

# SPICE @ ESA

## Status and Future Work

Marc Costa  
Bjoern Grieger  
Christophe Arviset



ESA SPICE Service

PDW#4  
18th June 2019, Flagstaff

# SPICE in a nutshell

SPICE is an information system that uses **ancillary data** to provide Solar System geometry information to scientists and engineers for planetary missions in order to plan and analyze scientific observations from space-born instruments. SPICE was originally developed and maintained by the Navigation and Ancillary Information Facility (NAIF) team of the Jet Propulsion Laboratory (NASA).

"Ancillary data" are those that help scientists and engineers determine:

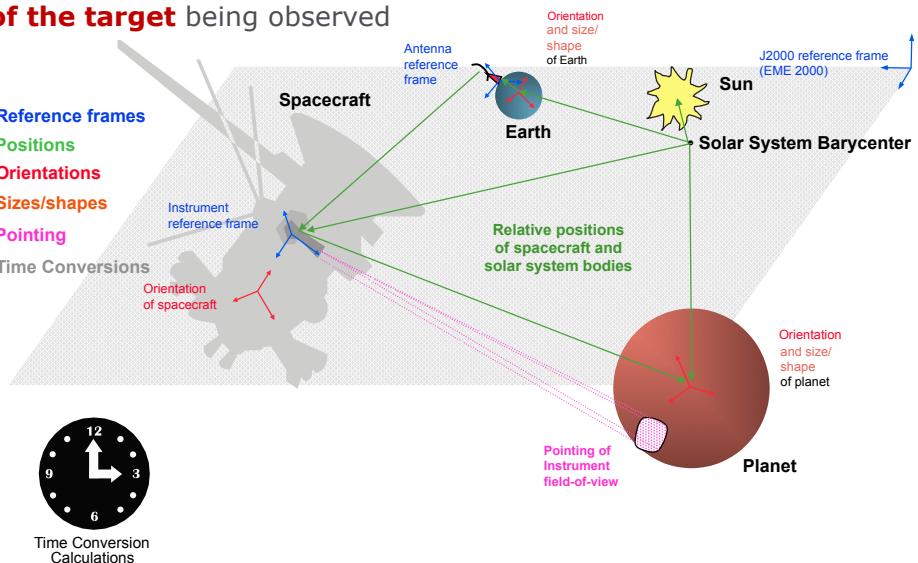
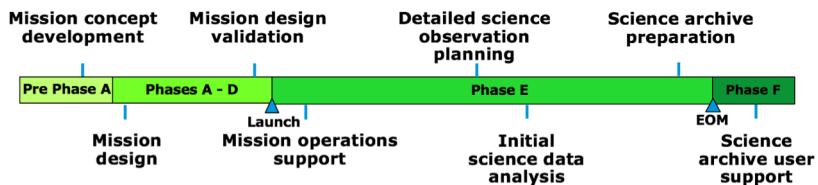
where the **spacecraft** was **located**

how the spacecraft and its instruments were **oriented** (pointed)

what was the **location, size, shape and orientation of the target** being observed

what **events were occurring** on the spacecraft

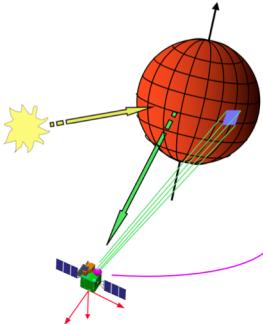
- SPICE provides users a large suite of SW used to read SPICE ancillary data files to compute observation geometry.
- The ancillary data (kernels) comes from: The S/C, MOC/SGS, S/C manufacturer and Instrument teams, Science Organizations.



# SPICE in a nutshell



Compute many kinds of observation geometry parameters at selected times

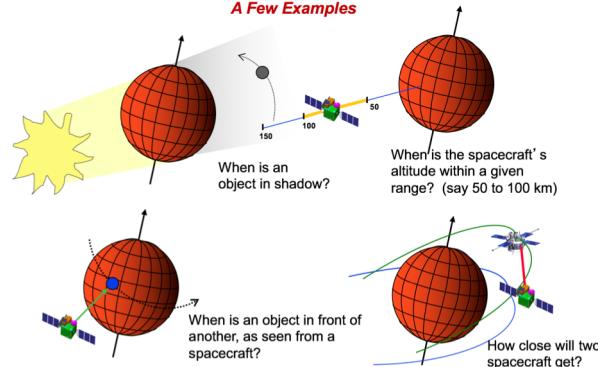


## A Few Examples

- Positions and velocities of planets, satellites, comets, asteroids and spacecraft
- Size, shape and orientation of planets, satellites, comets and asteroids
- Orientation of a spacecraft and its various moving structures
- Instrument field-of-view location on a planet's surface or atmosphere

Find times when a specified “geometric event” occurs, or when a specified “geometric condition” exists

## A Few Examples



## Time conversions

UTC to ET mapping (“generic” LSK file)

Universal Time Coordinated (UTC)

Orbiter on-board clock (SCLK)

## Time conversions

ET to orbiter on-board clock mapping (“orbiter” SCLK file)

## Position Vectors

Earth position relative to Solar System barycenter (“planet ephemeris” SPK file)

Ephemeris Time (ET)

Z<sub>E</sub> X<sub>E</sub> Y<sub>E</sub>

Orbiter on-board clock (SCLK)

Z<sub>O</sub> X<sub>O</sub> Y<sub>O</sub>

## Frame Orientations

Orbiter frame orientation relative to J2000 frame (“orbiter” CK file)

Rover position relative to the landing site (lander) (“rover” SPK file)

Z<sub>J2000</sub>

X<sub>J2000</sub> Y<sub>J2000</sub>

Z<sub>R</sub>

X<sub>R</sub> Y<sub>R</sub>

Rover frame orientation relative to local level frame (“rover” CK file)

Landing site (lander) position relative to the Mars center (“landing site” SPK file)

Z<sub>L</sub>

X<sub>L</sub> Y<sub>L</sub>

Z<sub>M</sub>

X<sub>M</sub> Y<sub>M</sub>

Local level frame orientation relative to planet body-fixed frame (“mission” FK file)

Mars position relative to the Solar System barycenter (“planet ephemeris” SPK file)

Orbiter position relative to the center of Mars (“orbiter” SPK file)

Planet body-fixed frame orientation relative to J2000 frame (“generic” PCK file)

# Using SPICE

- We want to analyze Phobos images from the HRSC instrument in MEX, more concretely images that with good resolution taken less than 1.000 km from Phobos → **WebGeocalc or GF System**
- Then we could constrain our search in the PSA UI.

 **Jet Propulsion Laboratory**  
California Institute of Technology

**WebGeocalc**  
Version 2.1.0 (4386 N66 11-FEB-2019)

**Input Values**

Calculation type	Distance Event Finder
Target	PHOBOS
Observer	MARS EXPRESS
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time range	2010-01-01 to 2010-12-01, step 6 hours
Event condition	is less than 1000
Output time unit	seconds
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

**Tabular Results**

Click a value to save it for a subsequent calculation.

Save All Intervals

	Start Time	Stop Time	Duration (secs)
1	2010-02-28 16:18:07.102645 UTC	2010-02-28 16:29:03.406319 UTC	656.30367434
2	2010-08-24 08:22:05.025171 UTC	2010-08-24 08:32:29.283596 UTC	624.25842512
3	2010-08-27 20:28:51.926715 UTC	2010-08-27 20:34:53.806208 UTC	361.87949306

**planetary science archive**  
PSA 5.4.1

Show All Hide All

Number of selected products: 0

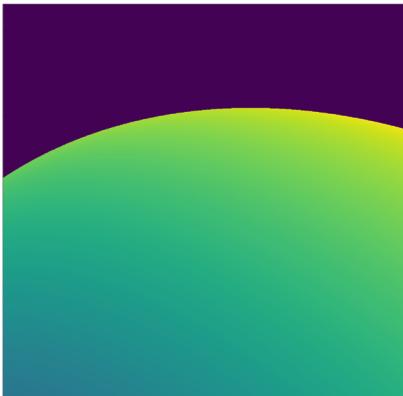
	Postcard	Product Identifier	Observation Start Time
<input type="checkbox"/>		H8512_0000_SR2.IMG	2010-08-27 20:33:05.658
<input type="checkbox"/>		H8512_0008_SR2.IMG	2010-08-27 20:32:00.355
<input type="checkbox"/>		H8512_0007_SR2.IMG	2010-08-27 20:31:58.175
<input checked="" type="checkbox"/>		H8512_0006_SR2.IMG	2010-08-27 20:31:56.243
<input type="checkbox"/>		H8512_0005_SR2.IMG	2010-08-27 20:31:54.063
<input type="checkbox"/>		H8512_0004_SR2.IMG	2010-08-27 20:31:51.883
<input type="checkbox"/>		H8512_0003_SR2.IMG	2010-08-27 20:31:49.703
<input type="checkbox"/>		H8512_0002_SR2.IMG	2010-08-27 20:31:47.523
<input type="checkbox"/>		H8512_0001_SR2.IMG	2010-08-27 20:31:45.095

# Using SPICE

```
load last meta-kernel  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```

# Using SPICE

```
load last meta-kernel  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```



# Using SPICE

load last meta-kernel  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf

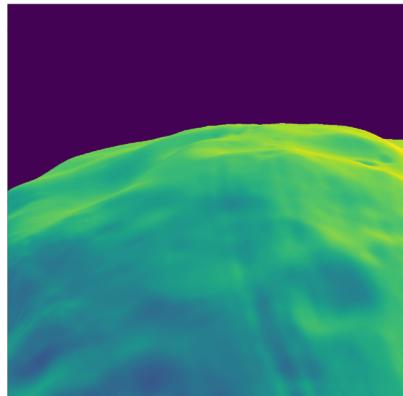
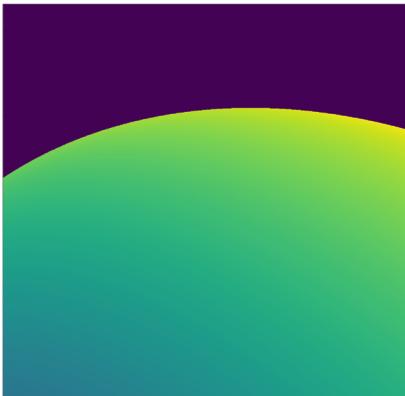
load last meta-kernel  
**load dsk**  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf



# Using SPICE

```
load last meta-kernel  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```

```
load last meta-kernel  
load dsk  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```

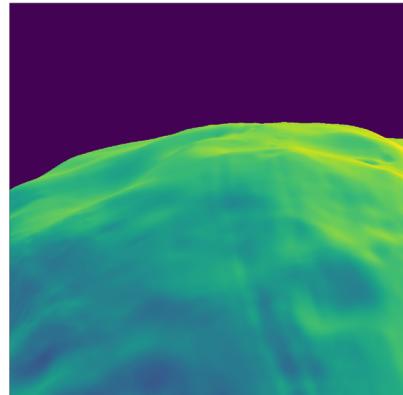
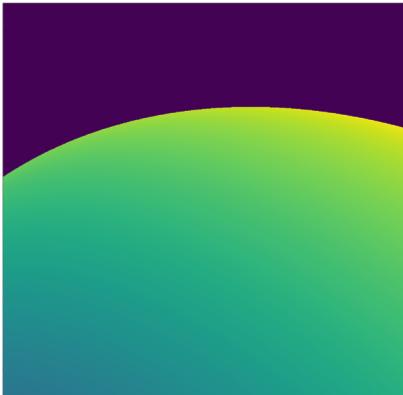


# Using SPICE

load last meta-kernel  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf

load last meta-kernel  
**load dsk**  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf

load last meta-kernel  
load dsk  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf  
adjust matrix

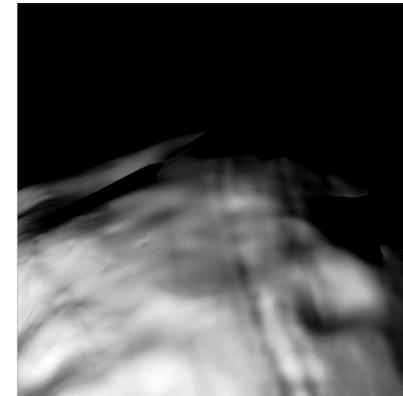
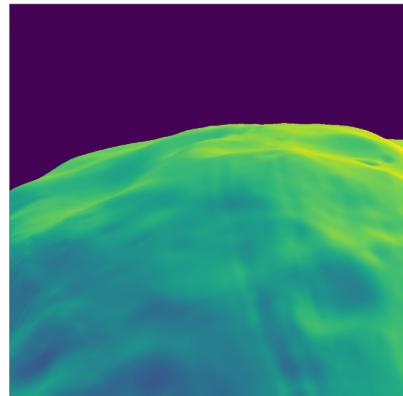
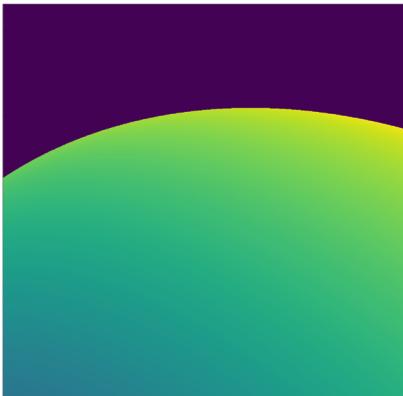


# Using SPICE

load last meta-kernel  
 loop per pixel  
 spice.getfov  
 spice.sincpt  
 spice.illumf

load last meta-kernel  
**load dsk**  
 loop per pixel  
 spice.getfov  
 spice.sincpt  
 spice.illumf

load last meta-kernel  
 load dsk  
 loop per pixel  
 spice.getfov  
 spice.sincpt  
 spice.illumf  
 adjust matrix

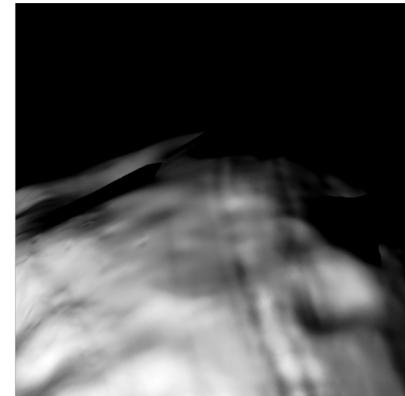
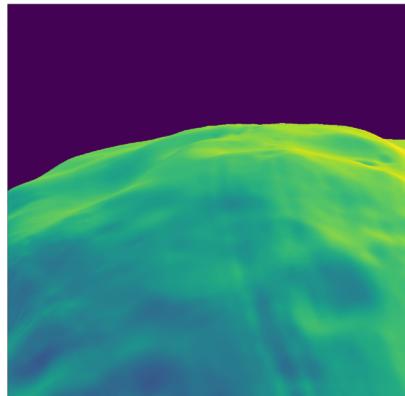
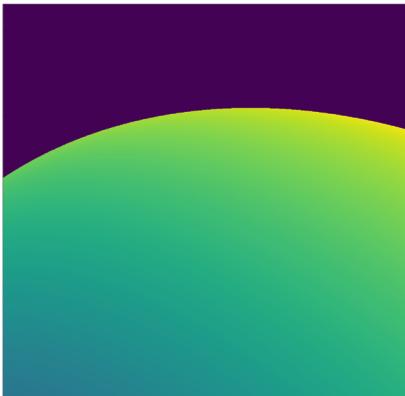


# Using SPICE

```
load last meta-kernel  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```

```
load last meta-kernel  
load dsk  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```

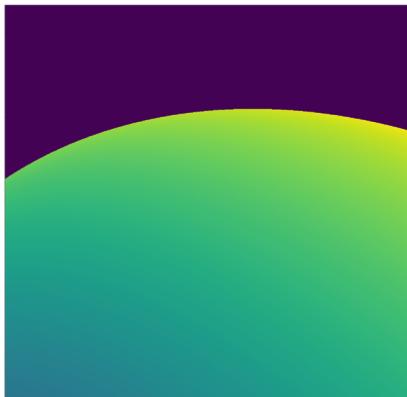
```
load last meta-kernel  
load dsk  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf  
adjust matrix
```



# Using SPICE

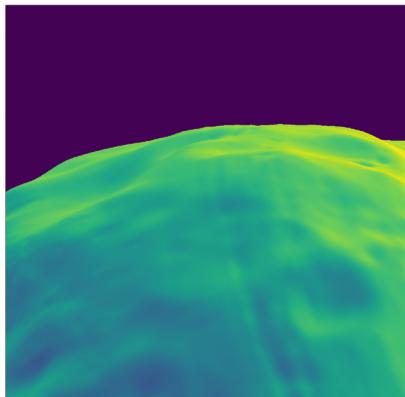
```
load last meta-kernel
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
```

Lines of code?



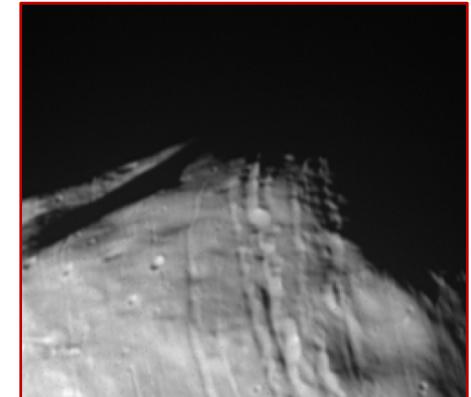
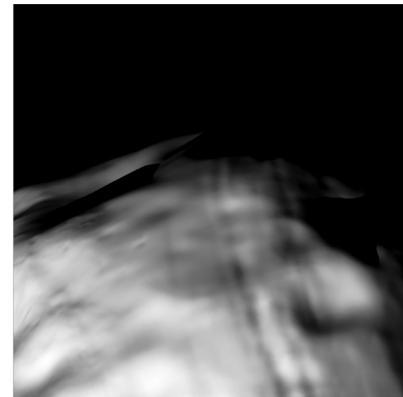
```
load last meta-kernel
load dsk
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
```

Lines of code?



```
load last meta-kernel
load dsk
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
adjust matrix
```

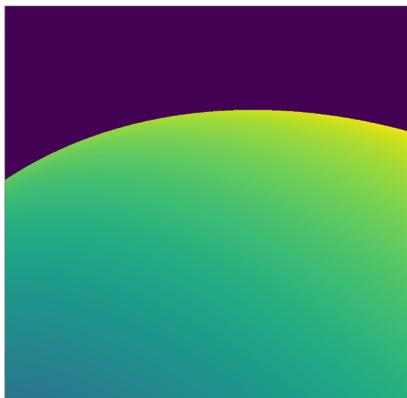
Lines of code?



# Using SPICE

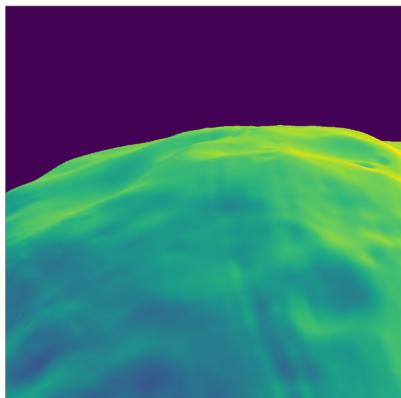
```
load last meta-kernel
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
```

Lines of code? **37**



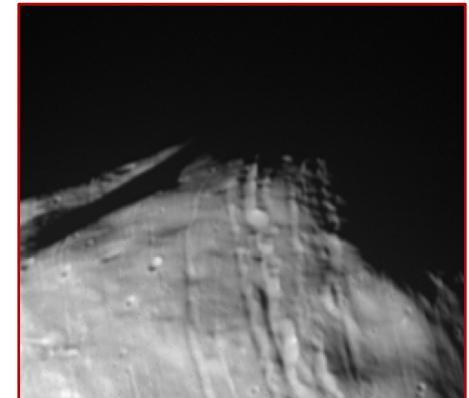
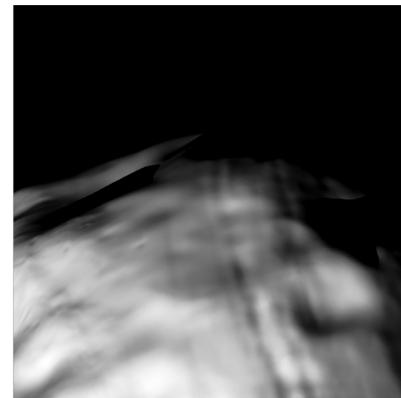
```
load last meta-kernel
load dsk
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
```

Lines of code?



```
load last meta-kernel
load dsk
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
adjust matrix
```

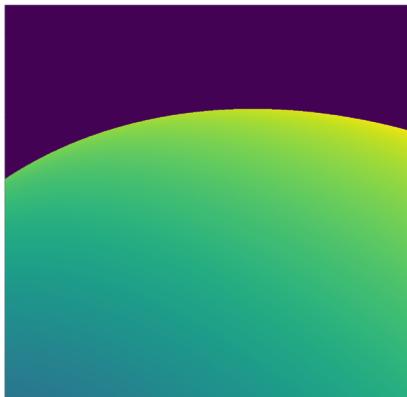
Lines of code?



# Using SPICE

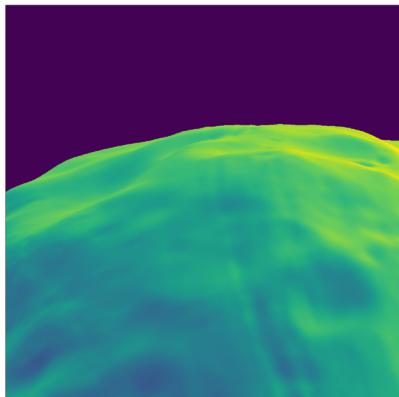
```
load last meta-kernel
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
```

Lines of code? **37**



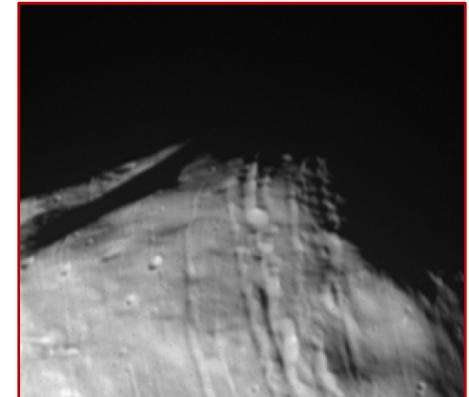
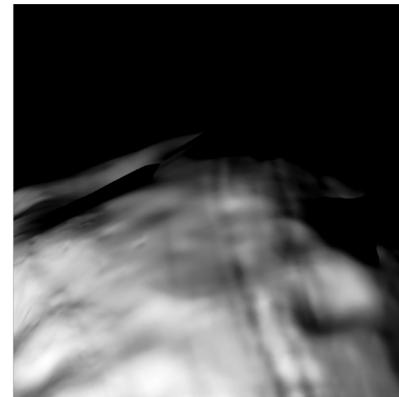
```
load last meta-kernel
load dsk
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
```

Lines of code? **38**



```
load last meta-kernel
load dsk
loop per pixel
spice.getfov
spice.sincpt
spice.illumf
adjust matrix
```

Lines of code?



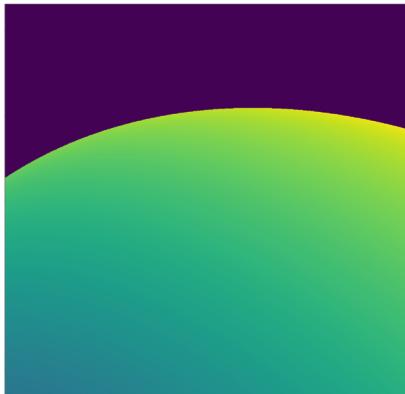
# Using SPICE

```
load last meta-kernel  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```

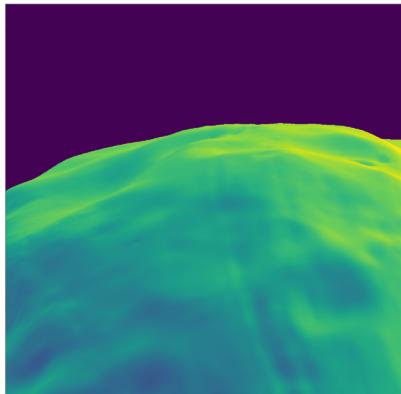
```
load last meta-kernel  
load dsk  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf
```

```
load last meta-kernel  
load dsk  
loop per pixel  
spice.getfov  
spice.sincpt  
spice.illumf  
adjust matrix
```

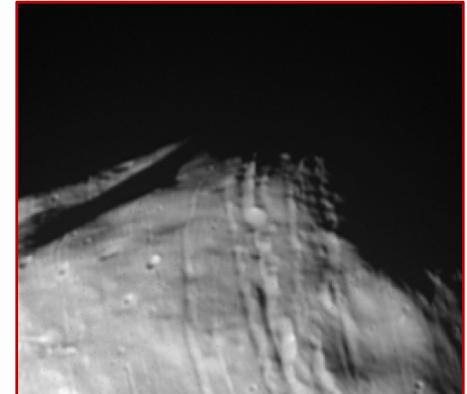
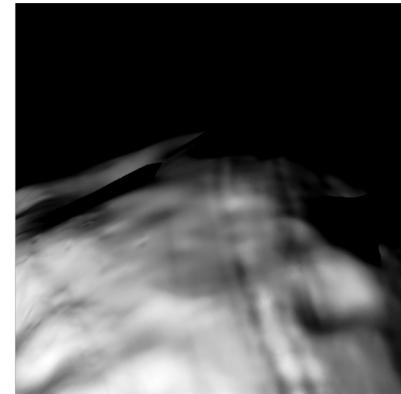
Lines of code? 37



Lines of code? 38



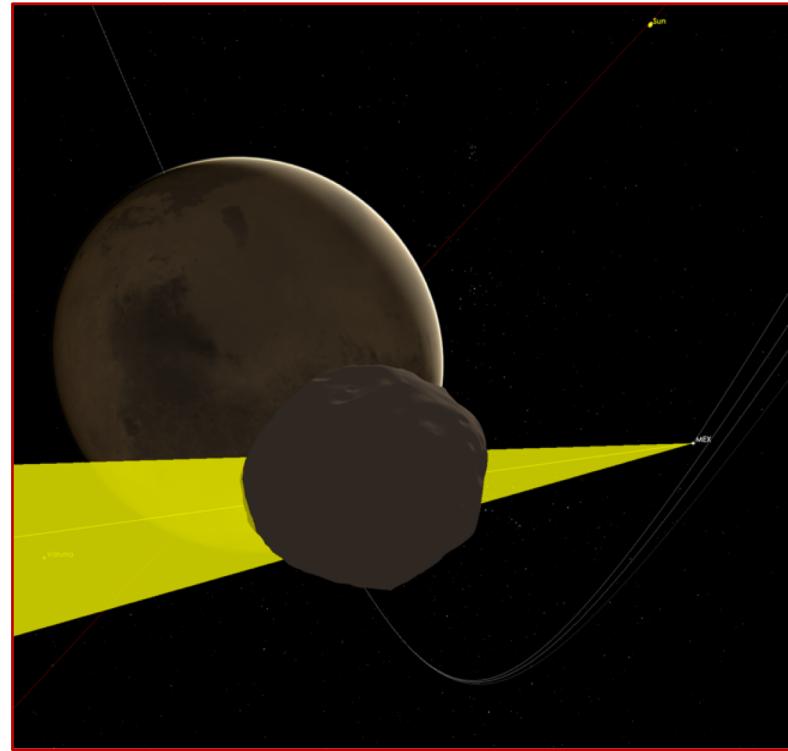
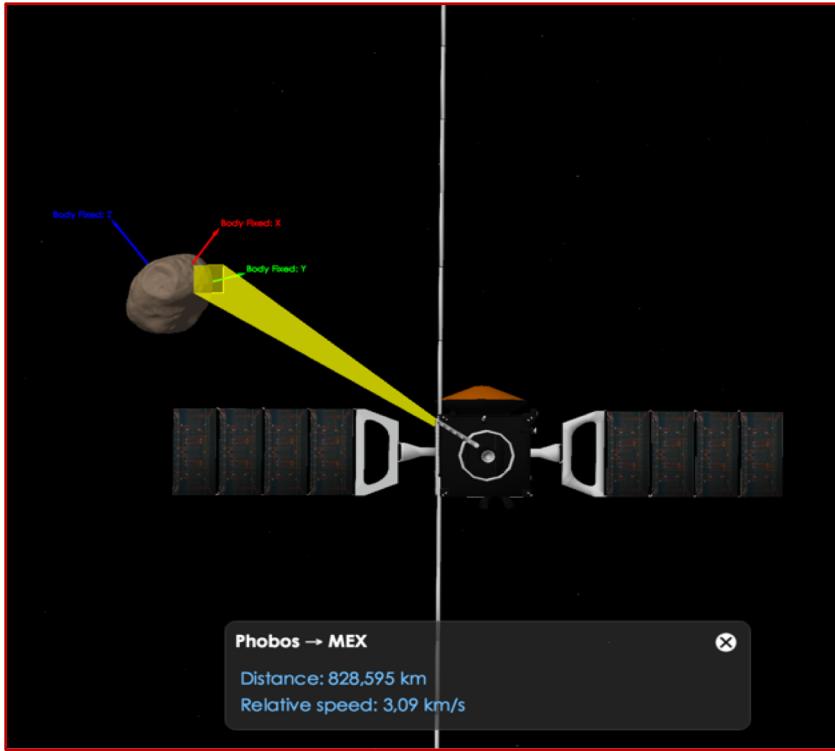
Lines of code? 75



[ftp://spiftp.esac.esa.int/temp/acardesin/spirec\\_sketch-hrsc\\_head-phobos.html](ftp://spiftp.esac.esa.int/temp/acardesin/spirec_sketch-hrsc_head-phobos.html)

# Using SPICE

- We can use Cosmographia to asses the geometry of the observation and to double-check that the kernels are correct





The **ESA SPICE Service (ESS)** based at ESAC leads the SPICE operations for ESA's planetary missions. Its main activities are:

- Generate, develop, maintain and archive the SPICE Kernel Datasets (SKD) for the ESA Planetary Missions (and Solar Orbiter);
- Develop and operates software to convert orbit, attitude, telemetry and spacecraft clock correlation data into the corresponding SPICE formats;
- Provide consultancy and support to the Science Ground Segments and the Science Community of the planetary missions for SPICE and ancillary data management.



ESA SPICE Service are: Marc Costa Sitja, Bjoern Grieger and (sometimes) a trainee, the group is managed by Christophe Arviset.

ESS also provides an instance of [WebGeocalc](#) and the [Cosmographia](#) configuration for ESA missions:

- **WebGeocalc** is a web-based interface to some SPICE Functions, extremely powerful for quick-look data analysis
- **Cosmographia** is a 3D-Visualization Tool for a full SPICE Scenario.

We provide **SPICE Training Classes** in Europe in a biannual basis. Next training 17<sup>th</sup>-19<sup>th</sup> June 2019.

This morning we held a 90 minute mini-workshop

Recording of last SPICE Training at ESAC is available in YouTube

# SPICE Kernel Dataset

- The main purpose is to provide a complete, consistent, high-quality, validated and up-to-date **SPICE Kernel Dataset (SKD)**
- A **SKD** consists on a complete set of SPICE Kernels that cover the whole mission lifespan including long term predicted trajectory and orientation. Kernels in a SKD can be classified in two main types:
  - **Setup kernels (STK)** [FK, IK, PCK, LSK] are developed by the ESA SPICE Service (ESS) and are reviewed and iterated with the SGS and with the Instrument Teams when need be during the whole duration of the mission.
  - **Time-varying kernels (TVK)** [SPK, CK, SCLK, MK] are generated by ESS with an operational pipeline and the source data is provided by the Flight Dynamics or the given SGS Downlink group in terms of OEMs, AEMs and Housekeeping TM data.
- The **SKD** will contain the following information:
  1. Set of Reference Frames of interest for geometry computations (**FK**)
  2. FoV and boresight modeling for remote and in situ sensors -at least- (**IK**)
  3. Test trajectories and orientation for the S/C (**SPK, CK**)
  4. OBT to UTC/CAL time conversion (**SCLK**)
  5. Measured and Reconstructed trajectory and orientation for the S/C (**SPK, CK**)
  6. Orientation of S/C parts and/or instruments (**CK from HK Telemetry**)
  7. Digital Shape Models of extended bodies (**DSK**)

(Completed with generic kernels for target body ephemeris and models, leap-seconds, etc)
- ESS has focused efforts on systematically providing meta-kernels (**MK**) to users.



It is also important to distinguish between SKDs published in the ESA FTP (Study and Operational) and BitBucket and the peer-reviewed **and PSA-PDS compliant Archived SKDs** (following the PDS3 and PDS4 standards from the PDS and IPDA).

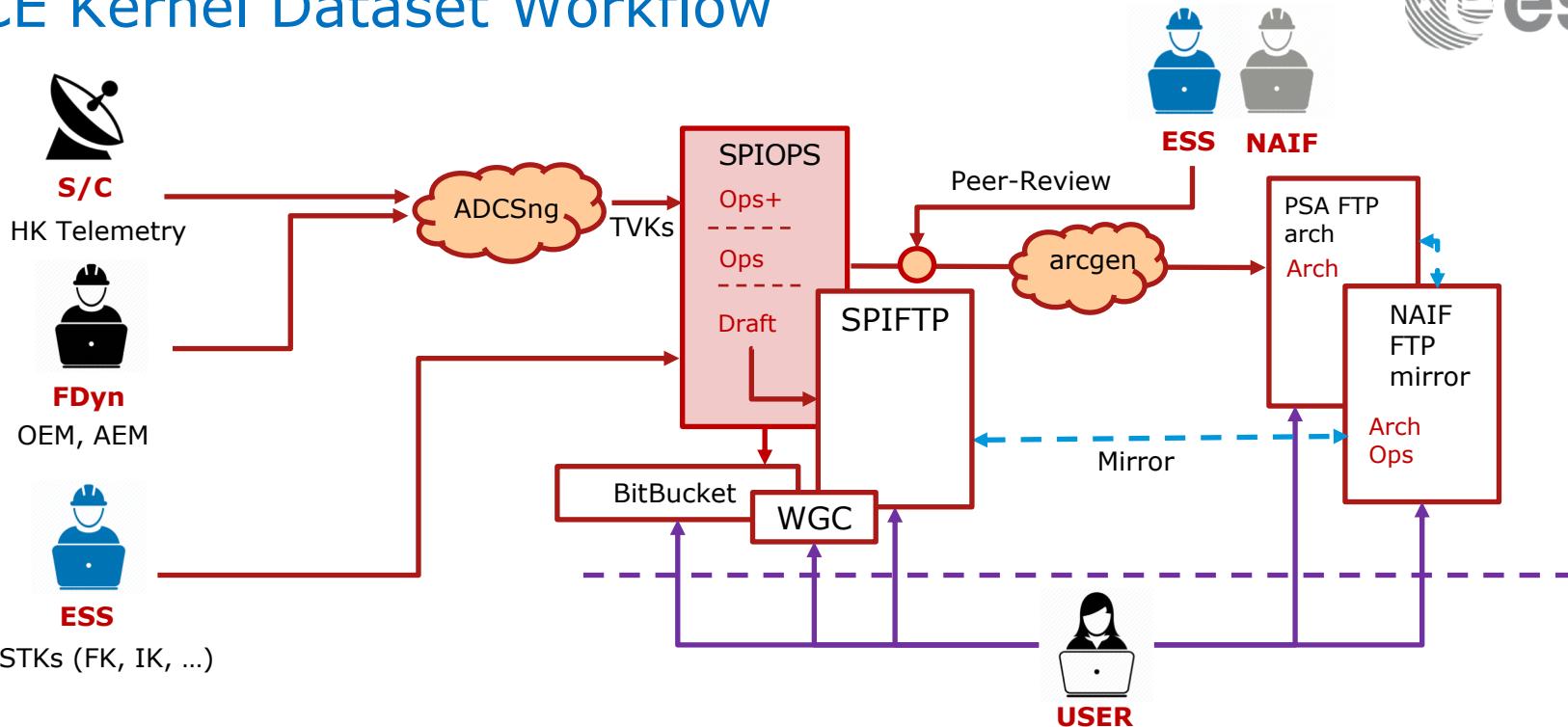
## Version and Configuration Control

- All SKDs are under configuration control and new releases happen constantly.
- SKDs are released on a regular basis when **STKs are updated** and when in operations are time tagged in a daily/weekly basis when **TVKs are updated**.
- For **operational/studies** SKD at the user's end the version is indicated by the meta-kernel used and has two components:
  - Latest SKD Version (three digit version X.Y.Z X: Major, Y: Minor, Z: Bug-fix)
  - Latest SPICE Operational pipeline run (day and count)
  - e.g.: **em16\_ops\_v410\_20190617\_002.tm**
- For **archived** SKDs the increment (version) of the archive is provided in the meta-kernel name. e.g: **em16\_v001.tm**

## The distribution of SKDs is done via:

- An operational FTP with all the kernels that were ever produced: <ftp://spiftp.esac.esa.int/data/SPICE>
- A permanent link to a ZIP file that contains the latest operational subset of the SPICE Kernels (as in BitBucket) (in the future we are exploring using Nexus repos and Dockers, if you have any advice or ideas please come to us)
- A BitBucket Git repository with a given subset of the SPICE Kernels (operational, planning, archived etc.).  
**[Https://repos.cosmos.esa.int/socci/projects/SPICE\\_KERNELS](https://repos.cosmos.esa.int/socci/projects/SPICE_KERNELS)**
  - This ensures inter-operability in between internal and external users and with WGC

# SPICE Kernel Dataset Workflow



- The Auxiliary Data Conversion System next-generation (ADCSng) generates the time-varying kernels when the mission is in operations and provides up-to-date time correlation, trajectory and orientation information to users.

# ESA SKD Status



Mission	Status	BitBucket	Archive	WGC	Cosmo	Frames and sensors	Predicted Orbit & Attitude	Reconstructed Orbit & Attitude	OBT conversion	S/C Element and Payload Orientation
ExoMars2016	OPERATIONAL		PDS4							
Mars Express	OPERATIONAL		PDS3							
BepiColombo	OPERATIONAL		PDS4							
Solar Orbiter	STUDIES									
JUICE	STUDIES		PDS4							
ExoMarsRSP	STUDIES		PDS4							
Hera	STUDIES		PDS4							
EnVision	STUDIES		PDS4							
Rosetta	LEGACY		PDS3							
Venus Express	LEGACY		PDS3							
SMART-1	LEGACY		PDS3							
Chandrayaan-1	LEGACY		PDS3							
(Cassini-) Huygens	LEGACY		PDS3							
Giotto	LEGACY		PDS3							

ESA UNCLASSIFIED - For Official Use

Christophe Arviset, Marc Costa, ESA SPICE Service | IPDA Face-to-face meeting | 12/07/2018 | Slide 21



European Space Agency

# SKDs in the PSA/PDS

**PDS3 SPICE datasets and PDS4 SPICE bundles for the archive** produced by the ESS are/will be available from the PSA UI, the PSA FTP server and the NAIF FTP server. They are to be published every 4-6 months.

## PDS3 Archives

- Current PDS3 Archived SPICE datasets available:
  - Mars Express **Last updated 2013-05-29** Next increment ~ Fall 2019
  - Rosetta **Last updated 2016-03-31** Next increment ~ Summer 2019
  - Venus Express **Last updated 2013-02-13** Next increment ~ Early 2020
- ESS still has not generated any PDS3 Archived SPICE Kernel Dataset increment, as soon as the process is started, the Rosetta and Venus Express datasets will be quasi complete (pending post-operational and MEX will recover the usual 6 months increment cadence).
- In the meanwhile instrument teams and other actors are encouraged to use the operational kernels.

## PDS4 Archives

- First PDS4 SPICE Bundle will be **ExoMars2016 and will be released Summer 2019**.
- BepiColombo will follow hopefully before the end of 2019.
- The PDS4 approach with SPICE is to minimize the effort required to archive SPICE kernels; the idea is to apply minimal changes to an operational SPICE Kernel Dataset. The rationale behind is that SPICE has proven to already be very well documented.
- Bundle structure agreed with NAIF, used by JAXA as well and adopted by the IPDA.
- More details available at the PSA PDS4 Archiving Guide, available under request.

# What is next for ESS ?

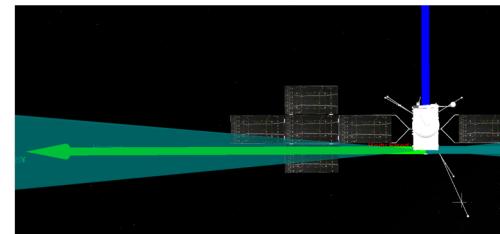
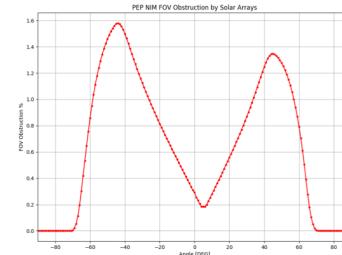
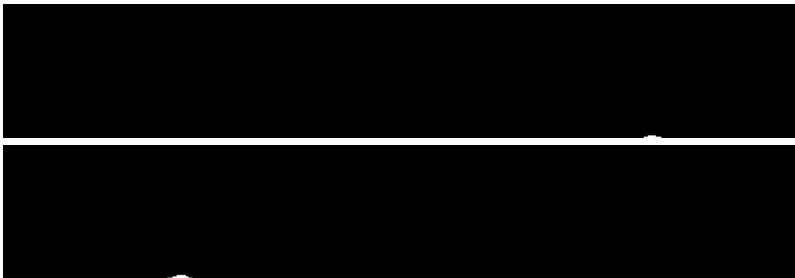


- Final review and implementation of several SPICE Kernel Datasets: ExoMars2016, BepiColombo, Mars Express and Rosetta
- Consolidation of the existing Operational Pipeline for all missions (BitBucket, etc.)
- Streamline the generation of SPICE Archives
- Generation of non-SPICE Ancillary Data PDS3 Datasets and PDS4 Bundles for the archive
- Development of a SPICE Validation Pipeline (including reporting publicly available): [spice.esac.esa.int/status](http://spice.esac.esa.int/status)
- Consolidation of ESS Python packages for the public
- Implementation of DSKs for extended bodies in all SKDs including:
  - Targets (Small Bodies, planet DEMs, etc)
  - Spacecrafts (Bus, rotating Solar Arrays, HGAs, etc)

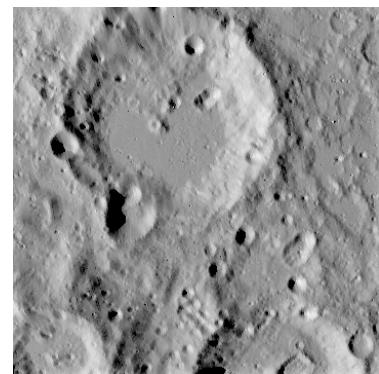
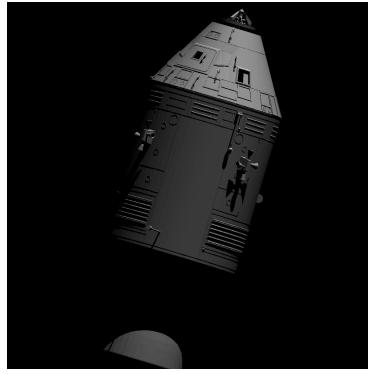
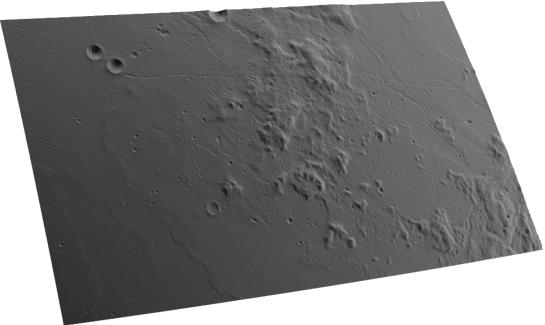
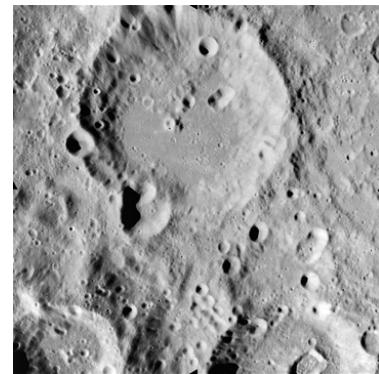
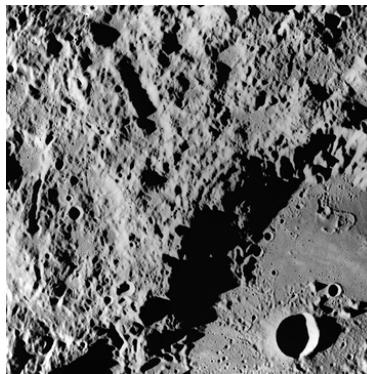
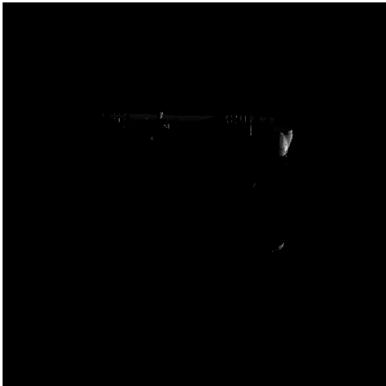
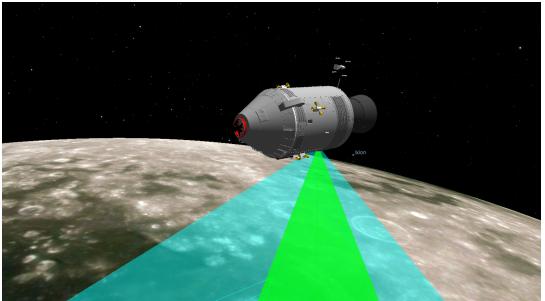
# What is next for ESS ?



- Final review and implementation of several SPICE Kernel Datasets: ExoMars2016, BepiColombo, Mars Express and Rosetta
- Consolidation of the existing Operational Pipeline for all missions (BitBucket, etc.)
- Streamline the generation of SPICE Archives
- Generation of non-SPICE Ancillary Data PDS3 Datasets and PDS4 Bundles for the archive
- Development of a SPICE Validation Pipeline (including reporting publicly available): [spice.esac.esa.int/status](http://spice.esac.esa.int/status)
- Consolidation of ESS Python packages for the public
- Implementation of DSKs for extended bodies in all SKDs including:
  - Targets (Small Bodies, planet DEMs, etc)
  - Spacecrafts (Bus, rotating Solar Arrays, HGAs, etc)



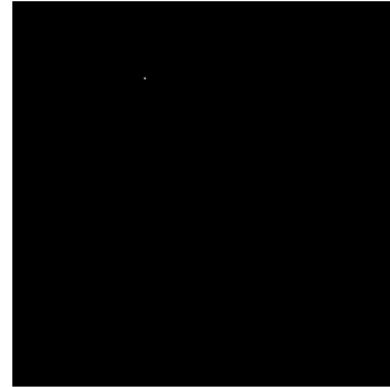
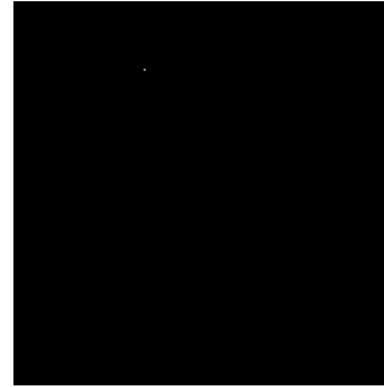
# What is next for ESS ?



# What is next for ESS ?



- Final review and implementation of several SPICE Kernel Datasets: ExoMars2016, BepiColombo, Mars Express and Rosetta
- Consolidation of the existing Operational Pipeline for all missions (BitBucket, etc.)
- Streamline the generation of SPICE Archives
- Generation of non-SPICE Ancillary Data PDS3 Datasets and PDS4 Bundles for the archive
- Development of a SPICE Validation Pipeline (including reporting publicly available): [spice.esac.esa.int/status](http://spice.esac.esa.int/status)
- Consolidation of ESS Python packages for the public
- Implementation of DSKs for extended bodies in all SKDs including:
  - Targets (Small Bodies, planet DEMs, etc)
  - Spacecrafts (Bus, rotating Solar Arrays, HGAs, etc)
- Providing reconstructed attitude



# Keeping in touch



<https://github.com/esaSPICEservice>

<https://twitter.com/SpiceEsa>

<https://tinyurl.com/y77bxntk>

## COMMUNICATE

- Everything is accessible from: <http://spice.esac.esa.int>
- Contact the service via e-mail [esa\\_spice@sciops.esa.int](mailto:esa_spice@sciops.esa.int)
- Stay tuned. You can join one the mission-specific mailing list: [spice\\_mex@sciops.esa.int](mailto:spice_mex@sciops.esa.int)
- You can also join the OpenPlanetary Slack channel: <http://openplanetary.co>

## COLLABORATE

- If you are a SPICE Kernel producer or a bi-product of your investigations are Ancillary Data (Reconstructed Trajectory, S/C Orientation, Natural Body Ephemeris) please contact us and share your data with the community.