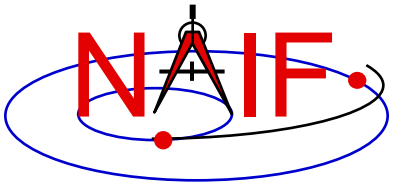
Summary of Key Points

June 2019



Which Pieces of SPICE Must I Use?

- There's not a simple answer
  - Depends on what activity or mission you are working on
  - Depends on what computation(s) you wish to make
- Don't feel overwhelmed
  - Many seemingly complex computations can be made using just a few SPICE APIs
- The next several charts highlight some key points
  - We assume you have already looked at the major SPICE tutorials, or already have some familiarity with SPICE
  - We assume you have successfully downloaded and installed the SPICE Toolkit
- Consider printing this tutorial and keeping it near your workstation

Reminder of Key Subsystems

Navigation and Ancillary Information Facility

SPK: Position (and velocity) of things ("ephemeris objects")

PCK: Size/shape/orientation of solar system bodies

- For binary PCKs, only orientation is provided; use a text PCK to obtain size/shape
- See also DSK below

IK: Instrument field-of-view geometry (see also FK below)

CK: Orientation of spacecraft or spacecraft structures that rotate

FK: Definition and specification details for many reference frames; also includes instrument mounting alignments

DSK: High fidelity shape data, better than what's in a text PCK

(But limited availability)

LSK: Time conversion: UTC (SCET) $\Longleftrightarrow$ ET (TDB)

SCLK and LSK: Time conversion: SCLK $\Longleftrightarrow$ ET (TDB)

Primary Kernel Interfaces - 1

Navigation and Ancillary Information Facility

Which SPICE APIs are most commonly called to use data obtained from a given kernel type?

SPK	SPKEZR, SPKPOS, SPKCOV, SPKOBJ
PCK	SXFORM, PXFORM, SPKEZR, SPKPOS, BODVRD
IK	GETFOV, G*POOL
CK	SXFORM, PXFORM SPKEZR, SPKPOS, CKCOV, CKOBJ (CKGPAV, CKGP)
FK	SXFORM, PXFORM, SPKEZR, SPKPOS
LSK	STR2ET, TIMOUT, SCE2C, SCT2E, SCE2S, SCS2E
SCLK	SCS2E, SCE2S SXFORM, PXFORM, SPKEZR, SPKPOS
DSK*	SINCPT, LATSRF, ILLUMF, SRFNRM, LIMBPT, TERMPT, ...

* Partial implementation starting with N66 Toolkits

Notes: FURNSH is used to load (provide access to) all SPICE kernels.
API names shown are for FORTRAN versions:
- **use lower case and add an "_c" when using C**
- **use lower case and prepend "cspice_" when using Icy (IDL) and Mice (MATLAB)**

Primary Kernel Interfaces - 2

Navigation and Ancillary Information Facility

For a given high-level module, which kinds of kernels will or may be needed?

	Kernel Type(s) Needed							
Module Name	**SPK**	**PCK**	**IK**	**CK**	**FK**	**LSK**	**SCLK**	**DSK**
SPKEZR, SPKPOS	Y	M		M	M	L	M	
SXFORM, PXFORM	M	M		M	L	M	M	
CKGP, CKGPAV	M	M		Y	M	L	L	
GETFOV			Y					
G*POOL		M	M					
STR2ET, TIMOUT						Y		
SCS2E, SCE2S						Y	Y	
CHRONOS (time conversion app.)	M	M		M	M	Y	M	

Yes = is needed
Likely = very likely needed
Maybe = may be needed

Primary Kernel Interfaces - 3

Navigation and Ancillary Information Facility

More: for a given high-level module, which kinds of kernels will or may be needed?

Kernel Type(s) Needed

Module Name	SPK	PCK	IK	CK	FK	LSK	SCLK	DSK*
SINCPT	Y	L	M	M	L	L	M	M
DSKXV, DSKXSI	M	M	M	M	M	M	M	Y
LATSRF		M		M	M	M	M	M
ILUMIN, ILLUMG, ILLUMF	Y	L		M	M	L	M	M
SUBPNT, SUBSLR	Y	L		M	M	L	M	M
GFOCLT, OCCULT	Y	L		M	M	L	M	M
SRFNRM		M		M	M	M	M	M
LIMBPT	Y	L		M	M	L	M	M
TERMPT	Y	Y		M	M	L	M	M

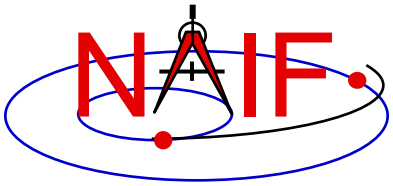
* Partial implementation starting with N66 Toolkits

Yes = is needed
Likely = likely needed
Maybe = may be needed

Kernel "Coverage" Cautions

Navigation and Ancillary Information Facility

- **Your set of kernels must:**
  - **contain data for all "objects" of interest**
    - » **Sometimes you must include intermediary objects that provide a connection (recall the chaining discussion in the SPK tutorial)**
  - **contain data covering the time span(s) of interest to you**
    - » **Watch out for data gaps within that time span**
    - » **Watch out for the difference between ET and UTC**
      - **The difference as of 2017 January 01 is ~69.182 seconds (ET > UTC)**
  - **contain all the kernel types needed by SPICE to answer your question**
    - » **As the previous charts show, you may need one or more kernels that are not obvious**
  - **be managed (loaded) properly if there are overlapping (competing) data within the set of files you are using**



How Can I Find Possibly Useful Toolkit Modules?

- Review the previous charts
- Look at the appropriate SPICE tutorial(s)
- Look at the "Most Used xxx APIs" document
  …/doc/html/info/mostused.html
- Search the permuted index:
  - spicelib_idx for the FORTRAN toolkits …/doc/html/info/spicelib_idx.html
    - » This index also correlates entry point names with source code files.
  - cspice_idx for the C toolkits …/doc/html/info/cspice_idx.html
  - icy_idx for the IDL toolkits …/doc/html/info/icy_idx.html
  - mice_idx for the MATLAB toolkits …/doc/html/info/mice_idx.html
- Read relevant portions of a SPICE "required reading" technical reference document (e.g. "spk.req")
  - …/doc/html/req/spk.html for the hyperlinked html version (best)
  - …/doc/spk.req for the plain text version

How Can I Understand How To Use Those Modules?

Navigation and Ancillary Information Facility

- The primary user-oriented documentation about each module is found in the "header" located at the top of each source code file and also in the module's HTML page in the API reference guide.
  - You can "Google" an API name to see its header
    - » For example: spkezr, spkezr_c, or cspice_spkezr (for Icy or Mice)
  - (More documentation is found at the additional entry points for those FORTRAN modules that have multiple entry points.)
- Reference documentation for major subsystems is found in like-named "required reading" documents (e.g. spk.req, ck.req, etc.)
- The SPICE tutorials contain much helpful information.
- NAIF's self-training materials provide an orderly approach to learning about SPICE:
  - https://naif.jpl.nasa.gov/naif/self_training.html

Does NAIF Provide Any Examples?

Navigation and Ancillary Information Facility

- Nearly all module headers contain one or more working examples
- "Most Useful SPICELIB Subroutines" has code fragments …/doc/html/info/mostused.html
- The "required reading" reference documents often contain examples …/doc/html/req/index.html
- The "Program_<language>" tutorial contains a substantial working example
- Some simple "cookbook" programs are found in the Toolkit …/src/cookbook/…
- Make use of the SPICE Programming Lessons available from the NAIF server
  - ftp://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Lessons/