Uptec F10041

Examensarbete 30 hp
Juli 2010

Determination of Ion Number Density from Langmuir Probe Measurements in the Ionosphere of Titan

Oleg Shebanits

UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Determination of Ion Number Density from Langmuir Probe Measurements in the Ionosphere of Titan

Oleg Shebanits

Saturn’s largest moon, Titan, presents a very interesting subject for study because of its atmosphere’s complex organic chemistry. Processes taking place there might shed some light on the origins of organic compounds on Earth in its early days. The international spacecraft Cassini-Huygens was launched to Saturn in 1997 for a detailed study of the gas giant and its moons, specifically Titan.
The Swedish Institute of Space Physics in Uppsala has manufactured the Langmuir probe instrument for the Cassini spacecraft now orbiting Saturn, and is responsible for its operation and data analysis. This project concerns the analysis of Titan’s ionosphere measurements from this instrument, from all “deep” flybys of the moon (<1400km altitude) in the period October 2004 - April 2010.
Using the Langmuir Probe analysis tools, the ion flux is derived by compensating for the atmospheric EUV extinction (that varies with the photoelectron current from the probe). The photoelectron current emitted from the probe also gives an artifact in the data that for this project needs to be deducted before analysis. This factor has already been modeled, while the extinction of Titan’s atmosphere has only been taken into account on event basis (not systematically). The EUV corrected ion flux data is then used to derive the ion number density in Titan’s atmosphere, by setting up an average ion mass altitude distribution (using the Ion Neutral Mass Spectrometer results for comparison) and deriving the spacecraft speed along the Cassini spacecraft trajectory through Titan’s ionosphere.
The ion number density results proved to correlate very well with the theoretical ionospheric profiles on the day side of Titan (see graphical representation in the Results section). On the night side, a perturbation of the ion flux data was discovered by comparison with Ion Neutral Mass Spectrometer data, supporting earlier measurements of negative ions reported by Coates et al 2009.
The project was carried out at the Swedish Institute of Space Physics (Institutet för Rymdfysik, IRF) in Uppsala.

Handledare: Jan-Erik Wahlund, Karin Ågren
Ämnesgranskare: Mats André
Examinator: Tomas Nyberg
ISSN: 1401-5757, Uptec F10041

Sammanfattning

Saturnus största måne Titan är ett väldigt intressant forskningsobjekt på grund av dess atmosfärs komplexa organiska kemi. Processer som pågår i Titans täta atmosfär kan hjälpa oss att förstå ursprunget till organiska föreningar på Jorden i dess unga ålder. Den internationella rymdsonden Cassini-Huygens blev uppskjuten mot Saturnus 1997, för att i detalj undersöka gasjätten och dess månar, speciellt Titan.

Institutet för Rymdfysik (IRF) i Uppsala är ansvariga för operation och dataanalys av Langmuirsonden ombord Cassini som ligger i omloppsbanan kring Saturnus sedan 2004. Detta projekt omfattar analys av Langmuirsondens mätningar av Titans jonosfär från alla ”djupa” förbiflygningar av månen under perioden oktober 2004 – april 2010.

Med hjälp av analysverktygen för Langmuirsonden, tas jonflödet fram efter kompensation för den atmosfäriska EUV extinktionen som ger upphov till fotoelektronströmmen från sonden. Fotoelektronströmmen som utsänds från proben ger en artefakt i data och måste (för detta projekt) korrigeras före analysen. Denna faktor är redan bestämd, men extinktionen av Titans atmosfär har endast korrigerats för i enstaka fall. Det korrigerade datat används för att få fram jondensiteten i Titans atmosfär genom att en genomsnittlig jonmass/höjd fördelning antas (jämförs med resultat från INMS-instrumentet) och kombineras med den beräknade hastighet som Cassini håller i banan genom jonosfären.

Projektet utfördes vid Institutet för Rymdfysik, Uppsala.

Acknowledgements

First of all I'd like to thank my supervisors, Jan-Erik Wahlund and Karin Ågren, for their guidance and for giving me this unique and exciting opportunity to work in the frontline of the Saturn system research. I would also like to thank the rest of the Cassini/Titan group at IRF Uppsala, Michiko Morooka and Niklas Edberg, for letting me use their Matlab programs and helping to understand them, it inspired me on several ideas of the data analysis and visualization. A special thanks to Ronan Modolo (LATMOS-IPSL, UVSQ, CNRS-INSU[1], Guyancourt, France), for taking his time to in great detail explain his and Malin Westerberg's photoelectron current routines and theory behind them. Another special thanks to Kathleen E. Mandt (Southwest Research Institute, San Antonio, US), for providing the INMS data which made possible a large part of the interpretation of results. Thanks to Anders Eriksson for sharing his extensive Matlab and SSH knowledge and to Bertil Segerström for his supreme technical support. Thanks to Mats André for revising and improving the final report. And last but not least, thanks to my colleagues, Marco Chiaretta, Christian Hånberg and Madeleine Holmberg for their friendship and creative discussions during the breaks, as well as the rest of the IRF Uppsala personnel for the warm atmosphere of space physics science.

[1] See Abbreviations below

Contents

Abbreviations

amu – atomic mass unit

DOY – Day Of Year (1-365)

ELS – Electron Spectrometer

ESA – European Space Agency

EUV – Extreme UltraViolet (radiation)

IND – Ion Number Density

INMS – Ion Neutral Mass Spectrometer

IRF – *Institutet för RymdFysik*, Swedish Institute of Space Physics

JPL – Jet Plasma Laboratory (NASA)

ASI - *Agenzia Spaziale Italiana*, Italian Space Agency

LATMOS-IPSL, UVSQ, CNRS-INSU - *Laboratoire Atmospheres Milieux et Observations Spatiales - Institut Pierre Simon Laplace / Universite de Versailles Saint Quentin / Centre National de la Recherche Scientifique - Institut National des Sciences de'Univers*

LP – Langmuir Probe

LT – Local Time

MIM – Mean Ion Mass

NASA - National Aeronautics and Space Administration

RPWS – Radio and Plasma Wave Science

s/c – Spacecraft

TCC – Titan-Centered Co-rotational

TSE – Titan Solar Ecliptic

1 Introduction

Saturn – Titan system

The Saturn System has striking similarities to the Solar System itself. Saturn is the sixth planet of our Solar System, easily recognized among others due to its distinct rings. It has more than 60 known moons, each with its own unique history and features. For instance, icy Enceladus has geysers coming from its subterranean ocean, which also makes it a possible host for extraterrestrial life. Another possibility for life to thrive in the Saturn system is on Saturn's largest moon Titan, the only moon in the Solar System with a fully developed atmosphere. And tiny Rhea even has its very own ring system (*Jones et al*, 2008[I]).

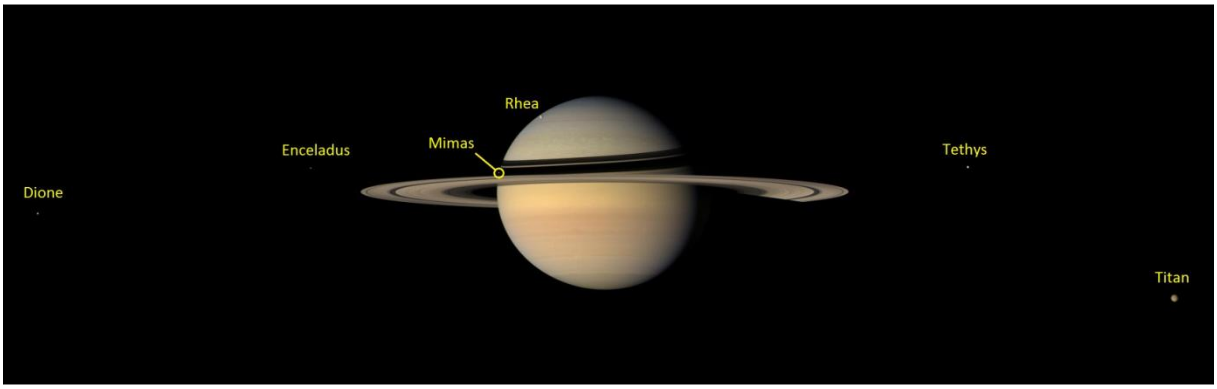

Figure 1.1 Saturn as seen by Cassini from a close encounter with the moon Iapetus. Source: NASA JPL[II]

The atmospheric layer of Titan is in fact so thick that the moon was mistaken to be the largest in the Solar System, corrected to the second largest (after Jupiter's Ganymede) only after arrival of Voyager 1 in 1980. This project is devoted solely to Titan and one of its atmospheric features, namely ionosphere.

Titan's Ionosphere

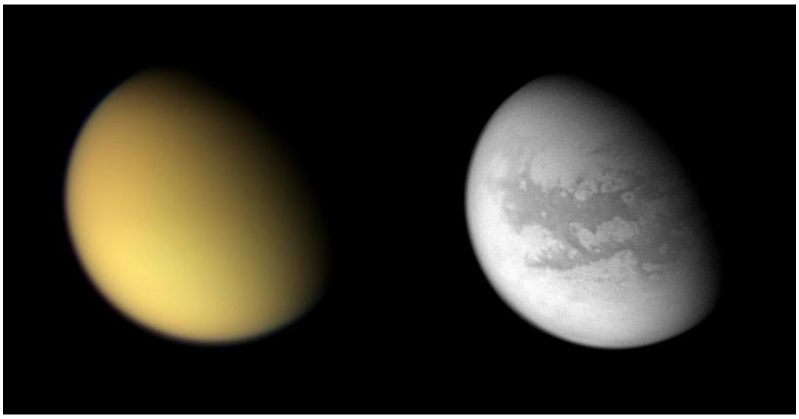

Figure 1.2 Titan's atmosphere in natural colors (left) and infra-red, revealing the surface (right). Source: NASA JPL[II]

The thick atmosphere of Titan (Figure 1.2) consists mainly of nitrogen N_2, with a few percent by number density of methane CH_4 (*Waite et al.*, 2007)[III], and a number of hydrocarbons. Under solar EUV radiation, X-rays and impact of Saturn's magnetospheric plasma, Titan's atmosphere becomes ionized and forms an ionospheric layer between 1000 and 2000km, with a typical "banana shape" ion profile on the day side due to solar radiation being dominant (Figure 1.3). The ionosphere consists mostly of positive ions, with a small part of recently discovered heavy negative ions (up to 10000 amu/q) at altitudes below 1150km (*Coates et al.*, 2007)[IV].

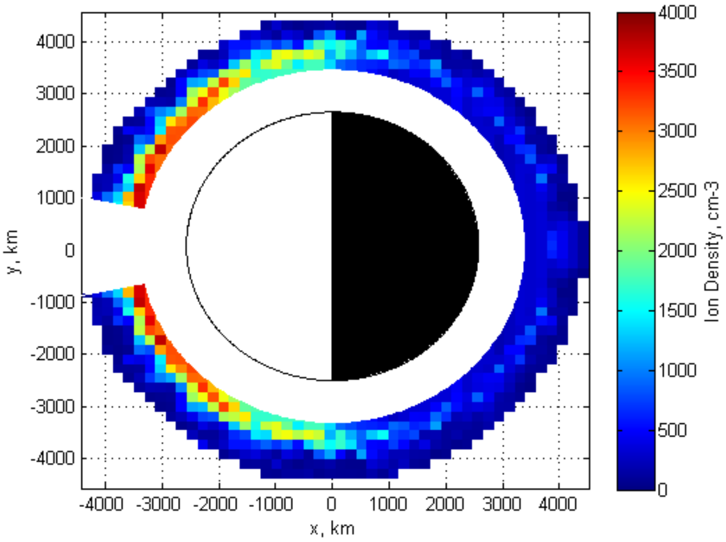

Figure 1.3 "Banana-shape" of Titan's Ionosphere Ion profile, showing day (left) and night (right) sides of the moon. The data is limited by Cassini's coverage of the moon.

Along with the discovery of liquid hydrocarbon lakes on Titan's surface by the Huygens probe, the moon resembles early Earth, although at lower temperature, thus making it one of the most interesting study objects in the Solar System, offering answers to questions about the origins of complex organic compounds (or even life) on Earth and a possibility of hosting extraterrestrial microorganisms (*McKay et al*, 2005[V]).

Cassini Equinox Mission

Cassini-Huygens spacecraft is a joint NASA/ESA/ASI mission that was launched in 1997 and arrived at the Saturn system in 2004. The two-parts composed spacecraft is named after Italian-French astronomer Giovanni Domenico Cassini, who gave name to the main orbiter, and Dutch astronomer, mathematician and physicist Christiaan Huygens, who gave name to the probe aimed at descending through Titan's atmosphere. Both scientists are famous for their discoveries of Saturn moons and ring system. Huygens probe detached from the Cassini spacecraft on December 25th 2004 and successfully landed on Titan's surface on January 14th 2005, transmitting measured data on the way down. After a funding extension in 2008 the mission was renamed to Cassini Equinox Mission.

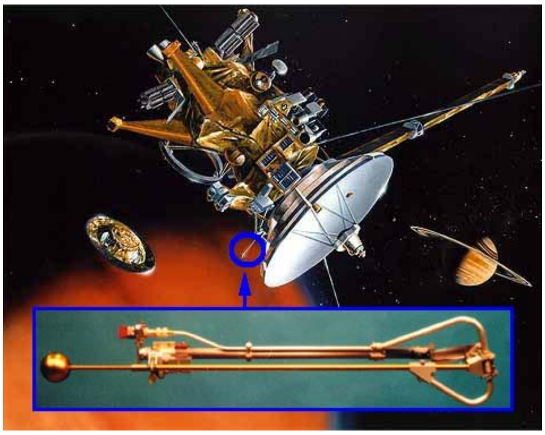

Figure 1.4 Artist's concept of the Cassini spacecraft just after detaching the Huygens probe. The image shows placement of the LP (blue ring) and the instrument itself in close-up. Source: Swedish Institute of Space Physics, NASA JPL.

The Langmuir Probe

A vast majority of visible matter in our universe is in its forth state, i.e. an ionized gas called plasma. All stars, nebulas and interstellar dust partly or completely consist of charged particles and ions. When radiation from a star hits an atmosphere, electrons are removed from neutral atoms and molecules, forming a plasma layer of ions and free electrons, ionosphere. Plasma behaves differently from neutral gas because it can be affected by external electromagnetic fields. Plasma is also a conductor and can therefore carry electromagnetic fields itself, resulting in quite complex interactions between the external fields and the charged particles in plasma.

The Langmuir Probe instrument onboard Cassini was developed and manufactured by the Swedish Institute of Space Physics in Uppsala (IRF) to measure plasma parameters such as ion and electron densities, temperatures, ion speed, mean ion mass and EUV intensity - about the same parameters a weather station measures on Earth (wind speed, temperature and pressure). Thus it is a perfect tool for studying ionospheric plasma.

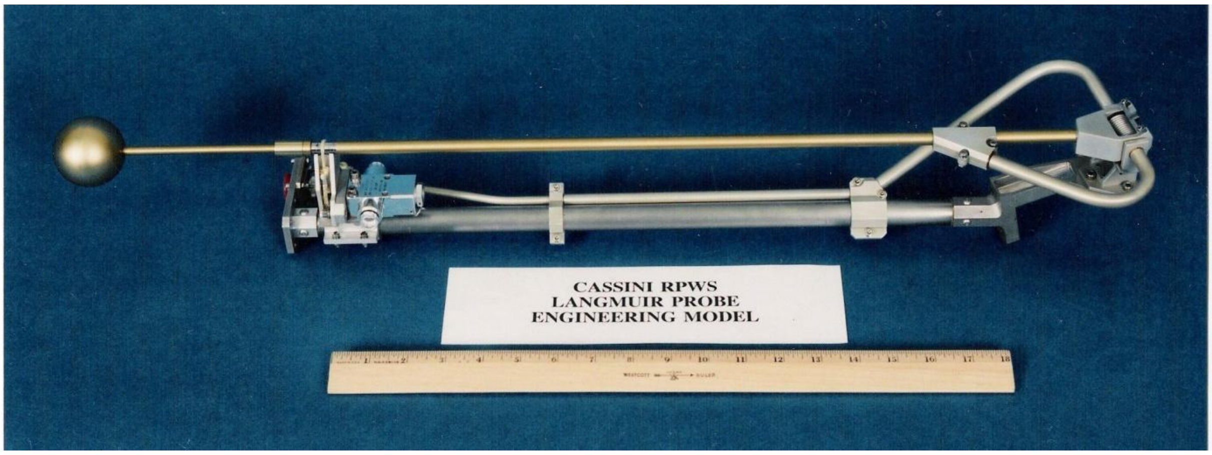

Figure 1.5 Cassini's Langmuir Probe, diameter 5 cm. The picture is taken from Swedish Institute of Space Physics homepage[VI].

The principle behind the LP is simple: by putting a spherically shaped (for symmetry) probe in plasma and applying electric potential (voltage U_{bias}), particles with opposite charge are attracted. If U_{bias} is positive, electron current (also negligible negative ion current) is received, if the applied voltage is negative, positive ion current is received. The LP does frequent voltage sweeps, measuring the current. During a sweep in Saturn's magnetosphere (once every 10 min) voltage is varied from -32V to +32V in under 0.5s and 512 steps and then resetting back to -32V. For reference, a measured floating potential U_{float} is used (related to the spacecraft potential). Around Titan, the sweep is done once every 24 seconds, the interval is ±4V divided in 1024 steps for more precise measurements. Unfortunately, determining U_{float} requires the electron part of the analysis (which was not derived for all flybys at the time), but it can be approximated by U_{zero}, a potential of zero current through the probe. For this project, only the ion part was needed, hence only the bias range of $U_{float} + U_{bias} < 0$ in the sweeps is of interest. More detailed theory is presented in next chapter.

Up until the start of and during this project, Cassini has completed 68 flybys of Titan (labeled TA, TB, TC and T3-T67). Out of those, 15 were excluded (**TC, T4, T7, T9, T11, T24, T35, T37, T45, T53, T60, T63, T64, T66** and **T67**) due to either too large distance from Titan (> 2000km altitude, i.e. outside the ionosphere) or broken/missing data from the Langmuir Probe instrument, leaving 53 flybys with ionospheric sweep data in an altitude interval 900-2000km.

2 The Model

2.1 Theory and Observations

As mentioned before, during a deep (<1400 km altitude) flyby of Titan, the LP sweeps U$_{bias}$ from -4V to +4V in 1024 steps, and the ion current is extracted when $U_{float} + U_{bias} < 0$. It is also possible to measure the ion current for positive voltage because ions are too heavy to be repelled and still hit the probe, but that current is negligible compared to the electron current.

The measured ion current I_i then is

$$I_i = I_{i0}(1+\chi_i) \tag{2.1.1}$$

where

$$\begin{cases} I_{i0} = -q_i n_i A_{LP}\sqrt{\dfrac{v_i^2}{16}+\dfrac{eT_i}{2\pi m_i}} \\ \chi_i \approx \dfrac{U_{float}+U_{bias}}{\frac{m_i v_i^2}{2e}+T_i} \end{cases} \tag{2.1.2}$$

q_i, n_i, m_i and T_i are ion charge, number density, mass and temperature (in electron-volt, eV) respectively, A_{LP} is area of the probe and e is electron charge.

Assuming $\frac{m_i v_i^2}{2e} >> T_i[eV]$, $-e = q_i$ and substituting spherical area of the probe $A_{LP} = 4\pi r_{LP}^2$ into Equation (2.1.2), yields

$$\begin{cases} I_i \approx -en_i A_{LP}\dfrac{|v_i|}{4}\left(1+\dfrac{2e}{m_i v_i^2}\left[U_{float}+U_{bias}\right]\right) \\ \dfrac{dI_i}{dU_{bias}} = en_i A_{LP}\dfrac{e}{2m_i|v_i|} = en_i \pi r_{LP}^2 \dfrac{2e}{m_i|v_i|} \end{cases} \tag{2.1.3}$$

From measurements, the ion current is derived as a fitted DC model

$$\begin{cases} m = a + b\cdot U_{bias} + I_{ph} \\ a = I_{i0} = -en_i \pi r_{LP}^2 |v_i| \\ b = -\dfrac{dI_i}{dU_{bias}} = -en_i \pi r_{LP}^2 \dfrac{2e}{m_i|v_i|} \end{cases} \tag{2.1.4}$$

where m is the measured current (affected by the constant U_{float}), a is the actual ion current (gives the ion flux, $n_i|v_i|$), b is the voltage derivative of m and I_{ph} is the photoelectron current. After analysis of the raw LP data (with Beta_Ion program), measured m and calculated b are received, yielding the ion number density and speed:

$$a \cdot b \propto n_i^2$$
$$a / b \propto v_i^2, m_i \qquad (2.1.5)$$

Ion speed at low altitudes is assumed to be equal to spacecraft speed (around 6km/s) due to frequent collisions between the ions and neutrals as well as both being at least 2000 times heavier than electrons and thus having much slower speed at the same kinetic energy levels (temperature).

2.2 LP data analysis tools, general

Beta_Ion and SweepMap

Beta_Ion and *SweepMap* are programs for analysis of LP output data. Beta_Ion is a modified version of original program Beta but without the electron analysis part. Beta_Ion extracts the ion part of the LP sweeps ($U_{float} + U_{bias} < 0$) shown in Equation (2.1.4): ion current $m=I_i$, its derivative $b=dI_i/dU$, and the zero crossing potential U_{zero}. Figure 2.1 shows part of the Beta analysis with electron parts included. The solid green lines mark the first electron population, its peak defining U_{float} (the lowest plot). U_{zero} is the voltage of zero current through the probe, as shown in the middle plot. Although the difference between U_{float} and U_{zero} is not negligible, the behavior of both is very similar, allowing for a good enough approximation for the cases when U_{float} is unavailable.

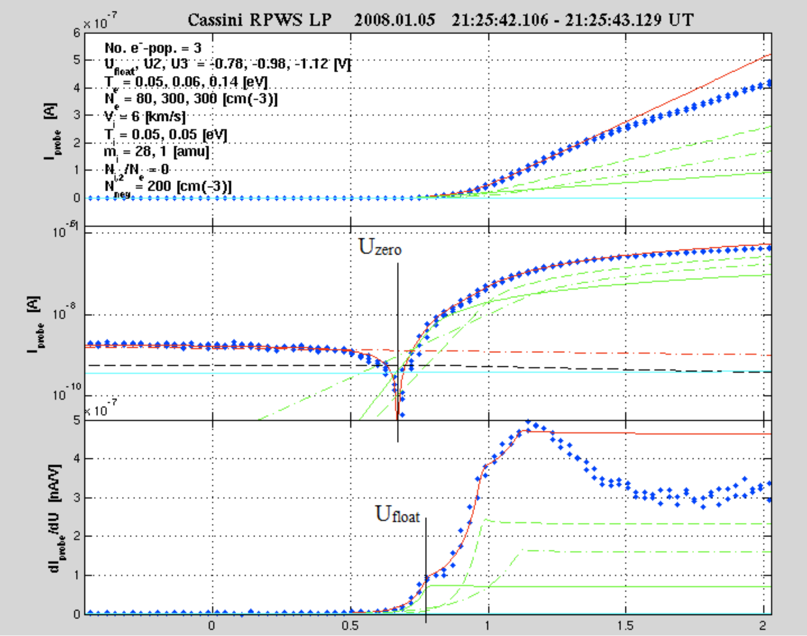

Figure 2.1 An LP sweep during Titan flyby T40, x-axis is voltage U (in V). The ion part is seen on the left side of the first plot, where the voltage drops below zero. The ion current fit is shown as a solid teal line. U$_{float}$ is derived from the peak of the first electron population (solid green line).

SweepMap gives a color-coded plot of the measured current *I* and *dI/dU* versus U_{bias}, plotted over time from all sweeps in a flyby time interval (Figure 2.2). SweepMap plots were used as complimentary to Beta_Ion output to identify ionospheric passages. Both programs have to be run individually for each flyby.

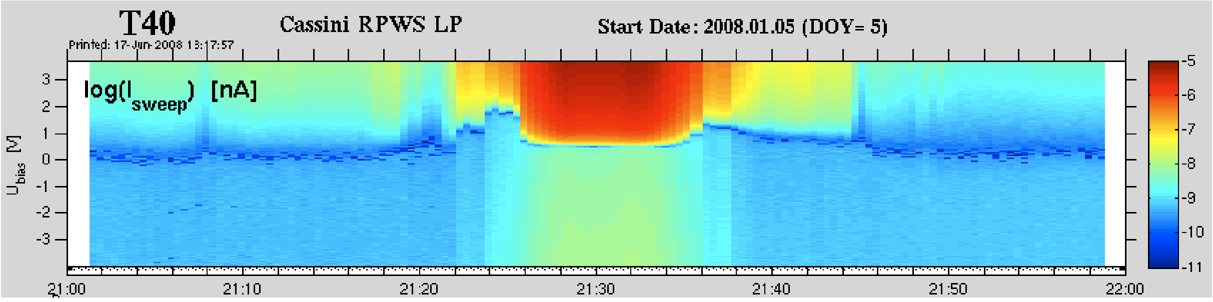

Figure 2.2 SweepMap output for T40. Redder parts of the spectra mark the electron current from the passage through the ionosphere.

Cassini RPWS ephemerides

The Cassini ephemeris tool from the Cassini Radio and Plasma Wave Science team homepage[VII] was used to get coordinates of Cassini in Titan Centered Co-rotational (TCC) and Titan Solar Ecliptic (TSE) systems. The coordinate systems are defined as follows: TCC has x-axis pointing at Saturn, z-axis perpendicular to Titan's orbital plane (along the moon's orbital rotational axis) and y-axis in the opposite direction of Titan's orbital velocity. TSE has x-axis pointing at the Sun (not same as x_{corot}), z-axis perpendicular to Saturn's orbital plane (along the planet's orbital rotational axis) and y-axis in the opposite direction of Saturn's orbital velocity. Both systems are right-handed ($\vec{z} = \vec{x} \times \vec{y}$).

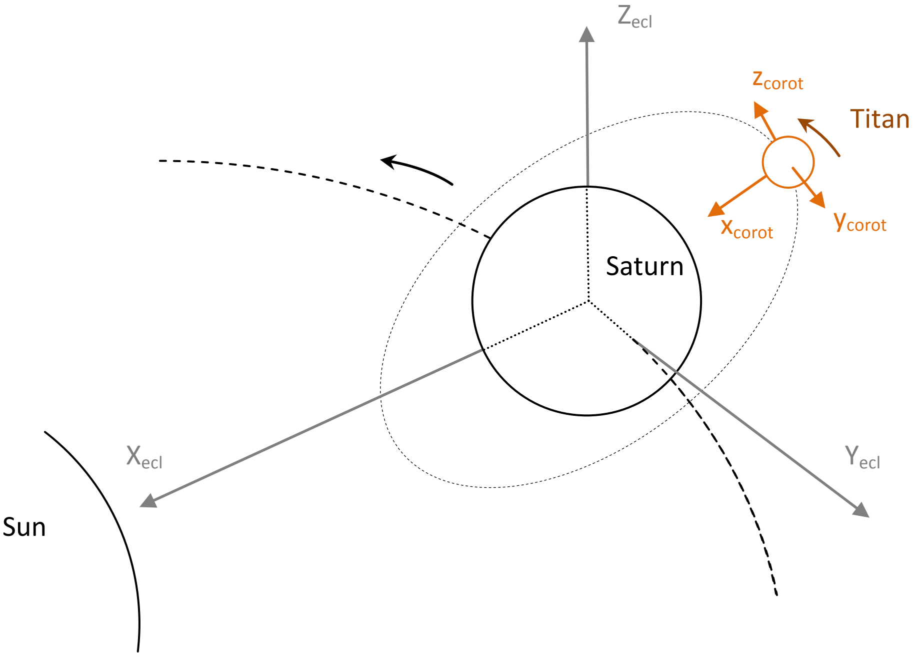

Figure 2.3 Co-rotational and Ecliptic coordinate systems. The TSE system is the same as Saturn Solar Ecliptical, but centered on Titan.

The Cassini ephemeris tool gives time (*Year, DOY, h-m-s*), space coordinates *x, y, z,* distance to origin *R* and spatial components of velocity v_x, v_y, v_z as well as speed $|v|$ in a desired coordinate system.

The TCC system is only used to identify passages of Titan's wake in Saturn's magnetospheric co-rotating plasma, purely for flyby classification reasons as Saturn's magnetospheric plasma plays only a minor role in the formation of the day-side ionosphere. TSE ephemerides were used to extract the exact position of the spacecraft relative to Titan and the Sun during the sweeps, ultimately yielding altitude and Solar Zenith Angle parameters. Ephemerides have a default time interval of one minute and must be interpolated onto LP sweep time vector given by the instrument. Although it is possible to set any time interval in the ephemeris tool, such direct approach would require a 10 milliseconds accuracy to match the time of LP sweeps, making each ephemeris data file colossal in size and therefore very slow to process.

Neutrals Population Grid and Photoelectron Current

Modeling and calculating of neutral densities, photoionization and the photoelectron current profile along Cassini trajectories was performed with Matlab programs written by Ronan Modolo and Malin Westerberg (2007), with a slight modification to match the coordinate systems used in this project (TSE). The programs create a 3D-grid around Titan (±5500km from center of the moon in x, y and z directions) with homogenously spaced data points (every 100km).

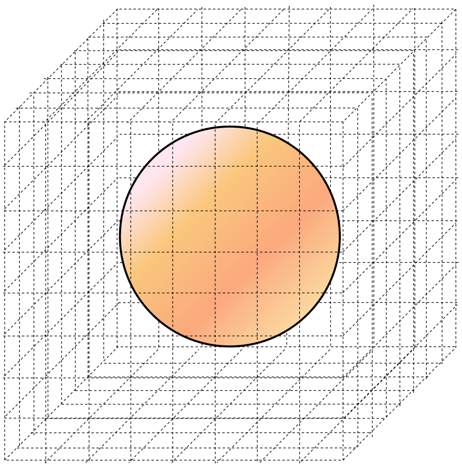

Figure 2.4 The 11000x11000x11000km grid box around Titan, to scale. Titan in this picture is shown without the atmospheric layer.

At each point, the neutral particle density is calculated for nitrogen N_2 and methane CH_4. The optical depth for both species is then derived as a function of altitude, wavelength and solar zenith angle. In this model only wavelengths up to 100nm are used, meaning that Ly-α ultraviolet radiation (121.6nm) is not included. With the optical depth modeled onto the grid, the intensity of the solar EUV along the Cassini trajectory can be calculated using a 3D-interpolation of the grid points onto LP sweep points (position of which in space is specified by the interpolated ephemeris coordinates). Thus, the photoelectron current can be estimated using the F10.7 index. F10.7 is a measurement of the solar radiation flux (given in W/m^2Hz) for the wavelength of 10.7 cm. The measurement is performed by the solar radio telescope of the Canadian National Research Council Dominion Radio Astrophysical Observatory (Penticton, Canada). However, for this project, the F10.7 proved to be insufficient. Instead, the theoretical F10.7 photoelectron current has been normalized (by maximum value) to LP measurements outside Titan's ionosphere (photoelectron current dominating).

Photoelectron current data correction

With the photoelectron current calculated, the ion current measurements can be corrected using the same Matlab programs that were written and used for the determination of the solar EUV and ion flux in Saturn's magnetosphere by Madeleine Holmberg (2009), again with slight modifications to better suit Titan's surroundings: the photoelectron and spacecraft attitude correction (when the LP is in the shadow of Cassini) were removed for passages of Titan's shadow. Figure 2.5 shows the corrected ion current for T51.

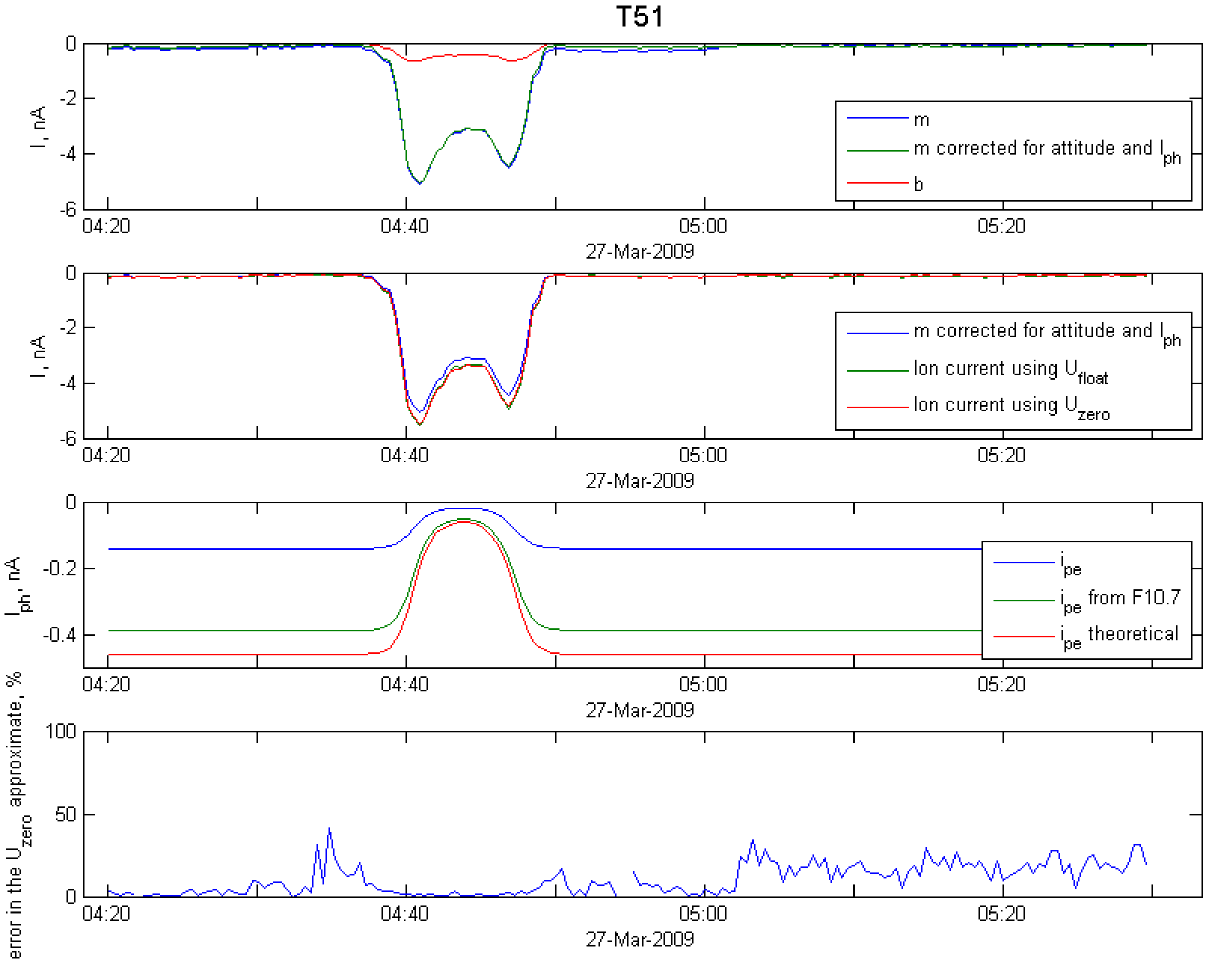

Figure 2.5 Photoelectron correction of the ion current. The first plot shows raw outputs *m* and *b* seen in Equation (2.1.4), as well as *m* corrected for attitude and photoelectron current. The second plot shows the ion current calculated with *U_{float}* resp *U_{zero}*. The third plot shows photoelectron currents, F10.7 model (green), theoretical calculated from the 3D-grid box (red) and normalized to LP measurements outside ionosphere (blue). The last plot shows the difference between *U_{float}* and *U_{zero}* in percent.

Ion Neutral Mass Spectrometer (INMS) contribution

The INMS instrument onboard Cassini measures mean mass of positive ions and neutral particles in the upper atmosphere (therefore also ionosphere) of Titan and the magnetosphere of Saturn, as well as particle environments of Saturn's rings and icy moons[VIII]. There are however three limits for the INMS instrument: it is only able to measure positive ion masses, only up to 100 amu (meaning that the measurements are often not very accurate at low altitudes) and does not operate on every flyby of Titan. For this project, INMS data from only 14 flybys was available.

If ion mass and ion number density (from the ion current) are known, ion speed can be derived from Equation (2.1.4):

$$|v_i| = -en_i\pi r_{LP}^2 \frac{2e}{m_i b} \qquad \textbf{(2.2.1)}$$

2.3 LP data analysis tools, project-specific

Mean Ion Mass (MIM)

With the corrected LP measurements of the ion current and the assumption that v_i=v_{sc}, the mean ion mass is calculated using Equation (2.1.5):

$$\frac{a}{b} = |v_i| / \frac{2e}{m_i |v_i|} = \frac{m_i v_i^2}{2e}$$
$$\Rightarrow m_i = \frac{2ea}{bv_i^2} \qquad \textbf{(2.3.1)}$$

The output is stored as raw data (MIM over altitude and SZA) and as a surface function fitted to the results (Figure 3.6), the same pattern is also used for saving the ion number density results. The surface function can be evaluated to extract all interpolated and original data points.

Ion Number Density (IND)

The IND is calculated with two different methods.

1. For all 14 flybys where the INMS MIM data is available (**T5**, **T17**, **T18**, **T21**, **T26**, **T32**, **T36**, **T39**, **T40**, **T48**, **T50**, **T51**, **T57** and **T59**), the IND is calculated from Equation (2.1.4):

$$n_i = \frac{1}{e\pi r_{LP}^2} \sqrt{\frac{abm_i}{2e}} \qquad \textbf{(2.3.2)}$$

where *a* is the corrected ion current and *b* is the dI_i/dU_{bias} (Beta_Ion outputs). This offers the most exact results due to all parameters being measured, but because only 14 flybys are covered the interpolation results have poorer quality. In addition, because INMS measurements of the ion mass are limited to 100 amu, m_i is not very accurate below ≈1000 km.

2. For all deep flybys (≤1400 km) except those excluded due bad or broken data, the IND is calculated from Equation (2.1.5), assuming v_i=v_{sc}:

$$n_i = -a \big/ ev_i \pi r_{LP}^2 \qquad \textbf{(2.3.3)}$$

This method offers better data resolution for the interpolation, but is only accurate up to approximately 1400km altitude where the assumption v_i=v_{sc} holds.

Both methods are therefore complimentary to each other.

Ion Speed

The assumption v_i=v_{sc} is tested by comparing the spacecraft speed with the ion speed derived from the LP and INMS data with Equation (2.2.1). As mentioned above and seen in the Results section below, above 1400km altitude this assumption no longer holds. This is due to low ion and neutral densities which reduce collision interactions and allow for a longer mean free path.

3 Results

Titan's ionosphere is (in theory) symmetrical around the x_{ecl} axis of the TSE system, hence a convenient representation format of the output data has been chosen in the polar spherical coordinates based on the TSE system (Figure 3.1), with Solar Zenith Angle as one dimension, position vector $\vec{r}$ converted to altitude as second and the data value as third. The data from measurement points is then interpolated onto a 3D-surface. This surface is stored as a Matlab surface function (`sfit` object) and can be evaluated (using the `feval` function) at any desired point within the covered range (see example below).

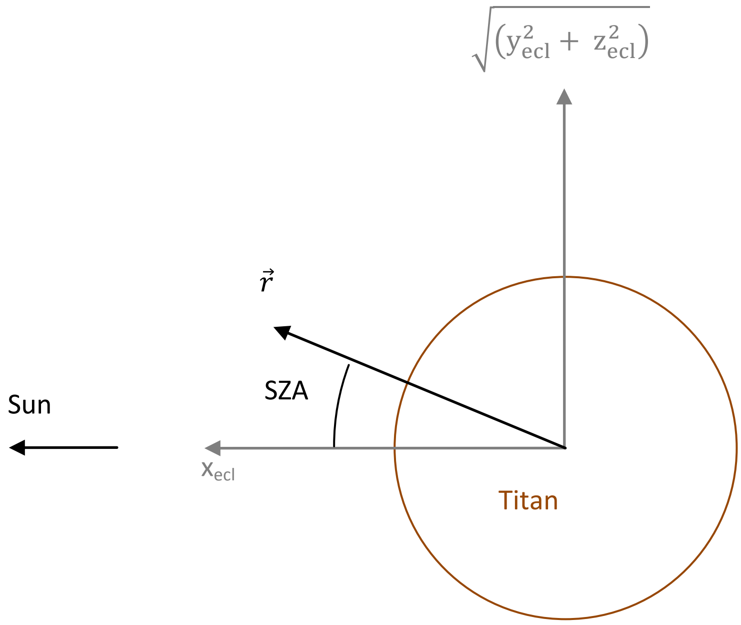

Figure 3.1 Solar Zenith Angle in Titan Solar Ecliptic system. The y and z axis are collapsed into one due to symmetry, and the SZA is counted from x axis to the position vector $\vec{r}$, ranging from 0° (noon, Sun in zenith) to 180° (midnight)

The ion number density altitude distribution

As expected theoretically, the ion population shows typical distribution curves (Figure 3.3) due to an exponential increase in neutral particle density along with an exponential decrease in solar EUV intensity. Figure 3.2 shows the ion number density over altitude and SZA, with contribution from all "deep" flybys of Titan in the time interval October 2004 - April 2010:

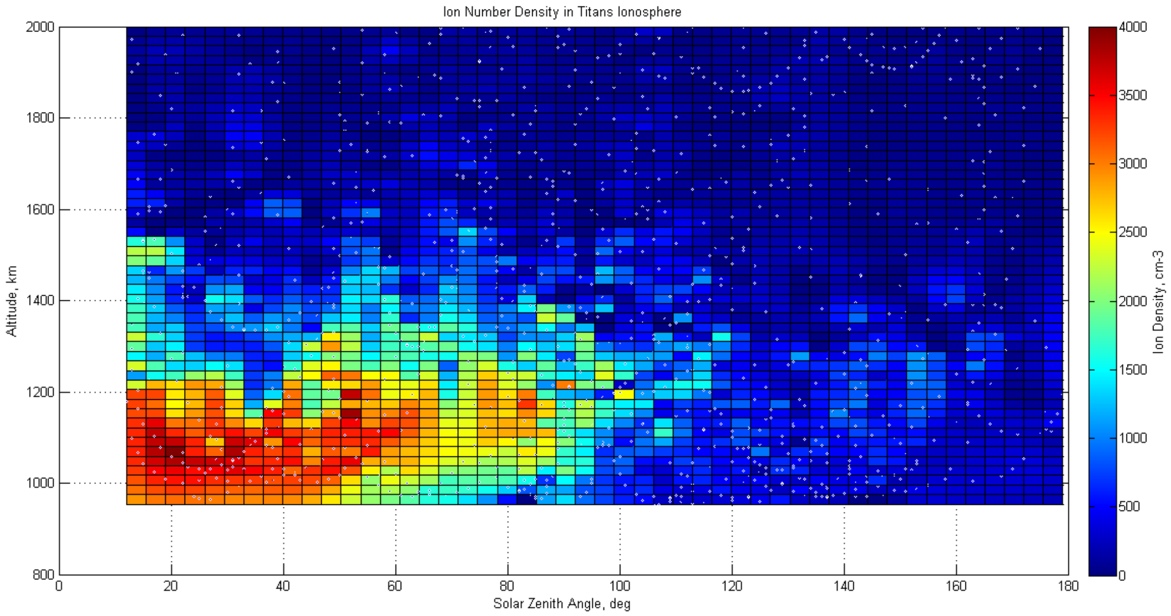

Figure 3.2 The ionosphere of Titan in polar coordinates. The ion density was interpolated over altitude and SZA using LP sweeps along the spacecraft trajectory (data points marked by white dots), and is limited by Cassini's coverage of the moon.

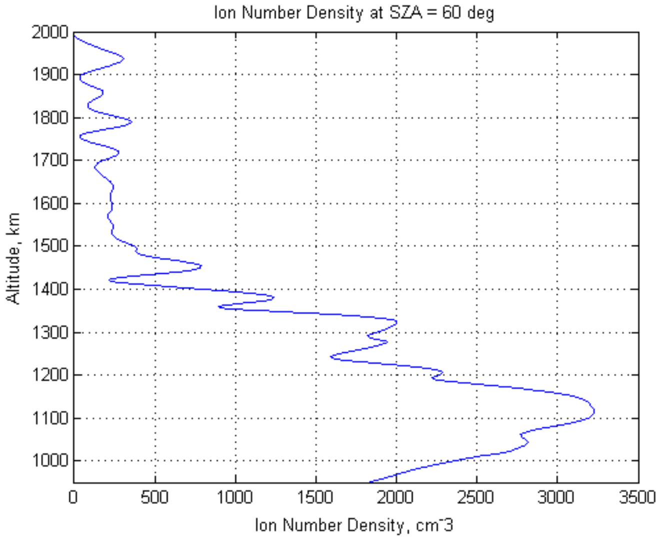

Figure 3.3 Ion Number Density in Titan's ionosphere at 60° SZA.

Figure 3.3 is actually a vertical "slice" of Figure 3.2, cut out from 60° to 60.1° using `feval` function. In this calculation of only the LP measurement data was used and the IND was derived according to Equation (2.3.3). For comparison, Figure 3.4 shows the result of using the more exact Equation (2.3.2) utilizing INMS contributed mean ion mass:

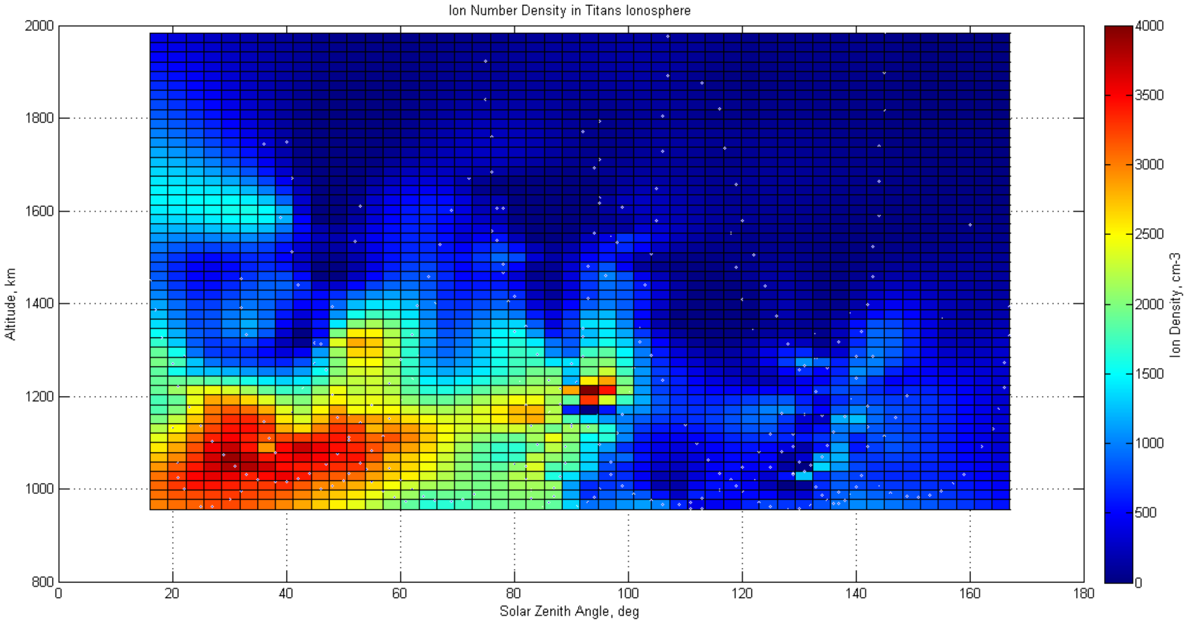

Figure 3.4 Ionosphere of Titan in polar coordinates. Data points (white dots) are limited to flybys where INMS data is available.

This more accurate result suffers from low resolution because it only includes the 14 flybys where the INMS data was available, as well as limitations of the INMS instrument which cannot see ion masses above 100 amu.

The mean ion mass altitude distribution and the evidence of the presence of negative ions

The INMS MIM data can be visualized with the same interpolation method as the IND data. Comparing the INMS MIM data (Figure 3.5) to the MIM calculated from LP measurements (Figure 3.6), it is easy to see a significant difference on the night side of Titan: INMS shows the highest ion mass (around 50 amu) whereas ion mass seen by LP is very low (around 10 amu). Such a big difference on the night side can be explained by the presence of negative ions, also discovered by the ELS instrument (*Coates et al* 2007, *Coates et al* 2009[IX]). See the Discussion section for details.

It should be noted that the reddest spots seen on Figure 3.6 are spikes up to 400 amu, which clearly illustrates that the v_i=v_{sc} assumption does not hold above 1400 km altitude.

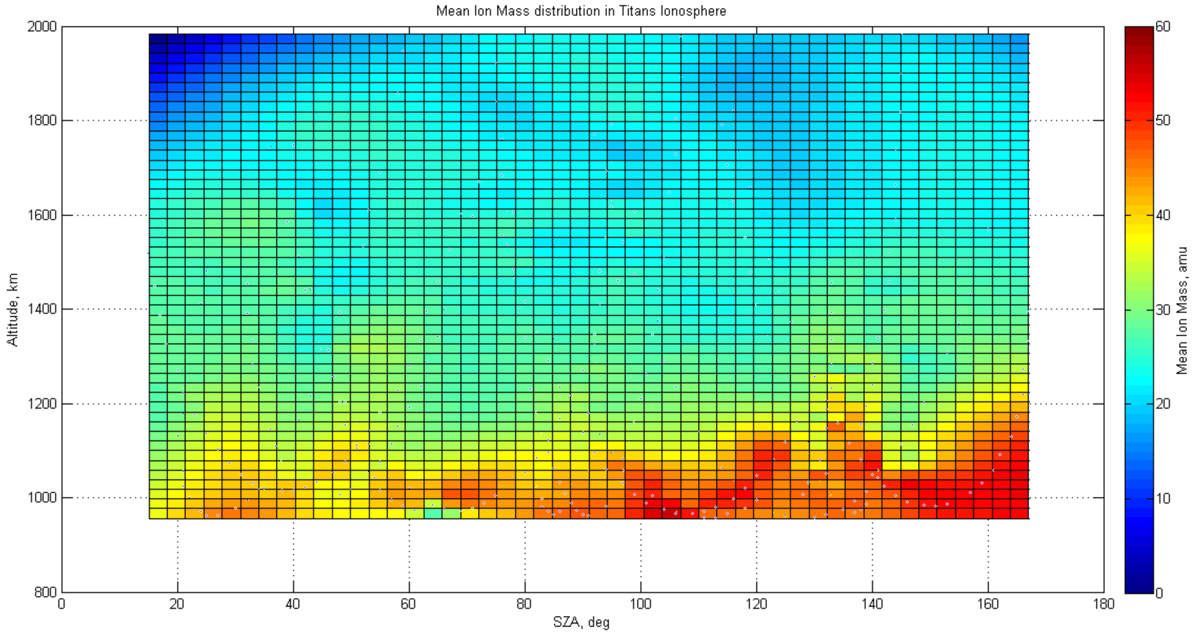

Figure 3.5 INMS Mean Ion Mass data interpolated over altitude and Solar Zenith Angle.

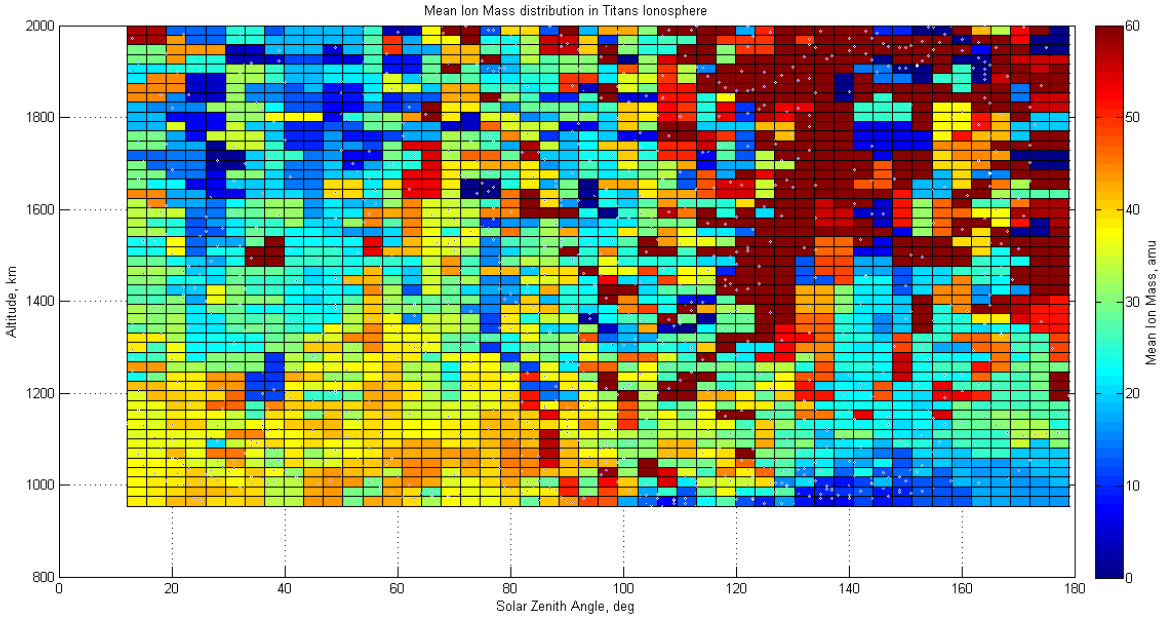

Figure 3.6 Mean Ion Mass calculated from LP data using all flybys included in the analysis, interpolated over altitude and Solar Zenith Angle.

The mentioned difference in mean ion mass is also clearly seen in Figure 3.7 which displays the difference between the s/c speed and the ion speed derived from *b*, the slope of the ion current from Equation (2.2.1).

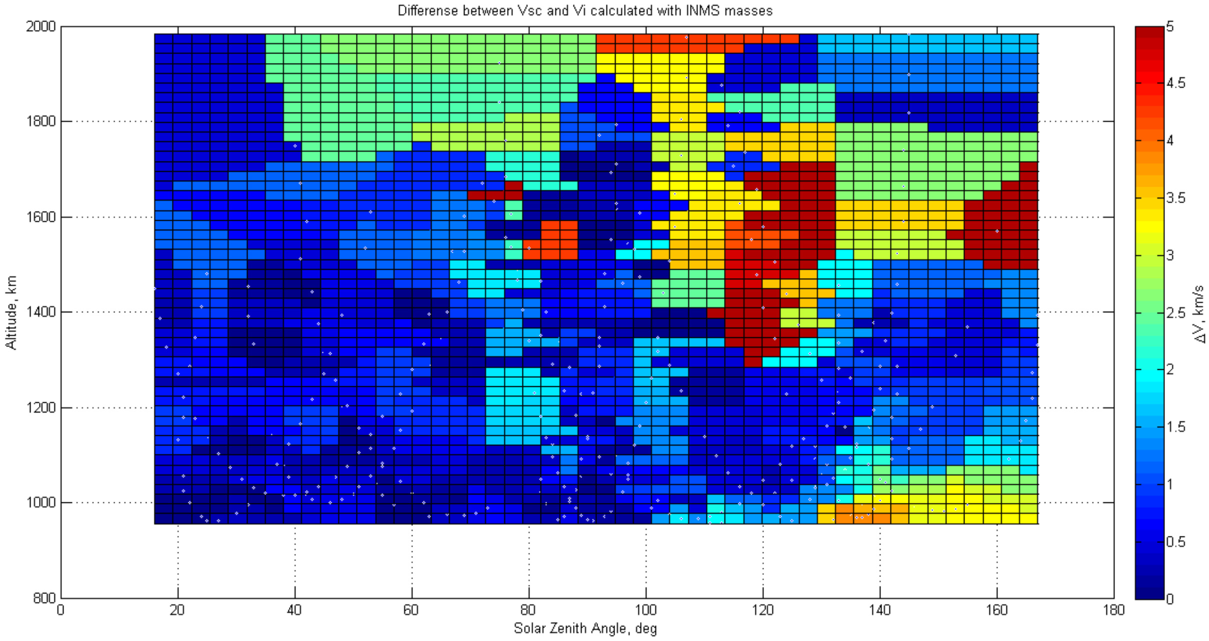

Figure 3.7 Absolute value of the difference between spacecraft speed v_{sc} and ion speed v_i calculated from the LP measurements and the INMS mass data.

4 Discussion

According to *Coates et al* 2009, the negative ions are present across all solar zenith angles and have charge of ≈4e. Those ions give an artifact in the measured ion current at low altitudes, by lifting the ion current curve (left side of Figure 2.1) and thus reducing the ion number density which is linearly proportional to the measured ion current. Accounted for negative ions with masses ≥ 100 amu, the ion current becomes:

$$\begin{cases} a \approx -en_i \pi r_{LP}^2 \left|v_i\right| + q_{neg} n_{neg} \pi r_{LP}^2 \left|v_{neg}\right| \approx -\pi r_{LP}^2 v_{sc} \left(en_i - q_{neg} n_{neg}\right) \\ b \approx \dfrac{2e}{v_{sc}} \pi r_{LP}^2 \left(e \dfrac{n_i}{m_i} + q_{neg} \dfrac{n_{neg}}{m_{neg}} \exp\left(\chi_{neg}\right) \right) \\ en_e = en_i - q_{neg} n_{neg} \end{cases} \tag{2.3.1}$$

where v_{neg}, n_{neg} and m_{neg} are the negative ion speed, number density resp. mass and n_e is the electron number density. Because of the quasineutrality, the ion number density displayed on Figure 3.2 is actually the electron number density, whereas the positive ion number density itself should be somewhat higher.

Even accounted for the negative ion current, the presented IND result is still quite accurate because the negative ions would not affect the ion current derivative (slope) *b*. In addition, the ionosphere is always dominated by the positive ions (comparison of dayside regions in Figure 3.5 and Figure 3.6 suggests that the negative ion number densities on the dayside are negligible, within ≈10% error). Estimation of the behavior of exponential part in the ion current derivative *b* in Equation (2.3.1) gives large masses of negative ions, which means that the entire exponential term holding all negative ion parameters becomes insignificant and thus no information on the negative ions can be acquired. Nevertheless, the effects of their presence are visible in the data.

IND improvements

A first improvement of the calculations would be to get rid of U_{zero} approximations of U_{float} as the latter becomes available for more flybys, ultimately eliminating the need for U_{zero}. The next step would be a more accurate estimation of the maximum photoelectron current, taking into account plasma currents affecting the probe in the magnetosphere of Saturn.

MIM improvements

It is possible to use interpolation and receive more detailed MIM distribution. However, because each flyby sweeps a very specific path in Titans ionosphere, the MIM data would have to be grouped by dividing the flyby trajectories into regions (on SZA basis, for instance). An attempt for such grouping was made, but the groups turned out to have far too few flybys each for a reasonably good approximation. An interpolation without grouping is also an option. The surface function shown in Figure 3.5 can be evaluated to match the data points of LP measurements. This option was turned down due to, again, low resolution of the INMS data set (only 14 flybys). It is impossible to have ion mass data for each flyby included in the project (thus yielding a viable alternative to the method involving ion speed approximation) because the INMS instrument is not operating during each flyby of Titan.

Further studies

The most interesting subject would probably be further investigation of the negative ions in Titan's ionosphere, due to the important role they play in the chemistry of Titan's atmosphere. The suggested improvements of the analysis could most certainly be done within another master thesis project.

It might also be interesting to see the 3D IND altitude distribution in different coordinate systems, for instance, in Titan Solar Ecliptic coordinates without collapsing y and z axis by symmetry, and in Titan Centered Co-rotational coordinates which would reveal ion distributions around the poles and equator.

On the night side of Titan, the atmospheric ionization is no longer dominated by the solar radiation and effects from the impact of Saturn's magnetospheric plasma are could be considerable. Studying those effects could shed more light on the atmospheric chemistry of Titan.

5 References

[I] G. H. Jones, E. Roussos, N. Krupp, U. Beckmann, A. J. Coates, F. Crary, I. Dandouras, V. Dikarev, M. K. Dougherty, P. Garnier, C. J. Hansen, A. R. Hendrix, G. B. Hospodarsky, R. E. Johnson, S. Kempf, K. K. Khurana, S. M. Krimigis, H. Krüger, W. S. Kurth, A. Lagg, H. J. McAndrews, D. G. Mitchell, C. Paranicas, F. Postberg, C. T. Russell, J. Saur, M. Seiß, F. Spahn, R. Srama, D. F. Strobel, R. Tokar, J.-E. Wahlund, R. J. Wilson, J. Woch, D. Young, *The Dust Halo of Saturn's Largest Icy Moon, Rhea*, 2008

[II] http://saturn.jpl.nasa.gov/photos/index.cfm

[III] J. H. Waite Jr., D. T. Young, T. E. Cravens, A. J. Coates, F. J. Crary, B. Magee, J. Westlake, *The Process of Tholin Formation in Titan's Upper Atmosphere*, 2007

[IV] A. J. Coates, F. J. Crary, G. R. Lewis, D. T. Young, J. H. Waite Jr. and E. C. Sittler Jr., *Discovery of heavy negative ions in Titan's ionosphere*, 2007

[V] C.P. McKay, H.D. Smith, *Possibilities for methanogenic life in liquid methane on the surface of Titan*, July 2005

[VI] www.space.irfu.se/cassini/cpr4.jpg

[VII] http://cassini.physics.uiowa.edu/cassini/

[VIII] http://saturn.jpl.nasa.gov/spacecraft/cassiniorbiterinstruments/instrumentscassiniinms/

[IX] A.J.Coates, A.Wellbrock, G.R.Lewis, G.H.Jones, D.T.Young, F.J.Crary, J.H.Waite Jr., *Heavy negative ions in Titan's ionosphere: Altitude and latitude dependence*, 2009

Other related articles:

K. Ågren, J.-E.Wahlund, P.Garnier, R.Modolo, J.Cui, M.Galand, I.Müller-Wodarg, *On the ionospheric structure of Titan*, 2009

J.-E. Wahlund, M.Galand, I.Müller-Wodarg, J.Cui, R.V.Yelle, F.J.Crary, K.Mandt, B.Magee, J.H.WaiteJr, D.T.Young, A.J.Coates, P.Garnier, K. Ågren, M.André, A.I.Eriksson, T.E.Cravens, V. Vuitton, D.A.Gurnett, W.S.Kurth, *On the amount of heavy molecular ions in Titan's ionosphere*, 2009

6 Appendix

This section includes only the Matlab programs developed during this project for calculating the final results presented in the Results section. Short descriptions of the programs are embedded in the code comments (green text). The data required by these programs is received from the voluminous analysis tools for the Langmuir Probe and the modified Matlab routines developed during several other studies. Those routines and the LP analysis tools are not presented here.

Short summary of the presented programs:

- **A1**, `ion_massalt.m` – filters the INMS MIM data, removing off-the-chart values in the upper ionosphere
- **A2**, `IMAD_fit_test.m` – interpolates the filtered INMS MIM data, matching the time vector to the LP measurements
- **A3** `IMAD_exact.m` – calculates the IND and $|v_i-v_{sc}|$, using Equation (2.3.2) and Equation (2.2.1) resp. MIM is imported from the INMS and compared to the MIM calculated from LP measurements using Equation (2.3.1). Produces a data matrix for each result, from the 14 flybys with the INMS data available.
- **A4** `IMAD_proxy.m` – calculates the MIM using Equation (2.3.1) and the IND with two methods for comparison, using Equation (2.3.2) and Equation (2.3.3). Produces a data matrix for each result, all 53 flybys included in the project are used.
- **A5** `Mesh_rcv.m` – a function for extracting the data points with their values and resp. altitude/SZA coordinates, which are used for interpolation and plotting of the results with `sftool`

A1 ion_massalt.m

```
% this code loads INMS mean ion mass data, removes bad values in upper
% ionosphere and saves corrected data in a dat-file
% plot shows corrected data only

clear all;
figure;

grp1 = [18 51];
grp2 = [5 32 59];
grp3 = [21 50 57];
grp4 = [26];
grp5 = [17 40 48];
grp6 = [36 39];
k1 = 1;
k2 = 1;
k3 = 1;
k4 = 1;
k5 = 1;
k6 = 1;
mrkr = [ '+'; 'x'; 'o'];

%----------- ion mass

% set the best looking plot as "expected" and remove off-the-chart values
% in upper ionosphere (>1400km)
proper = [load('t18_inms_mean_mass_04-14-10.dat', 'file')];% load('t50_inms_mean_mass_04-14-
10.dat', 'file')];
err = logical(1 - (proper(:,6) > 1) - isnan(proper(:,6)));
```

```
proper = proper(err,:);
fixp = polyfit(proper(:,2), proper(:,5), 4);
fix = polyval(fixp, 800:0.01:2000);

%subplot(1,2,1);
%hold all
%set(gca, 'ColorOrder', jet);
legendstrm={};
for flynum = 1:65
    if exist(['t' num2str(flynum) '_inms_mean_mass_04-14-10.dat'], 'file') == 2
        temp = load(['t' num2str(flynum) '_inms_mean_mass_04-14-10.dat']);
        % indexing good values only
        indm = temp(:,6) < 1;
        temp = temp(indm,:);


        % this loop off-the-chart values
        indoff = temp(:,2) > 1400 & abs(temp(:,5) - polyval(fixp, temp(:,2))) > 4;
        temp = temp(logical(1-indoff),:);

        % check for remaining NaNs
        if temp == NaN
            disp(['NaN in ' num2str(flynum)]);
        end

        % plotting
        h = plot((temp(:,5)), temp(:,2), '.');
        hold all
        % disp(['plotting mass for ' num2str(flynum)]);
        legendstrm = {legendstrm{:},['T' num2str(flynum)]};

        % sorting markers and colors to 6 groups (chosen manually by Solar Zenit Angle)
        if sum(flynum == grp1) == 1
            set(h, 'Color', [1 0 0]);
            set(h, 'Marker', mrkr(k1));
            k1 = k1+1;
        end
        if sum(flynum == grp2) == 1
            set(h, 'Color', [1 130/255 0]);
            set(h, 'Marker', mrkr(k2));
            k2 = k2+1;
        end
        if sum(flynum == grp3) == 1
            set(h, 'Color', [0.9 0.9 0]);
            set(h, 'Marker', mrkr(k3));
            k3 = k3+1;
        end
        if sum(flynum == grp4) == 1
            set(h, 'Color', [0 190/255 0]);
            set(h, 'Marker', mrkr(k4));
            k4 = k4+1;
        end
        if sum(flynum == grp5) == 1
            set(h, 'Color', [0 170/255 1]);
            set(h, 'Marker', mrkr(k5));
            k5 = k5+1;
        end
        if sum(flynum == grp6) == 1
            set(h, 'Color', [120/255 0 1]);
            set(h, 'Marker', mrkr(k6));
            k6 = k6+1;
        end

        % save corrected data in a dat-file
        dlmwrite(['T' num2str(flynum) '_INMS_MIM.dat'], temp, '\t');
    end
end
ylabel('altitude, {km}'); %xlim([20 55]);
xlabel('mean ion mass, {amu}');
xlim([15 55]); ylim([900 2000]);
legend(legendstrm);
%plot(fix, 800:0.01:2000, '-k', 'LineWidth',3); hold all
hold off


%---------- ion density
% %subplot(1,2,2);
% %hold all
% %set(gca, 'ColorOrder', jet);
% figure;
% legendstri={};
% for flynum = 1:65
%     if exist(['t' num2str(flynum) '_inms_mean_mass_04-14-10.dat'], 'file') == 2
%         temp = load(['t' num2str(flynum) '_inms_mean_mass_04-14-10.dat']);
%         % indexing good values only
%         indm = temp(:,6) < 1;
%
%         % plotting
%         h = plot(temp(:,3), temp(:,2));
%         hold all
%         %disp(['plotting ion density for ' num2str(flynum)]);
%         legendstri = {legendstri{:},['T' num2str(flynum)]};
%
%         % sorting markers to 6 groups (chosen by Titan Local Time)
%         if sum(flynum == grp1) == 1
%             set(h, 'Color', [1 0 0]);
%         end
%         if sum(flynum == grp2) == 1
%             set(h, 'Color', [1 130/255 0]);
%         end
%         if sum(flynum == grp3) == 1
%             set(h, 'Color', [0.9 0.9 0]);
%         end
%         if sum(flynum == grp4) == 1
%             set(h, 'Color', [0 190/255 0]);
%         end
%         if sum(flynum == grp5) == 1
%             set(h, 'Color', [0 170/255 1]);
%         end
%         if sum(flynum == grp6) == 1
%             set(h, 'Color', [120/255 0 1]);
%         end
%     end
% end
% ylabel('altitude, {km}'); %xlim([20 55]);
% xlabel('ion mass, {amu}');
% ylim([900 2000]);
% legend(legendstri);
% hold off
%
% dlmwrite('MAD_Ion.dat', [alt mfit], '\t');
% plot(mm, alt, '*', mfit, alt); legend('alt', 'alt pol');
```

A2 *IMAD_fit_test.m*

```
% load mean ion mass dat-files (run ion_massalt.m)
% fits timevector of INMS to timevector of LP
% interpolates MIM to match new timeline
% saves result in T#_MIMfit.dat
% plots MIM-altitude distribution over solar zenith angle (MIM is
% colorcoded)

clear all
close all

path(path, '/home/olch/titaneph'); % cassini ephemerides here
path(path, '/home/olch/INMS'); % cassini INMS data here

MIM_mesh = nan(1201,181); % empty matrix for color coding

%flynum = input('input fly-by number: ');
for flynum=1:65
if exist(['T' num2str(flynum) '_INMS_MIM.dat'], 'file') == 2


T_INMS = load(['T' num2str(flynum) '_INMS_MIM.dat']);
T_EPH = load(['T' num2str(flynum) '_Ecl.dat']);

T_EPH(:,10) = T_EPH(:,10) -2575; % radial dist to altitude
```

```
T_EPH = [toepoch(T_EPH(:,1:6)) T_EPH(:,7:14)]; % [time x y z alt vx vy vz v]


[~, indmin] = min(T_EPH(:,5)); % approximately closest approach
timecon = T_EPH(indmin,1); % normalization constant, needed for a good polyfit
T_EPH(:,1) = T_EPH(:,1) - timecon;


P = polyfit(T_INMS(:,1), T_INMS(:,2), 8);

% find closest approach
Pmin = polyfit(T_EPH(:,1),T_EPH(:,5),8);
altfun = @(x) Pmin(1)*x^8 + Pmin(2)*x^7 + Pmin(3)*x^6 + Pmin(4)*x^5 + Pmin(5)*x^4 +
Pmin(6)*x^3 + Pmin(7)*x^2 + Pmin(8)*x + Pmin(9); % polynomial P
timemin = fminbnd(altfun,-100,100);

T_EPH(:,1) = T_EPH(:,1) - timemin; % shift normalized time scale so that 0 corresponds to
closest approach
timecon = timecon + timemin; % change normalization constant to match closest approach

T_EPH = T_EPH(T_EPH(:,1)>=T_INMS(1,1) & T_EPH(:,1)<=T_INMS(end,1) ,:); % limit to INMS data
range



MIM = interp1(T_INMS(:,1),T_INMS(:,5),T_EPH(:,1), 'cubic');
MIM(MIM<0) = NaN;
%MIM = roundn(MIM,-3);

% figure(flynum);
% % this shows MIM interpolates vs original values
% plot(MIM, T_EPH(:,1), '-', T_INMS(:,5), T_INMS(:,1), ':');


T_INMS = [T_EPH(:,1)+timecon, MIM, polyval(P,T_EPH(:,1)), T_EPH(:,2), T_EPH(:,3), T_EPH(:,4)];
% OBS!data order changed [ephepoch massfit altfit x y z]
T_EPH(:,1) = T_EPH(:,1)+timecon;
save(['T' num2str(flynum) '_MIMfit'], 'T_INMS', 'T_EPH', '-v7.3');

% SZA = Solar Zenith Angle, 0 pointing to Sun
SZA = 90-(180/pi*atan(T_EPH(:,2)./sqrt(T_EPH(:,3).^2 + T_EPH(:,4).^2))); % arctan( x /
sqrt(y^2 + z^2) )


% color corresponds to mean ion mass
for i = 1:length(SZA)
    for j = 1:length(T_INMS(:,3))
        if i == j
            MIM_mesh(round(T_INMS(j,3))-799, round(SZA(i))+1) = MIM(i);
            % following makes data points into triplets for better visual,
            % comment before saving the output mesh matrix
%             MIM_mesh(round(T_INMS(j,3))-799+1, round(SZA(i))+1) = MIM(i);
%             MIM_mesh(round(T_INMS(j,3))-799-1, round(SZA(i))+1) = MIM(i);
        end
    end
end

end % if exist...

end % for flynum =...


set(0, 'DefaultTextFontName', 'Courier New');

if 1
figure;
% limiting axes, -90 is midnight, 90 is noon; 800-2000 is altitude
axis([0 180 800 2000]);
hp = pcolor(0:180, 800:2000, MIM_mesh);
% get(hp)
% set(hp,'LineWidth',5);

  % Set color-limits
  c_min = 0;
  c_max = 60;
  c_limit = [c_min c_max];
  set(gca, 'CLim', c_limit);

  % scaler at side
  colormap('jet');
  h_bar = colorbar;
  scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
  set(h_bar, 'CLim', c_limit);
  set(get(h_bar,'ylabel'),'String', 'Mean Ion Mass, {amu}');

title('Mean Ion Mass over Altitude Distribution vs Solar ZenithAngle. Gray line is the
terminator');
xlabel('Solar Zenith Angle, 0 is Sun-direction');
ylabel('altitude, {km}');

shading flat;
hold on

% plot light limit (Titan's shadow)
lightx = 0:-1:-6000;
plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9 0.9],
'LineWidth', 1);
hold off
end % if switch
```

A3 IMAD_exact.m

```
% Ion Mass-Altitude Distribution Calc
% only flybies for which INMS data exists here
% INMS mass data is used for ion number density Ni

clear all
%close all
% constants
r_lp = 2.5; % LP probe radius, cm
e = 1.602176487e-19; % electron charge, C
Rt = 2575; % radius of Titan, km
amu = 1.660538782e-27; % atomic mass, kg

% ion current:
% a = -e * Ni * (4*pir_lp^2) * Vi / 4
%   a = -e * Ni * pi * r_lp^2 * Vi
% ion current derivative wrt U bias
% b = -e * Ni * (4*pir_lp^2) * (2*e/4/mim) / Vi
%   b = -e  * Ni * pi * r_lp^2 * 2 * e / mim / Vi

% file paths
path(path, '../titaneph'); % cassini ephemerides for titan flybys here
path(path, '../Titan_Ion'); % cassini LP data here

IMAD_mesh = nan(1201,181); % empty matrix for color coding
IMAD_mesh_prx = nan(1201,181); % empty matrix for color coding
MIM_mesh_prx = nan(1201,181); % empty matrix for color coding
Verr_mesh = nan(1201,181); % empty matrix for color coding

% define fly-by
%flynum = input('enter fly-by number (T#): ');

% figure;

for flynum = 1:65

    if exist(['T' num2str(flynum) '_MIMfit.mat'], 'file') == 2

    disp(['processing T' num2str(flynum)]);


    % loading data
    load(['T' num2str(flynum) '_final.mat']); % contents: a, a0, a_err, b
    % ephdata: [yyyy mm dd hh mm ss x y z r vx vy vz v] coord in km, velocity in km/s

    % if a is NaN (Ufloat not available) set a to proxy value a0 (using zero
    % crossing potential)
    if sum(isnan(a)) == length(a)
        a = a0;
        disp('Warning: ion current approximation, U0 ~ -Ufloat');
    end



    load(['T' num2str(flynum) '_MIMfit.mat']);
    % format: T_INMS = [ephepoch massfit altfit x y z] T_EPH = [ephepoch x y z alt vx vy vz v]

    eph_temp = [];
    a_temp =  [];
    b_temp =  [];
    a0_temp =  [];
    a_err_temp =  [];

    % find ephemerides and current values matching INMS MIM_fit data (by time/alt)
    for i = 1:length(T_INMS(:,1))
        indeph = toepoch(ephdata(:,1:6)) == T_INMS(i,1);
        a_temp = [a_temp; a(indeph)];
        a0_temp = [a0_temp; a0(indeph)];
        a_err_temp = [a_err_temp; a_err(indeph)];
        b_temp = [b_temp; b(indeph)];

    end

    a = a_temp;
    b = b_temp;
    a0 = a0_temp;
    a_err = a_err_temp;


    % assign variables
    t = T_INMS(:,1);
    mim = T_INMS(:,2);
    alt = T_INMS(:,3);

    % SZA = Solar Zenit Angle, 0 pointing to Sun
%    SZA = 90-(180/pi*atan(T_INMS(:,4)./sqrt(T_INMS(:,5).^2 + T_INMS(:,6).^2))); % arctan( x /
sqrt(y^2 + z^2) )
    SZA = 90-(180/pi*atan(T_EPH(:,2)./sqrt(T_EPH(:,3).^2 + T_EPH(:,4).^2))); % arctan( x /
sqrt(y^2 + z^2) )


    % fix bad values of a (can't be positive)
    inda = a > 0;
    a(inda) = NaN;
    inda0 = a0 > 0;
    a0(inda0) = NaN;


    % ion density derived from a*b
        Ni = sqrt(a.*b.*mim*amu/2/e)/e/pi/r_lp^2 *1e-11; % 1e-2 is m^-1 to cm^-1 + 1e-9 is nA
to A
    % ion speed derived from a/b
        Vi = sqrt(2*e.*a./b./mim/amu)/1000; % km/s

    % same using Uzero
        Ni0 = sqrt(a0.*b.*mim*amu/2/e)/e/pi/r_lp^2 *1e-11;
        Vi0 = sqrt(2*e.*a0./b./mim/amu)/1000; % km/s

    % approximate Ni, set Vi = Vsc
        Niprx = -a./T_EPH(:,9)/pi/r_lp^2/e *1e-14; % 1e-5 is km^-1 to cm^-1 + 1e-9 is nA to A
        Niprx0 = -a0./T_EPH(:,9)/pi/r_lp^2/e *1e-14;
        mimprx = 2*e*a./b./T_EPH(:,9).^2 /amu *1e-6;
        if sum(mimprx>300) >= 1
            disp('VERY LARGE mimprx')
        else
            if sum(mimprx>100) >= 1
            disp('LARGE mimprx')
            end
        end

        % when activating don't forget figure before loop and hold off after
%         subplot(311); plot(T_EPH(:,5),[mim mimprx])
%         hold all
%         subplot(312); plot(T_EPH(:,5),[Ni, Ni0])
%         hold all
%         subplot(313); plot(T_EPH(:,5),[Niprx, Niprx0])
%         hold all

        Ni_err = sum(Ni == Niprx);
        mimprx(mimprx>60) = 60;
        % mesh matrix for mimprx
    for i = 1:length(SZA)
        for j = 1:length(T_INMS(:,3))
            if i == j
                MIM_mesh_prx(round(T_INMS(j,3))-799, round(SZA(i))+1) =  mimprx(i);
                % following is purely for cosmetic purposes
%                 MIM_mesh_prx(round(T_INMS(j,3))-799+1, round(SZA(i))+1) = mimprx(i);
%                 MIM_mesh_prx(round(T_INMS(j,3))-799-1, round(SZA(i))+1) = mimprx(i);
            end
        end
    end

        % mesh matrix for Ni from a*b
    for i = 1:length(SZA)
        for j = 1:length(T_INMS(:,3))
            if i == j
                IMAD_mesh(round(T_INMS(j,3))-799, round(SZA(i))+1) =  Ni(i);
                % following is purely for cosmetic purposes
%                 IMAD_mesh(round(T_INMS(j,3))-799+1, round(SZA(i))+1) = Ni(i);
%                 IMAD_mesh(round(T_INMS(j,3))-799-1, round(SZA(i))+1) = Ni(i);
            end
        end
    end

        % mesh matrix for Ni from Vi=Vsc proxy
    for i = 1:length(SZA)
        for j = 1:length(T_INMS(:,3))
            if i == j
                IMAD_mesh_prx(round(T_INMS(j,3))-799, round(SZA(i))+1) = Niprx(i);
                % following is purely for cosmetic purposes
%                 IMAD_mesh_prx(round(T_INMS(j,3))-799+1, round(SZA(i))+1) = Niprx(i);
%                 IMAD_mesh_prx(round(T_INMS(j,3))-799-1, round(SZA(i))+1) = Niprx(i);
            end
        end
    end
        % mesh matrix for Vsc - Vi
    for i = 1:length(SZA)
        for j = 1:length(T_INMS(:,3))
            if i == j
                Verr_mesh(round(T_INMS(j,3))-799, round(SZA(i))+1) = abs(T_EPH(i,9) - Vi(i));
                % following is purely for cosmetic purposes
%                 Verr_mesh(round(T_INMS(j,3))-799+1, round(SZA(i))+1) = T_EPH(i,9) - Vi(i);
%                 Verr_mesh(round(T_INMS(j,3))-799-1, round(SZA(i))+1) = T_EPH(i,9) - Vi(i);
            end
        end
    end

    end % if exist...
end % for flynum...

%  hold off

set(0, 'DefaultTextFontName', 'Courier New');

if 0
%------------------------- plot for mimprx
    figure;
    % limiting axes, 180 is midnight, 0 is noon; 800-2000 is altitude
    axis([0 180 800 2000]);
    pcolor(0:180, 800:2000, MIM_mesh_prx);

      % Set color-limits
      c_min = 0;
      c_max = 60;
      c_limit = [c_min c_max];
      set(gca, 'CLim', c_limit);

      % scaler at side
      colormap('jet');
      h_bar = colorbar;
      scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
      set(h_bar, 'CLim', c_limit);
      set(get(h_bar,'ylabel'),'String', 'Ion Density, {cm-3}');

    title('Vi = Vsc method: Mean Ion Mass over Altitude Distribution vs Solar Zenit Angle.
Gray line is the terminator');
    xlabel('Solar Zenit Angle, 0 is Sun-direction');
    ylabel('altitude, {km}');

    shading flat;
    hold on

    % plot light limit (Titan's shadow)
    lightx = 0:-1:-6000;
    plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9
0.9], 'LineWidth', 1);
    hold off
%--------------------------------------------------------

%------------------------- plot for Vsc-Vi
    figure;
    % limiting axes, 180 is midnight, 0 is noon; 800-2000 is altitude
    axis([0 180 800 2000]);
    pcolor(0:180, 800:2000, Verr_mesh);

      % Set color-limits
      c_min = -10;
      c_max = 10;
      c_limit = [c_min c_max];
      set(gca, 'CLim', c_limit);

      % scaler at side
      colormap('jet');
      h_bar = colorbar;
      scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
      set(h_bar, 'CLim', c_limit);
      set(get(h_bar,'ylabel'),'String', 'Ion Density, {cm-3}');

    title('Vsc - Vi Altitude Distribution vs Solar Zenit Angle. Gray line is the terminator');
    xlabel('Solar Zenit Angle, 0 is Sun-direction');
    ylabel('altitude, {km}');

    shading flat;
    hold on

    % plot light limit (Titan's shadow)
    lightx = 0:-1:-6000;
    plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9
0.9], 'LineWidth', 1);
    hold off
%--------------------------------------------------------

%------------------------- plot for for Ni from a*b
    figure;
    % limiting axes, 180 is midnight, 0 is noon; 800-2000 is altitude
    axis([0 180 800 2000]);
    pcolor(0:180, 800:2000, IMAD_mesh);

      % Set color-limits
      c_min = 0;
      c_max = 4000;
      c_limit = [c_min c_max];
      set(gca, 'CLim', c_limit);

      % scaler at side
      colormap('jet');
      h_bar = colorbar;
      scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
      set(h_bar, 'CLim', c_limit);
      set(get(h_bar,'ylabel'),'String', 'Ion Density, {cm-3}');

    title('a*b method: Ion Number Density over Altitude Distribution vs Solar Zenit Angle.
Gray line is the terminator');
    xlabel('Solar Zenit Angle, 0 is Sun-direction');
    ylabel('altitude, {km}');

    shading flat;
    hold on
    % plot light limit (Titan's shadow)
    lightx = 0:-1:-6000;
    plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9
0.9], 'LineWidth', 1);
    hold off
%--------------------------------------------------------


%------------------------- plot for Ni from Vi=Vsc proxy
    figure;
    % limiting axes, 180 is midnight, 0 is noon; 800-2000 is altitude
    axis([0 180 800 2000]);
    pcolor(0:180, 800:2000, IMAD_mesh_prx);

      % Set color-limits
      c_min = 0;
      c_max = 4000;
      c_limit = [c_min c_max];
      set(gca, 'CLim', c_limit);

      % scaler at side
      colormap('jet');
      h_bar1 = colorbar;
      scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
      set(h_bar1, 'CLim', c_limit);
      set(get(h_bar1,'ylabel'),'String', 'Ion Density, {cm-3}');

    title('Vi=Vsc method: Ion Number Density over Altitude Distribution vs Solar Zenit Angle.
Gray line is the terminator');
    xlabel('Solar Zenit Angle, 0 is Sun-direction');
    ylabel('altitude, {km}');

    shading flat;
    hold on

    % plot light limit (Titan's shadow)
    lightx = 0:-1:-6000;
    plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9
0.9], 'LineWidth', 1);
    hold off
% -------------------------------------------------------------
end % if switch
```

A4 *IMAD_proxy.m*

```
% Ion Mass-Altitude Distribution Calc
% all fly-bies, LP measurements only

clear all
close all
% constants
r_lp = 2.5; % LP probe radius, cm
e = 1.602176487e-19; % electron charge, C
Rt = 2575; % radius of Titan, km
amu = 1.660538782e-27; % atomic mass, kg

% ion current:
% a = -e * Ni * (4*pir_lp^2) * Vi / 4
%   a = -e * Ni * pi * r_lp^2 * Vi
% ion current derivative wrt U bias
% b = -e * Ni * (4*pir_lp^2) * (2*e/4/mimprx) / Vi
%   b = -e  * Ni * pi * r_lp^2 * 2 * e / mimprx / Vi

% file paths
path(path, '../titaneph'); % cassini ephemerides for titan flybys here
path(path, '../Titan_Ion'); % cassini LP data here

IMAD_mesh = nan(1201,181); % empty matrix for color coding
IMAD_mesh_prx = nan(1201,181); % empty matrix for color coding
MIM_mesh_prx = nan(1201,181); % empty matrix for color coding
```

```
% define fly-by
%flynum = input('enter fly-by number (T#): ');

% figure;

for flynum = 1:65

    if exist(['T' num2str(flynum) '_final.mat'], 'file') == 2

    disp(['processing T' num2str(flynum)]);


    % loading data
    load(['T' num2str(flynum) '_final.mat']); % contents: a, a0, a_err, b
    % ephdata: [yyyy mm dd hh mm ss x y z r vx vy vz v] coord in km, velocity in km/s

    % if a is NaN (Ufloat not available) set a to proxy value a0 (using zero
    % crossing potential)
    if sum(isnan(a)) == length(a)
        a = a0;
        disp('Warning: ion current approximation, U0 ~ -Ufloat');
    end

    T_EPH = [nan(length(ephdata(:,1)),1), ephdata(:,7:9), ephdata(:,10)-Rt, ephdata(:,11:14)];
% new format: [epoch x y z alt vx vy vz v]
    indeph = T_EPH(:,5) < 2000; % limit to max 2000km altitude
    T_EPH = T_EPH(indeph,:);

    a = a(indeph);
    b = b(indeph);
    a0 = a0(indeph);
    a_err = a_err(indeph);


    % assign variables
%    t = T_EPH(:,1);
    alt = T_EPH(:,5);
    Vsc = T_EPH(:,9);

    % SZA = Solar Zenit Angle, 0 pointing to Sun
    SZA = 90-(180/pi*atan(T_EPH(:,2)./sqrt(T_EPH(:,3).^2 + T_EPH(:,4).^2))); % arctan( x /
sqrt(y^2 + z^2) )


    % fix bad values of a (can't be positive)
    a(a>0) = NaN;
    a0(a0>0) = NaN;

    % mean ion mass in amu:
    mimprx = a./b./Vsc.^2*2*e/amu *1e-6; % 1e-6 is from division by km^2/s^2 (converted toto
m^2/s^2)

    % ion density derived from a*b
        Ni = sqrt(a.*b.*mimprx*amu/2/e)/e/pi/r_lp^2 *1e-11; % 1e-2 is m^-1 to cm^-1 + 1e-9 is
nA to A
    % ion speed derived from a/b
        Vi = sqrt(2*e.*a./b./mimprx/amu/1000); % km/s

    % same using Uzero
        Ni0 = sqrt(a0.*b.*mimprx*amu/2/e)/e/pi/r_lp^2 *1e-11;
        Vi0 = sqrt(2*e.*a0./b./mimprx/amu/1000); % km/s


        mimprx(mimprx>60) = 60; % limiting mass for plotting

    % approximate Ni, set Vi = Vsc
        Niprx = -a./Vsc/pi/r_lp^2/e *1e-14; % 1e-5 is km^-1 to cm^-1 + 1e-9 is nA to A
        Niprx0 = -a0./Vsc/pi/r_lp^2/e *1e-14;

%         % when activating don't forget figure before loop and hold off after
%         subplot(311); plot(T_EPH(:,5),mimprx)
%         hold all
%         subplot(312); plot(T_EPH(:,5),[Ni, Ni0])
%         hold all
%         subplot(313); plot(T_EPH(:,5),[Niprx, Niprx0])
%         hold all
```

```
        % mesh matrix for MIM
    for i = 1:length(SZA)
        for j = 1:length(T_EPH(:,5))
            if i == j
                MIM_mesh_prx(round(T_EPH(j,5))-799, round(SZA(i))+1) = mimprx(i);
            % following makes data points into triplets for better visual,
            % comment before saving the output mesh matrix
%                 MIM_mesh_prx(round(T_EPH(j,5))-799+1, round(SZA(i))+1) = mimprx(i);
%                 MIM_mesh_prx(round(T_EPH(j,5))-799-1, round(SZA(i))+1) = mimprx(i);
            end
        end
    end


        % mesh matrix for Ni from a*b
    for i = 1:length(SZA)
        for j = 1:length(T_EPH(:,5))
            if i == j
                IMAD_mesh(round(T_EPH(j,5))-799, round(SZA(i))+1) = Ni(i);
            % following makes data points into triplets for better visual,
            % comment before saving the output mesh matrix
%                 IMAD_mesh(round(T_EPH(j,5))-799+1, round(SZA(i))+1) = Ni(i);
%                 IMAD_mesh(round(T_EPH(j,5))-799-1, round(SZA(i))+1) = Ni(i);
            end
        end
    end

        % mesh matrix for Ni from Vi=Vsc proxy
    for i = 1:length(SZA)
        for j = 1:length(T_EPH(:,5))
            if i == j
                IMAD_mesh_prx(round(T_EPH(j,5))-799, round(SZA(i))+1) = Niprx(i);
            % following makes data points into triplets for better visual,
            % comment before saving the output mesh matrix
%                 IMAD_mesh_prx(round(T_EPH(j,5))-799+1, round(SZA(i))+1) = Niprx(i);
%                 IMAD_mesh_prx(round(T_EPH(j,5))-799-1, round(SZA(i))+1) = Niprx(i);
            end
        end
    end


    end % if exist...
end % for flynum...

% hold off

set(0, 'DefaultTextFontName', 'Courier New');



if 1

%------------------------- plot for Ni from a*b
    figure;
    % limiting axes, -90 is midnight, 90 is noon; 800-2000 is altitude
    axis([0 180 800 2000]);
    hp = pcolor(0:180, 800:2000, MIM_mesh_prx);
    % get(hp)
    % set(hp,'LineWidth',5);

      % Set color-limits
      c_min = 0;
      c_max = 60;
      c_limit = [c_min c_max];
      set(gca, 'CLim', c_limit);

      % scaler at side
      colormap('jet');
      h_bar = colorbar;
      scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
      set(h_bar, 'CLim', c_limit);
      set(get(h_bar,'ylabel'),'String', 'Mean Ion Mass, {amu}');

    title('Vi=Vsc: Mean Ion Mass over Altitude Distribution vs Solar Zenit Angle. Gray line is 
the terminator');
    xlabel('Solar Zenit Angle, 0 is Sun-direction');
    ylabel('altitude, {km}');
    shading flat;
    hold on

    % plot light limit (Titan's shadow)
    lightx = 0:-1:-6000;
    plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9
0.9], 'LineWidth', 1);
    hold off
%--------------------------------------------------------------




%------------------------- plot for Ni from a*b
    figure;
    % limiting axes, 180 is midnight, 0 is noon; 800-2000 is altitude
    axis([0 180 800 2000]);
    pcolor(0:180, 800:2000, IMAD_mesh);

      % Set color-limits
      c_min = 0;
      c_max = 5000;
      c_limit = [c_min c_max];
      set(gca, 'CLim', c_limit);

      % scaler at side
      colormap('jet');
      h_bar = colorbar;
      scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
      set(h_bar, 'CLim', c_limit);
      set(get(h_bar,'ylabel'),'String', 'Ion Density, {cm-3}');

    title('a*b method: Ion Density over Altitude Distribution vs Solar Zenit Angle. Gray line
is the terminator');
    xlabel('Solar Zenit Angle, 0 is Sun-direction');
    ylabel('altitude, {km}');

    shading flat;
    hold on

    % plot light limit (Titan's shadow)
    lightx = 0:-1:-6000;
    plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9
0.9], 'LineWidth', 1);
    hold off
%--------------------------------------------------------------


%------------------------- plot for Ni from Vi=Vsc proxy
    figure;
    % limiting axes, 180 is midnight, 0 is noon; 800-2000 is altitude
    axis([0 180 800 2000]);
    pcolor(0:180, 800:2000, IMAD_mesh_prx);

      % Set color-limits
      c_min = 0;
      c_max = 5000;
      c_limit = [c_min c_max];
      set(gca, 'CLim', c_limit);

      % scaler at side
      colormap('jet');
      h_bar1 = colorbar;
      scale_data = linspace( c_limit(1), c_limit(2), length( colormap ));
      set(h_bar1, 'CLim', c_limit);
      set(get(h_bar1,'ylabel'),'String', 'Ion Density, {cm-3}');

    title('Vi=Vsc method: Ion Density over Altitude Distribution vs Solar Zenit Angle. Gray
line is the terminator');
    xlabel('Solar Zenit Angle, 0 is Sun-direction');
    ylabel('altitude, {km}');

    shading flat;
    hold on

    % plot light limit (Titan's shadow)
    lightx = 0:-1:-6000;
    plot((90-atan(lightx./2575)*180/pi), sqrt((lightx).^2 + 2575^2)-2575, 'Color', [0.9 0.9
0.9], 'LineWidth', 1);
    hold off
%-------------------------------------------------------------
end % if switch
```

A5 Mesh_rcv.m

```
% function to extract row and column indexes and values from a mesh-matrix
% mesh-matrix contains NaNs inhomogenously distributed, which are ignored
function [row col val] = Mesh_rcv(mesh)
mesh1 = mesh;
mesh1(logical(isnan(mesh))) = 0; % set NaN to 0 in order to use find
[row, col, val] = find(mesh1); % indexes and values of non-zero entries
row = row + 800;
```