UPTEC F05 063

Examensarbete 20 p
4 juli 2005

The first Rosetta Earth flyby

Trajectory, attitude and radiation information for LAP operations

---

Magnus Billvik

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

The first Rosetta Earth flyby

Magnus Billvik

The Rosetta spacecraft, launched by the European Space Agency in March 2004, performed an Earth flyby in March 2005. Before reaching its final destination, the comet 67P/Churyumov-Gerasimenko, it will perform another two Earth flybys and one Mars flyby. During the planetary flybys the science instruments on board Rosetta can be calibrated and tested as well as providing new scientific results on the planetary environments. For planning the LAP (Langmuir Probe) operations before the first Earth flyby and to help analyzing the measured data, the LAP team at the Swedish Institute of Space Physics, Uppsala Division (IRF-U) needed background information about the flyby in terms of spacecraft trajectory, radiation doses on the electronics and information on how the spacecraft attitude affects the LAP environment. For this purpose a Matlab program was written with routines for reading, treating and visualizing the trajectory and attitude data. After transforming the Rosetta trajectory to geographical coordinates we could use the SPace ENVironment Information System (SPENVIS) to estimate the amount of radiation from the Van Allen radiation belts. The total expected radiation dose was estimated to be less than 100 Rad, meaning no risk of damaging the electronics, and this resulted in the decision to keep the instrument switched on during the flyby. Matlab routines were also developed to estimate when the two Langmuir probes were likely to be sunlit and when in eclipse behind the spacecraft body or any of its bigger instruments. The flyby was successful and analyzing the post-flyby LAP data, we could explain well the variations in LAP photocurrent from our estimations. This result verified that the Matlab routines work well and that they can be used, and further improved, for the three future planetary flybys.

Handledare: Anders Eriksson
Ämnesgranskare: Mats André
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F05 063
Tryckt av: IRF-U

Contents

1 Introduction

In 1993 the European Space Agency (ESA) approved the International Rosetta Mission as a Cornerstone Mission in ESA's *Horizons 2000 Science Programme*. Since then, scientists and engineers from all over Europe and the United States have been working within the Rosetta project. The goal of this unique space expedition is to gain knowledge about the content and origin of comets in our Solar System, by putting a lander on the surface of a comet. For this purpose, a number of scientific instruments are mounted on board the spacecraft for measuring physical properties of the space environment. One of these instruments is the dual Langmuir probe (LAP), and this thesis concentrates on the LAP instrument during Rosetta's first so-called Earth flyby.

This chapter will explain the background of this thesis. Section 1.1 gives an overview of the Rosetta Project together with the objectives of this thesis. Section 1.2 describes the Rosetta spacecraft and its equipment. In Section 1.3 the space environment around the Earth is briefly summarized, followed by some theory about the LAP instrument in Section 1.4.

1.1 The Rosetta Project - an overview

Comets are the oldest objects around the Sun and can therefore in some sense be considered as primitive building blocks of our Solar system. We know a great deal about comets from spacecraft flybys and photographs but there are still many unanswered questions, regarding for example their origin and exact contents. Studying comets and their history will help us to understand whether life on Earth began with the help of so-called comet seeding. [ESA, 2005a]

On 2nd March 2004 an Ariane V rocket was launched from French Guyana in South America by ESA, carrying a 12-cubic-meter spacecraft into space. As an important event of the Rosetta project, this successful lift-off was the start of a 12 year journey for the Rosetta spacecraft to reach a comet called 67P/Churyumov-Gerasimenko. Upon reaching the comet, Rosetta will go into orbit around it and release a small lander onto its surface, which will examine the geology, atmosphere, magnetic fields and radiation at the surface as well as analyzing samples from the comet's nucleus. This is the first space mission ever to put a lander on a comet. The Rosetta orbiter itself will follow the comet in its orbit for more than a year, studying its evolution as it approaches perihelion.

Rosetta will circle the Sun four and a half times before reaching the comet. A few times during this time the spacecraft passes close to Earth and Mars. The reason for this is to let it gain gravitational energy from the planets. The energy is needed for Rosetta to increase its speed enough to catch up with the comet at a distance of five AU [1] from the Sun. Rosetta will pass close by Earth three times and once come near Mars. During these planetary flybys the path of the spacecraft will be inside the magnetospheres, and sometimes even the ionospheres, of the planets, allowing the instruments from the *Rosetta Plasma Consortium (RPC)* to measure physical quantities such as magnetic field strength, temperature and plasma density. [ESA, 2005a]

A group at the *Swedish Institute of Space Physics, Uppsala Division (IRF-U)*, deeply involved in the Rosetta mission, is responsible for the Langmuir probe instrument (LAP) on board the spacecraft (see Section 1.4). The scientists decide how the instrument should be configured during the mission to get the best measurement results. To do this, one has to have a model of the space environment around the spacecraft and to know how the environment can affect, or even damage, the instruments.

The objectives of this thesis are to:

- create a Matlab tool for analyzing the Rosetta trajectory and attitude data provided by ESOC that can be used for all planetary flybys during the Rosetta mission
- provide the Rosetta LAP team at IRF-U with background information for planning the first Earth flyby, including presenting the trajectory and altitude in suitable coordinate systems
- estimate to what extent the LAP electronics would be exposed to radiation from the van Allen belts. This estimate lead to the decision that it was safe to have the instrument turned on during the flyby
- investigate when the spacecraft position and attitude will place the Langmuir probes in sunlight and when in eclipse, when they will be in wake behind the spacecraft and when they will experience a plasma flow. This is to verify the calculations of the spacecraft attitude and to make easier the analysis of the measurements from the Langmuir probes.

An overview of the Rosetta path through the inner solar system is shown in Figure 1, and the mission time table is according to Table 1. The whole mission is divided into five periods and the graphs show two-dimensional projections of the path of Rosetta and the comet onto the ecliptic plane. a) shows the path from launch to the first Earth flyby, b) covers the time from the first Earth flyby to the Mars flyby, c) the following time to the second Earth flyby, d) the following time to the third Earth flyby and e) the last period of the mission before reaching the comet.

---

[1] Astronomical Unit = the mean Sun-Earth distance = 149 598 000 km

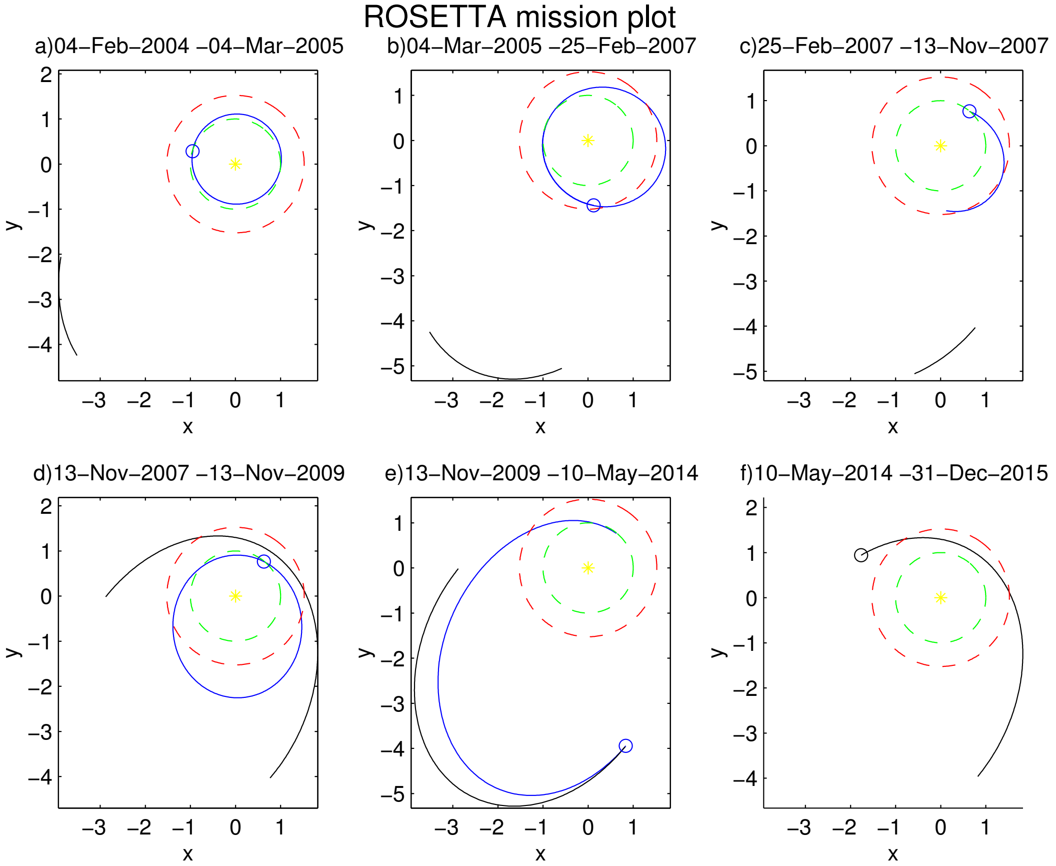


Figure 1: The path Rosetta has to travel to reach the comet between Mars and Jupiter, projected in the ecliptic plane. The path of Rosetta in blue, the path of the comet in black and the orbits of Earth and Mars in green and red respectively. Note that the comet completes about $1\frac{1}{2}$ revolution around the Sun between the Rosetta launch and their final rendez-vous in 2014. The circles denote the end of the time intervals. All motion is counterclockwise around the Sun.

1.2 The spacecraft and its equipment

The main structure of Rosetta has the dimensions 2.8 x 2.1 x 2.0 meters. It is, simply speaking, an aluminum box filled with computers, electronics and propellant tanks. On the outside are mounted two large solar panels for power supply as well as the thrusters[2], the sensors of all the scientific instruments and the communication antennas. The small landing module is also mounted on one of the sides. The launch weight of the whole spacecraft is approximately three tonnes, of which half is propellant[3] and about 100 kg is the lander.

Even though Rosetta will travel quite far out in the solar system (just outside the Jupiter orbit), ESA decided to rely solely on solar power in this project. The two solar panels on board, each with a length of 32 m, generate a maximum of

---

[2] Small rockets with a max force of 10 N used for orbit and attitude corrections [ESA, 2005a].

[3] Fuel and oxidiser

Step	**Date**		**Event**
1	2004	Mar 2	Launch
2	2005	Mar 4	Earth flyby
3	2007	Feb 25	Mars flyby
4	2007	Nov 13	Earth flyby 2
5			First asteroid flyby
6	2009	Nov 13	Earth flyby 3
7			Second asteroid flyby
	2011	Feb	Enter hibernation mode
8	2014	Jan	Exit hibernation mode
	2014	May	Rendezvous manoeuvre
	2014	Aug	Global mapping
9	2014	Nov	Lander delivery
	2015	Aug	Perihelion passage
	2015	Dec	End of mission

Table 1: Timetable for the Rosetta mission.

8 500 W for Rosetta's instruments and subsystems, and a minimum of 400 W when furthest out at 5.25 AU where the radiation from the Sun is only 4% of that at the Earth. To save energy and operational costs Rosetta will go into *hibernation mode* much of the time during the outward journey, with almost all systems switched off.

For communicating with the Earth, Rosetta is equipped with a high-gain antenna, which is a big parabolic dish, 2 meters in diameter, mounted on one side of the spacecraft. The radio waves are transmitted and received in two frequency bands: S-band (2 GHz) and X-band (8 GHz). The main ground antenna used for Rosetta is ESA's station in New Norcia in Western Australia, though half a dozen of ESA's and NASA's other deep space antennas are used from time to time. The largest distance the radio signals will have to travel in space is more than 1000 million kilometers, which will take up to 50 minutes. Therefore, Rosetta is made to be as "intelligent" as possible and does not always depend on orders from Earth. [ESA, 2005a]

On board Rosetta there are a number of scientific instruments provided by different groups of researchers across Europe and the USA. There are in total 16 instruments, six of them forming the *Rosetta Plasma Consortium (RPC)*. The RPC includes five instruments to measure the physical properties of the comet's nucleus, examine the structure of the inner coma, monitor cometary activity and study the comet's interaction with the solar wind. The dual Langmuir probe instrument (LAP), provided by IRF-U, is one of these five instruments. Figure 2 shows how the RPC instruments are mounted on Rosetta, including the two Langmuir probes each mounted on the tip of a deployable boom.

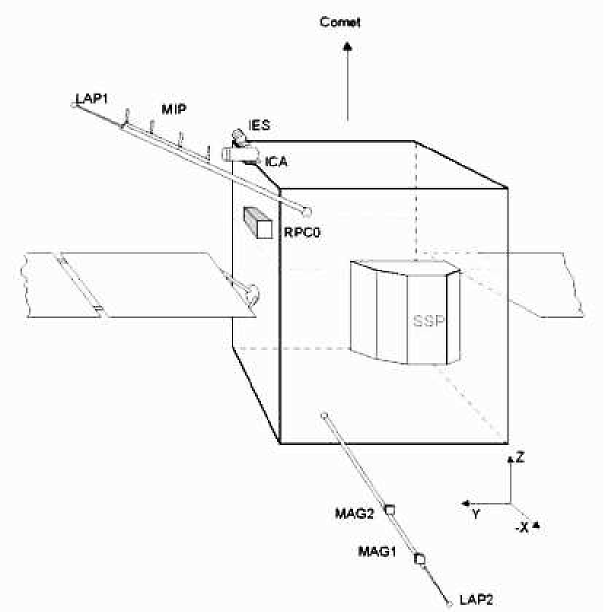


Figure 2: Sketch of the Rosetta spacecraft showing the configuration of the RPC sensors. [Trotignon et al., 1999]

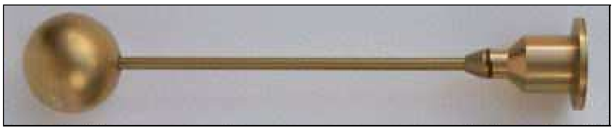


Figure 3: One of the two Langmuir probes mounted on Rosetta. The diameter of the spherical sensor is 50 mm. (http://www.space.irfu.se/rosetta/)

1.3 The space environment around the Earth

Figure 4 shows schematically the space environment around the Earth. The solar wind, coming from the left in the figure, interacts with the Earth's magnetic field, which is compressed on the day side and drawn out on the night side to form the region known as the *magnetosphere*. The magnetosphere is bounded by the *magnetopause*, and drawn out into a long *geomagnetic tail* on the night side. The tail consists of two *lobes*, separated by a *neutral sheet*. In front of the magnetosphere, a *bow shock* forms, separating the supersonic solar wind from the subsonic flow in the magnetosheath. Inside the magnetosphere, the *plasmasphere* is a toroidal plasma region of dense cold plasma close to the Earth, with the high energy particles trapped in the *Van Allen radiation belts* forming similar structures (both these regions are located in the area marked *trapping region* in Figure 4). Outside the trapping region, the magnetospheric plasmas are generally tenuous (a few particles per cm^3), cold in the tail lobes and hot elsewhere. [Kivelson and Russell, 1997].

The first Earth flyby is performed at a minimum altitude of about 2 000 km from the Earth's surface on its day side, and with an inclination to the equator of

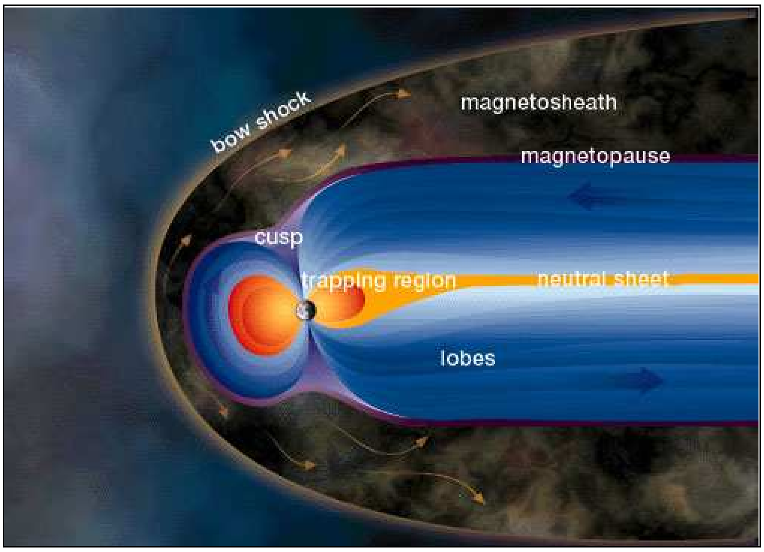


Figure 4: Earth's magnetosphere and its interaction with the solar wind. The inner and outer van Allen Belts are shown in orange/red. (Image from http://www.windows.ucar.edu/)

less than 40°. Rosetta enters the magnetosphere from the deep geomagnetic tail, crosses the Van Allen radiation belts and goes deep into the plasmasphere, before exiting through the magnetopause and bow shock.

The inner and outer Van Allen radiation belts, containing a lot of trapped high-energy protons and electrons respectively, form torii along the Earth's equator. Spacecrafts traveling through these radiation belts are exposed to the high-energy particles which can penetrate through the hull of the spacecraft and may damage the electronics inside. In some situations when the instruments are exposed to a lot of radiation it may even be necessary to turn off the electronics to minimize the risk of damaging it.

A high-energy proton in the radiation belts around the Earth typically has kinetic energy in the order of tens of MeV and electrons 1-10 MeV. As the masses of the two particle types differ a lot (the proton is almost 2 000 times heavier than the electron) the chance of being hit by a fast electron is much higher than that of being hit by a slow proton and in terms of radiation doses the electrons often give the highest contribution.

1.4 Some theory on the Langmuir probe

The main objective of the dual Langmuir probe instrument, LAP, is to study the plasma density, temperature and flow velocity near the comet. When placed in space plasma, the charged particles constituting the plasma collide with the probe. By applying a voltage difference between the spacecraft and the probe and measuring the current flowing from/to the probe one gets a current/voltage relation looking like the graph in Figure 5. From an "I/V curve" like this, being the result of a voltage sweep measurement, it is possible to calculate the plasma density and the electron- and proton temperatures of the surrounding space environment. In equations (1) to (8) we present the relation between biased voltage and probe current as derived by Mott-Smith and Langmuir in 1926 [Mott-Smith and Langmuir, 1926] in a so called OML-approach. For a more thorough explanation on the theory of the Langmuir probe please refer to [Behlke et al., 2000].

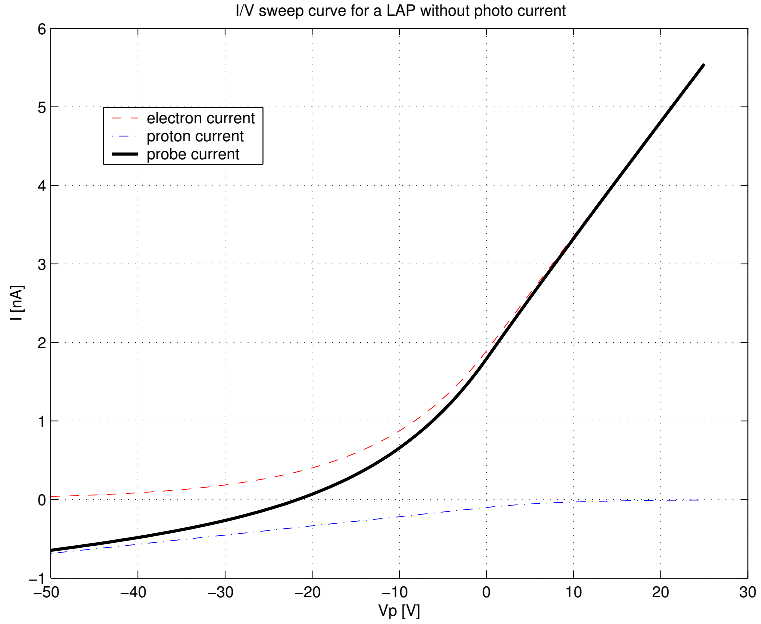


Figure 5: The relation between the bias voltage V_p and the probe current for an ideal probe in a collision-less homogeneous solar wind plasma. The probe current is the sum of the electron current and the ion current. Plasma parameters: $n_e = 2.5 \cdot 10^6 \ m^{-3}, n_i = 7 \cdot 10^6 \ m^{-3}, T_e = 15 \cdot 10^4 \ K$ and $T_i = 10^4 \ K$

Let V_p and V_s be the voltage of the probe and the spacecraft with respect to the plasma. We apply a bias voltage U_B to the probe, so

$$V_p = V_s + U_B \tag{1}$$

If we define

$$\chi_j = \frac{q_j V_p}{2\pi m_j} \tag{2}$$

then, for a positive bias potential, the electron current and ion current are, in the case of spherical probes, given by

$$I_e = I_{e0}(1 - \chi_e) \tag{3}$$

$$I_i = I_{i0} e^{-\chi_i} \tag{4}$$

and for negative bias potential

$$I_e = I_{e0} e^{-\chi_e} \tag{5}$$

$$I_i = I_{i0}(1 - \chi_i) \tag{6}$$

where

$$I_{j0} = -A_P n_j q_j \sqrt{\frac{k_B T_j}{2\pi m_j}}, \tag{7}$$

A_P is the area of the probe, n_j is the number density of particle species j (ions or electrons), m_j the particle mass, q_j the charge of the particle and T_j is the temperature. Subscript e and i means electrons and ions/protons respectively. The current is defined as positive when going *from* the probe *to* the plasma. To get the total probe current we now just take the sum of the electron and ion currents

$$I = I_e + I_i. \tag{8}$$

When placed in sunlight the probes will also emit a *photoelectron current* (or a *photocurrent*) in addition to the plasma electron and ion current. Photons in the UV range[4] coming from the Sun will hit the probes and release electrons from its surface, causing an electron current which affects the I/V-curve. As the electrons leave the probe the photocurrent counts as *negative* according to the definition above. In tenuous magnetospheric and solar wind plasmas the photocurrent will be the dominating current and a freely floating probe (to which the net current would be zero) will thus be brought to a positive potential in order to attract back many of the emitted photoelectrons. To estimate the magnitude of the photocurrent one has to consider the area of the probe projected to the Sun, the surface properties of the probe, the distance to the Sun and the solar spectrum. It is thus difficult to find a valid theoretical expression for the photocurrent and in this treatment we adopt the empirical expressions suitable for the near-Earth space environment:

$$I_{ph} = -I_{ph}^0, \qquad V_p < 0 \tag{9}$$

$$I_{ph} = -I_{ph}^0 e^{-V_p/T_f}, \quad V_p > 0 \tag{10}$$

As seen in the equations above, for negative potentials all photoelectrons can escape from the probe and the photocurrent will be saturated at the constant value. For probes at positive potentials a higher potential means that the probe will collect more photoelectrons. The total probe current is now the sum of the electron current, the ion current and the photocurrent:

$$I = I_e + I_i + I_{ph} \tag{11}$$

---

[4] The ultraviolet region of the solar spectrum (10-380 nm)

Typical values for a sunlit probe on Rosetta are: $I_{ph}^0 = 80$ nA and $T_f = 2$ eV. An I/V curve for a sunlit probe can look like the plot in Figure 6. The photocurrent for low bias voltage clearly dominates over the electron and ion currents.

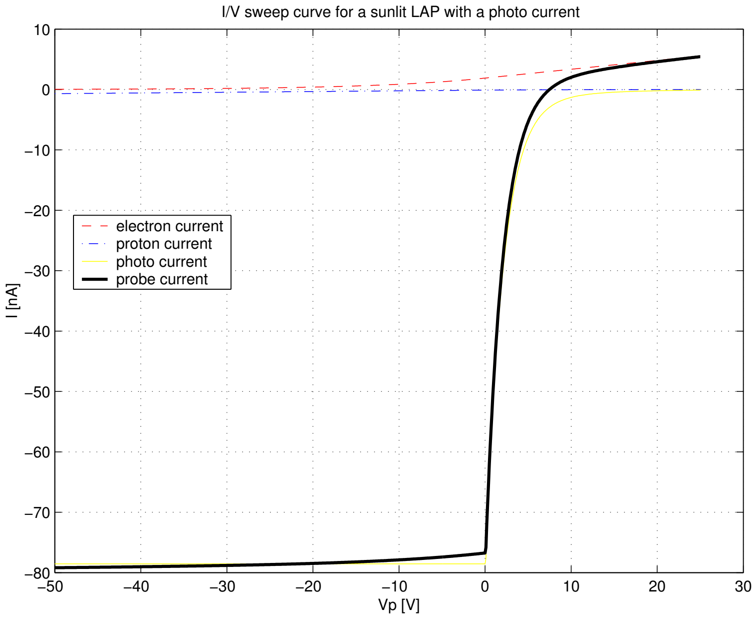


Figure 6: The current/voltage relation for a sunlit probe in the same plasma as in Figure 5.

2 Spacecraft trajectory information

A main prerequisite for all space missions is to have a well-known trajectory for the spacecraft. When planning the path of a spacecraft one must calculate very accurately the position for all times so that things like planetary flybys and encounters with minor planets occur as intended. Even more important is to know where the spacecraft is during the actual flight, as well as its attitude (orientation of the spacecraft axes in space). For this specific study the trajectory for Rosetta is needed for calculating the amount of radiation from the *van Allen radiation belts* as well as its attitude to see when the probes will be sunlit/in eclipse and when they are in the wake of the spacecraft.

At ESA all satellite operations are managed by the *European Space Operation Centre (ESOC)*. This is also the authority providing the trajectory data to all the scientific groups working within the space missions. In the case of the Rosetta project the latest update of the trajectory and attitude data is made available via the *Data Distribution System (DDS)* [ESA, 2003a]. The data files, covering both the time passed since launch and the future part of the Rosetta mission, are constantly being updated to match the spacecraft's real (measured) position in space.

There are a number of different tools for reading, calculating and presenting trajectory data. One example is the program package *SPICE*[5], used by many research teams to convert between different coordinate systems. Another program is the *Orbit Visualization Tool (OVT)*[6], developed at IRF-U and widely used for modeling the Earth space environment for satellites with geocentric orbits. NASA's *Jet Propulsion Laboratory (JPL)* provides an on-line system known as *HORIZONS*, for generating (geocentric) trajectories for some known objects like planets, comets and satellites. Despite the range of available software for handling trajectory data, easy and understandable Matlab routines for that purpose are needed for easy inclusion in the data analysis routines developed at IRF-U and elsewhere, and also as a flexible tool for planning of operations.

2.1 The equipment and data inputs

When working on my Diploma work, much of the time has been spent on programming coordinate transformation routines and getting the data in a format that is easy to work with. The inputs I have used in terms of trajectory and attitude data are the following:

- A file containing the Rosetta coordinates in a heliocentric frame of reference, given for a number of time points covering the whole 12-year mission,

---

[5] http://naif.jpl.nasa.gov/naif/pds.html

[6] http://ovt.irfu.se

provided by ESOC on the DDS. [ESA, 2003a]

- One file for each planetary flyby with the same information as above but in a planetary-centered reference frame (same source as above).
- A file with the spacecraft attitude covering the whole mission, i.e. actual data for past time and plans for the future (same source as above).
- Coordinate files for the Sun, Moon and Mars provided by JPL through the *HORIZONS* website[7].

The trajectory files provided by ESOC that was used for the first Earth flyby include the Cartesian (x,y,z) coordinates for Rosetta in heliocentric and geocentric reference frames, as well as Rosetta's (Cartesian) velocity vector at each point in time. All files begin with a header including information about the coordinate system and the time system. Thereafter comes the data, each variable in a separate column and with the date and time column to the left. A sample of a file header is shown in Appendix A.1. Refer to [ESA, 2003a] for a closer description of the data format.

2.2 Coordinate systems and transformations

To be able to reach the objectives of this thesis it is necessary to express the Rosetta trajectory in a number of different coordinate systems. Three main systems are used (see Table 2 for details).

- A geocentric system having one axis always pointing towards the Sun, providing a good overview of the Earth flyby (GSE).
- A geocentric system rotating with the Earth, providing data for calculating the trapped radiation as well as a plot of the trajectory expressed in geographical longitude and latitude (GEO).
- A Rosetta centered system with axes pointing along the spacecraft axes, which is suitable to represent for example the motion of the Sun as seen from the Langmuir probes.

The coordinates systems used in this thesis are listed in Table 2 together with the definition of the axes and the center of the system.

In the raw data files from ESOC the coordinates are given in a system called J2000. To keep the names of the variables and coordinate systems as consistent as possible the J2000 is, in this report and in the associated program, hereafter called GEI. A more thorough explanation of the transformation routines is found in Section 4.4.

---

[7] http://ssd.jpl.nasa.gov/horizons.html

Name		**Center**	**x-axis**	**z-axis**
GEI	Geoc. Equatorial Inertial	Earth	aries	Earth spin axis
GEA	Geoc. Ecliptic Aries	Earth	aries	ecliptic north
GSE	Geoc. Solar Ecliptic	Earth	earth-sun line	ecliptic north
GEO	Geographic	Earth	lat=long=0	Earth spin axis
ROS	Rosetta Internal System	Rosetta	fixed in s/c	fixed in s/c

Table 2: The used coordinate systems and where their axes point.

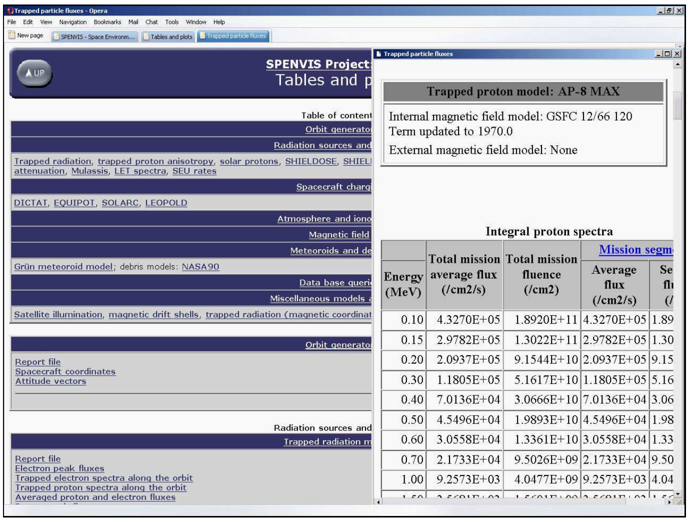
3 Radiation

3.1 Model

Even though the particle density in the Van Allen belts is low, the speed of the particles contribute to a high electron- and proton flux. This, together with the fact that the regions extend over quite a large volume around the Earth, makes the radiation belts a potential danger for the spacecraft electronics when traveling through. What decides the total dose of radiation that, in the case of this thesis, the RPC electronics inside Rosetta will be exposed to depends mainly on the following:

- The Rosetta path through the radiation belts
- The placement of the the RPC electronics inside the spacecraft
- The spacecraft hull and instrument box (material, thickness)

Many theoretical models treating the space environment around the Earth have been developed, and for this thesis we use the *SPace ENVironment Information System (SPENVIS)*[8], an ESA sponsored system developed by the *Belgian Institute for Space Aeronomy*. The SPENVIS tool is a web-based collection of models for the space environment and its effects on spacecrafts, developed mainly for satellites orbiting the Earth.



Trapped proton model: AP-8 MAX

Internal magnetic field model: GSFC 12/66 120
Term updated to 1970.0
External magnetic field model: None

Integral proton spectra

Energy (MeV)	Total mission average flux (/cm2/s)	Total mission fluence (/cm2)	Mission segm	
			Average flux (/cm2/s)	Se fl (/
0.10	4.3270E+05	1.8920E+11	4.3270E+05	1.89
0.15	2.9782E+05	1.3022E+11	2.9782E+05	1.30
0.20	2.0937E+05	9.1544E+10	2.0937E+05	9.15
0.30	1.1805E+05	5.1617E+10	1.1805E+05	5.16
0.40	7.0136E+04	3.0666E+10	7.0136E+04	3.06
0.50	4.5496E+04	1.9893E+10	4.5496E+04	1.98
0.60	3.0558E+04	1.3361E+10	3.0558E+04	1.33
0.70	2.1733E+04	9.5026E+09	2.1733E+04	9.50
1.00	9.2573E+03	4.0477E+09	9.2573E+03	4.04

Figure 7: Screenshot from the the web-based space environment modeling tool SPENVIS.

To be able to use SPENVIS for calculating the radiation on Rosetta during its first Earth flyby the spacecraft trajectory must be introduced in the model and

---

[8] http://www.spenvis.oma.be/spenvis/

this proved to be a problem. The present version of SPENVIS can only treat closed elliptical orbits around the Earth which have been calculated using SPENVIS' own orbit generator or the predefined orbits of known satellites. For this specific case though, thanks to the helpful staff at *Belgian Institute for Space Aeronomy*, we were able to upload the Rosetta coordinates to SPENVIS after transforming them to a geographical coordinate system.

When running SPENVIS, we used the NSSDC models AP-8 and AE-8[9] for trapped protons and electrons, respectively, in the terrestrial radiation belts. Running the model using the above mentioned models resulted in values for the proton and electron flux and fluence for a number of different particle energies as functions of time. For the solar wind we used the conditions for solar max to make it a "worst case" scenario, even though it was not the actual case.

To estimate the total radiation dose (energy deposited in the target), we used the SPENVIS implementation of the SHIELDDOSE-2 model (v 2.10) developed by NIST[10], assuming a silicon target inside a sphere of 1, 2 or 3 mm or behind a 1 mm semi-infinite slab of aluminum. This is a very rough approximation of the RPC electronics box, placed on the inside of Rosetta's $+y$ side according to Figure 2 (where the box is named RPC0), but nevertheless it is enough to estimate the order of magnitude. The modeling can be refined using SPENVIS' "Sectoring analysis for more complex geometries" and "Multi-Layered Shielding Simulation (Mulassis)" but the results from the first run suggest that this is not necessary.

---

[9] http://see.msfc.nasa.gov/ire/models.htm

[10] http://www.nist.gov

3.2 SPENVIS results

Predicted fluxes of particles during the Rosetta Earth flyby are shown in Figure 8. The plots show the flux of protons with energy content above 10 MeV and 30 MeV and the flux of electrons with energies above 1 MeV and 5 MeV. The time-integrated flux for the full flyby, i.e. the total number encountered or the particle fluence, is found in Table 3 and illustrated as a cumulative plot in Figure 9. The estimated total radiation doses are listed in Table 4.

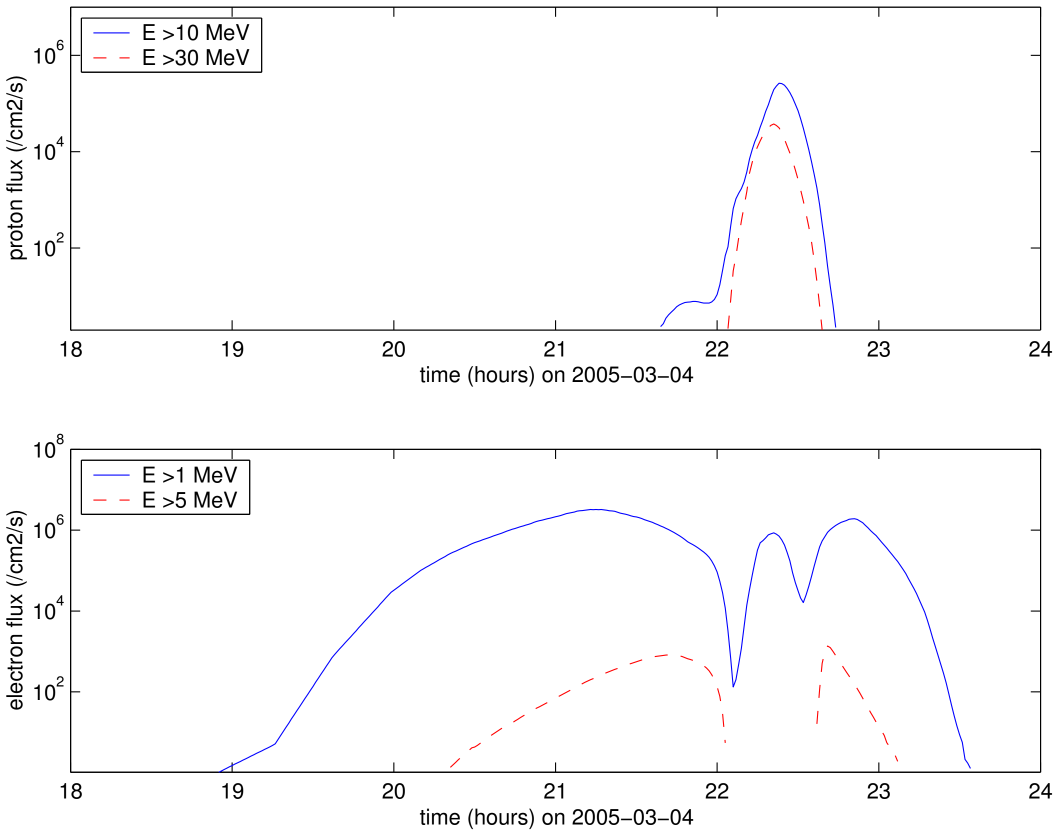


Figure 8: Predicted fluxes of protons above 10 MeV and above 30 MeV, and of electrons above 1 MeV and 5 MeV, from the SPENVIS implementation of the AP-8 and AE-8 models for solar max conditions.

Particles		**Fluence** $(\mathbf{cm^{-2}})$
Protons	>10 MeV	$1.5 \cdot 10^8$
Protons	>30 MeV	$2.1 \cdot 10^7$
Electrons	>1 MeV	$1.1 \cdot 10^{10}$
Electrons	>3 MeV	$2.2 \cdot 10^6$

Table 3: Predicted fluencies of high energy particles

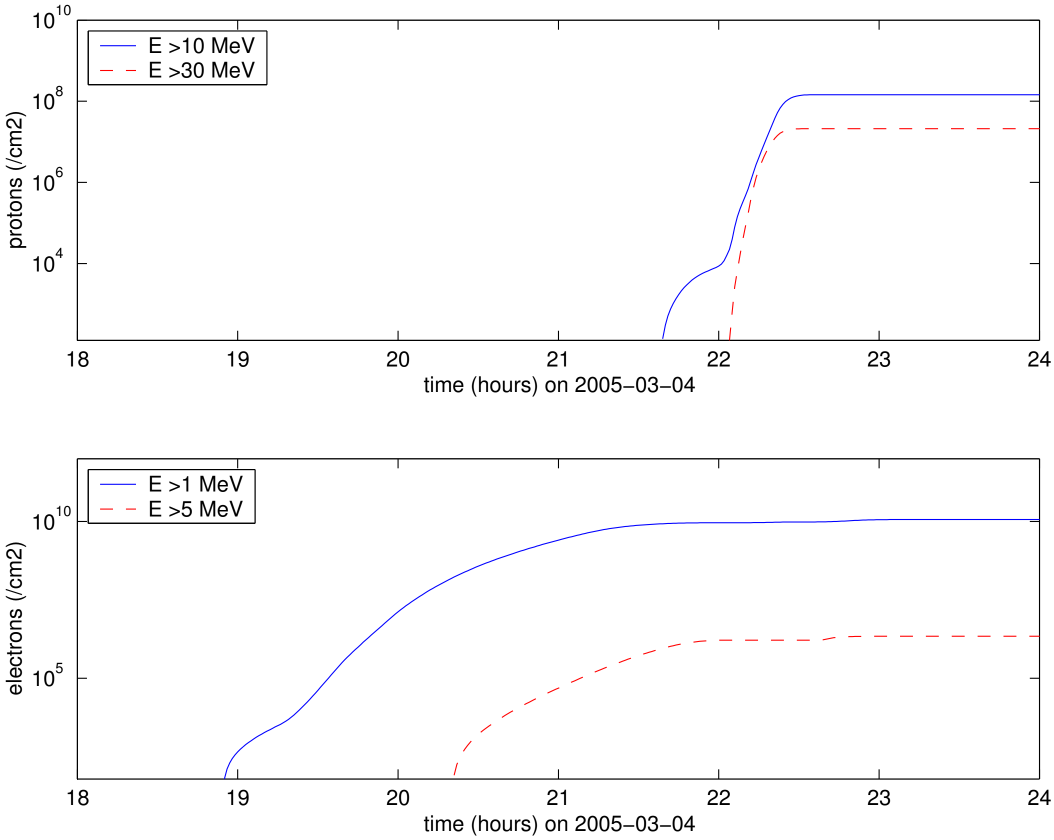


Figure 9: Predicted fluence of protons above 10 MeV and 30 MeV, and of electrons above 1 MeV and 3 MeV, from the SPENVIS implementation of the AP-8 and AE-8 models.

Al thickness	**Total dose [Rad]**	**From p+ [Rad]**	**From e- [Rad]**
1 mm (sphere)	660	72	579
2 mm (sphere)	186	20	162
3 mm (sphere)	66	10	54
1 mm (slab)	80	13	65

Table 4: Radiation dose expected for the first Rosetta Earth flyby from protons and electrons, for a silicon target within an aluminum sphere of given thickness, or behind a semi-infinite 1 mm Al plate. The ”total dose” column also includes small contributions from bremsstrahlung and solar protons, though the radiation belt particles clearly dominate.

3.3 Conclusion on radiation

The exact fluencies can of course vary a lot with the actual magnetospheric conditions, but the results above nevertheless provide a baseline for estimating the possible impact of the radiation belts. To put them into perspective, we can compare to what we normally expect to find in interplanetary space. As suggested by [Feynman et al., 1990] typical yearly averaged fluxes of solar proton event particles vary between 10^7 and 10^{10} protons/year above 30 MeV, with 10^9 a reasonable number a few years after solar max. This would suggest that for the protons, the radiation belt passage gives a dose equivalent to what we may expect to get in about a week of operations in interplanetary space. Hence, there is no reason to worry about total dose effects.

For the RPC electronics box, a relevant model may be a 0.5 mm thick Al sphere, representing the RPC-0 electronics box, behind a 1 mm semi-infinite Al slab, representing the spacecraft. The last row of Table 4 may thus be taken as an upper limit to what may be expected. The total dose on RPC main electronics should thus be small, below 100 Rad. As noted above, the effects of this dose are largely independent on whether we are on or off, so this has little impact on operations.

Regarding Single Event Upsets (SEU): No thorough investigation has been done to see whether the LAP electronics are likely to be a target to bit-flips and latch-ups, but according to [Åhlén, 2005] and the results presented above showing a total radiation of 100 Rad during some 20 minutes, the LAP team need not worry about the instrument being damaged.

Letting the LAP electronics be turned on showed to be a good decision. The radiation from the Van Allen belts caused no problems whatsoever for any of the RPC instrument and the LAP data was successfully delivered to Earth on 10th March [ESA, 2005b].

4 Matlab routines for trajectory data

The Matlab program described in this section is available to download from the web site: http://www.space.irfu.se/rosetta/sci. The main program files are also found printed in Appendix A.2.

To produce a useful tool for reading, treating and visualizing the trajectory data that can also be used in the future for upcoming planetary flybys, a Matlab program was written. The basic tasks of the computer program are to:

- read the Rosetta trajectory and attitude data files provided by ESOC
- read the planetary trajectory data files provided by NASA JPL Horizon System
- fit the different data to match in time
- perform coordinate transformations
- calculate time and index for closest approach
- visualize the trajectory and attitude of Rosetta during the flybys in an understandable way
- calculate when LAP probe 1 and 2 are in wake and when they are sunlit

This chapter is written as a handbook for the Matlab program, describing the routines for reading and writing data files, calculating and transforming coordinates and visualizing the results. Matlab version: 6.0.0.88, Release 12.

4.1 Program structure and the program files

The program as a whole consists of some 30 Matlab files including all functions, as shown in Table 5. The main program file `go.m` calls external routines for reading and treating the data as well as for visualizing the results. The idea has been to make the program useful for future flybys and the structure is made so that it should be easy to add new routines, such as new coordinate transformation functions, and to easily change the input data as it is constantly being updated. When running `go.m` the program goes through the following procedure:

- a choice for the user to select what event should be considered, i.e either one of the four planetary flybys or the whole mission
- reading of the ”raw” data from text files for the current event

- interpolation of the Rosetta trajectory and attitude data to match the constant time steps for the planetary trajectory
- calculating things like the time for closest approach and the Julian day numbers for the data
- performing coordinate transformations
- presenting the trajectories (and other data) in a number of different plots

4.2 Reading the input files

On the basis of which event is considered the program `readdata.m` calls different functions for reading trajectory and attitude data from the correct files. The data are stored in matrices and arrays in the raw format (see Section 4.3). It also defines a number of variables unique for the current event (such as the planetary radius and the title of the graphs), so that the data can later be treated as similarly as possible whatever event was chosen. The variables `att` and `moon` tell whether Rosetta attitude data and Moon trajectory data respectively have been read (=1) or not (=0).

4.3 Global variables

Few variables are cleared during the running of the program. This is to save as much data as possible that can be interesting, but it also results in a huge amount of data taking up a lot of computer memory. The naming of the variables is done as consistently as possible according to a few rules:

- Variables for the raw trajectory data are in capitals (RR, VR, RS, RM and Q).
- Generally in the beginning of variable names `r` means position vector and `v` means velocity vector, followed by the body itself: `r` for Rosetta, `s` for the Sun and `m` for the moon. `Asc` refers to the spacecraft attitude.
- The calculated (transformed) coordinates are stored in variables with names such as `rr_gse` which in this case means Rosetta's (xyz) position vector expressed in the GSE coordinate system.
- `i` often means index, `iclosest` for example is the index number for the closest approach in the trajectory data variables

Trajectory data, attitude data and time data are all stored in a similar way. Even though the dimensions of the matrices differ (time is one dimensional, trajectory data two dimensional and attitude data three dimensional) they all have the same length. This is a result of the interpolation that is done to make all data match each other in time.

Filename		**Description**
go.m	p	the main program, calling the other programs
readdata.m	p	reads the input data
readjpltraj.m	f	reads data from a JPL file
readros.m	f	reads data from an ESOC trajectory file
readatt.m	f	reads data from an ESOC attitude file
coordtransform.m	p	performs the coordinate transformations
vis.m	p	visualizes the resulting data
cart2sphere.m	f	convert cartesian coordinates to spherical
drawrosetta.m	f	draws a box model of Rosetta seen from a probe
drawmodellonglatf.m	f	draws a filled box model of Rosetta
drawmodellonglat.m	f	draws a box model of Rosetta
gea2gei.m	f	transformation from GEA to GEI
gea2gse_a.m	f	transformation from GEA to GSE for att. data
gea2gse.m	f	transformation from GEA to GSE
gea2mea.m	f	transformation from GEA to MEA
gei2gea_a.m	f	transformation from GEI to GEA for att. data
gei2gea.m	f	transformation from GEI to GEA
gei2geo.m	f	transformation from GEI to GEO (xyz)
geo2longlat.m	f	transformation from GEO (xyz) to GEO (long,lat)
imagexy.m	f	calculates image x,y values from spherical coorinates
P.m	f	rotation matrix (around x axis)
Q.m	f	rotation matrix (around y axis)
R.m	f	rotation matrix (around z axis)
point3d.m	f	plots a pointing vector in a 3D graph
point.m	f	plots a pointing vector in a 2D graph
circle.m	f	draws a circle in 2D
sphere.m	f	draws a sphere in 3D
splot3.m	f	draws 2D projections of a 3D curve
ssphere.m	f	draws 2D projections of a 3D sphere
att_e1.txt	d	spacecraftattitude data for the whole mission
traj_comet.txt	d	trajectory data for the comet
traj_mars_m_earth.txt	d	trajectory data for Mars
traj_m_e1.txt	d	Moon trajectory data, Earth flyby 1
traj_m_e2.txt	d	Moon trajectory data, Earth flyby 2
traj_m_e3.txt	d	Moon trajectory data, Earth flyby 3
traj_r_e1.txt	d	Rosetta trajectory data, Earth flyby 1
traj_r_e2.txt	d	Rosetta trajectory data, Earth flyby 2
traj_r_e3.txt	d	Rosetta trajectory data, Earth flyby 3
traj_r_m.txt	d	Rosetta trajectory data, Mars flyby
traj_r_whole.txt	d	Rosetta trajectory for the whole mission
traj_s_e1.txt	d	Sun trajectory, Earth flyby 1
traj_s_e2.txt	d	Sun trajectory, Earth flyby 2
traj_s_e3.txt	d	Sun trajectory, Earth flyby 3
traj_s_m_earth.txt	d	Sun trajectory, Mars flyby

Table 5: The matlab files making up the program. p=program, f=function, d=data file.

4.3.1 Time variables

There are a few different ways of expressing the time for the data used in this program. These are the *Julian day number*, the *Modified Julian day number* and the date and time (yyyy-mm-dd HH:MM:SS) [Hapgood, 1992]. The main variable used in the program for keeping track of the time is `jd` which is the Julian day number. In addition the variable `time` (a four row matrix containing the day of the month, hour, minute and second) is used for visualizing purposes, and the modified Julian day, `mjd`, used for the transformation into geographical coordinates (GEO).

The Julian day number (JD) is defined as the float number counting the days since 12:00 January 1, 4713 B.C[11]. The Modified Julian day (MJD) is similar but uses another offset: 00:00 November 17, 1858, corresponding to the Julian Day 2400000.5 [Hapgood, 1992]. Thus, mathematically:

$$MJD = JD - 2400000.5 \tag{12}$$

For calculating the Julian day number the Matlab function `datenum` is used. The function is given the date in vector form and returns a day number, by adding a constant value (`timediff`) the Julian day number is obtained. As mentioned earlier in the report an interpolation is also done so that the data is defined for constant (one minute) time steps. The reason for this is both the origin of the planetary trajectory data and that it simplifies the visualizing, making it easy to plot for example Rosetta's position for every hour.

4.4 Coordinate transformation routines

A list of the used coordinate systems and a description of where their axes are pointing is shown in Table 2 in Section 2, and a flowchart over the transformation procedure used in the program is shown in Figure 10. Each square in the diagram represents a set of spatial vectors expressed in different coordinate systems, grouped in ”rows” according to the type of vectors (attitude, Rosetta trajectory, Sun and Moon trajectory). The arrows show the transformations and the resulting data (red squares) is achieved by combining data from different types of vectors.

The diagram in Figure 10 shows the coordinate transformation procedure. Definitions of the coordinate systems are found in Table 2. The routines for converting between GEA, GEI, GSE and GEO are taken from [Hapgood, 1992] and definitions of the basic $\Re^3$ rotation matrices (in the program called P,Q and R) from [Weisstein, 1999]. All functions for transforming between geocentric systems work in the same way, receiving a two dimensional matrix with vectors in the old sys-

---

[11] from Wikipedia online encyclopedia (http://en.wikipedia.org)

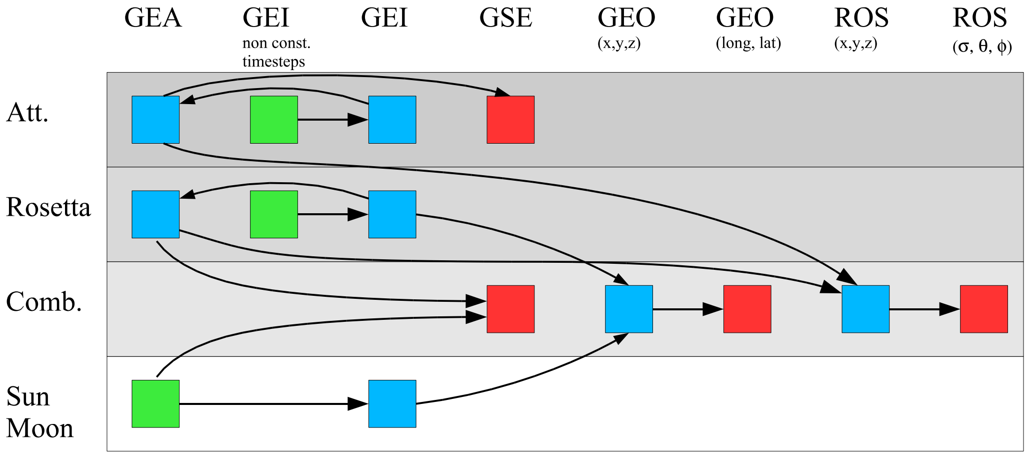


Figure 10: Flowchart showing the transformation between the coordinate systems. Green boxes represent the provided raw data and red boxes the resulting (useful) data. The arrows show the necessary coordinate transformations.

tem (and possibly additional data needed for the transformation) and returning the transformed coordinates in the same format but in the new system. Transforming vectors from a geocentric ecliptic system into Rosetta's frame of reference (GEA→ROS) however, is a bit different. Here a translation of the vectors is first performed to the spacecraft and then rotated using the attitude matrix.

4.5 Attitude data handling

The spacecraft attitude data files provided by ESOC have a structure similar to that of the spacecraft trajectory files, but instead of having six columns defining Rosetta's position and velocity vector in space there are four columns defining the so called *quaternions* for each point in time. From the four quaternions in turn, it is possible to calculate the three axes of the spacecraft in the GEA/J2000 system. This is done by defining a 3x3 matrix where the rows represent the spacecraft's three axes and the columns represent their respective x,y and z-component in GEA/J2000. Refer to [ESA, 2003b] and [ESA, 2001] for details on how to calculate the axes. By definition, this matrix made up by the three spacecraft base vectors expressed in a geocentric system is also the rotation matrix from the geocentric system to Rosetta's system. Similarly, its inverse is the rotation matrix from Rosetta's system to the geocentric system.

4.6 Probe exposure to sunlight and plasma flow

The two Langmuir probes on board Rosetta are mounted on booms (2.3 m and 1.7 m respectively [ESA, 2001]) reaching out from the spacecraft. What they measure in terms of probe currents is, except for the variable biased voltage, determined by the space environment around the probes. To make a complete model of what they travel through in terms of plasma density, temperature, solar radiation and wake effects, is of course a very complicated task and well outside the scope of this thesis. However, with the trajectory and attitude data for Rosetta as well as the planetary trajectories and by knowing the dimensions of the spacecraft, it is possible to calculate when the probes will be sunlit and when they will be shadowed by Rosetta. In a similar way the spacecraft velocity vector gives a rough estimation of when the probes are in the wake (see figure 12).

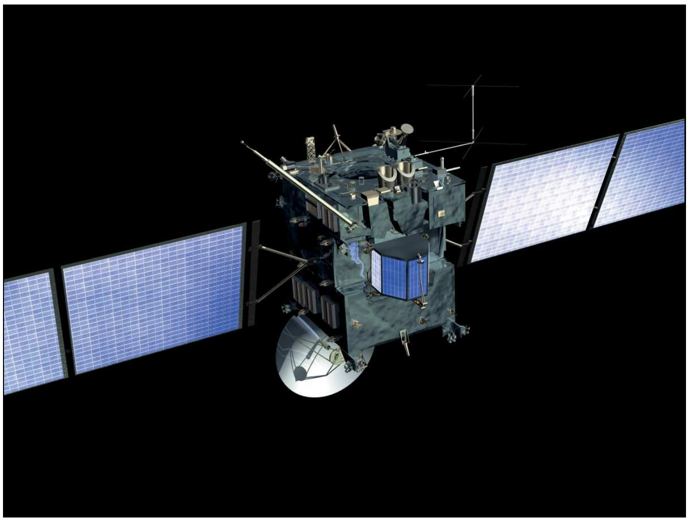


Figure 11: A fully equipped Rosetta with the HGA completely unfolded. Facing the viewer is the -x side with the small lander mounted and the LAP 1 boom, LAP 2 is missing in this model. (from http://esamultimedia.esa.int/images/Science/)

One feature on the spacecraft that affects the amount of sunlight hitting the probes is the position and direction of the high-gain antenna (HGA). The big antenna disc, measuring some two meters in diameter, can be folded and turned in different directions, and sometimes appears in front of the Sun as seen by probe 2. In this work, we have not included any data on the actual HGA position, but consider the two scenarios having it completely folded in and completely unfolded. As another simplification we do not include the lander or details like instruments and thrusters protruding up to a decimeter from the spacecraft box structure. As the solar panels sometimes can shadow probe 1, a variable defines the solar panel rotation angle around the y axis. By default this angle is set so that for each probe the panels block the sun maximally. This should be a fairly good assumption, using the fact that the solar panels are always oriented so that their surface is perpendicular to the direction to the Sun. Figure 11 shows a computer model of Rosetta with all instruments mounted.

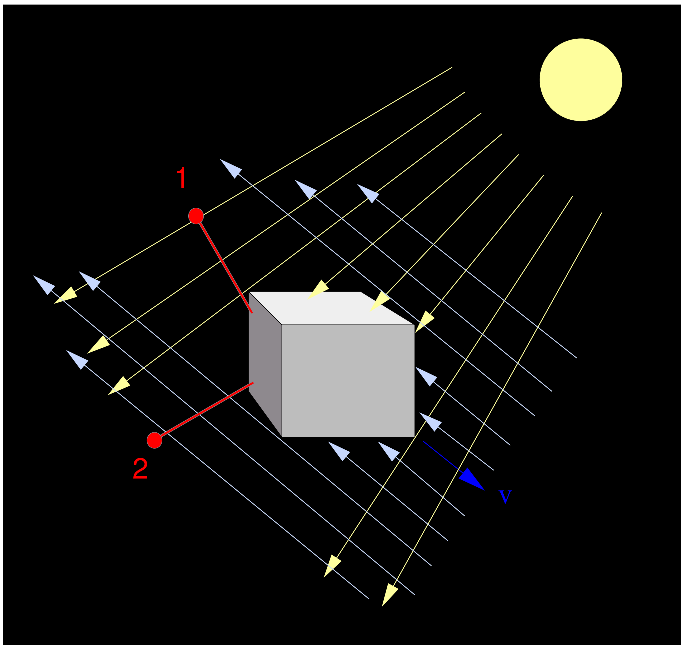


Figure 12: Rosetta's two Langmuir probes mounted on their booms. Here number 1 is exposed to sunlight (yellow arrows) and placed in the wake, number 2 is in eclipse but experiences the plasma flow (light blue arrows) which is a consequence of Rosetta's velocity through the plasma (dark blue arrow).

As the spacecraft attitude matrix is three dimensional and Matlab (6.0) cannot perform multiplication with matrices having more than two dimensions, this is done inside a loop. For every step in time the Rosetta velocity vector, the Sun, Earth and the moon position vectors (given in the GEA system) are rotated to the ROS system using the rotation matrix defined by Rosetta's current attitude vectors expressed in GEA (see Section 4.5). Moving the center of the coordinate system to one of the probes will not affect the rotated vectors (the minimal parallax can be neglected). For short vectors however, this is not the case. A box model of the Rosetta spacecraft will look different if seen from probe 1 and probe 2. Figure 13 shows what the observer would see if placed on the two probes, looking in the direction of Rosetta's x-axis. The motion of the sun and the velocity vector (transformed into Rosetta-centered spherical coordinates from the Cartesian coordinates) would in these plots be represented by a curve, sometimes going "behind" the spacecraft model and sometimes outside it.

To determine when the probes are in eclipse and in wake the program performs a basic image analysis routine (this is done in the visualization file `vis.m`):

The chosen probe plot (shown in figure 13) is saved as an image file (PNG format), manually converted to an indexed image and loaded back into Matlab

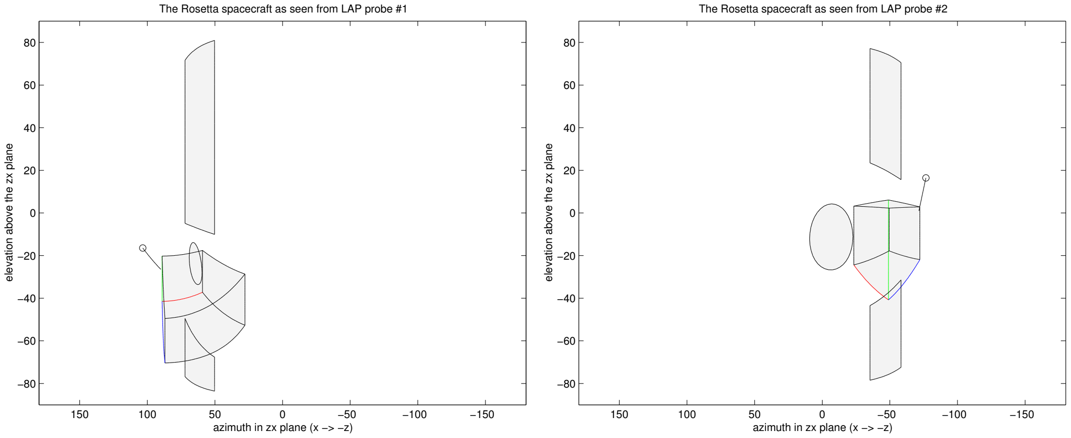


Figure 13: A box model of Rosetta with the high-gain antenna deployed, as seen from LAP 1 (left figure) and LAP 2 (right figure) respectively. The Rosetta's x,y,z axes are colored in red, green and blue respectively.

which then stores the image as a matrix (each element representing the color of the corresponding pixel). Thereafter the program loops through every step in time, calculates which pixel corresponds to the current (spherical) coordinate pair for Sun's position and the velocity vector and determines if it is outside the spacecraft (white) or a part of the spacecraft (other than white). This results in two one-dimensional arrays (`wake#` and `eclipse#` where # stands for the chosen probe number) with the same length as the time vector. A `0` in the arrays corresponds to "outside the spacecraft" and a `1` to "behind the spacecraft".

The algorithm described above is thereafter repeated twice, first time with the high-gain antenna included in the spacecraft model and the second time with the antenna disc having 110% of its actual size. Taking the mean value of the sunlit/wake arrays for the three cases gives a resulting vector from which we get an idea of how close the probes are to being in eclipse and wake respectively. Table 4.6 explains the meaning of each value.

value	**placement of vector in the graph**
`1`	behind the spacecraft model
`0.66`	behind the 100% size antenna disc
`0.33`	behind the 110% size antenna disc
`0`	not behind anything

Table 6: Placement of the vectors corresponding to the elements in the `eclipse` and `wake` arrays.

4.7 Visualizing

When running the visualization program `vis.m` the user can chose between a number of plot types and options. They are:

- trajectory plots in the GSE coordinate system with the planet and Moon motion included, a three dimensional plot as well as two dimensional projections
- same trajectory plots as above but with small colored vectors indicating the spacecraft's x (red), y (green) and z (blue) axes at certain points in time, preferably every hour
- a plot in the geographical system for the Earth flybys, showing Rosetta's path on a world map and its position at the closest approach
- a plot over the Sun's motion (and the motion of the velocity vector) in the ROS system, translated to the two LAP probes respectively
- an option to save the Rosetta box model image file described above for a chosen probe
- an option to calculate the expected eclipse/wake data from the saved image file and store it in variables
- plots showing the illumination and plasma exposure for each probe
- a plot over the whole Rosetta mission in a heliocentric coordinate system, divided up into sub-plots covering the interesting time periods
- an option to set the time span that should apply on the plot data, making it possible to "zoom in" interesting time periods
- an option to store the trajectory and attitude (or whatever) data in a text file with tab separated columns
- an option to store the eclipse/wake data in a file, to be used for comparison with measured data (see Section 6)

5 The first Earth flyby

The following plots present the trajectory and attitude of Rosetta's first Earth flyby in different ways. These plots were used as background information for deciding which LAP modes that were to be run during the flyby. In particular, the attitude information was used to determine the illumination and exposure to the ram flow for each probe (Section 5.2) to ensure proper bias settings of the instrument. In the titles of the plots the time span is expressed in the compact format (dd/mm/yy@HH:MM - dd/mm/yy@HH:MM).

5.1 The trajectory of the 1st Earth flyby

Figure 14 shows Rosetta's altitude vs. time for the hour of the closest approach. In Figure 15 the trajectory of Rosetta is plotted in 3-D GSE: Rosetta approaches the Earth from the night side, turns 90 degrees on the day side and leaves it in the direction of Earth's velocity relative to the Sun. Figure 16 shows Rosetta's path in geographical coordinates with a world map behind, showing that closest approach occurred when Rosetta was above the Pacific Ocean just off the coast of Mexico. The plots in figure 17 and 18 show the attitude of Rosetta during the flyby in 2-D and 3-D, red line representing the x-axis, green line the y-axis and blue line the z-axis.

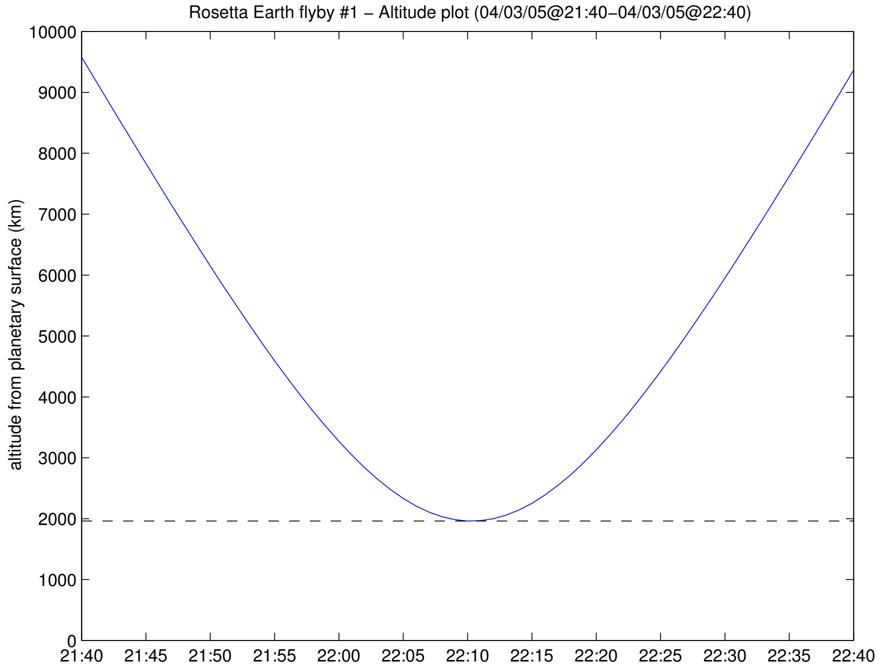


Figure 14: Altitude plot for the 1st Earth flyby, showing the Rosetta-Earth surface distance during its travel through the magnetosphere and the closest approach at 1960 km at 22:10 UT.

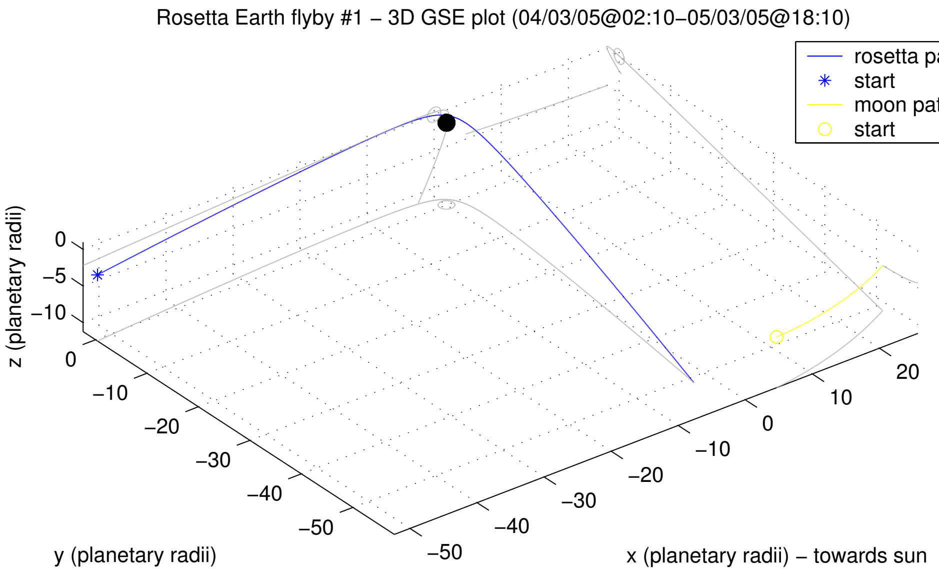


Figure 15: 3-D GSE plot for the first Earth flyby. The Sun is in the positive x direction and the Earth motion relative to the Sun is in negative y direction (anti-clockwise around the Sun).

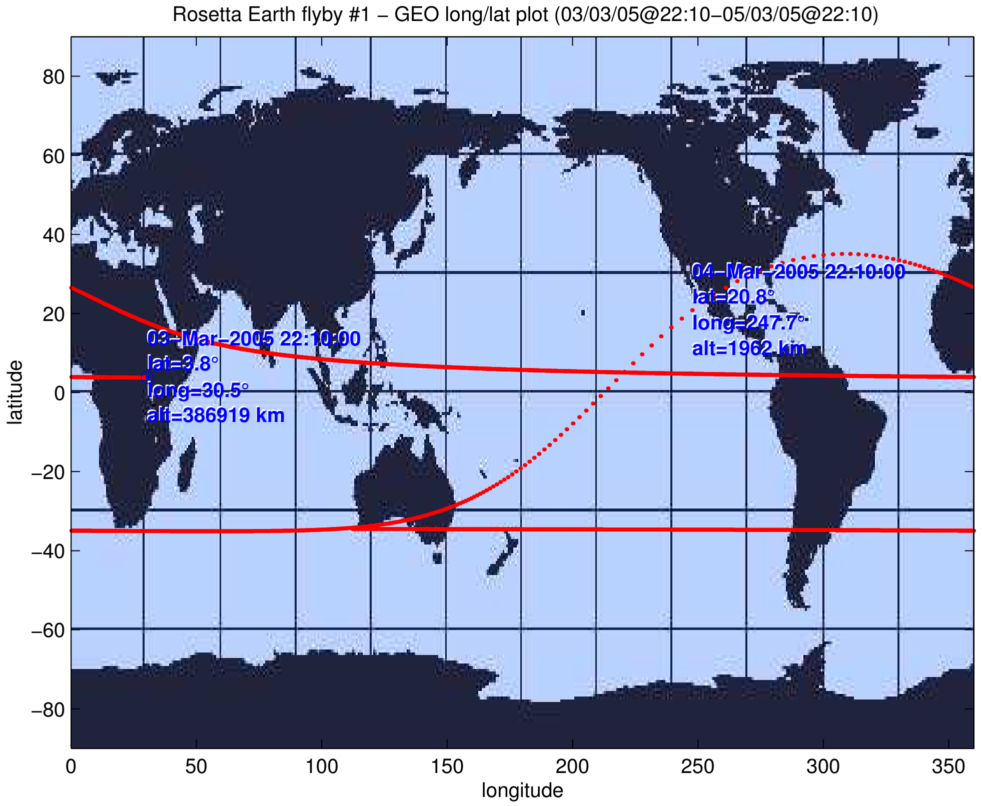


Figure 16: The first Earth flyby plotted in GEO long/lat coordinates on a world map.

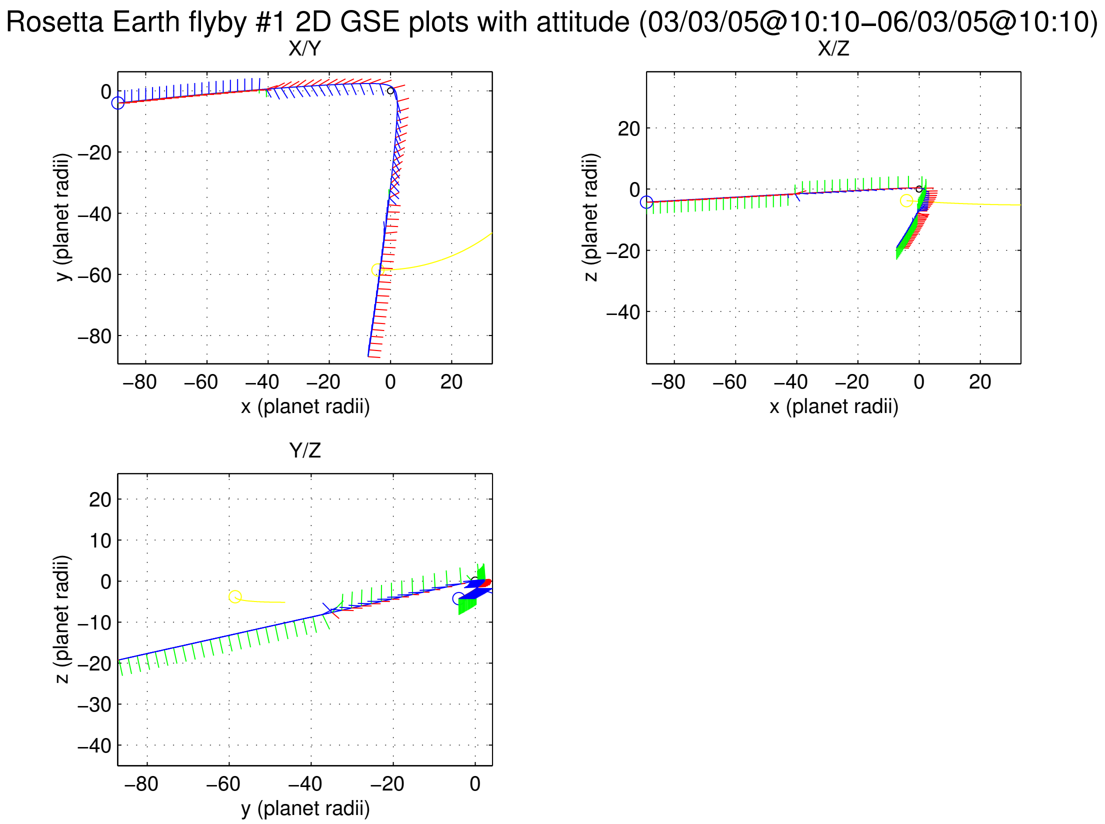


Figure 17: 2D plots of Rosetta's first Earth flyby with s/c attitude information. The small colored vectors represent the Rosetta axes (red, green, blue for x, y, z respectively) plotted for every hour.

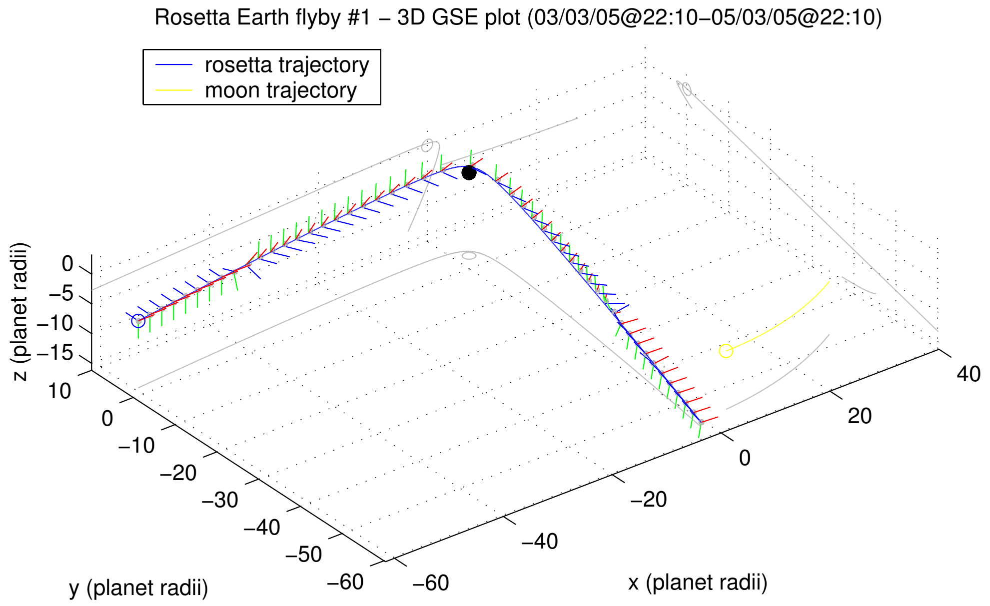


Figure 18: 3D plot of Rosetta's first Earth flyby with s/c attitude information for every hour.

5.2 Probe exposure to sunlight and plasma flow

In Figure 19 the motion of the Sun is plotted in polar coordinates together with a box model of Rosetta as seen from probe 1 and 2. As one can see from the plot, the circular high-gain antenna (which in this picture is completely deployed) is a good blocker of sunlight for probe 2, and its configuration therefore affects a lot what LAP 2 measures in terms of photocurrent. The diagrams in Figure 20 show the velocity vector of Rosetta and the spacecraft box model as seen from probe 1 and 2, which gives a rough estimation of when the probe will be in wake behind the spacecraft as long as the plasma velocity w.r.t. the Earth is much smaller than the velocity of Rosetta in the same frame of reference.

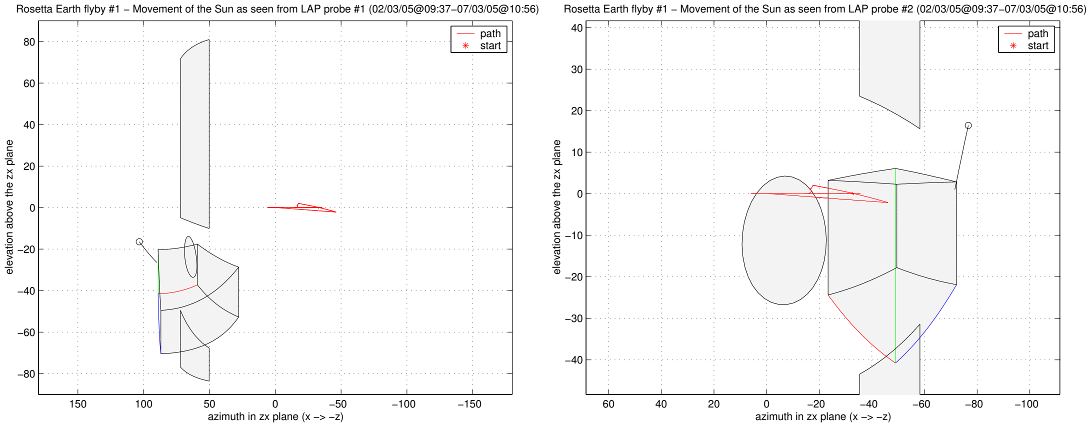


Figure 19: Movement of the Sun as seen from probe 1 (left) and 2 (right) in polar coordinates. The three colored edges of the spacecraft model define the Rosetta coordinate system.

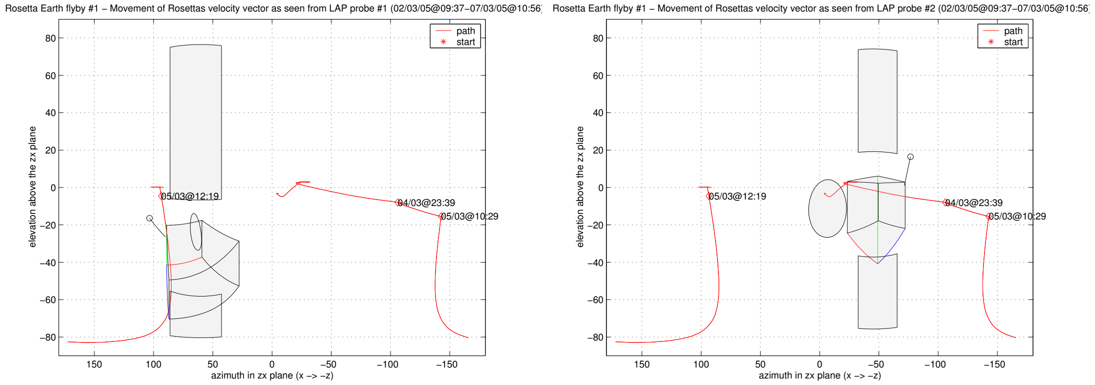


Figure 20: Motion of the velocity vector as seen from probe 1 (left) and 2 (right), in polar coordinates. The three colored edges of the spacecraft model define the Rosetta coordinate system.

Using the function for calculating sunlit and wake data, described in Section 4.6, for probe 1 and 2 respectively resulted in the plots in Figures 21 and 22. We see that probe 1 is not likely to be in eclipse at all during the flyby, but possibly in wake for a short time around noon on 5 March. The calculated results for probe 2 on the other hand is more interesting. The configuration of the high-gain antenna plays an important role here as can be seen from the plot. If the antenna is completely folded in the probe will be sunlit most of the time, whereas if fully deployed it is likely to put the probe in shadow during almost the whole flyby.

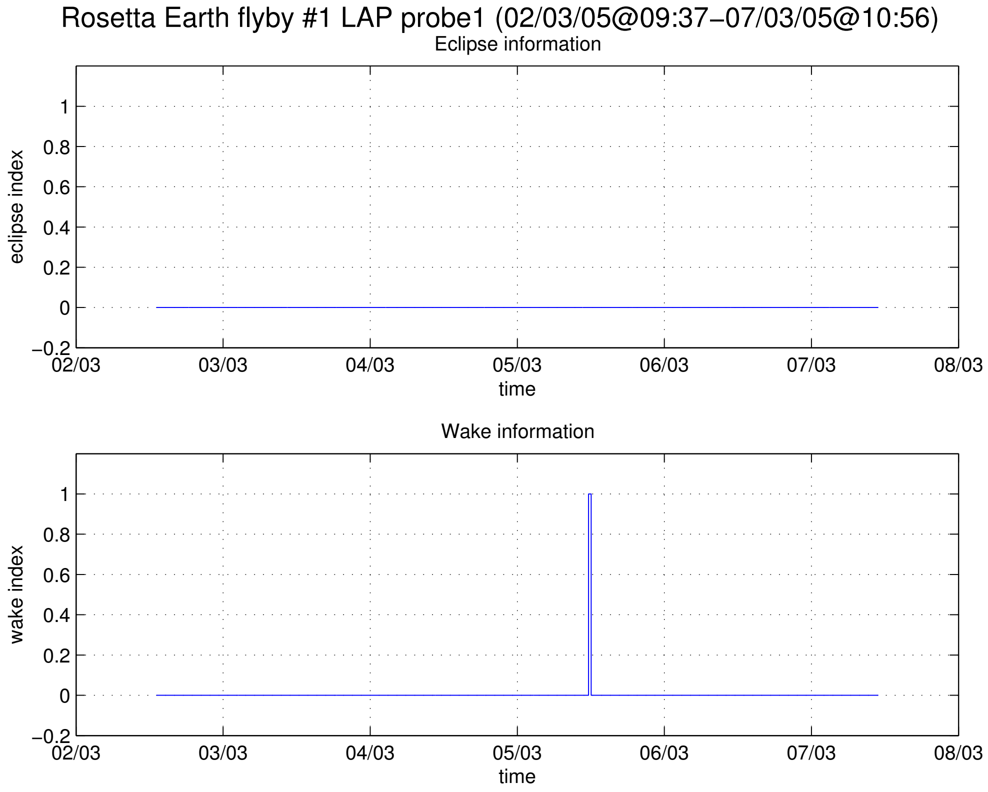


Figure 21: Plots over the estimated eclipse and wake information for probe 1, refer to Table 4.6 for an explanation of the y-axis values.

The sunlit/eclipse index is useful when analyzing the LAP data as it tells when to expect a photocurrent and when not to. The wake index, on the other hand, is of less interest as it only counts with the spacecraft velocity relative to the Earth. Outside the magnetosphere the solar wind speed is typically ten times greater than the speed of Rosetta, resulting in a wake on the shadow side of the spacecraft. For March 4 and 5, when Rosetta is inside the magnetosphere, looking at the wake index may have some relevance though.

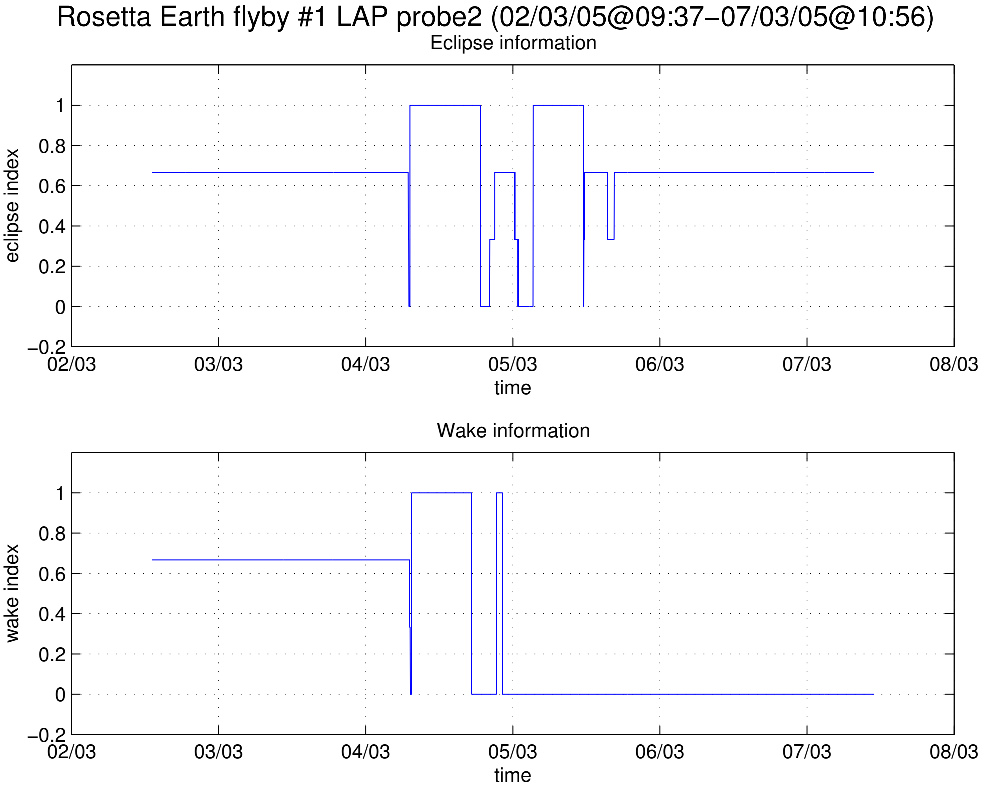


Figure 22: Plots over the estimated eclipse and wake information for probe 2, refer to Table 4.6 for an explanation of the y-axis values.

5.3 The future planetary flybys

Running the Matlab program with the data for Rosetta's second and third Earth flyby and for the Mars flyby results in the following plots shown in Figures 23 to 30.

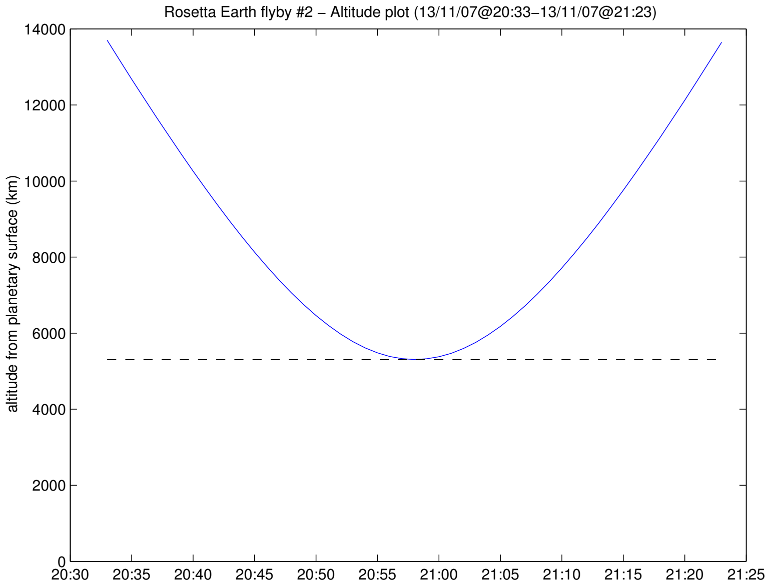


Figure 23: Altitude plot for the second Earth flyby

For the second Earth flyby the geometry of the trajectory looks completely different from the first, as seen in Figure 24. Rosetta approaches the Earth from the dayside, passes through the bowshock and into the magnetosphere. After a closest distance of 5,500 km to the Earth surface Rosetta then leaves on the nightside with an increased inclination.

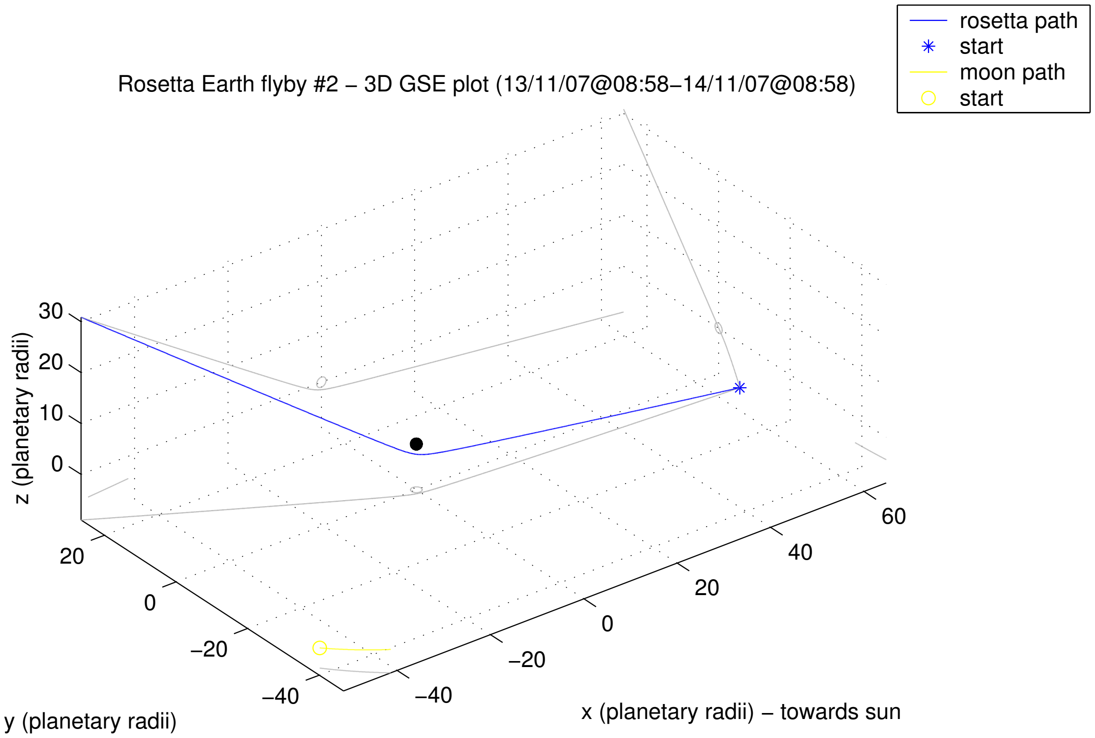


Figure 24: 3D plot of Rosetta's second Earth flyby in GSE

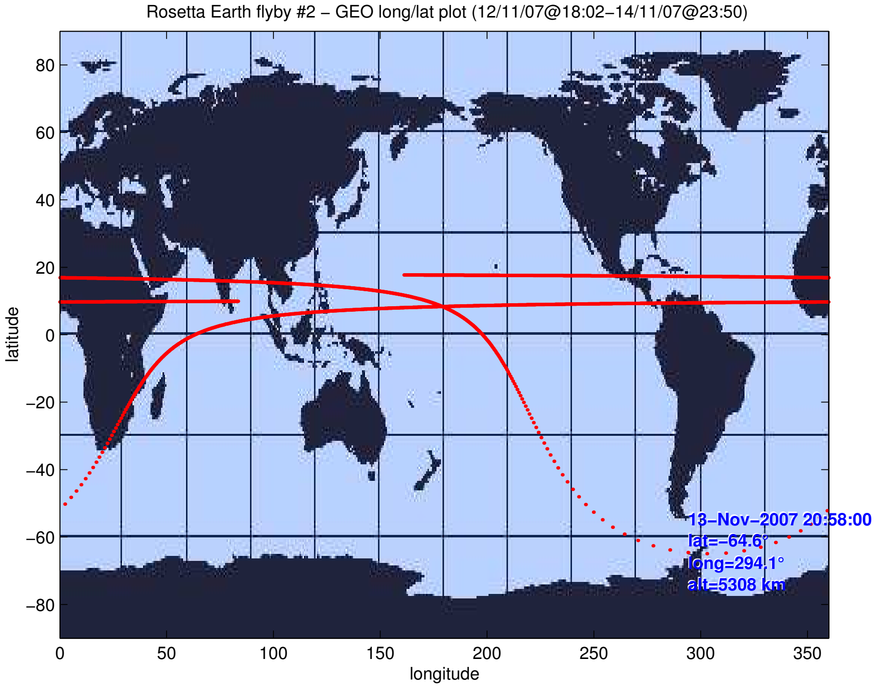


Figure 25: The second Earth flyby plotted in GEO long/lat coordinates on a world map.

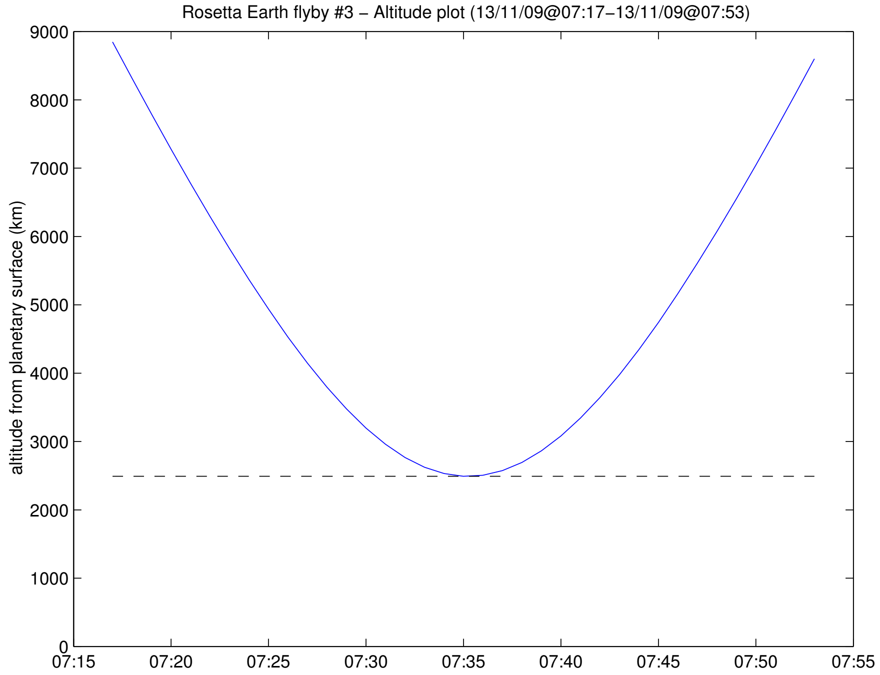


Figure 26: Altitude plot for the third Earth flyby

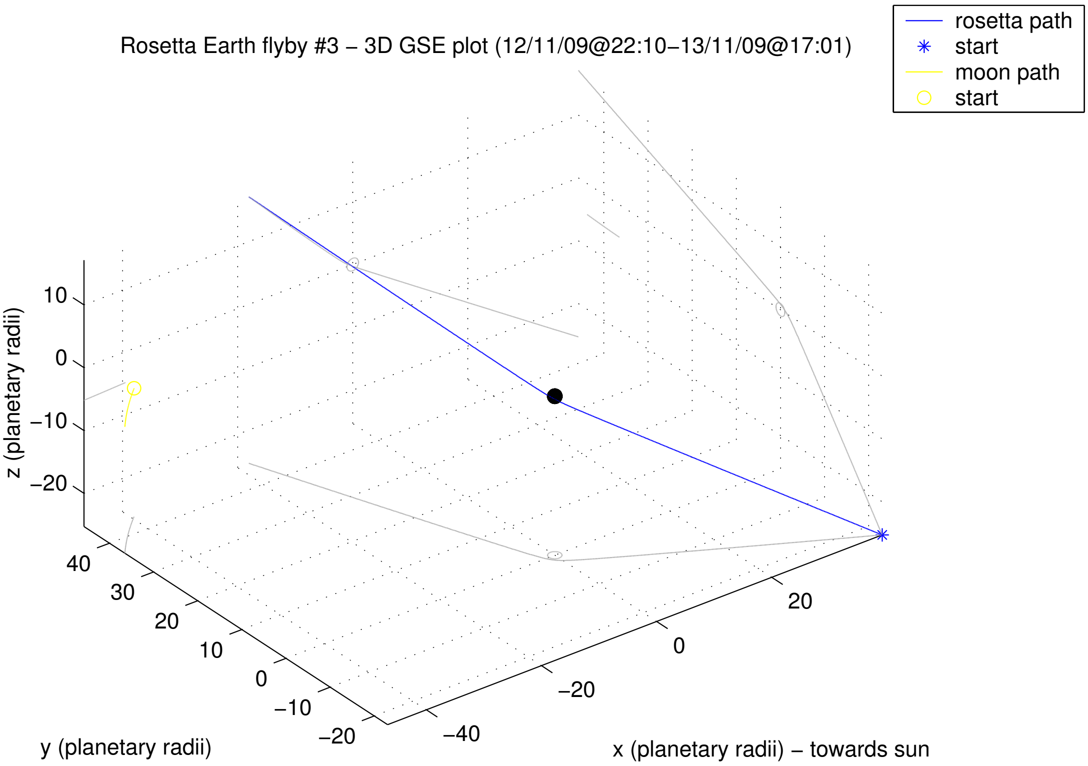


Figure 27: 3D plot of Rosetta's third Earth flyby in GSE

The third (and last) Earth flyby has similar geometry as the second, but with higher inclination and closer approach (2 500 km from the Earth surface).

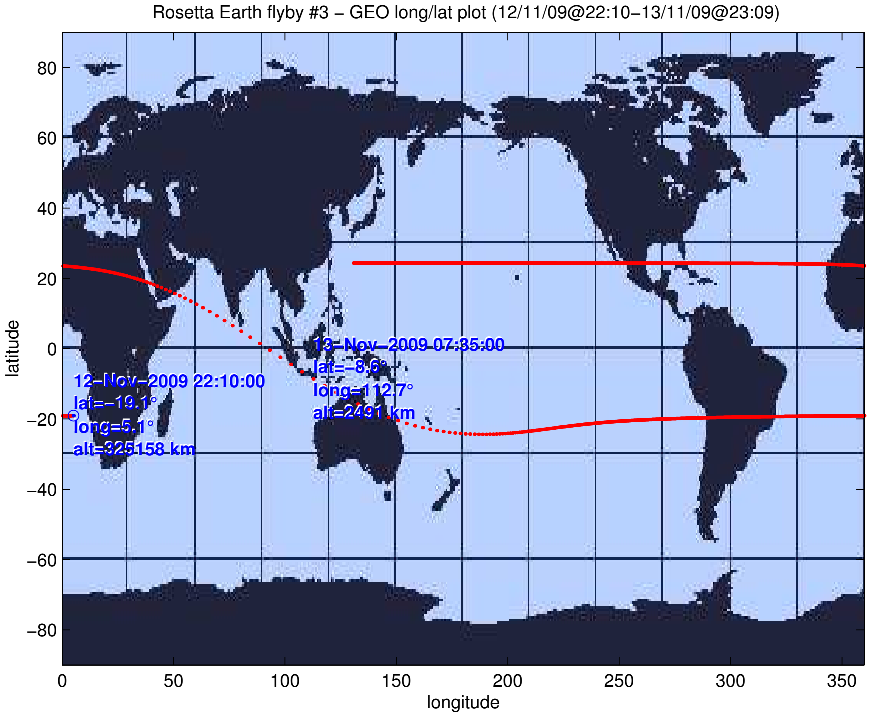


Figure 28: The third Earth flyby plotted in GEO long/lat coordinates on a world map.

The Mars flyby is performed at a minimum altitude of only 260 km providing an opportunity for the RPC team to do interesting measurements well inside the ionosphere of the planet.

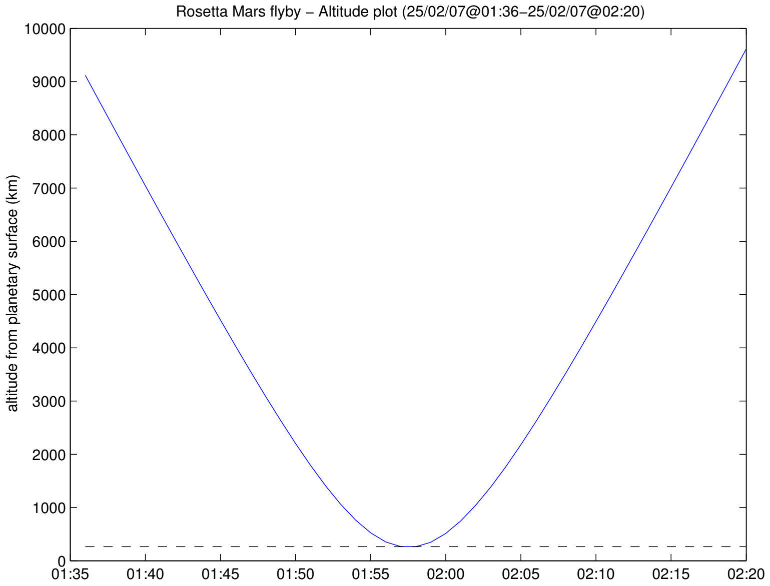


Figure 29: Altitude/time plot for the Mars flyby. Rosetta is intended to go as close as 260 km above the Martian surface.

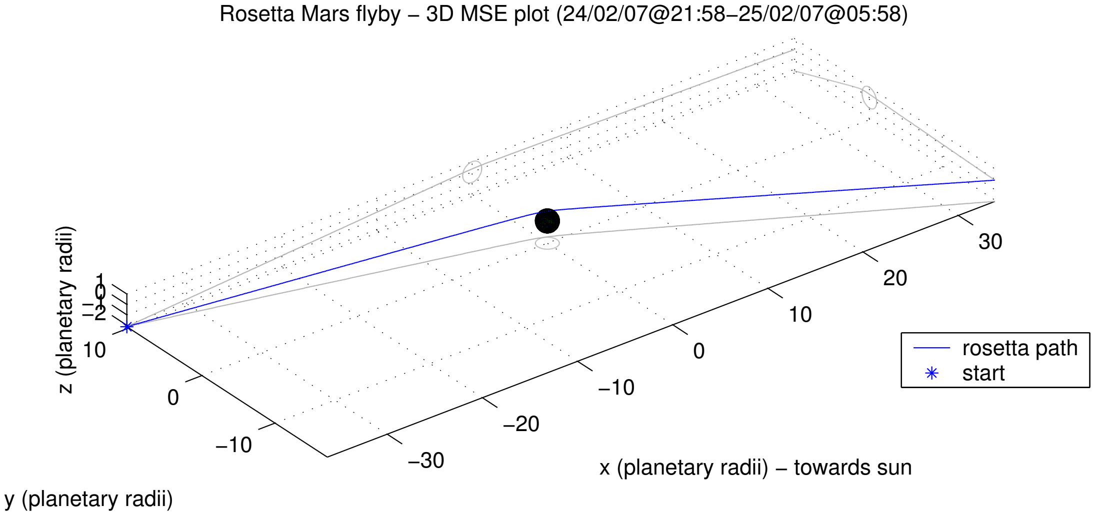


Figure 30: 3-D plot of the Mars flyby (8 hours time span) in a Mars-centered ecliptic coordinate system, x axis pointing towards the Sun.

5.4 Observing Rosetta from Uppsala

On the late evening of March 4, 2005 some members of the Swedish LAP team and a few scientists from IRF-U and Uppsala University gathered at the Westerlund Telescope at the Uppsala Astronomical Observatory in the Ångström Laboratory. The goal was to get a glimpse of the Rosetta spacecraft during its closest approach. With good visual conditions (clear sky, late night) and the possibility to track Rosetta's path with the telescope, Rosetta was observed as a white fast-moving dot.

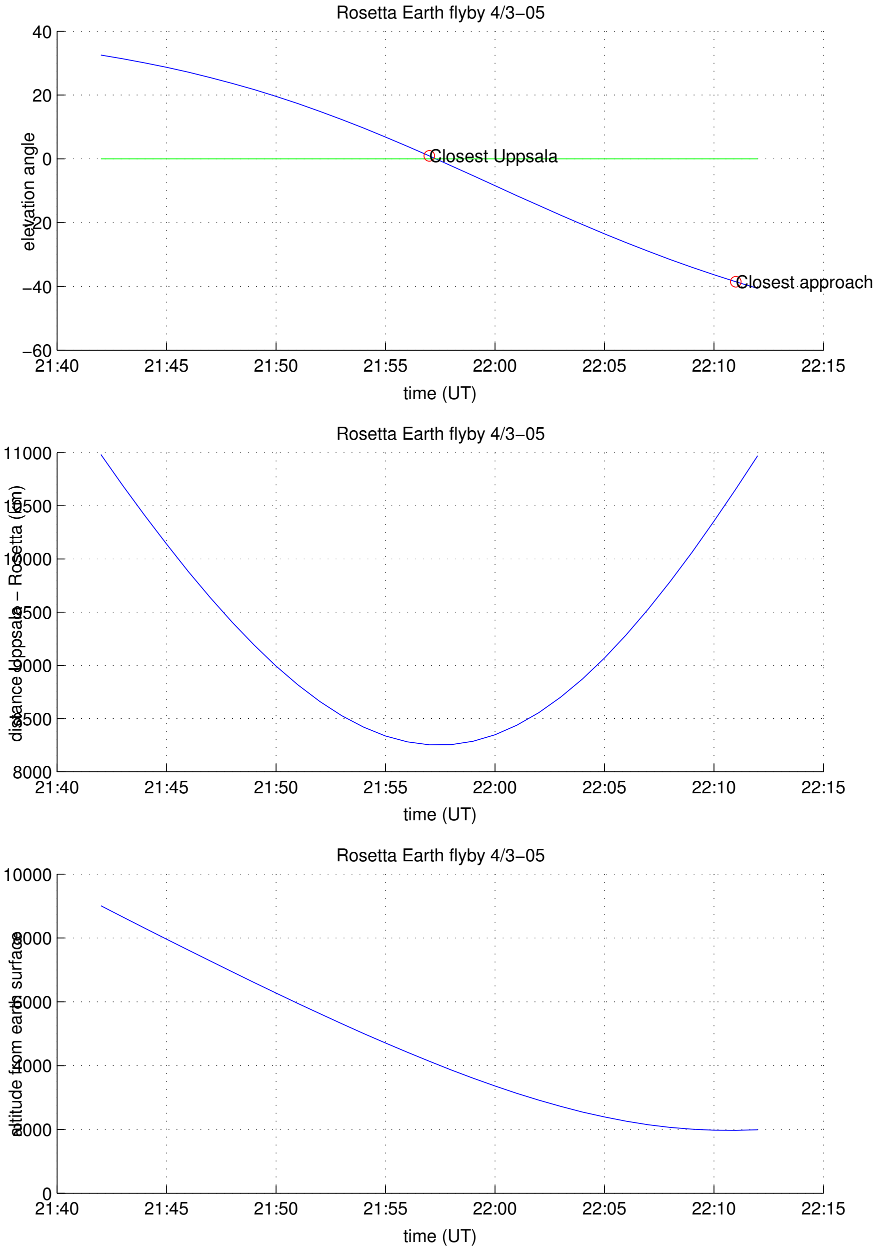


Figure 31: Distance and angle plots for Rosetta as seen from Uppsala on March 4 2005. Times are in UT. As seen in the first plot the minimum Uppsala-Rosetta distance occurred just before Rosetta went down behind the horizon.

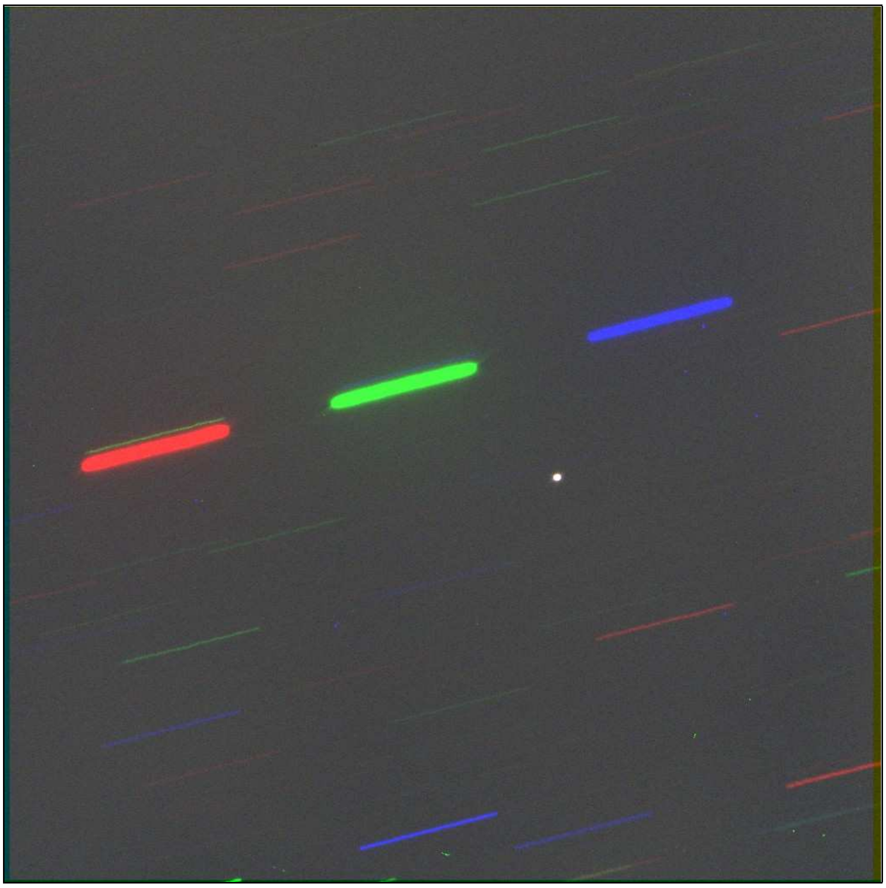


Figure 32: Photo of Rosetta (the white dot) through the Westerlund Telescope at Ångström Laboratory. The colored lines are the red, green and blue camera exposures of a nearby star. (Photo: Ola Karlsson, Department of Astronomy and Space Physics, Uppsala University)

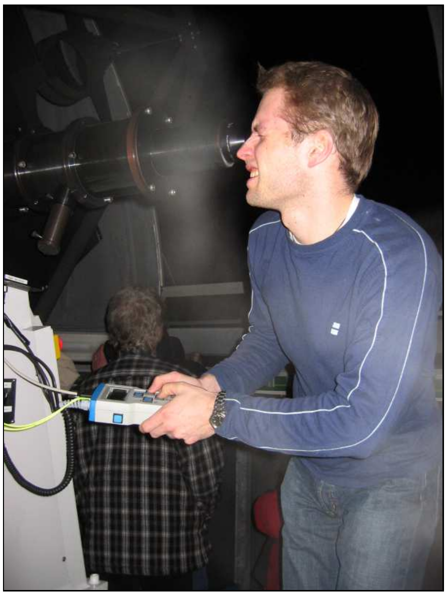


Figure 33: Magnus observing Rosetta through the Westerlund telescope, and tracking its path down towards the horizon in west. (Photo: Anders Eriksson, IRF-U)

6 Measured LAP data

The analysis of the data measured by LAP 1 and 2 that is presented in this section was performed at IRF-U by Anders Eriksson.

6.1 Data from the first Earth flyby

Figure 34 shows the LAP probe 1 current/voltage relation for a sweep measurement (I/V curve) from March 1, i.e. three days before the flyby when Rosetta was still in the magnetotail. One can easily see the photocurrent in the I/V-curve, being the saturated current for high negative voltage.

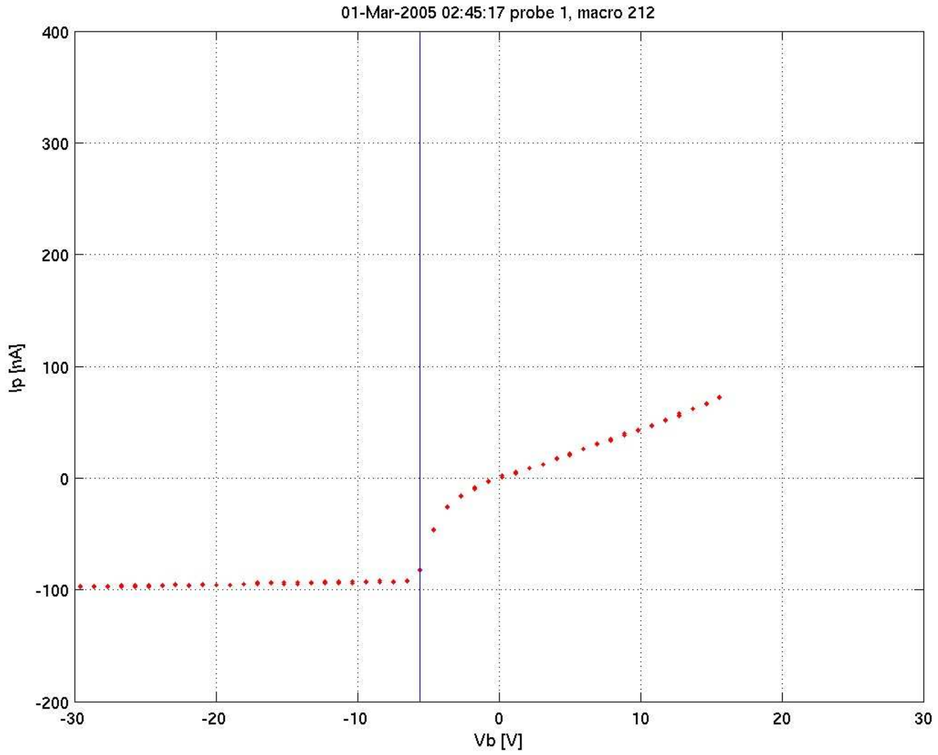


Figure 34: Typical I/V curve from LAP probe 1 (measured on March 1, 2005)

Figure 35 shows the I/V relation for LAP 1 and 2 for the whole flyby. The measured probe current is color coded with a span roughly from -100 nA for low bias voltage up to +300 nA for high bias voltage. The white areas represent missing or filtered data. Figure 36 shows the same data but with a logarithmic current scale.

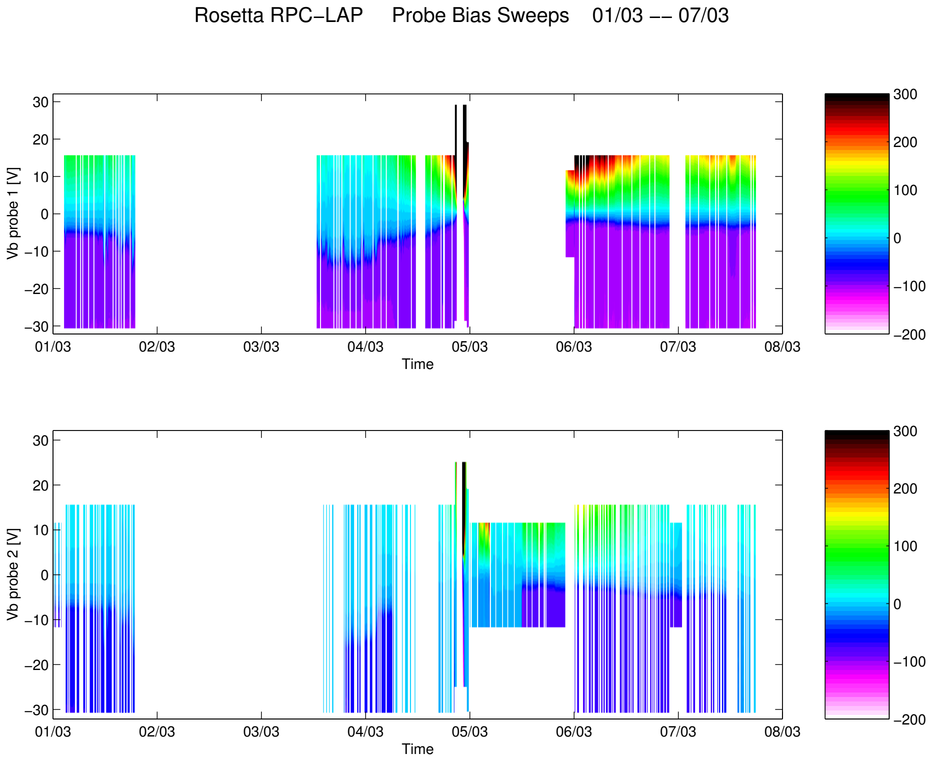


Figure 35: LAP sweep, current in nA

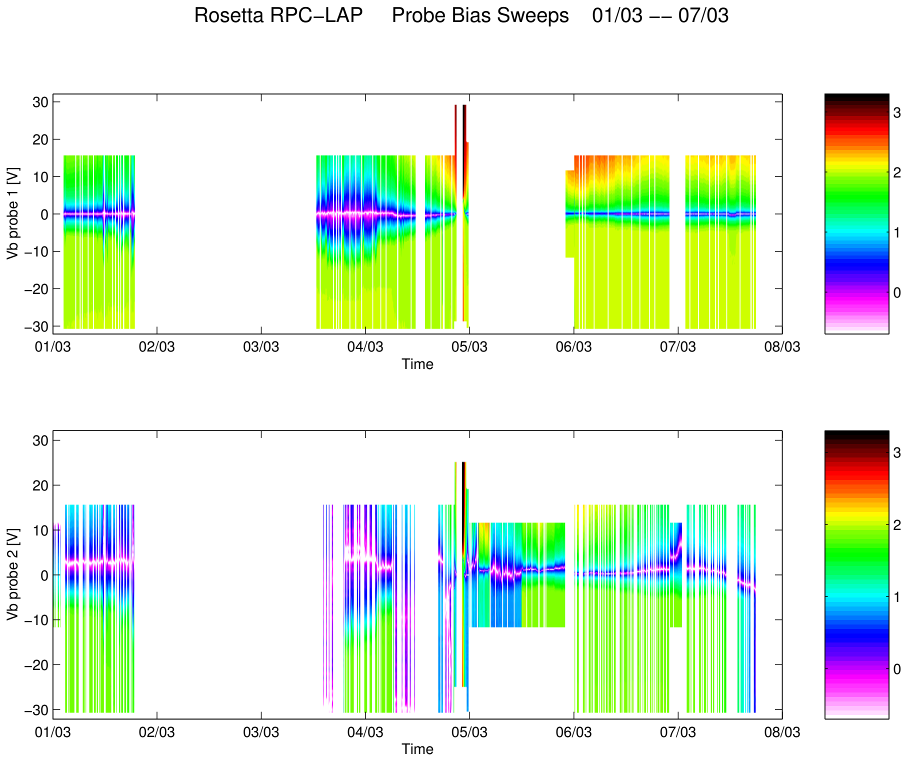


Figure 36: LAP sweep, current in logarithmic scale

In Figure 37 we compare the measured photo current (effectively the probe current for high negative bias voltage) and the estimated sunlit/eclipse information calculated from the attitude data (see Sections 4.6 and 5.2).

LAP 1: A fairly constant current of 80-90 nA suggests that probe 1 was well sunlit during the whole time span, which agrees with our estimations. After the time of closest approach the photocurrent seems to increase to nearly 100 nA. This is probably due to changed probe orientation with respect to the Sun and the boom on which it is mounted.

LAP 2: The photocurrent is considerably lower than for LAP 1 (60-70 nA). This can be explained by the fact that the booms, on which the probes are mounted, have a non-zero diameter and block the sunlight differently for LAP 1 and 2. The angle between the boom and the probe-Sun vector is more than 90° for LAP 1 during the flyby, but for LAP 2 the same angle varies from almost zero to about 45° allowing the boom to shadow part of the probe sensor from the Sun. The angles mentioned are estimations based on Figures 13 and 19. Generally, the abrupt jumps in the measured data to zero photocurrent seem to agree fairly well with the estimated periods of eclipse, with the exception of the hours around the time for closest approach. At that time the increase in plasma density makes it hard to define the photocurrent from the LAP data. The small jumps in photocurrent at midnight 6/3 and a day later could be explained by changes in HGA configuration, and the same is probably the case for the slow decrease during the later half of 7/3.

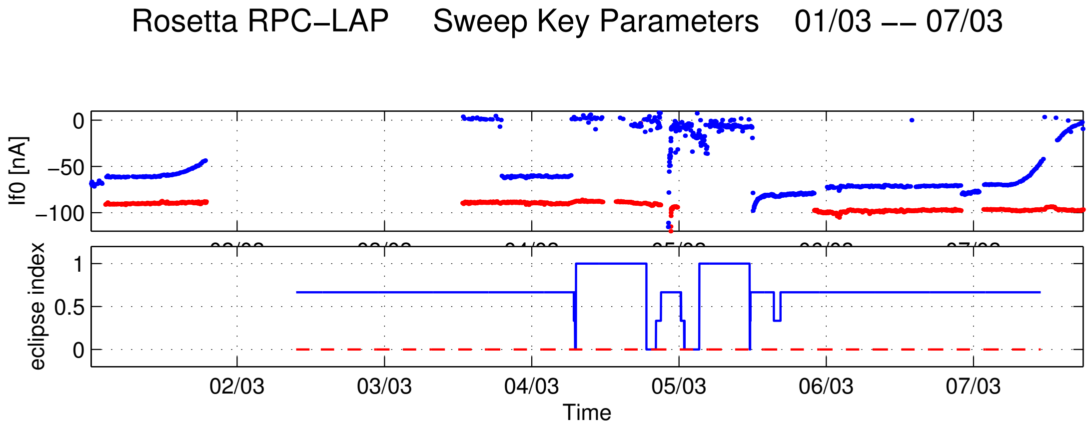


Figure 37: LAP photo current in comparison with the calculated eclipse and wake data. The missing data for the time before noon 2/3 is due to the short time span of the trajectory files. An extrapolation of the spacecraft position for this time gives the same values as for 3/3.

6.2 Data from the ”LAP dance” manoeuvre

A comparison between measured photocurrent and eclipse index was also done for the so-called ”LAP dance” event, on October 10 2004 02:00-14:30. During this period, Rosetta was rotated around its y axis (the solar paned axis) so that the LAP performance could be tested. A seen in Figure 38, the variations in the measured photocurrents agree well with what was expected from the calculated eclipse indecies.

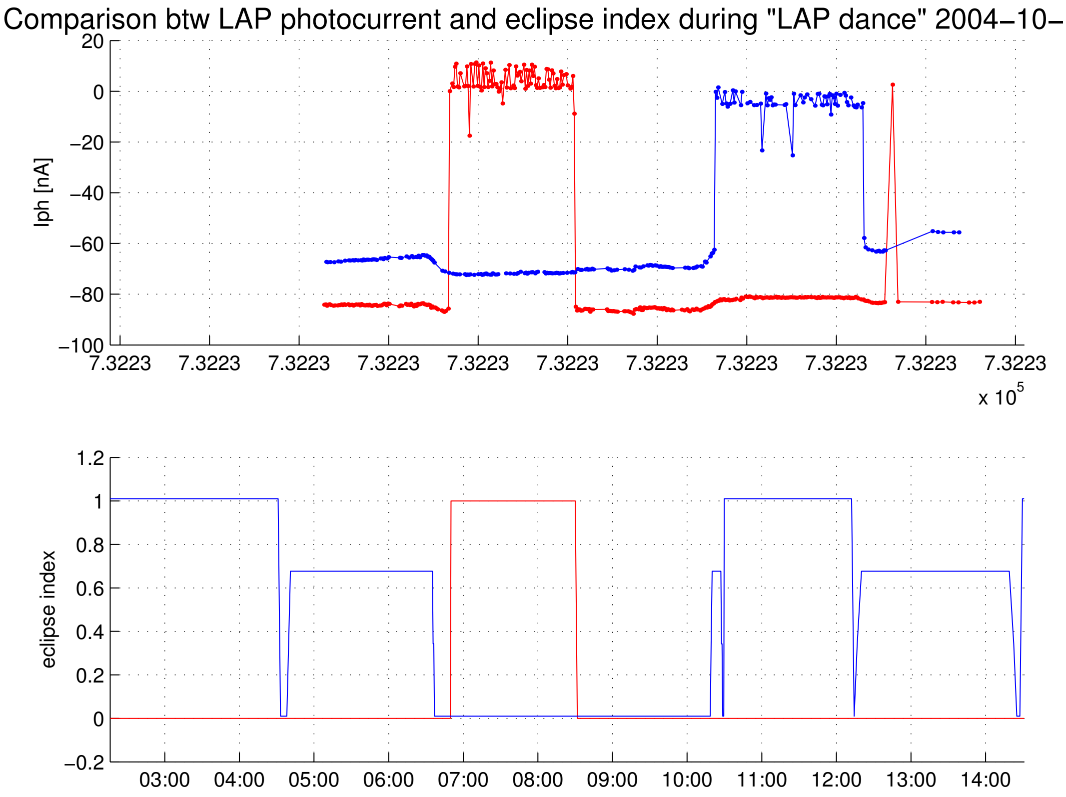


Figure 38: LAP photo current in comparison with the calculated eclipse and wake data for the ”LAP dance” manoeuvre. Red=LAP1, blue=LAP2.

7 Conclusion and discussion

The development of Matlab routines for handling the trajectory and attitude data provided by ESOC for the Rosetta flybys has helped the planning of LAP configurations for the 1st Earth flyby and also resulted in a useful tool for the three future planetary flybys. A lot can be done to improve further the Matlab routines, adapting the program for the Mars flyby occurring in 2007.

As for the radiation on Rosetta from the Van Allen radiation belts during the flyby, the results from SPENVIS showed that we could expect a total dose on the LAP electronics of less than 100 Rad. The total amount of trapped protons with energy above 30 MeV hitting Rosetta during the flyby was estimated to be in the order of $2 \cdot 10^7$ protons/cm^3. As this corresponds to a week of traveling in typical solar wind conditions, there was no reason to worry about the LAP electronics being damaged by the radiation dose and so it was decided to keep the electronics turned on during the flyby.

Using the Matlab routines and a box model of the Rosetta spacecraft we made a theoretical estimation of when the Langmuir probes were likely to be sunlit and when they would be shadowed by the spacecraft. A comparison with the measured data from the first Earth flyby, looking specifically at the photocurrent as a function of time, showed that with our model we could explain the main current variations to be the result of changes in spacecraft attitude and antenna position. From the model we could also explain why the photocurrent measured by LAP 1 was significantly larger than that of LAP 2. A similar comparison for the "LAP dance" event on October 10 2004 02:00-14:30 verified the model. The Matlab routines can also be used to estimate when the probes are likely to be in or out of the wake behind the spacecraft. No thorough analysis of the LAP data has been performed trying to see wake effects, but it could be an interesting issue for continued study.

Some future improvements of the Matlab routines would be:

- For the sun/eclipse routines, include a variable that defines the position of the high gain antenna. This needs information from ESOC about the HGA orientation.
- Include the positions of other RPC instruments (mainly MAG, ICA and IES), for calculating sunlit/eclipse index for those.
- Define a suitable coordinate system for the Mars flyby.

8 Acknowledgements

I would like to express my gratitude to a few people for their support and assistance while working on this project. My supervisor Anders Eriksson at the *Swedish Institute of Space Physics, Uppsala Division (IRF-U)*: Thank you for defining the project and providing me with invaluable information and thoughts. Your great amount of enthusiasm has encouraged me during the whole project. Thanks go also to Mats André (IRF-U) for initiating the project together with Anders. Daniel Heynderickx of the SPENVIS team at the Belgian Institute for Space Aeronomy is thanked for the help with introducing our Rosetta trajectory into the SPENVIS model package. I also would like to thank Jean-Gabriel Trotignon, Centre National de la Recherche Scientifique, Orléans, France, for the comments on my results of the trajectory calculations, Lennart Åhlén at IRF-U for information about radiation doses and Jorge Diaz del Rio Garcia at ESA for locating an error in my Matlab routines. For correcting my written English I thank my friend Anna Star (Otago University, Dunedin, New Zealand), my friend Dag Sehlin and Harley Thomas at IRF-U. Finally my colleagues Erik Engwall, Lisa Rosenqvist and Lars Norin at IRF-U are recognized for their help with providing literature and tips on writing in LaTeX.

References

[Behlke et al., 2000] Behlke, R., Sundkvist, D., and Tjulin, A. (2000). Langmuir probes, in S. Høymork (ed.). *Sensors and Instruments, IRF*, 2 ed:121–146.

[ESA, 2005a] ESA (2000-2005a). Rosetta Mission homepage. http://www.esa.int/SPECIALS/Rosetta/.

[ESA, 2001] ESA (2001). *Astrium GmbH ROSETTA Doc. RO-DSS-RP-1004*. European Space Agency.

[ESA, 2003a] ESA (2003a). *RO-ESC-IF-5003/MEX-ESC-IF-5003*. European Space Agency.

[ESA, 2003b] ESA (2003b). *ROSETTA/MARS EXPRESS Mission Control System (MCS) Data Delivery Interface Document DDID Doc. RO-ESC-IF-5003/MEX-ESC-IF-5003 Appendix H Issue 2*. European Space Agency.

[ESA, 2005b] ESA (2005b). *Rosetta Mission Operations Report, RO-ESC-RP-5018*. European Space Agency.

[Feynman et al., 1990] Feynman, J., Armstrong, T. P., and Silverman, S. (1990). Solar proton events during solar cycles 19, 20, and 21. *Solar Physics*, 126:385–401.

[Hapgood, 1992] Hapgood, M. A. (1992). Space physics coordinate transformations: A user guide. *Planetary and Space Science*, 40:71l–717.

[Kivelson and Russell, 1997] Kivelson, M. G. and Russell, C. R. (1997). *Introduction to Space Physics*. Cambridge University Press.

[Mott-Smith and Langmuir, 1926] Mott-Smith, H. M. and Langmuir, I. (1926). The theory of collectors in gaseous discharges. *Physical Review*, 28:727–763.

[Trotignon et al., 1999] Trotignon, J.-G., Boström, R., Burch, J. L., Glassmeier, K.-H., Lundin, R., Norberg, O., Balogh, A., Szegö, K., Musmann, G., Coates, A., Åhlén, L., Carr, C., Eriksson, A., Gibson, W., Kuhnke, F., Lundin, K., Michau, J.-L., and Szalai, S. (1999). The Rosetta Plasma Consortium: Technical realization and scientific aims. *Adv. Space Res.*, 24:1149–1158.

[Weisstein, 1999] Weisstein, E. W. (1999). ”Rotation Matrix” From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/RotationMatrix.html.

[Åhlén, 2005] Åhlén, L. (2005). Personal conversation with Lennart Åhlén.

A Appendix

A.1 File header for an ESOC trajectory data file

Rosetta trajectory data file (the last two columns missing)

```
ESOC_TOS_GFI_ORBIT_FILE_VERSION = 1.0


META_START
OBJECT_NAME          = ROSETTA
OBJECT_ID            = 226
CENTER_NAME          = EARTH
REF_FRAME            = EME2000
TIME_SYSTEM          = TDB
START_TIME           = 2005-03-02T09:36:59.97989711
STOP_TIME            = 2005-03-03T10:11:04.18299999
CREATION_DATE        = 2005-04-01T09:45:42
FILE_TYPE            = ORBIT FILE
VARIABLES_NUMBER     = 6
DERIVATIVES_FLAG     = 1
META_STOP

2005-03-02T09:36:59.97989711 -0.89504738140457869D+06  0.21986508254732937D+06  0.44337084295865148D+05  0.38770500975830942D+01
                              0.33497712843115389D+06 -0.75629846067945502D+05 -0.11971260360163962D+05  0.35363456884686263D-01
2005-03-02T10:06:21.32775342 -0.88821790883938421D+06  0.21832318914391313D+06  0.44093039345022698D+05  0.38777774880374025D+01
                              0.33503997496645682D+06 -0.75640254514640386D+05 -0.11971192825765180D+05  0.36000712227075189D-01
2005-03-02T10:35:24.15597085 -0.88145897224262427D+06  0.21679729609686023D+06  0.43851561031483965D+05  0.38785101563210387D+01
                              0.33510327750613459D+06 -0.75650849421387509D+05 -0.11971197514847427D+05  0.36645077048163652D-01
2005-03-02T11:04:08.68135411 -0.87476974862820120D+06  0.21528721435663907D+06  0.43612617867282490D+05  0.38792480630474633D+01
                              0.33516703264732257D+06 -0.75661629294097045D+05 -0.11971273779575346D+05  0.37296618233170105D-01
2005-03-02T11:32:35.11869977 -0.86814942267817853D+06  0.21379275669929801D+06  0.43376178700777709D+05  0.38799911696172478D+01
                              0.33523123705493018D+06 -0.75672592664464188D+05 -0.11971420983409960D+05  0.37955402907057283D-01
2005-03-02T12:00:43.68079281 -0.86159718675055879D+06  0.21231373772584379D+06  0.43142212715293026D+05  0.38807394382118900D+01
                              0.33529588746143377D+06 -0.75683738089877414D+05 -0.11971638501019930D+05  0.38621498441379111D-01
2005-03-02T12:28:34.57838830 -0.85511224094630359D+06  0.21084997387503955D+06  0.42910689429904727D+05  0.38814928317804749D+01
                              0.33536098066585127D+06 -0.75695064153171028D+05 -0.11971925718181643D+05  0.39294972455678118D-01
2005-03-02T12:56:08.02019506 -0.84869379316789599D+06  0.20940128343433773D+06  0.42681578699962185D+05  0.38822513140263895D+01
                              0.33542651353187626D+06 -0.75706569462165266D+05 -0.11972282031634792D+05  0.39975892822694509D-01
2005-03-02T13:23:24.21286175 -0.84234105916981993D+06  0.20796748654903247D+06  0.42454850717352645D+05  0.38830148493940952D+01
                              0.33549248298758629D+06 -0.75718252649589471D+05 -0.11972706849014588D+05  0.40664327666558073D-01
2005-03-02T13:50:23.36096475 -0.83605326260132017D+06  0.20654840522971339D+06  0.42230476010523402D+05  0.38837834030559808D+01
                              0.33555888602403528D+06 -0.75730112372726013D+05 -0.11973199588726222D+05  0.41360345365893968D-01
2005-03-02T14:17:05.66699832 -0.82982963504179660D+06  0.20514386335810117D+06  0.42008425444273780D+05  0.38845569408992886D+01
                              0.33562571969365136D+06 -0.75742147313001056D+05 -0.11973759679807268D+05  0.42064014557986223D-01
2005-03-02T14:43:31.33136638 -0.82366941602917935D+06  0.20375368669134221D+06  0.41788670219329084D+05  0.38853354295131464D+01
                              0.33569298110993655D+06 -0.75754356175893161D+05 -0.11974386561854169D+05  0.42775404136914080D-01
2005-03-02T15:09:40.55237630 -0.81757185308161762D+06  0.20237770286483821D+06  0.41571181871708075D+05  0.38861188361756933D+01
                              0.33576066744554136D+06 -0.75766737690435984D+05 -0.11975079684864926D+05  0.43494583259256328D-01
```

A.2 The Matlab routines - source code

A.2.1 Main program

go.m

```
% Rosetta Earth and Mars Flyby Modeling Tool - main program
%
% Coordinate systems (all geocentric):
% ------------------------------------
% GEI (Geocentric Equatorial Inertial): X-aries,          Z-geo (rot) north
% GEA (Geocentric Ecliptic Aries):      X-aries,          Z-ecliptic north
% GSE (Geocentric Solar Ecliptic):      X-earth-sun line, Z-ecliptic north   <-- The most useful!
% GEO (Geographic):                     X-lat=long=0,     Z-geo (rot) north
% ROS (Rosetta Inertial System)         X-High-gain ant.  Z-the side with the ICA instrument
% MEI (Mars-centric Equat. Inertial):   X-aries           Z-geo (rot) north
% MEA (Mars-centric Ecliptic Aries):    X-aries           Z-ecliptic north
% MSE (Mars-centric Solar Ecliptic):    X-mars-sun line   Z-ecliptic north
% HEI (Heliocentric Equat. Inertial):   X-aries           Z-geo (rot) north
% HEA (Heliocentric Ecliptic Aries):    X-aries           Z-ecliptic north


if (~exist('dontread') | dontread==0)
    clear all;
    read=1;                   % read the trajectory and attitude data from files
end

% DEFINITIONS & CONSTANTS
rad_sun=695000;                              % radius of the sun (km)
rad_earth=6371.2;                            % radius om earth (km)
rad_mars=3397;                               % radius of mars (km)
inclination=23.439291*pi/180;                % Earths inclination angle in rad
disp('Rosetta Planetary Swingby Modeling tool - Magnus Billvik, May 2005');
disp('Select ROSETTA event to load trajectory and attitude data.');
disp('1 - Earth flyby 2005');
disp('2 - Earth flyby 2007');
disp('3 - Earth flyby 2009');
disp('4 - Mars flyby 2007');
disp('5 - whole mission');
event=input('?');

% To transform the Rosetta dates to julian days, Matlabs "datenum" is used:
timediff=datenum([-4713 1 1 12 0 0])+327;    % julian days between JD=0 and the year 0 ...
                                             % ..with a 327 day correction for I don't know what!
                                             % Add this value to the 'datenum' value to get Julian Days!
if read==1
    readdata;
end

% juilan date for rosetta position data
jdr=datenum(dater(1,:),dater(2,:),dater(3,:),dater(4,:),dater(5,:),dater(6,:))-timediff;

% Here Rosetta's (GEI or MEI) coordinates are interpolated to match the time points
% for the coordinates of Sun and Moon (1 minute resolution). Also Mars' gea coords.

if event<=3
    rr_gei=zeros(3,length(jd)); % declare the variables to speed up
    vr_gei=zeros(3,length(jd));
    rr_gei(1,:) = interp1(jdr,RR(1,:),jd);      % interpolate the coordinates for every 1 minute
    rr_gei(2,:) = interp1(jdr,RR(2,:),jd);
    rr_gei(3,:) = interp1(jdr,RR(3,:),jd);
    vr_gei(1,:) = interp1(jdr,VR(1,:),jd);
    vr_gei(2,:) = interp1(jdr,VR(2,:),jd);
    vr_gei(3,:) = interp1(jdr,VR(3,:),jd);
    rs_gea=RS;
    if moon==1
        rm_gea=RMo;
        clear RMo;
    end

elseif event==4
    rr_mei=zeros(3,length(jd)); % declare the variables to speed up
    vr_mei=zeros(3,length(jd));
    rr_mei(1,:) = interp1(jdr,RR(1,:),jd);
    rr_mei(2,:) = interp1(jdr,RR(2,:),jd);
    rr_mei(3,:) = interp1(jdr,RR(3,:),jd);
    vr_mei(1,:) = interp1(jdr,VR(1,:),jd);
    vr_mei(2,:) = interp1(jdr,VR(2,:),jd);
    vr_mei(3,:) = interp1(jdr,VR(3,:),jd);
    rs_gea=RS;
    rma_gea = RMa;

elseif event==5
    dist_earth=1.496e8;
    dist_mars=2.28e8;

    if att==0
        rr_hei=RR;
        vr_hei=VR;
        rc_hei=RC;
        jd=datenum(dater')-timediff;    % define the time from the trajectory data
        number=length(rr_hei);
```

```

    elseif att==1  % if attitude data is loaded: interpolate the trajectory

        jda_tmp=datenum(datea(1,:),datea(2,:),datea(3,:),datea(4,:),datea(5,:),datea(6,:))-timediff;  % juilan date for rosetta attitude data

        % delete attitude elements with the same timestamp (it causes error in the interpolation)
        Q_mod(:,1)=Q(:,1);
        jda(1)=jda_tmp(1);
        j=2;
        for i=2:length(jda_tmp)
            jda(j)=jda_tmp(i);
            Q_mod(:,j)=Q(:,i);
            if jda(j)==jda(j-1)
                j=j-1;
            end
            j=j+1;
        end
        clear jda_tmp;

        % interpolate the trajectory for the time steps defined by attitude data
        rr_hei(1,:)=interp1(jdr',RR(1,:),jda);
        rr_hei(2,:)=interp1(jdr',RR(2,:),jda);
        rr_hei(3,:)=interp1(jdr',RR(3,:),jda);
        vr_hei(1,:)=interp1(jdr',VR(1,:),jda);
        vr_hei(2,:)=interp1(jdr',VR(2,:),jda);
        vr_hei(3,:)=interp1(jdr',VR(3,:),jda);
        rc_hei(1,:)=interp1(jdc',RC(1,:),jda);
        rc_hei(2,:)=interp1(jdc',RC(2,:),jda);
        rc_hei(3,:)=interp1(jdc',RC(3,:),jda);
        vc_hei(1,:)=interp1(jdc',VC(1,:),jda);
        vc_hei(2,:)=interp1(jdc',VC(2,:),jda);
        vc_hei(3,:)=interp1(jdc',VC(3,:),jda);

        number=length(rr_hei);

        % Interpolate the quaternions for every minute
        q1 = Q_mod(1,:);
        q2 = Q_mod(2,:);
        q3 = Q_mod(3,:);
        q4 = Q_mod(4,:);
        clear Q_mod;
%        keyboard;
%        q4(find(q4>0.6))=q4(find(q4>0.6))-0.9;

        % Define rosettas attitude vectors (from the quaternions) in GEI frame of referece for all times (3d matrix),
        % the Rosetta inertial x,y,z axis being the rows of Asc
        % Formula for converting quaternions to attitude vectors are taken from the ESA document file:
        % RO-ESC-IF-5003_2_0_DDID_Appendix_H_Data_Delivery_FD_Products_2003Oct23.pdf
        for i=1:number
            Asc_gei(1,:,i) = [q1(i)^2-q2(i)^2-q3(i)^2+q4(i)^2  2*(q1(i)*q2(i)+q3(i)*q4(i))      2*(q1(i)*q3(i)-q2(i)*q4(i))];
            Asc_gei(2,:,i) = [2*(q1(i)*q2(i)-q3(i)*q4(i))     -q1(i)^2+q2(i)^2-q3(i)^2+q4(i)^2  2*(q2(i)*q3(i)+q1(i)*q4(i))];
            Asc_gei(3,:,i) = [2*(q1(i)*q3(i)+q2(i)*q4(i))      2*(q2(i)*q3(i)-q1(i)*q4(i))     -q1(i)^2-q2(i)^2+q3(i)^2+q4(i)^2];
        end

        jd=jda;                 % define the time from the attitude data
%        clear q1 q2 q3 q4;
        clear jda jdr jdc;  % memory cleanup
    end

    istart=1;
    iend=number;

end


%clear RR VR RS RMa;
%clear jdmo jdr;


% ATIITUDE data - interpolate the quarternions and calculate the att. vectors (for flybys)
if att==1 & event<5

    jda_tmp=datenum(datea(1,:),datea(2,:),datea(3,:),datea(4,:),datea(5,:),datea(6,:))-timediff;  % juilan date for rosetta attitude data
    % delete attitude elements with the same timestamp (it causes error in the interpolation)
    Q_mod(:,1)=Q(:,1);
    jda(1)=jda_tmp(1);
    j=2;
    for i=2:length(jda_tmp)
        jda(j)=jda_tmp(i);
        Q_mod(:,j)=Q(:,i);
        if jda(j)==jda(j-1)
            j=j-1;
        end
        j=j+1;
    end
    clear jda_tmp;

    % Interpolate the quaternions for every minute
    q1 = interp1(jda,Q_mod(1,:),jd);
    q2 = interp1(jda,Q_mod(2,:),jd);
    q3 = interp1(jda,Q_mod(3,:),jd);
    q4 = interp1(jda,Q_mod(4,:),jd);
    clear Q_mod;
```

```

    % Define rosettas attitude vectors (from the quaternions) in GEI frame of referece for all times (3d matrix),
    % the Rosetta inertial x,y,z axis being the rows of Asc
    % Formula for converting quaternions to attitude vectors are taken from the ESA document file:
    % RO-ESC-IF-5003_2_0_DDID_Appendix_H_Data_Delivery_FD_Products_2003Oct23.pdf
    for i=1:number
        Asc_gei(1,:,i) = [q1(i)^2-q2(i)^2-q3(i)^2+q4(i)^2  2*(q1(i)*q2(i)+q3(i)*q4(i))      2*(q1(i)*q3(i)-q2(i)*q4(i))];
        Asc_gei(2,:,i) = [2*(q1(i)*q2(i)-q3(i)*q4(i))     -q1(i)^2+q2(i)^2-q3(i)^2+q4(i)^2  2*(q2(i)*q3(i)+q1(i)*q4(i))];
        Asc_gei(3,:,i) = [2*(q1(i)*q3(i)+q2(i)*q4(i))      2*(q2(i)*q3(i)-q1(i)*q4(i))     -q1(i)^2-q2(i)^2+q3(i)^2+q4(i)^2];
    end

    clear q1 q2 q3 q4; % memory cleanup
    clear jda;

end

coordtransform;


% define index variables for the interesting range of the flyby
% find the first and last index with traj data that is not NaN
if event<4 % for the earth flybys
    i=1;
    while isnan(rr_gea(1,i))
        i=i+1;
    end
    istart=i;
    i=number;
    while isnan(rr_gea(1,i))
        i=i-1;
    end
    iend=i;
    mindist=min(distance);
    iclosest=find(mindist==distance);
elseif event==4 % for the mars flyby
    i=1;
    while isnan(rr_mea(1,i))
        i=i+1;
    end
    istart=i;
    i=number;
    while isnan(rr_mea(1,i))
        i=i-1;
    end
    iend=i;
    mindist=min(distance);
    iclosest=find(mindist==distance);

end

disp('Calculations done!');

vis;
```

A.2.2 Program for reading the data files

readdata.m

```
% Reads all the (raw) trajectory data for Rosetta and the Sun and Moon from textfiles
% and saves the coordinates and velocity vectors in the arrays
% RR, RS, RM and VR, VS, VM respectively.

%function readtrajdata
%global number nlstart date jd mjd date dater inclination radiuse theta fb;
%global RR VR RM VM RS VS;

% Rosettas trajecotry, given in Geocentric Equatorial Aries coordinates

% standard start and end dates (outside the trajectory data time span)
startdate=[2000 1 1 0 0 0]’;
enddate=[2020 1 1 0 0 0]’;

switch event
case 1
    [RR, VR, dater] = readros(’traj_r_e1.txt’, startdate, enddate);
    [RS, dates, jds] = readjpltraj(’traj_s_e1.txt’);
    [RMo, datem, jdmo] = readjpltraj(’traj_m_e1.txt’);
    [Q, datea] = readatt(’attitude.txt’, dater(:,1), dater(:,length(dater)));
    savename=’efb1_gse.txt’;
    att=1;
    moon=1;
    tit=’Rosetta Earth flyby #1 ’;
    radius=rad_earth;
    % rename and clear variables
    jd=jds;
    time=dates;
    number=length(time);
    clear jdm jds datem dates;

case 2
    [RR, VR, dater] = readros(’traj_r_e2.txt’, startdate, enddate);
    [RS, dates, jds] = readjpltraj(’traj_s_e2.txt’);
    [RMo, datem, jdmo] = readjpltraj(’traj_m_e2.txt’);
    savename=’efb2_gse.txt’;
    att=0;
    moon=1;
    tit=’Rosetta Earth flyby #2 ’;
    radius=rad_earth;
    % rename and clear variables
    jd=jds;
    time=dates;
    number=length(time);
    clear jdm jds datem dates;

case 3
    [RR, VR, dater] = readros(’traj_r_e3.txt’, startdate, enddate);
    [RS, dates, jds] = readjpltraj(’traj_s_e3.txt’);
    [RMo, datem, jdmo] = readjpltraj(’traj_m_e3.txt’);
    savename=’efb3_gse.txt’;
    att=0;
    moon=1;
    tit=’Rosetta Earth flyby #3 ’;
    radius=rad_earth;
    % rename and clear variables
    jd=jds;
    time=dates;
    number=length(time);
    clear jdm jds datem dates;

case 4
    [RR, VR, dater] = readros(’traj_r_m.txt’, startdate, enddate);
    [RS, dates, jds] = readjpltraj(’traj_s_m_earth.txt’);
    [RMa, datem, jdma] = readjpltraj(’traj_mars_m_earth.txt’);
    savename=’mfb_gse.txt’;
    att=0;
    moon=0;
    tit=’Rosetta Mars flyby ’;
    radius=rad_mars;
    % rename and clear variables
    jd=jds;
    time=dates;
    number=length(time);
    clear jdm jds datem dates;

case 5
    startdate_tmp=input(’Attitude start date [yyyy mm dd HH MM SS] (0 to ignore attitude):’);
    enddate_tmp=input(’Attitude end date [yyyy mm dd HH MM SS] (0 to ignore attitude):’);
    att=0;
    if max(startdate_tmp~=0) & max(enddate_tmp~=0)
        startdate=startdate_tmp’;
        enddate=enddate_tmp’;
        att=1;
        [Q, datea] = readatt(’attitude.txt’, startdate, enddate);
    end
    [RR, VR, dater] = readros(’traj_r_whole.txt’, startdate, enddate);
    [RC, VC, datec] = readros(’traj_comet.txt’, startdate, enddate);

```

```
    savename='wholemission.txt';
    moon=0;
    tit='ROSETTA mission ';
    radius=rad_sun;
    jdr=datenum(dater')-timediff;
    jdc=datenum(datec')-timediff;
    % rename and clear variables
end

```

A.2.3 Coordinate transformation program

coordtransform.m

```
% Transforms the trajectory data between different coordinate systems
%
% Coordinate systems:
% ------------------------------------
% GEI (Geocentric Equatorial Inertial): X-aries,          Z-geo (rot) north
% GEA (Geocentric Ecliptic Aries):      X-aries,          Z-ecliptic north
% GSE (Geocentric Solar Ecliptic):      X-earth-sun line, Z-ecliptic north   <-- The most useful!
% GEO (Geographic):                     X-lat=long=0,     Z-geo (rot) north
% ROS (Rosetta Inertial System)         X-High-gain ant.  Z-the side with the ICA instrument
% MEI (Mars-centric Equat. Inertial):   X-aries           Z-geo (rot) north
% MEA (Mars-centric Ecliptic Aries):    X-aries           Z-ecliptic north
% MSE (Mars-centric Solar Ecliptic):    X-mars-sun line   Z-ecliptic north
% HEI (Heliocentric Equat. Inertial):   X-aries           Z-geo (rot) north
% HEA (Heliocentric Ecliptic Aries):    X-aries           Z-ecliptic north


% For the Earth flybys
if event<=3

    % SUN & MOON traj from GEA(J2000) to GEI
    disp('- gea -> gei');
    rs_gei=gea2gei(rs_gea, inclination);  % sun's coord
    if moon==1
        rm_gei=gea2gei(rm_gea, inclination);  % moons coord
    end


    % ROSETTA traj and ATTITUDE from GEI to GEA(J2000)
    disp('- gei -> gea');
    rr_gea=gei2gea(rr_gei, inclination);    % rosettas coord
    vr_gea=gei2gea(vr_gei, inclination);    % rosettas velocity
    if att==1
        Asc_gea(1,:,:)=gei2gea_a(Asc_gei(1,:,:),inclination);
        Asc_gea(2,:,:)=gei2gea_a(Asc_gei(2,:,:),inclination);
        Asc_gea(3,:,:)=gei2gea_a(Asc_gei(3,:,:),inclination);
    end

    % GEA(J2000) to GSE
    disp('- gea -> gse');
    rr_gse=gea2gse(rr_gea, rs_gea);  % rosettas coord
    vr_gse=gea2gse(vr_gea, rs_gea);  % rosettas velocity
    rs_gse=gea2gse(rs_gea, rs_gea);  % suns coord
    rm_gse=gea2gse(rm_gea, rs_gea);  % moons coord
    if att==1
        Asc_gse(1,:,:)=gea2gse_a(Asc_gea(1,:,:), rs_gea);    % rosettas x-axis
        Asc_gse(2,:,:)=gea2gse_a(Asc_gea(2,:,:), rs_gea);    % rosettas y-axis
        Asc_gse(3,:,:)=gea2gse_a(Asc_gea(3,:,:), rs_gea);    % rosettas z-axis
    end

    if att==1
        % GEA(J2000) to ROS
        disp('- gea -> ros (xyz)');
        rv_ros=zeros(3,number); % declare the variables to speed up the loop!
        re_ros=zeros(3,number);
        rs_ros=zeros(3,number);
        rm_ros=zeros(3,number);
        % translation and rotation to rosetta's ref. frame
        % by def. Asc_gea(:,:,i) is the rotation matrix from GEA to ROS
        for i=1:number
            vr_ros(:,i)=Asc_gea(:,:,i)*vr_gea(:,i);                % rosettas velocity vector
            re_ros(:,i)=Asc_gea(:,:,i)*(-rr_gea(:,i));             % earth's position
            rs_ros(:,i)=Asc_gea(:,:,i)*(rs_gea(:,i)-rr_gea(:,i));  % sun's position
            rm_ros(:,i)=Asc_gea(:,:,i)*(rm_gea(:,i)-rr_gea(:,i));  % moon's position
        end


        % ROS(x,y,z) to ROS(rho, phi, theta in km and degrees resp.)
        disp('- ros (xyz) -> ros (rho,phi,theta)');
        sphi=zeros(1,number); % declare the variables to speed up the loop!
        stheta=zeros(1,number);
        srho=zeros(1,number);
        mphi=zeros(1,number);
        mtheta=zeros(1,number);
        mrho=zeros(1,number);
        vphi=zeros(1,number);
        vtheta=zeros(1,number);
        vrho=zeros(1,number);
        for i=1:number
            % use this when using visualize2 (zx plane instead of xy-plane)
            [sphi(i), stheta(i), srho(i)]=cart2sphere(rs_ros(1,i), rs_ros(2,i), rs_ros(3,i));
            [mphi(i), mtheta(i), mrho(i)]=cart2sphere(rm_ros(1,i), rm_ros(2,i), rm_ros(3,i));
            [vphi(i), vtheta(i), vrho(i)]=cart2sphere(vr_ros(1,i), vr_ros(2,i), vr_ros(3,i));
        end
        sphi=sphi*180/pi;   % to make it degrees, not radians
        mphi=mphi*180/pi;
        vphi=vphi*180/pi;
        stheta=stheta*180/pi;
        mtheta=mtheta*180/pi;
        vtheta=vtheta*180/pi;
```

```
    end


    % Transformation from GEI to GEO
    disp('- gei -> geo (xyz)');
    rr_geo=gei2geo(rr_gei, jd);  % rosettas coord
    rs_geo=gei2geo(rs_gei, jd);  % suns coord
    rm_geo=gei2geo(rm_gei, jd);  % moons coord


    % calculate the geographical coordinates (lat, long)
    disp('- geo (xyz) -> geo (long,lat)');
    [longitude, latitude]=geo2longlat(rr_geo);

    distance=sqrt(rr_gse(1,:).^2+rr_gse(2,:).^2+rr_gse(3,:).^2);


% For the Mars flyby

elseif event==4

    % translate Sun's pos to mars-centric MEA system
    rs_mea=gea2mea(rs_gea, rma_gea);

    % transform Sun's pos from MEA to MSE
    rs_mse=gea2gse(rs_mea, rs_mea);

    % transform Rosetta's pos from MEI to MEA
    rr_mea=gei2gea(rr_mei, inclination);

    % transform Rosetta's pos from MEA to MSE
    rr_mse=gea2gse(rr_mea, rs_mea);

    distance=sqrt(rr_mse(1,:).^2+rr_mse(2,:).^2+rr_mse(3,:).^2);




else    % For the whole mission
    disp('- hei -> hea');
    rr_hea=gei2gea(rr_hei, inclination);
    vr_hea=gei2gea(vr_hei, inclination);
    rc_hea=gei2gea(rc_hei, inclination);

    if att==1
        Asc_gea(1,:,:)=gei2gea_a(Asc_gei(1,:,:),inclination);
        Asc_gea(2,:,:)=gei2gea_a(Asc_gei(2,:,:),inclination);
        Asc_gea(3,:,:)=gei2gea_a(Asc_gei(3,:,:),inclination);

        % HEA(J2000) to ROS
        disp('- hea -> ros (xyz)');

        rs_ros=zeros(3,number);

        % translation and rotation to rosetta's ref. frame
        % by def. Asc_gea(:,:,i) is the rotation matrix from HEA/GEA to ROS
        for i=1:number
            vr_ros(:,i)=Asc_gea(:,:,i)*vr_hea(:,i);                % rosettas velocity vector
            rs_ros(:,i)=Asc_gea(:,:,i)*(-rr_hea(:,i));             % sun's position
        end


        % ROS(x,y,z) to ROS(rho, phi, theta in km and degrees resp.)
        disp('- ros (xyz) -> ros (rho,phi,theta)');
        sphi=zeros(1,number); % declare the variables to speed up the loop!
        stheta=zeros(1,number);
        srho=zeros(1,number);
        vphi=zeros(1,number);
        vtheta=zeros(1,number);
        vrho=zeros(1,number);
        for i=1:number
            [sphi(i), stheta(i), srho(i)]=cart2sphere(rs_ros(1,i), rs_ros(2,i), rs_ros(3,i));
            [vphi(i), vtheta(i), vrho(i)]=cart2sphere(vr_ros(1,i), vr_ros(2,i), vr_ros(3,i));
        end
        sphi=sphi*180/pi;   % to make it degrees, not radians
        vphi=vphi*180/pi;
        stheta=stheta*180/pi;
        vtheta=vtheta*180/pi;
    end
end
```

A.2.4 Program for visualizing

vis.m

```
disp(’Select visualization type/option:’);
disp(’a - distance plot’);
disp(’b - 3D GSE plot’);
disp(’c - 2D GSE plots’);
disp(’d - 3D GSE plot with s/c attitude’);
disp(’e - 2D GSE plots with s/c attitude’);
disp(’f - GEO plot with world map (only earth flybys)’);
disp(’g - LAP: sun and velocity vector’);
disp(’h - LAP: make image file’);
disp(’i - LAP: Calculate "sunlit/eclipse and wake/non wake" from image file’);
disp(’j - LAP: Plot "sunlit/eclipse and wake/non wake" vs. time’);
disp(’k - Whole mission plot’);
disp(’l - set current time span’);
disp(’m - dump trajectory data file’);
disp(’n - dump eclipse/wake data file’);
disp(’o - LAP: plot measured photo-current data in lap plot’);
item=input(’?’,’s’);

if isempty(get(0, ’CurrentFigure’))
    fign=1;
else
    fign=get(0, ’CurrentFigure’)+1;
end

windowsize=[0 0 1275 945]; % for maximizing the window size (probably different for different computers...)

if event<4
    x=rr_gse(1,:)/radius;
    y=rr_gse(2,:)/radius;
    z=rr_gse(3,:)/radius;
    mx=rm_gse(1,:)/radius;
    my=rm_gse(2,:)/radius;
    mz=rm_gse(3,:)/radius;
    index=[istart:iend];
    indexx=[istart:60:iend];
    coordsyst=’ GSE’;
elseif event==4
    x=rr_mse(1,:)/radius;
    y=rr_mse(2,:)/radius;
    z=rr_mse(3,:)/radius;
    index=[istart:iend];
    indexx=[istart:60:iend];
    coordsyst=’ MSE’;
elseif event==5
    x=rr_hea(1,:)/radius;
    y=rr_hea(2,:)/radius;
    z=rr_hea(3,:)/radius;
    index=[istart:iend];
%    indexx=[istart:60:iend];
    coordsyst=’ HEA’;
end



switch item

% Altitude plot
case ’a’
    figure(fign)
    set(gcf,’Position’, windowsize);
    plot(jd(index)+timediff, (distance(index)-radius));
    hold on;
    plot([jd(istart) jd(iend)]+timediff, [1 1]*(distance(iclosest)-radius), ’k--’);
    ax=axis
    axis([ax(1) ax(2) 0 ax(4)]);
    ylabel(’altitude from planetary surface (km)’);
    titlestr=strcat(tit, ’ - Altitude plot (’,datestr(jd(istart)+timediff,20), ’@’,...
        datestr(jd(istart)+timediff,15), ’-’,datestr(jd(iend)+timediff,20),’@’,...
        datestr(jd(iend)+timediff,15),’)’);
    title(titlestr);
    datetick(’x’, 15);


% Plot 3D path
case ’b’
    figure(fign)
    set(gcf,’Position’, windowsize);
    plot3(x(index),y(index),z(index));
    hold on;
    plot3(x(istart),y(istart),z(istart),’*’);
    if moon==1
        plot3(mx(index),my(index),mz(index),’y’);    % moon
        plot3(mx(istart), my(istart), mz(istart), ’yo’); % plot a ring at the start position
    end
    sphere(1,0);
    axis(’equal’);
    grid on;

    splot3(x(index),y(index),z(index),axis)
```

```
    if moon==1
        splot3(mx(index),my(index),mz(index),axis);
    end
    ssphere(1,axis);

    xlabel('x (planetary radii) - towards sun');
    ylabel('y (planetary radii)');
    zlabel('z (planetary radii)');
    titlestr=strcat(tit, ' - 3D ', coordsyst,' plot (',datestr(jd(istart)+timediff,20),...
        '@', datestr(jd(istart)+timediff,15), '-',datestr(jd(iend)+timediff,20),'@',...
        datestr(jd(iend)+timediff,15),')');
    title(titlestr);
    if moon==1
        legend('rosetta path', 'start', 'moon path', 'start');
    else
         legend('rosetta path', 'start');
    end
    axis('equal');

% Plot 2D projections
case 'c'
    figure(fign)
    set(gcf,'Position', windowsize);

    subplot(2,2,1)
    plot(x(index),y(index));            % rosetta
    hold on;
    plot(x(istart), y(istart), '*');    % plot a ring at the start position
    if moon==1
        plot(mx(index),my(index), 'y');     % moon
        plot(mx(istart), my(istart), 'y*'); % plot a ring at the start position
    end
    circle(0,0,1,'k');                  % earth
    axis('equal');
    grid on;
    xlabel('x (planet radii)');
    ylabel('y (planet radii)');
    title('X/Y');

    subplot(2,2,2)
    plot(x(index),z(index));            % rosetta
    hold on;
    plot(x(istart), z(istart), '*');    % plot a ring at the start position
    if moon==1
        plot(mx(index),mz(index),'y');      % moon
        plot(mx(istart), mz(istart), 'y*'); % plot a ring at the start position
    end
    circle(0,0,1,'k');                  % earth
    axis('equal');
    grid on;
    xlabel('x (planet radii)');
    ylabel('z (planet radii)');
    title('X/Z');

    subplot(2,2,3)
    plot(y(index),z(index));            % rosetta
    hold on;
    plot(y(istart), z(istart), '*');    % plot a ring at the start position
    if moon==1
        plot(my(index),mz(index),'y');      % moon
        plot(my(istart), mz(istart), 'y*'); % plot a ring at the start position
    end
    circle(0,0,1,'k');                  % earth
    axis('equal');
    grid on;
    xlabel('y (planet radii)');
    ylabel('z (planet radii)');
    title('Y/Z');

    subplot(2,2,4)
%    plot(0,0);
%    plot(0,0,'*');
    legend('path', 'start');

    suptitle(strcat(tit, ' 2D', coordsyst,' plots (',datestr(jd(istart)+timediff,20), '@',...
        datestr(jd(istart)+timediff,15), '-',datestr(jd(iend)+timediff,20),'@',...
        datestr(jd(iend)+timediff,15),') [printed ', datestr(now), ']'));

% Plot 3D projections and attitude vectors
case 'd'
    figure(fign)
    set(gcf,'Position', windowsize);

    % following is the coordinates for a square representing rosetta (only to make the viewing of the xyz axis easier)
    ros3dmodela_sc=0.5*[ 1 0 1 ; 1 1 1 ; 1 1 0 ; 1 0 0 ; 0 0 1 ; 0 1 1 ; 0 1 0 ; 0 0 0 ; 1 0 1 ; 1 1 1 ; 1 1 0 ; 1 0 0 ]';
    ros3dmodelb_sc=0.5*[ 1 1 1 ; 1 1 0 ; 1 0 0 ; 1 0 1 ; 0 1 1 ; 0 1 0 ; 0 0 0 ; 0 0 1 ; 0 0 1 ; 0 1 1 ; 0 1 0 ; 0 0 0 ]';
    x1=Asc_gse(1,1,:); x2=Asc_gse(1,2,:); x3=Asc_gse(1,3,:); % xyz gse components of the rosetta's x-axis
    y1=Asc_gse(2,1,:); y2=Asc_gse(2,2,:); y3=Asc_gse(2,3,:); % xyz gse components of the rosetta's y-axis
    z1=Asc_gse(3,1,:); z2=Asc_gse(3,2,:); z3=Asc_gse(3,3,:); % xyz gse components of the rosetta's z-axis

    plot3(x(index),y(index),z(index));       % rosetta
    hold on;
    if moon==1
```

```
        plot3(mx(index),my(index),mz(index),'y');  % moon
        plot3(mx(istart), my(istart), mz(istart), 'yo');            % plot a ring at the start position
        splot3(mx(index),my(index),mz(index),axis);
    end
    sphere(1,0);            % earth
    splot3(x(index),y(index),z(index),axis);                    % gray 2d-projections on the sides
    ssphere(1,axis);                                            % earth
    plot3(x(istart), y(istart), z(istart), 'o');                % a ring at the start position

    % plot the 3d cube
    for i=1:length(indexx)
        a=inv(Asc_gse(:,:,indexx(i)))*ros3dmodela_sc; % transform to GSE ref frame
        b=inv(Asc_gse(:,:,indexx(i)))*ros3dmodelb_sc;
        for l=1:12
            a(1,l)=a(1,l)+x(indexx(i)); % translation to earth centered origo
            a(2,l)=a(2,l)+y(indexx(i));
            a(3,l)=a(3,l)+z(indexx(i));
            b(1,l)=b(1,l)+x(indexx(i));
            b(2,l)=b(2,l)+y(indexx(i));
            b(3,l)=b(3,l)+z(indexx(i));
            plot3([a(1,l) b(1,l)], [a(2,l) b(2,l)], [a(3,l) b(3,l)], 'color', [0.7 0.7 0.7]);
        end
    end
    % plot rosettas x,y and z axis in rgb colours
    point3d(x(indexx), y(indexx), z(indexx), x1(indexx), x2(indexx), x3(indexx), 3, 'r');
    point3d(x(indexx), y(indexx), z(indexx), y1(indexx), y2(indexx), y3(indexx), 3, 'g');
    point3d(x(indexx), y(indexx), z(indexx), z1(indexx), z2(indexx), z3(indexx), 3, 'b');

    axis('equal');
    grid on;
    xlabel('x (planet radii)');
    ylabel('y (planet radii)');
    zlabel('z (planet radii)');
    titlestr=strcat(tit, ' - 3D ', coordsyst,' plot (',datestr(jd(istart)+timediff,20),...
        '@', datestr(jd(istart)+timediff,15), '-',datestr(jd(iend)+timediff,20),'@',...
        datestr(jd(iend)+timediff,15),')');
    title(titlestr);
    legend('rosetta trajectory', 'moon trajectory');

    clear ros3dmodela_sc ros3dmodelb_sc x1 x2 x3 y1 y2 y3 z1 z2 z3 a b;


% Plot 2D path and attitude vectors
case 'e'

    x1=Asc_gse(1,1,:); x2=Asc_gse(1,2,:); x3=Asc_gse(1,3,:); % xyz gse components of the rosetta's x-axis
    y1=Asc_gse(2,1,:); y2=Asc_gse(2,2,:); y3=Asc_gse(2,3,:); % xyz gse components of the rosetta's y-axis
    z1=Asc_gse(3,1,:); z2=Asc_gse(3,2,:); z3=Asc_gse(3,3,:); % xyz gse components of the rosetta's z-axis

    figure(fign)
    set(gcf,'Position', windowsize);

    subplot(2,2,1)
    plot(x(index),y(index));        % rosetta
    hold on;
    plot(x(istart), y(istart), 'o');    % plot a ring at the start position
    if moon==1
        plot(mx(index),my(index),'y');      % moon
        plot(mx(istart), my(istart), 'yo'); % plot a ring at the start position
    end
    circle(0,0,1,'k');                  % earth
    % plot rosettas x,y and z axis in rgb colours
    point(x(indexx), y(indexx), x1(indexx), x2(indexx), 4, 'r');
    point(x(indexx), y(indexx), y1(indexx), y2(indexx), 4, 'g');
    point(x(indexx), y(indexx), z1(indexx), z2(indexx), 4, 'b');
    axis('equal');
    grid on;
    xlabel('x (planet radii)');
    ylabel('y (planet radii)');
    title('X/Y');

    subplot(2,2,2)
    plot(x(index),z(index));        % rosetta
    hold on;
    plot(x(istart), z(istart), 'o');    % plot a ring at the start position
    if moon==1
        plot(mx(index),mz(index),'y');      % moon
        plot(mx(istart), mz(istart), 'yo'); % plot a ring at the start position
    end
    circle(0,0,1,'k');                  % earth
    % plot rosettas x,y and z axis in rgb colours
    point(x(indexx), z(indexx), x1(indexx), x3(indexx), 4, 'r');
    point(x(indexx), z(indexx), y1(indexx), y3(indexx), 4, 'g');
    point(x(indexx), z(indexx), z1(indexx), z3(indexx), 4, 'b');
    axis('equal');
    grid on;
    xlabel('x (planet radii)');
    ylabel('z (planet radii)');
    title('X/Z');


    subplot(2,2,3)
    plot(y(index),z(index));        % rosetta
```

```
    hold on;
    plot(y(istart), z(istart), ’o’);    % plot a ring at the start position
    if moon==1
        plot(my(index),mz(index),’y’);      % moon
        plot(my(istart), mz(istart), ’yo’); % plot a ring at the start position
    end
    circle(0,0,1,’k’);                  % earth
    % plot rosettas x,y and z axis in rgb colours
    point(y(indexx), z(indexx), x2(indexx), x3(indexx), 4, ’r’);
    point(y(indexx), z(indexx), y2(indexx), y3(indexx), 4, ’g’);
    point(y(indexx), z(indexx), z2(indexx), z3(indexx), 4, ’b’);
    axis(’equal’);
    grid on;
    xlabel(’y (planet radii)’);
    ylabel(’z (planet radii)’);
    title(’Y/Z’);

    suptitle(strcat(tit, ’ 2D’, coordsyst,’ plots with attitude (’,datestr(jd(istart)+timediff,20),...
        ’@’, datestr(jd(istart)+timediff,15), ’-’,datestr(jd(iend)+timediff,20),’@’,...
        datestr(jd(iend)+timediff,15),’)’));
    clear x1 x2 x3 y1 y2 y3 z1 z2 z3;


% Plot Rosetta’s path on a World map
case ’f’
    figure(fign)
    set(gcf,’Position’, windowsize);
    [world, col]=imread(’world_blueblack.png’,’png’);
    newplot;
    set(gca,’YDir’,’normal’);
    %image([-10 350], [85 -60], world, ’YData’, [85 -60]), colormap(col);
    image([0 360], [90 -90], world, ’YData’, [90 -90]), colormap(col);
    set(gca,’YDir’,’normal’);
    hold on;
    plot(longitude(istart:iend),latitude(istart:iend),’r.’);    % the s/c path
    plot(longitude(istart),latitude(istart),’b.’);              % the start of the data

    temp{1,1}=datestr(jd(iclosest)+timediff);
    temp{2,1}=strcat(’lat=’, num2str(latitude(iclosest),’%3.1f’), ’\circ’);
    temp{3,1}=strcat(’long=’, num2str(longitude(iclosest),’%3.1f’), ’\circ’);
    temp{4,1}=strcat(’alt=’, num2str(round(distance(iclosest)-radius),’%4.0f’), ’ km’);
    h=text(longitude(iclosest)-0.4, latitude(iclosest)+0.4, temp, ’FontWeight’, ’bold’);
    set(h, ’Color’, ’w’);
    h=text(longitude(iclosest), latitude(iclosest), temp, ’FontWeight’, ’bold’);
    set(h, ’Color’, ’b’);

    temp{1,1}=datestr(jd(istart)+timediff);
    temp{2,1}=strcat(’lat=’, num2str(latitude(istart),’%3.1f’), ’\circ’);
    temp{3,1}=strcat(’long=’, num2str(longitude(istart),’%3.1f’), ’\circ’);
    temp{4,1}=strcat(’alt=’, num2str(round(distance(istart)-radius),’%4.0f’), ’ km’);
    h=text(longitude(istart)-0.4, latitude(istart)+0.4,temp, ’FontWeight’, ’bold’);
    set(h, ’Color’, ’w’);
    h=text(longitude(istart), latitude(istart),temp, ’FontWeight’, ’bold’);
    set(h, ’Color’, ’b’);

    plot(longitude(istart), latitude(istart),’bo’);

    xlabel(’longitude’);
    ylabel(’latitude’);
    titlestr=strcat(tit,’ - GEO long/lat plot (’,datestr(jd(istart)+timediff,20), ...
        ’@’, datestr(jd(istart)+timediff,15), ’-’,datestr(jd(iend)+timediff,20),’@’,...
        datestr(jd(iend)+timediff,15),’)’);
    title(titlestr);
    axis([0 360 -90 90]);

    clear temp h;

% Plot sun vector or velocity vector as seen from LAP
case ’g’
    proben=input(’LAP probe number: ’);
    temp=input(’Sun vector (1), velocity vector (2) or none (3)’);
    relradius=input(’High-gain antenna relative radius (1=100%): ’);
    dontplot=0;

    switch temp
    case 1
        phi=sphi;
        theta=stheta;
        titlestr=strcat(tit,’ - Movement of the Sun as seen from LAP probe #’, num2str(proben),...
            ’ (’,datestr(jd(istart)+timediff,20), ’@’, datestr(jd(istart)+timediff,15), ’-’,...
            datestr(jd(iend)+timediff,20),’@’,datestr(jd(iend)+timediff,15),’)’);
    case 2
        phi=vphi;
        theta=vtheta;
        titlestr=strcat(tit,’ - Movement of Rosettas velocity vector as seen from LAP probe #’, ...
            num2str(proben), ’ (’,datestr(jd(istart)+timediff,20), ’@’, datestr(jd(istart)+timediff,15),...
            ’-’,datestr(jd(iend)+timediff,20),’@’,datestr(jd(iend)+timediff,15),’)’);
    case 3
        dontplot=1;
        titlestr=strcat(’The Rosetta spacecraft as seen from LAP probe #’, num2str(proben));
    end

    % to make the plots look fine, take away the point before the "jump" (but save the orig. values)
```

```
    figure(fign)
    set(gcf,'Position', windowsize);

    if dontplot==0
        phi_save=phi;
        theta_save=theta;
        itmp=find(abs(diff(phi))>300);
        phi(itmp)=NaN;
        theta(itmp)=NaN;

        plot(phi(index), theta(index), 'r');
        hold on;
        plot(phi(istart), theta(istart), 'r*');
    end
    drawrosetta(proben, relradius);
    if dontplot==0
        plot(phi(istart), theta(istart), 'r*');
        plot(phi(index), theta(index), 'r');
        grid on;
   end
    axis([-180 180 -90 90]);
    set(gca,'XDir','reverse');
    xlabel('azimuth in zx plane (x -> -z)');
    ylabel('elevation above the zx plane');
    title(titlestr);
    if dontplot==0
        legend('path', 'start');
    end

    clear phi theta phi_save theta_save;


% Save LAP image file
case 'h'
    proben=input('LAP probe number: ');

    figure(fign)
    set(gcf,'Position', windowsize);
    set(gca,'XDir','reverse');          % reverses the phi-axis
    set(gcf,'Position',[0 0 1275 945]); % maximizes the window size
    axis([-180 180 -90 90]);

    drawrosetta(proben,0);              % draw the spacecraft without the antenna
    set(gca,'XDir','reverse');          % reverses the phi-axis
    axis([-180 180 -90 90]);
    filename=strcat('image_p', num2str(proben), '_1');
    print(gcf, '-dpng', '-r400', filename)

    hold off;
    drawrosetta(proben,1);              % draw the spacecraft with the antenna
    set(gca,'XDir','reverse');          % reverses the phi-axis
    axis([-180 180 -90 90]);
    filename=strcat('image_p', num2str(proben), '_2');
    print(gcf, '-dpng', '-r400', filename)

    hold off;
    drawrosetta(proben,1.1);            % draw the spacecraft with an extra big antenna (110% radius)
    set(gca,'XDir','reverse');          % reverses the phi-axis
    axis([-180 180 -90 90]);
    filename=strcat('image_p', num2str(proben), '_3');
    print(gcf, '-dpng', '-r400', filename)


    disp('Images saved in file "image_pX_X.png". Manually convert the files to INDEXED GIF before calculating wake/eclipse data.');

    clear filename;


% Calculate wake- and eclipse data from image file
case 'i'
    proben=input('LAP probe number: ');
    clear scindex vcindex;
    scindex=zeros(3,length(sphi));
    vcindex=zeros(3,length(sphi));
    for n=1:3               % loop through the three image files
        filename=strcat('image_p',num2str(proben),'_', num2str(n));
        disp(strcat('Reading image file "',filename , '.gif"...'));
        [I, colmap] = imread(filename,'gif');
        disp('Cropping image...');
        img = imcrop(I,[418 181 2477 1954]);
        colbg=img(1,1);     % index number for the (white) background color
        disp('Calculating...');
        disp(strcat('Size of image matrix: ',num2str(size(img))));

        % this loop checks whether the coordinate is "behind" the s/c or visible from the probe
        for i=1:length(sphi)
            if isnan(sphi(i))       % if data values are missing..
                scindex(n,i)=NaN;
                vcindex(n,i)=NaN;
            else
                [sx,sy]=imagexy(sphi(i),stheta(i));     % calculate the "pixel" corresponding to a
                [vx,vy]=imagexy(vphi(i),vtheta(i));     % certain phi-theta pair
                scindex(n,i)=img(sy,sx);                % (color) value for the image coordiante/pixel
```

```
                vcindex(n,i)=img(vy,vx);
                if scindex(n,i)==colbg                  % if visible (sunlit) set the value to 0
                    scindex(n,i)=0;
                else                                    % if in the shadow of the s/c set the value to 1
                    scindex(n,i)=1;
                end
                if vcindex(n,i)==colbg                  % if not in wake
                    vcindex(n,i)=0;
                else                                    % if in wake
                    vcindex(n,i)=1;
                end
            end
        end
    end



    switch proben
    % add together the values calculated from the three images
    % to get non-binary variables for the eclipse and wake data
    case 1
        wake1=(vcindex(1,:)+vcindex(2,:)+vcindex(3,:))/3;
        eclipse1=(scindex(1,:)+scindex(2,:)+scindex(3,:))/3;
        disp('Wake-data for the current time span saved in variable "wake1"');
        disp('Eclipse-data for the current time span saved in variable "eclipse1"');
    case 2
        wake2=(vcindex(1,:)+vcindex(2,:)+vcindex(3,:))/3;
        eclipse2=(scindex(1,:)+scindex(2,:)+scindex(3,:))/3;
        disp('Wake-data for the current time span saved in variable "wake2"');
        disp('Sunlit-data for the current time span saved in variable "Eclipse2"');
    end

    clear vcindex scindex sx sy vx vy I colmap bgcol filename;


% Plot wake- and sunlit data vs. time
case 'j'
    proben=input('LAP probe number: ');
    switch proben
    case 1
        wake=wake1;
        eclipse=eclipse1;
    case 2
        wake=wake2;
        eclipse=eclipse2;
    end

    figure(fign)
    set(gcf,'Position', windowsize);

    subplot(2,1,1);
    plot(jd(index)+timediff,eclipse(index));
    grid on;
    axis([jd(istart)+timediff jd(iend)+timediff -0.2 1.2]);
    datetick('x',19);
    xlabel('time');
    ylabel('eclipse index');
    ax=axis;
    title('Eclipse information');

    subplot(2,1,2);
    plot(jd(index)+timediff,wake(index));
    grid on;
    axis([jd(istart)+timediff jd(iend)+timediff -0.2 1.2]);
    datetick('x',19);
    xlabel('time');
    ylabel('wake index');
    ax=axis;
    title('Wake information');

%    suptitle(strcat(tit, ' LAP probe #',num2str(proben), ' [printed ', datestr(now), ']'));
    suptitle(strcat(tit, ' LAP probe ',num2str(proben), ' (',datestr(jd(istart)+timediff,20),...
        '@', datestr(jd(istart)+timediff,15),'-',datestr(jd(iend)+timediff,20),'@',...
        datestr(jd(iend)+timediff,15),')'));

    clear wake Eclipse ax titlestr;

case 'k'
    % plot the whole mission
    figure(fign)
    set(gcf,'Position', windowsize);
    AU=149597870.691;                   % define the astronomical unit
    plot3(rr_hea(1,:), rr_hea(2,:), rr_hea(3,:));
    hold on;
    plot3(rc_hea(1,:), rc_hea(2,:), rc_hea(3,:), 'k--');
    plot3(0,0,0,'y*');
    % plot mars and earth orbit
    theta=(0:1:360);
    [xx,yy] = pol2cart(theta*pi/180,1);
    plot3(dist_earth*xx, dist_earth*yy, zeros(1,length(xx)),'g--');
    plot3(dist_mars*xx, dist_mars*yy, zeros(1,length(xx)),'r--');
    axis equal;
    grid on;
```

```
    xlabel('x');
    ylabel('y');
    zlabel('z');
    title(strcat('ROSETTA mission 3D plot (J2000) [printed ', datestr(now), ']'));

    d(1,:)=[2004 02 04];    % launch
    d(2,:)=[2005 03 04];    % earth flyby 1
    d(3,:)=[2007 02 25];    % mars flyby
    d(4,:)=[2007 11 13];    % earth flyby 2
    d(5,:)=[2009 11 13];    % earth flyby 3
    d(6,:)=[2014 05 10];    % comet approach
    d(7,:)=[2015 12 31];    % end of mission
    r1=find(min(abs(datenum(d(1,:))-datenum(dater(1:3,:)')))==abs(datenum(d(1,:))-datenum(dater(1:3,:)'))); % rosetta
    r2=find(min(abs(datenum(d(2,:))-datenum(dater(1:3,:)')))==abs(datenum(d(2,:))-datenum(dater(1:3,:)'))); % indecies
    r3=find(min(abs(datenum(d(3,:))-datenum(dater(1:3,:)')))==abs(datenum(d(3,:))-datenum(dater(1:3,:)'))); % for the
    r4=find(min(abs(datenum(d(4,:))-datenum(dater(1:3,:)')))==abs(datenum(d(4,:))-datenum(dater(1:3,:)'))); % dates
    r5=find(min(abs(datenum(d(5,:))-datenum(dater(1:3,:)')))==abs(datenum(d(5,:))-datenum(dater(1:3,:)')));
    r6=find(min(abs(datenum(d(6,:))-datenum(dater(1:3,:)')))==abs(datenum(d(6,:))-datenum(dater(1:3,:)')));
    c1=find(min(abs(datenum(d(1,:))-datenum(datec(1:3,:)')))==abs(datenum(d(1,:))-datenum(datec(1:3,:)'))); % comet
    c2=find(min(abs(datenum(d(2,:))-datenum(datec(1:3,:)')))==abs(datenum(d(2,:))-datenum(datec(1:3,:)'))); % indecies
    c3=find(min(abs(datenum(d(3,:))-datenum(datec(1:3,:)')))==abs(datenum(d(3,:))-datenum(datec(1:3,:)'))); % for the
    c4=find(min(abs(datenum(d(4,:))-datenum(datec(1:3,:)')))==abs(datenum(d(4,:))-datenum(datec(1:3,:)'))); % dates
    c5=find(min(abs(datenum(d(5,:))-datenum(datec(1:3,:)')))==abs(datenum(d(5,:))-datenum(datec(1:3,:)')));
    c6=find(min(abs(datenum(d(6,:))-datenum(datec(1:3,:)')))==abs(datenum(d(6,:))-datenum(datec(1:3,:)')));
    c7=find(min(abs(datenum(d(7,:))-datenum(datec(1:3,:)')))==abs(datenum(d(7,:))-datenum(datec(1:3,:)')));

    irs=[r1(1) r2(1) r3(1) r4(1) r5(1)];  % make an array with (the first of) the resp. dates
    ire=[r2(1) r3(1) r4(1) r5(1) r6(1)];
    ics=[c1(1) c2(1) c3(1) c4(1) c5(1) c6(1)];
    ice=[c2(1) c3(1) c4(1) c5(1) c6(1) c7(1)];

    figure(fign+1)
    set(gcf,'Position', windowsize);
    xmin=min(rc_hea(1,:));
    xmax=max(rc_hea(1,:));
    ymin=min(rc_hea(2,:));
    ymax=max(dist_mars);
    subname='abcdef';

    for i=1:6
%        figure(fign+i)
        hold on;
        subplot(2,3,i);
        if i<6
            plot(rr_hea(1,irs(i):ire(i))/AU, rr_hea(2,irs(i):ire(i))/AU);
        end
        hold on;
        plot(rc_hea(1,ics(i):ice(i))/AU, rc_hea(2,ics(i):ice(i))/AU, 'k');
        hold on;
        if i==6
            plot(rc_hea(1,ice(i))/AU, rc_hea(2,ice(i))/AU, 'ko');
        end
        plot(0,0,'y*');
        % plot mars and earth orbit
        plot(dist_earth*xx/AU, dist_earth*yy/AU,'g--');
        plot(dist_mars*xx/AU, dist_mars*yy/AU, 'r--');
        if i<6
 %           plot(rr_gea(1,ire(i))/AU, rr_gea(2,ire(i))/AU, 'bo');
        end
        axis([xmin xmax ymin ymax]/AU);
        axis equal;
%        grid on;
        xlabel('x');
        ylabel('y');
        title(strcat(subname(i), ') ', datestr(datenum(d(i,:))),' - ', datestr(datenum(d(i+1,:)))));
        grid on;
    end

    suptitle(strcat('ROSETTA mission plot'));

    clear xx yy;
%    subplot(2,3,6)
%    plot(0,0,'b',0,0,'k',0,0,'g',0,0,'r');
%    legend('path of Rosetta', 'path of the comet', 'earth orbit', 'mars orbit');
%    axis off;

case 'l'
    % set the current time span
    disp('Current time span:');
    disp(strcat('Minimum traj data index :1 :', datestr(jd(1)+timediff)));
    disp(strcat('Start index             :', num2str(istart),' :', datestr(jd(istart)+timediff)));
    disp(strcat('Closest approach index  :', num2str(iclosest), ' :', datestr(jd(iclosest)+timediff)));
    disp(strcat('End index               :', num2str(iend), ' :', datestr(jd(iend)+timediff)));
    disp(strcat('Maximum dtraj ata index :', num2str(length(jd)), ' :', datestr(jd(length(jd))+timediff)));
    istart=input('Enter new value for start index:');
    iend=input('Enter new value for end index:');

case 'm'
    % save coordinates
    [yy,mm,temp1,temp2,temp3,temp4]=datevec(jd+timediff);
    dd=time(1,:);
    HH=time(2,:);
    MM=time(3,:);
```

```
    X=rr_gse(1,:);
    Y=rr_gse(2,:);
    Z=rr_gse(3,:);
    if att==0
        savedata=[yy; mm; dd; HH; MM; X; Y; Z];
    else
        savedata=[yy; mm; dd; HH; MM; X; Y; Z; ];
    end

    fid=fopen(savename, 'w');

    fprintf(fid, '%04u %02u %02u %02u %02u %13e %13e %13e\n', savedata);

    fclose(fid);

    clear yy mm temp1 temp2 temp3 temp4 dd HH MM X Y Z savedata;
    disp(strcat('Data saved in "',savename,'".'));


case 'n'
    % save sunlit and wake data to use with "getswp"
    format long;
    savedata = [(jd+timediff)' eclipse1' eclipse2' wake1' wake2'];
    save sunwake.txt savedata -ascii -tabs -double
    disp('Sunlit/wake for the two probes saved in "sunwake.txt" in the format');
    disp('MATLAB day number, eclipse probe 1, eclipse probe 2, wake probe 1, wake probe 2');
    disp('1 represent eclipse resp. not wake, 2 represent eclipse resp. wake.');


case 'o'
    % plot "measured data color coded" dots in the long-lat plot as seen from one of the probes
    proben=input('LAP probe number: ');
    relradius=input('High-gain antenna relative radius (1=100%): ');

    load fb1photo;

    switch proben
    case 1
        it=t1;
        iphoto=if1*1e9;
        titlestr=strcat(tit,' - Sun vector as seen from LAP probe #1 (',datestr(jd(istart)+timediff,20),...
            '@', datestr(jd(istart)+timediff,15), '-',datestr(jd(iend)+timediff,20),'@', ...
            datestr(jd(iend)+timediff,15),') [printed ', datestr(now), ']');
    case 2
        it=t2;
        iphoto=if2*1e9;
        titlestr=strcat(tit,' - Sun vector as seen from LAP probe #2 (',datestr(jd(istart)+timediff,20),...
            '@', datestr(jd(istart)+timediff,15), '-',datestr(jd(iend)+timediff,20),'@', ...
            datestr(jd(iend)+timediff,15),') [printed ', datestr(now), ']');
    end

    sphiint=interp1(jd+timediff,sphi,it);
    sthetaint=interp1(jd+timediff,stheta,it);
    ik=1/abs(max(iphoto)-min(iphoto));

    figure(fign)
    set(gcf,'Position', windowsize);
    drawrosetta(proben, relradius);

    % just for testing...
%    angleshift=-15;
    angleshift=0;

    for i=1:length(sphiint)
        if ~isnan(iphoto(i))
            plot(sphiint(i)+angleshift, sthetaint(i), '.', 'color', ...
                [1-abs(iphoto(i)-min(iphoto))*ik 0 abs(iphoto(i)-min(iphoto))*ik]);
        end
    end
    axis([-180 180 -90 90]);
    grid on;
    set(gca,'XDir','reverse');
    xlabel('azimuth in zx plane (x -> -z)');
    ylabel('elevation above the zx plane');
    title(titlestr);
    disp(strcat('RED=', int2str(min(iphoto)), ' nA, BLUE=', int2str(max(iphoto)), ' nA.'));
%    clear it iphoto sphiint sthetaint ik;
end
```