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

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. [1] 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 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 LAT/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 (Figure 1).

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.

Figure 2 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, 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 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.

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

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. Figure 3 provides a graphical peek at some of these uses. Other uses include antenna scheduling and pointing, tuning of ground station transmitters and receivers, telecommunications and spacecraft thermal analyses, telescope pointing for terrestrial observations, and support for public outreach.

4. DON'T STAND STILL!

Seeing the already very broad use of SPICE might suggest there's little left to do in terms of system development. Of course this is not the case! Every aspect of SPICE could use improvements and additions to better facilitate its use throughout the space science community, to improve interoperability with other computational elements within space science, and to lower the learning curve. As noted in the next sections of this paper the NAIF Team has already embarked on some such improvements, is about to initiate still others, and is contemplating what further should be done. NAIF hopes those who read this paper will be able to capitalize on recent and pending improvements, and will also offer suggestions for further SPICE development.

5. RECENT ADDITIONS TO SPICE

While SPICE has been around for quite some time, it continues to grow to support the evolving field of space science.

5.1. New Calculations

Many new or improved calculations have been added to the SPICE Toolkit. This includes an entire class of new calculation types, referred to as the "geometry finder" (GF) or "event finder" subsystem. Traditional SPICE calculations performed with SPICE involve calculating quantities of interest such as distances, vectors, angles, or orientations for specified times. The GF subsystem solves the inverse problem: it finds times when specified geometric conditions are met. Examples:

- Within user-specified time bounds, find all the time intervals when Saturn occults Titan as seen by the Cassini spacecraft.
- Within user-specified time bounds, find the time(s) when the angular separation between comet C-G and the sun, as seen by the Rosetta spacecraft, is at a global or local maximum.
- Within user-specified time bounds, find the time intervals when the Akatsuki spacecraft's altitude above Venus is between 400 and 450 km.

5.2. New Interfaces

For some years SPICE has been offered in four languages: Fortran 77, C, IDL and MATLAB. (The MATLAB version reportedly works under OCTAVE as well.) NAIF has been working towards the addition of a Java Native Interface (JNISpice), currently in alpha-test status but available to interested persons as it's been pretty well tested. It has been suggested that NAIF add a Python binding to the CSPICE Toolkit. We found that several users have already done this and likely some of those implementations can work for others; check here for more information: http://naif.jpl.nasa.gov/naif/links.html.

Realizing that not everyone has the skills or time to write their own SPICE-based geometry calculator, NAIF recently implemented a geometry engine with a Graphical User Interface accessed through a user's browser. This WebGeocalc (WGC) tool uses simple menus and GUI widgets to allow a user to select a computation to be made and the SPICE kernels to be used. The WGC server performs the computations needed and returns numerical and optional graphical results to the customer's browser. An ability to perform chaining using outputs from one computation as inputs to a subsequent computation makes WGC quite powerful. The WebGeocalc tool may be accessed at this location: http://naif.jpl.nasa.gov/naif/webgeocalc.html

5.3. New Models for Solar System Bodies

For many years the only means within SPICE for modeling a planet, satellite, comet or asteroid has been to use a tri-axial ellipsoid or oblate spheroid. This has become a severe limitation as target bodies have ever more physical diversity, e.g. think of the Rosetta mission's primary target, comet 67P/Churyumov-Gerasimenko, as modern missions fly closer to those target bodies, and as modern instruments achieve ever higher resolution.

To address this shortcoming NAIF is currently adding to the SPICE system two new shape model containers, both falling under the name Digital Shape Kernel (DSK). One of those containers is designed to hold a set of tessellated triangular plates, a popular shape model representation for small, irregularly shaped objects. The second container is designed to hold traditional digital elevation model (DEM) representations of large bodies with measureable terrain.

NAIF is not a fundamental producer of shape data; rather, NAIF provides container mechanisms for the shape data produced by the science/engineering community. Why do this? Software that is part of the DSK subsystem not only packages tessellated plate data or digital elevation data, but it also provides means to compute an assortment of observation geometry parameters relative to these bodies–quantities such as altitude, LAT/LON and lighting angles that are important to scientific investigations and mission engineering applications. Thus a scientist or engineer who has elected to use SPICE for other kinds of space geometry computations will not have to exit the SPICE domain in order to make computations involving tessellated plate or DEM data.

Alpha-test SPICE Toolkits that handle the tessellated plate model have been in use for some time, and this portion of the DSK subsystem will be officially added to SPICE Toolkits in the Fall of 2016. An early-stage version of the DEM style of DSK has been successfully deployed in support of the Soil Moisture Active-Passive (SMAP) earth science mission. Official integration into SPICE Toolkits of this part of the DSK subsystem will come at a future date. In parallel NAIF plans to prepare DSKs using the best publically available DEM data sets for the moon, Mercury and Mars. But any DSK user will be able to make her/his own DSKs for any body for which there are tessellated plate data or DEM data available.

5.4. Mission Visualization Tool

SPICE has always been in the numbers game, used to calculate numeric quantities used in downstream applications, in user-made graphing or visualization tools, or in publications. Some SPICE users have indeed implemented their own visualization tools that use some or much of SPICE, and some of these are available to the community. But over the years SPICE users asked NAIF if it could provide a generic, fully SPICE-enabled visualization tool having characteristics similar to the SPICE Toolkits: portable, easy to use, free and not export limited.

The NAIF Group had been experimenting with an open source tool named Celestia, then learned of a new tool named Cosmographia by the same author. It appeared Cosmographia could be the basis for a NAIF-supported visualization tool. After successful discussions with Cosmographia's author, NAIF embarked on making modifications to the tool to enhance its SPICE compatibility, to add more functionality, and to provide a variety of user-focused tutorial materials.

More information about Cosmographia and access to the download page is available here:
http://naif.jpl.nasa.gov/naif/cosmographia.html

5.5. Performance Enhancements

The SPICE software was architected to provide correct, high-precision results and broad flexibility while also alerting users to erroneous inputs wherever possible: high speed was not a consideration. Nevertheless some customers asked if performance could be improved. The NAIF Team looked into this question and found several areas where modest changes could be implemented in SPICE software that would improve run-time performance in some situations while not altering functionality or user interface. The very latest SPICE Toolkit–version N65–contains these improvements, and the next Toolkit will benefit from some additional performance enhancements.

5.6. A Running Summary of Changes

Each SPICE Toolkit includes a simple text file named "whats.new" that summarizes all of the changes in each new Toolkit release. This same file, named "What's New in SPICE," is available under each Toolkit language page on the NAIF website–for example here for the C Toolkits:
http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/index.html

6. SPICE 2.0

Perhaps the most often requested enhancement to the SPICE system is the provision of thread-safe and object-oriented versions of the Toolkits. NAIF requested and has received funding aimed at designing and building such Toolkits, and has recently hired new staff who can help with this work; we are off and running on what we are calling SPICE 2.0. We'll take this chance to add still other enhancements to the Toolkits. How long this development will take is unknown. We do note that we realize we must maintain all of the existing Toolkits as well.

7. LOOKING FURTHER AHEAD

How can the SPICE system and NAIF's services further evolve to accommodate the challenges of "big data" and space science outside of the planetary community?

7.1. Engaging the Heliophysics Community

Trying to "push" SPICE onto the heliophysics community from the top seems unlikely to succeed. There has already been some adoption of SPICE within heliophysics coming from the grassroots level, and simply allowing this hands-off growth to proceed at its own pace may be the best approach. But NAIF is open to alternate ideas.

7.2. Adding Heliophysics-focused Functionality

Adding new capabilities that could be of interest to heliophysics could facilitate this process. In this regard, a soon to be released collection of dynamic reference frames, many of which are popular in that community, should help. Further, NAIF should seek to improve dialogue with practitioners within that community to understand what further enhancements would be desirable.

7.3. Maintaining High Quality

Avoiding bugs through extensive testing, maintaining the very highest accuracy standards, and providing extensive user-focused documentation are important aspects of achieving broad community use. These efforts take a good deal of time but are well worth it. Adding automation wherever possible could help reduce the time spent on quality control; this is particularly pertinent in the development and testing of examples showing how to use SPICE APIs.

7.4. Improving Ancillary Data Management

For the purposes of this paper let us define "observation geometry" to be parameters such as altitude, instrument pointing direction, surface intercept location (LAT/LON), lighting angles (phase, incidence and emission) that are derived from "ancillary data files" such as spacecraft trajectory and attitude, target body ephemeris, size, shape and orientation, and instrument mounting alignment and field-of-view geometry. Robotic space science missions tend to produce a large number of ancillary data files, so it can be difficult for the individual scientist to know which ones to use for a given task.

For some (mostly archival) situations the introduction of meta kernels–logical aggregations of complete, consistent sets of ancillary data–has greatly improved the kernel management conundrum. But there are other situations, especially for real time flight operations, where improvements are needed. Provision of web-based tools allowing a user to quickly compare a set of SPICE ancillary data files would be a big help.

7.5. More Emphasis on Interoperability

As it is currently architected the core SPICE system is pretty much a stand-alone capability; users must incorporate SPICE APIs into their own application programs. The relatively new WebGeocalc (WGC) tool mentioned in Section 5.2 is somewhat an exception; it provides a convenient graphical user interface to a geometry engine on a remote server. But the output of WGC is restricted to simple CSV or text format files and PNG plots. A tighter connection to information system components addressing big data issues would be helpful. One example of improving this sort of interface being implemented now is the addition to WGC of VOTable[2] as an output format.

In addition to its GUI interface the WebGeocalc tool has a programmatic interface based on RESTful URLs and JSON payloads, although this has not yet been deployed. Members of the VO community have suggested NAIF look into providing Simple Application Messaging Protocol[3] (SAMP) interfaces to SPICE-based web services such as WGC.

8. CONCLUSIONS

The multi-mission SPICE system has been quite successful in supporting NASA and worldwide planetary missions, and some heliophysics missions as well. However, as the number of missions, mission operators and science data sets grow, as instruments become ever more complex, and as interest in using data from multiple instruments and missions simultaneously becomes the norm, a system such as SPICE needs to continually evolve to keep up with user needs. The NAIF team has taken some steps in this direction, but a quicker pace is called for, along with more dialogue with leaders in interoperability and open access.

[1] Acton, C.H.; (1996) Planetary and Space Science, Vol. 44, No. 1, pp. 65-70.

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.

---

[2] http://www.ivoa.net/documents/VOTable/
[3] http://www.ivoa.net/documents/SAMP/

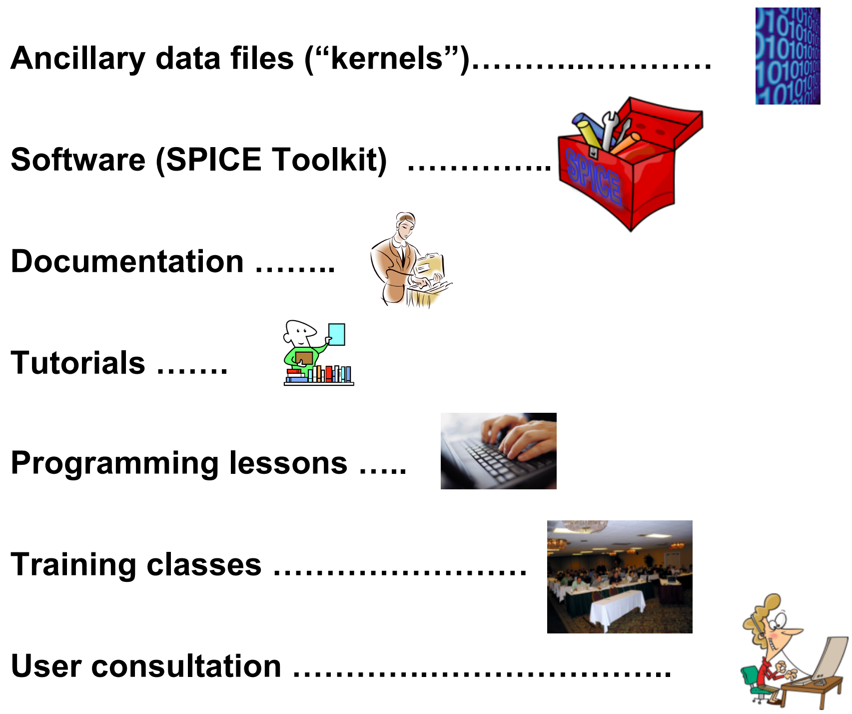

Fig.1 SPICE System Components

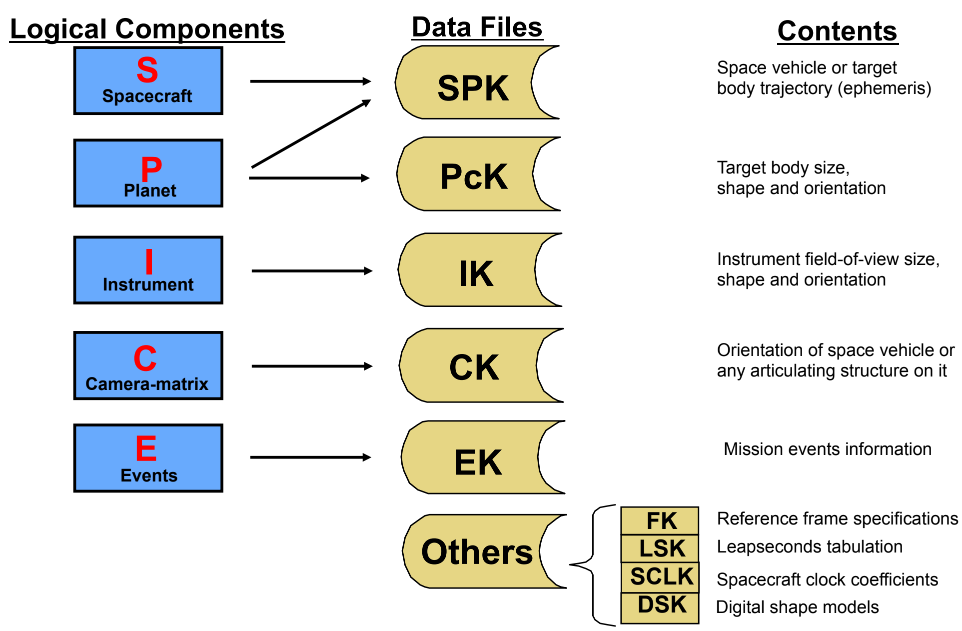

Fig.2 SPICE Data Overview

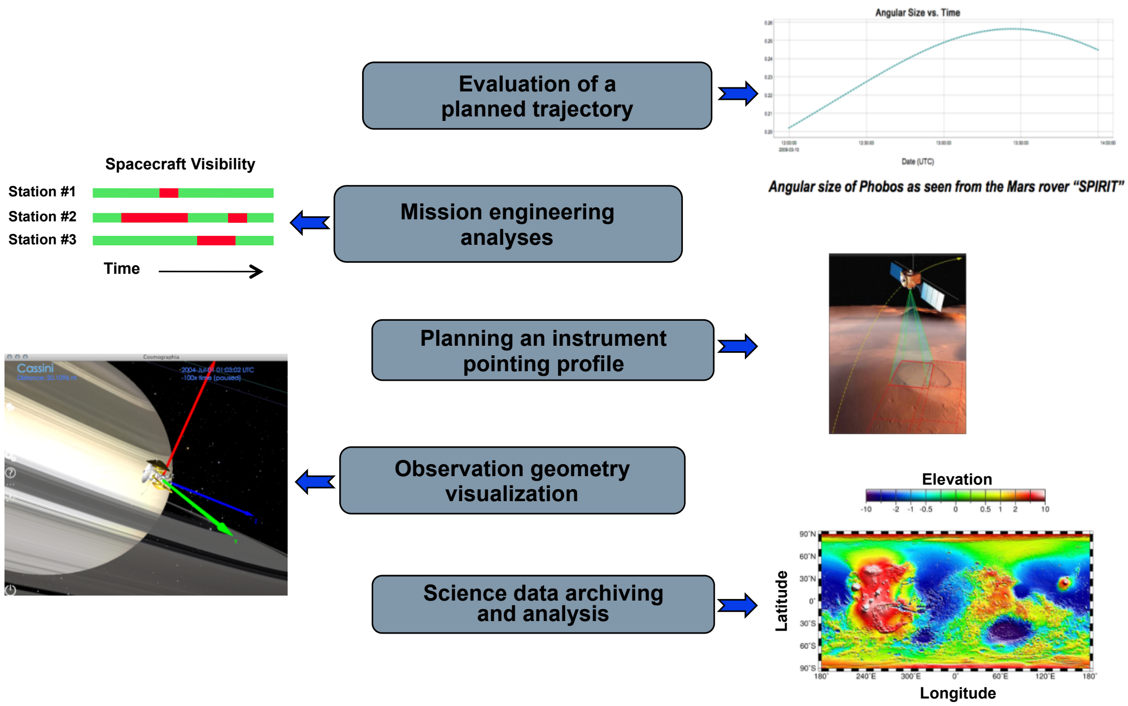

Fig. 3 Examples of how SPICE is used