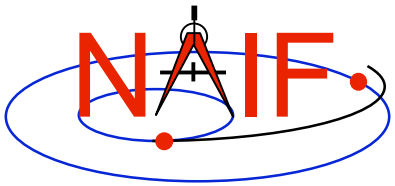


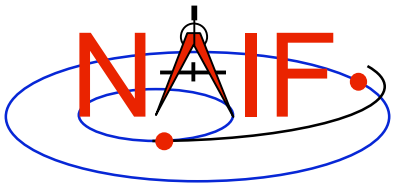
Navigation and Ancillary Information Facility

“Camera-matrix” Kernel CK

(Orientation or Attitude Kernel)

Emphasis on reading CK files

March 2006



CK File Contents - 1

- **A CK file holds orientation data for a spacecraft or a moving structure on the spacecraft**
  - **"Orientation data" ⇒ orientation matrix used to rotate vectors from a base reference frame ("from" frame) into a target reference frame ("to" frame)**
  - **Optionally may include angular velocity of rotation of the "to" frame with respect to the "from" frame**
- **A CK file should also contain comments--sometimes called metadata--that provide some details about the purpose for the particular CK, when and how it was made, and what time span(s) the data cover**

CK File Contents - 2

- **A single CK file can hold orientation data for just one, or for any combination of spacecraft or their structures**
  - **Some examples**
    1. **Huygens Probe**
    2. **Cassini Orbiter and its CDA instrument mirror**
    3. **Mars Express Orbiter, PFS scanner, Beagle Lander**
    4. **MRO orbiter, MRO high gain antenna, MRO solar arrays**
- **Most CKs contain data for just one structure**

CK File Varieties

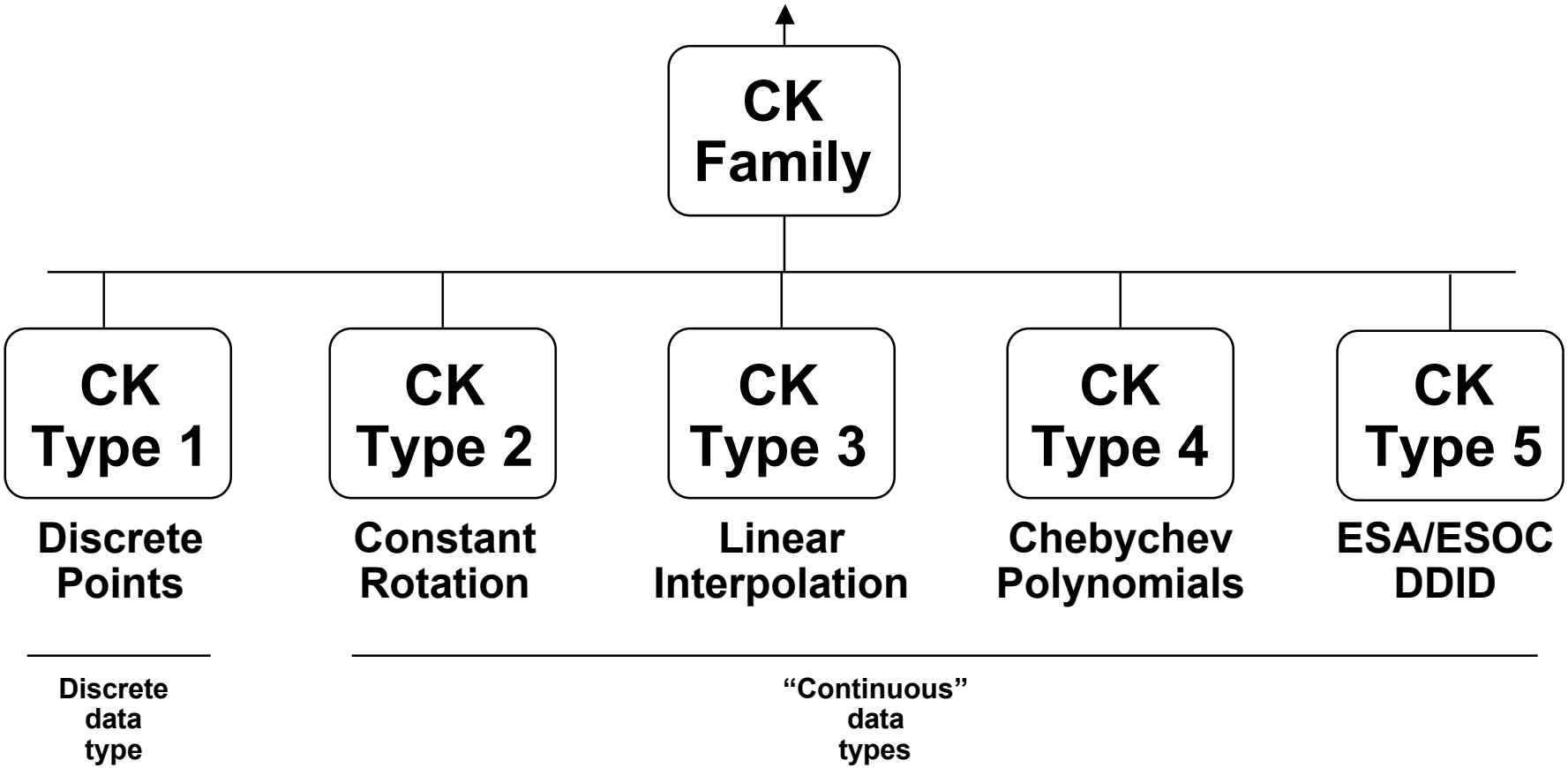
- "Reconstruction"
  - A CK file can be made from actual orientation telemetry returned from a spacecraft or other structure
  - This kind of file is generally used for science data analysis or spacecraft performance analysis

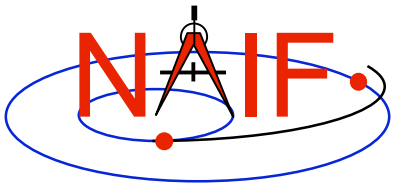
- "Predict"
  - A CK file can be made using information that predicts what the orientation will be some time in the future
    - » Input data usually come from a modeling program, or a set of orientation rules
  - This kind of file is generally used for engineering assessment, science observation planning, software testing and quick-look science data analysis
    - » If it has good fidelity, such a file might be used to "fill in the data gaps" of a reconstruction CK file

CK Data Types Concept

Navigation and Ancillary Information Facility

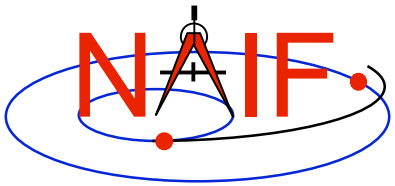
The underlying data are of varying types, but the user interfaces to each of these are the same.





Kernel Data needed

- To get orientation (rotation matrix) and angular velocity of a spacecraft or one of its moving structures, one needs three SPICE file types: CK, SCLK, LSK, and may also need an FK
  - CK contains spacecraft or other structure orientation
  - SCLK and LSK contain time correlation coefficients used to convert between ephemeris time (ET) and spacecraft clock time (SCLK)
    - » Sometimes an LSK is not needed in this conversion, but best to have it available as it is needed for other purposes as well
  - FK associates reference frames with CK IDs
    - » Needed if high level SPICE interfaces are used to access CK data (see next page)



What SPICE Routines Access CKs?

- **High-level SPICELIB routines are used more often than the "original" CK readers to access CK data. These high-level routines are:**
  - **Position or state transformation matrix determination**
    - **PXFORM: return a rotation matrix (3x3) from one frame to another, either of which can be a CK-based frame**
    - **SXFORM: return a state transformation matrix (6x6) from one frame to another, either of which can be a CK-based frame**
  - **Position or state vector vector determination**
    - **SPKPOS: return a position vector (1x3) in a specified frame, which can be a CK-based frame or have a CK-based frame as one of the "links" in its chain**
    - **SPKEZR: return a state vector (1x6) in a specified frame, which can be a CK-based frame or have a CK-based frame as one of the "links" in its chain**
  - **Use of the above mentioned routines is discussed in the frames (fk and using_frames) and spk tutorials**

- **The "original" CK access routines are CKGP and CKGPAV**
  - **Use of these routines is the subject of the remainder of this tutorial**

One Example of How To Read a CK File

Navigation and Ancillary Information Facility

Initialization ... typically once per program run

Tell your program which SPICE files to use ("loading" files)

```
CALL FURNSH( 'lsk_file_name' )
CALL FURNSH( 'sclk_file_name' )
CALL FURNSH( 'ck_file_name' )
```

Better yet, use a "furnsh kernel" to load all the needed files.

Loop ... do as often as you need

Convert UTC to SCLK ticks *

```
CALL STR2ET( 'utc_string', tdb )
CALL SCE2C ( spacecraft_id, tdb, sclkdp )
```

Get rotation matrix, or rotation matrix and angular velocity at requested time

```
CALL CKGP   (instid,sclkdp,tol,'ref_frame',cmat,   clkout,found)
```

or

```
CALL CKGPAV (instid,sclkdp,tol,'ref_frame',cmat,av,clkout,found)
```

inputs | outputs

* Although most spacecraft have a single on-board clock and this clock has the same ID as the spacecraft, the user should know which SCLK was used to tag data in a CK file to specify correct ID in a call to SCE2C.

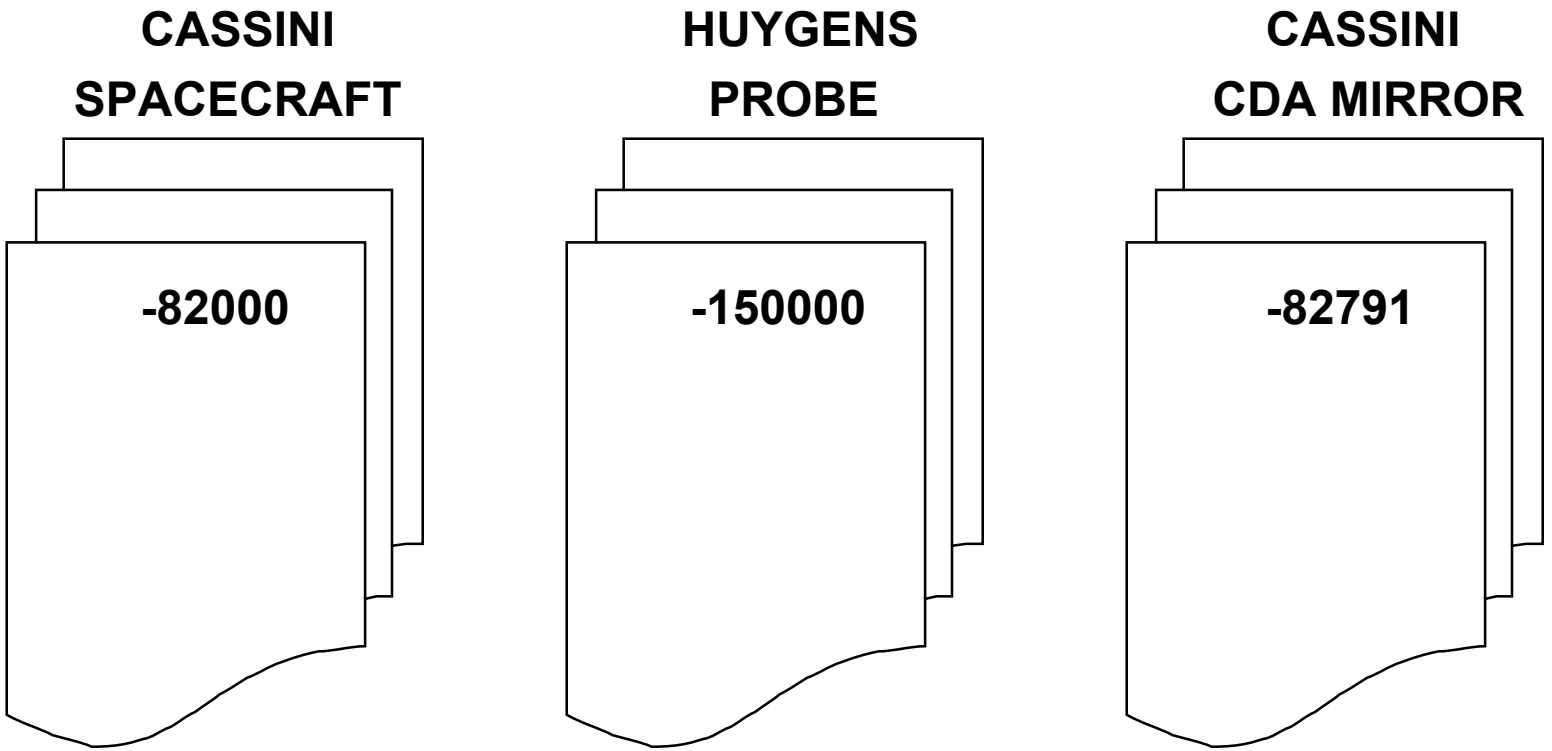
Arguments of CKGPAV

instid	NAIF ID for the spacecraft, instrument or other structure for which the orientation is to be returned
sclkdp	the time at which the orientation matrix and angular velocity are to be computed. The time system used is encoded spacecraft clock time (SCLK). The units are ticks since the beginning of the mission
tol*	the tolerance, expressed as number of SCLK ticks, to be used in searching for and computing the orientation data
ref_frame	the name of the reference frame with respect to which the orientation is to be computed
cmat	the 3x3 rotation matrix that you requested
av	the angular velocity that you requested
clkout	the exact time for which the orientation and angular velocity was computed
found	the logical flag indicating whether the orientation and angular velocity data were found. Note that if the loaded CK file(s) do not contain angular velocity data, CKGPAV will return a FALSE found flag even if orientation could have been computed. If "found" is .FALSE., then the values of the output arguments "cmat" and "av" are undefined and should not be used! ALWAYS CHECK THE FOUND FLAG RETURNED BY CKGPAV and CKGP!

* tol is explained in detail later on

An Example of Project CK Files

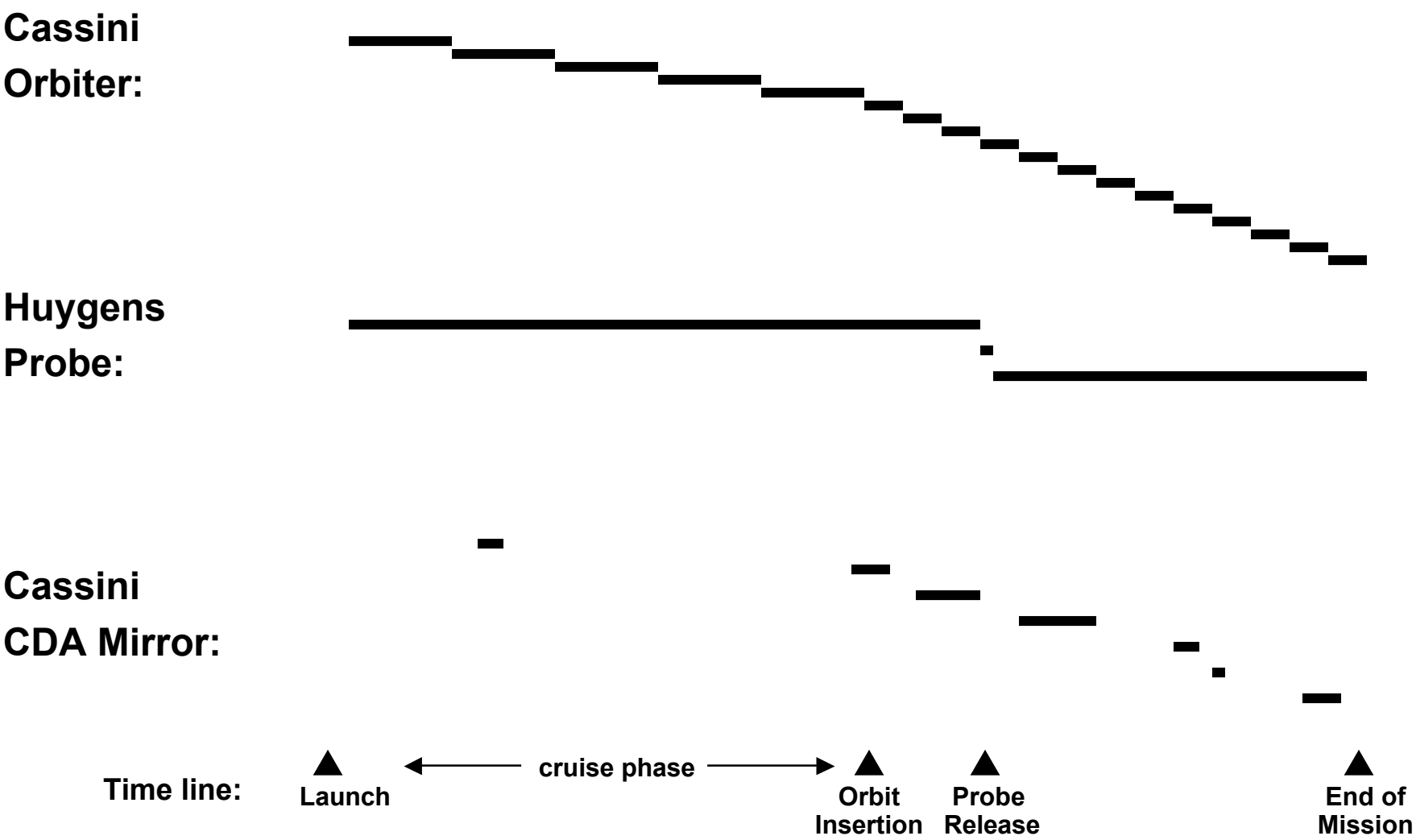
Navigation and Ancillary Information Facility



Users' programs must be able to load as many of these files as needed to satisfy one's requirements. It is strongly recommended that such programs have the flexibility to load a list of CK files provided to the program at run time; this is easily accomplished using the Toolkit's FURNSH routine.

Sample* CK File Coverage - 1

Navigation and Ancillary Information Facility



* Note: This may not be a bona fide Cassini/Huygens scenario; it is a highly simplified illustration of some of the possibilities for orientation delivery on a planetary mission.

Sample CK Data Coverage - 2

Navigation and Ancillary Information Facility

Even though a CK production and delivery schedule may suggest that CK files provide continuous coverage for the interval of time for which they were generated, in reality this is rarely the case -- CK files almost always contain gaps in coverage. Below is an example of this.

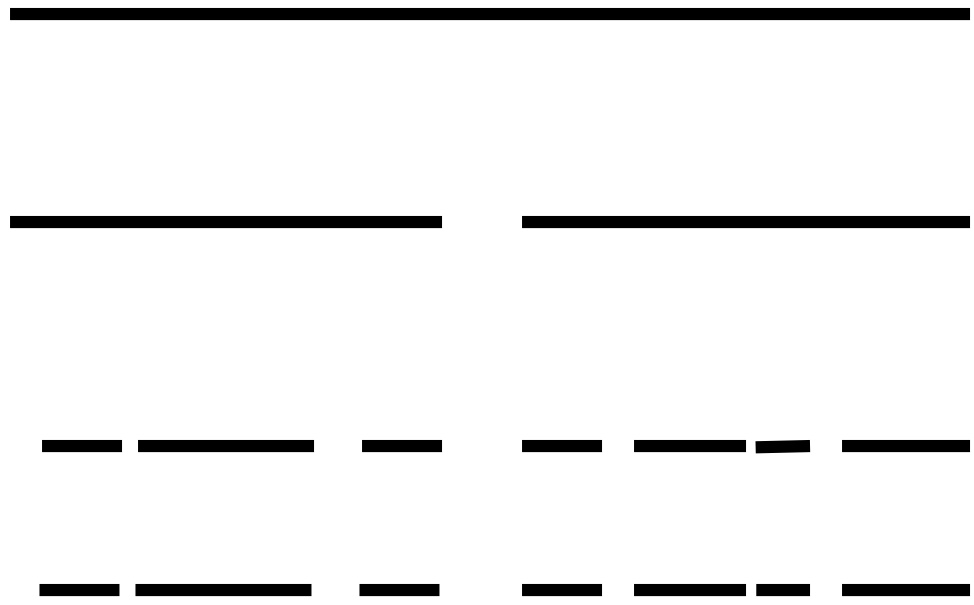
Coverage of …

a CK file:
as appears in file name/comments

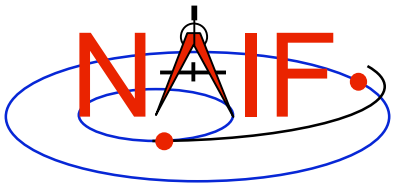
CK file segments:
as appears in ckbrief/spacit summary

Segments with "continuous" data:
(Types 2 - 5)

Summary of pointing available to the CK reader for this CK file:



Black lines represent
Interpolation intervals



What is an Interpolation interval?

- An interpolation interval is a time period for which the CK reader routines can compute and return pointing
  - For Types 2, 3, and 5 the pointing is computed by interpolating between the attitude data points that fall within the interval
  - For Type 4 the pointing is computed by evaluating polynomials covering the interval
  - The notion of an interpolation interval is not relevant to Type 1 (discrete pointing instances)
- The time periods between interpolation intervals are gaps during which the CK readers are not able to compute pointing
- The start and stop time for each interpolation interval is stored in a separate block within the CK segments, and can be modified without changing the data itself
  - The CKSPANIT and CKSMRG programs can be used to modify such information in type 3 CK segments

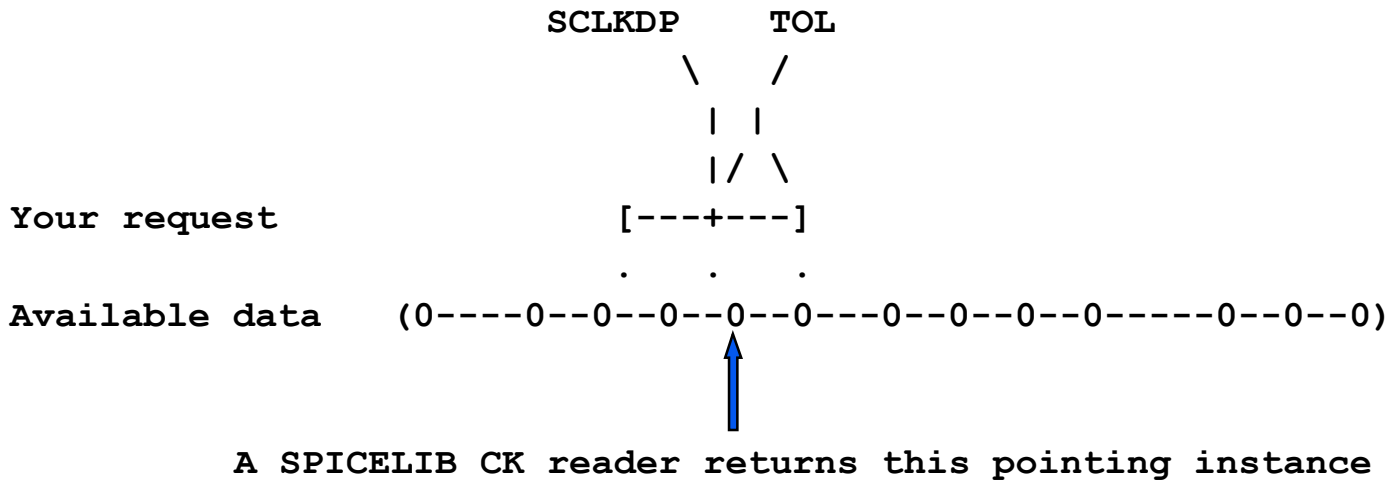
The Meaning of Tolerance - 1

- **The CKGP and CKGPAV routines employ a time tolerance "tol," measured in ticks, in executing pointing lookups.**
  - **No matter whether your CK is a discrete type (Type 1) or a continuous type (Types 2 - 5), if pointing information is not found within +/- tol of your pointing request time, no pointing will be returned.**
  - **For Type 1 (discrete) CKs, the pointing instance closest to your request time will be returned, as long as it is within tol of your request time.**
    - » **If the nearest pointing instances on each side of your request time are equidistant from your request time, the later instance will be selected.**
  - **For Types 2 - 5 (continuous pointing), pointing for exactly your request time will be returned if this time falls anywhere within an interpolation interval.**
  - **For all Types, the time tag associated with the pointing data will also be returned**

The Meaning of Tolerance - 2

Navigation and Ancillary Information Facility

When reading a Type 1 CK containing discrete pointing instances

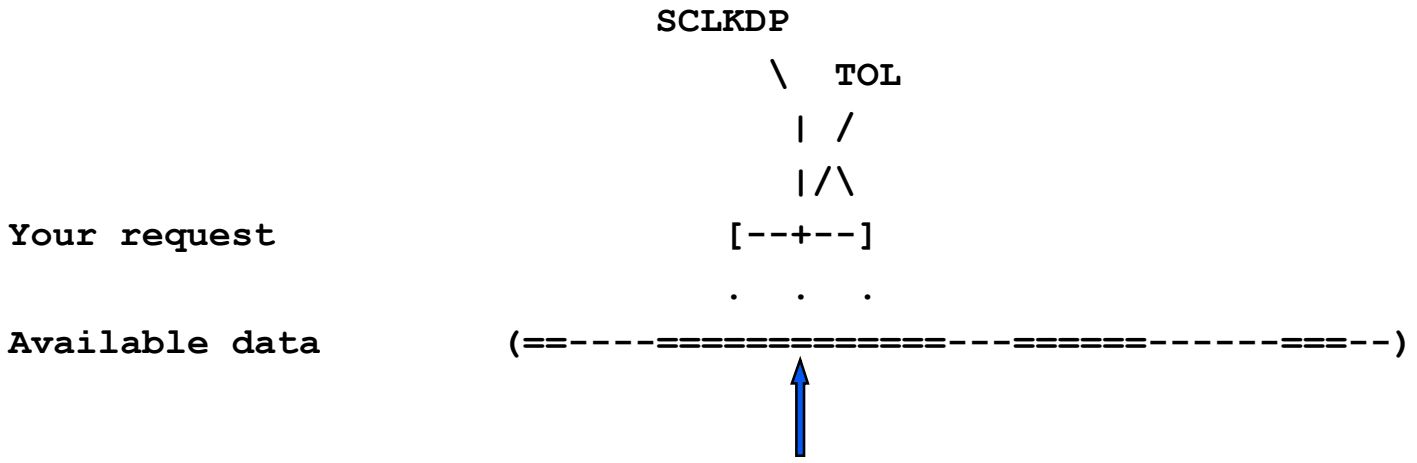


- "0" is used to represent discrete pointing instances (quaternions)
- "()" are used to represent the end points of a segment within a CK file
- SCLKDP is the time at which you have requested pointing
- TOL is the time tolerance you have specified in your pointing request
- The quaternions occurring in the segment need not be evenly spaced in time

The Meaning of Tolerance - 3

Navigation and Ancillary Information Facility

When reading a Type 2, 3, 4 or 5 CK (continuous pointing), with a "pointing request" that falls within a span of continuous pointing (an "interpolation interval")



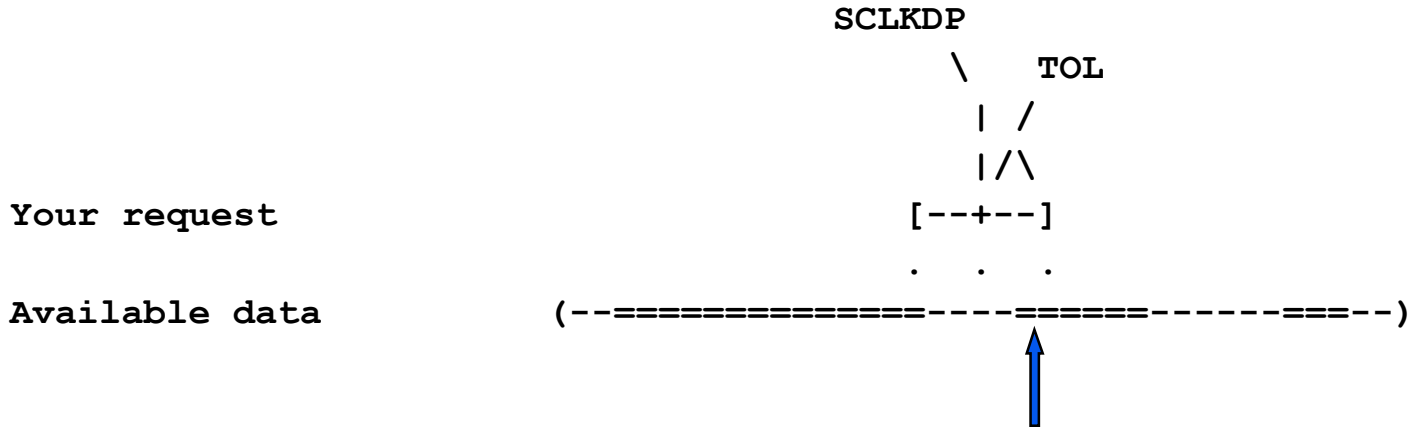
A SPICELIB CK reader returns pointing at precisely the requested epoch

- "==" is used to indicate interpolation intervals of continuous pointing
- "()" are used to represent the end points of a segment within a CK file
- SCLKDP is the time at which you have requested pointing
- TOL is the time tolerance you have specified in your pointing request; for this particular case it does not come into play
- The quaternions occurring in the periods of continuous pointing need not be evenly spaced in time

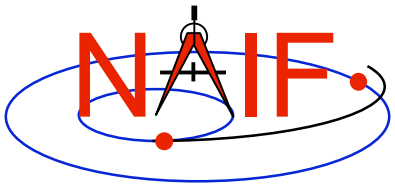
The Meaning of Tolerance - 4

Navigation and Ancillary Information Facility

When reading a Type 2, 3, 4 or 5 CK (continuous pointing), with a "pointing request" that is NOT within a span of continuous pointing (an "interpolation interval")



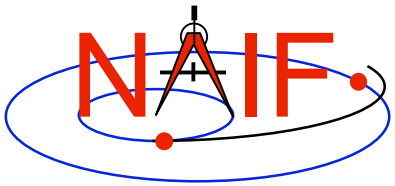
- "==" is used to indicate interpolation intervals of continuous pointing
- "()" are used to represent the end points of a segment within a CK file
- SCLKDP is the time at which you have requested pointing
- TOL is the time tolerance you have specified in your pointing request
- The quaternions occurring in the periods of continuous pointing need not be evenly spaced in time



Summarizing a CK file - 1

- A brief summary of a CK can be made using the Toolkit utility CKBRIEF
  - At your command prompt, type the program name followed by the names of CK, LSK and SCLK files (in that order)

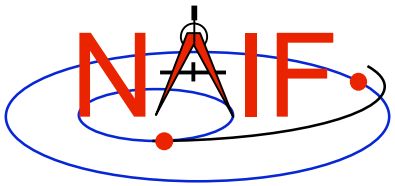
```
% ckbrief cas_050215.bc naif0007.tls cassini.tsc

CKBRIEF Version: 1.0.0. SPICE Toolkit Version: N0050.

Summary for: cas_050215.bc

Object:  -82000
  Interval Begin ET         Interval End ET           AV
  ------------------------- ------------------------- ---
  2005-FEB-15 08:01:04.183 2005-FEB-15 09:01:04.183 Y
```



Summarizing a CK file - 2

• A detailed summary of a CK can be made using the Toolkit utility SPACIT. See the SPACIT User's Guide for details.

```
------------------------------------------------------------
   Segment ID      : CASSINI ATT: -Y TO TITAN, +Z - VELCTY
   Instrument Code: -82000
   Spacecraft      : Body -82, CAS
   Reference Frame: Frame 1, J2000
   CK Data Type    : Type 3
      Description  : Continuous Pointing: Linear Interpolation
   Available Data  : Pointing and Angular Velocity
   UTC Start Time  : 2005 FEB 15 07:59:59.999
   UTC Stop Time   : 2005 FEB 15 08:59:59.998
   SCLK Start Time: 1/1487147147.203
   SCLK Stop Time  : 1/1487150747.209
------------------------------------------------------------
```



Obtaining Coverage of CK File

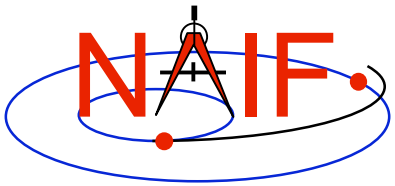
- A high-level subroutine interface for obtaining CK coverage information was introduced in the N0058 version of the SPICE Toolkit.
  - Call CKOBJ to find the set of structures for which a specified CK provides data.
    - » INPUT: an CK file name and initialized SPICE integer "Set" data structure. The set may optionally contain ID codes obtained from previous calls.
    - » OUTPUT: the input set, to which have been added (via set union) the ID codes of structures for which the specified CK provides coverage.

```
CALL CKOBJ ( CK, IDS )
```

  - Call CKCOV to find the window of times for which a specified CK file provides coverage for a specified structure:
    - » INPUT: a CK file name, structure ID code, flag indicating whether angular rates are needed, flag indicating whether coverage on segment or interval level is to be returned, tolerance, output time system, and initialized SPICE double precision "Window" data structure. The window may optionally contain coverage data from previous calls.
    - » OUTPUT: the input window, to which have been added (via window union) the sequence of start and stop times of coverage intervals of the specified CK, expressed as encoded SCLK or ET seconds past J2000.

```
CALL CKCOV ( CK, ID, NEEDAV, LEVEL, TOL, TIMSYS, COVER)
```

- See the headers of these routines for example programs.
- Also see the CELLS, SETS and WINDOWS Required Reading for background information on these SPICE data types.



Additional Information on CK

- **For more information about CK, look at the following documents**
  - CK Required Reading
  - headers for the CKGP and CKGPAV routines
  - Most Useful SPICELIB Routines
  - CKBRIEF and SPACIT User's Guides
  - Frames tutorials: FK and Using Frames ← *don't miss these*
  - Porting_kernels tutorial
- **Related documents**
  - SCLK Required Reading
  - Time Required Reading
  - Frames Required Reading
  - NAIF_IDS Required Reading
  - Rotations Required Reading

Backup

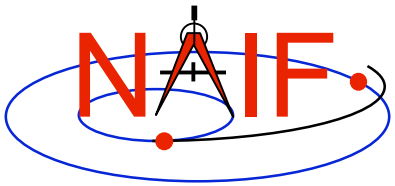
Examples of Problems Encountered When Using the CK Subsystem

Problems using CK - 1

- **The file, or files, you loaded do not contain orientation data for the object of interest.**
  - **Make sure the ID that you use in a call to CKGP or CKGPAV matches one in the CK file(s) you have loaded, or the frame name that you use in a call to SXFORM, PXFORM, SPKEZR, or SPKPOS is associated with one available in the loaded CK files**
- **CKGP or CKGPAV returns a transformation matrix and/or angular velocity that does not appear correct.**
  - **Probably the FOUND flag is "FALSE" and you are using data left over from a previous query. Remember to always check the FOUND flag! (If the FOUND flag is "TRUE" but the data seem bad, contact the data producer.)**

Problems using CK - 2

- **The file, or files, you loaded do not cover the time at which you requested orientation**
  - **Check file coverage on the segment level by summarizing the file(s) using CKBRIEF or SPACIT**
  - **Check interpolation interval coverage using CKBRIEF with option "-dump" (Toolkit version N0052 or later) or by examining comments provided in the comment area of the file - you may be asking for data within a coverage gap (outside of interpolation intervals)**
- **One of the frames routines (SPKEZR/SPKEZ, SPKPOS, SXFORM, PXFORM) signals an error**
  - **All frames routines read CK files using tolerance = 0**
    - » **For discrete CKs (Type 1) the orientation of a CK-based frame will be computed only if the time provided to a Frames routine exactly matches one of the times stored in the CK file; otherwise an error will be signaled.**
    - » **For continuous CKs (all but Type 1) the orientation of a CK-based frame will be computed only if the time provided to a frames routine falls within one of the interpolation intervals defined by the CK file; otherwise an error will be signaled.**

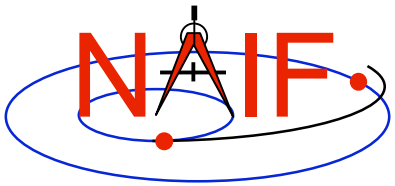


Problems using CK - 3

- **You've confirmed not having any of the previously described problems, but the FOUND flag comes back "FALSE" when using CKGPAV, or SXFORM or SPKEZR signals a frame related error.**
  - **You are using a SPICE routine that expects angular velocity as well as orientation, but the CK segments available at your requested epoch don't contain angular velocity.**
    - » **Routines expecting AV are: CKGPAV, SXFORM, SPKEZR**
    - » **Routines not expecting AV are: CKGP, PXFORM, SPKPOS**

Problems using CK - 4

- **The FOUND flag comes back "TRUE" when using CKGPAV but returned angular velocity does not appear correct.**
  - **While many other sources of the angular rate data, for example spacecraft telemetry, specify it relative to the spacecraft frame, SPICE CK files store it and CKGPAV returns it with respect to the base frame, so CK style may be unexpected.**
- **The FOUND flag comes back "TRUE" when using CKGP/CKPGAV but the quaternion computed from returned transformation matrix via a call to M2Q does not appear correct.**
  - **The quaternion returned by M2Q follows SPICE-style which is different from the quaternion styles used by some other sources of orientation data, for example spacecraft telemetry.**
    - **See headers of M2Q and Q2M routines and Rotations Required reading document for more details.**
    - **NAIF also prepared and can provide a "white paper" explaining differences between various quaternion styles commonly used in space applications.**



Problems using CK - 5

- **You're trying to use a CK file that is not properly formatted for your computer**
  - **You can read only a binary CK file with the CK subroutines; you can't read a "transfer format" file**
    - » **Although not required, binary CK files often have a name like "xxxxxx.bc"**
    - » **Although not required, transfer format CK files often have a name like "xxxxxx.xc" (formerly "xxxxxx.tc")**
  - **If using Toolkit Version N0051 or earlier you must have the proper kind of CK binary file for your computer (a native binary file)**
    - » **Sun, HP, SGI, MAC and NeXT all use the same (unix) binary standard**
    - » **DEC Alpha/Digital Unix and PC use the same binary standard**
    - » **DEC Alpha/VMS has its own binary standard**
    - » **The pair of utility programs named TOBIN and TOXFR, or the utility program SPACIT, can be used to port CK files between computers having incompatible binary standards**