta, alpha = np.polyfit(x, y, deg=1)
                plt.title("Extrapolation points = %s" % len(x))
                I_ph_temp    = -alpha/beta
                xval = np.linspace(I_ph_temp/nA,np.max(x/nA),
                    10)
                t0.append(Start_times[0])
                V_SC.append(Vk)
                plt.plot(xval, beta*xval + alpha/nA)
                plt.show()

              i += 1


    def plot_step_one(self, directory, macro, step,
        macro_only_p1,
                      macro_only_p2, all_p1_macros,
                         all_p2_macros,
            start_datetime,stop_datetime,probe_id,pbs,
            pkbo, swp_res, min_pass, ill, pklnamestring):
      nA = 1e-9
      pkbo = pkbo[0]
```

```
    pbs = pbs[0]
    for j in range(len(probe_id)):
      for i in range(len(macro)):



        if probe_id[j] == "2" and macro[i] in macro_only_p1:
          pass
        elif probe_id[j] == "1" and macro[i] in macro_only_p2:
          pass
        else:
          A, B, I      = self.get_data(directory, macro[i],
              start_datetime, stop_datetime, probe_id[j])

          I_ph         = []
          t0           = []
          V_SC         = []

          i = 0
          while (i+1)*swp_res < len(A):
            V_B          = B['BIAS']
            Start_times      =
                A.index.values[i*swp_res:(i+1)*swp_res]
            Stop_times       = A.loc[Start_times, u'
                STOP_TIME(UTC)']
            SAA          = A.loc[Start_times, u' Illumination']
            Vk           = A.loc[Start_times,  u' Vph_knee']

            slope      = []
            intercept    = []

            I_pbs      = []
            bias_pbs   = []

            neg_slope    = []
            I_pbs_neg    = []
            for i in range(len(Start_times)):
              I_Swp = I.loc[Start_times[i], '0':]
              if SAA[i] == ill:
                if "NaN" in str(Vk[Start_times[i]]):
                  print "passed", str(Vk[Start_times[i]])
                else:
                  # print "macro",  macro[i]
                  # print "vk",   -Vk[Start_times[i]]

                  V_pbs_center  = -float(Vk[Start_times[i]]) -
                      pkbo
                  V_pbs_min     = (V_pbs_center - pbs/2)
                  V_pbs_max     = (V_pbs_center + pbs/2)
                  pbs_index     =
                     np.arange((np.abs(V_B-V_pbs_min)).argmin(),
```

```
                    (np.abs(V_B-V_pbs_max )).argmin(), 1)

                # pbs_index    =
                    np.arange((-int(round(np.float(Vk[Start_times[i]])))
                    - pkbo - pbs/2)*step[i] +
                    int(round(np.min(V_B)))*step[i],
                #
                    (-int(round(np.float(Vk[Start_times[i]])))
                    - pkbo + pbs/2)*step[i] +
                    int(round(np.min(V_B)))*step[i], 1)

                bias_pbs_temp    = np.array(V_B[pbs_index[0]
                      : pbs_index[-1]]).astype(np.float)
                I_pbs_temp       =
                    np.array(I_Swp[pbs_index[0] :
                    pbs_index[-1]]).astype(np.float)

                plt.figure()
                plt.plot(V_B, I_Swp, label="$I_S$")
                plt.plot(bias_pbs_temp, I_pbs_temp,
                    label="pbs")
                plt.legend()
                plt.show()
              if len(bias_pbs_temp) > 10:
                temp_slope, temp_intercept  =
                    np.polyfit(bias_pbs_temp, I_pbs_temp,
                    deg=1)
                ####################
                ## Negative slopes sorting
                ############################


                if temp_slope > 0:
                  I_pbs.append((V_pbs_center)*temp_slope +
                      temp_intercept)
                  bias_pbs.append(bias_pbs_temp)

                  intercept.append(temp_intercept)
                  slope.append(temp_slope)

                elif np.isnan(temp_slope) == True:
                  pass
                else:
                  neg_slope.append(temp_slope)
                  I_pbs_neg.append((V_pbs_center +
                      pbs/2)*temp_slope + temp_intercept)

            # x       = np.array(Swpfit['I_pbs'])
            # y       = np.array(Swpfit['slope'])
```

```
            # plt.plot(x/nA,y/nA, '.')
            # if len(x) < min_pass:
            #   pass
            # else:

            #   beta, alpha = np.polyfit(x, y, deg=1)
            #   plt.title("Extrapolation points = %s" %
                len(x))
            #   I_ph_temp   = -alpha/beta
            #   xval = np.linspace(I_ph_temp/nA,np.max(x/nA),
                10)
            #   t0.append(Start_times[0])
            #   V_SC.append(Vk)
            #   plt.plot(xval, beta*xval + alpha/nA)
            #   plt.show()

            # i += 1




def slope_dists_short(self, directory, probe_id, macro,
    step,
 macro_only_p1, macro_only_p2, start_datetime,
     stop_datetime,
 pkbo, pbs, ill, min_pass, swp_res, pklnamestring):

  for j in range(len(probe_id)):

    slopes_over_I     = []
    macro_dist_list   = []
    for i in range(len(macro)):
      # print "macro", macro[i], "probe", probe_id[j],
        "step", step[i]

      temp_slopes   = []
      temp_step     = step[i]
      if probe_id[j] == "2" and macro[i] in macro_only_p1:
        pass
      elif probe_id[j] == "1" and macro[i] in macro_only_p2:
        pass
      else:
        A, B, I     = self.get_data(directory, macro[i],
            start_datetime, stop_datetime, probe_id[j])

        k = 0
        while (k+1)*swp_res < len(A):
          V_B         = B['BIAS']
          Start_times     =
             A.index.values[k*swp_res:(k+1)*swp_res]
```

```
            Stop_times      = A.loc[Start_times, u'
                STOP_TIME(UTC)']
            SAA             = A.loc[Start_times, u' Illumination']
            Vk              = A.loc[Start_times,  u' Vph_knee']

            if macro[i] in ["506", "525"]:
              pkbo = [4]
              pbs  = [6]
              Swpfit    = self.get_Swpfit(pbs[0], pkbo[0],
                  V_B, Start_times, Stop_times, SAA, Vk, I,
                  ill, temp_step, macro[i])
            else:
              Swpfit    = self.get_Swpfit(pbs[0], pkbo[0],
                  V_B, Start_times, Stop_times, SAA, Vk, I,
                  ill, temp_step, macro[i])

            x       = np.array(Swpfit['I_pbs'])
            y       = np.array(Swpfit["slope"])

            if len(x) < min_pass:
              pass
            else:
              temp_slopes.extend(y/x)
            k += 1
          if len(temp_slopes) !=0:

            slopes_over_I.append(temp_slopes)
            macro_dist_list.append(macro[i])

          else:
            pass
        print "##############################################"
        print len(slopes_over_I), len(macro_dist_list)
        print str(start_datetime)[0:10]
        np.save("/home/sigvesh/Master/Thesis-Git-Repository/" +
                "Python/slope_dist/%s_slopelist_date_%s_P-%s_R-%s.npy"
                % (pklnamestring, str(start_datetime)[0:10],
                    probe_id[j], swp_res), slopes_over_I)
        np.save("/home/sigvesh/Master/Thesis-Git-Repository/" +
                "Python/slope_dist/%s_macrolist_date_%s_P-%s_R-%s.npy"
                % (pklnamestring, str(start_datetime)[0:10],
                    probe_id[j], swp_res), macro_dist_list)


        slopes_over_I    = np.load("/home/sigvesh/Master/" +
          "Thesis-Git-Repository/Python/slope_dist/" +
          "%s_slopelist_date_%s_P-%s_R-%s.npy"
          % (pklnamestring, str(start_datetime)[0:10],
              probe_id[j], swp_res))
```

```
        macro_dist_list = np.load("/home/sigvesh/Master/" +
          "Thesis-Git-Repository/Python/slope_dist/" +
          "%s_macrolist_date_%s_P-%s_R-%s.npy"
          % (pklnamestring, str(start_datetime)[0:10],
              probe_id[j], swp_res))

        print slopes_over_I
        def plot_beta_hist(ax, slopes, macro, length):
          # binwidth =
          # plt.xlim([-0.02,0])
          ax.hist(slopes,
              histtype="stepfilled",bins=np.arange(min(slopes),
              max(slopes) + binwidth, binwidth), alpha=0.6,
              normed=True, label="macro %s" % macro)
          ax.set_xlabel("$(dI_S/dV)/I_S$      [1/V]")
          ax.set_ylabel("Counts [#]")

        fig, ax = plt.subplots()
        for i in range(len(macro_dist_list)):
          plot_beta_hist(ax, slopes_over_I[i],
              macro_dist_list[i], len(slopes_over_I[i]))
        plt.title("Probe %s, date: %s - %s" % (probe_id[j],
            str(start_datetime)[0:10], str(stop_datetime)[0:10]))
        plt.legend(loc=2)
        plt.show()
        plt.savefig("/home/sigvesh/Master/Thesis-Git-Repository/Python/biweekly_slope
            % (pklnamestring, str(start_datetime)[0:10],
            probe_id[j], swp_res))
        plt.close()


    def run(self):
      directory      = "/home/sigvesh/Data/csv_datafiles/all/"
      filenames      = os.listdir(directory)
      plt.style.use('ggplot')


      macro          = [x[23:26] for x in filenames]
      macro          = sorted(macro)

      # letter_counts = Counter(macro)
      # df = pd.DataFrame.from_dict(letter_counts,
          orient='index')
      # df.columns = ["Number of files"]
      # df = df.sort_index()
      # df.plot(kind='bar')
      # plt.xlabel("Macro name")
      # plt.ylabel("Number of files")
      # plt.show()

      macro          = list(set(macro))
```

```
# 304 p1 p2, 307 p1 p2, 510p2, 916p1 p2, 926 p1 p2
    excluded (does not have enough data to calc l_ph)

# raw_macros    = ['202', '203', '212', '301', '305',
    '306', '410', '411', '412', '414', '416',
#           '417', '501', '505', '506', '510', '515',
    '516', '517', '525', '600', '601', '602', '603',
#           '604', '610', '611', '612', '613', '615',
    '616', '617', '624', '710', '805', '807', '814',
#           '816', '827', '901', '904', '910', '914']

# clean_macros  = ['202', '203', '212', '301', '305',
    '306', '410', '411', '412', '414', '416',
#           '417', '501', '510', '600', '601', '602',
    '603',
#           '610', '611', '612', '613', '615', '616',
    '710', '805', '814',
#           '816', '901', '904', '910', "914"]

all_p1_macros = ['202', '203', '212', '305', '306',
    '414', '417', '501', '505', '506', '510',
           '515', '516', '517', '525', '600', '601',
              '602', '603', '604', '610', '611', '612',
              '613',
           '615', '617', '624', '805', '807', '814',
              '816', '827', '901', '904', '910', '914']


all_p2_macros   = ['202', '212', '301',  '305', '306',
    '410', '411', '412', '414', '416', '501',
           '505', '506', '515', '516', '517', '525',
              '600', '602', '603', '604', '610', '611',
           '612', '613', '616', '624', '710', '901',
              '904',  '914']


macro_only_p1   = ["203", "417", "510", "601", "615",
    "617", "805", "807", "814", "816","827", "910"]

macro_only_p2   = ["301", "410", "411", "412", "416",
    "616", "710"]


# macros_U5     = ["517", "612", "710"]
# macros_U25    = ["412", "416", "417", "624", "615",
    "616", "617", "914"]


# plotting dates
```

```
    # start_datetime      = ["2015-01-01T00:00:00.000000"]
    # stop_datetime       = ["2015-02-01T00:00:00.000000"]
    # pklnamestring     = "comparison_of_two_macros"

    # start_datetime      = ["2014-03-01T16:00:59.531502"]
    # stop_datetime       = ["2016-09-01T17:57:34.004082"]
    # pklnamestring     = "trusted_overview"

    # start_datetime      = ["2014-03-01T16:00:59.531502"]
    # stop_datetime       = ["2016-09-01T17:57:34.004082"]
    # pklnamestring     = "expanded_overview"

    # start_datetime      = ["2014-03-01T16:00:59.531502"]
    # stop_datetime       = ["2015-08-13T02:03:00.000000"]
    # pklnamestring     = "trusted_inbound"

    # start_datetime      = ["2015-08-13T02:03:00.000000"]
    # stop_datetime       = ["2016-09-01T17:57:34.004082"]
    # pklnamestring     = "outbound"



    # start_datetime      = pd.date_range('2014-03-01',
        periods=31, freq='2w', dtype='datetime64[ns]')
    # stop_datetime     = pd.date_range('2014-04-01',
        periods=31, freq='2w', dtype='datetime64[ns]')
    # pklnamestring       = "expanded_Monthly"


    start_datetime      = pd.date_range('2014-06-01',
        periods=62, freq='2w', dtype='datetime64[ns]')
    stop_datetime     = pd.date_range('2014-06-15',
        periods=62, freq='2w', dtype='datetime64[ns]')
    pklnamestring       = "expanded_double_weekly"

    # start_datetime      = pd.date_range('2015-10-01',
        periods=31, freq='m', dtype='datetime64[ns]')
    # stop_datetime     = pd.date_range('2015-11-01',
        periods=31, freq='m', dtype='datetime64[ns]')
    # pklnamestring     = "trusted_monthly"


    # start_datetime      = ['2015-04-01T00:00:00.0000000']
    # stop_datetime       = ['2015-05-01T00:00:00.0000000']
    # pklnamestring     = "noise_cluster_April_2015"

    ill       = 1.0           # Probe Illumination 1,0 = on,
        off

    probe_id    = ["1","2"]
    min_pass    = [10, 20, 30]
```

```
swp_res      = [15, 40, 50]

min_pass     = [20]
swp_res      = [40]



################ expanded macro list
   ##############################################3#

# macros = ["506", "525", "410", "412", "414", "416",
    "417", "510", "525", "516", "517", "610", "611",
    "612", "613", "604"
#        , "614", "624", "634", "615", "616", "617",
    "710", "814", "805", "815", "816", "807", "817", "827"
#        , "900", "910", "901", "903", "904", "914", "905"]
# steps  = [2, 2, 4, 4, 4, 2, 4, 4, 4, 2, 2, 2, 2, 2, 2,
    4, 4, 4, 4, 4,
#        4, 4, 2, 4, 4, 4, 4, 4, 4, 4, 4, 2, 4, 4,4, 4, 2]

macros = ["516", "517", "604", "610", "612", "613",
    "615", "624", "901", "910", "914", "506", "525"]
steps  = [2, 2, 4, 2, 2, 2, 4, 4, 4, 2, 4, 2, 2]

####################Expanded filtered macro list

# macros = ["516", "517", "610", "612", "613", "615",
    "901", "910", "914"]
# steps  = [2, 2, 2, 2, 2, 4, 4, 2, 4]


pkbo         = [20]
pbs          = [10]

# special_macros  = ["506", "525"]
# special_steps   = [2,2]

#####################trusted macro
    list#################################################################
# macros      = ["412", "416", "417", "624", "615",
    "616", "617", "914", "517", "612", "710"]
# steps       = [4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2]


for i in range(len(start_datetime)):
  for j in range(len(swp_res)):


    # self.run_pkl(directory, macros, steps,
        macro_only_p1, macro_only_p2, all_p1_macros,
        all_p2_macros,
```

```
        #   start_datetime[i],stop_datetime[i],probe_id,pbs,
            pkbo, swp_res[j], min_pass[j], ill, pklnamestring)

        # self.overview_plot_with_filter(start_datetime[i],
            stop_datetime[i], min_pass[j], swp_res[j],
            probe_id,
                        # macros, macro_only_p1,
                            macro_only_p2, pklnamestring)

        self.slope_distribution_analysis(directory, probe_id,
            macros, steps, macro_only_p1, macro_only_p2,
                   start_datetime[i], stop_datetime[i], pkbo,
                       pbs, ill, min_pass[j], swp_res[j],
                       pklnamestring)


        self.slope_dists_short(directory, probe_id, macros,
            steps, macro_only_p1, macro_only_p2,
                   start_datetime[i], stop_datetime[i], pkbo,
                       pbs, ill, min_pass[j], swp_res[j],
                       pklnamestring)


if __name__ == '__main__':
  test = Photo()
  test.run()
    # Ps: I've decided to sell my vacuum. Well, it was just
        gathering dust!
```