Internship Report (2 months)

Mercury science through databases and remote sensing observations: Increase the scientific exploitation of Mercury in view of BepiColombo

Emma Vellard

Internship, Master 1

ESA supervisor(s): Santa Martinez (ESAC), Sebastien Besse (ESAC), Claudio Munoz (ESAC), Thomas Cornet (ESAC)

June 2022

Table of contents

Chapter 1

Abstract

1.1 English

The planet Mercury is about to be reached by the ESA/JAXA BepiColombo spacecraft in 2025. Prior to this mission, the planet was studied by two NASA space missions: Mariner10 and MESSENGER. They have provided data that are still used by researchers and a project is about to change the way they use it: the MeSS.

In order to better understand the science and history of Mercury, it is necessary to explore thousands of complex data sets on surface composition, materials, phenomena and surface images in relation with several parameters such as spacecraft orbital plane, mission's events, data quality, footprint size or location. The MeSS project was developed to better bring together all these elements in order to find links between certain physical aspects.

The current project consists of a database of MASCS data from the MESSENGER mission. The aim of this internship is to extend the database by adding new datasets, to improve the way they are used and to increase the efficiency to get better computation times.

During these two months, a map of Low Reflectance Materials was added to the database and the computation time on the VIS and NIR instruments was improved by modifying the data characteristics.

1.2 French

La planète Mercure est sur le point d'être atteinte par la sonde spatiale ESA/JAXA BepiColombo en 2025. Avant cette mission, la planète a été étudiée par deux missions spatiales de la NASA : Mariner10 et MESSENGER. Elles ont fourni des données qui sont toujours utilisées par les chercheurs et le projet MeSS est sur le point de changer la façon dont ils les utilisent.

Afin de mieux comprendre la science et l'histoire de Mercure, il est nécessaire d'explorer des milliers de jeux de données complexes sur la composition de la surface, les matériaux, les phénomènes et l'image de la surface en fonction de plusieurs paramètres tels que le plan orbital du vaisseau spatial, les événements de la mission, la qualité des données, la taille de l'empreinte ou l'emplacement. Le projet MeSS a été développé pour mieux rassembler tous ces elements afin de trouver des liens entre certains aspects physiques.

Le projet actuel est composé d'une base de données constituée des données MASCS de la mission MESSENGER. Le but de ce stage est de développer la base de données en ajoutant de nouveaux jeux de données, d'améliorer la façon dont ils sont utilisés et d'augmenter l'efficacité pour obtenir de meilleurs temps de calcul.

Pendant ces deux mois, une carte des matériaux à faible réflectance a été ajoutée à la base de données et le temps de calcul sur les instruments VIS et NIR a été amélioré en modifiant les caractéristiques des données.

Chapter 2

Context

2.1 Mercury

Mercury is the smallest planet in our Solar System and the closest planet to the Sun. It is in the most intense solar radiation environment, and it is the most affected by solar tides. The temperature range of its surface is the most extreme in the Solar System, from -180 to 430 degrees Celsius. The planet rotates very slowly, one day on Mercury takes 59 Earth days. Because of its faster orbit, Mercury year takes 88 Earth days. Despite the intense heat the planet faces as it rotates, areas that are permanently shaded, such as some polar craters, may hold deposits of ice. As one of the very few terrestrial planets, Mercury plays an important role in comparative planetology studies of processes relevant to our own planet Earth [1].

The surface of Mercury is scattered by numerous geological elements: impact basins, heavily cratered terrains, scarps or volcanoes.

Craters cover 85% of the surface and are surrounded by plains of volcanic origin. The planet was bombarded by comets and asteroids during its history and received impacts over its entire surface during its period of intense crater formation, facilitated by the lack of any atmosphere to slow impactors down [2]. Mercury presents different crater aspects according to their age. The youngest ones (less than 3.7 Ga years) are known as smooth craters because of the few impacts that have occurred, they were not affected by the 'late heavy bombardment' as the oldest ones were [3]. One of the largest craters is the 1500 km wide Caloris basin (*Figure* 2.1). The large number of comet impacts indicates a geology that has been inactive since the formation of the planet.

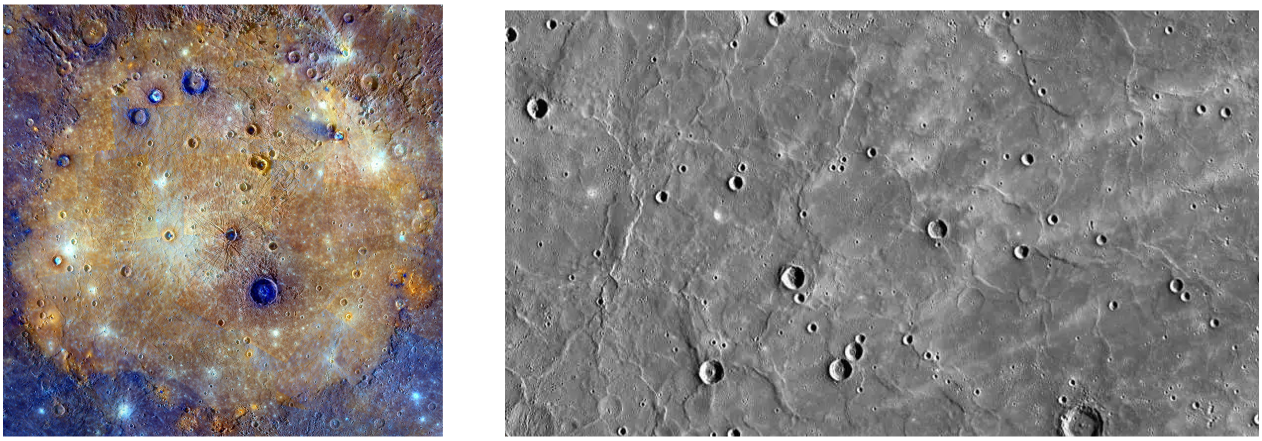


Figure 2.1: Caloris Basin (left) (*source: NASA Science*) and Northern volcanic plains (right) (*source: Planetary Society*)

Mercury's northern region is recovered by expansive smooth plains, created in the past by a huge amounts of volcanic material flooding across Mercury's surface. The volcanic lava flows buried craters, leaving only traces of their rims visible. These volcanic plains are younger than Mercury's rougher surface because they have fewer impact craters.

Mercury's crust is mainly composed of silicate rocks and metal. It is rich in volatile elements such as sulfur, carbon, potassium, chlorine, and sodium [4], and altered by different phenomena such as hollows or faculae (*Figure* 2.2). Hollows are bright rounded depression over the surface within craters and faculae are high-albedo surficial deposits typically tens of km in diameter that have distinct outer edges [5]. The faculae, pyroclastic deposits, are located principally on the floor of craters [6]. These two phenomena are not yet very well known and are the subject of several research topics.

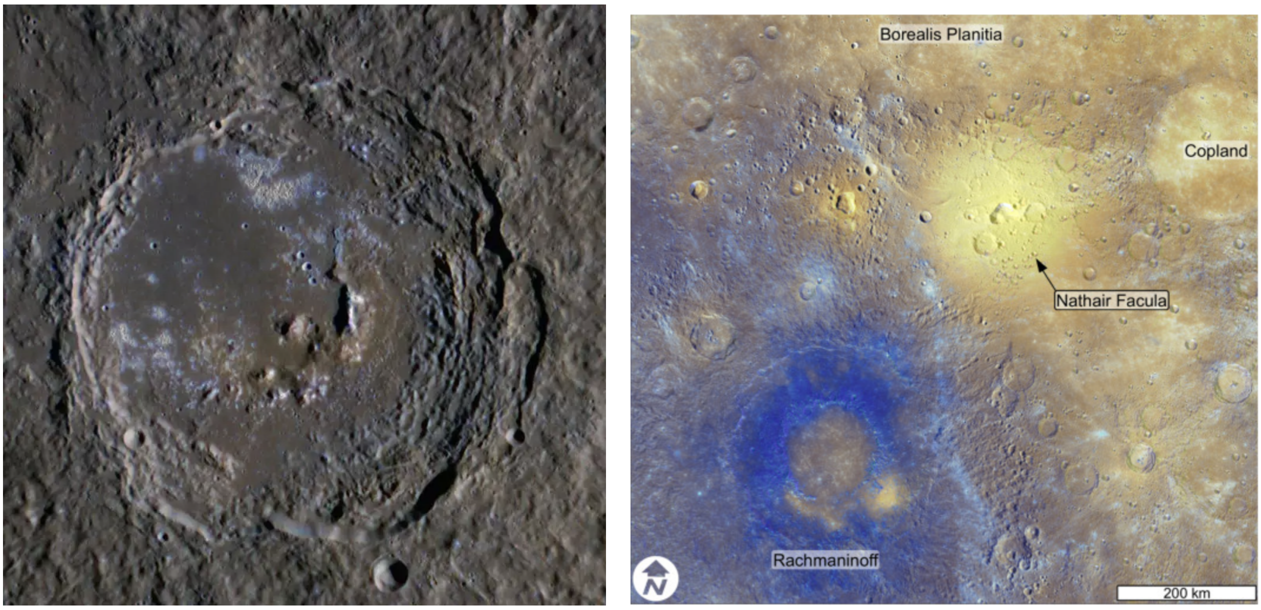


Figure 2.2: Zeami crater dotted by hollows (left) (*source: Planetary Society*) and Nathair facula (right) [7]

2.2 Space missions on Mercury

Mercury is the least explored terrestrial planet in our Solar System. So far, only three space probes have approached the planet: Mariner10, MESSENGER and BepiColombo and it was not until 2011 that the first was put into orbit.

Three main reasons make it difficult to explore Mercury:

- The planet is difficult to reach: Because of Mercury's proximity to the Sun, a spacecraft must constantly brake against the gravitational pull of the star and this requires a lot of energy. To achieve this, there are two options: the first one is to have a huge spacecraft with lots of fuel, and the second one is to use the gravity of the other planets to slow down on the way, which increases a lot the journey to the planet's orbit[1].
- The planet's surface is too hot to get into close orbit: Sunlight around Mercury is about 10 times more intense than near Earth. Moreover, the planet's scorched surface radiates heat back to space then probes have to endure temperatures close to 450 degrees at 200km to the surface.
- The planet is difficult to observe: As the innermost planet of the Solar System, it appears close to the Sun. The observation from Earth is only possible before sunrise or after sunset and close to the horizon. Moreover, the intense sunlight due to its proximity to the Sun, can damage the optics of telescopes both from the ground and from space.

Only three mission have been able to study Mercury so far. Based on the results of the first one, scientist and engineers have been able to design two more precise missions, MESSENGER and BepiColombo, to answer the questions raised by the previous one.

The spacecraft Mariner 10 was the first to reach Mercury and the objectives were to study the surface and atmosphere characteristics of Mercury. The spacecraft imaged 45% of the planet surface during three flybys[2] of the planet. The 2,800 pictures taken reveal a Moon-like surface with craters and large basins.

More than thirty years later, MESSENGER mission entered into Mercury orbit in 2011 and retrieve the main data used these days by the scientists. The main objectives concerning planetary surface science were the following: study the

[1]The BepiColombo probe takes seven years and nine gravity assist manoeuvres to orbit Mercury, while Solar Orbiter reaches its target orbit around the Sun in less than two years.

[2]Flyby: Flight past a point, especially the close approach of a spacecraft to a planet or moon for observation.

planet geologic history, characterize the chemical composition of Mercury's surface, determine the volatile inventory at the poles. MESSENGER spacecraft was composed of different scientific instruments to answer these questions such as Mercury Dual Imaging System (MDIS) and Mercury Atmospheric and Surface Composition Spectrometer (MASCS), which will be detailed later.

In December 2025, the BepiColombo mission, developed by European Space Agency (ESA) and Japan Aerospace Exploration Agency (JAXA), will reach Mercury. The spacecraft is composed of two different satellites: Mercury Planetary Orbiter (MPO) and Mercury Magnetospheric Orbiter (MMO-MIO) which will for the first time study the planet simultaneously. The main objectives concerning planetary surface science are the following: exo- and endogenic surface modifications: cratering, tectonics, polar deposits and volcanism, Mercury's figure, interior structure, and composition, study the origin and evolution of a planet close to its parent star.

2.3 Remaining questions

These three missions provide numerous data sets that are currently spread in archives, websites or papers depending on the spacecraft, instrument and their relevance to research projects. The data are complex to obtain and to analyse because not always provided in the right format.

In order to better understand Mercury and to open new perspectives for planetary research, the MeSS[1] project has been under development since 2015. The main objectives are to facilitate access to all types of data such as spectra or maps, to store them in a single place and to provide innovative research methods that could lead to new results.

The MeSS project is able to respond to different situations. Firstly, the researcher will obtain more easily and in a way that is adapted to his/her project the desired data thanks to the database infrastructure. In another case, the researcher can explore the data provided by the MeSS in order to find new correlations or analyses that have not been investigated before.

[1]MeSS: Mercury Surface Spectroscopy

Chapter 3

MeSS

The short term objective of the MeSS is to store in a same place the data obtained by several instruments onboard the MESSENGER mission, then to extend it to some derived products from research results.

3.1 MeSS infrastructure

The MeSS projects is composed of the **MASCS** database which is currently used by researchers [8] [9] [10]. It provides the different data obtained by the MASCS[2] instrument on the MESSENGER spacecraft. This instrument is consisted of a small Cassegrain telescope which simultaneously fed an Ultraviolet and Visible Spectrometer (UVVS) and a Visible and Infrared Spectrograph (VIRS). It was designed to measure profiles with altitude of known exospheric species, to search for previously undetected exospherics species and to observe Mercury's surface in the far and middle ultraviolet at a spatial scale of 10 km or smaller.

The **MASCS** database includes only the VIRS data because it is the one which characterizes the surface. The primary data of VIRS are reflectance[3] spectra data and cover the wavelength ranges of the visible (VIS), from 300 to 1050 nm, and of the near infrared (NIR), from 850 to 1450 nm. The VIRS was designed to map the surface reflectance over Mercury's surface on spatial scales of 5 km.

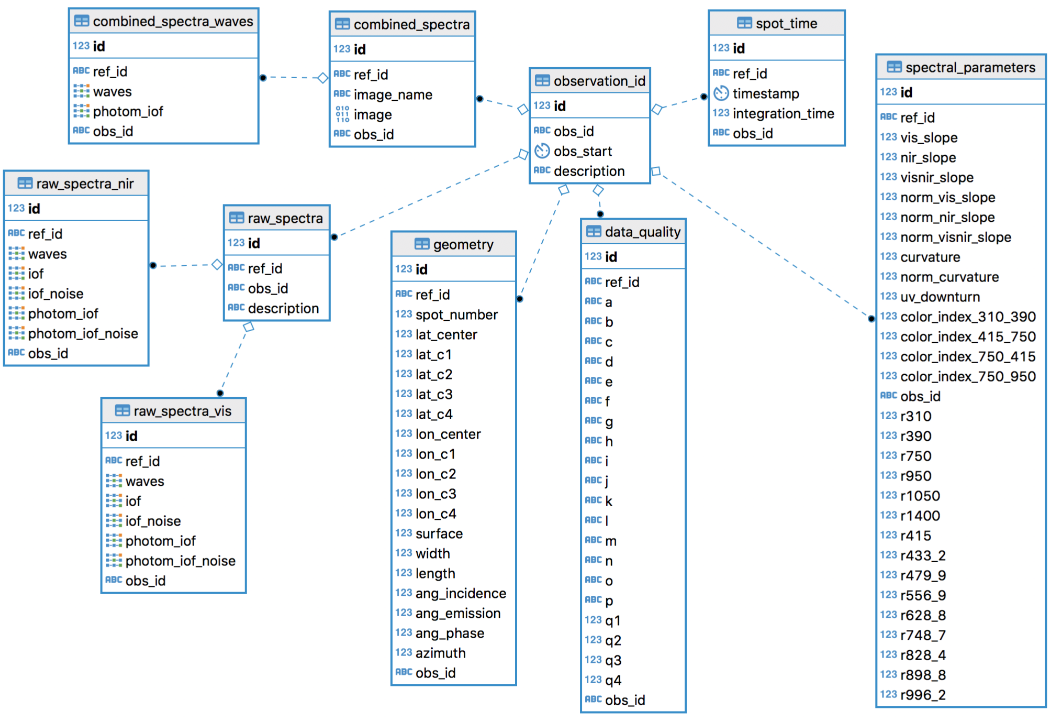


Figure 3.1: Current **MASCS** database structure

[2] MASCS: Mercury Atmospheric and Surface Composition

[3] Reflectance: the fraction of light reflected by the object over the solar flux coming to the object

About 4.8 millions spectra are stored in the database. They are characterised by different parameters such as the quality of the samples, in the table *data_quality*, or the integration time, in the table *spot_time* for each footprint[1].

The footprint of the instrument have different oval shapes according to the spacecraft altitude and tilt (*Figure* 3.2). They are stored in the table *geometry*. Each footprint is spotted according to five points which give the position in latitude and longitude of the center and the four extremities. The table also provides the width, length, area of each footprint and the detailed observation geometry such as azimuth, incidence, emission and phase angles of each instrument's measurements.

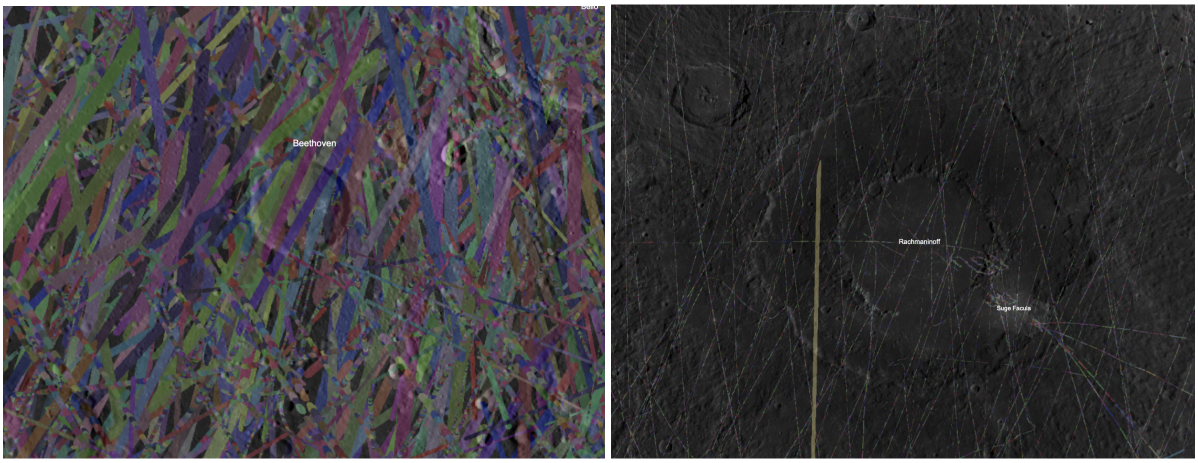


Figure 3.2: Footprints over Beethoven (left) and Rachmaninoff craters (right) (source: MESSENGER Quickmap)

The reflectance spectra included in the MASCS database have been selected according to several parameters. Afterwards, they are stored in two different tables, *raw_spectra_nir* and *raw_spectra_vis* according to their wavelength range. The table *combined_spectra_waves* corresponds to the association of the VIS and NIR data to build full spectra from 300 to 1450 nm such as in Figure 3.3 [5].

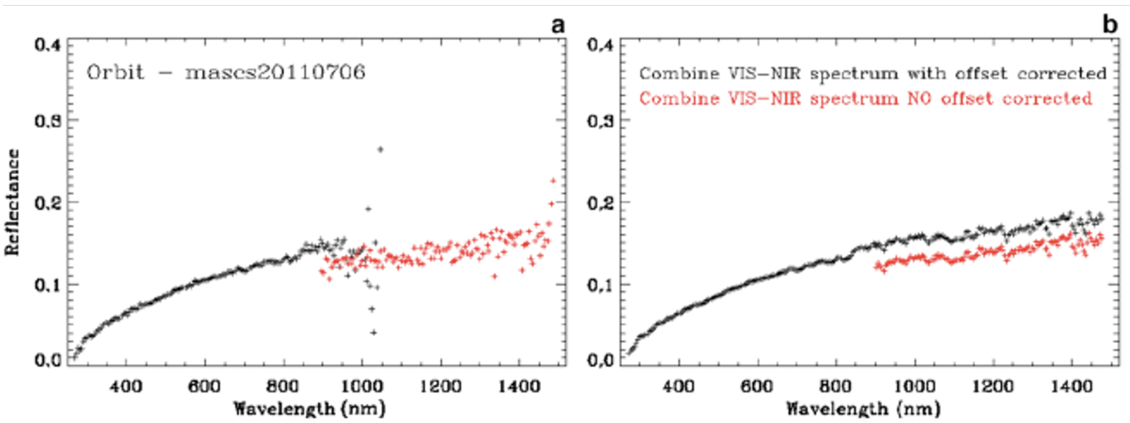


Figure 3.3: Combined spectra

[1]A footprint corresponds to the ground area retrieve by the spacecraft during one of its orbit

This database is currently used by researchers and is able to provide several type of results according to the query done [11]. Figures 3.4, 3.5 and 3.6 illustrate typical uses of the database for extracting and filtering data.

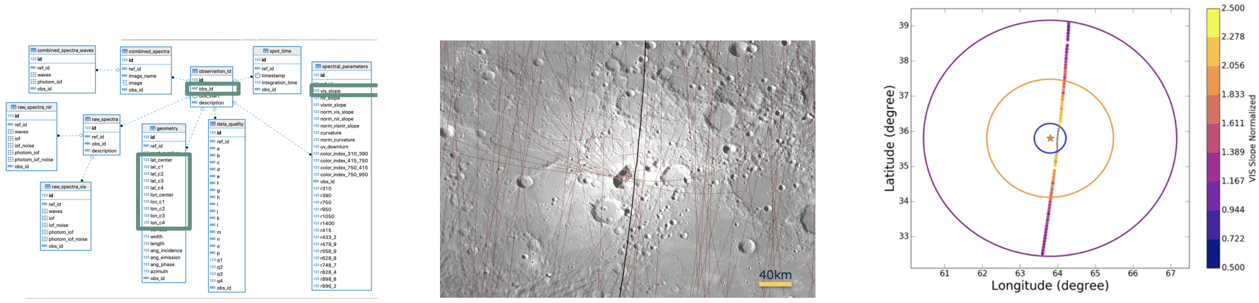


Figure 3.4: Nathair facula: The query is done over three different tables. The user chooses a specific orbit, the latitude and longitude surrounding the Nathair facula and the corresponding visible slope. The resulting plot is the intensity of the VIS slope over the Nathair facula according to specific footprints which cross it. The blue line is the volcanic vent, the orange one is the previous measurement of pyroclastic deposit and the purple one is the new measurement researchers highlight according to their study thanks to the database.

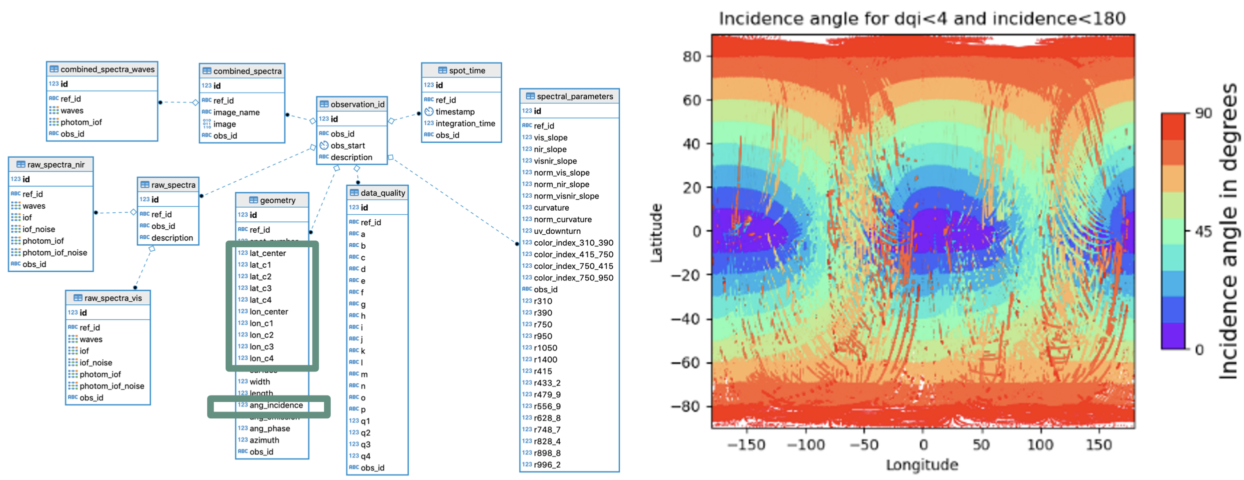


Figure 3.5: Incidence angle over Mercury surface: The query is now performed on a single table. The user selects the complete planetary scale and the angle of incidence on it. The resulting plot highlights the incidence condition over the planet and provides to researchers the possible observation conditions.

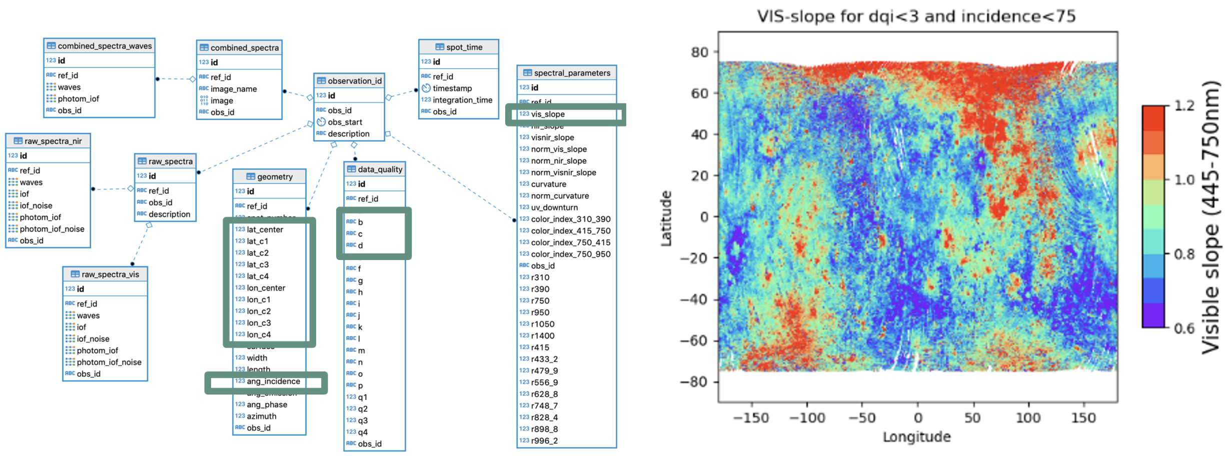


Figure 3.6: Planet scale properties: The query covers three different tables. For the complete planet, the user selects the visible slope for an incidence below 75 degrees and specific quality parameters. According to Figure 3.5, the map has been cropped on the upper and lower latitude. Such a map is useful for highlighting certain surface features such as the presence of volcanic plains like the red part on the upper part of the figure.

3.2 MeSS new infrastructure

As part of this work three new tables have included and populated in the **MASC** database (*Figure* 3.7). Moreover the table *raw_spectra* has been renamed *calibrated_spectra* because of a lack of clarity.

The current priorities were to improve the query time of the tables *calibrated_spectra_vis* and *calibrated_spectra_nir*, by adding two new tables named *reduced_calibrated_spectra_vis* and *reduced_calibrated_spectra_nir* with reduced data. The performances of the **MASCS** database were low to extract reflectance spectra from these data tables, to be used to access the most calibrated data products of the database (queries on the order of minutes). A second objective is expand the usage of the database by including new data set on low reflectance materials at the surface of Mercury in the table *derived_product*.

The following parts are using codes detailed in Chapter 6.

In a first place, before interacting with the database, the user needs to get access to it. The connection function (*Appendix* 6.1) allows to connect to it, by using the connection information provided by the user. This function will be used in the following codes.

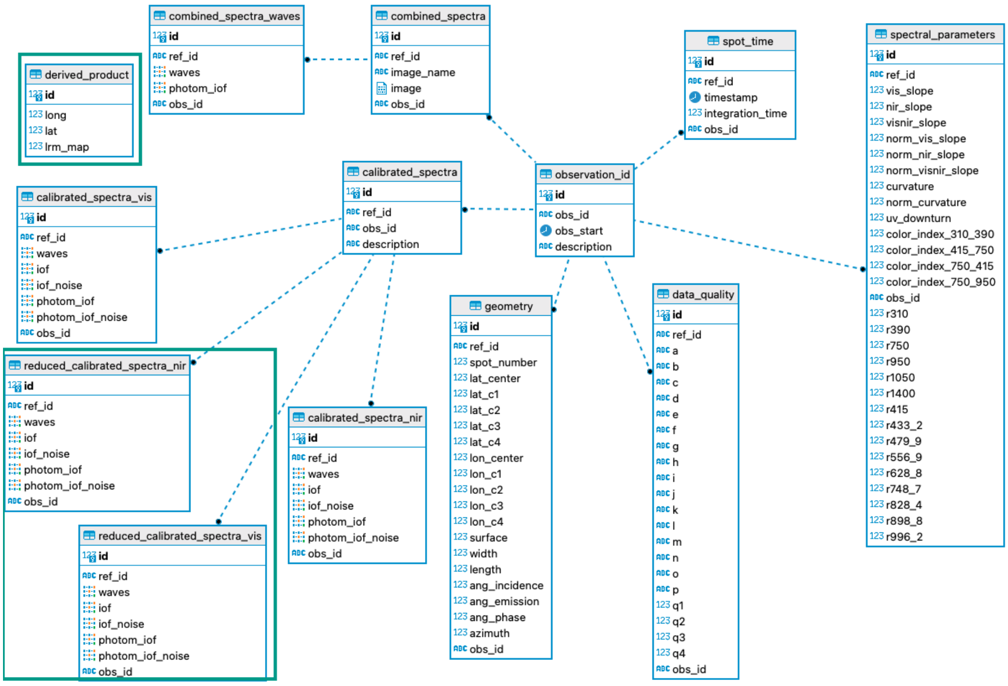


Figure 3.7: New **MASCS** database structure

3.2.1 Reduced values

In order to offer faster queries over the *calibrated_spectra_nir* and *calibrated_spectra_vis* tables, their data need to be reduced by adapting their digit numbers and type from float with 1e-10 precision to saturated values flagged as -999. The data are multiplied by 1e3 to keep the precision small, lighter regarding to storage size.

In a first place, the two *reduced_calibrated_spectra_nir* and *reduced_calibrated_spectra_vis* tables has been initialized into PostgreSQL language thanks to the code in Appendix 6.2.1. The structure is the same as the one of *calibrated_spectra_nir* and *calibrated_spectra_vis* tables in order to let the possibility to use the detailed data for a better precision or reduced data for a faster query. The two tables have quite the same structure which is composed of the data for each instrument in the columns *waves*, *iof*, *photom*, which are respectively wavelength, reflectance and reflectance corrected for the different viewing geometries, *iof_noise*, *photom_noise*, which contains additionally the noise of the previous data. The *id*, *ref_id* and *obs_id* are respectively the row identifier in the database, the footprint identifier and the orbit identifier of the spacecraft (*Figure* 3.8).

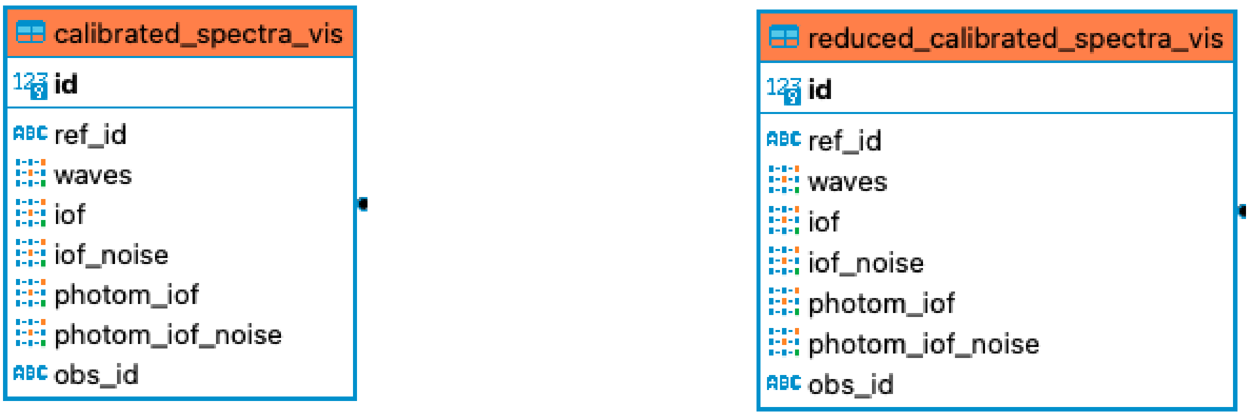


Figure 3.8: Calibrated spectra tables for VIS instrument

Once the empty tables had been created, it was a matter of recovering the data from the two previous tables, then adapting them in an appropriate form and adding them to the database. The code in Appendix 6.2.2 allows to connect to the database first with the function seen previously, then to make a SQL query on the current database to obtain the data to be reduced:

SELECT DISTINCT o.obs_id FROM observation_id o INNER JOIN calibrated_spectra_vis v ON v.obs_id = o.obs_id

It separately selects the orbits concerning the VIS and NIR instruments in a variable *df_vis_global*. Then, looping over this list, the pattern is the same for the VIS and NIR instruments. First, the following query is performed:

SELECT * FROM calibrated_spectra_vis WHERE obs_id like obs_id

It selects only those orbits where the VIS data were available. Then the data concerned are selected and adapted to the form from -999 to 999. Due to the multiplication by a factor of 1e3, the stored data must be divided by the same amount before being used as a post-processing step, after the query. They are then added to the database using the function *insert_into_reduced_calibrated_table* in the Appendix 6.2.3. This function fills the table according to the following SQL command line:

INSERT INTO reduced_calibrated_spectra_vis (ref_id, waves, iof, iof_noise, photom_iof, photom_iof_noise, OBS_ID)
VALUES (ref_id, waves_array, iof_array, iof_noise_array, photon_iof_array, photom_iof_noise_array, obs_id)

It inserts into the new table *reduced_calibrated_spectra_vis* the data recovered and modified previously in the columns which make up the table.

The code is quite similar for the NIR instrument as only the name *vis* is changed to *nir*.

3.2.2 Derived products

In order to continuously making research more convenient, some paper results could be compared to other data set in order to find, for instance, correlation between terrains identified within previous research works and our MASCS data.

An important goal of Mercury research science is to determine the composition of its surface. The Low Reflectance Materials (LRM) are quite an interesting aspect to maybe be able to explain presence of carbon on cratered terrains.

In a paper by [4], the global distribution and spectral properties of the LRM were studied on Mercury and presented in the form of maps (*Figure* 3.9). These map are composed of almost 4.5 million pixels and covers all Mercury surface. The data has been recovered by the instrument MDIS[1] on board the MESSENGER spacecraft. The aim is to import such map into the MeSS database architecture to provide at a same time a data set that can be used to extract MASCS spectral data where LRM are present and a new instrument data set that can be compared to the previously added one.

[1]MDIS: Mercury Dual Imaging System

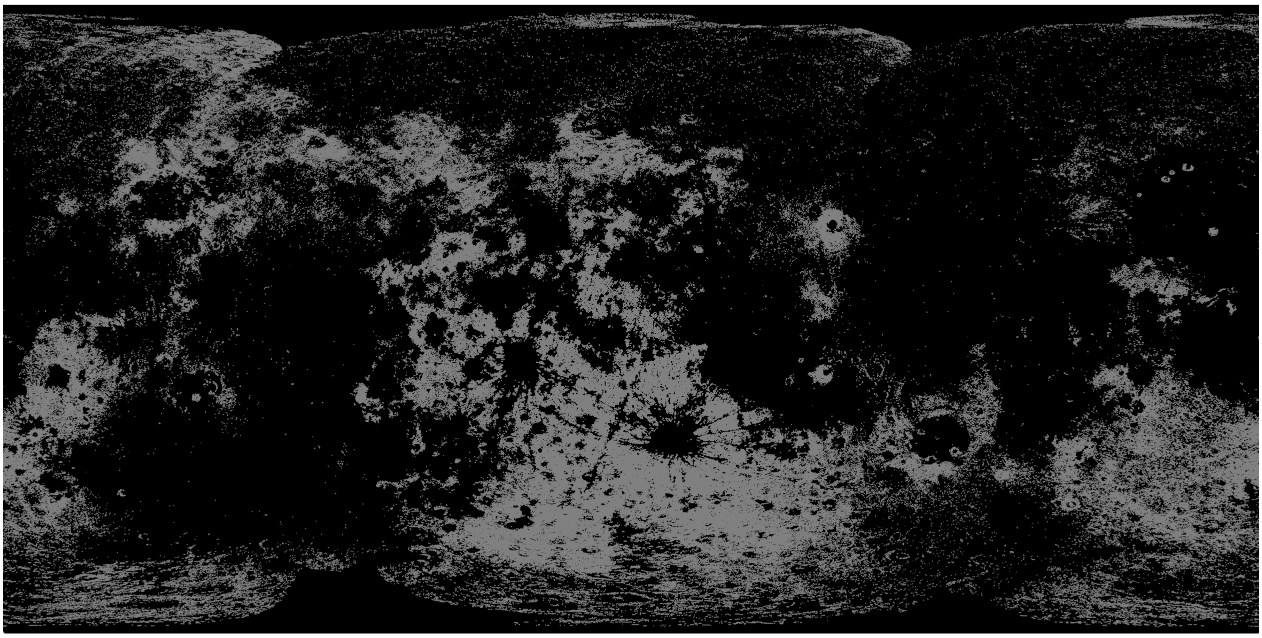


Figure 3.9: Low Reflectance Materials map [4]

To import such a map into the database, several steps are necessary. The file linked to the article is a *.tiff* file which must be georeferenced thanks to softwares such as ArGis Pro or GDal. Once georeferenced to accurately take into account geographic projection effects, the map can be exported to record the coordinates and map values for each pixel. The outgoing file is a *.csv* (*Figure* 3.10) that consists of four columns: an row identifier *OID*, longitude *X* and latitude *Y* of each pixel and their value, *0* for no low reflectance material, *1* otherwise.

```
OID_,X,Y,Value
0,-179.939999999999998,89.939999999999998,0
1,-179.819999999999993,89.939999999999998,0
2,-179.699999999999989,89.939999999999998,0
3,-179.580000000000013,89.939999999999998,0
4,-179.460000000000008,89.939999999999998,0
5,-179.340000000000003,89.939999999999998,0
6,-179.219999999999999,89.939999999999998,0
7,-179.099999999999994,89.939999999999998,0
8,-178.979999999999990,89.939999999999998,0
9,-178.860000000000014,89.939999999999998,0
```

Figure 3.10: CSV file for Low Reflectance Material map

In a first place, the new table *derived_product* as been initialized into PostgreSQL language thanks to the code in Appendix 6.2.1. The table contains the id reference for each pixel, the coordinates of pixels in the columns *lat* and *long* which are respectively the latitude and longitude and the value of the concerned pixel in the column *LRM_map* (*Figure* 3.11 *left*).

123 id	123 long	123 lat	123 lrm_map
3,164	-169.020004272	87.059997559	0
3,165	-168.899993896	87.059997559	0
3,166	-168.779998779	87.059997559	1
3,167	-168.660003662	87.059997559	1
3,168	-168.539993286	87.059997559	1

Figure 3.11: Low Reflectance Materials table characteristics: table structure (left) and its values (right)

The code in Appendix 6.3.2, first obtained the values of each pixel by reading the csv file. Then, the values are stored in the list *derived_product* and used in the function *insert_into_derived_product_table* detailed in Appendix 6.3.3. This function converts the latitude and longitude to a float and the pixel value to an integer. These values are then included in the database using the SQL command line:

INSERT INTO derived_product (lat, long, lrm_map) VALUES (lat, long, LRM_map)

It inserts into the table *derived_product* the data recovered in the columns which make up the table. The table *derived_product* has now the following form (*Figure* 3.11 *right*).

Chapter 4

Results

Through the developed methods in Part 3.2, the data are now usable in the MeSS project as additional data sets. The following part highlights the improvements these data sets bring and the added value they provide to users.

4.1 Reduced values

Reducing these values brought two significant improvements to the database: the size of the content and the query time were reduced.

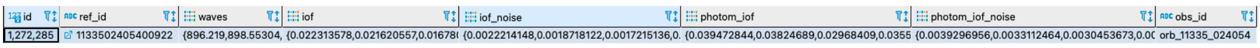
Content size

These modifications allow to reduce the size of the data set as the Figure 4.1 shows. The tables has been approximately reduced from 6.9 GB to 3.3 GB for the NIR instruments and from 14 GB to 4.9 GB for the VIS instrument. This space saving is important for future data sets that are supposed to be added to the project.

calibrated_spectra_nir	6.9G	reduced_calibrated_spectra_nir	3.3G
calibrated_spectra_vis	14G	reduced_calibrated_spectra_vis	4.9G

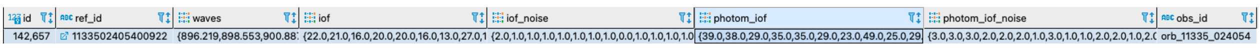
Figure 4.1: Size comparison for the calibrated spectra tables

The appearance of the data form is displayed on the figures 4.2 and 4.3. For the same orbit ('*orb_11335_024054*') and footprint ('*1133502405400922*') for the NIR instruments, it is possible to highlight the effects of the modification to reduced values. Only the four columns *iof*, *iof_noise*, *photom_iof* and *photom_iof_noise* have been affected by the reduction. For instance, the first value of *iof* was changed from 0.022313578 to 22.0. The first value of *photom_iof* has also been modified from 0.039472844 to 39.0.



id	ref_id	waves	iof	iof_noise	photom_iof	photom_iof_noise	obs_id
1,272,285	1133502405400922	{896.219,898.55304,	{0.022313578,0.021620557,0.01678(	{0.0022214148,0.0018718122,0.0017215136,0.	{0.039472844,0.03824689,0.02968409,0.0355	{0.0039296956,0.0033112464,0.0030453673,0.0(	orb_11335_024054

Figure 4.2: Data form for *iof* columns of detailed values for the NIR instrument



id	ref_id	waves	iof	iof_noise	photom_iof	photom_iof_noise	obs_id
142,657	1133502405400922	{896.219,898.553,900.88	{22.0,21.0,16.0,20.0,20.0,16.0,13.0,27.0,1	{2.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,0.0,1.0,1.0,1.0,1.0	{39.0,38.0,29.0,35.0,35.0,29.0,23.0,49.0,25.0,29	{3.0,3.0,3.0,2.0,2.0,2.0,1.0,3.0,1.0,1.0,2.0,2.0,1.0,2.(	orb_11335_024054

Figure 4.3: Data form for *iof* columns of reduced values for the NIR instrument

The figures 4.4 and 4.5 show the two different tables (detailed and reduced) for two specific footprints. The query over the database was done with the code in Appendix 6.2.4. In each case, when displaying the reflectance in terms of the wavelength, the data are consistent. The rounding of data creates some difference over a few points but the general aspect is still the same.

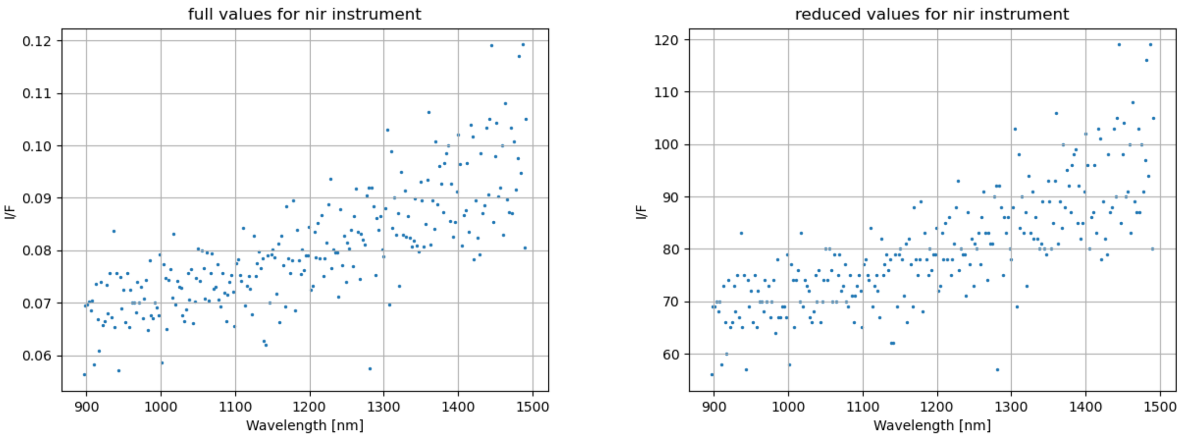


Figure 4.4: Comparison between detailed and reduced values for NIR instrument (set 1)

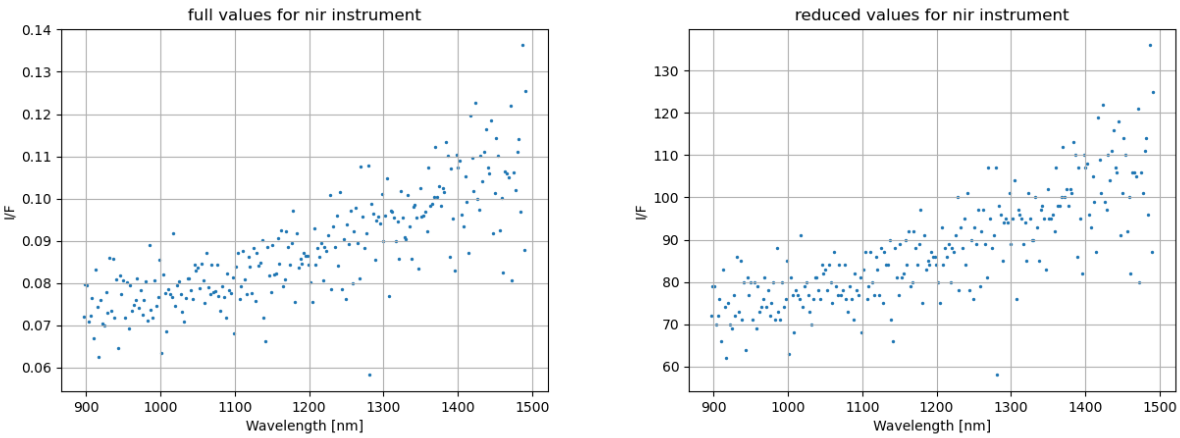


Figure 4.5: Comparison between detailed and reduced values for NIR instrument (set 2)

The reduced values tables offer researchers a choice between more accurate data and longer search time or faster search time at the expense of less accurate data.

Query time

The query time has also been reduced for the both set of examples seen above (*Figures* 4.4 *and* 4.5) as detailed in the Table 4.1. The values still seems to be quite similar but the query has been done over a a little data sample, about 6000 records. For larger ones, difference is more noticeable.

To further improve the query time, the data might be converted to integers. This format is lighter, could free up space in the database and reduce the time needed. In the future, the MeSS project will contain much more data and the queries will become increasingly complex.

	set 1	set 2
Detailed values	2.769 sec	2.721 sec
Reduced values	2.187 sec	2.472 sec

Table 4.1: Query times for two data sets

4.2 Derived products

In addition to the data reduction, the MeSS has been improved in terms of content. The addition of research result such as the Low Reflectance Materials map is a step forward in planetary research. Instead of making comparisons by hands, researchers are now able to select a specific area to study its LRM composition in detail and to compare it to different data sets.

It is possible to make a query on a specific area such as the Rachmaninoff crater in the northern hemisphere of Mercury (*Figure* 4.6). The location of the crater can be selected from the coordinates:

minimum latitude	16
maximum latitude	35
minimum longitude	43
maximum longitude	70

Table 4.2: Rachmaninoff crater location

Because of the georeferencing, the coordinates are the same between the LRM map and the database coordinates.

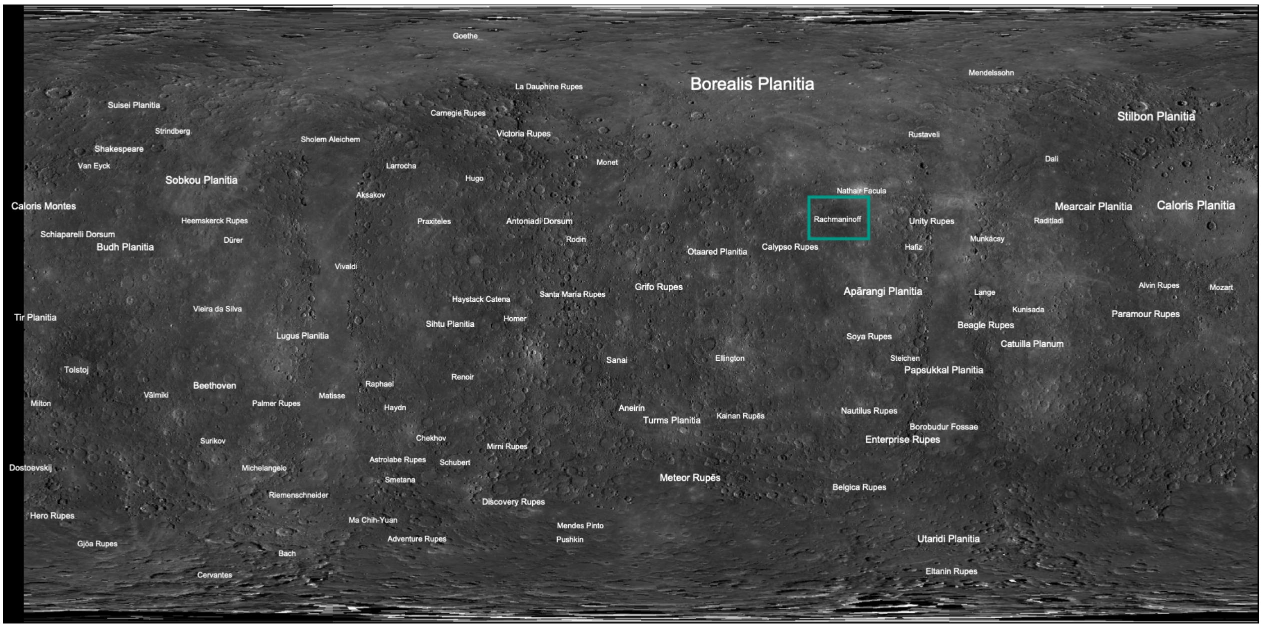


Figure 4.6: Rachmaninoff crater location *(source: MESSENGER Quickmap)*

According to the LRM map, this crater is surrounded by low reflectance materials (*Figure* 4.7 *left*). The same result can be verified by doing a query over the database, thanks to code in Appendix 6.5. Once this query is done, it is possible to display the low reflectance materials on the selected area (*Figure* 4.7 *right*). Compared to the LRM map itself, the database map is the same and can be adapted to any area of Mercury.

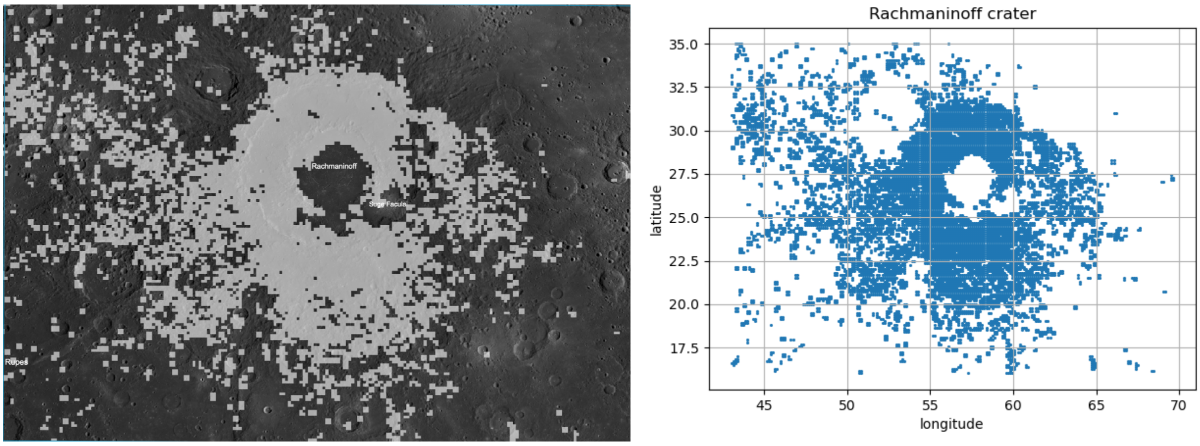


Figure 4.7: LRM of Rachmaninoff crater: LRM map [4] (left) Result of a query on the database (right)

The next main objective is to be able to compare map of such materials or components in terms of spectral behaviour in the MASCS data (visible slope, near-infrared slope etc) and highlight the possible correlation over a map of a specific area with the presence (or not) of different spectral elements.

To reach this goal, a first step is under development. The *derived_product* table must be linked with the footprint of the MASCS intrument. In this way, the presence of LRM in a specific footprint during a specific orbit could be confirmed or not.

Chapter 5

Conclusion

5.1 MeSS project

The MeSS project is a breakthrough regarding to the Mercury research. Being able to link and retrieve within seconds data acquired by different instruments over the full MESSENGER mission duration in an unique query according to a specific area is a helpful support for researchers. The database can be used to extract and map, for example, spectral parameters or specific reflectance values at a given wavelength for direct interpretations. It can also now be used more efficiently to access the original data sets thanks to my contribution in the creation of the reduced data tables. It is now possible to compare data from different sources and to find or eliminate potential correlations faster. Moreover the data can be adapted according to the research's needs. It is possible to choose between a more accurate data set, regardless of the computation time, or a faster query time for less accurate data. The project is able to adapt to any type of request and is always open to adding new data sets, as I demonstrated with the inclusion of the Low Reflectance Material map, which makes it essential for the current Mercury research project.

Thanks to this internship, I was able to develop my computer skills in Python and PostgreSQL in a more practical way and to develop codes in a complex environment and to optimise them according to their use. Being part of a project such as BepiColombo is an opportunity to get closer to the scientific team, the role of everyone and to understand how such projects are managed.

These few months at ESAC[2] have been conducive to the development of my international and scientific knowledge by following weekly meetings on the evolution of the BepiColombo project and weekly Science talks about the current research project new results. I had also the chance to follow on site events provided by ESA such as the Gaia data release.

5.2 Prospects

The first few months were essential to understand the structure of the project and how the data can be adapted for research. The MeSS database still need to be improved in order to become an essential tool for planetary researchers. In the next few months, some aspects could be interesting to develop such as the content of the database, its efficiency and the preparation for the BepiColombo observations.

Content

For now, the MeSS is composed essentially on the MASCS instrument's data. In order to get a larger overview of the science on Mercury, some other instruments of the MESSENGER spacecraft can be included in. The most interesting ones are the following:

- The MDIS[3] device is composed of two cameras, a multispectral Wide Angle Camera (WAC) and a monochrome Narrow Angle Camera (NAC). This instrument that mapped land forms, tracked variations in surface spectra and gathered topographic information. MDIS instrument can be useful to get the context by comparing image and data plot obtained, for instance, with the MASCS data set. This database is in development.
- The MLA[4] device determined the planet topography. This instrument can provide data for detailed analysis of the already existing data set.

[2]ESAC: European Space Astronomy Center
[3]MDIS: Mercury Dual Imaging System
[4]MLA: Mercury Laser Altimeter

Planetary research is not only depending on the data get by spatial mission. A large part of the research is also done in laboratory. Laboratory research is useful for example to prove the presence of a specific material and the chemical reactions that may occur. Being able to compare such result to the data already provides by the database would be a step forward in understanding Mercury.

Efficiency

Improve the efficiency is still an important aspect for the database and a time saving for the researchers. The performance is depending on how the table are developed and linked, how the data are stored (type, digits), of the quantity of data contains in the database. A significant improvement can range from seconds to tens of minutes.

By adding data sets on the project, numbers of indexes and relations will drastically increase. In order to keep a sustainable use, the data format and database structure need to be optimised. A solution can be to store each instrument in a separated database, such as for **MASCS** and **MDIS**.

BepiColombo observations

The project is a way to prepare the BepiColombo observations. The spacecraft will give the first results in 2026 and by this time, the database should be sufficiently developed to be able to compared the results of these last years studies to the new BepiColombo data. In addition, the current analysis will help the observations by guiding the study of a specific area or a specific material to be analysed. A strong knowledge of the MESSENGER data will help to fill the gap with BepiColombo.

During the flyby of MESSENGER, with the constraints of the orbits, some areas, such as Nathair or Lermontov craters, require new measurements concerning these missing data. The trajectories of BepiColombo can be chosen according to this lack of knowledge. There are likely other ways to make use of the MeSS to further refine the SIMBIO-SYS observation plan.

Chapter 6

Appendix

The following codes are the result of joint work with the MeSS team. For confidential reasons, the 'X' will replace the true values.

6.1 Connection function

This function allows to connect to the database. It is used by most of the codes below.

```
def connect(config_connection):
    """
    Connects to postgres handling connection errors by using the try+catch statement.
    in case of error stop execution.
    :param config_connection a python dictionary defining user,password, database,host,
    :return None
    """
    try:
        # connect to the PostgreSQL server
        database = config_connection['database']
        logging.info('Connecting to the PostgreSQL database {}...'.format(database))
        cnx = postgres.connect(**config_connection)

        return cnx

    except (Exception, postgres.DatabaseError()) as error:
        logging.error(error)
        logging.error('Exit since cannot connect to DB')
        sys.exit()
```

6.2 Reduced values

6.2.1 Table creation for reduced values

This code allows to create the table *reduced_calibrated_spectra_vis* in an SQL console. The table *reduced_calibrated_spectra_nir* is using the same pattern.

```
CREATE TABLE reduced_calibrated_spectra_vis(
    id SERIAL PRIMARY KEY,
    ref_id varchar(255) NOT NULL,
    waves real[] NOT NULL,
    iof int[] NOT NULL,
    iof_noise int[] NOT NULL,
    photom_iof int[] NOT NULL,
    photom_iof_noise int[] NOT NULL,
    obs_id varchar(255) NOT NULL,
    CONSTRAINT ref_id
    FOREIGN KEY (ref_id)
    references calibrated_spectra(ref_id)
    ON DELETE CASCADE
```

```
);
```

6.2.2 Modification of reduced values

This code allows to import the reduced values for the BIR and VIS instruments in new tables previously created.

```
import sys
import logging
import pandas as pd
import numpy as np

import psycopg2 as postgres

import psycopg2.extensions

psycopg2.extensions.register_type(psycopg2.extensions.UNICODE)
psycopg2.extensions.register_type(psycopg2.extensions.UNICODEARRAY)

from mascs_importer.postgres.mascspostgresconnector import MascsPostgresConnector

if __name__ == '__main__':

    config_connection = {'host': 'X',
                         'user': 'X',
                         'database': 'X',
                         'password': 'X',
                         'port': X
                         }

    connection = connect(config_connection)
    db = MascsPostgresConnector(config_connection=config_connection)

    print('\nObservation_id contents\n')
    df_vis_global = pd.read_sql_query("select distinct o.obs_id from observation_id
    o inner join calibrated_spectra_vis v on v.obs_id = o.obs_id ", con=connection)

    list_obs_id = df_vis_global['obs_id'].to_list()

    for obs_id in list_obs_id:

        print('obs_id: {}'.format(obs_id))
        print('VIS')

        df_vis = pd.read_sql_query("SELECT * FROM {} WHERE {} like
        '{}'".format('calibrated_spectra_vis', 'obs_id', obs_id), con=connection)

        reduced_values_vis = []

        for i in range(df_vis.index.size):

            ref_id = df_vis['ref_id'][i]
            waves = df_vis['waves'][i]
            iof = np.clip((df_vis['iof'][i]), -999, +999)
            iof_noise = np.clip(df_vis['iof_noise'][i], -999, +999)
            photom_iof = np.clip(df_vis['photom_iof'][i], -999, +999)
```

```
            photom_iof_noise = np.clip(df_vis['photom_iof_noise'][i], -999, +999)

            # Reduce contents translating float to integer and cutting
            # For instance if stored values are * 1000, values must be divided by
            1000 when extracted
            iof = [int(v * 1E03) for v in iof]
            iof_noise = [int(v * 1E03) for v in iof_noise]
            photom_iof = [int(v * 1E03) for v in photom_iof]
            photom_iof_noise = [int(v * 1E03) for v in photom_iof_noise]

            reduced_values_vis = [ref_id, waves, iof, iof_noise,
                                  photom_iof, photom_iof_noise, obs_id]

            db.insert_into_reduced_calibrated_table(reduced_values_vis,
                                                    'reduced_calibrated_spectra_vis')

        print('NIR')

        df_nir = pd.read_sql_query("SELECT * FROM {} WHERE {} like
        '{}'".format('calibrated_spectra_nir', 'obs_id', obs_id), con=connection)

        reduced_values_nir = []

        for i in range(df_nir.index.size):
            ref_id = df_nir['ref_id'][i]
            waves = df_nir['waves'][i]
            iof = np.clip((df_nir['iof'][i]), -999, +999)
            iof_noise = np.clip(df_nir['iof_noise'][i], -999, +999)
            photom_iof = np.clip(df_nir['photom_iof'][i], -999, +999)
            photom_iof_noise = np.clip(df_nir['photom_iof_noise'][i], -999, +999)

            # Reduce contents translating float to integer and cutting
            # For instance if stored values are * 1000, values must be divided by
            1000 when extracted
            iof = [int(v * 1E03) for v in iof]
            iof_noise = [int(v * 1E03) for v in iof_noise]
            photom_iof = [int(v * 1E03) for v in photom_iof]
            photom_iof_noise = [int(v * 1E03) for v in photom_iof_noise]

            reduced_values_nir = [ref_id, waves, iof, iof_noise,
                                  photom_iof, photom_iof_noise, obs_id]

            db.insert_into_reduced_calibrated_table(reduced_values_nir,
'reduced_calibrated_spectra_nir')

connection.close()
```

6.2.3 Importation of reduced values

This code allows to import the reduced values in the new tables previously created.

```
def insert_into_reduced_calibrated_table(self, values, table_name):
    """
    Insert values corresponding to reduced_data in table
    """
```

```
try:

    cursor = self.cnx.cursor()

    col_names = '(ref_id, waves, iof, iof_noise, photom_iof,
                photom_iof_noise, OBS_ID)'
    ref_id = values[0]
    obs_id = values[-1]
    waves_array = values[1]
    iof_array = values[2]
    iof_noise_array = values[3]
    photon_iof_array = values[4]
    photom_iof_noise_array = values[5]
    my_values = "('{}', ARRAY{}, ARRAY{}, ARRAY{}, ARRAY{}, ARRAY{},'{}')".format
                (ref_id, waves_array, iof_array, iof_noise_array,
                photon_iof_array, photom_iof_noise_array, obs_id)

    # logging.debug('insert into {} {} values {}'.format
                (table_name, col_names, my_values))
    cursor.execute(sql.SQL("insert into {} {} values {}".format
                (table_name, col_names, my_values)))

    self.cnx.commit()

except (Exception, postgres.DatabaseError) as error:
    print(error)
    sys.exit(1)
```

6.2.4 Query over reduced values

This code allows to make a query over the database. It is using a external class, called Pymess, which cannot be shared in this report for reasons of confidentiality.

```
import matplotlib.pyplot as plt
from mascs_queries import mascs_queries as mq
import sys

mess = mq.Pymess(typeconn='esac-test')

out = mess.query(minlat=0, maxlat=20, minlon=0, maxlon=20,
query_tables=['geometry', 'data_quality', 'spectral_parameters',
              'spot_time', 'reduced_calibrated_spectra_nir'])

plt.figure()
plt.scatter(x=out['reduced_calibrated_spectra_nir']['waves'].values[2],
            y=out['reduced_calibrated_spectra_nir']['iof'].values[2],
            s=2)

plt.ylabel('I/F')
plt.xlabel('Wavelength [nm]')

plt.title('reduced values for nir instrument')
plt.grid()
plt.savefig('reduced_values_nir')
```

6.3 LRM Map

6.3.1 Table creation for LRM map

This code allows to create the table *derived_product* in an SQL console.

```
CREATE TABLE derived_product(
        id SERIAL PRIMARY KEY,
        ids int NOT NULL,
    lat real NOT NULL,
    long real NOT NULL,
    lrm_map int NOT NULL
);
```

6.3.2 Importation of LRM map

This code allows to import the LRM map values in a new table previously created. It use the code 6.3.3 above.

```
import sys
import logging
import csv
import psycopg2 as postgres
import psycopg2.extensions

psycopg2.extensions.register_type(psycopg2.extensions.UNICODE)
psycopg2.extensions.register_type(psycopg2.extensions.UNICODEARRAY)

from mascs_importer.postgres.mascspostgresconnector import MascsPostgresConnector

if __name__ == '__main__':

    config_connection = {'host': 'X',
                         'user': 'X',
                         'database': 'X',
                         'password': 'X',
                         'port': X
                         }

    connection = connect(config_connection)
    db = MascsPostgresConnector(config_connection=config_connection)

    file = open('con_sc_1bit_1band_sample_concentrated_lrm_560_params_bw.csv')

    data = csv.reader(file)

    derived_product = []

    next(data)

    for row in data:
        derived_product.append(row)

    db.insert_into_derived_product_table(derived_product, 'derived_product')
```

```
file.close()
connection.close()
```

6.3.3 LRM importation function

This function is the developed function to include the LRM map data into the new table previously created.

```
def insert_into_derived_product_table(self, values, table_name):
    """
    Insert values corresponding to derived_product in table
    """

    try:

        cursor = self.cnx.cursor()

        col_names = '(lat, long, lrm_map)'

        for row in values:
            lat = float(row[1])
            long = float(row[2])
            LRM_map = int(row[3])

            my_values = "('{}', '{}', '{}')".format(lat, long, LRM_map)

            # logging.debug('insert into {} {} values {}'.format
                             (table_name, col_names, my_values))
            cursor.execute(sql.SQL("insert into {} {} values {}".format
                            (table_name, col_names, my_values)))

        self.cnx.commit()

    except (Exception, postgres.DatabaseError) as error:
        print(error)
        sys.exit(1)
```

6.4 Part of the query function

This function allows to make a query over the database. It is used in the code 6.5 for the LRM map of the Rachmaninoff crater.

```
def query_derived_product(self, minlat=None, maxlat=None, minlon=None, maxlon=None,
export_output=False, query_tables=['derived_product']):
    """
    Performs a query based on a set of parameters used to subset the data.

    :param minlat: minimum latitude
    :param maxlat: maximum latitude
    :param minlon: minimum longitude
    :param maxlon: maximum longitude
    :param query_tables: List of the tables that will be queried
    (name re-used for the output)
    :keyword export_output: True or False to write dat file with results
```

```
:returns: The routine returns a dictionary of dataframes containing:
    -
"""

# First build the selection criteria string that will be used
select = ''
if minlat is not None:
    select = " ".join((select, 'and derived_product.lat>{}'.format(minlat)))
if maxlat is not None:
    select = " ".join((select, 'and derived_product.lat<{}'.format(maxlat)))
if minlon is not None:
    select = " ".join((select, 'and derived_product.long>{}'.format(minlon)))
if maxlon is not None:
    select = " ".join((select, 'and derived_product.long<{}'.format(maxlon)))
select = " ".join((select, 'and derived_product.lrm_map = 1'))

# remove the first "and" from the select string
select = " ".join(('where', select[4:]))

# prepare the base queries on the different tables to which
criteria will be appended
all_queries = {}
for t in query_tables:

    if t.find('derived_product') != -1:
        base = '''select derived_product.* from derived_product'''

    all_queries[t] = " ".join((base, select))#, 'ORDER BY derived_product.id'))

out = {}
for t_q in list(all_queries.keys()):
    log.info('Doing query on {}'.format(t_q))  # all_queries[t_q])
    out[t_q] = pd.read_sql_query(all_queries[t_q], con=connection)
    log.info('{} query done: {} records'.format(t_q, out[t_q].shape[0]))

# Export the output if required
(one file per table, + 1 note per file which reports the query done)
if export_output:
    for t_q in out:
        query_str = " ".join((all_queries[t_q], select, 'ORDER BY
        geometry.ref_id'))
        out[t_q].to_csv(os.path.join(self.dl_url,
                                     'MeSS_{}_lon{}_{}_lat{}_{}.dat'.format
                                     (t_q, minlon, maxlon, minlat, maxlat)))
        with open(os.path.join(self.dl_url,
                               'MeSS_{}_lon{}_{}_lat{}_{}_note.dat'.format
                               (t_q, minlon, maxlon, minlat, maxlat)), 'w+')
                               as f:
            f.write('Query performed: \n')
            f.write(query_str)
        f.close()
        log.info('Stored MeSS database outputs in %s', os.path.join
        (self.dl_url, 'MeSS_{}_lon{}_{}_lat{}_{}.dat'.format
        (t_q, minlon, maxlon, minlat, maxlat)))
        log.info('Stored MeSS database query in %s', os.path.join
        (self.dl_url, 'MeSS_{}_lon{}_{}_lat{}_{}_note.dat'.format
```

```
            (t_q, minlon, maxlon, minlat, maxlat)))

    return out
```

6.5 Rachmaninoff crater query

This code is an example of a query done on the database in order to plot the LRM map for the Rachmaninoff crater. It is using a external class, called Pymess, which cannot be shared in this report for reasons of confidentiality.

```
import matplotlib.pyplot as plt
from mascs_queries import mascs_queries as mq
import sys

mess = mq.Pymess(typeconn='esac-test')

out = mess.query_derived_product(minlat=16, maxlat=35, minlon=43, maxlon=70,
query_tables=['derived_product'])

plt.figure()
plt.scatter(x=out['derived_product']['long'],
            y=out['derived_product']['lat'],
            s=1, marker='.')

plt.ylabel('latitude')
plt.xlabel('longitude')

plt.title('Rachmaninoff crater')
plt.grid()

plt.savefig('rachmaninoff_crater')

sys.exit()
```

Bibliography

[1] Larry R. Nittler Sean C. Solomon and Brian J. Anderson. Mercury: The View after MESSENGER.

[2] Belton M. J. S. McElroy M. B. Broadfoot A. L., Kumar S. Mercury's atmosphere from mariner 10: Preliminary results. 185:166–169.

[3] David A. Rothery. Volcanism on mercury.

[4] Rachel L. Klima, Brett W. Denevi, Carolyn M. Ernst, Scott L. Murchie, and Patrick N. Peplowski. Global distribution and spectral properties of low-reflectance material on mercury. 45(7):2945–2953.

[5] S. Besse, A. Doressoundiram, and J. Benkhoff. Spectroscopic properties of explosive volcanism within the caloris basin with MESSENGER observations: MERCURY's EXPLOSIVE VOLCANISM. 120(12):2102–2117.

[6] David T. Blewett Sean C. Solomon Lionel Wilson Scott L. Murchie Mark S. Robinson Brett W. Denevi Deborah L. Domingue Laura Kerber, James W. Head. The global distribution of pyroclastic deposits on mercury: The view from messenger flybys 1–3. (4):1895–1909.

[7] David A. Rothery, Océane Barraud, Sebastien Besse, Cristian Carli, David L. Pegg, Jack Wright, and Francesca Zambon. On the asymmetry of nathair facula, mercury. 355:114180.

[8] S. Besse, A. Doressoundiram, O. Barraud, L. Griton, T. Cornet, C. Muñoz, I. Varatharajan, and J. Helbert. Spectral properties and physical extent of pyroclastic deposits on mercury: Variability within selected deposits and implications for explosive volcanism. 125(5).

[9] O. Barraud, A. Doressoundiram, S. Besse, and J. M. Sunshine. Near-ultraviolet to near-infrared spectral properties of hollows on mercury: Implications for origin and formation process. 125(12).

[10] Océane Barraud, Sébastien Besse, Alain Doressoundiram, Thomas Cornet, and Claudio Muñoz. Spectral investigation of mercury's pits' surroundings: Constraints on the planet's explosive activity. 370:114652.

[11] Sebastien Besse, Claudio Munoz, Oceane Barraud, Thomas Cornet, and Alain Doressoundiram. Mercury spectroscopic surface - MeSS - a million spectra database for scientific exploration of mercury's surface.