COMET INTERCEPTOR GIT SET UP

Prepared by	Jaime Fernández Diz
Document Type	MAN - Manual / User Guide / Handbook
Reference	
Issue/Revision	.
Date of Issue	29/07/2022
Status	N/A

APPROVAL

Title	Comet interceptor git set up		
Issue Number		Revision Number	
Author	Jaime Fernández Diz	Date	29/07/2022
Approved By		Date of Approval	

CHANGE LOG

Reason for change	Issue Nr	Revision Number	Date

CHANGE RECORD

Issue Number	Revision Number		
Reason for change	Date	Pages	Paragraph(s)

DISTRIBUTION

Name/Organisational Unit

Table of Contents

List of figures

1. INTRODUCTION

This document aims to quickly explain how to prepare and use the parametric trajectory generator of comet interceptor and its Cosmographia catalogues at user level, without entering in algorithms or mathematical considerations of the model. It also includes explanations of some specific files that are not intended to being modified during the execution of the program but might be useful to change in the future.

2. DOWNLOAD AND SET UP

In order to visualize the predefined trajectories, you will need the Comet Interceptor git repository and Spice Enhanced Cosmographia:

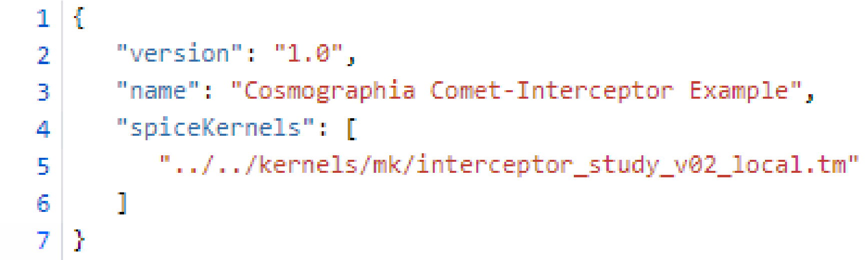
- Comet interceptor git repository: Browse SPICE_Kernels / COMET-INTERCEPTOR - SOCCI Bitbucket (esa.int)
- Spice Enhanced Cosmographia: SPICE-enhanced Cosmographic Components (nasa.gov)

If new trajectories are going to be generated, also python 3 and the mkspk tool provided by NAIF are needed:

- mkspk: Application and Utility Programs (nasa.gov). Here, select your operative system and find the mkspk program. Then download it and copy it in comet-interceptor/misc/ParametricTrajectory.

In order to be able to open the catalogue in Cosmographia, some modifications have to be done to the files:

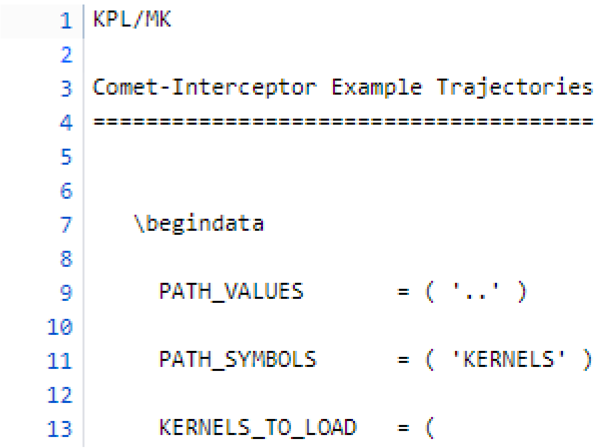
1. In the folder comet-interceptor/misc/cosmo, open spice_interceptor.json and change the “spiceKernels” variable to “../../kernels/mk/interceptor_study_v02.tm” (remove the _local part of the name). Be careful with the version number, as it could have also changed.



```
1 {
2     "version": "1.0",
3     "name": "Cosmographia Comet-Interceptor Example",
4     "spiceKernels": [
5         "../../kernels/mk/interceptor_study_v02_local.tm"
6     ]
7 }
```

Figure 1: delete the _local part of the string in line 5.

2. In the folder comet-interceptor/kernels/mk open interceptor_study_v02.tm and interceptor_study_parametric.tm and change the PATH_VALUES to the complete path to said files (ending in the kernels folder).



```
KPL/MK

Comet-Interceptor Example Trajectories
=====================================


   \begindata

     PATH_VALUES       = ( '..' )

     PATH_SYMBOLS      = ( 'KERNELS' )

     KERNELS_TO_LOAD   = (
```

Figure 2: change the .. in PATH_VALUES for the path to this file.

3. In order to load the catalogue in Cosmographia, open it and go to file > open catalog and in comet-interceptor/misc/cosmo, select load_interceptor_001.json for the predefined trajectories, or load_interceptor_parametric.json for the user defined ones. Note that by default there are no user defined trajectories in the repository, so the parametric trajectory generator has to be run at least once before loading load_interceptor_parametric.json or Cosmographia will show errors.

3. PARAMETRIC TRAJECTORY GENERATOR USAGE

To run the parametric trajectory generator, go to comet-interceptor/misc/ParametricTrajectory and run the code trajectory_generator_v02.py in any way you want (through Spyder, with the command window, with the Anaconda prompt…)

The program should generate the trajectory files and open Cosmographia with the catalogue loaded.

If an error arises during the execution of the program, the remaining steps can be completed manually to be able to visualize the trajectory anyways.

3.1. Input files are created, but not the spk files.

During the execution, the program should create three files called input_A.txt, input_B1.txt and input_B2.txt, and place them in comet-interceptor/misc/ParametricTrajectory. Those files are not in the git repository, but are overwritten each time the program is run, so make sure that they are the last version of the file, and not some old ones.

The trajectory generator then calls the program mkspk for creating the spice trajectory files (which are in comet-interceptor/kernels/spk and are called CI_SCA_parametric.bsp, CI_SCB1_parametric.bsp, CI_SCB2_parametric.bsp). Those files are also not present in the git repository and are overwritten in each execution of the program.

If the input files are created, but the spk files are not:

1. Check that mkspk is in the proper folder (misc/ParametricTrajectory) and that it is the version suitable for your operative system.
2. If the problem persists, create manually the spk files:
   - 2.1. Open a terminal in misc/ParametricTrajectory
   - 2.2. Call mkspk (with the command “mkspk -setup <file>) with each of the set up files in the folder.

Once the spk files are created, the catalogue load_interceptor_parametric.json can be opened as explained in section 2.

3.2. SPK files are created, but cosmographia is not opened.

At the end of the execution of the parametric trajectory generator, Cosmographia should be opened with the catalogue loaded. If this does not happen, but the spk files are created, it should be possible to load the catalogue manually like in the previous sections.

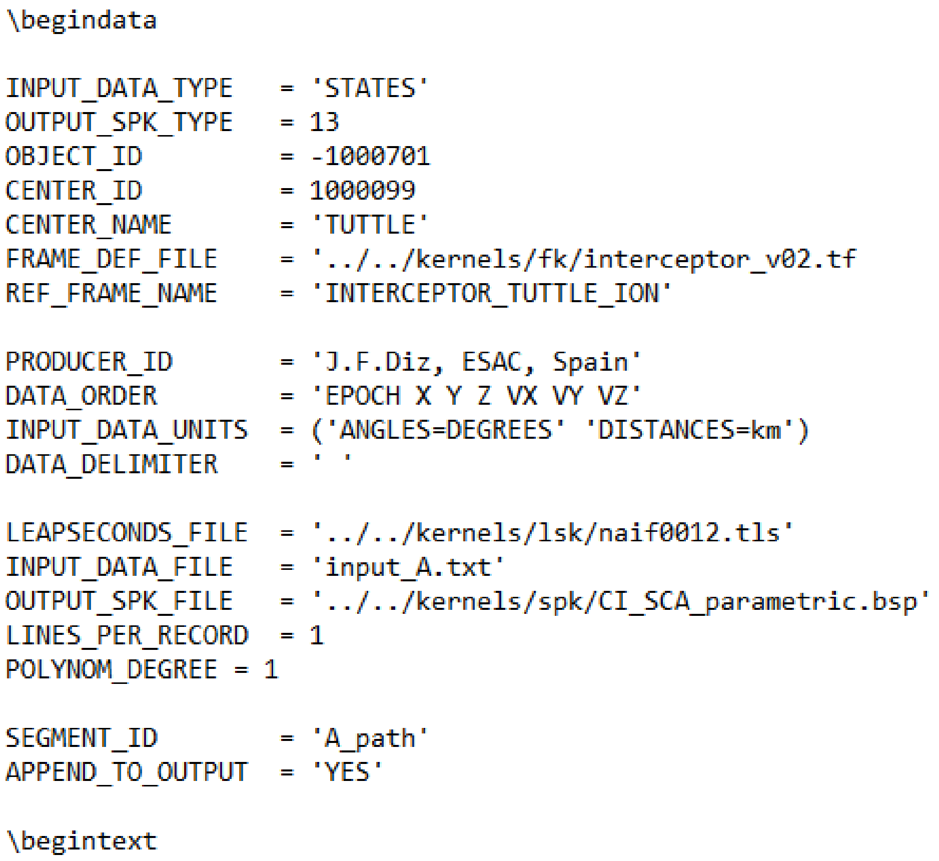
4. SET UP FILES

The set up files contained in comet-interceptor/misc/ParametricTrajectory are the files used by mkspk for creating the trajectory of the bodies in a given reference frame and with regard to a given object.

If during the course of the project, the trajectory of the spacecrafts needs to be defined in other reference frame, the variable REF_FRAME_NAME that appears in the set up files will have to be updated. If the reference frame is not one of the spice defaults, then the FRAME_DEF_FILE should be updated with the relative path to a fk file which contains the definition of the new reference frame.

The input and output files can also be defined in the setup files, as can be seen in the third block of variables. Those, however, are optional, and can be given as inputs when the mkspk program is called in the command window.

More information about the setup files is available in the mkspk user manual.



```
\begindata

INPUT_DATA_TYPE   = 'STATES'
OUTPUT_SPK_TYPE   = 13
OBJECT_ID         = -1000701
CENTER_ID         = 1000099
CENTER_NAME       = 'TUTTLE'
FRAME_DEF_FILE    = '../../kernels/fk/interceptor_v02.tf
REF_FRAME_NAME    = 'INTERCEPTOR_TUTTLE_ION'

PRODUCER_ID       = 'J.F.Diz, ESAC, Spain'
DATA_ORDER        = 'EPOCH X Y Z VX VY VZ'
INPUT_DATA_UNITS  = ('ANGLES=DEGREES' 'DISTANCES=km')
DATA_DELIMITER    = ' '

LEAPSECONDS_FILE  = '../../kernels/lsk/naif0012.tls'
INPUT_DATA_FILE   = 'input_A.txt'
OUTPUT_SPK_FILE   = '../../kernels/spk/CI_SCA_parametric.bsp'
LINES_PER_RECORD  = 1
POLYNOM_DEGREE = 1

SEGMENT_ID        = 'A_path'
APPEND_TO_OUTPUT  = 'YES'

\begintext
```

Figure 3: setup file example.