UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

---

Titulació:

MASTER'S DEGREE IN SPACE AND AERONAUTICAL ENGINEERING

Alumne (nom i cognoms):

ÁLVARO BOIX CANDIL

Enunciat TFM:

STUDY: THE ION PROPULSION SYSTEM AND TRAJECTORY OF DAWN SPACECRAFT.

Director del TFM:

MANEL SORIA GUERRERO

Codirector del TFM:

JOSEP ORIOL LIZANDRA DALMASES

Convocatòria de lliurament del TFM:

QUATRIMESTRE PRIMAVERA CURS 2018/19

I declare that,

the work in this Master Thesis is completely my own work,

no part of this Master Thesis is taken from other people's work without giving them credit,

all references have been clearly cited,

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by the Universitat Politècnica de Catalunya - BarcelonaTECH.

ÁLVARO BOIX CANDIL		20/06/2019
_____	_____	_____
Student Name	Signature	Date

Title of the Thesis:

STUDY: THE ION PROPULSION SYSTEM AND TRAJECTORY OF DAWN SPACECRAFT.

Abstract

La missió Dawn representa un assoliment revolucionari en la investigació espacial, ja que va ser la primera astronau en orbitar dos cossos extraterrestres diferents: els asteroides Vesta i Ceres. Va ser dirigit pel Jet Propulsion Laboratory (JPL) des del seu disseny, llançament al 2007, fins el seu final el 2018. El seu objectiu va ser assolit gràcies al seu Sistema de Propulsió Iònica (IPS), el qual va fer possible les maniobres i trajectòries requerides per aconseguir els objectius de la missió.

Aquesta tesi se centra en l'estudi d'aquest innovador IPS. Específicament posa el focus en el seu funcionament durant el transcurs la missió i en entendre per què opera d'una manera o d'una altra. A més, el treball presenta d'una manera detallada sobretot la maniobra més característica de la missió Dawn al complet: la trajectòria d'escapament de Vesta per viatjar a Ceres.

Per aconseguir això, la biblioteca i el kit d'eines del JPL anomenats SPICE s'utilitza durant totes les computacions com a conseqüència de la gran fiabilitat de la seva font d'informació. D'aquestes dades proporcionades per SPICE, els càlculs son duts a terme en relació amb el IPS de Dawn com la seva trajectòria, força d'empenta (thrust) i, fins i tot, la potència d'entrada donada per les plaques solars de l'aeronau. Un dels objectius d'aquesta tesi és demostrar la veracitat de les dades proporcionades per SPICE així com les seves eines de programació, la qual es confirma després d'analitzar els resultats extrets i comparar-los amb informació corroborada.

A més, juntament amb els càlculs de SPICE, els mateixos paràmetres de l'IPS són estimats utilitzant alguns models teòrics relacionats amb maniobres espacials similars a les de la Dawn. La viabilitat de l'aplicació d'aquests procediments establerts es vol demostrar també en el cas de cada valor determinat.

Després de les computacions, es troba que aquests mètodes són vàlids per la seva similitud absoluta amb els resultats obtinguts amb SPICE. La trajectòria de pujada amb baixa empenta estimada resulta en una trajectòria d'escapament aproximada, i el valor de la força d'empenta donat per un escapament amb baixa empenta és increïblement similar al de SPICE, fins i tot tenint en compte els valors petits de la propulsió iònica en comparació amb els motors d'hidrazina comuns. Per últim, gràcies a la determinació de la potència d'entrada del IPS, donada per la orientació i eficiència de les plaques solars, és possible verificar els valors reals de la força d'empenta, els quals encara milloren els resultats extrets prèviament.

Aquesta tesi significa un primer acostament a l'anàlisi d'una missió espacial i els seus elements, i representa un punt de partida cap a un estudi més divers o més profund dels seus paràmetres.

Abstract

Dawn mission represents a revolutionary achievement in space research due to the fact that it was the first spacecraft to orbit two different extraterrestrial bodies, the asteroids Vesta and Ceres. It was managed by NASA's JPL since its design, its launch in 2007 until its end in November 2018. Its objective was accomplished thanks to its Ion Propulsion System (IPS), which made possible the required manoeuvres and trajectories required to fulfil the mission's objectives.

This thesis focuses on the study of this innovative IPS, specifically on how it works during the mission and trying to understand why it operates in one way or another. Specially, the thesis centres in the most characteristic manoeuvre of the entire Dawn mission, the escape trajectory from Vesta to begin cruise to Ceres.

In order to do this, the JPL's SPICE library and toolkit is employed during the whole computations due to its high reliable source of data. From this information provided by SPICE, some calculations are performed regarding the Dawn IPS such as its trajectory, thrust force and even though its input power given by the spacecraft's solar arrays. One of this thesis's goals is to prove the reliability of this SPICE given data as well as its programming tools, which is confirmed after analysing the extracted results and comparing them to corroborated information.

Furthermore, together with the SPICE computations, the same IPS parameters are estimated employing some theoretical models related to space manoeuvres similar to the ones performed by Dawn. The feasibility of the application of these established procedures is also wanted to be demonstrated in the case of each determined value.

After the calculations, it is found out that these methodologies are validated due to the absolute similarity to the results obtained with SPICE. The estimated low thrust climb trajectory results in a truly approximate escape trajectory, and the thrust force value provided by a low thrust escape is incredibly similar to the SPICE one, even taking into account the definitely low values of ion propulsion compared to the common hydrazine thrusters. Last but not least, thanks to the determination of the IPS input power, given by the solar arrays' orientation and efficiency, it is possible to verify the real values of the thrust force, which even improved the results extracted previously.

This thesis means a first approach to the analysis of a space mission and its elements, and it represents a starting point into a more various or deeper study of the parameters of them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

One of the main contemporary objectives of space research is the exploration of new and unknown bodies, especially the farther ones from the Earth. These planets, satellites or asteroids provide innovative data about different and strange environments in contrast to the Earth, which enlarges the scientific knowledge of mankind.

This project focuses on NASA's Dawn mission (2007-2018), which, apart from this science innovation in the space investigation field, studied the two biggest asteroids in the Solar System's main asteroid belt, which represent very distinctive targets. The particularity of these asteroids, Vesta and Ceres (being the last one also a dwarf planet), is that they were formed in the first phases of the Solar System creation, so they are called as protoplanets.

Furthermore, their contrasting differences may make possible the study of two different regions or eras of the Solar System origin. The analysis of the data of these two protoplanets is probable to be beneficially advantageous for scientists of many fields due to the fact that it may lead to the origin of our planet, the Earth.

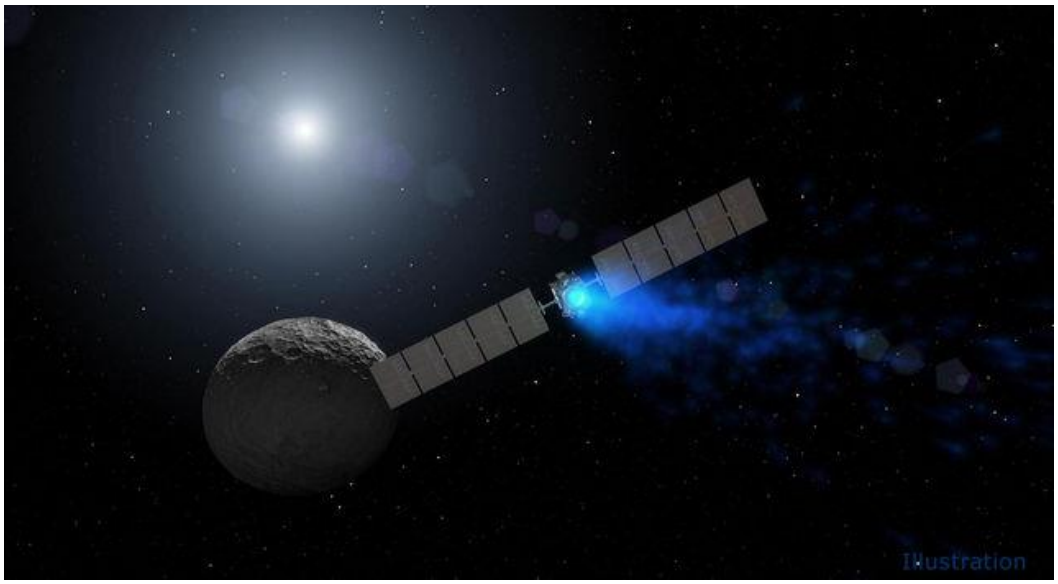


Figure 1. Dawn spacecraft arriving to Ceres asteroid [1].

In order to make this double journey possible as well as the abundant science orbits around the two asteroids, a new and revolutionary ion propulsion system was introduced in the Dawn spacecraft. The low but continuous thrust made possible the necessary manoeuvres to perform in a right way all the desired trajectory along the entire mission.

Nevertheless, the knowledge and work on this type of propulsion is very limited to most of the scientific community. Therefore, this thesis centres on the study

of this Dawn Ion Propulsion System's performance by determining the path carried out by the spacecraft during crucial selected phases of the mission in order to understand in a better way its operation. One of the specific mission stages studied in this project is the escape of Dawn from the gravitational influence of Vesta, the first visited asteroid, which enabled the beginning of the second cruise trajectory to Ceres.

To fulfil all these thesis' goals, a database library and a programming toolkit from NASA's Jet Propulsion Laboratory called both as SPICE, is employed during the entire project. SPICE provides useful data from the mission and the required set of codes to read them for the user. The reliability of SPICE is also one of the objectives of this project.

From all this, it can be established that the procedure of this thesis is getting the data from the SPICE library, reading and computing it by programming and extracting the desired results. Moreover, the project also contains computation processes in order to analyse the extracted data with existing theoretical methodologies to develop a practical study of the Dawn Ion Propulsion System performance. The resultant useful parameters computed in this thesis are the ones related to the trajectory (position and velocity), the thrust force applied, and even the input power of the propulsion system, which is directly determined by the spacecraft's solar arrays.

As the work focuses on the propulsion system of the spacecraft, it does not examines or interpret data supplied by the observation of Vesta and Ceres, such as photos or mappings. This information could be deeply studied in future projects thanks to this first approach to the study of a space mission using SPICE.

Because of this, the scope of this thesis is centred in the investigation of the operation of the Dawn Ion Propulsion System by employing the public SPICE database and toolkit provided by JPL. The only requirements for the development of this project are having previously acquired the basic knowledge about ion propulsion and the Dawn mission, and the use of a computer connected to Internet that includes a programming software to create and execute the necessary computing codes.

CHAPTER 2. DAWN SPACECRAFT

Dawn comprises an innovative spacecraft whose purposes mean an absolute revolution for the current and future development of the space research. Its goals can be clearly identified although they work together to accomplish the established objectives. These goals are mainly two, and they will constitute the two main blocks of this section. These goals are:

- ✓ A journey to the two largest asteroids of the main asteroid belt, **Ceres and Vesta**. It also includes their observation and study to acquire more knowledge about these bodies, which could bring some clues about the past of the Earth.

- ✓ The implementation of an **ion propulsion system** in a non-test mission. It is in charge of all the required spacecraft propulsion in order to perform the mission in the right way.

2.1 Mission overview

Dawn is a space mission managed by the Jet Propulsion Laboratory (JPL) for the National Aeronautics and Space Administration (NASA) that belongs to the Discovery Program. This program comprises a set of scientific space missions which explore the Solar System and contains a low-cost budget, in contrast to New Frontiers or Flagship programs.

Figure 2. Dawn mission logo [2].

One characteristic about this Discovery program is that missions are proposed by independent scientific teams and then are selected to become part of the program. After the NASA mission Deep Space 1 proved in 1998 the capability of performing a fly-by of an asteroid and a comet employing a new electrostatic ion thruster called NASA Solar Technology Application Readiness (NSTAR) propelled by Xenon, a group of scientists led by Christopher T. Russell

presented the Dawn mission. Then, in 2001, Dawn and Kepler missions were finally selected for the Discovery program [3].

After some changes in the mission and many cancellations, Orbital Sciences Corporation, which was the developer and manufacturer of the spacecraft, assumed all the cost in 2006. Finally, the mission was not cancelled and was rescheduled for 2007, when the spacecraft was launched and the Dawn journey began.

Dawn mission was able to reach two new revolutionary achievements, which made it unique and the first step of the great future space research. First of all, it was the first spacecraft to orbit two extraterrestrial bodies, Vesta and Ceres. Both are considered asteroids and protoplanets although Ceres is also a dwarf planet. These categories make these bodies ideal candidates to study the origin of the Solar System.

Until now, the past chemical-powered propulsion systems made possible to perform several fly-bys around some Solar System's bodies or to orbit a single one, but Dawn was able to travel to Vesta, orbit it during a long period of time, and then escaping from its gravitational influence to travel to Ceres and orbit it in the same way as Vesta.

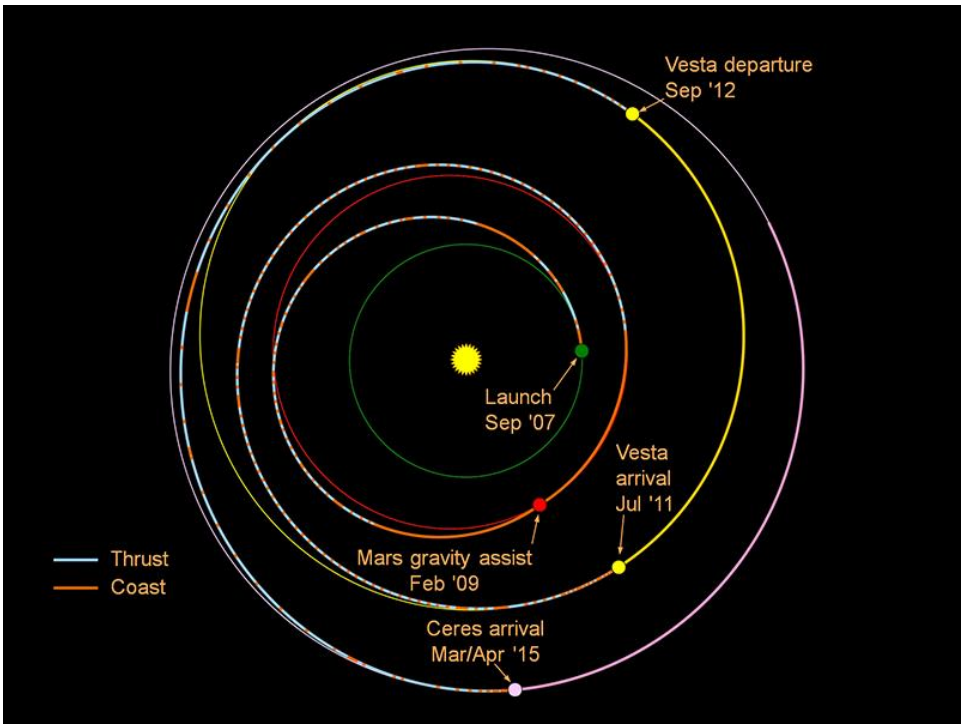


Figure 3. Dawn mission total trajectory [4].

This means a truly feat due to the small size of these bodies and, consequently, poor gravity force exerted by them. However, it also became an advantage when it came to the escape manoeuvre in Vesta. In order to give Dawn the required impulse to arrive to Vesta, the spacecraft did a gravity assist when it approached to Mars orbit [5].

The other important milestone achieved by Dawn was being the first to visit a dwarf planet when it arrived to Ceres. It just was reached four months before the known New Horizons mission got to Pluto [6]. The mission key dates are summarised chronologically below:

1) 27th September 2007 → Launch.
2) 17th February 2009 → Mars gravity assist.
3) 16th July 2011 → Vesta arrival.
4) 5th September 2012 → Vesta departure.
5) 6th March 2015 → Ceres arrival.
6) 1st July 2016 → End of main mission and start of extended mission.
7) 1st November 2018 → End of Dawn mission.

As it is seen in this summary, the spacecraft finished its main mission in Ceres in 2016 and some extended mission options were analysed and discussed. Finally, Dawn remained orbiting around Ceres orbit until it ran out hydrazine propellant, employed for the Attitude Control System (ACS) of the spacecraft. Due to this, the solar panels could not be oriented towards the Sun and the antenna was not able to point to the Earth, so the communications stopped. Therefore, on the 1st November 2018, Dawn said “good night” and began a silent stay around Ceres that will last at least 20 years [7].

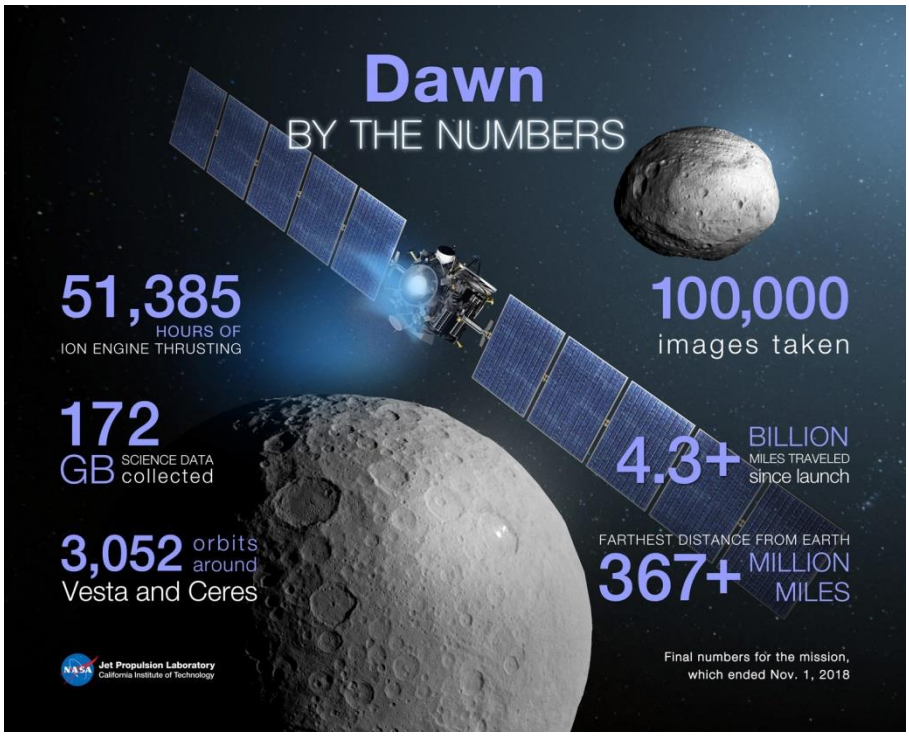


Figure 4. Dawn achievements during its whole mission [8].

As it is exposed in the figure above (information from the 1st November 2018, just at the mission end), since its launch, Dawn mission has brought to the scientific community a completely useful collection of data that has been, are and will be employed to extract valuable analysis and conclusions. Their study provides knowledge about not only the asteroids themselves but also the origin and past of the Solar System, especially the Earth.

In order to obtain all this worthwhile information, a set of scientific instruments were integrated in the spacecraft as payload. These instruments are a framing camera (FC), a visible and infrared spectrometer (VIR) and a Gamma Ray and Neutron Detector (GRaND) [3]. One of the mission phases when these payload instruments came into play was the orbiting around Vesta, which is also a stage that this project focuses on.

The orbits around Vesta used for data obtaining are called as scientific orbits, which are four [9]. These four orbits encompass not only different altitudes with respect to Vesta but different orbital planes.

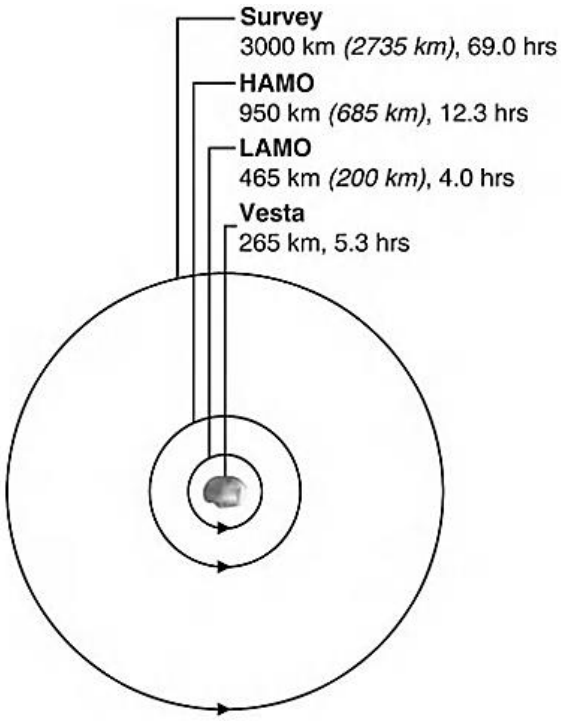


Figure 5. Scientific orbits around Vesta performed by Dawn [3].

These scientific orbits exposed in the figure above are:

- **Survey orbit** → 11th – 31st August 2011.
- **HAMO-1 (High Altitude Mapping Orbit)** → 29th September – 2nd November 2011.
- **LAMO (Low Altitude Mapping Orbit)** → 12th December 2011 – 1st May 2012.

- **HAMO-2** → 15th June – 25th July 2012.

From the dates of these orbits, it is clearly extracted that LAMO orbit was the one used for most of the data collection due to its long duration compared with the other. Furthermore, HAMO orbit is carried out twice because in the second orbit its inclination angle is changed. However, this is going to be checked later in the computations chapter.

2.2 Ion Propulsion System

The Dawn Ion Propulsion System (IPS) is the element that most characterises the whole spacecraft. Consequently, the IPS is what makes Dawn capable to perform the innovative manoeuvres and trajectories that differentiate it from other missions during the whole journey.

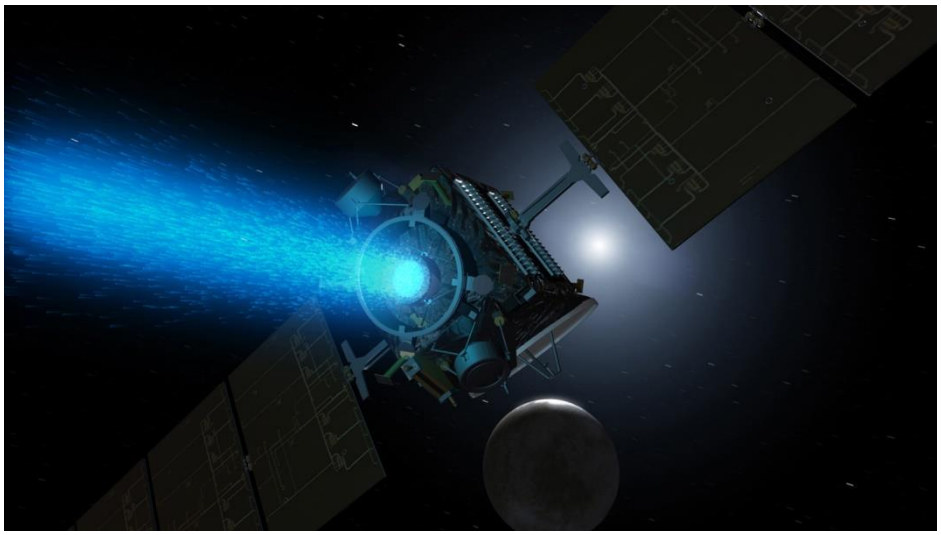


Figure 6. Dawn IPS central thruster application at Ceres approach [10].

As it was said before, the Dawn mission came up thanks to the successful Deep Space 1 Mission, which demonstrated the good operability of the NSTAR electrostatic ion thruster. For this reason, it was decided that the same NSTAR thruster would be integrated in the Dawn IPS. So, in order to cover all the required thrusting directions, three of these thrusters were included in the IPS, which are called **Flight Thruster-1** (FT-1), 2 and 3 [3].

The NSTAR thruster is able to exert a thrust force from 18.8 to 91.0 mN and its input power range is from 0.5 to 2.3 kW. Moreover, although there are three thrusters, only one of them can operate at a time.

This electrostatic ion thruster operates with Xenon as propellant. Inside it, the injected Xenon atoms are bombarded with electrons threw by an electron gun. This ionisation process is greatly enhanced by the magnetic field inside the thruster. From this, positive ions are generated and then, they are

electrostatically accelerated thanks to high-voltage positive and negative grid electrodes. Due to this abrupt acceleration, they are propelled out of the thruster and finally produce the desired thrust force to the spacecraft in the opposite direction. Just when they are out, these positive ions are neutralised thanks to an electron beam located at the exit.

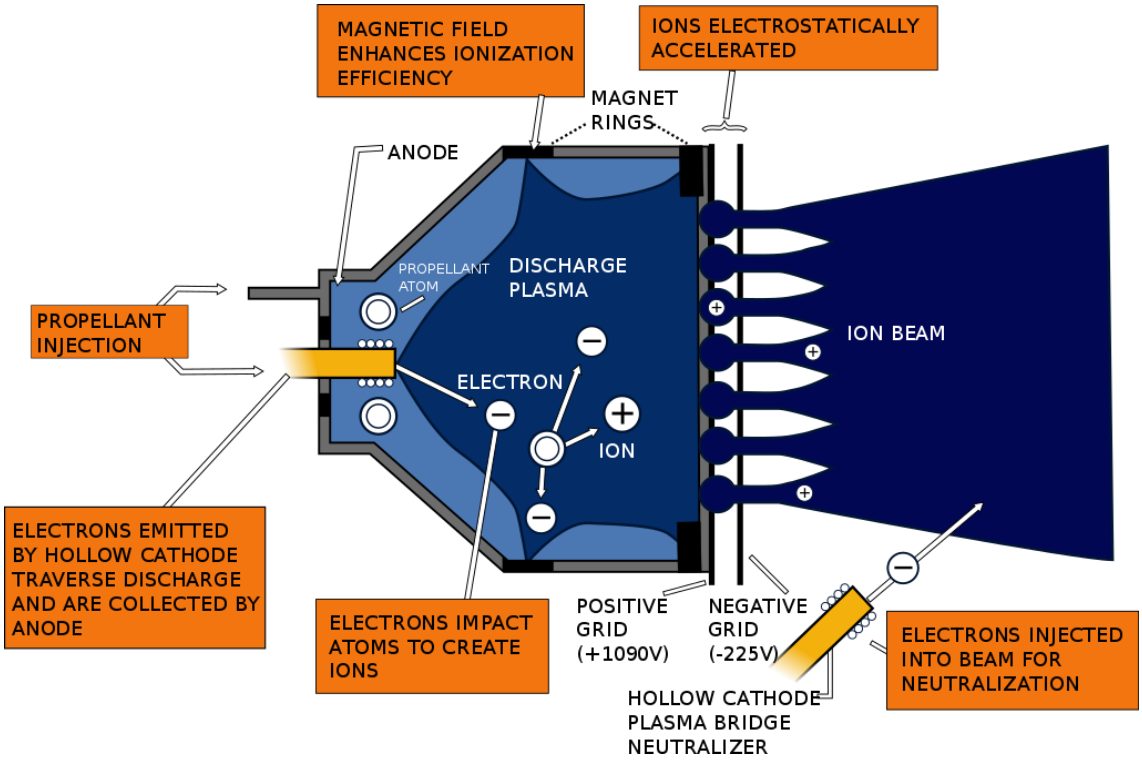


Figure 7. Dawn IPS NSTAR thruster operation process [11].

Apart from these ion thrusters, three **Power Processor Units (PPUs)** were also tested in the DS1 mission and included in the Dawn IPS. Their function is to convert the input power received from the solar arrays into the specific current required to start and operate the ion thrusters. This current can be distributed in two thrusters although only one is able operate at a time.

Another significant element of the Dawn IPS that complements these two main previous sections is the **Xenon Feed System (XFS)**. The most relevant elements of this set of components related to Xenon are its main storage tank, two plenum tanks which were also used in the DS1, and the Xenon Control Assembly (XCA). This XCA is responsible for the flow rate control on the Xenon distribution, related to the pressure regulation employed by the system. The methodology is the same as the DS1 one too.

In addition to the NSTAR thrusters, a set of three **Thruster-Gimbal Assemblies (TGAs)** are just installed next to them. These TGAs can operate in 2 axes (±8º in one axis and ±12º in the other) and are able to adjust the orientation of the ion thrusters depending on the required operation thanks to the two actuators that each TGA incorporates.

Finally, just placed before the PPUs there are two **Digital Control Interface Units (DCIUs)**, which are in charge of monitoring and controlling the previously mentioned systems, specifically the PPUs operation, the XFS valves and the TGAs actuators. It has greatly high efficiency, being even greater when the input power is higher.

Focusing on the PPUs, which affect directly to the ion thrusters, the DCIUs receive the IPS orders from the Dawn command and data subsystem to be capable of starting up the thrusters and making them work in the selected throttle level or even changing this throttle level to another one.

These all subsystems included in the Dawn IPS are represented in the IPS block diagram displayed below.

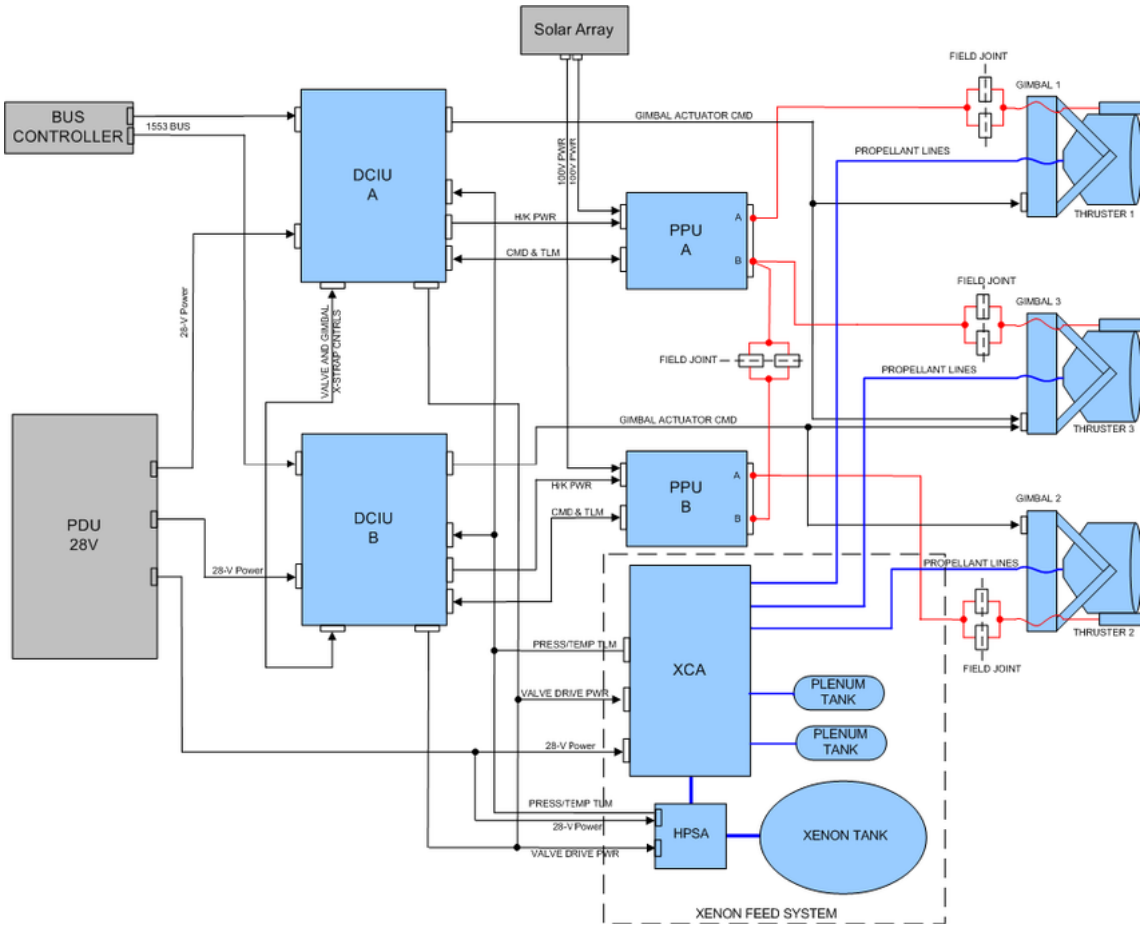


Figure 8. Dawn IPS components block diagram [3].

Now, having described the characteristics of the Dawn mission and spacecraft related to this project, it is time to focus on the employed software, which will be deeply defined and analysed in the following chapter.

CHAPTER 3. SPICE

From the beginning of this thesis, it was settled that the study of the Dawn Ion Propulsion System is performed through the SPICE database library and toolkit. As SPICE will be employed in the MATLAB software, it can also be called as MICE.

SPICE is an information database designed by the NASA's Navigation and Ancillary Information Facility (NAIF), which is established in the Navigation and Mission Design Section at the Jet Propulsion Laboratory (JPL) [12] [13]. It comprises all the NASA's space exploration missions, such as Dawn, as well as other space agencies' missions such as Rosetta from ESA of Hayabusa from JAXA. The provided data of these missions depends on their complexity and their confidentiality, and it includes from the spacecraft's position to information given by the scientific instruments, so its database is under continuous update.

Figure 9. NAIF logo [12].

The main objective of SPICE is to support and facilitate the research and interpretation of the investigation work performed by scientists. It helps to improve the accuracy of the study of a specific spacecraft or any of its systems, such as in this thesis, the spacecraft's scientific payload and how it is used, and the study of any explored body of the Solar System and its near surroundings.

SPICE is definitely well recognised as an accurate database which provides information considered as real one due to the fact that the data is extracted directly from NASA's Planetary Data System, managed by the United States government employees at the Goddard Space Flight Centre [14].

Apart from the SPICE database, the NAIF also designed a specific set of files and functions called toolkit expressly designed for the employed computing software and the computer operating system. Obviously, this set of files is under regular update too.

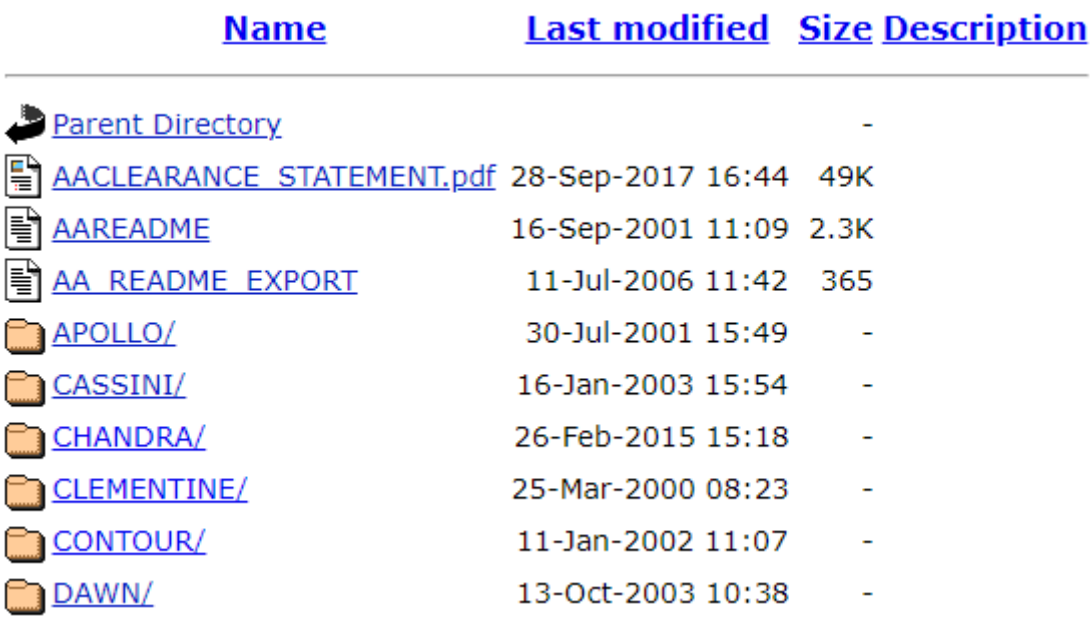
In this chapter, the structure and the different characteristics of the SPICE database and toolkit will be explained together with its application in the MATLAB software.

3.1 Database library

This section's goal is to describe the organisation of the SPICE database together with its various features. Due to the way it is distributed, it can be said that this database structure can be called as SPICE database library.

The SPICE database library is divided in a wide number of levels and sublevels, characterised by its folders and subfolders. Moreover, each level is usually accompanied by its corresponding readme text file, which describes in detail important files or facts belonging to the data provided that folder.

As it was just mentioned, the library comprises information about different NASA and other space agencies missions, so the first level of this NAIF library is a list of the space exploration missions' folders available to be studied. Besides, there are other folders which mainly contains two types of information: about the SPICE toolkit and about the Solar System elements. Furthermore, the readme file of this level contains general information about the SPICE database and about some of its folders.



Name	Last modified	Size	Description
Parent Directory		-	
AACLEARANCE_STATEMENT.pdf	28-Sep-2017 16:44	49K	
AAREADME	16-Sep-2001 11:09	2.3K	
AA_README_EXPORT	11-Jul-2006 11:42	365	
APOLLO/	30-Jul-2001 15:49	-	
CASSINI/	16-Jan-2003 15:54	-	
CHANDRA/	26-Feb-2015 15:18	-	
CLEMENTINE/	25-Mar-2000 08:23	-	
CONTOUR/	11-Jan-2002 11:07	-	
DAWN/	13-Oct-2003 10:38	-	

Figure 10. Library first level's mission folders [15].

In the case of this project, the folders employed to extract the required information are the "*DAWN*" and "*generic_kernels*". As its name says, the "*DAWN*" folder provides all the available data from Dawn spacecraft, and the "*generic_kernels*" provides data from the Solar System bodies and features that appear in the future computations.

Then, navigating inside these folders, it is possible to reach the desired final levels to extract the needed data for the calculations. In these final levels, the data files are found as well as the corresponding readme text file, which contains information about the nomenclature of the data files and their content. The groups of information contained in these data files are called **kernels**. As the kernels data can vary in an extensive way, the files can have a different format depending on their kernels, so the way of reading these kernels files types in the programming software varies too.

For example, one of the first objectives in the computation codes is to determine the position of the Dawn spacecraft. In order to accomplish this, it is possible to use the kernel file "*dawn_rec_090801-090915_090923_v1.bsp*", which is found in the "*spk*" folder of the kernels folder of the principal "*DAWN*" one. Thanks to the readme file [16] it is possible to know information about the file by its nomenclature.

This kernel file information extracted from the nomenclature is distributed along its keywords described below:

a) The word *dawn* obviously refers to the Dawn spacecraft, *rec* refers to the reconstructed trajectory produced by the Dawn navigation system.

b) *090801-090915* refers to the beginning and end date (from the 1st of August 2009 to the 15th of September 2009) of this trajectory in this file.

c) *090923* refers to the delivery date of the file by Dawn (23rd of September 2009).

d) Then, there is the version of the file, which in this case it is the first one (*v1*).

e) Finally, the file name extension "*.bsp*" indicates that the file is a binary SPK one.

It can also be seen that the keywords are separated by a "_" symbol.

The way of how these kernel files are downloaded and applied to the programming software in order to perform the required computation will be explained in the next section.

3.2 Programming with SPICE

This section describes the set of files and functions that the SPICE database software contains as well as how to download and work with the previous kernel files data with these given routines.

Figure 11. SPICE database library logo [17].

3.2.1 SPICE toolkit

First of all, the SPICE toolkit is downloaded from the NAIF page. As it was said before, there is a wide variety of installing options, depending on the programming software and the operative system. In the case of this thesis, the SPICE toolkit used is the one for MATLAB and compatible with Windows. Now, having downloaded and unzipped it, there is a collection of folders inside the folder "*mice*" with many different files.

In order to make a practical description of the toolkit, the most important files for this project are defined below [18]:

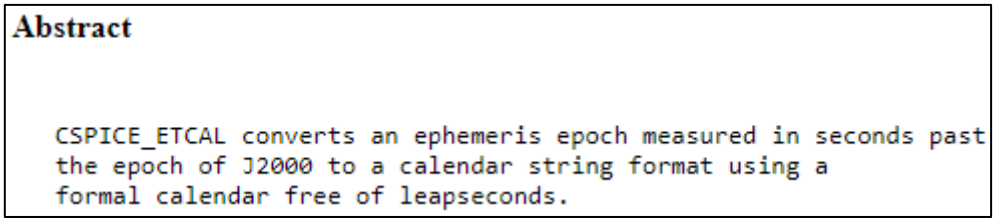
- A broad number of application program interfaces (APIs), which contains the vast set of sub-routines and functions that configure the SPICE toolkit. They make possible the reading and study of the kernels data from the library files. All the provided functions that will be used in the computations procedures are stored in the "*mice/src/cspice*" folder, which will be the first of the two paths required the MATLAB software code.

- A library "*cspice.lib*" constructed by all the given APIs described in the previous point. This library is stored in the "*mice/lib*" folder, which will be the second of the two paths required the MATLAB software code.

- An assortment of HTML text files assigned to describe each function included in the first point. It includes the description of its objective in an abstract, the input elements, the resultant output values and how are these input and output parameters presented. Furthermore, the HTML files expose some example about some of the uses of their corresponding function.

  For example, one basic function employed in all the codes of this project is the *cspice_etcal*, which transforms a determined number of seconds in

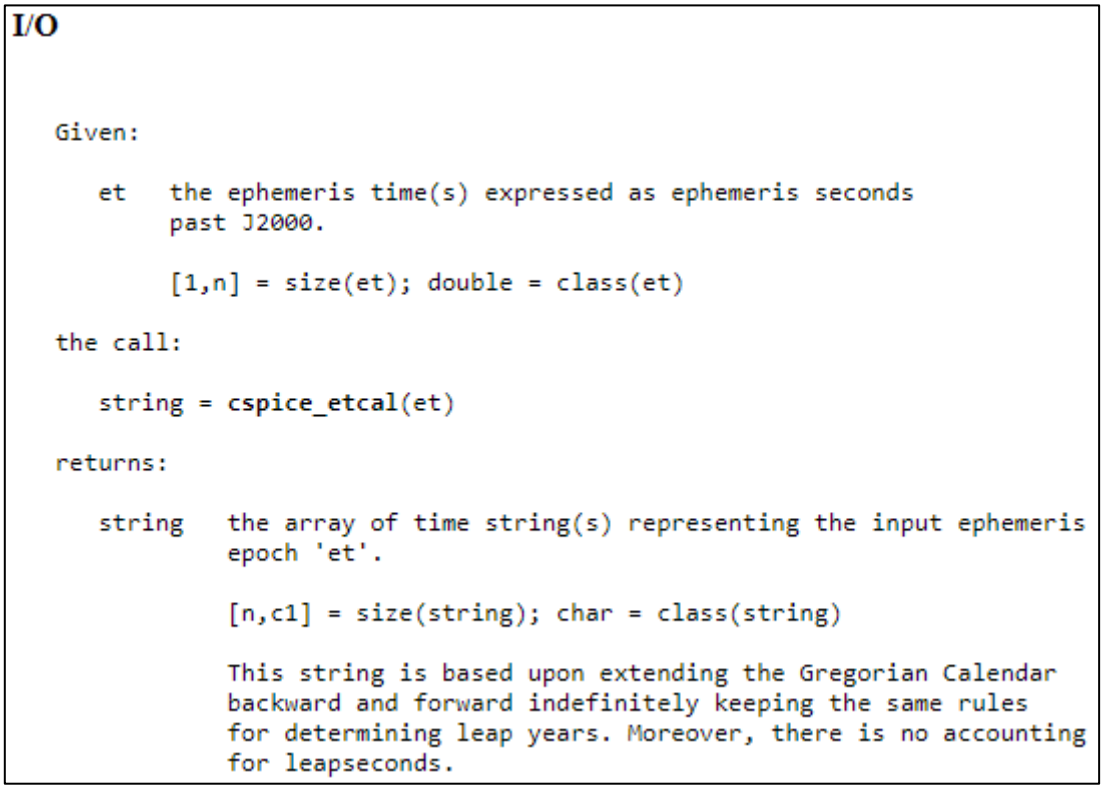
the J2000 epoch to a string of the equivalent calendar date. Although there will be a section in this chapter about the description of the most important functions employed in this thesis, the following figures corresponds to parts of the HTML definition file of this function.



Abstract

CSPICE_ETCAL converts an ephemeris epoch measured in seconds past
the epoch of J2000 to a calendar string format using a
formal calendar free of leapseconds.

Figure 12. *cspice_etcal* SPICE function HTML file abstract [19].



I/O

Given:

et the ephemeris time(s) expressed as ephemeris seconds
past J2000.

[1,n] = size(et); double = class(et)

the call:

string = **cspice_etcal**(et)

returns:

string the array of time string(s) representing the input ephemeris
epoch 'et'.

[n,c1] = size(string); char = class(string)

This string is based upon extending the Gregorian Calendar
backward and forward indefinitely keeping the same rules
for determining leap years. Moreover, there is no accounting
for leapseconds.

Figure 13. cspice_etcal SPICE function HTML file inputs and outputs [19].

3.2.2 “SPICEing” MATLAB

Now, from the collection of files above it is possible to apply the functions supplied by the SPICE toolkit to the MATLAB software in order to create the desired codes to perform the computations of this project. This section exposes the basic structure that the MATLAB shall have to make calculations from the SPICE kernels data.

Figure 14. MATLAB software logo.

Initially the kernels have to be loaded on the computer. It is very important to understand that the once the kernel files are downloaded in the PC, they will be able to be employed whenever the user wants, even when the MATLAB session is closed and opened again. So the kernels data are not saved in the MATLAB session but in a specific folder of the computer.

First of all, the parameter *HOMESPICE* is the first to come into play. It indicates the directory where the *mice* folder, the one which allocates all the SPICE toolkit files, is located within the computer. This directory is usually for Windows "*C:\...\folders\...\mice*".

After this parameter, the second key one is the *METAKR*. It enumerates the list of kernel files employed in the specific code. Its structure follows a matrix of 2 columns for each kernel file. The first column indicates the web link of the determined kernel file in the SPICE database library website. Then, the second column only indicates the identification name of the specific kernel file. Using the same example of the previous section, for the kernel file "*dawn_rec_090801-090915_090923_v1.bsp*", the first column would be "*https://naif.jpl.nasa.gov/pub/naif/DAWN/kernels/spk/dawn_rec_090801-090915_090923_v1.bsp*" and the second one the identification name of the file itself.

Now, having identified the wanted kernel files and the location of the SPICE toolkit folder, it is possible to import these kernels to their corresponding place in the toolkit. For this, a new folder called *kernels* where the kernel files will be imported is created in the *mice* folder. The *HOMESPICE* and *METAKR* parameters are introduced in the *initSPICEv* function as input variables. This *initSPICEv* function is not included in the SPICE toolkit but it is the intermediary between the parameters and the SPICE function that loads the kernels into the computer. This function has no output variables because its mere goal is the kernels load. So now the *initSPICEv* function is executed.

Firstly, this *initSPICEv* function adds the two paths for MATLAB mentioned before, the one to the SPICE toolkit functions and the one to the toolkit library file. Then, a loop analysing each kernel file begins.

Inside this loop, the steps followed are the ones below:

1) The location of the selected kernel file inside the *kernels* folder is created. This new kernel location is stored in the parameter *kfile*.

Inside this parameter, the small function *osi*, which changes the nomenclature of the directory depending on the operative system (Mac or Windows), is also applied. This *osi* function is neither part of the SPICE toolkit.

2) It is checked whether the specific kernel file already exists or not in the *kernels* folder.

3) If this kernel file is not already present in the folder, it is downloaded and placed in the folder. This procedure is performed thanks to the MATLAB function *websave*, which takes the new created kernel file location from *kfile* parameter and the NAIF website link from the first column of the *METAKR* parameter. Moreover, the MATLAB command window indicates that the kernel file is being downloaded.

4) Finally, there is the kernel file in the wanted folder but it cannot be opened by only using MATLAB functions, so here is when the SPICE toolkit functions come into play. The *kfile* parameter is introduced as an input parameter into the *cspice_furnsh*, which is the one that reads the kernel data inside the file and makes it available to be studied.

From here now, the code is can be fully personalised depending on its objective and the kernel data is not downloaded or called again anymore. In the case of this thesis there is one computation step which is always performed, the time determination. However, its detailed explanation will be shown in the next chapter, the computations one.

Finally, here there is a scheme of the steps that has to be followed during this SPICE beginning procedure.

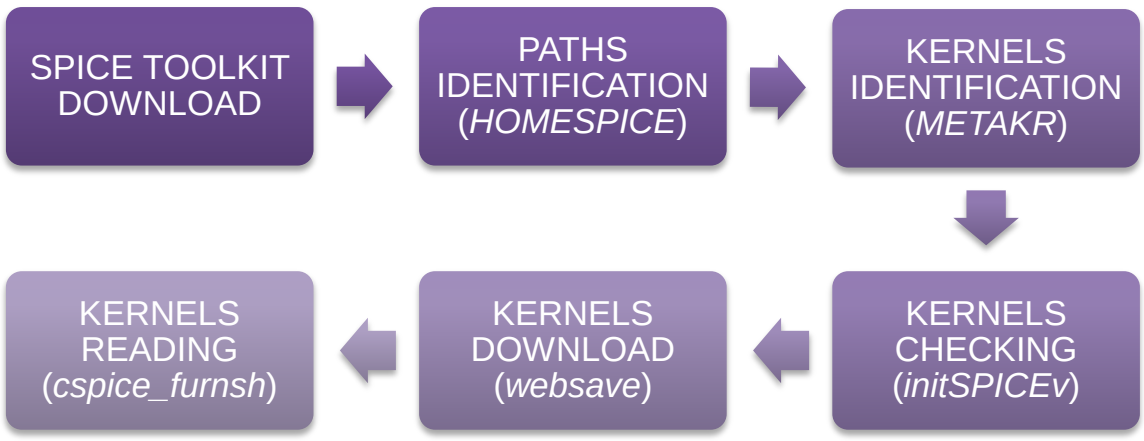


Figure 15. Required steps to make the SPICE toolkit ready to use.

CHAPTER 4. DAWN TRAJECTORY AND IPS COMPUTATIONS

Since last chapter explained in detail the SPICE database library and toolkit and how to settle a MATLAB code ready to use them, the computations of many parameters of the Dawn spacecraft during its mission have been performed employing SPICE.

The created codes have been separated in two main blocks:

- Codes that contains calculations from the kernels provided by the SPICE database library data using the functions supplied by the SPICE toolkit.

  The objective of these codes is to mainly analyse the performance of the Dawn spacecraft and its Ion Propulsion System through real data from the mission.

- Codes that contains calculations from initial data provided by the SPICE database kernels but applying to them theoretical models and methodologies. They compute parameters already given by SPICE database or others from external reliable sources.

  The goal of these codes is to prove the validity of the application of the theoretical expressions in specific Dawn performance and their accuracy in their resultant values.

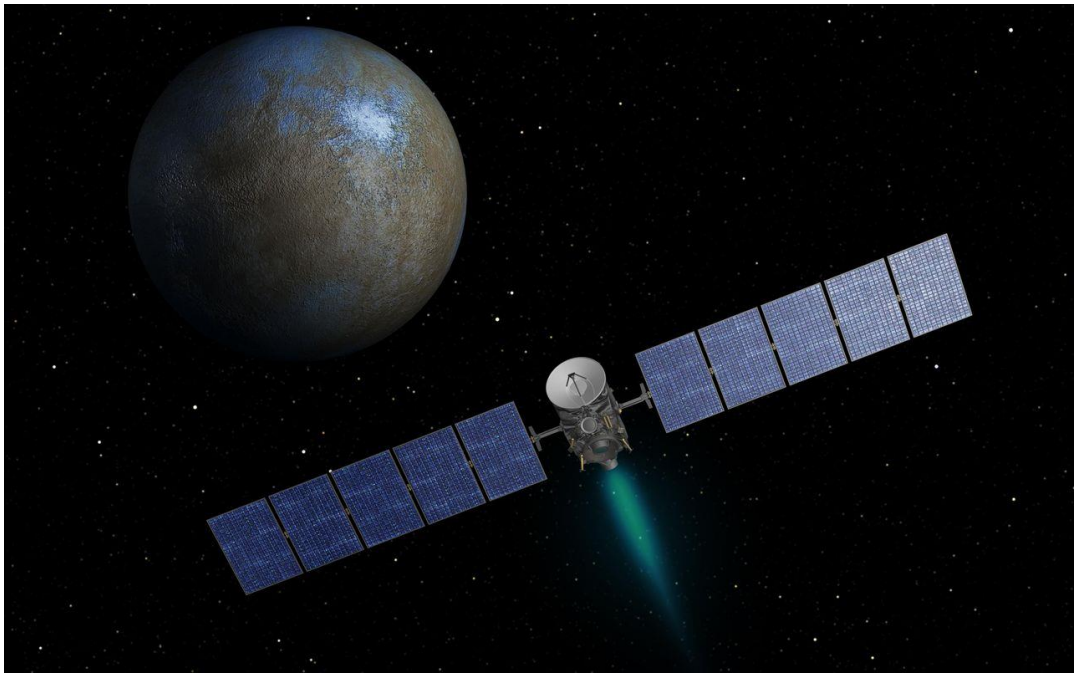


Figure 16. Dawn spacecraft around dwarf planet Ceres [20].

4.1 From SPICE

This section focuses on the first block defined above, the one that details the codes built from the SPICE database library and toolkit. In order to understand better the complexity of the codes, they are exposed from the most basic ones to the ones with more specific computations.

From this, codes included in this section are the following:

- ✓ **Time period.** The code that determines the period of time in which all the calculations in the next codes are performed.

  This simpler but relevant code is included in the entire project's code due to its necessity at the computation of any parameter in the Dawn mission. Then, when it is executed in the further codes, these desired parameters are calculated during this period of time.

- ✓ **Position and velocity.** The code that determines the position and velocity of an object, the Dawn spacecraft mainly in this case, respect to another object which acts as the reference system.

  This code allows the study and analysis of the spacecraft or other selected object from the point of view of a second one and makes possible the representation of the trajectory performed during a period of time determined by the first code of this list.

- ✓ **Vesta scientific orbits.** From the previous codes, if the trajectory of Dawn is represented in the period of time while it orbits the asteroid Vesta, it is possible to observe the scientific orbits that the spacecraft performed to accomplish its mission.

  Then, extracting the scientific orbit trajectories plots, these are compared to the ones described in the mission objectives.

- ✓ **Vesta escape trajectory.** One of the innovations of the Dawn mission was the possibility of orbiting two different celestial bodies, so the key operation to accomplish this is the escape from the first body to start a cruise trajectory to the second one.

  The trajectory and other significant parameters are extracted from this mission stage where the spacecraft exited from Vesta's gravitational influence, which was the first of the two body objectives.

4.1.1 Time computation

The main objective of this code is to create a parameter which includes a period of time determined by the user. Then, this time parameter can be used in all the desired computations during the entire project.

As the wanted solution is a parameter employed in further codes, this period of time computation is not a main MATLAB code but a MATLAB function. As it happens with all MATLAB functions, this one is located inside other main codes and has some specific input and output parameters. The input parameters are defined in the main code before the function is executed and the output ones are extracted from the function execution and can be then employed in the following code lines.

Furthermore, all the MATLAB functions have an own nomenclature employed in main codes to call and execute them. Regarding this time period function, it appears in all codes as *time_computation*.

In order to describe this time period function as it is applied in all the computations codes, this section is divided in three parts:

- The **input** parameters required to execute the function, provided by the main code.

- The **output** parameters, which represent the final solution of the function is used in the main code computations.

- The **development** procedures followed inside the function execution work.

4.1.1.1 Input parameters

First of all, these input parameters are always introduced in all main codes just before the *time_computation* execution, although their values depend on the studied case. These input variables are two: the collection of the dates of the beginning and the end of the desired period of time, and the parameter that determines the time distribution of the output parameters.

On the one hand, there is the parameter about the initial and final dates of the time period. Its content must be introduced by the user depending on the wanted case, so its format and the way of introducing its data have to be clearly understandable for the user. This is the reason why the selected format for this variable is simply typing the first three letters of the month's name, then the day number just followed by a comma symbol and finally the year's number.

All these elements are separated by one space. They all form what is called in MATLAB as a string, a sequence of different characters delimited two apostrophe symbols. For example, if the selected date is the 2nd January 2010, it would be introduced as 'Jan 02, 2010'. As the required introduced dates are two, the initial and the final one of the time period, this parameter encompasses

both dates' strings, so they are inserted between curly brackets and separated by commas, creating a strings vector. This strings vector is called in all codes as ***dates***. For example, if the time period goes from the 2nd January 2011 to the 3rd February 2012, the *dates* parameter would be {'Jan 02, 2011','Feb 03, 2012'}.

On the other hand, there is the parameter that defines the size of the content of the desired output variable, so it is only a number. Although it has its own section, it is important to know that the final parameter is a vector of seconds between the selected initial and final dates. This parameter establishes the size of this output vector, which directly defines the time step in seconds between the elements of the vector. This implies that this variable is closely related to the accuracy of the following computations.

The bigger this parameter value is, the smaller the time step is and the more accurate the computations are due to the fact that the calculations are executed in more occasions within the same period of time. Of course, the consequent drawback of this parameter value increment is that the computations performed afterwards in the code would last more due to the necessity of a higher computing efficiency. This parameter is called as ***STEP***.

4.1.1.2 Output parameters

The output parameters are elements that are used in the subsequent code computations or representations, given by the *time_computation* function execution. As it happened in the input variables, there are two outputs: the just mentioned time period vector, which contains the time values between the initial and final dates, and a dates' strings vector employed in the plots of the calculated values.

Firstly, the main output parameter is the time period vector. It contains the time values of the selected period of time corresponding to the TDB seconds past from the J2000 epoch. The J2000 epoch is established at the 1st January 2000 at 12:00 TT (Terrestrial Time). This means that the time values inserted in this vector correspond to the seconds spent since this precise moment in time. These values are determined from the input initial and final dates in the *dates* parameter and the time step between each time value is determined by the *STEP* variable. This J2000 time values vector is called in the following code lines as *times_sec*.

Then, the other output parameter is called *dates_plot*. This is not a necessary variable but it eases the understanding of the representations of the parameters computed in the code. As the seconds past since the J2000 epoch may be a useful value at computing but not having a time reference for the user, they are converted into time dates in order to introduce them in the code plots. As they cannot be all introduced in the plots, the *times_sec* vector is divided in 10 equal parts, so the 11 time dates that delimit these portions are the ones represented in the plot. Finally, this *dates_plot* vector is a strings vector which contains 11 strings.

4.1.1.3 Development procedures

This section describes the process inside the *time_computation* function when it is executed. Its explanation will follow the line steps performed in the function code.

These *time_computation* function steps are:

1) From the *dates* parameter, the function ***cspice_str2et***, which is included in the SPICE toolkit, is applied to the dates strings. This function is able to convert any string which represents a date to the time value of the seconds spent since the J2000 epoch. So, in this step, the dates' string vector is converted into a number vector called *et* of both initial and final dates transformed into these J2000 epoch seconds.

2) The code line just after the first step is the creation of the ***times_sec*** parameter. From the *STEP* variable and the J2000 epoch seconds of the initial and final date in the *et* vector, the time values between these initial and final ones are generated by distributing them in a number of portions corresponding to the *STEP* value. So, the *times_sec* length corresponds to the *STEP*.

3) This *times_sec* vector contains the time values in J2000 epoch seconds, so just after its creation, it is converted into the same vector but with the time values in J2000 epoch dates, creating the new *times_dates* vector. This is possible thanks to the application of the SPICE toolkit function ***cspice_etcal*** to the *times_sec* vector.

   This function transforms a J2000 epoch seconds time value into a calendar date string with a different format respect to the input parameter *dates*. This format follows the structure of beginning with the year, then the first three letters of the month's name, the day number and finally the hour, minutes and seconds. In order to make it simpler, the code only takes the first 11 characters, which correspond to the year, month and day.

4) Now, having the *times_dates* vector with all the time values of the time period expressed in calendar dates, the *dates_plot* vector is created. As it was explained in the output parameters section, all the time period is divided in 10 equal parts, so 11 elements equally separated of the *times_dates* are taken to generate the *dates_plot* parameter.

Now, after all these steps, the required output parameters have been computed, so they are ready to get into action when they are needed in the other codes. The scheme below sums up all the procedures performed during this *time_computation* function.

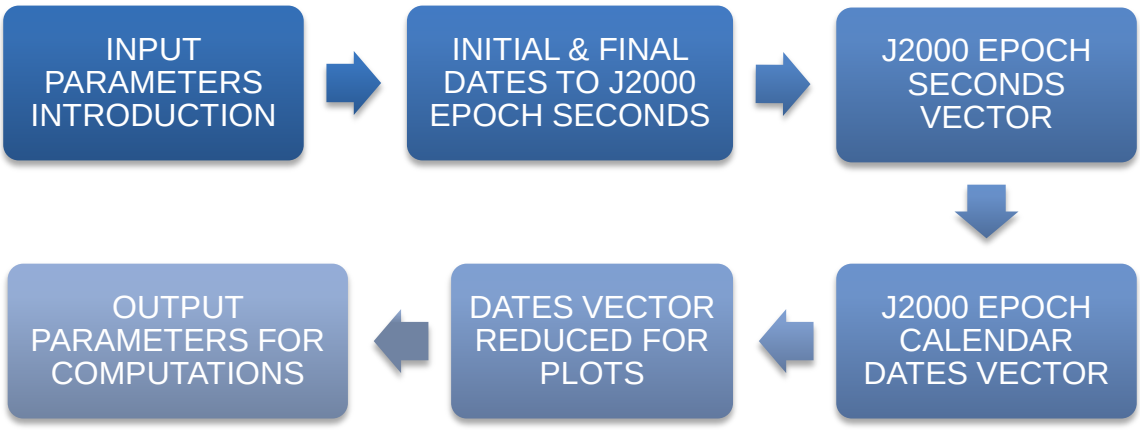


Figure 17. Required steps at the execution of the time_computation function.

From now on, the SPICE introduction explained in the previous chapter and this *time_computation* function are always introduced in all the following computing codes.

4.1.2 Position and velocity

Now, having the basic initial structure to create a code using the SPICE library and toolkit, the first step is to compute the parameter that corresponds to the highest level of this thesis, the Dawn trajectory, that is, its position during a specific period of time. Furthermore, as they are closely related, the velocity during this trajectory will also be determined.

This section is divided in two parts:

- **Code explanation**. The code to perform the trajectory (position) and velocity computation description step by step.

- **Calculations examples**. From the described code, two specific cases of the Dawn mission will be tested and analysed.

4.1.2.1 Code explanation

First of all, the MATLAB program created to be in charge of this trajectory and velocity computation is exposed and studied. It is important to know that this code is able to determine these two parameters since just after Dawn launch and deployment from its rocket from Earth until the last data received by the spacecraft.

Figure 18. Dawn just after its deployment and ready to start its journey [21].

Besides, it is taken into account that the output parameters of the *time_computation* function are fully available in the established format, as well as the required kernel files, which will be specifically mentioned and justified in all the codes examples concerning these trajectory computations and the ones of the other sections.

Now, regarding on the code calculations objectives, it is noticeable that the resultant parameters do not represent a complex challenge by their own, as their computations is mainly based in a function included in the SPICE toolkit collection, the one called ***cspice_spkezr***. It signifies a key SPICE function in this project due to its use in almost all the computing codes.

The execution of the *cspice_spkezr* function is performed just after the *time_computation* line, so there are no steps required before it can be operative. Moreover, it is important to remark that the resultant positions and velocities are always with respect to a body different from the spacecraft which acts as the reference frame. This is why the computed velocity is always defined with the specification of the reference body.

The introduction of the input parameters of this function, which are five, follows the structure exposed below:

a) **Target body**. Here, the name of the body whose trajectory is computed is introduced. The resultant position and velocity determined in this function correspond to the ones of this body.

   The name of this target body is typed by the user in capital letters as a string of characters delimited by two apostrophes, and it usually corresponds to the common name of the body. Nevertheless, the SPICE

library provides a list of the introducing names of the available bodies. For example, if the selected reference body is the Earth, the introduced string would be ‘EARTH’.

b) **Ephemeris time**. Here is where the *times_sec* vector extracted from the *time_computation* function is introduced. It demands that the time values must be expressed in seconds past since J2000 epoch, so the format of the vector is right the one required, so it is not modified before its introduction.

c) **Time reference frame**. This input variable establishes the reference frame concerning time used in the computation of the position and velocity.

   It is noticeable that it is directly related to the just previous parameter, the ephemeris time, which has to concur with this reference frame. It is introduced as a characters string just as the target body. Besides, the SPICE library includes a list of the possible time reference frames available at computing.

d) **Aberration corrections**. Here, some corrections such as accounting for the one-way light time and stellar aberration can be considered if necessary.

e) **Observing body**. Here, the name of the body which defines the reference frame is introduced. The position and velocity parameters of the studied body are always computed with respect to this one.

   The name of this observing body is also typed in the same format as the target one, and can be provided by the SPICE library too.

Then, having introduced these input variables into the *cspice_spkezr* function, it is executed and provides two different output parameters, which are the following ones:

- **State vector**. This is the output variable required to accomplish this code’s objectives. This vector is called in the code as *state*.

  It comprises a matrix of six rows, where the first three ones contain the x-, y- and z- coordinates respectively of the target body position with respect to the observer body in kilometres (km), and the other three ones contain the x-, y- and z- components respectively of the velocity vector of the target body relative to the observer’s one in kilometres per second (km/s).

It can be seen that this unique state vector encompasses all the parameters established at the beginning of this section which wanted to be computed, so the target's trajectory can be extracted from the position vector included in the first three rows and the velocity vector is extracted from the other three ones.

Then, these six rows are repeated for each time step, whose quantity is determined by the size of the *times_sec* vector or, consequently, by the value of the *STEP* variable. It is clear that most of the time spent by the program to execute the code is mainly caused by this state vector computation, which has to be looped *STEP* times, which proves that this variable has a huge influence regarding execution time.

- **Light time**. This output variable provides the value of the one-way light time in seconds between the observer and target bodies at each time step.

  This one-way light time is not part of this section's goals, so it acts as an additional data in this project. It would be certainly useful in the case of an analysis regarding communications between the spacecraft and the Earth's ground station or even concerning payload instrumentation such as radars or cameras.

Now, focusing on the output state vector, the necessary computed parameter, it is possible to generate a representation of the trajectory and relative velocity of the target's body. This representation has the purpose of making an easier, quicker and more accurate study of the resultant variables during the selected period of time.

So, first of all, the position and velocity vectors are built by extracting their coordinates and components respectively from the state vector. Regarding the position vector, the vectors *x*, *y* and *z*, which correspond to the coordinates, are created. Then, as complementary information, a vector called *range* is formed. This ***range*** vector contains the distance between the target and observer bodies by calculating the module of the position vector employing the three coordinates.

Having these variables, two plots are generated: one for the trajectory and another for the range. The trajectory plot is created by using the three position coordinates, so in the end, there is a 3-D plot. This 3-D plot is easily represented thanks to the MATLAB function ***plot3***, which works by only introducing the three coordinate vectors, just as the more common and simpler plotting MATLAB function *plot* used for 2-D graphs. Then, this function allows

this desired 3-D graph by joining all the positions of the target body during the selected period of time and generating the trajectory track.

The main drawback between this 3-D plot and the 2-D one is that the time dates are not represented so there is no time reference in the trajectory plot. In order to diminish this disadvantage, the *range* vector is plotted using the *plot* function by only introducing it as the input parameter. The usual procedure would be also introducing a time vector but, as the *times_sec* vector does not provide a clear reference in terms of dates, it is not introduced.

However, the previously defined *times_dates* vector really provides this useful time data, so it is integrated in the graph by adding a code line just after the range plot function which contains the MATLAB function ***set***. This *set* function adds properties to the previous plot, so, in this case, three parameters are introduced in it. The first one is *gca*, which states that the additional property concerns the plot axes, the second one is '*xticklabel*', establishing that the label of the x axis is modified, and finally the *dates_plot* vector, confirming that this vector is allocated in the x axis label. Thanks to this, the *range* vector data can be plotted together with its time period dates, allowing a more understandable comprehension of the trajectory.

Then focusing on the velocity, the procedure is the same as in the position vector. Firstly, three vectors *vx*, *vy* and *vz* are created containing the components of the velocity vector. Then, the total relative velocity is calculated by the module of these three components and is inserted in the ***v_tot*** variable.

Then, as it happened in the range representation, the *v_tot* parameter is introduced in the *plot* function to be plotted. Again, the *set* function executed just after this plot function to integrate the *dates_plot* vector in the label of the x axis of the graph.

Now, after these procedures, three plots and represented: the total trajectory of the target body from the observer one, which is located just in the centre of the 3-D plot, during the selected period of time, the distance between these two bodies along the time period with the reference dates indicated, and finally the relative velocity of the target body with respect to the observer one along the time period with the reference dates indicated too.

Finally, in order to close this and all the rest of the computing codes, the SPICE toolkit function ***cspice_kclear*** is executed. It makes the function of unloading all the kernels data that was previously loaded in the *initSPICEv* function at the beginning of the code. It has no input nor output parameters, so it is just called alone.

The following scheme summarises these exposed procedures performed during this position and velocity computing code.

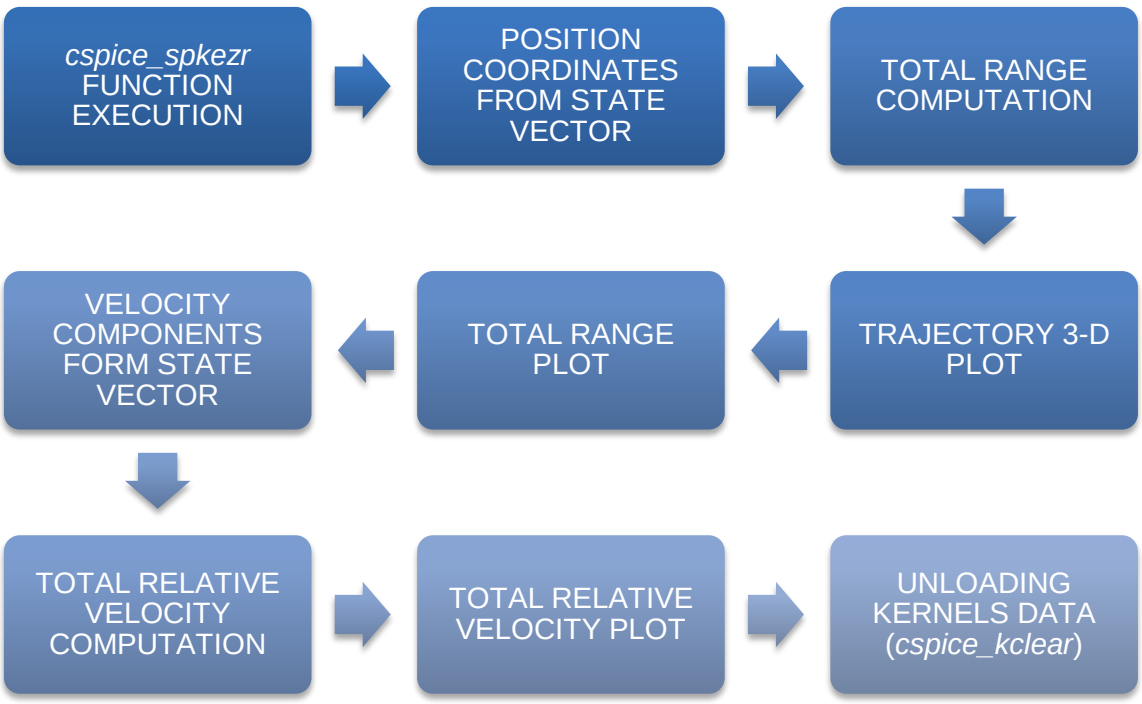


Figure 19. Required steps of the position and velocity computing code.

4.1.2.2 Examples

The explained procedure to determine and represent the position, trajectory and velocity of the target body with respect to the observer one within a selected period of time establishes the general basic structure of this computing code. Now, this section focuses on two specific cases in Dawn mission which demonstrates the right operability of the code and could bring some useful conclusion after their analysis for future computing codes.

The two selected test cases are:

- The trajectory of Dawn from its launch and deploy in the Earth to its arrival to Vesta, the first asteroid if the mission.

- The trajectory of Dawn around Vesta since its arrival until the beginning of its departure.

The first test case focuses on the journey of the Dawn spacecraft since it is launched and deployed in the Earth's surroundings until it arrives to Vesta, where it starts to perform scientific orbits.

First of all, the first specific parameter that has to be introduced by the user is the list of kernel files. In this case three types of kernels are introduced:

- Kernel file in charge of the conversion from UTC to Ephemeris Time, which is called as leapseconds kernel. This kernel file is mandatory for the execution of *time_computation* in order to achieve the right seconds spent from J2000 epoch. The latest version of this kernel file employed in this code and also in all the rest of computing codes is the ***naif0012.tls***, which is a unix-style text file.

- Kernel file which provides data of the orientation, size and shape of the most relevant bodies of the Solar System including the Sun, planets, Pluto, satellites, asteroids (such as Vesta and Ceres) and some comets provided by the IAU Working Group on Cartographic Coordinates and Rotational Elements. The latest version of it is ***pck00010.tpc***. As the time conversion kernel file, this is included in all the computing codes due to the need of orientation data from any Solar System body in all calculations.

- Kernel file which provides data of the trajectory and velocity of the Sun, the planets, Pluto and the Moon. This kernel file makes available the information of any of these bodies at the computation of their position and velocity when comparing them to the Dawn's ones, so it means a key kernel data. The latest version of it is ***de432s.bsp.***

- Kernel files that provide information about the trajectory of the Dawn spacecraft during a specific period of time. The trajectory provided by these files is the reconstructed one, which has the greatest accuracy of the multiple options that the SPICE library gives. Their nomenclature is deeply explained in section 3.1 in the previous chapter about SPICE.

  From it and knowing that Dawn is launched the 27th September 2007 arrives at Vesta the on July 2011, the selected files encompass all the period time between these two dates approximately. The first kernel file is the *dawn_rec_070927-070930_081218_v1.bsp* and the last one is the *dawn_rec_110328-110419_110420_v1.bsp*.

Then, after defining the kernel files of the code, the next variables introduced by the user have to be the input parameters of the *time_computation* function. Firstly, the initial and final dates of this specific time period have to be introduced in the *dates* variables. Knowing the Earth departure and Vesta arrival dates, this time period begins the 28th September 2007, the first day when Dawn data is available, and finishes the 16th April, before the spacecraft is influenced by Vesta's gravity. From this, these two dates are typed in the right mentioned date format.

Then, the other introduced variable is the *STEP*. After some tests, it is decided that the value 10^5 means an accurate value because it implies an acceptable computation time for usual practical tests and the time step is about 1120 seconds, which means about 20 min. From this, the *time_computation* function can be executed and the *times_sec* vector can be extracted from it.

Going into the state vector computation, the input parameters of the *cspice_spkezr* function shall be defined for this case. The target body clearly corresponds to Vesta spacecraft, so it is introduced as '*VESTA*'. The ephemeris time parameter is always the *times_sec* function, as its content depends on the *time_computation*'s input parameter *dates*. The time reference frame is directly determined by the *times_sec* vector, so knowing that its time reference is the J2000 epoch, the string introduced is '*J2000*'. Then, no aberration corrections are necessary so in its parameter the string '*NONE*' is typed. Finally, the observer body is the one around which the target body orbits, so in this case the string introduced is '*SUN*'.

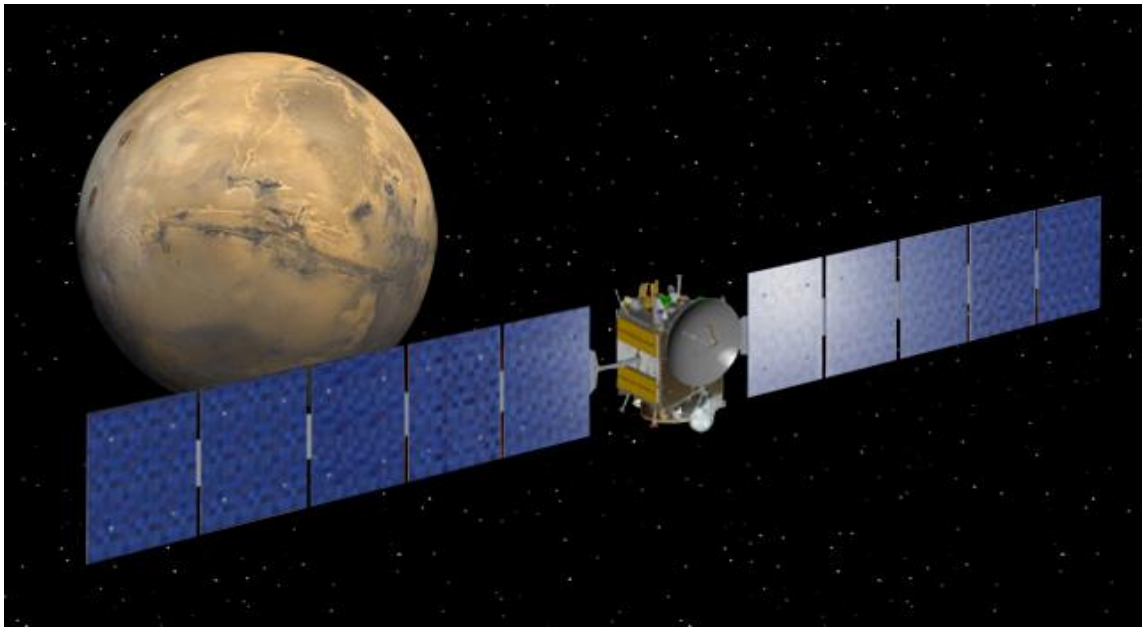


Figure 20. Gravity assist in Mars by Dawn [22].

Finally, having executed the *cspice_spkezr* function and obtained the state vector, this Dawn trajectory, range and velocity can be represented. As it is known, this trajectory shall not be a continuous low thrust climb, just because the spacecraft is impulse in the middle of its journey (17th February 2009) by Mars's gravitational influence [5]. This gravity assist process should be included in this trajectory and velocity plot.

Furthermore, a red big star marker is added in the 3-D trajectory plot just in the coordinate's origin of it. The objective of this star marker is to represent the Sun and makes the plot more understandable at having a reference of the distance between it and Dawn. Moreover, the axes labels are adapted to this trajectory case.

Then, having considered all this the trajectory and range plots for this case are the ones below.

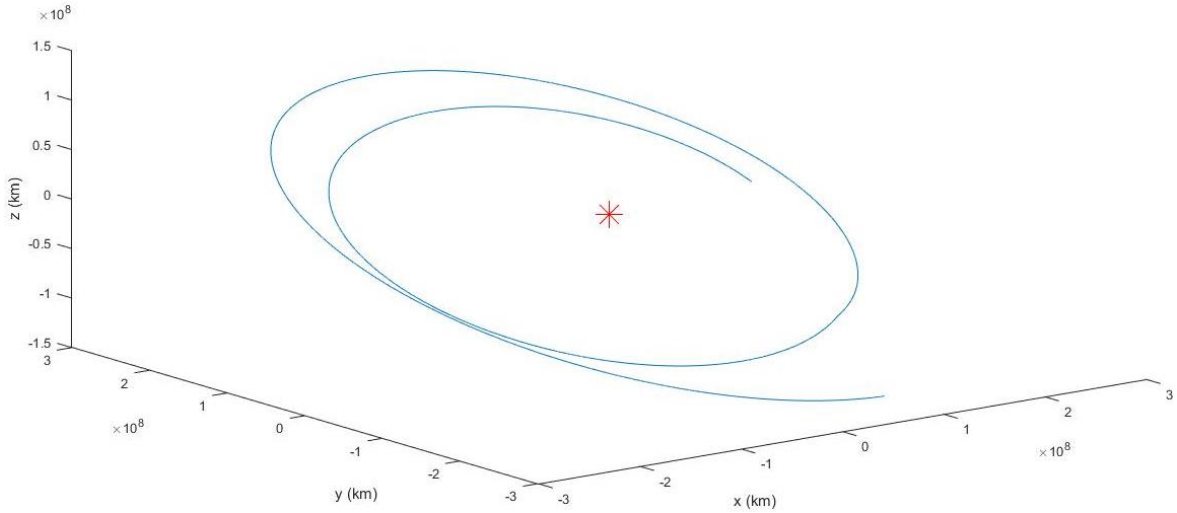


Figure 21. Dawn trajectory around the Sun from the Earth to Vesta.

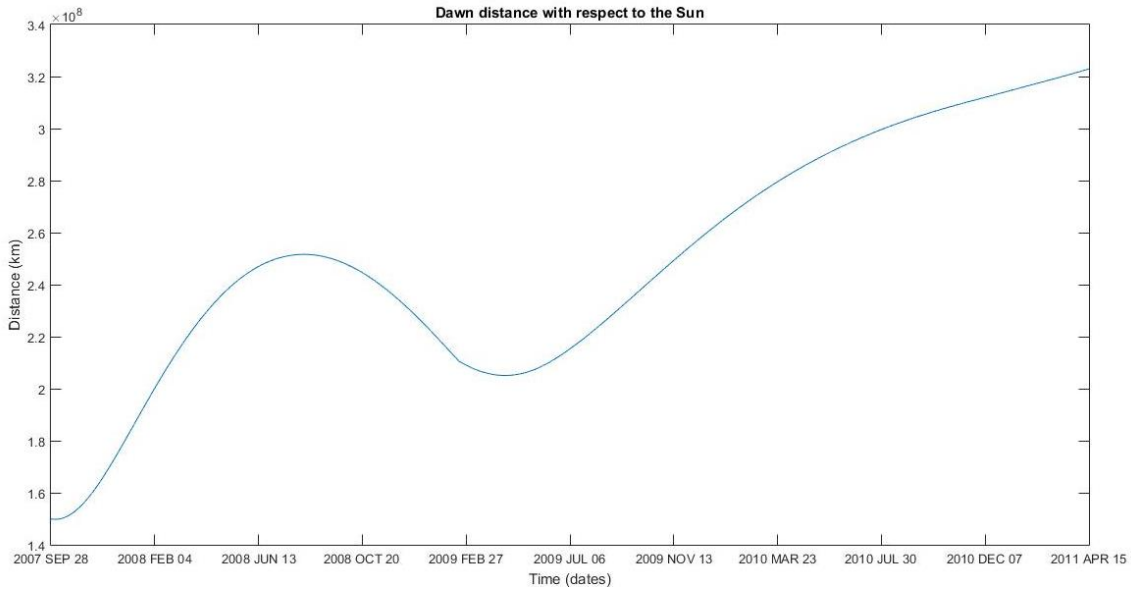


Figure 22. Dawn range from the Sun during the trajectory from the Earth to Vesta.

These two figures represent the Dawn journey to Vesta. As it was said before, it does not comprise a continuous climb trajectory but it is noticeable that the spacecraft begins with a positive climb low impulse until it begins to approach again the Sun by August 2008. Then, it is proved that the trend of the trajectory is clearly changed in February 2009 thanks to the Mars gravity assist. Finally, this assist impulses Dawn until reaching Vesta.

The trajectories of the Earth, Mars and Vesta could be also represented although after some tests it was demonstrated that this trajectory would generate a messy and not certainly understandable plot, so they can be checked in figure 3.

Then, the relative velocity of Dawn with respect to the Sun during this trajectory is also plotted below.

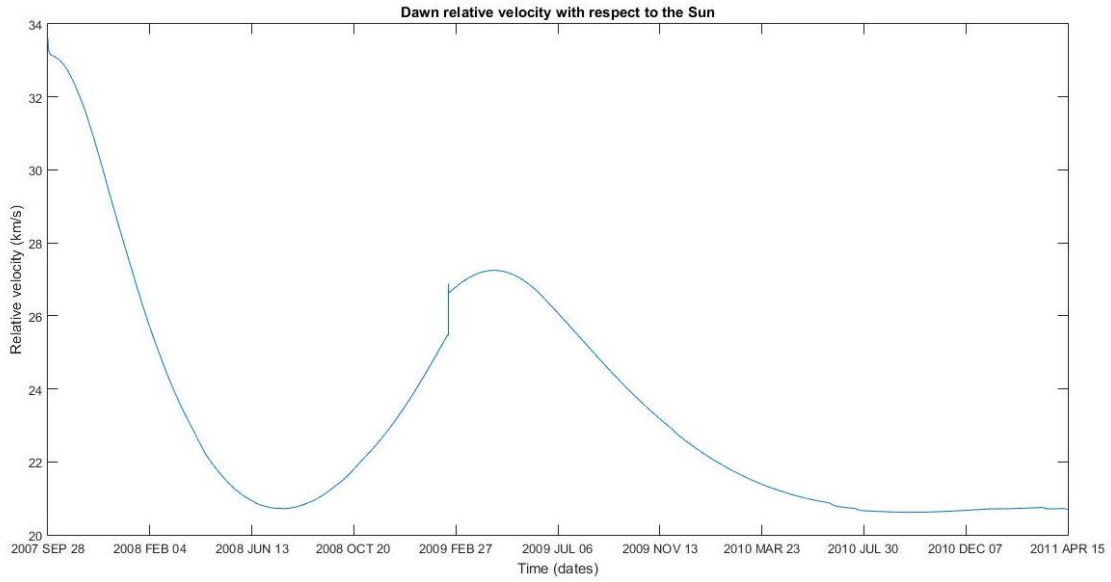


Figure 23. Dawn relative velocity with respect to the Sun during the trajectory from the Earth to Vesta.

Apart from its values, the trends in this velocity graph provide complementary information to the ones regarding the trajectory and range although they are inversely proportional. Firstly, there is the first launch impulse, so the relative velocity decreases until it starts to increase when Dawn approaches again. An important data from this trajectory stage represented in this plot is the abrupt velocity descent just in the first days, which mean the first impulse provided by the launching rocket.

Then, the gravity assist in Mars is clearly more noticeable in this figure, where an abrupt velocity ascent is performed by Dawn, enabling it to impulse until reaching Vesta with an approximate constant velocity. This continuous velocity demonstrates the softness of the spacecraft approach to the asteroid.

These examples certainly prove the reliability of the SPICE library data toolkit functions performance, which are able to represent trajectory and velocity plots with great accuracy and veracity, during this specific Dawn mission phase.

The other test example is the whole trajectory of Dawn around Vesta. It means a similar procedure but there are some small differences which are useful for the further computing codes, which focus more on Vesta. Its analysis will follow the same steps as in the initial Dawn journey to Vesta.

From the beginning, the first parameters modified by the user are the kernel files. There are two main differences in the *METAKR* matrix variable: the removal of the *de432s.bsp* kernel file due to the no need of position and velocity

data from the Sun, and the obvious change in the Dawn reconstructed trajectory kernels, replaced by the ones which encompass the Vesta orbits period.

As the Vesta observation phase of the Dawn mission begins in July 2011 (officially on the 16th), where the spacecraft arrives to its gravitational influence area, until it departs to Ceres in the beginning of September 2012 (officially on the 5th) [5]. As the official arrival and departure dates may not represent a practical plot and analysis of this trajectory around Vesta, after some tests it was established that the initial and final dates studied would be from the 19th July 2011 to the 2nd September 2012. From this, the kernel files go from *dawn_rec_110416-110802_110913_v1.bsp*, the initial one, to the final *dawn_rec_120724-120913_121213_v1.bsp*.

This change in the Dawn trajectory kernel files is also reflexed in the *dates* vector, where these initial and final dates are introduced. Moreover, regarding the *time_computation* function, the *STEP* variable value due to the accuracy that it provides, where the time step is now about 355 seconds (6 minutes) and the computing time cost is as acceptable as in the previous test example.

Finally, the last modifications are present in the state vector calculation function, where the *times_sec* vector contains different values due to the change of dates and now the observer object is Vesta and not the Sun, so the string '*VESTA*' is typed. Besides, the graphs' labels are adapted to the specific case and the red star marker that represented the Sun is removed.

Considering all these modifications and adaptations of this second test example, the trajectory and range plots are shown below.

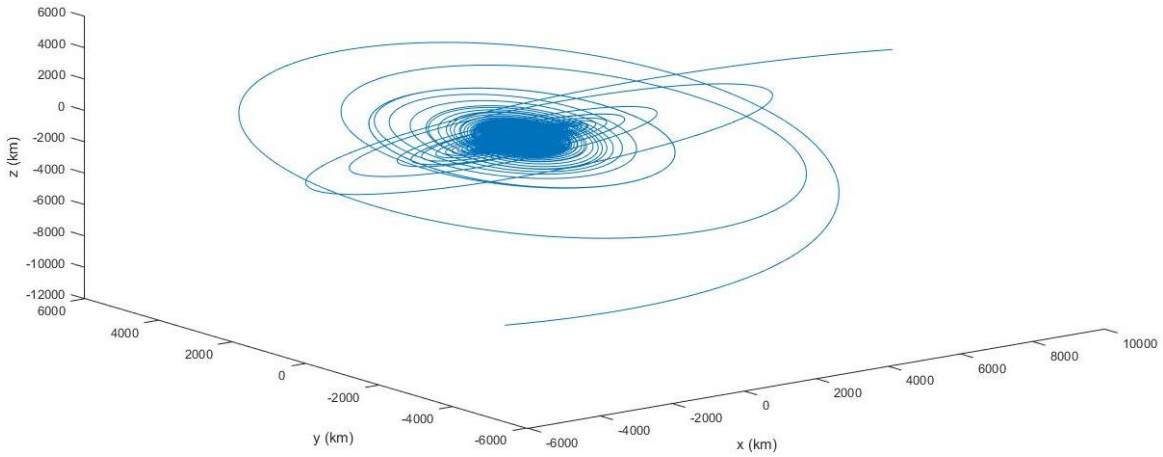


Figure 24. Dawn total trajectory around Vesta.

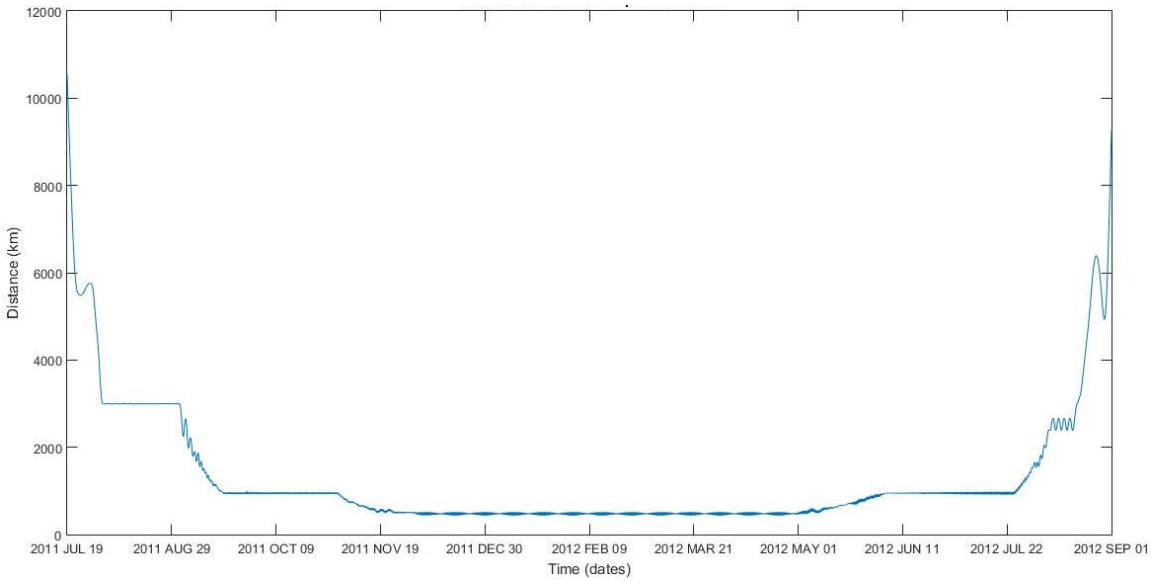


Figure 25. Dawn range from Vesta during its total orbit trajectory.

On the one hand, the figure 24 shows all the total trajectory that Dawn performed around Vesta during its whole stay and clearly proves the complexity of this entire trajectory, making it very difficult to study and analyse if it is totally plotted due to the huge amount of transfer orbits required by the performance of an IPS.

On the other hand, the figure 25 that displays the distance between the spacecraft and the asteroid eases a bit more this analysis, making possible the identification of the four scientific orbits performed by Dawn along Vesta's observation.

These orbits, previously mentioned in section 2.1, are the first survey orbit (11th - 31st August 2011), the first High Altitude Mapping Orbit HAMO-1 (29th September – 2nd November 2011), the Low Altitude Mapping Orbit LAMO (12th December 2011 – 1st May 2012) and the second High Altitude Mapping Orbit HAMO-2 (15th June – 25th July 2012) [9]. These dates coincide with the ones in this figure, demonstrating its correct operation.

Nevertheless, the transfer orbits are also present in both plots, so its comprehension can be truly improved. This is why the next section of this chapter focuses on the study of these scientific Vesta orbits and their upgraded representation and study.

Furthermore, as it happened in the Dawn's journey to Vesta trajectory example, the total velocity is also computed and plotted in order to provide additional results to the study.

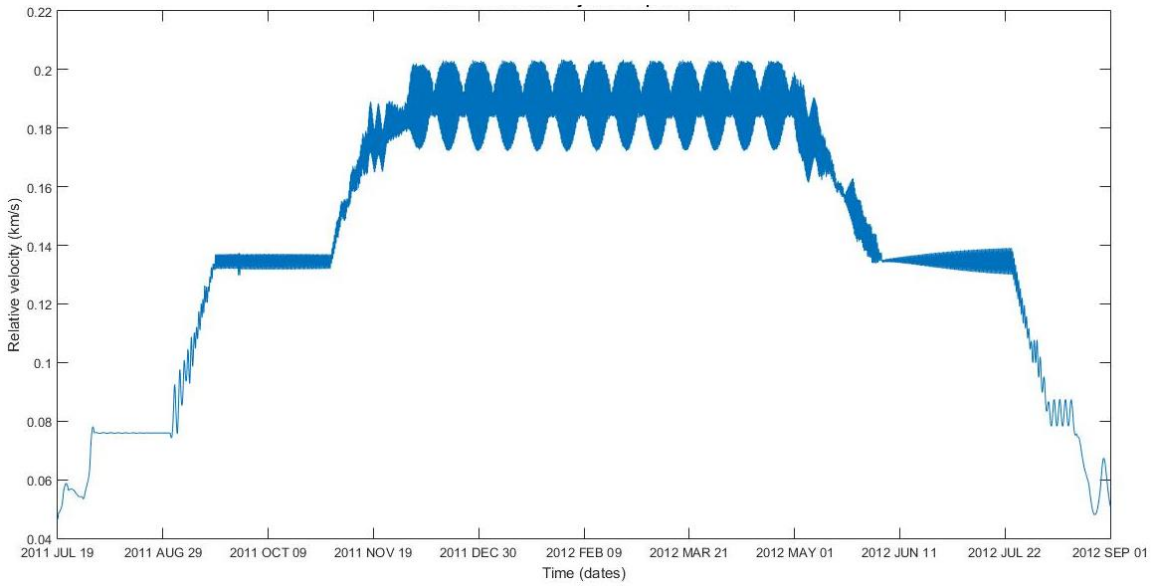


Figure 26. Dawn relative velocity with respect to the Vesta during its total trajectory.

As it happened in figure 25, from figure 26 it is possible to identity these four scientific orbits around Vesta with even more clarity. However, the transfer orbits again difficult the extraction of the desired conclusions, so this total velocity plot is also represented in the next section.

4.1.3 Vesta scientific orbits

After computing and plotting the Dawn trajectory, range and velocity during from its arrival to Vesta until its departure, it was concluded that an improved computing code had to be developed in order to ensure a practical and understandable study of this Dawn mission phase.

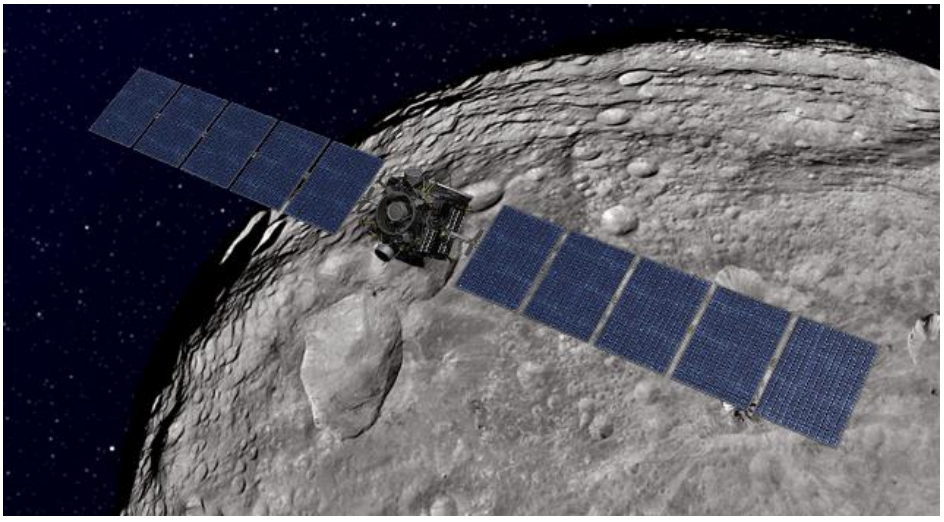


Figure 27. Dawn performing a scientific orbit around Vesta [23].

The procedure of this code is similar to the test example in term of content but the modifications focuses on its structure, as its main objective to provide the same resultant data concerning the most relevant parts of it. This means that the code section about the kernel files remains the same.

Just after this kernel files section is when the code begins to be changed with respect to the one before. Here, in a commented format the four scientific orbits around Vesta are identified and each one are assigned to a determined different numbers. These numbers are assigned considering the chronological order, so the survey orbit corresponds to number 1, HAMO-1 corresponds to 2, LAMO to 3 and HAMO-2 to number 4. The reason why the scientific orbits are assigned to number is revealed just the code line after this procedure.

Having settled the assigned numbers for each scientific orbit for the user information, the ***trajectory*** parameter appears. This variable has to contain one of these numbers, depending on the desired orbits which wants to be computed and studied. For example, if the trajectory, range and velocity of the HAMO-1 orbit are required, this variable value would be 2.

Then, after also defining the *STEP* value, which is the same as the previous codes, these *trajectory* and *STEP* parameters are the input of a new function called ***trajectory_Vesta***. Inside this function there is firstly again a visual explanation for the user of the four scientific orbits and their assigned number.

After these orbits and assigned number display, a *switch* block starts. A *switch* block in MATLAB allows the code execution to follow a determined path depending on the chosen value if the main parameter, where in this case it is the *trajectory* variable. From this, depending on the selected value of this variable, a different path option called as *case* will be executed. For these scientific orbits, their characteristic initial and final dates, which were mentioned in the section before, are what differentiate between themselves, so these path options are the parameter *dates* of each Vesta orbit. The relative length of this procedure is the main reason why a separate function was decided to be created.

So, having extracted the *dates* variable from the *switch* block depending on the chosen scientific orbit, the next process is the same for every possible orbit. The *time_computation* function is executed with the determined *dates* parameter, *times_sec* and *dates_plot* vectors are extracted from it, and then the known *cspice_spkezr* function is executed with the same inputs as in the test example. Finally, from the state vector, the position coordinates, range, velocity components and total velocity vectors are created just as in the test example. These vectors generated from the state vector plus the *dates_plot* variable are the outputs of this *trajectory_Vesta* function.

Returning to the main computing code, these outputs variables are employed in the same way as in the test example although now the data provided by them is specifically about the chosen scientific orbit. From this, the already explained plots of the Dawn trajectory, range to Vesta and total velocity relative to Vesta are executed and represented.

The only difference from the graphs performed in the test example is the addition of a star marker in the coordinate's origin to have the reference position of Vesta. This time, this star marker would be black instead of red due to the fact that the observer object is now an asteroid (Vesta) rather than a star (Sun).

As this new improved computing code can plot the trajectory, range and velocity of Dawn in all four scientific orbits and in order to not being tedious and repetitive, it is decided that LAMO orbit is the one whose plots will be exposed in this memory. This LAMO orbit was the one that lasted more and the one that was closer to Vesta's surface, so its scientific resultant value is clearly more relevant for the mission. Considering this, the LAMO trajectory and range graphs are represented below.

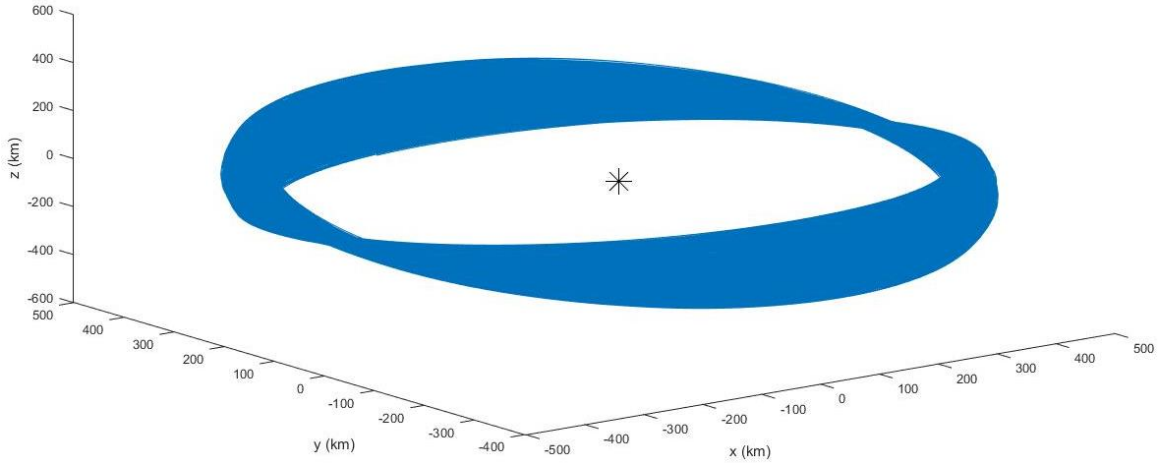


Figure 28. Dawn trajectory around Vesta during LAMO orbit.

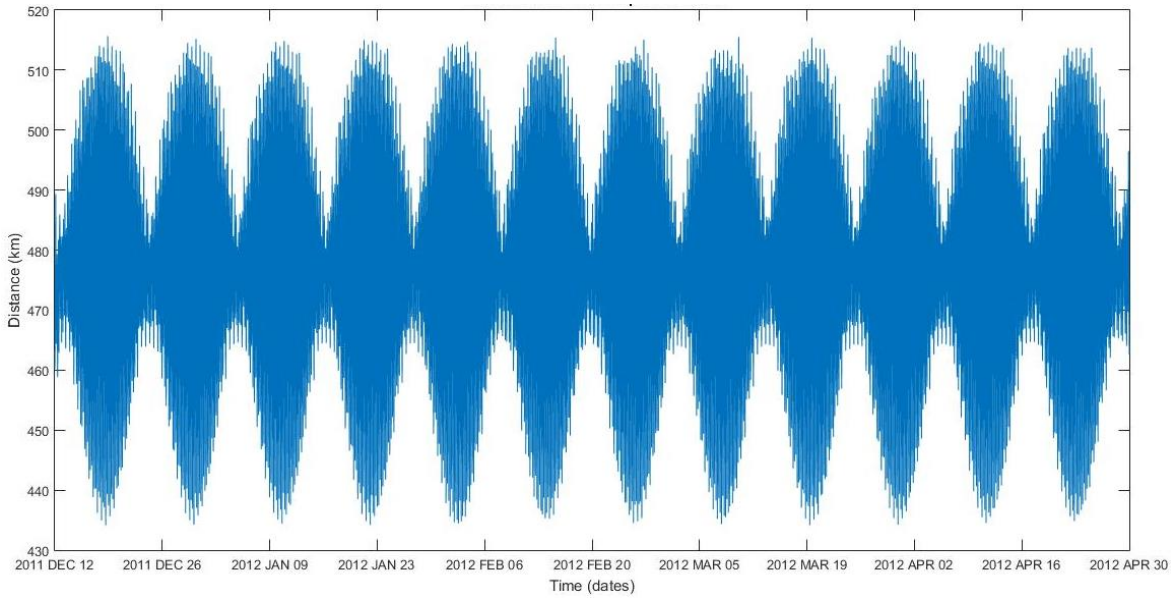


Figure 29. Dawn range from Vesta during LAMO orbit.

Now, form these figures and comparing them with the test example, it is proved that the representation of the Vesta scientific orbits is absolutely clearer and easier to extract an analysis. Figure 28 shows that the variations regarding the inclination during the four months of this orbit are a bit significant, which allowed the spacecraft to carry out the study of the asteroid in a wide area.

As figure 28 provides information about the inclination of the LAMO orbit, figure 29 supplies data about its height. It shows that along its time period, there were regular occasions when the spacecraft increased the eccentricity of its elliptical orbit, so the closest and farthest positions were considerably different, while between these periods the orbit remained almost circular. The difference between the minimum and maximum ranges in the most eccentric periods could even achieve the 18% of the maximum range, while in the almost circular orbits this difference is around the 2%. This allowed the spacecraft to have different perspectives of the surface performing a unique orbit.

Apart from these separated plots, if a comparison study was required instead of a deep and specific one, it is possible to plot the trajectory, range and velocity of all scientific orbits in a single graph. This is carried out introducing a loop in the code that executes the *trajectory_Vesta* for all *trajectory* values. So, having the values for each orbit, they are plotted using the *hold on* MATLAB instruction and adding the corresponding legend to identify all the orbits.

After adding all these changes, the plots of trajectory, range and velocity of all scientific orbits are displayed below.

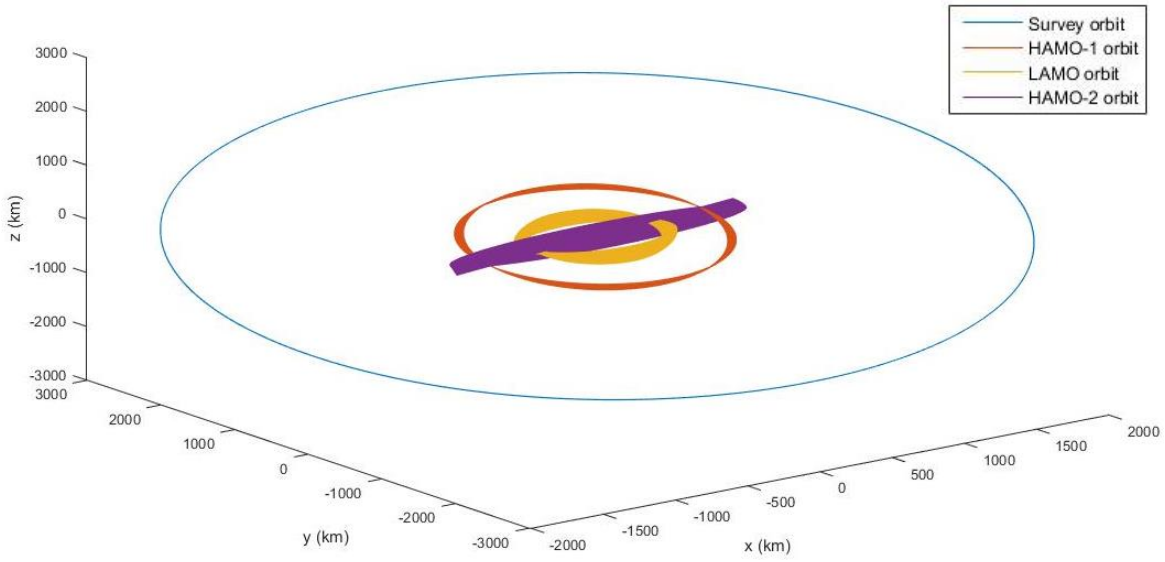


Figure 30. Dawn trajectory around Vesta during all scientific orbits.

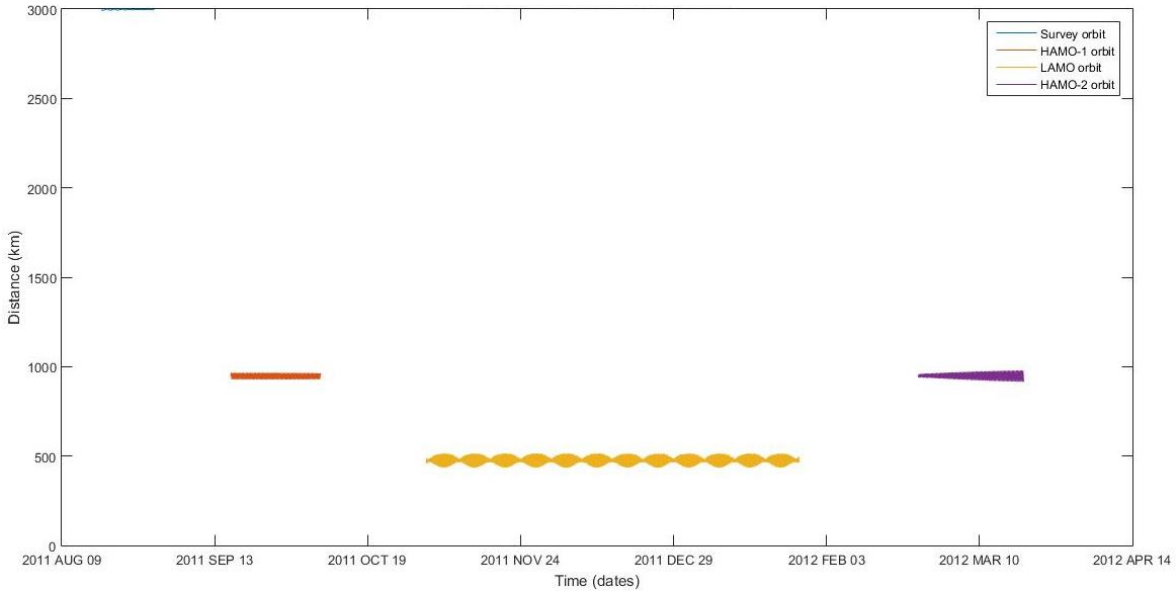


Figure 31. Dawn range from Vesta during all scientific orbits.

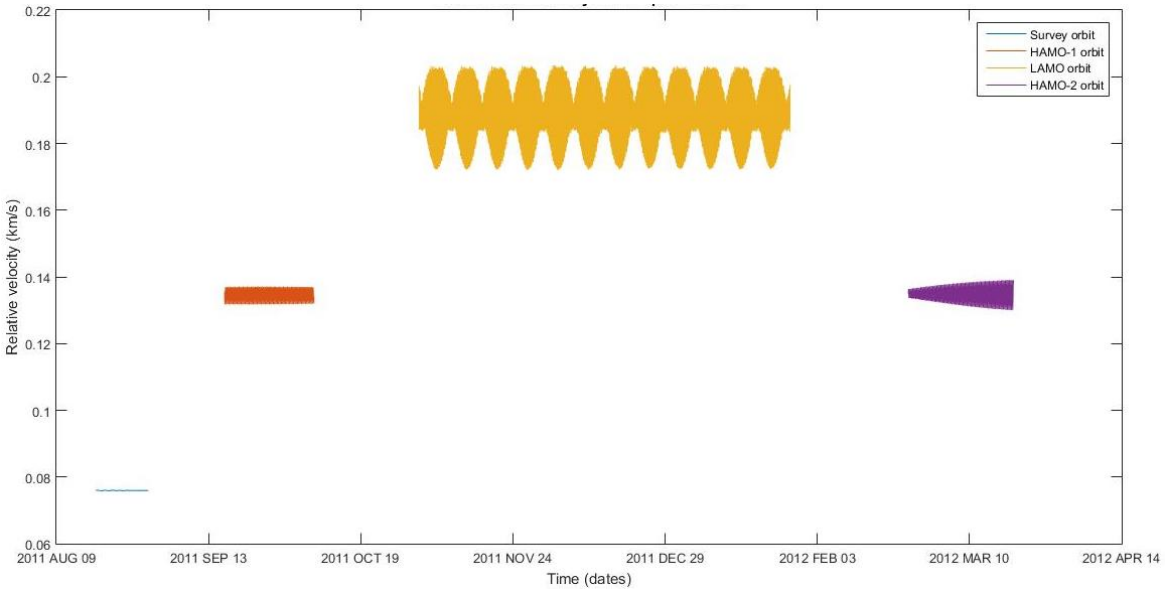


Figure 32. Dawn relative velocity with respect to Vesta during all scientific orbits.

As it was settled before, these plots do not provide an accurate representation from which exact values could be easily extracted or an exhaustive analysis cold be carried out. Instead of this, these figures make clearer the differences between the scientific orbits and their location along the Dawn mission around Vesta.

An important fact shown in figure 30 is that all the orbits are performed in approximately the same plane with the same inclination except the last one, HAMO-2. This means that the transfer orbits between LAMO and HAMO-2 must include a plane change.

Moreover, from figure 31 it can be seen that the survey orbit is significantly away from the rest of the orbits. Then, the HAMOs orbits have the same range although they are on different planes. Besides, HAMO-2 begins with an orbit close to a circular one but its eccentricity became bigger and bigger, what also makes it different from HAMO-1, which maintains its eccentricity constant. Finally, LAMO range is half of the HAMOs.

Figure 32 makes even more considerable the eccentricity changes of LAMO and HAMO-2 orbits, as these changes are more noticeable regarding the orbiting velocity. Furthermore, these changes are more significant as the spacecraft gets closer to the asteroid, as it is can be seen in these two orbits representations. In figure 31 the range changes are almost the same in these two orbits but figure 32 enhances these variations in the LAMO orbit due to its proximity to Vesta.

4.1.4 Vesta escape trajectory

The scientific orbits performed in different ranges mean a relevant characteristic stage of Dawn's mission. Nevertheless, it is not the one that makes this mission unique, but the fact that the spacecraft was able to orbit not only one small body such as Vesta but then a second one, Ceres. What makes this possible is the ability of Dawn to escape from Vesta's gravitational influence and begin a cruise journey to Ceres, so this specific case is computed, represented and studied in this section.

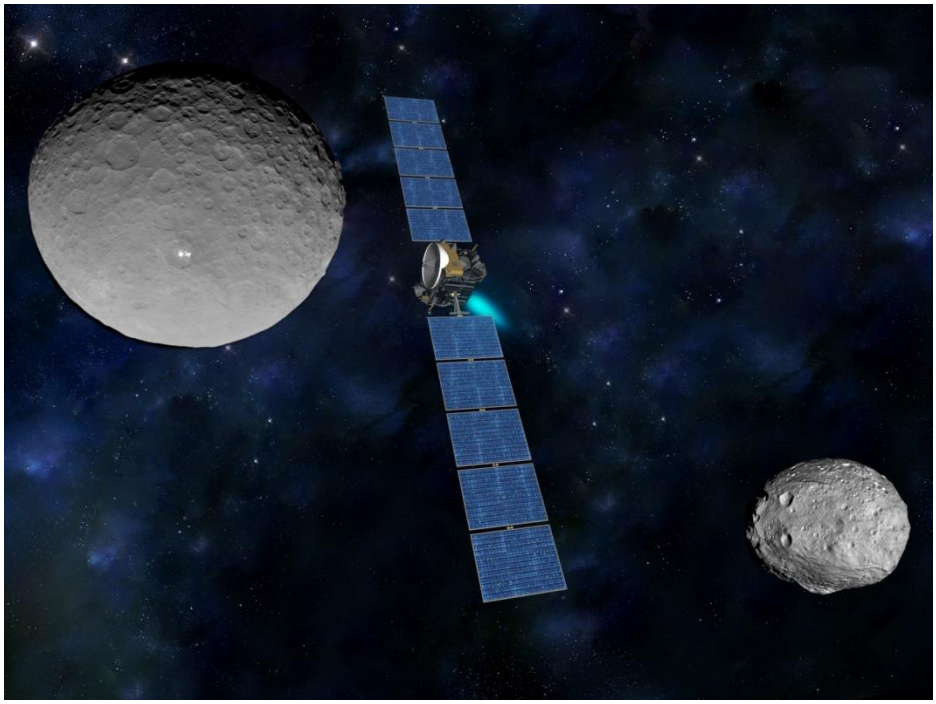


Figure 33. Dawn travelling from Vesta to Ceres [24].

In order to explain this computing code, the code of the position and velocity calculation is employed as the reference and the needed modifications and additions are described here.

From this, first of all there is the kernel files matrix. The required files are the time conversion and bodies' orientation ones, just as in the test examples, and then, a specific kernel file which provides data about Vesta trajectory is added called *vesta_1900_2100.bsp*.

Finally, the Dawn trajectory kernel files are the ones corresponding to the dates that encompass this escape trajectory. Knowing that Dawn departs from Vesta officially on the 5th September 2012 and after some tests, it is decided that the escape time period takes up from the 18th August to the 2nd September 2012. So, from this, the only demanded kernel file is *dawn_rec_120724-120913_121213_v1.bsp*.

Continuing through the code, the *dates* parameter is modified with the initial and final dates of the Vesta escape trajectory case and the value of the *STEP* variable is maintained due to its increase of accuracy, where the time step is now 129 seconds (2 minutes approximately). Then these parameters are introduced in the *time_computation* function and the needed *times_sec* and *dates_plot* vectors are extracted.

After this *time_computation* function, the state vector is computed using the same input parameters as in the Vesta test example, where only the content of *times_sec* is different. Having extracted the position coordinates vector from the state vector and then having calculated the range vector, these position coordinates are introduced in a position matrix called ***r***, in which the x coordinates are allocated in the first column, the y coordinates in the second column and the z coordinates in the third one. The same process is performed for the velocities, whose components are extracted and introduced in a ***v*** velocity vector the same way as in the position matrix *r*. The total velocity vector is computed too.

Now, having these parameters, a new reference system different from the one of Vesta provided by SPICE is created, as one of the main objectives of this code is to determine the performance of the Dawn IPS thrusters, the orientation of the thrust force and its value. Instead of x-, y- and z- components, the new ones are the tangential ***t***, the radial ***h*** and the elevation ***b***. This reference system is called as the **Spacecraft Trajectory Reference System**. It is considered that the thrust force can be distributed into these three components and the computation of their vales can provide a more practical analysis for the user than the force distributed in the SPICE reference system. The expressions used to compute these components are:

$$\vec{h} = \frac{\vec{r}}{|\vec{r}|} \qquad \textbf{(4.1)}$$

$$\vec{b} = \frac{\vec{r} \times \vec{v}}{|\vec{r} \times \vec{v}|} \qquad \textbf{(4.2)}$$

$$\vec{t} = \vec{b} \times \vec{h} \tag{4.3}$$

where *h* is the radial vector (pointing in the opposite direction of Vesta), *b* is the elevation vector, *t* is the tangential vector, *r* is the position vector and *v* is the velocity vector.

These expressions are calculated inside a loop of *STEP* length in order to encompass all the *r* and *v* rows together with the *beta* angle, the angle from the tangential vector to the projection of the thrust in the t-h plane, which points to the same direction as the total velocity. The expression of this ***beta*** angle is:

$$\beta = \sin^{-1} \frac{\vec{v} \cdot \vec{t}}{|\vec{v}|} \tag{4.4}$$

Then, having computed these new parameters, it is considered that the maximum variation of the angle between the thrust vector and its projection in the t-h plane is calculated by comparing this angle at the beginning and end of the escape trajectory. This angle is called as *alpha*, so the maximum variation is called ***alpha_max***. This maximum change is computed from the angle between the initial vector *b* and the final one. It is determined that this maximum variation is about 0.3014º, which means that the elevation component of the thrust has almost no relevance.

Then, knowing this maximum alpha, it is considered that the elevation component of the thrust is distributed in a sinusoidal way depending on the *theta* parameter. This parameter corresponds to the theta angle of the orbit, so when it is 90º or 270º, the alpha angle reaches its maximum. This expression can be written as:

$$\alpha = \alpha_{max} \cdot \sin\theta \tag{4.5}$$

where α is the alpha angle (*alpha* in the code), α_{max} is the *alpha_max* variable and θ is the theta angle of the orbit.

This theta angle is determined by executing the *cart2sph* MATLAB function, which from the Cartesian coordinates of a trajectory (*x*, *y* and *z* in this case) it computes the spherical coordinates. So, one of the outputs of this function is the theta angle, called as *theta* in the code.

Now, as the objective of this part of the code is determining the distribution of the thrust force in this Spacecraft Trajectory Reference System, the total Dawn IPS thrust force must be firstly computed. In order to make the most accurate determination, the input power supplied to the IPS is found out knowing that the thrust value depends on it. Moreover, this IPS input power depends on the distance between Dawn and the Sun, due to the fact that it is provided by the solar energy from the solar arrays of the spacecraft, so the first parameter computed is the Dawn range from the Sun.

It is can be deduced that this range is easily computed by executing the *cspice_spkezr* function considering Dawn as the target body and the Sun as the observer one and calculating the *range* vector. To connect this Sun range to the input power, it is necessary to look at a JPL article made for the Dawn operations at Ceres, which collects information about all the mission phases that were already finished, including the Vesta escape [25].

From this article, there is a table which provides information about the Dawn distance to the Sun at the different mission stages as well as the IPS input power during them.

Description	Time Period	Distance S/C to Sun (AU)	Power Level To IPS (kW)	Comments
Launch	09/27/2007	1.0	NA	
Initial Check-out	09/2007 - 12/2007	1.0 - 1.16	2.6	ΔV = 0.07 km/s
Cruise prior to MGA	12/2007 - 11/2008	1.16 - 1.40	2.6	ΔV = 1.80 km/s
Optimal Coast and Mars Gravity Assist (MGA)	11/2008 - 06/2009	1.40 - 1.60	NA	ΔV = 2.60 km/s (From MGA)
Cruise to Vesta	06/2009 - 07/2011	1.40 - 2.26	2.6 -1.7	ΔV = 4.84 km/s *
IPS Operations at Vesta	07/2011 - 09/2012	2.26 - 2.53	1.7 -1.3	ΔV = 0.25 km/s
Cruise to Ceres and Approach	09/2012 - 04/2015	2.51 - 2.84	1.3 -0.9	ΔV = 3.85 km/s **
Ceres Science Operations	04/2015 - 06/2016	2.84 - 2.93	0.9	ΔV = 0.33 km/s
Total From IPS				ΔV = 11.14 km/s
Mission Total				ΔV = 13.74 km/s

Figure 34. Dawn mission summary until Ceres arrival [25].

As this table relates these two required parameters and having computed the real Dawn range to the Sun, it is possible to extract the real power supplied to the IPS in this specific moment.

The range provided by SPICE is in kilometres so, converting it into Astronomical Units, the resultant range is **2.55 AU**, which is stretchy close to the largest range in the *IPS Operations at Vesta* mission phase of the table and the smallest one of the *Cruise to Ceres and Approach* one. Locating the Vesta escape between these two stages and looking at the IPS power level column, this input power is found to be **1.3 kW**.

Now, having this IPS input power value, it is required to relate it to the corresponding thrust value. The Dawn IPS is distributed into various throttle levels, so the total thrust force value is extracted from the level that corresponds to this computed IPS input power. The data of the Dawn IPS throttle levels can be found in another JPL paper that describes the Dawn IPS, where there are some tables of the thrusters' performance specifications [26]. One of them

exposes the relation between these throttle levels, IPS input power and thrust force.

Throttle Levels						
TH	ML	PPU Input Power (EOL)		Thruster Input Power (EOL)	Engine-U Thrust (EOL)	
		(W)	Uncertainty	(W)	(mN)	Uncertainty
15	111	2496	1.04%	2318	91.5	1.13%
14	104	2376	1.07%	2206	86.8	1.17%
13	97	2254	1.10%	2090	82.1	1.22%
12	90	2129	1.14%	1973	77.1	1.28%
11	83	2004	1.18%	1857	72.4	1.34%
10	76	1866	1.23%	1728	67.2	1.43%
9	69	1732	1.30%	1602	62.4	1.52%
8	62	1601	1.37%	1479	57.2	1.64%
7	55	1473	1.45%	1359	52.0	1.78%
6	48	1362	1.54%	1253	47.4	1.94%
5	41	1239	1.66%	1137	42.2	2.16%
4	34	1116	1.81%	1020	37.0	2.44%
3	27	992	2.00%	902	31.8	2.82%
2	20	812	2.21%	732	26.3	3.23%
1	13	714	2.11%	639	23.9	3.24%
0	6	606	2.06%	537	20.9	3.37%

Figure 35. Dawn IPS thrusters' performance specifications [26].

It can be seen that there are specified two input powers, one to the thruster and another to the PPU of the IPS. As it was described in section 2.2, the power supplied to the IPS is distributed firstly by the PPUs to the thrusters, which are placed at the end of the system. From this, it is deduced that the IPS input power is practically the same as the PPU input power, so, knowing that it is 1300 W, the corresponding throttle level is found to be the number **6**.

Now, in order to make a practical procedure of the computation of the thrust force from this throttle level, a new function called *Dawn_thrust* is created. The only input of this function is the determined throttle level (called as *throttle_level* in the code) and inside the function a *switch* block is again executed, just as in the scientific orbits function.

This switch block has as the main parameter the throttle level value and the different path options correspond to the thrust force values related to each throttle level extracted from the table in figure 35. In this Vesta escape trajectory case, as the level is number 6, the total resultant IPS thrust force is found to be

0.0474 N. This thrust value is also the only output parameter of this *Dawn_thrust* function.

Returning to the main computing code, it is time to compute the distribution of this thrust force during the escape trajectory. For this, a loop is generated of every position within the previously computed trajectory. Firstly, inside this loop, the *alpha* angle is computed from expression (4.5). Having this alpha angle computed as well as *beta* angle from the loop in charge of calculating the components of the Spacecraft Trajectory Reference System and the total thrust force value, it is possible to determine the components of this thrust force within this reference system.

The expressions followed to compute the thrust force components of this new reference system are:

$$F_{T-h} = F_T \cdot \cos\alpha \cdot \sin\beta \quad \textbf{(4.6)}$$

$$F_{T-t} = F_T \cdot \cos\alpha \cdot \cos\beta \quad \textbf{(4.7)}$$

$$F_{T-b} = F_T \cdot \sin\alpha \quad \textbf{(4.8)}$$

Where the $F_{T\text{-}h}$, $F_{T\text{-}t}$ and $F_{T\text{-}b}$ are the components of the thrust force in the Spacecraft Trajectory Reference System, F_T is the total IPS thrust force, α is the alpha angle and β is the beta angle.

Then, it is possible from these expressions to convert these components into Cartesian components of the SPICE default reference system. These Cartesian components are:

$$F_{T-x} = F_{T-h} \cdot h_x + F_{T-h} \cdot t_x + F_{T-b} \cdot b_x \quad \textbf{(4.9)}$$

$$F_{T-y} = F_{T-h} \cdot h_y + F_{T-h} \cdot t_y + F_{T-b} \cdot b_y \quad \textbf{(4.10)}$$

$$F_{T-y} = F_{T-h} \cdot h_z + F_{T-h} \cdot t_z + F_{T-b} \cdot b_z \quad \textbf{(4.11)}$$

where the $F_{T\text{-}h}$, $F_{T\text{-}t}$ and $F_{T\text{-}b}$ are the components of the thrust force in the Cartesian SPICE reference system.

Having computed all the desired parameters, it is time to plot the results. Firstly, as was done in the previous computing codes, the position, range and relative velocity of Dawn spacecraft during Vesta escape trajectory are represented employing the same format as until now. Vesta asteroid is again remarked with a big black star in the origin of coordinates and the dates introduced in the range and relative velocity plots are the ones determined before in the *dates_plot* vector.

From this, these three plots are the ones represented below:

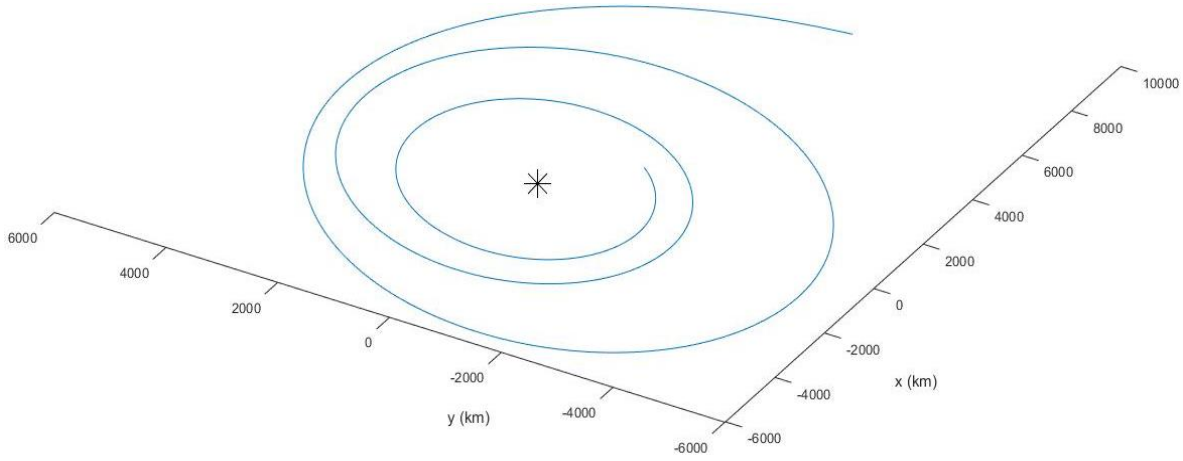


Figure 36. Dawn escape trajectory around Vesta.

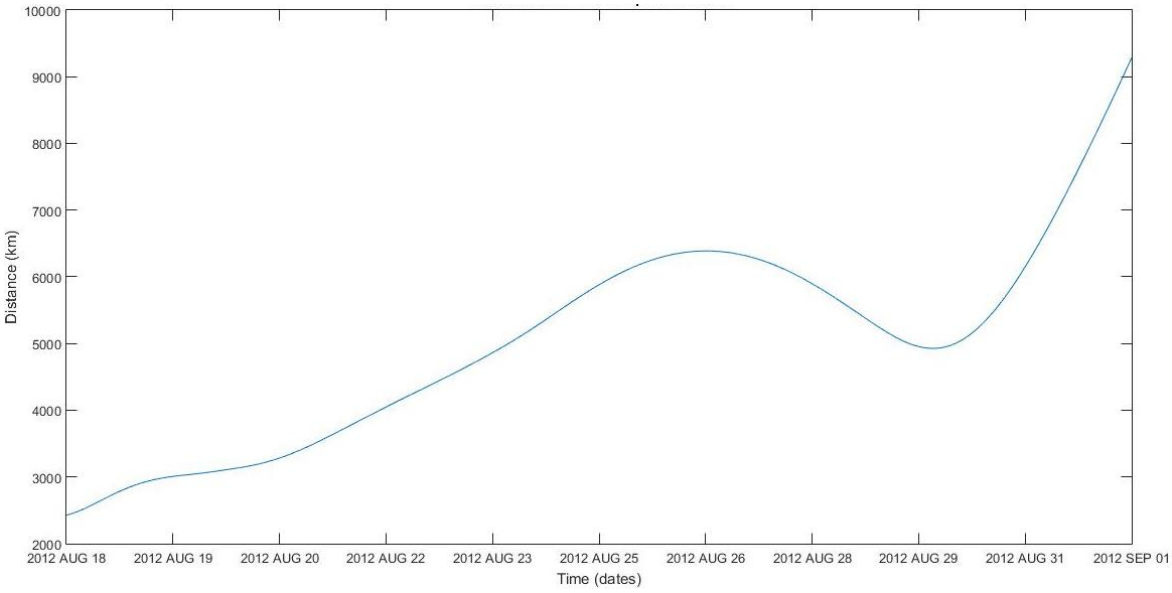


Figure 37. Dawn range from Vesta during the escape trajectory.

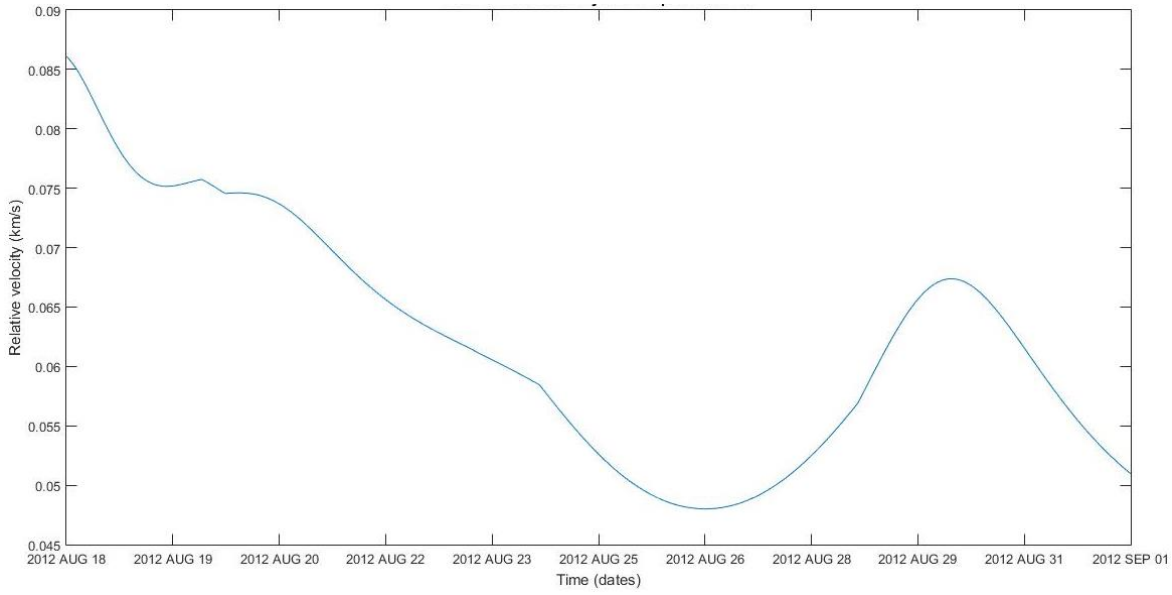


Figure 38. Dawn velocity relative to Vesta during the escape trajectory.

In a first look to these three figures, it can be noticeable that the escape trajectory around Vesta performed by Dawn does not follow a regular track due to the complex operation of the Dawn IPS.

One important characteristic of figure 36 is that, compared to other trajectory plots around Vesta, its axes are not placed in the same way. This is because of the plane change between LAMO and HAMO-2 scientific orbits, which provoked that the new inclination just coincided with the perspective of the user, so the trajectory was unable to be seen. For this reason, the axes were moved to a point where the trajectory could be clearly visible and legible.

Furthermore, this figure proves that the inclination change during the escape trajectory is almost negligible in a general view, which makes the trajectory easier to identify and to be studied with simpler methodologies. Moreover, as it is known that a plane change manoeuvre implies a great cost in terms of fuel consumption, making it even greater if it is combined with an orbit height change such as this, it is understandable that this plane change was performed just after the LAMO because low altitudes minimise this fuel consumption considerably.

Figure 37 provides more information about the type of movement that the spacecraft performs during the escape trajectory. In figure 36 it seems that Dawn performs a manoeuvre similar to a low-thrust continuous climb but this range plot confirms that it is not like that.

At first, it may seem that the spacecraft moves away from Vesta in a regular way but then some variations make the trajectory a unique one, with even a little approach before the last impulse to exit. Nevertheless, the next section 4.2 will focus on this escape trajectory in order to make an approximation from theoretical models. Its objective will be to prove if this orbit is certainly similar or not to a low-thrust continuous climb as well as other methodologies that will be explained in the section.

Then, figure 38 makes even more noticeable these variations along the escape trajectory, demonstrating its complexity when it comes to analysing it. Specially, the most relevant irregularity is present at the Dawn approach just after the final impulse, where the relative velocity highly increments.

Besides, the little deviations of figure 37 at the beginning of the manoeuvre stand out even more in this relative velocity plot. This again confirms that the variations are more significant in the velocity representation when the range is smaller.

After these trajectory known plots, the thrust force components distribution in the Spacecraft Trajectory Reference System along the Vesta exit trajectory is also and represented.

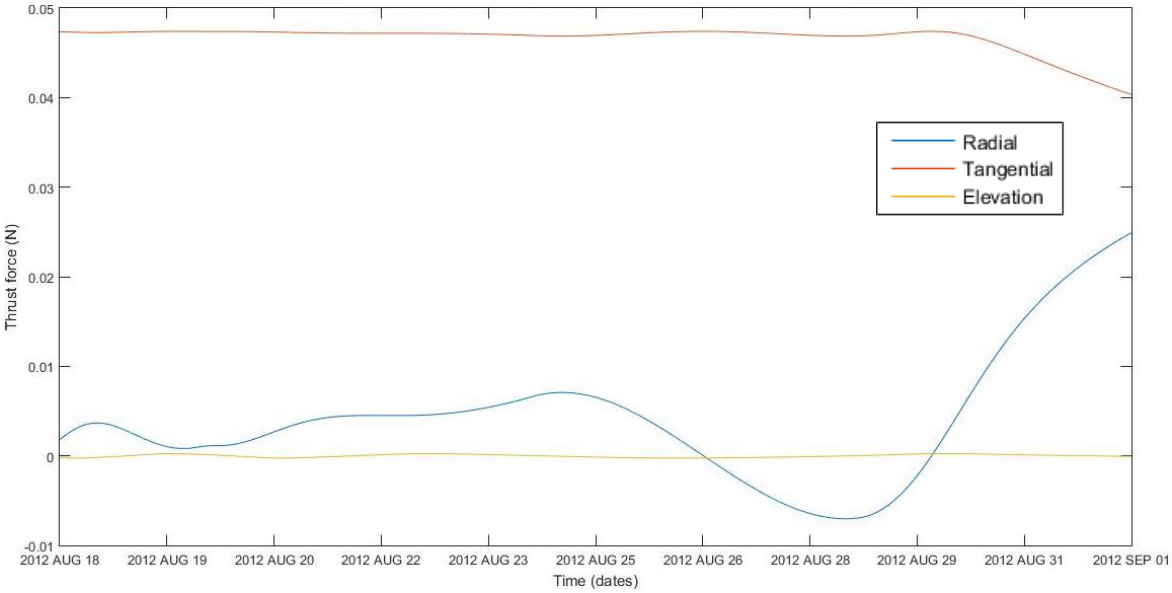


Figure 39. Dawn IPS thrust force components distribution.

This figure 39 shows not only the values of the IPS thrust force components in the Spacecraft Trajectory Reference System but also the division between them taking into account the total thrust value. The mean values provide useful conclusions as well as the evolution of the values along the escape trajectory.

Firstly, the elevation component has the same relevance as the one it was predicted, that is, almost none. The little ripples are caused by the sinusoid application of alpha angle established in expression (4.5), but the fact that the maximum alpha variation was only about 0.3014º makes even the peak values insignificant compared to the other components.

Focusing on the other two components, which really contribute at the thrust application, the tangential is the one which has clearly more prominence during most of the manoeuvre. Then, in the last phase of this escape trajectory, its importance is lowered to lead to a greater significance of the radial component, which achieves a thrust force value very close to the tangential one.

The evolution of these components has a lot of sense even if it seems that the thrusters' performance is totally irregular. At the beginning of the exit manoeuvre, the spacecraft's goal is to move away as much as possible from Vesta. This is why in figure 37 the range increases in an approximate constant way. In order to achieve this movement, the thrust must be applied in the tangential component. Therefore, this component is the predominant one.

Then, as Dawn has accomplished this first goal, the final objective is to obtain an enough impulse to finish the exit spiral and start a cruise trajectory to Ceres. This is reached thanks to an abrupt increment of the orbit eccentricity, a parameter which is increased thanks to a force applied in the radial direction, so this is why the radial thrust component rises at this final stage. Additionally, a low radial thrust force is applied pointing to Vesta to acquire an enough velocity (see figure 38) to get the impulse to escape from Vesta.

4.2 Theoretical escape from Vesta

As it was said at the beginning of this chapter, the second block of the created codes is the one which contains computations which are not performed by the SPICE database library and toolkit. Instead of that, the calculations are accomplished thanks to the application of theoretical models and methodologies in order to determine parameters are may be already given by the SPICE database or others provided by other sources with their reliability confirmed.

The main objective of this block is to check the accuracy of the use of these theoretical expressions when determining the behaviour of the Dawn spacecraft in specific cases and confirm if they are feasible or not. This section focuses on this second block, especially on the analysis of an already studied selected case in the Dawn spacecraft mission: the escape trajectory of the spacecraft from Vesta asteroid.

Figure 40. Dawn's journey from Vesta to Ceres beginning [27].

As it was mentioned in the section 4.1.4, this specific case of the Dawn mission is the one that characterises more the uniqueness of the spacecraft and its mission. What makes Dawn unique is the fact that the Dawn IPS enabled it to not only approach to one absolute small astronomical body compared to other past mission such as the asteroid Vesta and perform several scientific orbits at different altitudes around it, but also its capability to escape from the gravitational influence of this body, the achieving of a new cruise trajectory and finally again the performance of a second approach and scientific orbits around Ceres. Dawn has the distinction of being the first spacecraft to orbit two extraterrestrial bodies.

One of the designed codes that extract results employing exclusively data from SPICE was one that computed the Dawn spacecraft's escape from Vesta, which corresponds to section 4.1.4.

This code provided several useful results, such as the spacecraft's trajectory, relative velocity respect to Vesta, range from Vesta or the components of the thrust force given by the Dawn's IPS. All this data is referred to the time period when the spacecraft performed its final orbits around Vesta, was propelled to escape from Vesta's gravitational influence and began its second journey, which ended at Ceres. This time period corresponds to the one from 18th August to 2nd September, both in 2012.

From all this extracted results, the Dawn's trajectory will be the one which this section is going to focus on. This section's main objective is to compute the most possible accurate escape trajectory using only the given initial position and velocity of the spacecraft provided by the mentioned code. Then, the computed trajectory and the one provided by SPICE, that is, the trajectory considered as the real one, will be compared. Furthermore, from this calculated trajectory, the total relative velocity with respect to Vesta and the distance between the spacecraft and the asteroid are other parameters that will be compared.

Apart from this trajectory, two additional parameters will be determined. The first parameter will be the IPS thrust required to propel the spacecraft from the initial position to the final one considering that Dawn escapes from Vesta's gravitational influence at the final position. The computation of this theoretical IPS thrust and its comparison with the real exerted thrust will determine if the Dawn's trajectory of the selected dates is accurate enough to consider it as the Vesta escape trajectory.

The second calculated parameter will be the time of flight during the trajectory. The computation of this parameter will be performed considering different methodologies, so different values of this parameter will be determined. Finally, as it happens with the rest of the results, it will also be compared with the actual Dawn's time of flight along its real trajectory.

The kernel files needed to perform this code were the same as in the code of section 4.1.4 except the old Vesta kernel file (*vesta_1900_2100.bsp*), which was only needed to compute the position and velocity of Vesta respect to the Sun.

4.2.1 Trajectory

4.2.1.1 Computation procedure

As it was just mentioned, the computation of this theoretical escape trajectory will only have as inputs the initial position and velocity of the spacecraft. The trajectory will be assumed to be quasi-circular, with constant thrust and determined by a climb angle (γ in figure 41). Furthermore, the theoretical model employed in these calculations assumes that the spacecraft wants to travel from a closer orbit to a farther one applying a low thrust manoeuvre.

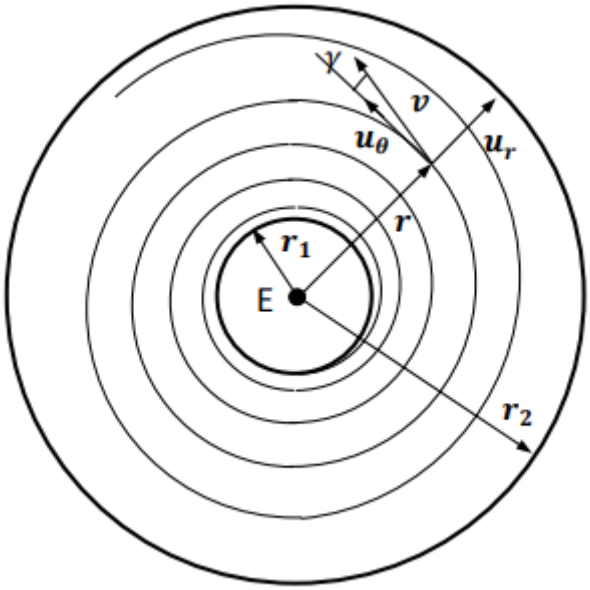


Figure 41. Low thrust change of orbit manoeuvre [28].

The first step is to create a 1x3 vector *r* and a 1x3 vector *v* which will allocate the *x*, *y* and *z* coordinates of this initial position and the v_x, v_y and v_z components of the velocity relative to Vesta respectively. This input data is given by the code in section 4.1.4 thanks to the *cspice_spkezr* function from SPICE. Moreover, as the computation is performed, the *r* and *v* vectors will become matrices which will allocate their corresponding data at each time step, as it happens in all the codes related to trajectory computations.

So, in order to begin the trajectory computation, these vectors together with the subsequent calculations, are introduced in a loop whose length corresponds to the already known *STEP* parameter, which determines the number of time steps present along the trajectory, that is, the trajectory accuracy.

Firstly, these vectors are employed to create a Spacecraft Trajectory Reference System, which was already explained in detail in section 4.1.4. The resulting vectors of this reference system are the vectors *h*, *b* and *t*.

Nevertheless, as it was also shown in section 4.1.4 that the change of plane was about 0.3º, so it can be considered as negligible. Therefore, the parameters along the *b* direction will be negligible too.

Following the Spacecraft Trajectory Reference System, there is the time counter represented in the *time* parameter. It accumulates the time spent of the trajectory by adding the time step period at every loop. It is also known that the time step (*time_step*) is the quantity of time spent between every computation step, so it is directly related to the *STEP* parameter.

After this, the climb angle *γ* is computed. This angle is determined by the angle between the total spacecraft velocity vector and the tangential trajectory vector (or the tangential velocity) [28]. Assuming the quasi-circular approximation,

$$1 - \sqrt{\frac{r_i}{r(t)}} = a_T \cdot t \cdot \sqrt{\frac{r_i}{\mu_V}} = \frac{a_T \cdot t}{v_i} \quad \textbf{(4.12)}$$

$$r(t) = \frac{r_i}{\left(1 - \frac{a_T \cdot t}{v_i}\right)^2} \quad \textbf{(4.13)}$$

$$v_r = \frac{dr}{dt} = \frac{2 \cdot r_i \cdot a_T}{v_i} \cdot \frac{1}{\left(1 - \frac{a_T \cdot t}{v_i}\right)^3} \quad \textbf{(4.14)}$$

$$v_t = \sqrt{\frac{\mu}{r(t)}} \quad \textbf{(4.15)}$$

$$\gamma = \tan^{-1} \frac{v_r}{v_t} \quad \textbf{(4.16)}$$

where r_i is the initial distance from Dawn to Vesta, $r(t)$ is the distance from Dawn to Vesta at a specific time, a_T is the spacecraft acceleration generated by the thrust force, t is the time spent from the initial position to a specific one, μ_V is the standard gravitational parameter of Vesta, v_i is the initial velocity of Dawn relative to Vesta, v_r is the radial component of the Dawn's velocity relative to Vesta, v_t is the tangential component of the Dawn's velocity relative to Vesta and γ is the climb angle.

The Dawn's acceleration generated by the thrust force can be computed employing the well-known Newton's Second Law:

$$a_T = \frac{F_T}{m_{s/c}} \quad \textbf{(4.17)}$$

Where F_T is the thrust force exerted by the Dawn IPS and $m_{s/c}$ is the current mass of the Dawn spacecraft.

This current mass of Dawn is given by the data provided by the JPL in an article just after the Dawn's arrival to Ceres [25]. This article also provides the data of the spacecraft's mass at launch.

Description	Mass, kg
Dry spacecraft and avionics (except IPS)	573
Science instruments	46
Hydrazine	45
Ion Propulsion System (IPS)	129
Xenon	425
Flight system mass at launch	**1218**

Figure 42. Dawn's total mass at launch [25].

Description		Xenon Allocation (kg)
Initial Checkout-	Actuals	3.1
Deterministic Thrusting To Vesta	Actuals	246.2
Vesta Operations	Actuals	10.3
Deterministic Thrusting To Ceres	Actuals	118.0
Approach to Ceres	Actuals	12.4
Allocation for Ceres Operations		9.1
Allocation for Leaks and Thruster Restarts		1.0
Main Tank Residuals		5.0
Margin		20.1
Totals	425.2	

Figure 43. Dawn's Xenon mass employed through different stages [25].

From figure 43 it is shown that the Xenon mass used until the final operations at Vesta is 259.8 kg and figure 42 exposes that the Dawn's total mass at launch was 1218 kg, so the remaining spacecraft mass at the end of Vesta operations and at the beginning of the cruise trajectory to Ceres is **958.4 kg**.

On the other hand, the thrust force data is given by the same JPL article, which provides the IPS input power at this stage of the mission, together with another JPL article which provides the correspondence of the IPS input power with the resultant thrust force given by the thruster [26]. This procedure was explained in detail in section 4.1.4, where this thrust force is needed to compute its distribution along the axes of the spacecraft trajectory reference system. This resultant thrust force exerted by the thrusters was found to be **0.0474 N**.

From these expressions, the resultant climb angle is computed. The next graph shows the evolution of this climb angle within the theoretical exit trajectory.

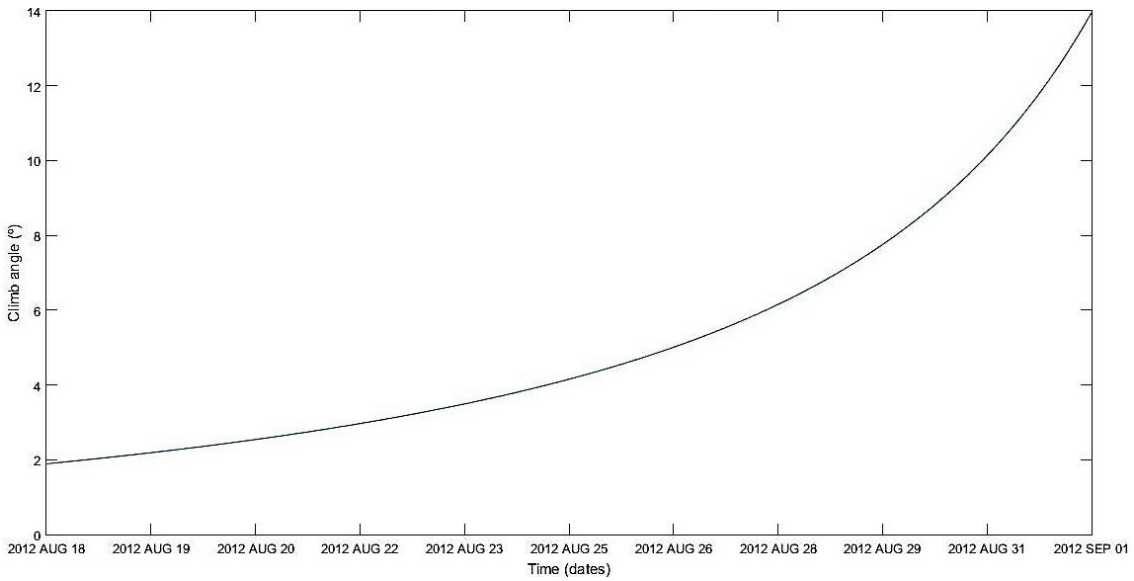


Figure 44. Climb angle along the theoretical exit trajectory.

As figure 44 shows, the climb angle of the spacecraft trajectory continuously increases, making this growth more prominent as Dawn moves away from Vesta. This increment is caused by the augmentation of the radial velocity due to the increase of the time spent as well as the decrease of the tangential velocity due to the fact that Dawn gets away continually from Vesta. This continuous but not constant increase of the climb angle makes the trajectory look like a spiral similar to the one of figure 41.

Now, having computed the Dawn's velocity relative to Vesta at both axes, tangential and radial to the actual orbit, this same velocity is now calculated in the Vesta's reference system, which is the one employed by SPICE. This means that the initial velocity given by the SPICE computations is no longer used and is substituted by this new calculated velocity. Knowing this, the v_r and v_t components are turned into the usual v_x, v_y and v_z components. This reference system change is performed by the following expressions.

$$v_x = v_r \cdot h_x + v_t \cdot t_x \quad \textbf{(4.18)}$$

$$v_y = v_r \cdot h_y + v_t \cdot t_y \quad \textbf{(4.19)}$$

$$v_z = v_r \cdot h_z + v_t \cdot t_z \quad \textbf{(4.20)}$$

where v_x, v_y and v_z are the components of the spacecraft velocity relative to Vesta, h_i are the components of the radial vector and t_i are the components of the tangential vector. As it was previously mentioned, the velocity at the *b* elevation vector is negligible. The components of the velocity in the Vesta's reference system are introduced in the *v* matrix.

Now, having the components of the velocity at Vesta's reference system, it is possible to compute the acceleration of the spacecraft along its exit trajectory. As it happened in the velocity, the acceleration is also composed by a radial and a tangential component.

A significant part of the total Dawn's total acceleration is the one supplied by the spacecraft's IPS, which was computed thanks to the formula **(4.17)** and the data provided by the JPL articles [25] [26]. Moreover, the other part of the acceleration that contributes with the computation of the total one is the gravitational acceleration generated by Vesta. From this, the determination of the total acceleration is given by the expressions below.

$$a_r = a_T \cdot \sin\gamma - \frac{\mu}{r(t)^2} \quad \textbf{(4.21)}$$

$$a_t = a_T \cdot \cos\gamma \quad \textbf{(4.22)}$$

$$a_x = a_r \cdot h_x + a_t \cdot t_x \quad \textbf{(4.23)}$$

$$a_y = a_r \cdot h_y + a_t \cdot t_y \quad \textbf{(4.24)}$$

$$a_z = a_r \cdot h_z + a_t \cdot t_z \quad \textbf{(4.25)}$$

Where a_r is the radial component of the Dawn's acceleration relative to Vesta, a_t is the tangential component of the Dawn's acceleration relative to Vesta, and a_x, a_y and a_z are the components of the spacecraft acceleration relative to Vesta in the Vesta's reference system.

The radial component of the total acceleration is determined by the acceleration produced by Dawn IPS and the one produced by the gravitational attraction of Vesta, and its tangential component is only influenced by the IPS thrusters too. The acceleration of the thrusters employed is split up in its two components of the Spacecraft Trajectory Reference System, so the climb angle must be used to compute these components.

Besides, as it happened in the velocity computation, the reference system is again changed to the Vesta's one, so the acceleration components within this reference system are calculated. Additionally, these acceleration components are introduced in a matrix called *a*, which refers to the acceleration at every time step and has the same format as the *r* and *v* matrices.

Finally, the only step left in the loop is determining the position and velocity of the following iteration, since they are the required parameters needed to perform all the procedure exposed until now. As a trajectory iterative process, the parameters applied to calculate this next-step data are the current position, velocity and acceleration, allocated in the matrices *r*, *v* and *a* respectively. These values are employed as well as the time step spent until the next position, allocated in the *time_step* parameter. As the acceleration is considered constant during the time step period, the expressions of the constantly accelerated motion are used.

$$r(t+1) = r(t) + v(t) \cdot t_s + a(t) \cdot t_s^2 \quad \textbf{(4.26)}$$

$$v(t+1) = v(t) + a(t) \cdot t_s \quad \textbf{(4.27)}$$

Where *r(t+1)* and *v(t+1)* are the position and velocity vectors respectively at the next time step period, and t_s is the time step period.

Having computed these parameters, the following iterations can be performed until the time step number achieves the *STEP* parameter value.

4.2.1.2 Results and plots

As it was explained at the beginning of this section, the most considerable result that will be determined, analysed and compared with the data from SPICE is the trajectory itself and then, from this trajectory, the distance between Dawn and Vesta and the total spacecraft's velocity relative to Vesta are secondary results that will also be taken into account.

First of all, from the procedure above, the resultant estimated exit trajectory around Vesta is the one below.

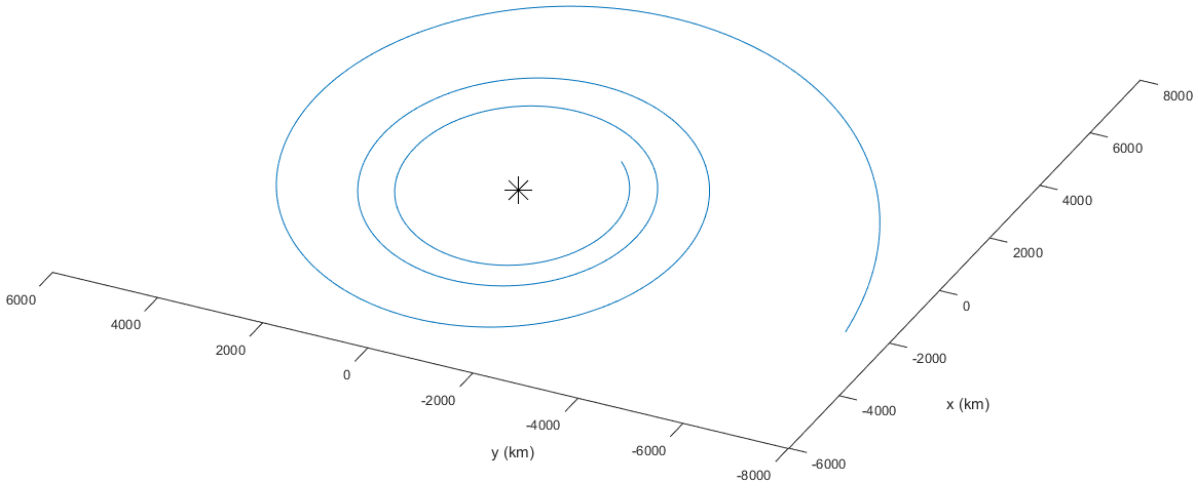


Figure 45. Theoretical Dawn exit trajectory around Vesta.

The blue line represents the theoretical Dawn exit trajectory and the black star represents the asteroid Vesta.

As it is shown in figure 45, the resultant theoretical exit trajectory is significantly similar to the trajectory given by the general low-thrust change of orbit manoeuvre represented in figure 41. Moreover, as it was mentioned in the computation procedure section, the spiral's climb angle increases as the spacecraft moves away from Vesta, so the trajectory does not represent a regular spiral due to a continuous climb. Furthermore, the negligibility of the plane change is also noticeable.

Although its ideal regularity, at first sight it seems that the computation procedure is able to estimate considerably a reliable approximation of the real Dawn's escape trajectory. However, this computed trajectory is below directly compared to the real one, which is provided by SPICE.

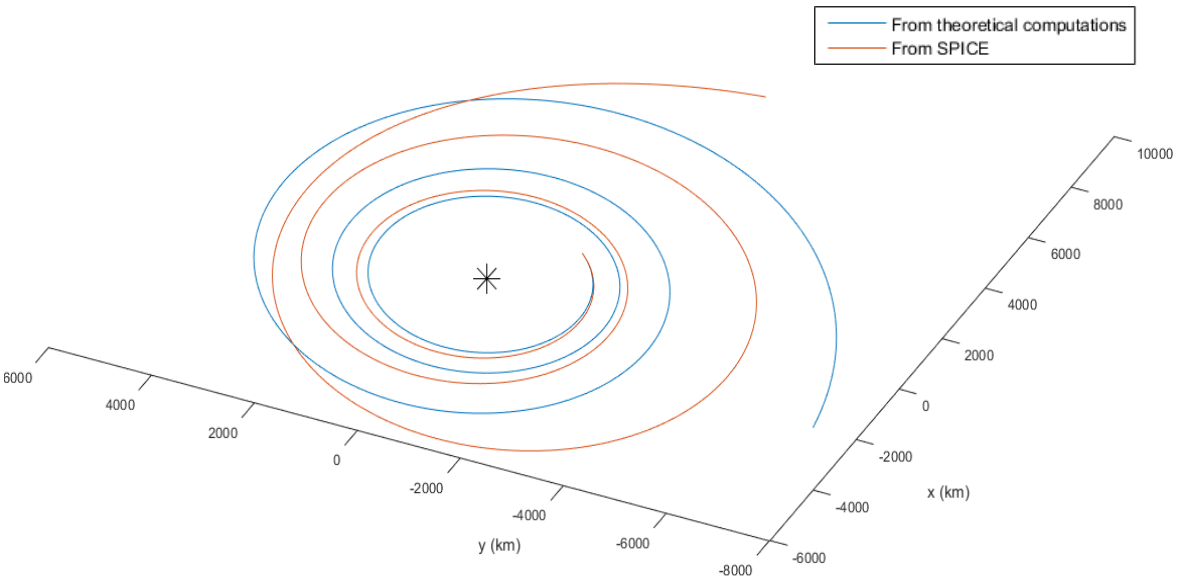


Figure 46. Theoretical and real (SPICE) Dawn exit trajectory around Vesta.

Now figure 46 definitely demonstrates the similarity between the real Dawn escape trajectory extracted from SPICE and the one resultant from the theoretical computations' procedure.

As it is known, the trajectory represented by the positions provided by SPICE is not regular due to the fact that the thruster is clearly not applied in a predictable way during the entire trajectory or even is directly not applied at some segments of it. This is why the computed trajectory is certainly very similar to the real one until the spacecraft acquires a greater orbit than the theoretical one (approximately after 2 orbits from the initial position), then it goes to a closer distance, even closer than the theoretical one, to finally achieve the final impulse to escape from Vesta's gravitational influence.

It must be taken into account that the computed trajectory considers it as a change of orbit manoeuvre, so that is why a final impulse is noticeable in the real trajectory but not in the theoretical one. Furthermore, the differences between both trajectories at the end of the escape phase make even more evident the non-regularity of the movements and manoeuvres of the Dawn spacecraft caused by the also irregularity of its IPS performance or application.

As it was said before, from this plot it can be extracted two significant parameters to understand better the comparison and make a more accurate analysis. These parameters are the total relative Dawn's velocity with respect to Vesta and the distance between the spacecraft and the asteroid. The only operation to compute these parameters is to calculate the module of the vectors *r* and *v*. After this brief step, the resultant computed ranges and total velocities compared to the real SPICE ones are the ones below.

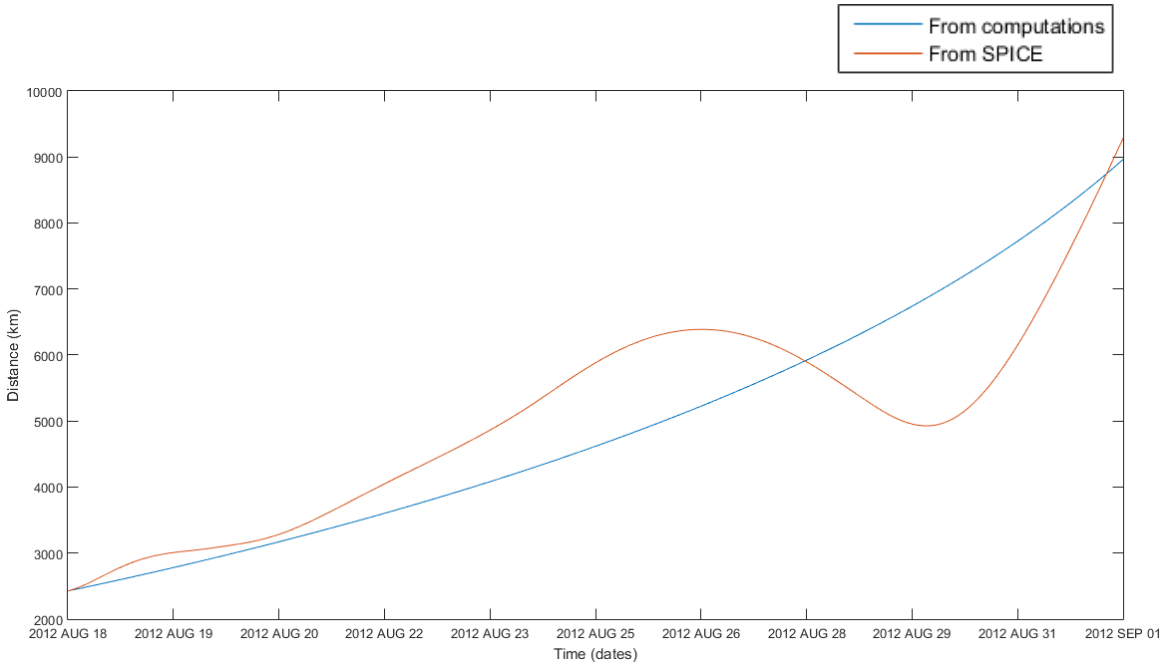


Figure 47. Theoretical and real (SPICE) distance between Dawn and Vesta during exit trajectory.

As it was deduced in the trajectories' analysis, the Dawn's range extracted from the theoretical computation increases gradually proportionally with the climb angle.

On the other hand, the real Dawn's range also increases at a similar rhythm as the theoretical one until half of the total escape time. Nevertheless, the acquirement of a greater orbit mentioned in the trajectories' analysis is clearly represented from the 23rd to the 28th of August. Furthermore, the subsequent approach is represented from the 28th until almost the end of the escape trajectory, when the final impulse is exerted in the real escape until achieving again a certainly similar value to the theoretical one.

This graph provides and confirms data crucial to prove the irregularity of the employment of the IPS but the valid precision of the low-thrust manoeuvre approximation to obtain an estimation of the real trajectory.

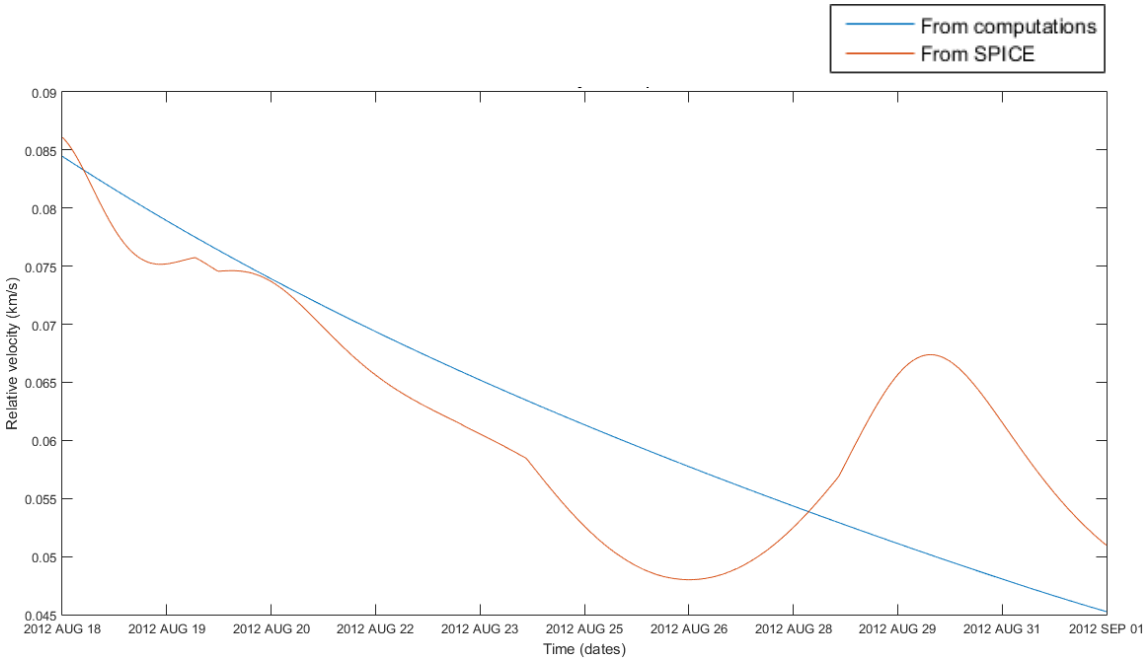


Figure 48. Theoretical and real (SPICE) Dawn's total velocity relative to Vesta.

Figure 48 exposes almost the same trajectories characteristics as in figure 47. As the range increases, the velocity decreases, so the evolution of this velocity is clearly related to the evolution of the range in figure 47. Besides, the variations of the real trajectory parameters respect to the computed ones are clearly remarked in this plot.

Finally, considering all these results and although the evident but unavoidable parameters deviations, it can be concluded that the low-thrust change of orbit manoeuvre can be considered as an accurate approximation of the trajectory that the Dawn spacecraft performed to escape from Vesta's gravitational influence.

All the computation steps of the theoretical Dawn escape trajectory can be synthesised in the following phases scheme which sums up the procedure performed until now in this section.

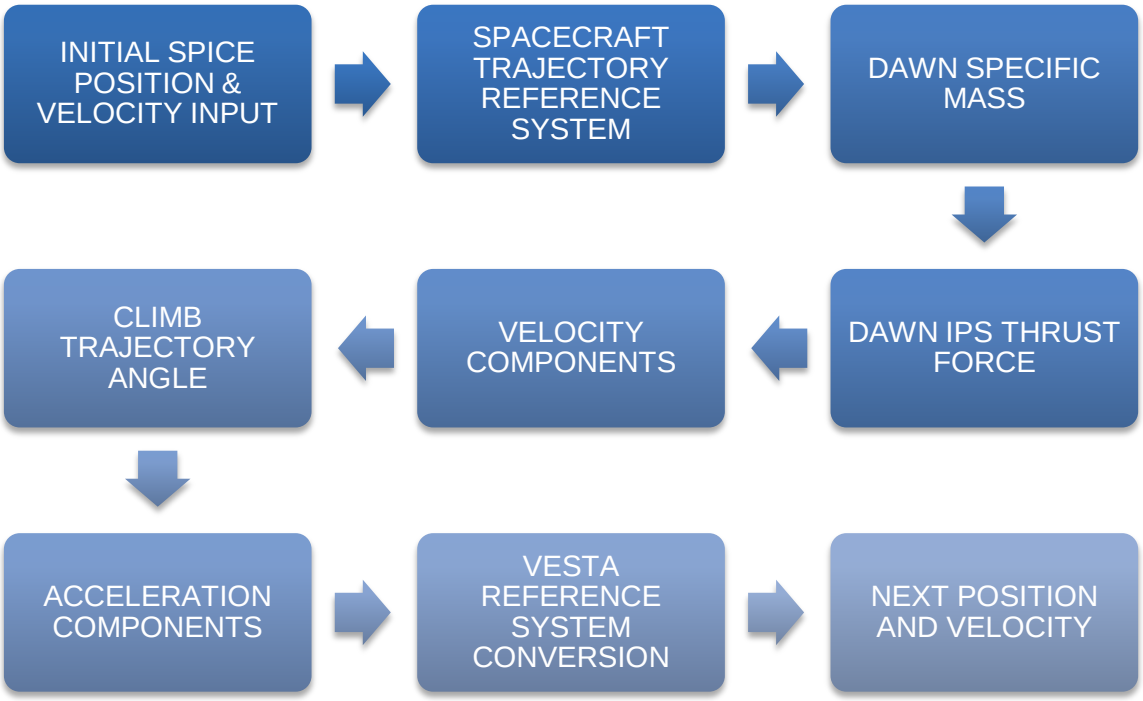


Figure 49. Required steps to compute the theoretical escape trajectory.

4.2.2 IPS thrust

As it was explained at the beginning of this chapter, the estimated Dawn's escape trajectory from Vesta was not the only parameter that would be computed although it is its main goal. Another parameter that may not seem as complex as the trajectory but it is considered a crucial datum of this whole project is the thrust force applied by the Dawn IPS thrusters. Its determination from theoretical models will prove if these models are accurate enough to consider them a correct approximation to the real value.

4.2.2.1 Computation procedure

Now, in order to compute the IPS thrust force, the conditions and considerations are different although the input data is almost the same as in the trajectory estimation.

First of all, the trajectory now is not considered a low-thrust change of orbit manoeuvre that supposes that the spacecraft moves away from a low orbit to a higher one by a constant low impulse, as it happened in the section 4.2.1. Now, the case considered is a trajectory performed by the spacecraft to escape from Vesta's gravitational influence at the final position of it. This means that these

conditions are more similar to the real ones given by SPICE, where it is supposed the spacecraft certainly escapes from Vesta at this final position, as it is has been considered until now.

Furthermore, the input data necessary to perform this thrust force computation comprises the range between Dawn and Vesta at the initial and final positions, as well as the total Dawn velocity relative to Vesta at the initial position. These 3 parameters are extracted from the results provided by SPICE. This implies a bit wider variety of input data than in the trajectory determination but not so precise, due to the only need of the modules of the data.

One of the objectives of this section is to find out the IPS thrust force so, as the data of the spacecraft mass is given in the previous section, the force can be computed employing the 2nd Newton's Law theorem. So, the resultant parameter which needs to be determined is the total acceleration generated by this IPS thrust force, so it will provide almost automatically this desired force value.

In order to compute the thrust acceleration, it is necessary to calculate the different components of this acceleration. The reference system employed to compute these components is the Spacecraft Trajectory Reference System, so these components will be the radial and tangential ones with respect to the Dawn's escape trajectory.

Firstly, the code focuses on the computation of the radial acceleration. As it was considered in the 4.2.1 section, the radial component vector points in the opposite direction of Vesta. The procedure to compute the radial thrust acceleration will be calculating the difference between the total radial spacecraft acceleration and the gravitational acceleration, just pointing opposite to the thrust one.

The theoretical model employed considers the radial and tangential spacecraft acceleration constant, so the desired resultant thrust is wanted to be constant too, then finally the gravitational acceleration will also be constant [29]. In order to implement this condition, the gravity employed for the computations will be the average one along the entire escape trajectory. Its expression is the one below.

$$\overline{\boldsymbol{a}_{\boldsymbol{grav}}} = \frac{\boldsymbol{\mu}}{|\bar{\boldsymbol{r}}|^{\boldsymbol{2}}} \qquad \textbf{(4.28)}$$

Where $\overline{a_{grav}}$ is the average gravitational acceleration and $|\bar{r}|$ is the Dawn average range to Vesta, that is, the average value from the initial to the final range during the escape trajectory.

After this, the radial spacecraft acceleration has to be determined. As it was just mentioned, both spacecraft trajectory acceleration components are considered constant. The procedure performed to determine the expression of the radial spacecraft acceleration is exposed below.

At first, the problem is considered as the spacecraft is in a circular orbit at initial radius r_i and begins performing a low-thrust manoeuvre until it reaches the escape velocity [29]. The equations of motion in polar coordinates are:

$$\frac{d^2r}{dt^2} - r \cdot \left(\frac{d\theta}{dt}\right)^2 + \frac{\mu}{r^2} = a_r \tag{4.29}$$

$$\frac{d}{dt}\left(r^2 \cdot \frac{d\theta}{dt}\right) = 0 \tag{4.30}$$

From (4.30) it can be extracted that

$$r^2 \cdot \frac{d\theta}{dt} = const = r_i^2 \cdot \frac{d\theta}{dt} = \sqrt{\frac{\mu}{r_i}} \tag{4.31}$$

Also, from (4.29),

$$\frac{d^2r}{dt^2} = \frac{1}{2} \cdot \frac{d}{dr} \cdot \left(\frac{dr}{dt}\right)^2 \tag{4.32}$$

Then,

$$\left(\frac{dr}{dt}\right)^2 = (r - r_i) \cdot \left[2 \cdot a_r - \frac{\mu}{r_i \cdot r^2} \cdot (r - r_i)\right] = v_r^2 \tag{4.33}$$

Taking into account that

$$v_{esc} = \sqrt{\frac{2\mu}{r}} \tag{4.34}$$

It all can be joined as

$$\frac{v_{esc}}{2} - \frac{\mu}{r} = \frac{1}{2} \cdot \left[\left(\frac{dr}{dt}\right)^2 + \left(r \cdot \frac{d\theta}{dt}\right)^2\right] - \frac{\mu}{r} = 0 \tag{4.35}$$

Finally, from the previous expression, it is found that the radial component of the acceleration follows the expression:

$$\boldsymbol{a_r = \frac{\mu}{2 \cdot r_i \cdot (r_{esc} - r_i)}} \tag{4.36}$$

Where a_r is the radial spacecraft acceleration and r_{esc} is the Dawn's range when it escapes from Vesta's gravitational influence, which is the same as the range at the final position of the real escape trajectory.

Then, having the radial component of the Dawn's acceleration, the tangential component must also be computed. As the only acceleration in the tangential axis is the one given by the spacecraft itself and not an external one, it is only required the calculation of is acceleration.

As it was just previously mentioned, the tangential acceleration component is also considered as a constant value, so the procedure performed to determine the expression of the tangential spacecraft acceleration is exposed below.

First of all, the considered case is again a circular orbit of initial radius r_i and the tangential acceleration applied to the spacecraft is constant [29]. The equation of motion taking into account all axes of the Spacecraft Trajectory Reference System is the following one.

$$\frac{dv}{dt}\vec{t} + \frac{v^2}{\rho}\vec{b} = a_t\vec{t} - \frac{\mu}{r^2}\vec{h} \tag{4.37}$$

The parameter ρ represents the curvature radius. Focusing on the tangential and elevation axes, there are two equations.

$$\frac{dv}{dt} = a_t - \frac{\mu}{r^2}\cdot\cos\varphi \tag{4.38}$$

$$\frac{v^2}{\rho} = \frac{\mu}{r^2}\sin\varphi \tag{4.39}$$

The parameter φ represents the flight-path angle, that is, the angle between the vectors r and v. Then it is known that the arc length of the orbit, denoted by s, the parameter that represents the distance covered along the orbit, verifies that

$$(ds)^2 = (dr)^2 + (rd\theta)^2 \tag{4.40}$$

$$r\cdot\frac{d\theta}{ds} = \sin\varphi \qquad \frac{dr}{ds} = \cos\varphi \tag{4.41}$$

From this, the equations of motion components become

$$v\cdot\frac{dv}{ds} = a_t - \frac{\mu}{r^2}\cdot\frac{dr}{ds} \tag{4.42}$$

$$\frac{v^2}{\rho} = \frac{\mu}{r}\cdot\frac{d\theta}{ds} \tag{4.43}$$

From the curvature it can be demonstrated that

$$\frac{1}{\rho} = \frac{1}{r}\cdot\left[1 - \left(\frac{dr}{ds}\right)^2 - r\cdot\frac{d^2r}{ds^2}\right]\left[1 - \left(\frac{dr}{ds}\right)^2\right]^{-\frac{1}{2}} \tag{4.44}$$

Finally, the equations of motion are

$$\frac{1}{2}\cdot\frac{dv^2}{ds} + \frac{\mu}{r^2}\cdot\frac{dr}{ds} = a_t \tag{4.45}$$

$$r\cdot v^2\cdot\frac{d^2r}{ds^2} + \left(v^2 + \frac{\mu}{r}\right)\left[\left(\frac{dr}{ds}\right)^2 - 1\right] = 0 \tag{4.46}$$

As a_t is constant, the equation (4.45) can be integrated as

$$v^2 = 2 \cdot s \cdot a_t + \mu\left(\frac{2}{r} - \frac{1}{r_i}\right) \tag{4.47}$$

Moreover, as a_t is small enough to consider $\frac{d^2r}{ds^2}$ almost zero, from equation (4.46) it can be extracted that $v^2 = \frac{\mu}{r}$, so the circularity of the orbit can be proved. Then, introducing this conclusion to equation (4.47), it is shown that

$$r = \frac{r_i}{1 - \frac{2 \cdot s \cdot a_t}{v_i^2}} \tag{4.48}$$

Now, from equation (4.34), which provides the escape velocity, equation (4.46) can be simplified to

$$2 \cdot r \cdot \frac{d^2r}{ds^2} = 1 - \left(\frac{dr}{ds}\right)^2 \tag{4.49}$$

Then, using equation (4.48),

$$s_{esc} = \frac{v_i^2}{2 \cdot a_t} \cdot \left[1 - \frac{1}{v_i} \cdot (20 \cdot a_t^2 \cdot r_i^2)^{\frac{1}{4}}\right] \tag{4.50}$$

Finally, it is found that the tangential acceleration component follows the expression:

$$\boldsymbol{a_t = \frac{r_i \cdot v_i^2}{\sqrt{20} \cdot r_{esc}^2}} \tag{4.51}$$

Now, after these computation justifications and determination of the expressions of the radial and tangential Dawn acceleration, called as *a_rad* and *a_tan* in the code respectively, it is possible to calculate the total spacecraft acceleration provided by the IPS thrusters.

$$\boldsymbol{a_T = \sqrt{\left(a_r - \overline{a_{grav}}\right)^2 + a_t^2}} \tag{4.52}$$

Finally, from this total acceleration and the Dawn's mass provided the previous section, the thrust force exerted by the IPS thrusters is

$$\boldsymbol{F_T = a_T \cdot m_{s/c}} \tag{4.53}$$

These computation steps of the thrust force can be synthesised in the following phases scheme which sums up the procedure performed until now in this section.

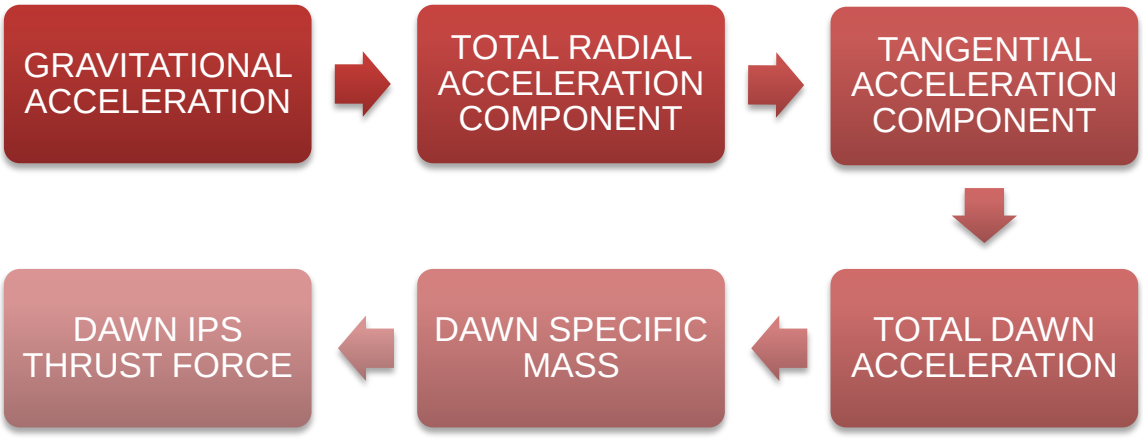


Figure 50. Required steps to compute the theoretical Dawn IPS thrust force.

4.2.2.2 Results

From this final expression it is possible to extract an exact value of the estimated Dawn IPS applied thrust. It must be remembered that from section 4.1.4 it was possible to determine the exact value of the real Dawn's thrust along the escape trajectory. This real thrust force was found to be **0.0474 N**. Now, from equation (4.53), the resultant estimated thrust force is found to be **0.0470 N**.

Table 1. Thrust force from SPICE and theoretical model.

COMPUTATION METHODOLOGY	THRUST FORCE
From SPICE (real)	**0.0474 N**
From escape theoretical model	**0.0470 N**

The resultant value of this thrust force implies that the theoretical model applied during all the exposed procedure can be considered as a reliable estimation of this parameter.

To be more concrete the only difference between the real and the computed value of this thrust force is about a 4 tenths of miliNewtons (0.8%), which means a greatly substantial accuracy, even taking into account the undoubtedly low values of the thrust force given by ion propulsion compared to other more usual propulsion systems.

4.2.3 Time of flight

Apart from the IPS thrust force, the other parameter that is estimated from the Dawn escape trajectory is the time of flight spent by the spacecraft to perform this manoeuvre. As it happened in the IPS thrust force computation, the resultant value will be extracted by a theoretical model.

In the case of the time of flight, 3 different theoretical methodologies will be employed, so 3 different values of this parameter will be calculated. Then, this 3 values will be compared each other and with the real one to determine which of them is closer to the real trajectory and even if any of them is feasible to be considered as an accurate approximation to the real case.

These 3 cases are the following ones:

- From theoretical escape trajectory, as in the previous thrust force section.
- From theoretical low-thrust change of orbit manoeuvre employed in the estimated escape trajectory.
- From theoretical low-thrust change of orbit manoeuvre considering real spacecraft velocity.

4.2.3.1 From escape trajectory

In order to compute the estimated time of flight from the theoretical escape trajectory model, it is needed to come back to the expressions employed to compute the tangential component of the spacecraft's acceleration [29].

Firstly, from equation (4.47) and (4.48), it can be deduced that

$$v^2 = {v_i}^2 - 2 \cdot s \cdot a_t \tag{4.54}$$

And then, taking into account that $v = {ds}/{dt}$, the time demanded by the spacecraft to achieve escape velocity is

$$\boldsymbol{TOF_{et}} = \frac{\boldsymbol{v_i}}{\boldsymbol{a_t}} \cdot \left[\mathbf{1} - \left(\frac{\mathbf{20} \cdot \boldsymbol{a_t}^2 \cdot \boldsymbol{r_i}^2}{\boldsymbol{v_i}^4} \right)^{\frac{\mathbf{1}}{\mathbf{8}}} \right] \tag{4.55}$$

where TOF_{et} is the estimated time of flight from the theoretical escape trajectory model, which corresponds to the *TOF_et* parameter in the code.

Finally, using this expression with the real values of the initial position and velocity, and with the estimated value of the tangential component of the Dawn acceleration, the computed time of flight value is **10.44 days**.

4.2.3.2 From low-thrust change of orbit manoeuvre and theoretical velocities

The first case of the computation of the estimated time of flight from the theoretical low-thrust change of orbit manoeuvre will take into account the conditions provided in the estimated trajectory computation section.

As it established a change of orbit but not an escape manoeuvre, the initial and final spacecraft's velocities correspond to the merely orbital, or circular, velocities at the initial and final positions. Then, this manoeuvre considered that the time of flight spent is directly proportional to the change of velocity achieved through it. Besides, it is considered that this change of velocities is only provoked by the acceleration provided by the Dawn IPS thrust force.

This implies that the time of flight follows the expression below [28].

$$\boldsymbol{TOF_{co-theo} = \frac{\Delta v_{orb}}{a_T}} \quad \textbf{(4.56)}$$

where $TOF_{co\text{-}theo}$ is he estimated time of flight from the low-thrust change of orbit manoeuvre with theoretical velocities, which corresponds to the *TOF_co_theo* parameter in the code, and Δv_{orb} is the change of the orbital velocities, which are extracted by the formula (4.15).

Finally, employing this expression with the values of the orbital velocity at the initial and final position of the estimated trajectory and with the real thrust force exerted by the Dawn IPS, the computed time of flight value is **9.47 days.**

4.2.3.3 From low-thrust change of orbit manoeuvre and real velocities

The second case of the computation of the estimated time of flight from the theoretical low-thrust change of orbit manoeuvre will also take into account the conditions provided in the estimated trajectory computation section. Nevertheless, the only change will be the values of the initial and final velocities, which will correspond to the real ones provided by SPICE.

Again, the expression employed to determine the time of flight will be very similar to the formula (4.56) but with these little changes [28].

$$\boldsymbol{TOF_{co-real} = \frac{\Delta v_{tot}}{a_T}} \quad \textbf{(4.57)}$$

where $TOF_{co\text{-}real}$ is he estimated time of flight from the low-thrust change of orbit manoeuvre with real SPICE velocities, which corresponds to the *TOF_co_real* parameter in the code, and Δv_{tot} is the change of the total real velocities, which are extracted from SPICE.

Finally, employing this expression with the values of the real spacecraft's velocity at the initial and final position of the real trajectory and with the real

thrust force exerted by the Dawn IPS, the computed time of flight value is **8.22 days.**

4.2.3.4 Results analysis

Now, from the calculations employing the different methods, there are 4 values of the time of flight spent by Dawn during its escape trajectory, which are exposed in the table below.

Table 2. Time of flight values extracted from different sources.

SOURCE EMPLOYED	TIME OF FLIGHT
Real (SPICE)	**15 days**
Escape trajectory model	**10.44 days**
Low-thrust change of orbit manoeuvre + orbital velocities	**9.47 days**
Low-thrust change of orbit manoeuvre + real velocities	**8.22 days**

The table 1.2 shows that any method achieves a very accurate resultant time of flight despite not being inaccurate either. The closest one is the theoretical escape trajectory one, which differs in almost a 33% of the real value provided by SPICE.

This proves that, although the resultant thrust force extracted from this theoretical model has a great similarity with the real Dawn IPS thrust force, this model is not so able to estimate the time of flight with the same accuracy. Although the real time of flight value has an error margin of some hours, this demonstrates the irregularity of the application and performance of the Dawn IPS.

Moreover, this table also makes more noticeable this irregularity when focusing on the difference between using orbital (theoretical) and real velocities in the low-thrust change of orbit manoeuvre model. As both initial velocities are practically the same, the great difference falls back onto the final velocities, where it is higher in the real velocities case, even seeing that the Dawn's range at the real trajectory is also higher, which would theoretically mean that the velocity should be lower.

Finally, the final approach of the time of flight different computations is displayed in the scheme below, where the accuracy of every method employed respect with the real SPICE value is represented.

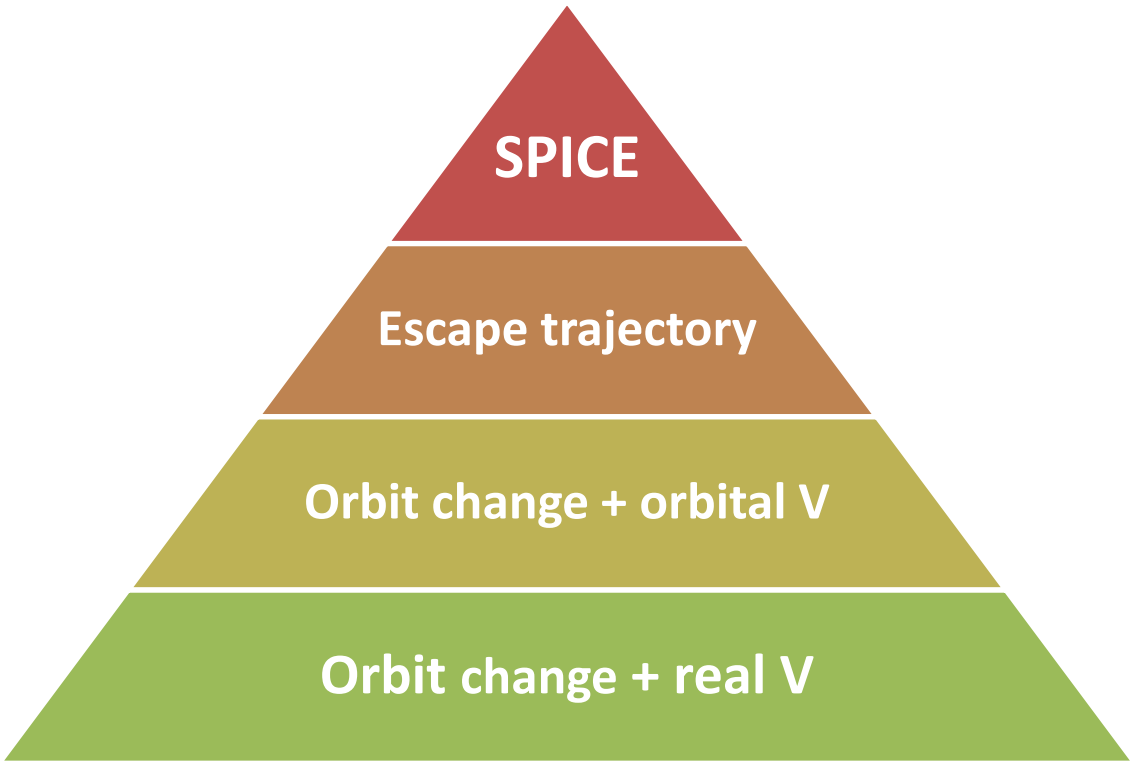


Figure 51. Level of accuracy of the time of flight calculation methods.

4.2.4 Solar arrays

As it has been mentioned, the Dawn IPS works thanks to the electric energy supplied by the solar arrays allocated in the spacecraft. Furthermore, the throttle level of the IPS is directly determined by the power provided by these solar arrays, being then controlled and distributed by the PPUs. In addition to this, the IPS is system that clearly consumes the largest fraction of the energy generated by the solar arrays, representing more than a half of its total value when the spacecraft is propelled [3].

All these reasons prove that the Electric Power System (EPS), which mostly consists in the two solar arrays of Dawn spacecraft, and the IPS are closely related. Indeed, the power supplied to the IPS in section 4.1.4 helped to calculate the thrust force considering that it was constant all along the escape trajectory. Nevertheless, the data was supplied directly from a JPL article, so in this section the power supplied to the IPS is computed employing SPICE.

Until now, SPICE has demonstrated that it is a reliable library and tool to compute parameters related to the spacecraft trajectory such as the position, velocity, or even the thrust force and the time of flight in the last sections. Now, the computations are focused on a different subsystem, the solar arrays of the Dawn EPS, specifically during the Vesta escape trajectory in order to compare them to the values of figure 34 and relate them to the thruster performance table of figure 35.

It must be mentioned that the author of this thesis expresses thanks to Roger Sala, a UPC student, for providing a truly helpful code, on which the following computing procedure is based.

4.2.4.1 Computation procedure

First of all, two new kernels are introduced in the *METAKR* matrix. One is about the frame information about the Dawn spacecraft and its solar arrays, which is called *dawn_v15.tf*. The second kernel file is really similar to the dawn trajectory's ones, but now it provides data about the solar arrays' orientation. Their nomenclature is almost same except the *rec* characters, which are replaced by the *sa*, relating it to the solar arrays. Because of this, the kernel files corresponding to the escape trajectory dates are introduced in the matrix.

After this change, the procedure is similar to the basic structure of the previous codes: the *time_computation* function is executed with the escape trajectory dates, and then the *cspice_spkezr* is computed introducing the Sun as the observer body in the last input.

Then, it begins the new calculations, which are all introduced in a loop of a *STEP* length, which means that all the computations are carried out at each *times_sec* vector time. It has to be mentioned that the two solar arrays are just placed in the +Y and –Y axes of the spacecraft, so they will be called as their "**+Y**" and "**-Y**" solar arrays respectively.

In this loop, the first computed parameters are the position rotation matrix of the frame of each solar array. For this, the SPICE function ***cspice_pxform*** is executed twice, one for each solar array. These solar arrays' position rotation matrices are determined from the ones of the frame of the spacecraft. Therefore, the input parameters are the known spacecraft reference frame, typed as '*DAWN_SPACECRAFT*', the desired solar arrays' reference frame, typed as *'DAWN_SA+Y'* and *'DAWN_SA-Y'*, and the specific epoch, where the corresponding value of the *times_sec* vector is introduced. The output matrices are *Matsa1* and *Matsa2*, referring the two solar arrays.

Then, as the wanted solar array vector is the normal one, a first vertical vector called ***boresightsa*** is established. Its values are zero for x and y axes, and 1 for the z axis. Then, the normal vectors of the solar arrays are calculated by the product of the rotation matrix and the reference normal vector. The resultant parameters are called *boresitesa1* and *boresitesa2* respectively.

Now, from the state vector previously computed, the coordinates of the position of the spacecraft are extracted and joined to create the position vector from the Sun to Dawn. This vector is called ***vectsundawn***.

Then, having the normal vector of each solar array and the position vector with respect to the Sun, it is possible to determine the angle between both vectors, obtaining the orientation of the solar arrays with respect to the Sun. These

orientation angles are called as *anglesa1* and *anglesa2* respectively. The expression employed to compute this angle is the one below.

$$\delta = \cos^{-1} \frac{\vec{n} \cdot \overrightarrow{S-D}}{\left|\overrightarrow{S-D}\right|} \quad \textbf{(4.58)}$$

where δ is the orientation angle of the solar arrays, $\vec{n}$ is the solar array normal vector and $\overrightarrow{S-D}$ is the vector from the Sun to Dawn. If this δ angle is 0º, the solar arrays are completely pointing towards the Sun. On the other hand, if it is 90º or more, the sunlight is not caught by the solar arrays.

Knowing this key parameter, it is time to start computing the solar energy variables. Firstly, the solar irradiance at Dawn is determined by knowing the range with respect to the Sun. Furthermore, having the solar irradiance on the Earth atmosphere's top, also known as solar constant, the inverse-square law is applied.

$$SI_D = SI_E \cdot \frac{r_E^2}{r_D^2} \quad \textbf{(4.59)}$$

where SI_D and SI_E are the solar irradiance at Dawn and Earth respectively in W/m^2, and r_D and r_E are the ranges from Dawn and Earth respectively to the Sun in Astronomical Units.

Now, this is the solar irradiance that reaches the spacecraft's solar arrays. However, as they have an orientation angle, it has to be taken into account. Knowing also the solar array's surface [30], the real input solar irradiance follows the expression below:

$$SI_{SA} = SI_D \cdot (S_{SA} \cdot \delta_{+Y} + S_{SA} \cdot \delta_{-Y}) \quad \textbf{(4.60)}$$

where SI_{SA} is the solar irradiance obtained by the solar arrays in W and S_{SA} is the solar array surface in m^2.

Finally, the power supplied to the Dawn systems is determined by the solar array's efficiency [3]. This power follows the expression below.

$$P_D = SI_{SA} \cdot e_{SA} \quad \textbf{(4.61)}$$

where P_D is the real power supplied to the different systems of Dawn spacecraft and e_{SA} is the efficiency of the solar arrays.

This Dawn input power does not represent the real IPS input power, but it is known that the non-IPS systems consume from 300 W when the spacecraft is on cruise to 750 W during asteroids observation and data downlink [3]. The escape trajectory would be placed near cruise operation due to the fact that the payload instruments are not working as in the scientific orbit. Nevertheless, considering this Vesta escape trajectory a delicate manoeuvre that requires to

be regularly monitored, a continuous data downlink is required, so this extra power will be greater than 300 W.

Although it is not possible to extract an exact value of the IPS input power due to this fact, the possible non-IPS consumed power will be taken into account at the Dawn input power representation.

All the computation steps of the Dawn IPS input power can be synthesised in the following phases' scheme which sums up the procedure performed until now in this section.

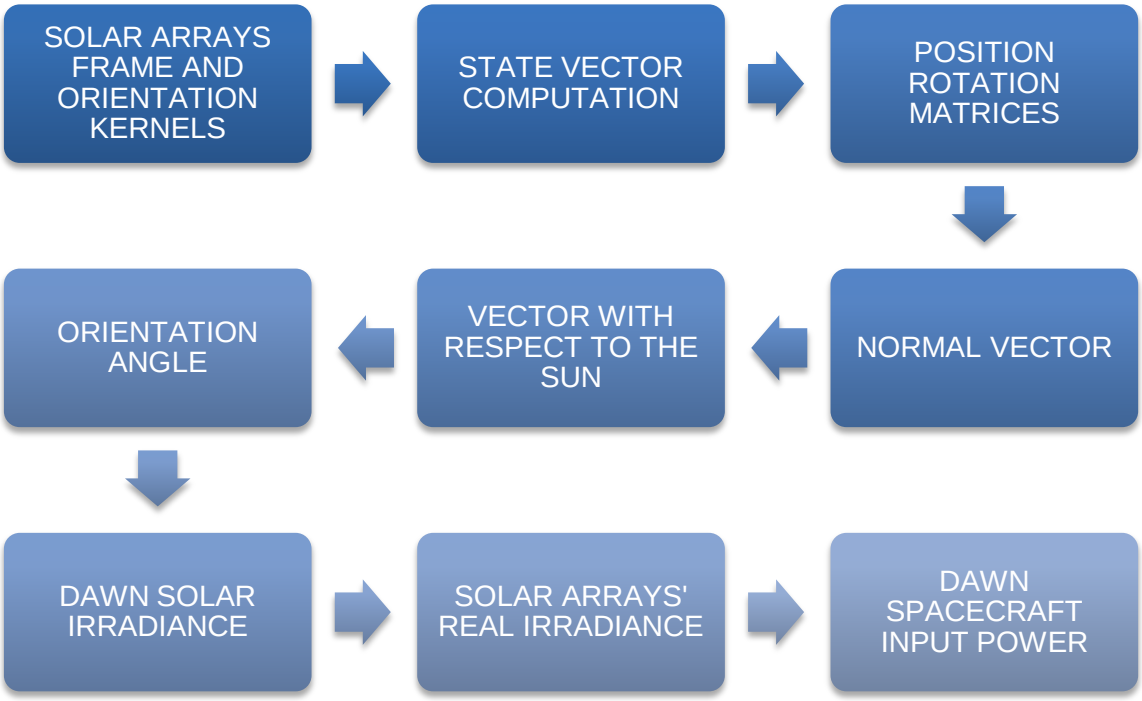


Figure 52. Dawn IPS input power computation procedure.

4.2.4.2 Plots and results

After this computations procedure, some results are extracted and analysed to understand the operation of the solar arrays in relation to the Dawn IPS. Although all the previous variables are computed in the code, the resultant key parameters that provide more useful and practical data to the objective of this section are represented in a plot. These two parameters are the **orientation angle** of the solar arrays and the **input power** to the Dawn systems.

From this and applying the previous expressions, the resultant orientation angle during the Vesta escape trajectory is represented below.

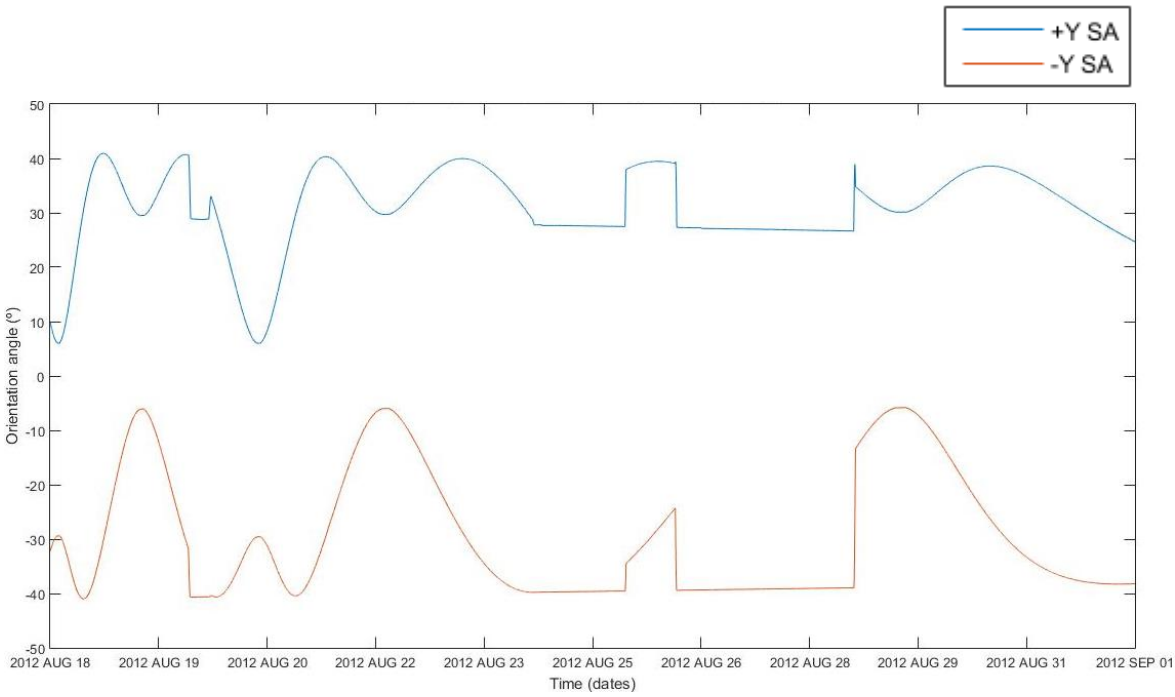


Figure 53. Dawn solar arrays orientation with respect to the Sun.

This figure 53 shows the orientation angle of the solar arrays +Y and –Y during the escape trajectory. It must be reminded that 0º represent a full facing towards the sunlight. It is proved that they work differently from the other but always taking into account the symmetry regarding the solar irradiance. Besides, focusing on the peaks, it can be seen that despite the different values, they work coordinated between them, placing the closest values to 0º at the same times.

Although it will be shown with a greater accuracy in next graph, this orientation variation may establish that, taking into account that the IPS throttle level depends on the input power, the thrust force exerted during the entire trajectory is not constant at all. Moreover, considering the peaks of the orientation angles, the phases when the solar arrays pointed more towards the sunlight are the initial stage until almost half of the trajectory and then the last stage of this escape.

Having this hypothesis, if the plots 36, 37 and 38 of the real SPICE escape trajectory are studied again, it is noticeable that this two more oriented phases coincide with the continuous climb that Dawn performs during the first half of the trajectory and the final impulse applied to end the escape and begin the cruise trajectory to Ceres.

However, this hypothesis will be confirmed or not thanks to the Dawn systems' input power represented below. It must be reminded that this power takes into account the non-IPS systems of the spacecraft but, recovering the deduction of the possible estimated value of this non-IPS power, the IPS input power can be approximated.

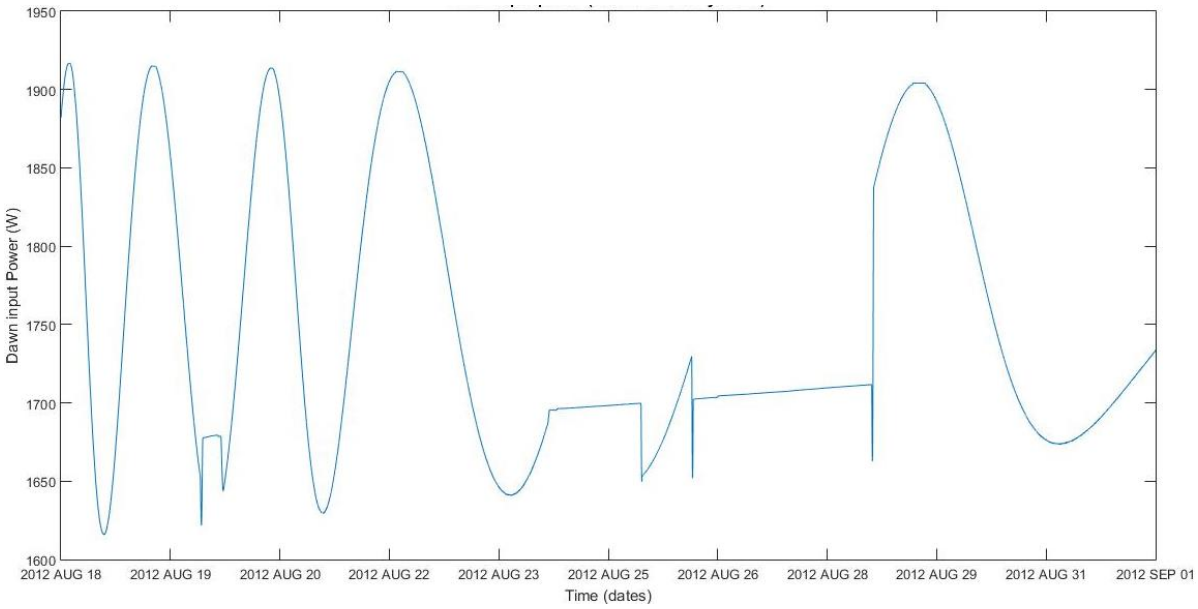


Figure 54. Dawn input power (IPS + non-IPS systems).

This figure 54 is clearly the combination of both solar arrays' orientation, whose different minimum and maximum peaks summed result the same input power, especially in the first half of the escape trajectory.

The trends of this graphs certainly confirms the hypothesis that the exerted IPS thrust force is not constant during the escape trajectory and its maximum peaks coincide with the first continuous climb of the spacecraft and the last escape impulse. Furthermore, an abrupt and clearly intentional "cut" is provoked between these two phases, causing the Dawn approach shown in figure 37 in order to obtain the required speed to carry out the last impulse.

Regarding the power values, although the maximum and minimum peaks may seem very far away one next to the other at first sight, the maximum variance is only about 300 W. Considering that the non-IPS consumed power during this escape trajectory is around 400 W [3], the resultant IPS input power range is from 1500 W to 1200 W.

These values translated into throttle levels, imply that the employed levels are from 5 (at the minimum power) to 7 (at the maximum power) [26]. This, converted into thrust represents that the exerted force ranges from 0.0422 N to 0.052 N. In the SPICE escape trajectory section, it was established that the thrust force was constant in throttle level 6 [25], so comparing it to these new results, it can be considered that it was a good approximation.

In addition to this, the theoretical computed thrust force value was also really similar to the estimated one, so the reliability of the theoretical model employed is again confirmed. These two demonstrations also prove the absolute accuracy of SPICE data and its tools operation.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

The main high level objective of this thesis established at the beginning of this document was to study the performance of the Dawn Ion Propulsion System. To accomplish this, the NASA's SPICE library and toolkit has been employed to carry out the necessary computations related to this spacecraft IPS.

From this high level objective, other goals emerged to accomplish an exhaustive work on this IPS study. They have been the computation of parameters related to the Dawn trajectory such as the position and velocity or the thrust force using SPICE, then the implementation of theoretical models to specific cases of Dawn mission to compute the same parameters and compare between them, and finally the calculation of the IPS input power from SPICE and JPL's data to provide a more accurate analysis of the previous computations.

From these procedures, the reliability of SPICE, the employed theory methodologies and the combination between them and data supplied by the Dawn mission's team is wanted to be validated.

Firstly, the computations performed using only SPICE database and its programming toolkit enabled to represent some key mission phases with a more than high accuracy, making even noticeable the irregularities of the IPS operation. They made possible a proper understanding of the application of the IPS in innovative situations such as the Vesta escape trajectory, a revolutionary manoeuvre in the space propulsion field. Here, the spacecraft's position and velocity were determined together with the thrust force applied and its distribution in the Spacecraft Trajectory Reference System. The trustworthiness reflected in the resultant values certainly confirmed the preciseness at its use in space mission analysis.

Then, a greater effort in complexity was required at the implementation of the theoretical models. From initial SPICE data, an estimation of the Vesta escape trajectory was plotted successfully, which resulted undoubtedly very similar to the one extracted only from SPICE. It represented a continuous low thrust climb manoeuvre trajectory.

Besides, the thrust force was also estimated by considering a low thrust escape trajectory model, which considered the thrust force as constant. The resultant thrust value extraordinarily close to the one extracted from the specification of the thruster and the JPL data, considering the low values of the thrust force regarding ion propulsion. Specifically, the difference between the computed and the considered as real one was only about 4 tenths of miliNewtons, which only meant a variation of only 0.8%.

In addition to the trajectory and the thrust force, the time of flight was also approximated employing different methodologies. The most similar one, the low thrust escape trajectory model, resulted in a difference of a 33% (5 days), which is much greater than the thrust one, but it can be considered acceptably

accurate taking into account the actual difficulties at estimating the real time of flight. Nevertheless, these three theoretical models applications demonstrated the possibility of connecting the data and results provided by SPICE and the implemented theory.

Finally, a deeper study of the IPS operation was carried through thanks to the computations regarding the EPS, specifically the Dawn solar arrays, due to the fact that the IPS work is directly determined by the power supplied by it. Firstly, it was possible to determine the orientation angle of the solar arrays thanks to the data provided by SPICE of the spacecraft's frame and position with respect to the Sun. Then, from this orientation angle, the solar arrays' efficiency and the Dawn's solar irradiance determined by the range to the Sun, the input power to all Dawn systems was able to be calculated. Again, this computing procedure was performed for the Vesta escape trajectory.

Knowing the power consumption of the non-IPS systems during this selected case, it was concluded that the thrust during this manoeuvre was not constant but it varied along only 3 throttle levels. The maximum variation between them was only about 0.01 N, but it is enough to not consider it as constant. It was concluded that during some periods the IPS exerted a higher thrust force due to the necessities of the manoeuvre, and in other phases the force was lower due to the intentional orientation of the solar arrays. Nevertheless, the average thrust force value coincided with the previously considered real value and the computed theoretical one, so this new solution does not discredit the previous considerations but improves them.

Thanks to these results, the SPICE database and toolkit as well as the employed theoretical expressions are again proved as reliable methods at space missions computations. Indeed, this thesis may mean a first approach to future studies regarding Dawn systems or other spacecraft, due to the fact that SPICE encompasses a huge variety of data, from this trajectory one to other related to the payload instrumentation. Moreover, a similar study can be performed focusing on other mission phases.

REFERENCES

[1] JPL, «Dawn Artist's Concept (Realistic Ceres),» 02 March 2015. [Online]. Available: https://www.jpl.nasa.gov/spaceimages/details.php?id=PIA20919. [Last access: 08 June 2019].

[2] NASA, «Dawn: About the mission,» [Online]. Available: https://www.jpl.nasa.gov/missions/dawn/. [Last access: 06 June 2019].

[3] C. T. Russell & C. A. Raymond, The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres, Springer, 2012.

[4] NASA/JPL-Caltech, «Dawn Journal: It's All About Grace Under Pressure for Dawn's Drop Into Orbit,» 31 January 2014. [Online]. Available: https://www.jpl.nasa.gov/blog/2014/1/its-all-about-grace-under-pressure-for-dawns-drop-into-orbit. [Last access: 08 June 2019].

[5] N. P. S. C. team, «Dawn Mission Timeline,» 11 December 2018. [Online]. Available: https://solarsystem.nasa.gov/missions/dawn/mission/toolkit/mission-timeline. [Last access: 08 June 2019].

[6] NASA, «New Horizons: Overview,» [Online]. Available: https://www.nasa.gov/mission_pages/newhorizons/overview/index.html. [Last access: 08 June 2019].

[7] NASA, «NASA's Dawn Mission to Asteroid Belt Comes to End,» 01 November 2018. [Online]. Available: https://www.jpl.nasa.gov/news/news.php?feature=7275. [Last access: 08 June 2019].

[8] N. P. Laboratory-Caltech, «Dawn by the Numbers,» 15 November 2018. [Online]. Available: https://solarsystem.nasa.gov/resources/2168/dawn-by-the-numbers/. [Last access: 08 June 2019].

[9] E. Lakdawalla, «What did Dawn learn at Vesta?,» 09 October 2014. [Online]. Available: http://www.planetary.org/blogs/emily-lakdawalla/2014/10091306-what-did-dawn-learn-at-vesta.html. [Last access: 08 June 2019].

[10] NASA, «Dawn mission overview,» [Online]. Available: https://solarsystem.nasa.gov/missions/dawn/overview/. [Last access: 08 June 2019].

[11] N. G. R. Center, «Overview of ion propulsion,» 22 December 2008. [Online]. Available: https://www.grc.nasa.gov/www/ion/overview/overview.htm. [Last access: 08 June 2019].

[12] NAIF, «NAIF index,» [Online]. Available: https://naif.jpl.nasa.gov/naif/index.html. [Last access: 08 June 2019].

[13] C. Acton, *NAIF AAREADME,* 2001.

[14] NAIF, *AA_README_EXPORT.*

[15] NAIF, «SPICE library index,» [Online]. Available: https://naif.jpl.nasa.gov/pub/naif/. [Last access: 08 June 2019].

[16] NAIF, *spk Dawn kernels aareadme.*

[17] NAIF, «SPICE system overview,» [Online]. Available:

http://www.iki.rssi.ru/naif/spiceovr.htm. [Last access: 08 June 2019].

[18] NAIF, «The SPICE toolkit,» [Online]. Available: https://naif.jpl.nasa.gov/naif/toolkit.html. [Last access: 08 June 2019].

[19] NAIF, *cspice_etcal html file.*

[20] NASA/JPL-Caltech, «Dawn Spacecraft Begins Approach to Dwarf Planet Ceres,» 29 December 2014. [Online]. Available: https://www.jpl.nasa.gov/news/news.php?feature=4425. [Last access: 09 June 2019].

[21] NASA/JPL, «Dawn Spacecraft Leaving Earth (Artist's Concept),» 29 May 2009. [Online]. Available: https://www.jpl.nasa.gov/spaceimages/details.php?id=PIA12032. [Last access: 09 June 2019].

[22] NASA/JPL, «NASA Spacecraft Falling For Mars,» 12 February 2009. [Online]. Available: https://www.nasa.gov/mission_pages/dawn/news/dawnf-20090212.html. [Last access: 09 June 2019].

[23] NASA/JPL-Caltech, «Dawn Orbiting Vesta,» 12 December 2011. [Online]. Available: https://www.jpl.nasa.gov/spaceimages/details.php?id=pia15174. [Last access: 09 June 2019].

[24] NASA/JPL-Caltech, «Dawn Celebrates Eight Years, Two Stellar Orbits,» 27 September 2015. [Online]. Available: https://www.jpl.nasa.gov/blog/2015/9/dawn-celebrates-eight-years-two-stellar-orbits. [Last access: 09 June 2019].

[25] C. E. Garner & M. D. Ryman, «In-Flight Operation of the Dawn Ion Propulsion System Through Survey Science Orbit at Ceres,» 2015.

[26] J. R. Brophy, M. G. Marcucci, G. B. Ganapathi, J. Gates i e. al., «Implementation of the Dawn Ion Propulsion System,» de *AIAA 2005-4071*, Tucson, Arizona, 2005.

[27] JPL, «Dawn Mission to Ceres/Vesta,» [Online]. Available: https://www.jpl.nasa.gov/infographics/infographic.view.php?id=10753. [Last accesss: 09 June 2019].

[28] O. Lizandra, *Space Propulsion: Module 1,* 2018.

[29] J. Gutierrez, *Low Thrust Trajectories,* 2018.

[30] S. B. Benson, «Solar Power for Outer Planets Study,» Greenbelt, 2007.