Analyzing the dayside terrestrial magnetopause using Cluster data

1. Introduction

In this lab you are introduced to handle real satellite data obtained by one of the Cluster satellites (see Section 2). You will be given tools to analyze magnetopause crossings and will be asked to estimate the magnetopause normal (**n**) and the current density (**J**).

--- ***Minimum variance of the magnetic field (MVAB)***

This is a simple method to estimate a local magnetopause normal (**n**) based on the magnetic field (**B**) change across the magnetopause. The method is described in the *How to determine a normal vector to the magnetopause boundary using MVAB and MVAE from spacecraft data,* by Lars-Göran Westerberg (from now on referred to as *LGW*). Read the sections related to MVAB carefully.

--- The magnetopause current

The magnetopause current **J** can be estimated from the change of magnetic field components across the magnetopause by using Ampere´s law:

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{J}.$$

2. The Cluster Satellites

The Cluster mission is a key project for the European Space Agency (ESA). It is composed of four spacecraft studying the magnetospheric structures and solar wind interactions (Escoubet et al 2001). The Cluster I mission came to an end in 1996, 37 seconds after launch, when the rocket Ariane-5 (carrying the four Cluster satellites) exploded. In the year 2000 the launch was successful, and four spacecraft were put in elliptical orbits around the Earth (Cluster II mission). Originally planned to last up until 2003, the mission has been extended several times and the satellites are still in operation. They are orbiting the Earth in a tetrahedral formation, with orbit perigee and apogee around 3 Re and 19 Re respectively, with an orbit period of 57 hours and a spin period of 4 seconds. The scientific objective for the mission is to study the interaction between the magnetosphere and the solar wind in key plasma regions as the magnetotail, the magnetospheric cusps, the auroral zone, the bow shock, and the magnetopause among others. The formation of four spacecraft allows for three-dimensional analysis of the physical structures and the possibility to distinguish between spatial and temporal phenomena.

In the analysis you are supposed to do now, you will however only use data from one of the satellites. These data (*20020509_data.mat*) can be found at and downloaded from Canvas.

3. Exercise 1

Perform the example in Section 2.1.1 of *LGW*, that is find the magnetopause normal for the magnetopause data referred to in *LGW.* The magnetic field data needed are given in the table below, given in nT. If done correctly you should get **n** = [0.8671 -0.4978 0.0187], corresponding to the smallest eigenvalue e = 7.1. This exercise is meant to be fairly straightforward and when made correctly you can use the code as a base to solve the next exercise.

Bx:	By:	Bz:
-13.6	-24.7	54.6
-14.8	-24.9	58.7
-13.4	-17.2	62.4
-14.0	-25.0	43.8
-7.1	-4.5	33.5
-0.9	-5.0	44.4
-10.0	-0.4	44.6
-6.1	-4.8	21.1
1.2	1.6	21.0
-3.4	-3.9	4.1
-0.9	1.2	5.0
-1.0	-1.5	12.3
11.0	13.2	29.7
19.1	34.4	20.1
24.9	50.1	1.9
29.2	47.1	-10.6

4. Exercise 2

Download the Cluster data file from Canvas (*20020509_data.mat*). These data corresponds to 6 hours of data (06:00-12:00, 2002-05-09), where the magnetopause sweeps over the satellite several times. The data parameters are specified in the Appendix. Plot times series of proton data (number density, velocity, temperature, plasma beta) and magnetic field data. Consider if a linear or logarithmic scale is best suitable for each parameter. Generally: if the parameter ranges over several orders of magnitude a logarithmic scale is to prefer

* Can you distinguish the magnetosphere region from the magnetosheath region? Discuss and motivate.

* How many magnetopause timings do you observe? And at what times do they (approximately) occur? (In order to do that you will need to plot for smaller time intervals)

* Use the MVAB method and estimate the magnetopause normal for each magnetopause crossing. In general, are the estimated normal directions what you expected, considering the spacecraft position in relation to Earth and an ideal textbook magnetopause?

* Estimate the current density for each crossing and report. Compare to typical magnetopause currents. Are your results reasonable?

5. Report

Hand in a report of your results and discussion. No MATLAB codes need to be included.

6. Appendix

--- The data set

You will be provided a mat-file (in Canvas, where all data you need is stored, and which can be uploaded in matlab. The mat-file is named data practical2.mat and below follows a clarification of the parameters:

BX; BY; BZ: the magnetic field x-, y-, and z-components, [nT]

X_gse; Y_gse; Z_gse: the spatial x-, y- and z-coordinates, [RE]

n_H: H+ number density, [cm^{-3}]

Tpar_H; Tperp_H: Parallel and perpendicular H+ temperature, [eV]

vpar_H; vperp_H: _ H+ parallel and perpendicular bulk velocity, [km/s]

matlab_time: time [matlab time]

--- Matlab time

The time format can seem to be a bit strange at first sight, but using some simple commands, it becomes manageable. Matlab time is nothing more than number of days from year 0000. The *datestr* command allows you to convert matlab time to normal time. For example

```
>> datestr(5)
```

```
        ans = 05-jan-0000
```

```
>> datestr(735664)
        ans = 06-mar-2014
```

More details (hours, minutes and seconds for example) can be achieved by writing

```
>> datestr(735664.34, 'dd-mm-yyyy HH:MM:SS')
        ans = 06-03-2014 08:09:36
```

You can also go the opposite way, converting a specified time into matlab time. This is necessary when you for example want to pick out data for a certain time interval: say between 15:15:00 to 16:45:00 on 11 Dec 2014. To find the corresponding matlab time you can write:

```
>> t_start = datenum('11-12-2014 15:15:00', 'dd-mm-yyyy HH:MM:SS');
>> t_end = datenum('11-12-2014 16:45:00', 'dd-mm-yyyy HH:MM:SS');
>> ii = find(mattime>t_start & mattime<t_end);
```

The parameter *ii* now contains the indices of all data points corresponding to the time interval of interest. Now you can plot any physical parameter (the proton density for example) for that time interval by writing

```
>>plot(mattime(ii), Hdens(ii))
```

Then use *datetick* to get the time axis in a familiar form.