Planetary and Space Science xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Planetary and Space Science

journal homepage: www.elsevier.com/locate/pss

A look towards the future in the handling of space science mission geometry☆,☆☆

Charles Acton*, Nathaniel Bachman, Boris Semenov, Edward Wright

Jet Propulsion Laboratory/California Institute of Technology, United States

ABSTRACT

The “SPICE” system[1] has been widely used since the days of the Magellan mission to Venus as the method for scientists and engineers to access a variety of space mission geometry such as positions, velocities, directions, orientations, sizes and shapes, and field-of-view projections (Acton, 1996). While originally focused on supporting NASA's planetary missions, the use of SPICE has slowly grown to include most worldwide planetary missions, and it has also been finding application in heliophysics and other space science disciplines. This paper peeks under the covers to see what new capabilities are being developed or planned at SPICE headquarters to better support the future of space science.

The SPICE system is implemented and maintained by NASA's Navigation and Ancillary Information Facility (NAIF) located at the Jet Propulsion Laboratory in Pasadena, California (http://naif.jpl.nasa.gov).

1. Introduction

The computation of observation geometry is an important step in the development of robotic science missions, the planning of science observations, and the analysis of the science data returned from those observations. In the early days of space exploration this sort of work was not well supported–there was little standardization applied; computations were sometimes erroneous or ill documented; computations were not made in a timely fashion; and scientists had little possibility for revising what was computed, how it was computed, when it was computed, and based on what source data it was computed.

The advent of the SPICE[2] system offered many changes to the handling of observation geometry data, putting much power in the hands of individual scientists and engineers. Users could now make the kinds of computations wanted, over time spans and cadences, or at specific instances of time of interest, and using the source data of interest. Of course this new freedom came with a cost: a would-be SPICE user needs learn enough about solar system geometry and SPICE software to be able to write her/his own geometry computation program.

2. A sound geometry footing

The SPICE system comprises both low-level data files–usually referred to as kernels–and software, mostly in the form of APIs (subroutines) used to read the kernels and then compute high-level, derived quantities of interest such as spacecraft altitude and the latitude (LAT), longitude (LON) and lighting angles where an instrument field-of-view intercepts a surface. A scientist or engineer integrates a few of the SPICE Toolkit APIs into her/his program as needed to obtain the observation geometry parameters needed for the job at hand. Also part of SPICE is a great deal of documentation, tutorials, programming lessons and training classes.

The SPICE software was originally offered in the Fortran 77 language for several platforms and operating systems, but in subsequent years versions have been added for C, IDL, MATLAB and Java Native Interface. Once offered, a computational capability is never removed or altered, thus assuring 100% backwards compatibility. All of the software is thoroughly tested before being released.

Fig. 1 provides a graphical depiction of the kinds of ancillary data used within the SPICE system. These data come from multiple sources, including flight dynamics teams, science operations centers, spacecraft and instrument builders, and science standards entities. Some of the kernels use binary formats to allow rapid access to data while others are in a text format to allow for human readability and easy editing.

This combination of ultra-stable software and data files has provided space science professionals a sound footing for building their own science or engineering applications that require precise knowledge

☆ The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

* Corresponding author.

E-mail address: charles.h.acton@jpl.nasa.gov (C. Acton).

[1] **S**pacecraft, **P**lanet, **I**nstrument, **C**amera-matrix, **E**vents

[2] https://naif.jpl.nasa.gov

http://dx.doi.org/10.1016/j.pss.2017.02.013

Received 28 September 2016; Received in revised form 24 January 2017; Accepted 22 February 2017

0032-0633/

Please cite this article as: Acton, C., Planetary and Space Science (2017), http://dx.doi.org/10.1016/j.pss.2017.02.013

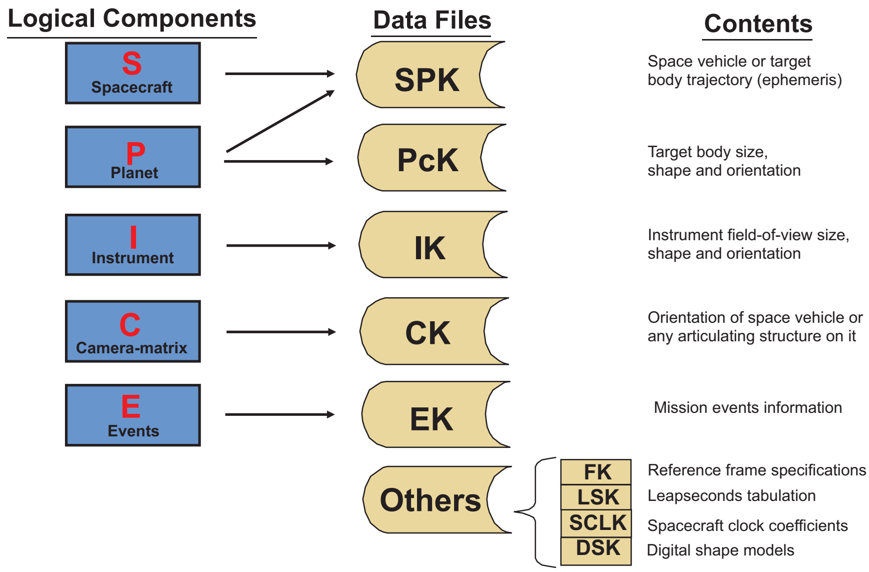

Fig. 1. SPICE Data Overview.

of mission geometry. While it takes non-trivial effort to learn to use SPICE, customers have found its reliability, stability and adaptability, as well as its no cost price tag and its lack of licensing or export restrictions to warrant the learning effort needed. The use of SPICE has slowly spread throughout the worldwide space science community.

3. More than just data analysis

While originally focused on aiding science data analysis and archiving, the use of SPICE has grown to cover the entire mission lifecycle, finding use in mission engineering applications as well as in all aspects of conducting a science investigation. Fig. 2 provides a graphical depiction of some of these other uses such as antenna scheduling and pointing, tuning of ground station transmitters and receivers, telecommunications and spacecraft thermal analyses, telescope pointing for terrestrial observations, and computations and visualizations supporting public outreach.

As a general rule one can imagine that almost any geometry related computation one might need in support of a robotic space science mission can be done using SPICE. How to make some such computations is quite obvious, being supported by very focused high-level APIs, while other such computations can be achieved after some thought by those quite familiar with SPICE, using lower-level APIs.

Part of the job of the NAIF Group is to continue adding new, high-level computational capability aimed at making the scientist's use of SPICE as easy and risk-free as possible without introducing new risk. Over the years NAIF has found this requires a delicate design balance.

4. Training

The computation of observation geometry, whether using SPICE or some other means, is inherently challenging. Essentially everything is

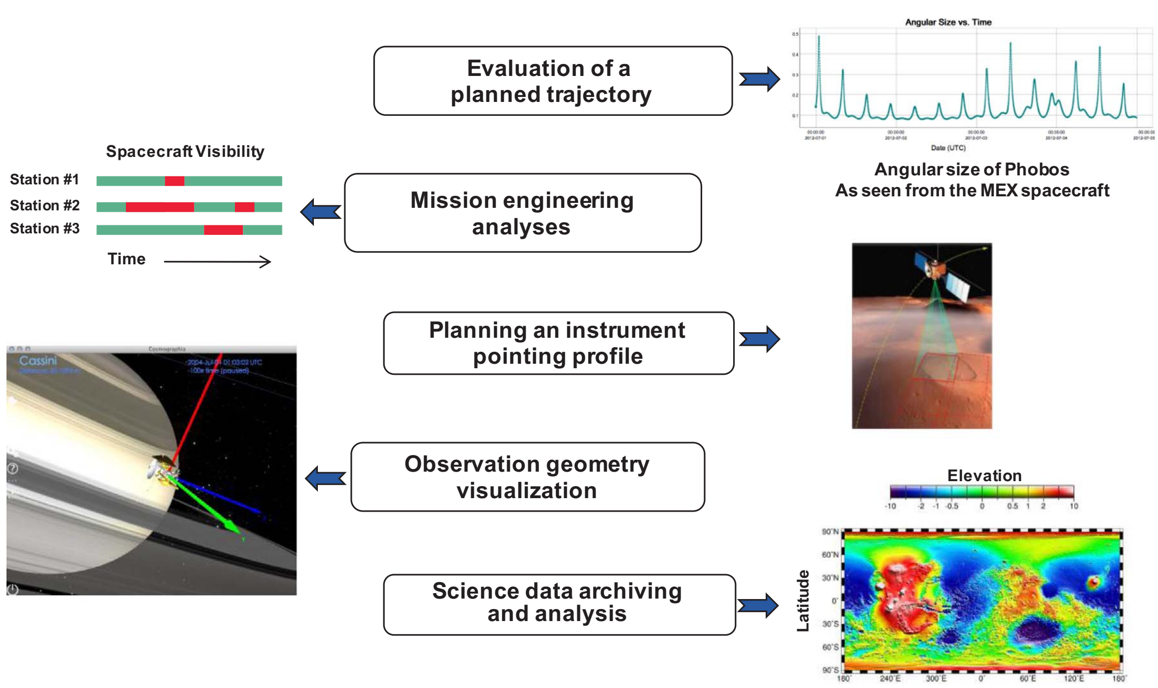

Fig. 2. Examples of how SPICE is used.