Frames Kernel
FK

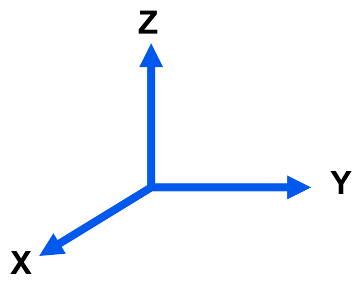
April 2016

Background

- First, reminders of what SPICE means by:
  - reference frame
  - coordinate system

What is a Reference Frame?

- Within SPICE, a reference frame is specified by three orthogonal axes

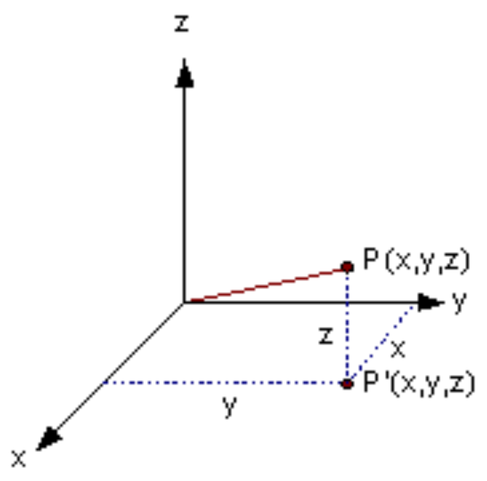


- It may be fixed in space (not rotating, not accelerating)
  - This is called an inertial frame
- It may be changing its orientation in space
  - This is called a non-inertial frame
- Every frame has a name assigned to it
  - You'll use these names as arguments in calling Toolkit APIs
- Every frame has an associated center location

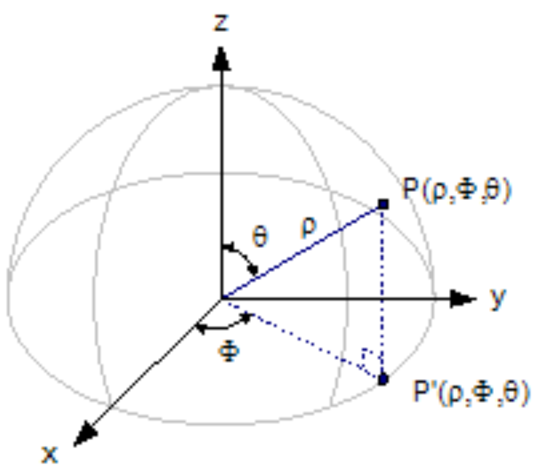
What is a Coordinate System?

Navigation and Ancillary Information Facility

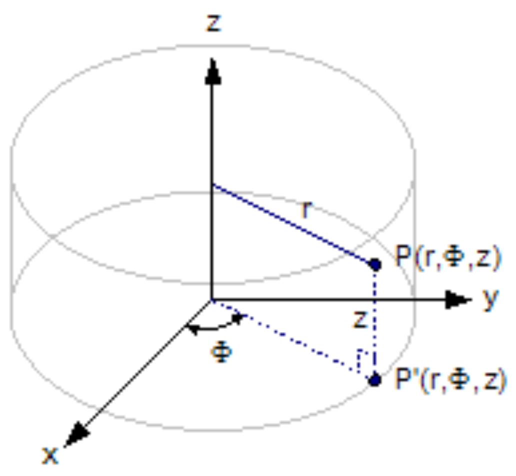
- **Within SPICE, a coordinate system is the method used to specify a vector within a reference frame. Examples:**



Rectangular



Spherical



Cylindrical

Introduction

Navigation and Ancillary Information Facility

What does the FRAMES subsystem do?

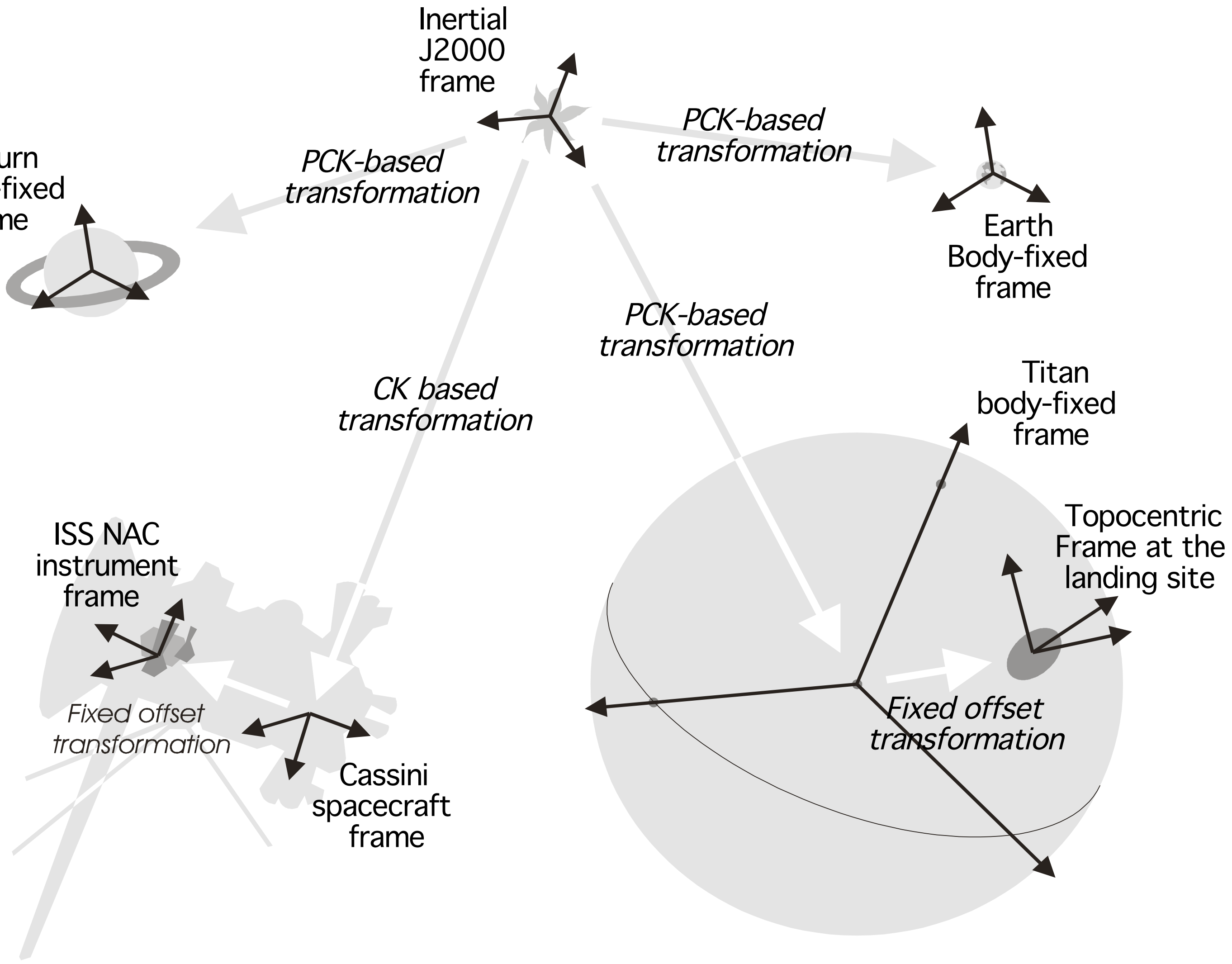
1. It establishes relationships between reference frames used in geometry computations – it "chains frames together" in a frame tree.

2. It connects frames with the sources of their orientation specifications. In some cases those data are included in the Frames kernel itself.

Based on these relationships and the orientation source information, the frames subsystem allows SPICE software to compute rotations between neighboring frames in the frame tree, and to combine these rotations in the right order, thus providing an ability to compute the orientation of any frame in the tree with respect to any other frame in the tree, at any time.

Sample Frame Tree

Navigation and Ancillary Information Facility



Frame Names

- Frame names are character strings used to identify frames to Toolkit APIs
- Examples of frame names:
  - J2000
  - IAU_MARS
  - DAWN_SPACECRAFT
  - MEX_OMEGA
  - DSS-14_TOPO

Where to Find the Names of Frames?

- Refer to the “NAIF IDs” Tutorial for an introduction to reference frame names and IDs
- Refer to the FRAMES.REQ document for the list of NAIF “built in” (hard coded) inertial and body-fixed frames
- Refer to a project’s Frames Kernel (FK) file for a list of frames defined for the spacecraft, its subsystems and instruments
- Refer to an earth stations FK for a list of frames defined for the DSN and other stations
- Refer to the moon FKs for names and descriptions of the body-fixed frames defined for the moon

Frame Class Specifications

Navigation and Ancillary Information Facility

Frame class	***Frame Defined in:***	***Orientation data provided in:***
Inertial	**Toolkit software**	**Toolkit software**
Body-fixed	**Toolkit software or FK**	**PCK**
CK based	**FK**	**CK**
Fixed offset	**FK**	**FK**
Dynamic	**FK**	**Toolkit software, or computed using FK, SPK, CK, and/or PCK**

Examples of Frame Classes

Frame class	***Frame Examples (with real frame names)***
Inertial	• ICRF (called J2000 in SPICE) • Planet Equator/Equinox of Epoch (MARSIAU, ...) • Ecliptic of Epoch (ECLIPJ2000, ...)
Body-fixed	• Solar system body IAU frames (IAU_MARS, IAU_SATURN, …) • High accuracy Earth frames (ITRF93, …) • High accuracy Moon frames (MOON_PA, MOON_ME)
CK-based	• Spacecraft (CASSINI_SC_BUS, …) • Moving parts of an instrument (MPL_RA_JOINT1, ...)
Fixed Offset	• Instrument mounting alignment (CASSINI_ISS_NAC, …) • Topocentric (DSS-14_TOPO, …)
Dynamic	• Geomagnetic • Geocentric Solar Equatorial • Planet true equator and equinox of date

Frames Kernel File Overview

Navigation and Ancillary Information Facility

- Uses the SPICE text kernel standards
- Loaded using the FURNSH routine
- Usually contains comprehensive information about the defined frames in the comment section(s) of the file
- Contains frame definition information consisting of a set of assignments in the data sections of the file. Below are examples defining a CK-based frame and a fixed-offset frame.

CK-based Frame Example

```
FRAME_DAWN_SPACECRAFT   = -203000
FRAME_-203000_NAME      = 'DAWN_SPACECRAFT'
FRAME_-203000_CLASS     =  3
FRAME_-203000_CLASS_ID  = -203000
FRAME_-203000_CENTER    = -203
CK_-203000_SCLK         = -203
CK_-203000_SPK          = -203
```

Fixed-offset Frame Example

```
FRAME_DAWN_FC1              =  -203110
FRAME_-203110_NAME          = 'DAWN_FC1'
FRAME_-203110_CLASS         =  4
FRAME_-203110_CLASS_ID      =  -203110
FRAME_-203110_CENTER        =  -203
TKFRAME_-203110_RELATIVE    = 'DAWN_SPACECRAFT'
TKFRAME_-203110_SPEC        = 'ANGLES'
TKFRAME_-203110_UNITS       = 'DEGREES'
TKFRAME_-203110_ANGLES      = ( 0.0, 0.0, 0.0 )
TKFRAME_-203110_AXES        = ( 1,   2,   3   )
```

- See the on-line FK tutorial for details about these assignments

APIs Using the Frames Subsystem

Navigation and Ancillary Information Facility

SXFORM, PXFORM, and PXFRM2	**returns state or position transformation matrix**

```
CALL SXFORM( 'FROM_FRAME_NAME', 'TO_FRAME_NAME', ET,            MAT6x6 )
CALL PXFORM( 'FROM_FRAME_NAME', 'TO_FRAME_NAME', ET,            MAT3X3 )
CALL PXFRM2( 'FROM_FRAME_NAME', 'TO_FRAME_NAME', ETFROM, ETTO, MAT3X3 )
```

SPKEZR SPKPOS	**returns state or position vector in specified frame**

```
CALL SPKEZR( BOD, ET, 'FRAME_NAME', CORR, OBS, STATE,  LT )
CALL SPKPOS( BOD, ET, 'FRAME_NAME', CORR, OBS, POSITN, LT )
```

The above are FORTRAN examples, using SPICELIB modules.

The same interfaces exist for C, using CSPICE modules, and for IDL (Icy) and MATLAB (Mice).

CK-Based Frames "Must Know"

Navigation and Ancillary Information Facility

These are VERY IMPORTANT points you must understand!

- **The frames routines (SPKEZR, SPKPOS, SXFORM, PXFORM) all 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 (Types 2 – 6) 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.
- **Using SPKEZR or SXFORM requires CKs that contain angular rate data**
  - Since these routines return a state vector (6x1) or state transformation matrix (6x6), angular rate must be present in the CK in order to compute vectors and matrices; if angular rate is not present an error will be signaled.
  - SPKPOS and PXFORM, which return a position vector (3x1) and a position transformation matrix (3x3) respectively, can be used when angular rate data are NOT present in a CK.
- **Ephemeris time input to Frames routines is converted to SCLK to access CKs**
  - SCLK and LSK kernels must be loaded to support this conversion.
  - The SCLK ID is specified in one of the CK frame definition keywords; if not, it's assumed to be the Frame ID divided by 1000.

Frame Tree Example: ASPERA Instrument on Mars Express

Navigation and Ancillary Information Facility

```
                               "J2000" <-inertial
                                   |
       +------------------------------------------------------+
       |                           |                          |
       |<-pck                      |                          |<-pck
       |                           |                          |
       V                           |                          V
   "IAU_MARS"                      |                     "IAU_EARTH"
 MARS  BODY-FIXED                  |<-ck               EARTH  BODY-FIXED
 ---------------                   |                   ----------------
                                   V
                           "MEX_SPACECRAFT"
       +------------------------------------------------------+
       |                                                      |
       |<-fixed                                               |<-fixed
       |                                                      |
       V                                                      V
"MEX_ASPERA_URF"                                     "MEX_ASPERA_IMA_URF"
-----------------                                    --------------------
       |                                                      |
       |<-ck                                                  |<-fixed
       |                                                      |
       V                                                      V
"MEX_ASPERA_SAF"                                       "MEX_ASPERA_IMA"
---------------------------------------------  ...     ----------------
       |                 |                  |
       |<-fixed          |<-fixed           |<-fixed
       |                 |                  |
       V                 V                  V
"MEX_ASPERA_ELS"   "MEX_ASPERA_NPI"   "MEX_ASPERA_NPD1"
----------------   ----------------   ----------------
```

Blue text indicates frame class

Navigation and Ancillary Information Facility

FK Utility Programs

- The following FK and frames utility programs are included in the Toolkit

FRMDIFF	samples orientation of a frame or compares orientation of two frames
CKBRIEF	summarizes coverage for one or more CK files
BRIEF	summarizes coverage for one or more binary PCK files

- These additional FK and frames utility programs are provided on the NAIF Web site

PINPOINT	creates SPK and topocentric frames FK files for fixed locations (ground stations, etc)
BINGO	converts FK files between UNIX and DOS text formats

Additional Information on FK

- For more information about FK and frames, look at the following documents
  - Frames Required Reading
  - Using Frames Tutorial
  - Dynamic Frames Tutorial
  - NAIF IDs Tutorial
  - Headers for the routines mentioned in this tutorial
  - Most Useful SPICELIB Routines
  - FRMDIFF User's Guide
  - Porting Kernels tutorial
- Related documents
  - CK Required Reading
  - PCK Required Reading
  - SPK Required Reading
  - Rotations Required Reading