“Camera-matrix” Kernel
CK

(Orientation or Attitude Kernel)

Emphasis on reading CK files

April 2016

CK File Contents - 1

- **A CK file holds orientation data for a spacecraft or a moving structure on the spacecraft**
  - "Orientation data" ⇒ quaternions, from which orientation matrices are formed by SPICE software. These matrices are used to rotate position vectors from a base reference frame (the "from" frame) into a second reference frame (the "to" frame)
    - » In SPICE this is often called the "C-matrix or "Camera matrix"
  - Optionally may include angular velocity of the "to" frame with respect to the "from" frame
    - » Angular velocity vectors are expressed relative to the "from" frame.

- **A CK file should also contain comments–sometimes called metadata–that provide some details about the CK such as:**
  - the purpose for this particular CK
  - when and how it was made
  - what time span(s) the data cover

CK File Contents - 2

- A single CK file can hold orientation data for 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

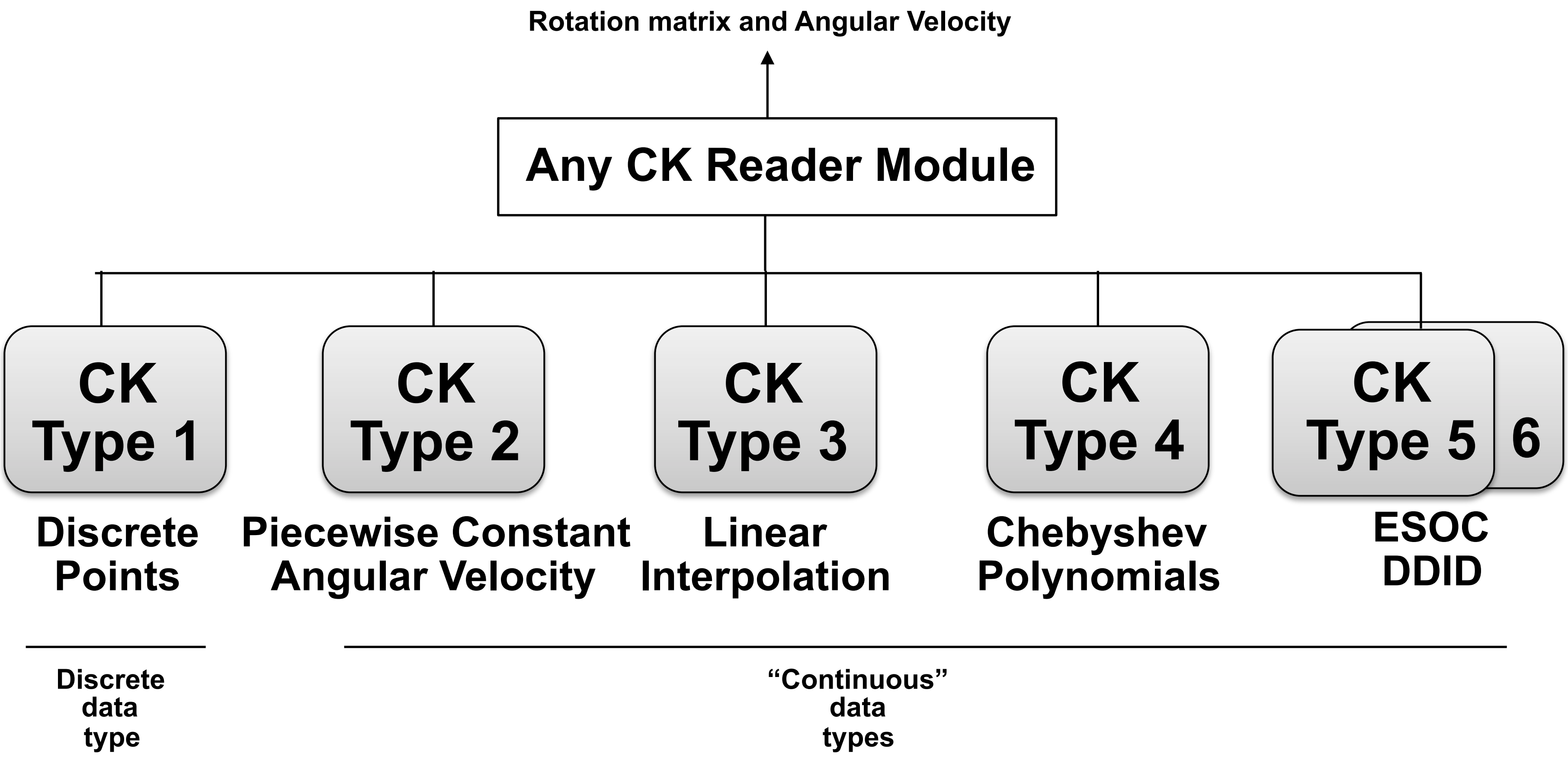
- In most cases CKs contain data for just one structure

CK File Varieties

- “Reconstruction” CK (also called “definitive” CK)
  - Made from orientation telemetry returned from a spacecraft or other structure
- “Predict” CK
  - 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
- Knowledge of CK variety might be implicit in the file naming schema, and/or might be provided in the comment section, but is not available using a SPICE API
- It is inadvisable that both “reconstruction” and “predict” data be combined in a single file

CK Data Types Concept

The underlying orientation data are of varying types, but the user interface to each of these CK types is the same.



Rotation matrix and Angular Velocity

Any CK Reader Module

CK Type 1

CK Type 2

CK Type 3

CK Type 4

CK Type 5

6

Discrete Points

Piecewise Constant Angular Velocity

Linear Interpolation

Chebyshev Polynomials

ESOC DDID

Discrete data type

“Continuous” data types

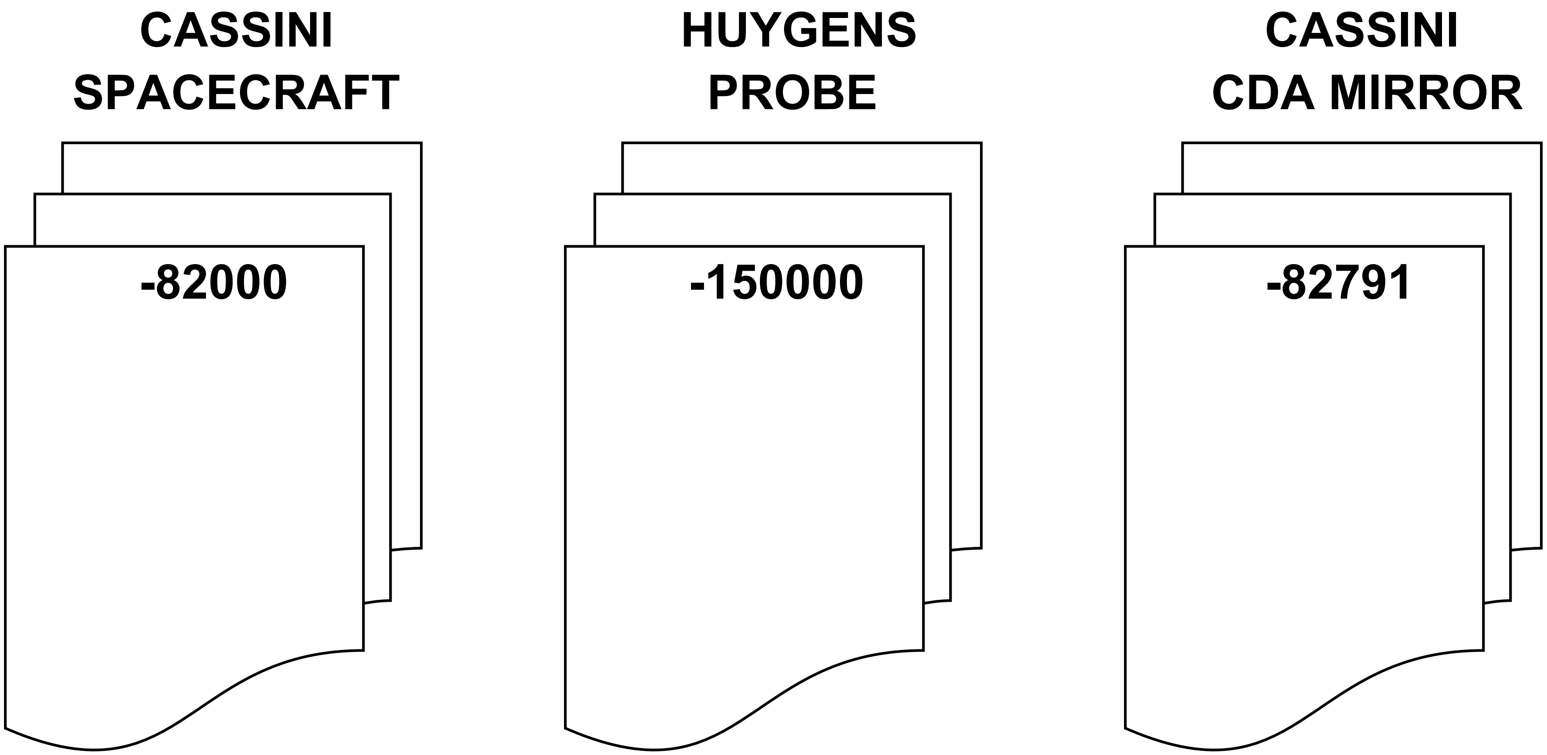
Kernel Data needed

- To obtain orientation one needs at least three SPICE kernel types: CK, SCLK, and LSK.
  - 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 it's best to have it available as it is usually needed for other purposes
- One may also need an FK if planning to access CK data via high level SPICE interfaces.
  - FK associates reference frames with CK data via CK IDs

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, PXFRM2: return a rotation matrix (3x3) from one frame to another, either of which can be a CK-based frame or have CK-based frames as "links" in its chain
    - SXFORM: return a state transformation matrix (6x6) from one frame to another, either of which can be a CK-based frame or have CK-based frames as "links" in its chain
  - Position or state vector determination
    - SPKPOS: return a position vector (3x1) in a specified frame, which can be a CK-based frame or have CK-based frames as "links" in its chain
    - SPKEZR: return a state vector (6x1) in a specified frame, which can be a CK-based frame or have CK-based frames as "links" in its chain
- Use of the above mentioned routines is discussed in the FK, Using Frames, and SPK tutorials
- The "original" CK access routines are CKGP and CKGPAV
  - Use of these routines is NOT described in this class tutorial: see the on-line version for those details

An Example of Project CK Files



CASSINI SPACECRAFT

-82000

HUYGENS PROBE

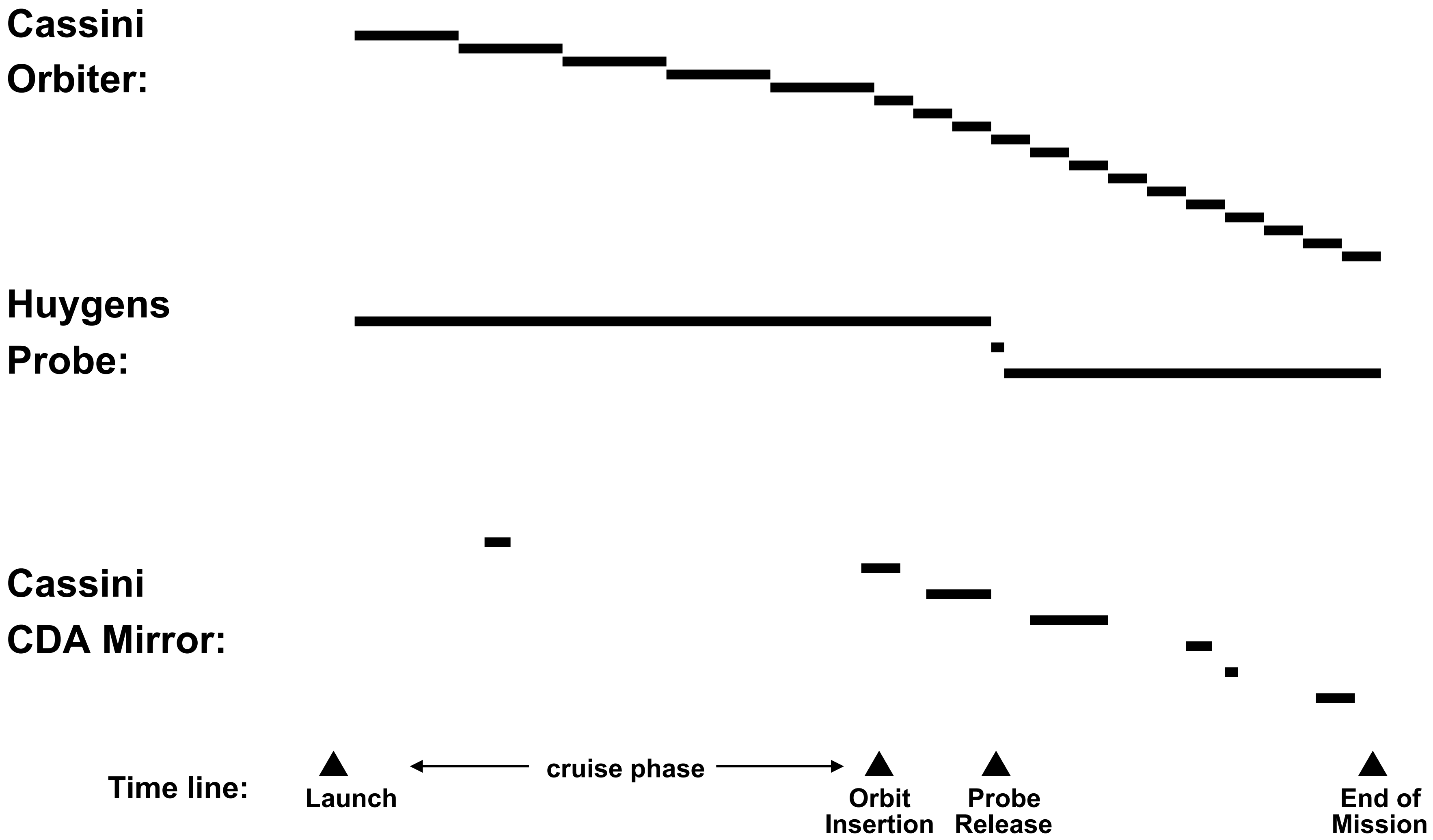
-150000

CASSINI CDA MIRROR

-82791

A user’s program must be able to load as many of these files as needed to satisfy his/her requirements. It is strongly recommended that user’s 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



Cassini Orbiter:

Huygens Probe:

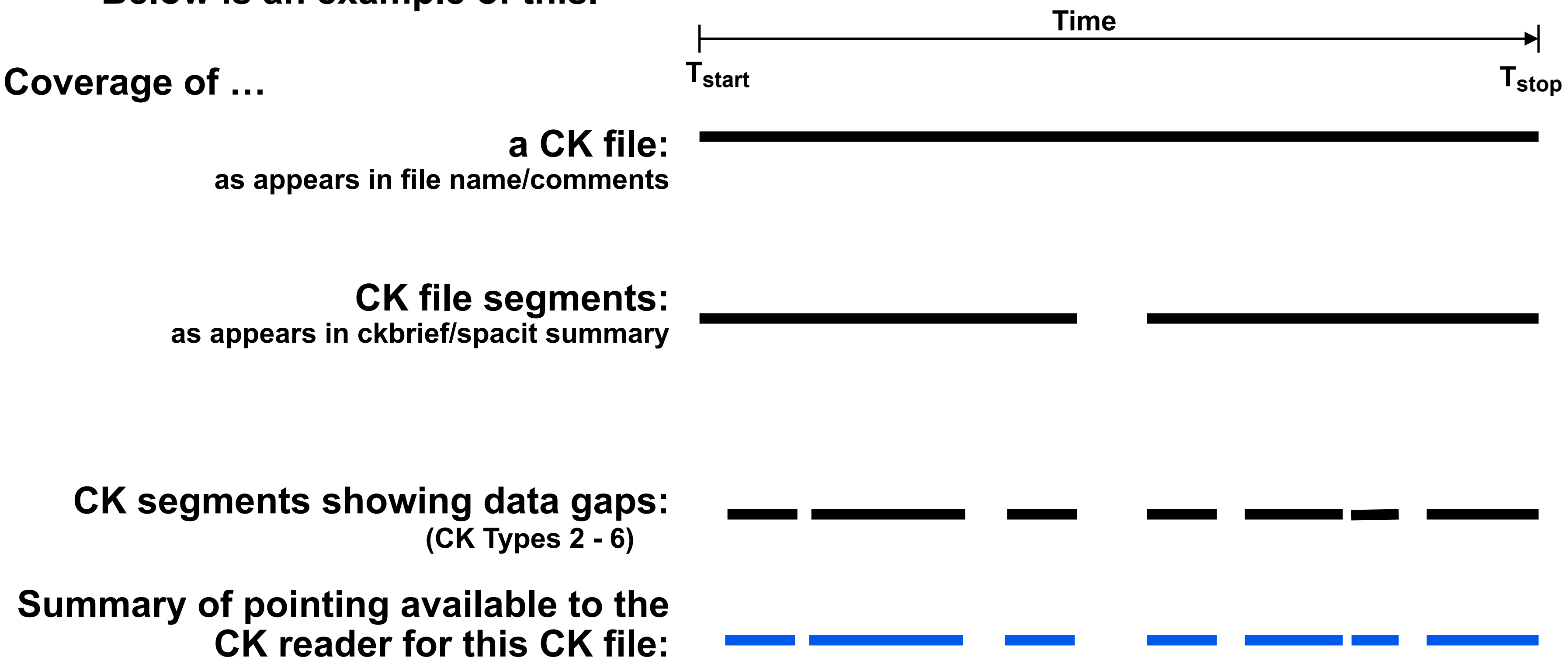
Cassini CDA Mirror:

Time line: Launch ← cruise phase → Orbit Insertion Probe Release End of Mission

* Note: This is not an actual 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

Even though a project's CK production process 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.


Time
T_{start}
T_{stop}
Coverage of …
a CK file:
as appears in file name/comments
CK file segments:
as appears in ckbrief/spacit summary
CK segments showing data gaps:
(CK Types 2 - 6)
Summary of pointing available to the
CK reader for this CK file:

Blue line segments represent interpolation intervals–times when pointing will be returned and the FOUND flag set to “TRUE.”

What is an Interpolation interval?

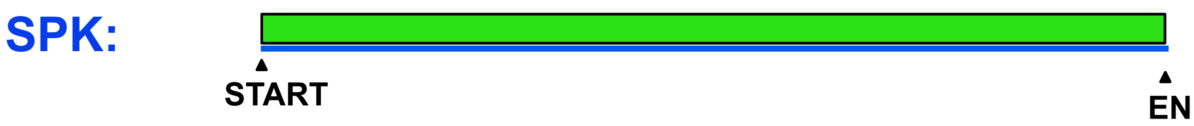
- An interpolation interval is a time period for which the CK reader routines can compute and return pointing.
  - For CK Types 3, 5 and 6 the pointing is computed by interpolating between the attitude data points that fall within the interval.
  - For CK Type 2 the pointing within each interval is computed by extrapolating from a single attitude and associated angular velocity.
  - For CK Type 4 the pointing is computed by evaluating polynomials covering the interval.
  - For CK Type 1 (discrete pointing instances) the notion of an interpolation interval is not relevant.

- The time periods between interpolation intervals are gaps during which the CK readers are not able to compute pointing.

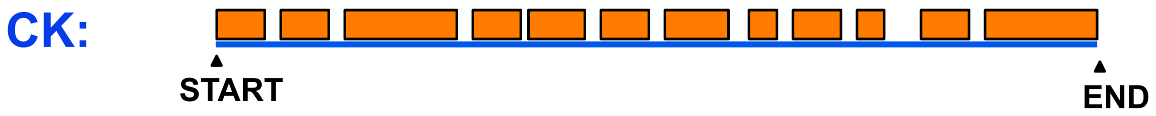
- The interpolation intervals in Type 3 CK segments can be modified without changing the actual pointing data.
  - The CKSPANIT and CKSMRG programs are used to make these changes.

Coverage - Contrast CK with SPK

- SPKs usually have continuous coverage for the objects included
  - You can get a result (an answer) at any time



- CKs often have gaps, sometimes very many!
  - Due to missing or bad telemetry packets from the spacecraft



- Any "high-level" SPICE API that needs to use a CK to help determine a reference frame orientation will NOT return a result for any time falling in a gap!
  - The orange bars above are called "interpolation intervals"
  - The white spaces between the orange bars are gaps

Determine Coverage Using APIs

- When using high-level routines* that need orientation data from a C-kernel, it's often a good idea to first determine what are the valid interpolation intervals in your CKs using the CKOBJ and CKCOV APIs.

- Then check each time of interest for your geometry calculations against the window of valid intervals before proceeding onwards.

* For example: SPKEZR, SPKPOS, SXFORM, PXFORM, SINCPT

Determine Coverage Using Utilities

- Three Toolkit utility programs can provide various kinds of CK summaries, including listings of gaps or of interpolation intervals
  - CKBRIEF
  - FRMDIFF
  - SPACIT

Summarizing a CK with CKBRIEF

- A summary of interpolation intervals in a CK can be made using CKBRIEF using its ‘-dump’ option

```
% ckbrief -dump 07102_07107ra.bc naif0008.tls cas00106.tsc

CKBRIEF Ver 3.2.0, 2006-11-02. SPICE Toolkit Version: N0061.

Summary for: 07102_07107ra.bc

Segment No.: 1

Object:  -82000
  Interval Begin ET        Interval End ET          AV
  ------------------------ ------------------------ ---
  2007-APR-12 00:01:06.462 2007-APR-12 05:58:02.576 Y
  2007-APR-12 05:58:22.576 2007-APR-12 21:34:26.221 Y
  . . .
```

continued on next page

Summarizing a CK with FRMDIFF

continued from previous page

- A summary of interpolation intervals in a CK can also be made using FRMDIFF with its ‘-t dumpc’ option
- A summary of gaps within interpolation intervals in a CK can be made using FRMDIFF with its ‘-t dumpg’ option

```
% frmdiff -t dumpg \
          -k cas_v40.tf naif0008.tls cas00106.tsc \
          -f 'YYYY-DOYTHR:MN:SC ::RND’ \
          07102_07107ra.bc
#
# . . . <FRMDIFF report header> . . .
#
# gap_start, gap_stop, gap_duration_sec, gap_duration_string
2007-102T05:56:57 2007-102T05:57:17  19.999   0:00:00:19.999
2007-102T21:33:21 2007-102T21:33:41  19.999   0:00:00:19.999
2007-102T21:34:57 2007-102T21:35:25  27.999   0:00:00:27.999
. . .
```

CK Utility Programs

- The following CK utility programs are included in the Toolkit:

CKBRIEF	summarizes coverage for one or more CK files
SPACIT	generates segment-by-segment summary of a CK file
COMMNT	reads, appends, or deletes comments in an CK file
MSOPCK	converts attitude data provided in a text file into a CK file
FRMDIFF	samples or compares orientation of CK-based frames

- These additional CK utility programs are provided on the NAIF Web site

CKSLICER	subsets a CK file
CKSMRG	merges segments in a type 3 CK file (*)
DAFCAT	concatenates together CK files (*)
CKSPANIT	modifies interpolation interval information in a Type 3 CK file
DAFMOD	alters structure or frame IDs in a CK file
PREDICKT	creates a CK file representing an orientation profile described by a set of orientation rules and a schedule
BFF	displays binary file format of a CK file
BINGO	converts CK files between IEEE and PC binary formats

(*) DAFCAT and SKSMRG are frequently used together to first merge many CK files into a single file using DAFCAT and then merge segments inside the merged file using CKSMRG.

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 FRMDIFF 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