USING THE SPICE SYSTEM TO HELP PLAN AND INTERPRET SPACE SCIENCE OBSERVATIONS

Charles H. Acton, Jr.

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA USA

ABSTRACT

A portable multimission information system named SPICE is used to assemble, archive, and provide easy user access to viewing geometry and other ancillary information needed by space scientists to interpret observations of bodies within our solar system. The modular nature of this system lends it to use in planning such observations as well. With a successful proof of concept on Voyager, the SPICE system has been adapted to the Magellan, Galileo and Mars Observer missions, and to a variety of ground based operations. Adaptation of SPICE for Cassini and the Russian Mars 94/96 projects is underway, and work on Cassini will follow,

SPICE has been used to support observation planning for moving targets on the Hubble Space Telescope Project. Applications for SPICE on earth science, space physics and other astrophysics missions are under consideration.

Key Words: Data analysis, observation planning, ancillary data, geometry, Level 6 data

1. INTRODUCTION

Distributed information systems are becoming popular within the space science community. With affordable access to substantial processor power, large scale mass storage and high speed networks, scientists today wish to—or must—design and operate their own science data analysis systems.

Doing one's own thing often requires a substantial investment of local resources: repeated at each end-user site, this may seem unduly wasteful. Yet local processing offers many advantages and may be the only means for obtaining the derived products needed for one's scientific pursuits. Local processing affords the scientist or other end user a level of knowledge about the mechanics of the data processing often missing when the processing is largely done at a central site. With good in-house tools the user has specific knowledge about the algorithms and processing parameters used and the processes applied as raw data (Level 0) are refined into useful products. More importantly, the user has the flexibility inherent in local control. Within some limits he may select what quantities are derived, which data are used in the process and when the processing takes place. The pipeline may be tuned or rearranged as data analysis requirements evolve or data sources change.

2. SPICE OVERVIEW

SPICE offers one approach for producing and using ancillary data within a distributed information system architecture. SPICE provides a portable, multi-discipline mechanism useful in both planning space science observations and in reducing the data obtained from those observations. Within the SPICE context, ancillary data are broadly defined as those used to determine:

- where an instrument was located while taking data,
- where the instrument was pointed and what targets it could see,
- how those targets would appear at the time of observation,
- how an instrument was acquiring data, and
- what else of significance to science data analysis was occurring.

In this context a "target" could be a whole planet, satellite, comet or asteroid, a surface or atmospheric feature, or a star or other galactic entity. Targets on planet earth are as valid as any others for application of SPICE technology.

2.1 SPICE Components

The principal components of the SPICE system are a set of elemental data files–called kernels–and software. Some of the SPICE software is used to produce kernel files, some is used to read those files to find information or calculate parameters needed to plan observations or interpret sensor data. The contents of the SPICE components are summarized in Figure 1. The reader can see that the acronym might better have been "SPICES."

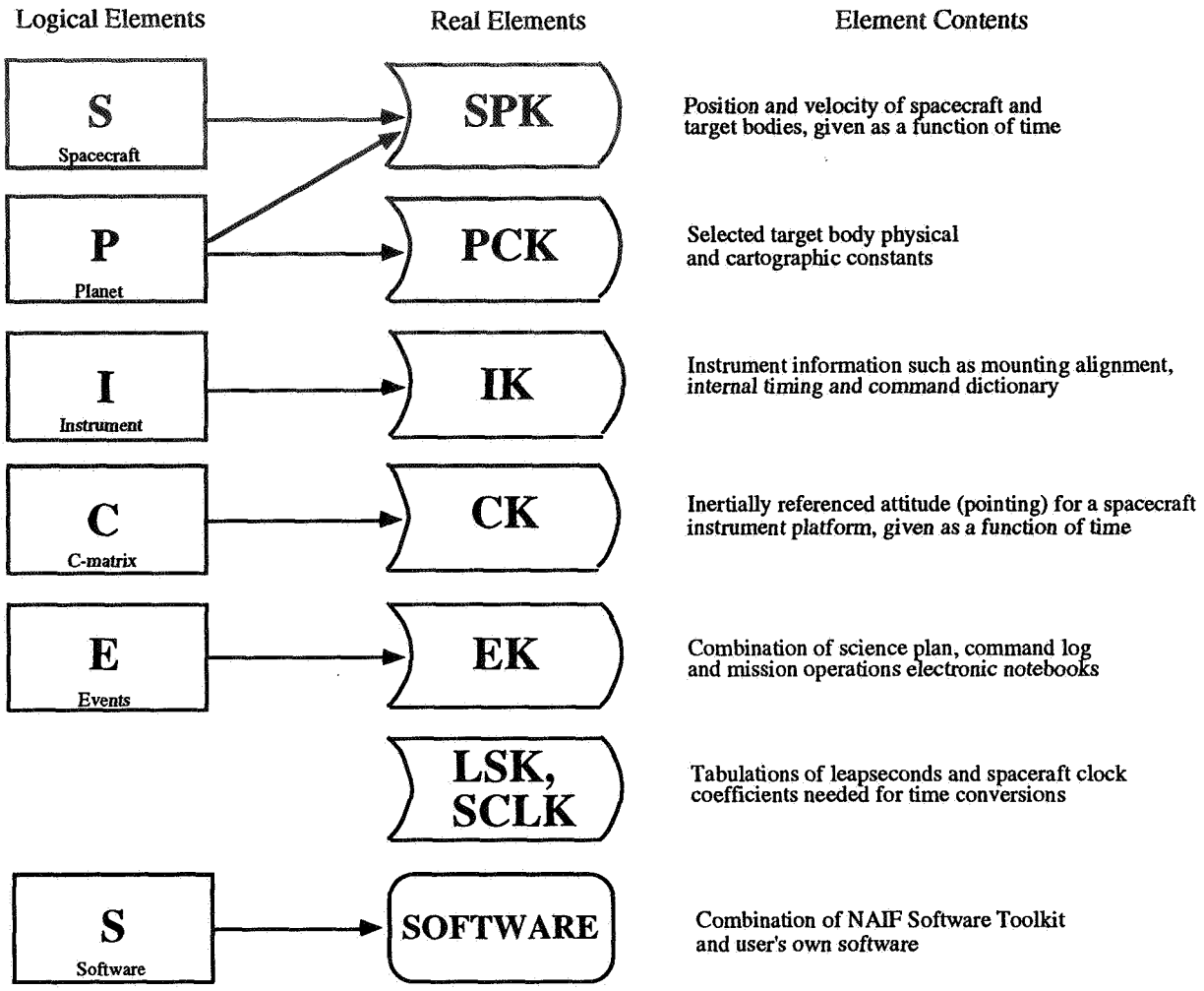

Figure 1. Principal Elements of the SPICE System

2.2 Making SPICE Kernel Files

Instances of each type of SPICE kernel file should be produced by the entity most qualified to do so—the person or group most familiar with the components and processes from which ancillary data are produced. For example, a project's navigation team responsible for estimating the orbit of a spacecraft would be responsible for producing SPK kernel files. Similarly a PI or facility instrument team would construct the I-kernel for that instrument (or would at least provide the information needed for constructing an I-kernel).

2.3 Distributing SPICE Products

Whatever the source, SPICE kernel files and NAIF Toolkit software are normally collected and cataloged in a central repository for subsequent distribution to end users. For a flight project the repository likely resides at the mission operations center for the duration of the project. In other cases the repository could be a local, national or international archive center established for the archive and distribution of discipline specific science data products.

A mission operations center might make automatic, regular distributions of project SPICE data to individuals or agencies directly associated with the project. Alternatively, an end user needing ancillary data could contact a mission or discipline archive center to order specific data of interest. The order might be effected through an on-line catalog, or it could be placed with the archive's staff using electronic mail or telephone.

SPICE data are delivered to the user by whatever means is available and best suited to both parties.

Magnetic tapes or diskettes are the traditional distribution media; these are still sometimes the best or only method available. Worldwide electronic networks offer convenience and immediacy for small to modest data volumes. CD-ROM and write once CD (WOCD) will soon play a major role.

2.4 Using SPICE Kernels and Toolkit Software

The scientist or engineer builds an application program to address a particular need. This program integrates appropriate subroutines from SPICELIB (SPICE Library)—the principal component of the NAIF Toolkit software—with the user's own subroutines. Some SPICELIB routines provide access to the data within SPICE kernel files. For example, calling SPICELIB routine SPKEZ returns the Cartesian state vector giving the position of a target relative to an observer. The "observer" is typically a spacecraft, but it could be any object for which ephemeris data is available within an SPK kernel file.

Other SPICELIB routines combine data extracted from the kernels to produce derived (higher level) geometric parameters or related ancillary information needed to plan observations or to interpret science data. Spacecraft altitude, lighting angles and latitude and longitude of an instrument's optical axis intercept with the target are typical of the computations available within SPICELIB.

3. INTENDED CHARACTERISTICS OF THE SPICE SYSTEM

3.1 Correct Results Are Obtained

It is imperative that, within realistic limits, an ancillary information system such as SPICE provide results that are "correct". Whether or not a SPICE user gets the intended results depends on several factors.

- Correct and accurate data must be assembled for placement in SPICE kernel files.
- Software used to produce SPICE kernels and subsequently to read them and compute derived quantities must be accurately specified and must perform as specified.
- SPICE kernel files must be correctly labeled.
- Customers–kernel producers as well as kernel users–must be able to determine with confidence which SPICELIB subroutines to use to yield the intended results. Customers must also realize when a desired functionality is not (directly) available within the SPICE library.

Computation precision and methods of handling exceptional cases are overarching factors that add another dimension to the issue of determining correctness.

Seeking out substantial, definitive customer feedback seems the best method of obtaining a composite measure of correctness covering the factors above. In this regard the SPICE system–in its current state–is not without shortcomings. Nevertheless, most users report success with a wide range of SPICE applications. The growing number of repeat customers and referrals further bolsters the sense of success.

3.2 Reasonably Easy to Use

Ideally all software would be easily used by experts and novices. But to a certain extent one must trade off simplicity against functionality; the easier a program is to use the more difficult it becomes to have it accomplish significant tasks. Similar tradeoffs are made between ease of use versus portability and broad applicability.

How easy it is to use a piece of software is very subjective. It would be misleading to assert that SPICE is easy to use. But it may be said that a substantial fraction of the NASA resources invested in SPICE development have been applied to documentation and tutorial materials created for end users, and these have helped the more than 200 SPICE users achieve their objectives.

3.3 Portable Data and Software

Early information systems were centralized—accessed by users who came to the mission operations center or who dialed in from a remote location. Portability was not an issue. More recently, data have been distributed with a file description document; it is the user's responsibility to decipher that document in order to write code to read a dataset. Now the trend is towards increasingly decentralized systems, with both data and allied software delivered to users operating on many different types of computing platforms.

Portability of SPICE kernel files is achieved by using text-format files Several of the kernel files exist only in text format and so are readily ported. Those kernel files that are normally in binary format are translated to text using a portable NAIF Toolkit utility program prior to being ported to a heterogeneous host. The same utility program is used to translate the file back to the binary format of the new host once the data have been moved there. This translation process takes a little extra time and transfers involve more data. But the scheme works across the many computer systems and networks in use by scientists and engineers.

Achieving software portability is an elusive goal, traded against robustness and ease of use, and achievable in only a marginal sense. SPICE uses ANSI FORTRAN 77 code, and while environmental dependencies cannot be totally eliminated, in SPICE they have been minimized, isolated and clearly documented. Changes needed to port the NAIF Toolkit software to several popular platform/ compiler combinations are implemented and tested before the Toolkit is released to the user community; the current complement is listed in Table 1.

Platforms To Which NAIF Staff Have Ported The Toolkit	Platforms To Which Others Have Ported the Toolkit
DEC VAX VMS	DEC VAX Ultrix
Hewlett Packard 700	Cray
Sun	Data General
PC/Lahey EMS-32 FORTRAN	Silicon Graphics (IRIS)
Macintosh/Language Systems FORTRAN	IBM mainframe
NeXT/Absoft FORTRAN	

Table 1. Examples of Platforms Where NAIF Software Is Operational

Several customers have inquired about the availability of the NAIF Toolkit in the C language—it is not. However, many customers successfully call SPICELIB routines from C programs. Most compilers support cross-language compatibility.

3.4 Flexibility - Separable and Extensible Components

To endure, a data system must be easily adapted to meet new or revised requirements.

While the SPICE kernels and the allied Toolkit software are designed to operate as an integrated system, subsets of SPICE also prove useful. One example of SPICE component separability can be found in the widespread use of SPK kernel files containing only planet and satellite ephemeris data (no spacecraft ephemeris) for terrestrial observation planning and data analyses.

Each kernel design had extensibility as a major objective. Two approaches were used. For those kernels that are text files, data are formatted in a simple but flexible **KEYWORD = VALUE** style. New keyword/value pairs can be added as needed. Descriptive text that is ignored by kernel access software can be inserted wherever appropriate.

The binary kernels are built upon specially developed data structures that were designed to accommodate growth. Each structure supports a family of data types, unlimited in number. As an example, the SPK kernel file family currently consists of eight data types, such as Chebyschev polynomials, conic elements and discrete state vectors.

3.5 Affordable

Economic considerations–always important–today receive extra attention from NASA managers. SPICE rates high in this regard, partly because of the flexibility and wide applicability noted earlier. These qualities allow the cost of system development and maintenance to be shared across a broad user base.

Equally important is the attention paid to ensuring that SPICE software maintenance costs will be reasonable. The subroutines are written in a consistent, easy-to-read style and are highly annotated–approximately 70 percent of the source code is documentation. The NAIF staff refrains from using FORTRAN features that frequently make code difficult to maintain: examples are COMMON blocks and EQUIVALENCE statements. But use of a few extensions to the ANSI FORTRAN 77 standard which promote code maintainability are permitted in the local master copy of the Toolkit. A precompiler is used to translate this code into pure ANSI FORTRAN 77 before Toolkit deliveries are made.

4. SPICE KERNEL FILE SPECIFICATIONS

4.1 SP Kernel File

SPK files contain data that, when interpreted by SPK "reader" subroutines, yield position and velocity of one "ephemeris object" relative to another at the time requested by the user. In the language of SPICE, spacecraft, planets, satellites, planet-satellite barycenters, comets, asteroids, the sun and the solar system barycenter are all ephemeris objects.

The position and velocity returned from SPK readers at the requested time are given as Cartesian vectors in a user-selected inertial reference frame, such as B1950 or J2000. Either true or apparent position may be requested. SPK kernel files are structured as binary, direct-access files to provide quick response to requests.

4.2 PC Kernel File

PCK files contain the names and values of physical and cartographic constants for target bodies—items that change infrequently. NAIF assembles and distributes a version of the PCK file that contains those data needed to determine spin axis orientation, size and shape (three axes) and location of the prime meridian for all planets and satellites, as determined by the International Astronomical Union. A PCK file could include similar kinds of target model data for comets or asteroids. Data in PCK files are most frequently used to define a body-fixed reference frame used to compute locations of objects in cartographic coordinates (latitude and longitude).

All PCK files use text format, so it is easy for users to change parameter values, add or delete data items, or add or delete target bodies in their local copies. Any text editor will do.

Data in a PCK file are formatted as KEYWORD=VALUE structures. Notes or references in free format text may be inserted where desired throughout the file. (Only the KEYWORD=VALUE data are read by Toolkit software.)

4.3 I Kernel File

IK files contain an assortment of science instrument-related information. A separate IK file is constructed for each instrument. These files are text format, using the same KEYWORD=VALUE structure used in the PCK file.

An IK file contains an instrument's mounting alignment relative to a spacecraft reference frame for which a CK file exists. An IK could include data that model articulation of an internal mirror used to point the instrument's optic axis. Also possibly found in IK files are an instrument's internal timing offset parameters, used to correlate data time tags with the instant that photons or particles impinged on the instrument's sensor.

An instrument's command dictionary, operating description and references to calibration files and pertinent instrument documents should be included in its IK file. Such text fields are not accessed by Toolkit software, but may be easily displayed or printed using standard computer utilities.

4.4 C Kernel File

CK files consist of time tagged quaternions used by Toolkit "readers" to construct the transformation matrix that rotates a vector from an inertial reference frame to coordinates fixed to an instrument platform. The data in a CK kernel file are derived from a spacecraft's attitude and articulation telemetry data.

CK kernel files use the same fundamental structure found in SPK files and offer the same flexibility of data content and quick data access. At present three CK data types are supported.

4.5 E Kernel File

The E-kernel consists of three distinct components: Science Plan, Events and Notebook. The Science Plan and Notebook components use identical text file structures. The Events component is a binary file, somewhat like the SPK and CK files, although the underlying data structure is different.

Time and "instrument" ID are the parameters common to all three EK components. Here, "instrument" is used in a broad sense: the spacecraft, the ground data system and any other mission element of interest may be treated as an EK "instrument" along with the bona-fide science data sensors.

Each science instrument team and mission operations team is encouraged to supply inputs to each of the EK components. The collection of inputs for each EK component type are merged in daily or weekly collections.

The Science Plan component records broad scientific objectives for the operation of an instrument over the applicable time period.

The Events component records the commands–specifically, English language transliterations–sent to the spacecraft and its instruments in order to carry out the objectives of the Science Plan.

The Notebook component is used to record notes about problems, unexpected situations or general information of interest observed by scientists and mission operations staff as the instrument and spacecraft commands get executed. In an ideal world where everything worked exactly as planned, the Notebook component might never be used.

4.6 Miscellaneous Kernel Files

Two miscellaneous kernel files are needed to support typical planetary science missions. One, the Leapseconds Kernel (LSK), contains a tabulation of the leapseconds and related data needed by SPICELIB subroutines used to convert between civil time (UTC) and ephemeris time (ET, also called Barycentric Dynamical Time or TDB). This conversion is needed because ET is the independent variable used in SPICE SPK and CK files. (It could also be used in EK files.)

Similarly, the Spacecraft Clock Kernel file (SCLK) contains a tabulation of data needed for conversion between ET and the spacecraft clock time tags assigned to science and engineering telemetry returned from the spacecraft.

4.7 Kernel File Labeling

All SPICE kernel files contain an area set aside for internal labels. These internal labels can be used to help construct an SFDU, PDS Label, FITS Label or other formal metadata structure required by a particular discipline information system.

5. NAIF TOOLKIT CONTENTS

The principal component of the NAIF Toolkit is a library of ANSI FORTRAN 77 subroutines and functions named "SPICELIB." Users of SPICE kernel files usually need little or no understanding of the file structures and specific contents since their interface to these datasets is always realized through the SPICE kernel file "reader" subroutines found in SPICELIB.

In addition to the kernel file readers, SPICELIB contains a broad set of subroutines designed to further assist scientists with the planning or interpretation of space science observations. Examples of the functional categories addressed by these modules are time translation, reference frame and coordinate conversion, solid geometry, vector and matrix algebra, conic element applications, string manipulation, parsing and array manipulation.

Beyond SPICELIB, the Toolkit contains "cookbook programs" that are samples of typical use of SPICELIB subroutines and functions. It also includes several important utility programs that provide conversion, summarizing and labeling functions for binary SPICE kernel files.

6. SPICE SYSTEM APPLICATIONS

The SPICE system has been built and tested in a predominantly planetary science environment; its use in supporting science observation planning and science data analyses in conjunction with planetary flight projects is well established. SPICE has also been applied in other space science disciplines. Table 2 summarizes current and planned applications. It also identifies examples of possible further use.

Current Applications	Tentative or Under Discussion	Examples of Future Possibilities
Voyager (Uranus/Neptune)	Radioastron/VSOP	Eos
Magellan	Spectrum-R/G	Interbol
Galileo	Clementine	Discovery Program
Hubble Space Telescope		Lunar Scout
Mars Observer		SIRTF
Mars 94/96		AXAF
Cassini		Pluto Flyby

Table 2. Current and Possible SPICE Applications

SPICE is also used informally to support a variety of other programs and applications; IUE observations of satellites, terrestrial observatory observations, mission concept studies and solar system dynamics analyses are among these.

With each new application the growing cadre of scientists and engineers already familiar with SPICE will find it easy to deal with processes requiring the use of ancillary data. And with each new SPICE application funding organizations will find the marginal cost for adapting and operating this proven information system to be very reasonable.

This research was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.