Introduction to SPICE

Spacecraft Guidance and Navigation

AY 2021-2022

POLITECNICO
MILANO 1863

About me

Alessandro Morselli

Space Engineer

alessandro.morselli@polimi.it

Skills

- Satellites and operations
- Modelling and simulations
- Software development
- Testing and validation

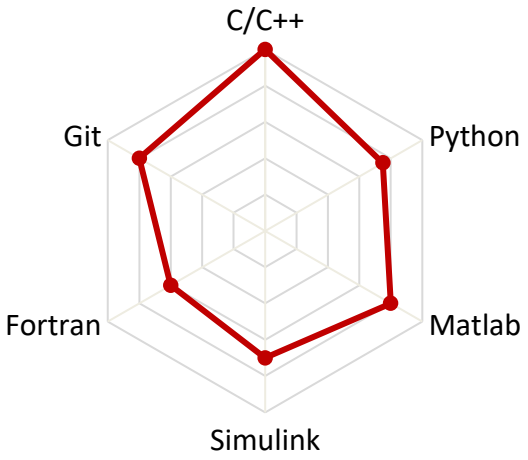
Education

- *2011-2014*
  PhD in Aerospace Engineering
  Politecnico di Milano
- *2008-2011*
  MSc in Space Engineering
  Politecnico di Milano
- *2005-2008*
  BSc in Aerospace Engineering
  Politecnico di Milano

Work experience

- *2021-present*
  PostDoc Researcher @ Polimi DART Lab
- *2015-2021*
  Flight Dynamics Engineer
  Deimos Space @ ESA/ESOC
- *2014-2015*
  Senior Project Engineer
  Dinamica Srl
- *2011*
  Junior Thermal Engineer
  Carlo Gavazzi Space/ OHB Italia

Programming Skills



Research Interests

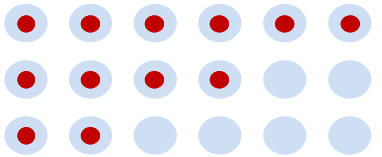
- Guidance, Navigation, Control
- Orbit propagation
- Numerical methods for astrodynamics
- Manoeuvre optimization
- Uncertainty propagation
- Space surveillance and tracking

ESA missions LEOPS & FD OPS:

Science missions:
MEX, Exomars, BepiColombo, Solar Orbiter (Rosetta, Lisa Pathfinder)

Earth orbiters:
Sentinels, Galileo, XMM/INTEGRAL, Swarm

LANGUAGES



POLITECNICO MILANO 1863

This week agenda

- **Today** ⟹ **SPICE overview and hands-on session**
  - Kernels
  - Time conversions
  - Reference frames and ephemeris

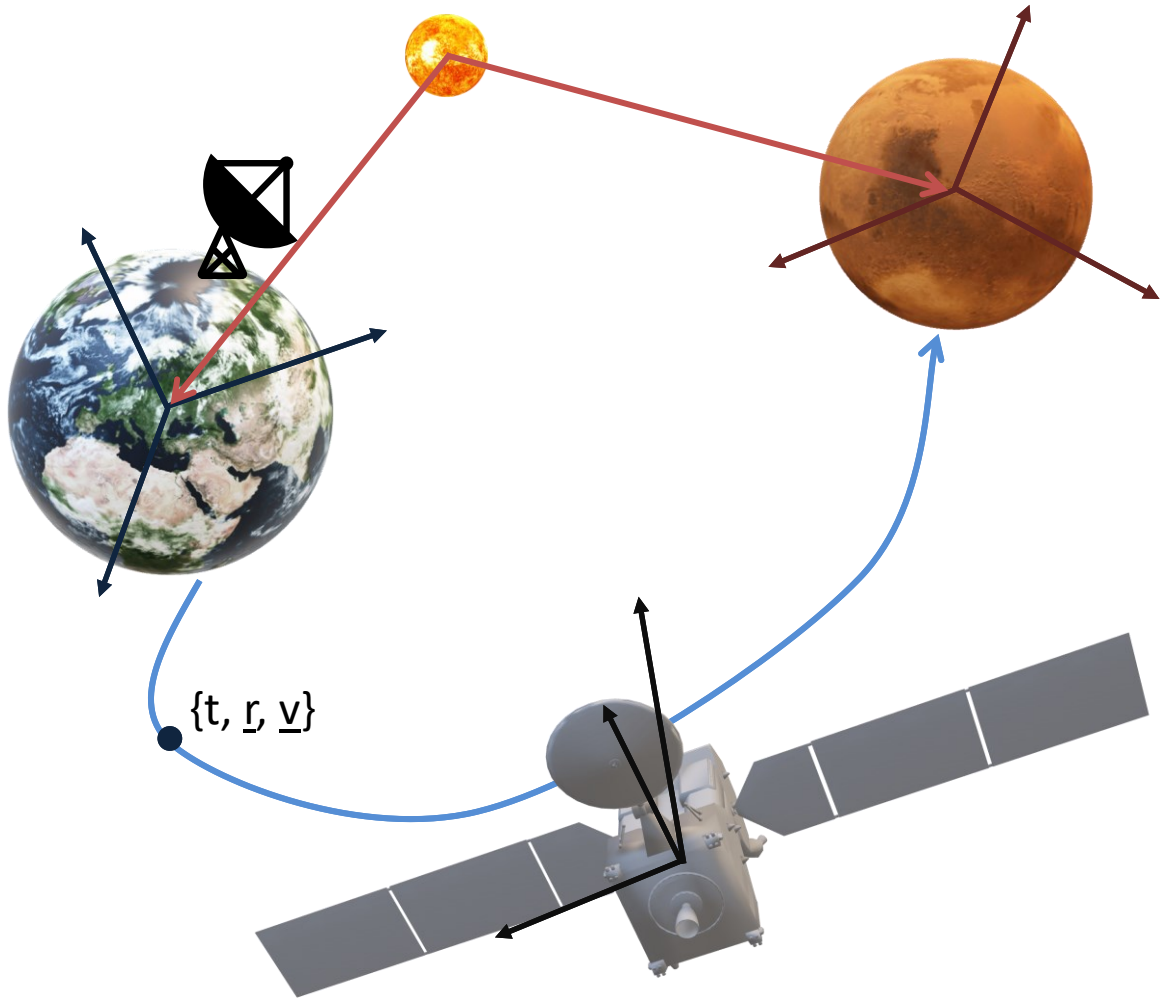
- **Tomorrow** ⟹ **SPICE application**
  - N-body propagator

- **Friday** ⟹ **SGP4 and TLEs**
  - SPG4 overview
  - SPG4 hands-on session

Objective ⟹ **provide the instruments and tools for the course assignments**

What ‘practical tasks’ are needed to perform Guidance and Navigation?

- Access planetary **ephemeris**
- Access/store the **state vector** of a spacecraft (or celestial body)
- Perform **time conversions**
- Convert between **reference frames**
    - Celestial bodies frames
    - Ground stations
    - Spacecraft
- Compute **observation geometries**
- Define attitude law and instruments pointing

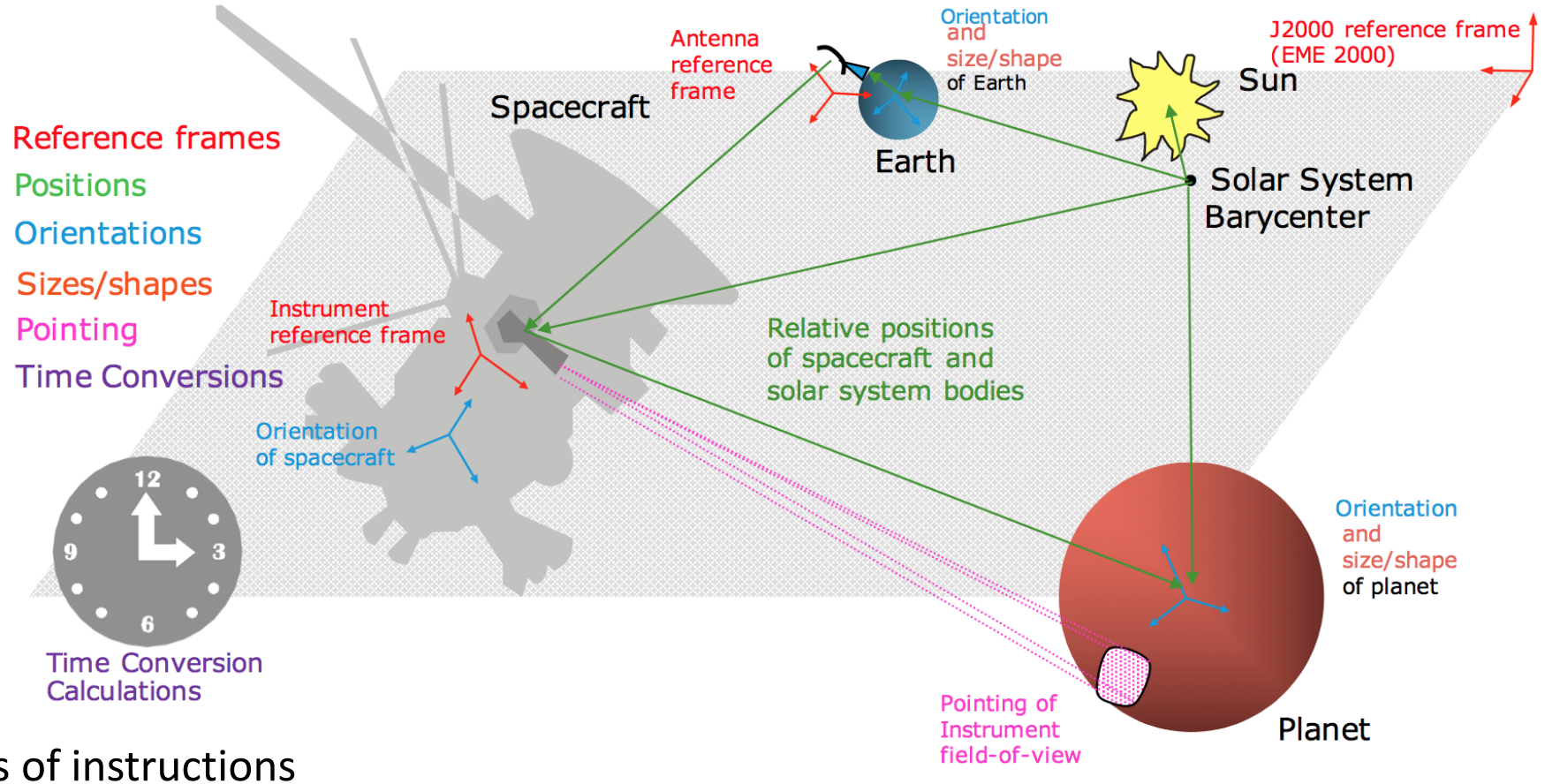


scifleet.esa.int/#/model-wall

What is SPICE?

NASA's Observation Geometry System for Space Science Missions

Satellite
Planet
Instrument
C-Matrix (camera)
Events



Useful software features:

- Common framework and sets of instructions
- Same reference frames used when calling functions and tools

Credits *NAIF Team.*

What is SPICE?

CONTENTS

- Library of subroutines
  - Only a handful of routines is usually needed
- Programs
  - SPICE data production
  - SPICE data management
- Documentation
  - Highly annotated source code
  - Technical reference manuals
  - User guides
  - Tutorials: naif.jpl.nasa.gov/naif/tutorials.html

VERSIONS

- Nine languages
  - Fortran 77
  - C
  - **MATLAB**
  - IDL
  - Java Native Interface (JNI)
  - Python, Julia, Ruby, Swift (provided by 3rd parties)
- Four platforms
  - PC/Linux
  - PC/Windows
  - Mac/OSC
  - Sun/Solaris
- Several compilers (Fortran and C)

Want to know more? Tutorial names here

NAIF online resources: naif.jpl.nasa.gov/naif/index.html

Where is SPICE used?

Data Restorations	Selected Past Users	Current/Pending Users	Examples of Possible Future Users
Apollo 15, 16 [L]	Magellan [L]	Cassini Orbiter	NASA Discovery Program
Mariner 2 [L]	Clementine (NRL)	Mars Odyssey	NASA New Frontiers Program
Mariner 9 [L]	Mars 96 (RSA)	Mars Exploration Rover	Lunar IceCube (Moorehead State)
Mariner 10 [L]	Mars Pathfinder	Mars Reconnaissance Orbiter	LunaH-Map (Arizona State)
Viking Orbiters [L]	NEAR	Mars Science Laboratory	Luna-Glob (RSA)
Viking Landers [L]	Deep Space 1	Juno	Aditya-L1 (ISRO)
Pioneer 10/11/12 [L]	Galileo	MAVEN	*Examples of Users not Requesting NAIF Help*
Haley armada [L]	Genesis	SMAP (Earth Science)	GOLD (LASP, UCF) (Earth Science) [L]
Phobos 2 [L] (RSA)	Deep Impact	OSIRIS REx	Hera (ESA)
Ulysses [L]	Huygens Probe (ESA) [L]	InSight	ExoMars RSP (ESA, RSA)
Voyagers [L]	Stardust/NExT	Mars 2020	Emmirates Mars Mission (UAE via LASP)
Lunar Orbiter [L]	Mars Global Surveyor	Europa Clipper	Hayabusa-2 (JAXA)
Helios 1,2 [L]	Phoenix	NISAR (NASA and ISRO)	Proba-3 (ESA)
	EPOXI	Psyche	Parker Solar Probe
	GRAIL	Lucy	EUMETSAT GEO satellites [L]
	DAWN	Lunar Reconnaissance Orbiter	MOM (ISRO)
	Messenger	Mars Express (ESA)	Chandrayan-2 (ISRO)
	Phobos Sample Return (RSA)	ExoMars 2016 (ESA, RSA)	Solar Orbiter (ESA)
	Venus Express (ESA)	Akatsuki (JAXA)	STEREO [L]
	Rosetta (ESA)	Korean Pathfinder Lunar Orbiter (KARI)	Spitzer Space Telescope [L]
[L] = limited use	Chandrayaan-1 (ISRO)	New Horizons	Kepler [L]
[S] = special services	Hayabusa (JAXA)	JUICE (ESA)	Hubble Space Telescope [S][L]
	Kaguya (JAXA)	Bepicolombo (ESA, JAXA)	James Webb Space Telescope [S][L]
	LADEE		Altius (Belgian earth science satellite)
	ISO [S] (ESA)		Armadillo (CubeSat, by UT at Austin)
Last updated: 1/7/20	Smart-1 (ESA)	Deep Space Network	Spectrum-RG (RSA)

NAIF has or had project-supplied funding to support mission operations, consultation for flight team members, and SPICE data archive preparation. NAIF also has PDS fu help scientists use SPICE data that have been officially archived at the NAIF Node of the PDS.

NAIF has or had NASA funding to support a foreign partner in SPICE deployment and archive review, and to consult with flight team SPICE users.

NAIF has token funding to consult with kernel producers at APL. APL provides support to science teams.

NAIF has or had modest PDS-supplied funding to consult on assembly of a SPICE archive.

Used since more than 20 years in NASA missions for:

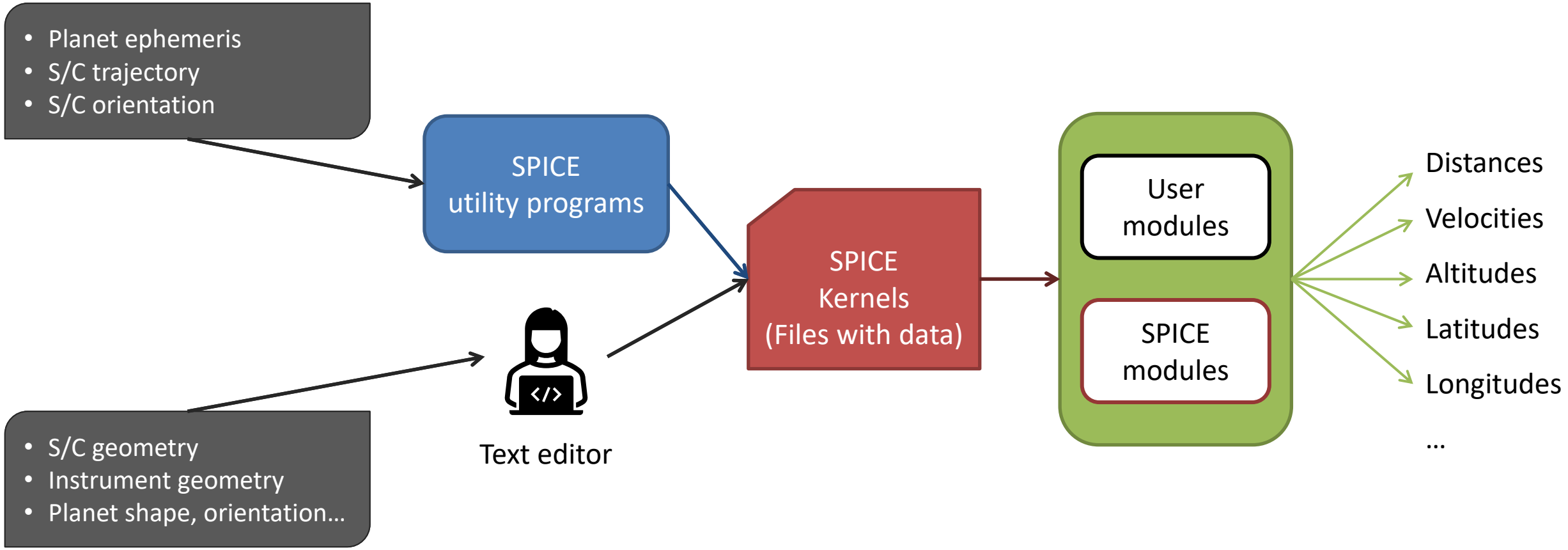
- Mission preparation
- Operations
- Mission planning
- Post-mission analysis
- Science data processing
- Visualization

ESA provides SPICE kernels to scientists who work with ESA's planetary missions

- Rosetta
- BepiColombo
- TGO
- …

ESA missions' SPICE kernels repo:
cosmos.esa.int/web/spice

SPICE overview



Data collection | Data preparation | Analysis and computation | Results

What is a kernel?

"Kernel" means File

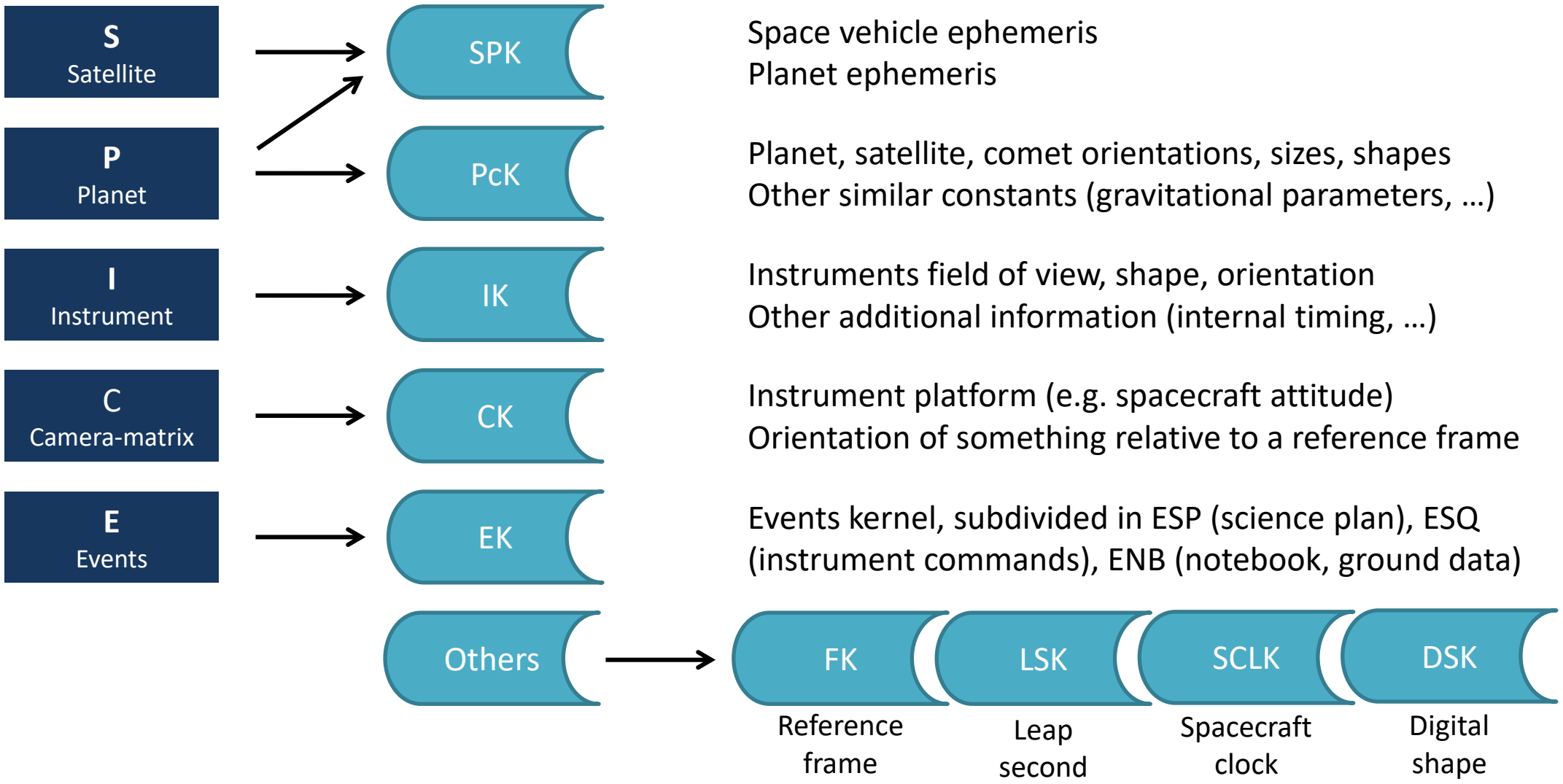
- **Binary** — **Not** Human Readable (Readable with **Utilities**)
- **Text** — Human Readable

Always used
Often used
Rarely used

- **SPK** Spacecraft and Planet Ephemeris
- **PCK** Planetary Constants
- **IK** Instrument
- **CK** Orientation
- **EK** Events

- **LSK** Leapseconds
- **MK** Meta-Kernels
- **FK** Reference Frame Specifications
- **SCLK** Spacecraft Clock
- **DSK** Digital Shape Kernel
- **DBK** Database Mechanism

SPICE Kernels



Space vehicle ephemeris
Planet ephemeris

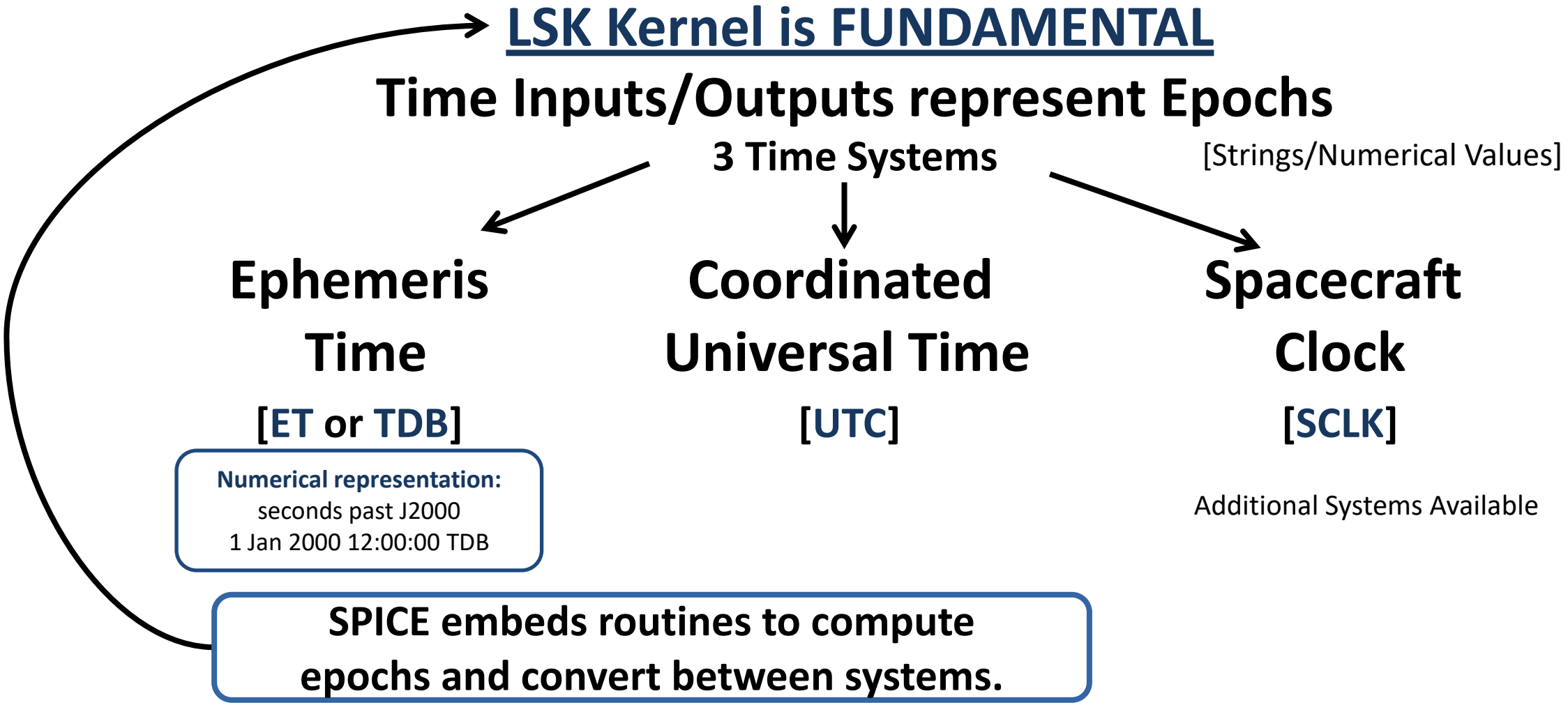
Planet, satellite, comet orientations, sizes, shapes
Other similar constants (gravitational parameters, …)

Instruments field of view, shape, orientation
Other additional information (internal timing, …)

Instrument platform (e.g. spacecraft attitude)
Orientation of something relative to a reference frame

Events kernel, subdivided in ESP (science plan), ESQ (instrument commands), ENB (notebook, ground data)

Time conversions and calculations



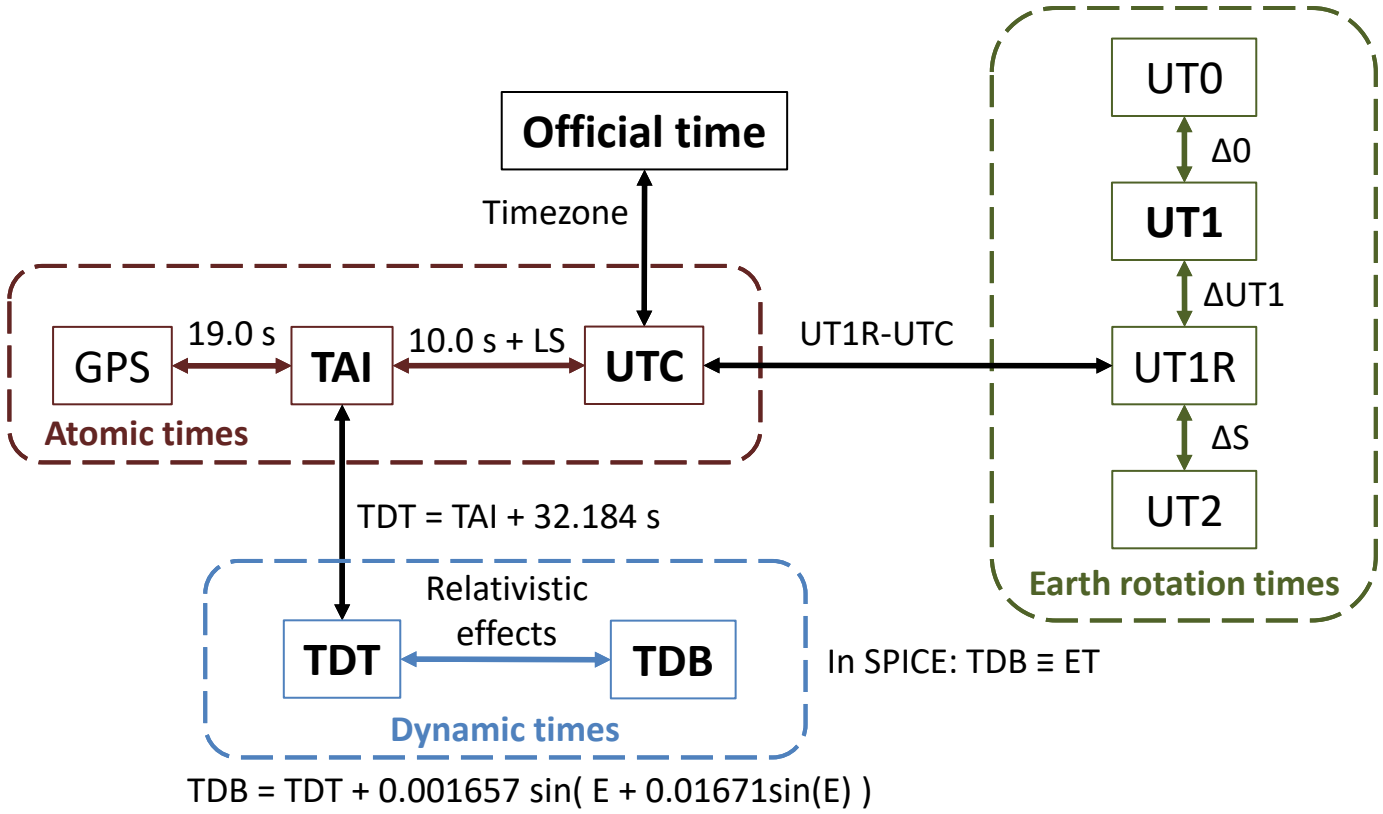
15_time.pdf

Some notes on Time Systems

- **International Atomic Time (TAI)**
  - Count of atomic seconds past astronomically determined midnight of 1 Jan 1958 00:00:00
- **Coordinated Universal Time (UTC)**
  - Civil time at Greenwich England (≈GMT)
  - TAI = UTC + 10.0 s + LS

- **UT1** is the hour representation of the angle between Greenwich zenith meridian and the 'computed mean Sun' location
- **UT1** and **UTC** get out of sync due to non-regularity of **Earth rotation**
  - Leap Second inserted if difference > 0.9 s

- **Barycentre Dynamical Time (TDB)**
  - Mathematical ideal, 'continuous' time
- **Terrestrial Dynamical Time (TDT) or Terrestrial Time (TT)**
  - Ideal time on Earth at sea level
  - Small variations (1.6 ms) from TDB due to relativistic effects

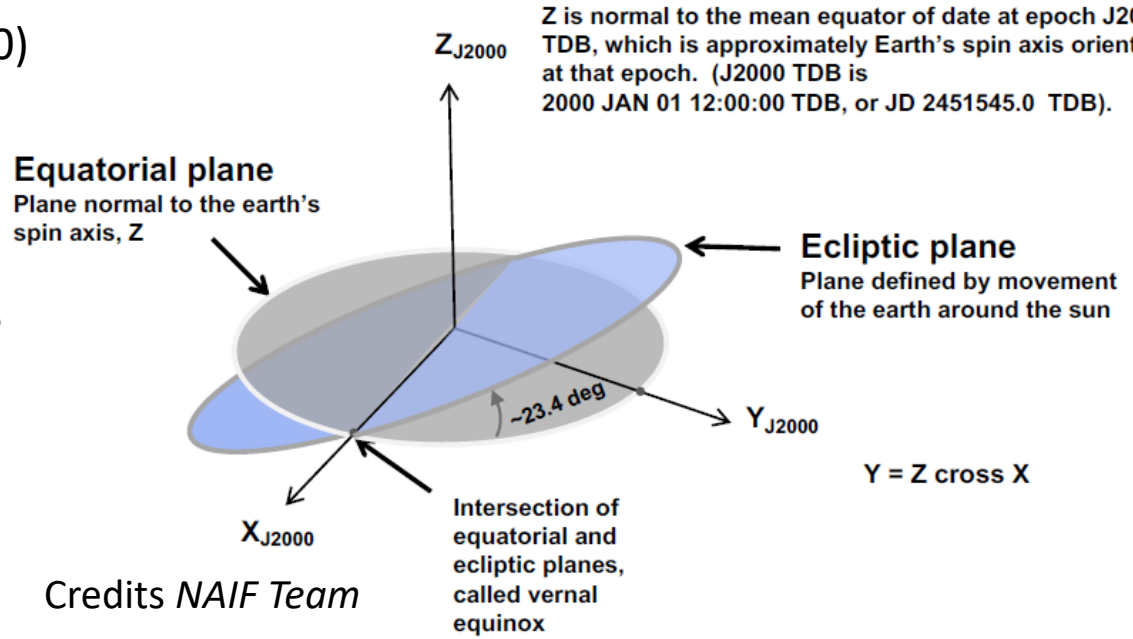


!

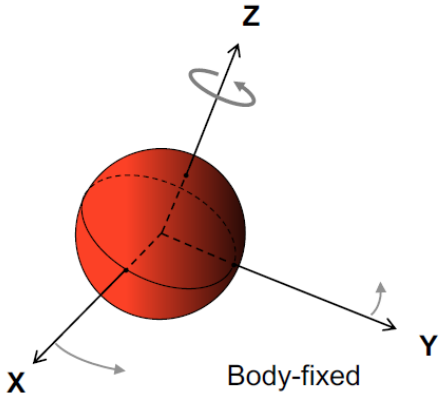
Leap seconds complicate computation of differences between UTC epochs:
- Knowledge of past leap seconds must be known → **LSK kernels** ☺
- Unknown/undetermined future durations (not predictable)
- Placed on June 30 or December 31

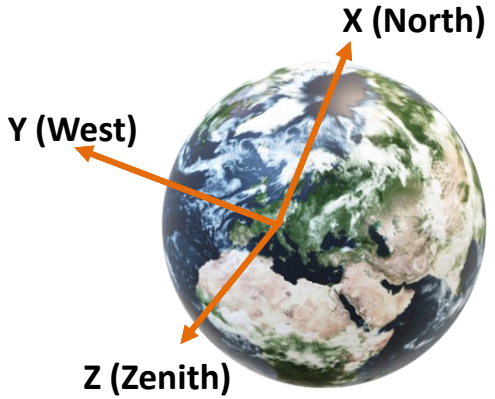
Reference frames and coordinate systems

- Inertial reference frame: **J2000** (or Earth Mean Equator J2000)
- **ICRF**: International Celestial Reference Frame
- **ICRF ≡ J2000 ≡ EME2000**
- Planet-fixed frames (IAU)
  - Z-axis is rotation axis, X-axis points towards planet prime meridian, Y-axis completes the right-hand frame
  - Name convention: **IAU_<planetName>**
- High-accuracy frames for Earth and Moon available
  - Accounts for nutation, precession and polar motion
- **Topocentric** frame:
  - Located on or near a body surface
  - Used for **stations**, rovers, or landers
  - One axis normal to spheroid or parallel to gravity
  - Many different definition possible!



Credits *NAIF Team*





17_frames_and_coordinate_systems.pdf

Ephemeris Subsystem

Based on **Ephemeris Objects** (spacecrafts, planets, satellites, comets, … everything)

Ephemeris Data are Saved in SPKs

↓

Position + Velocity = State

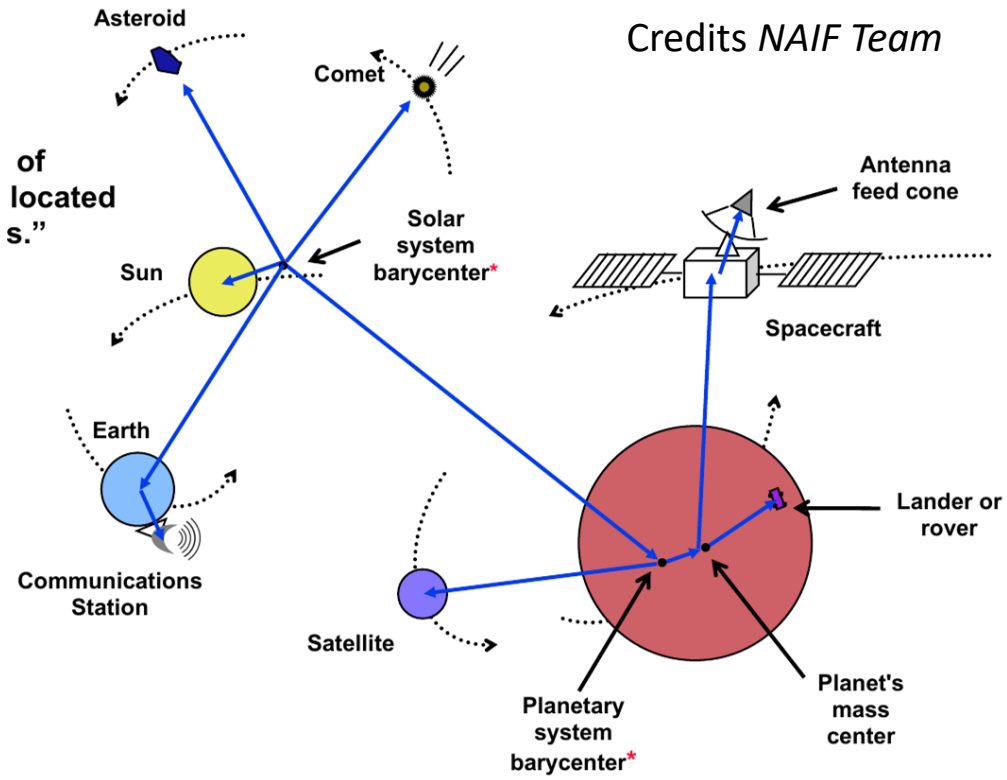
Reference Frames

- J2000 Inertial Frame
- Body-Fixed Frames
- Topocentric Frames
- …

Several Coordinate Systems Used

Corrections

- Light Time [LT]
- Stellar Aberration [S]

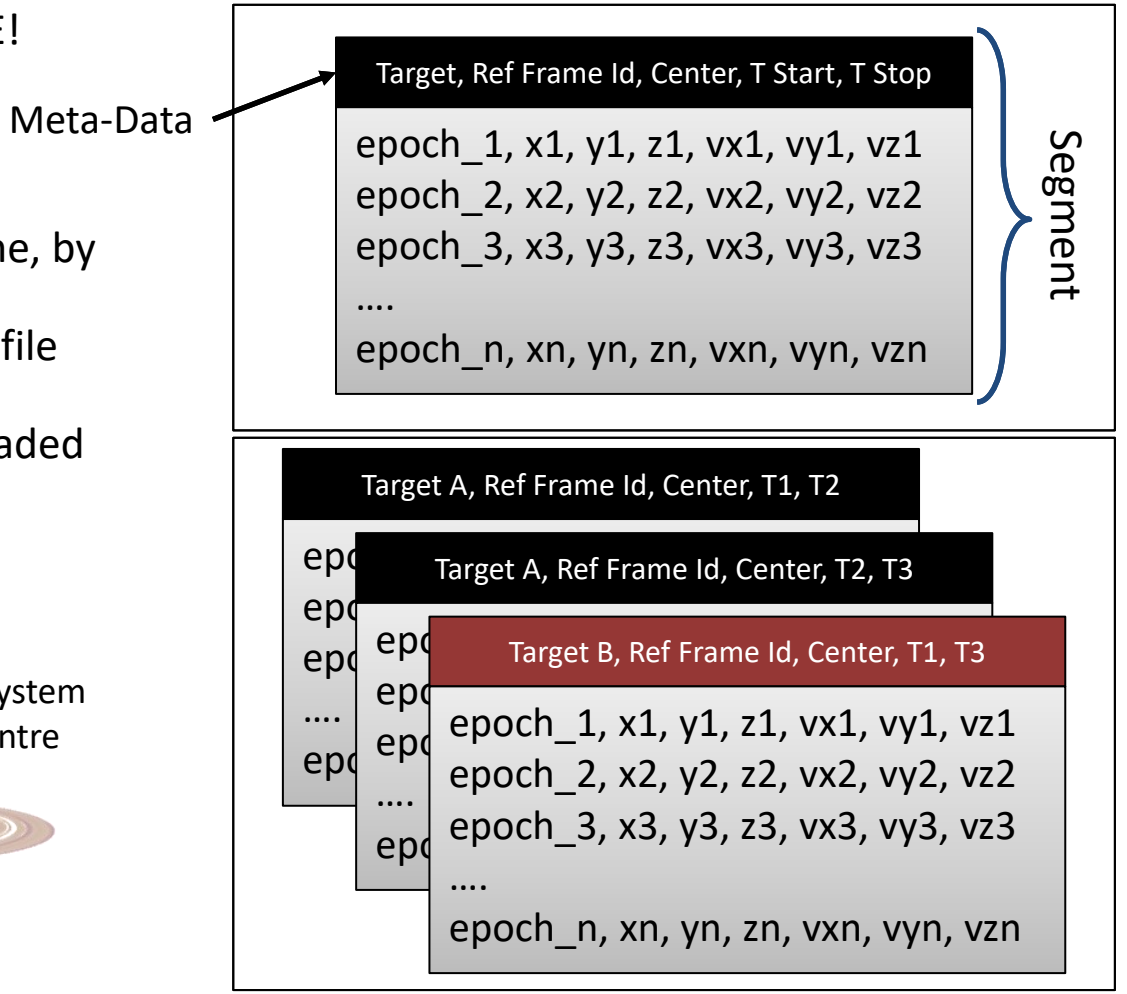


Credits *NAIF Team*

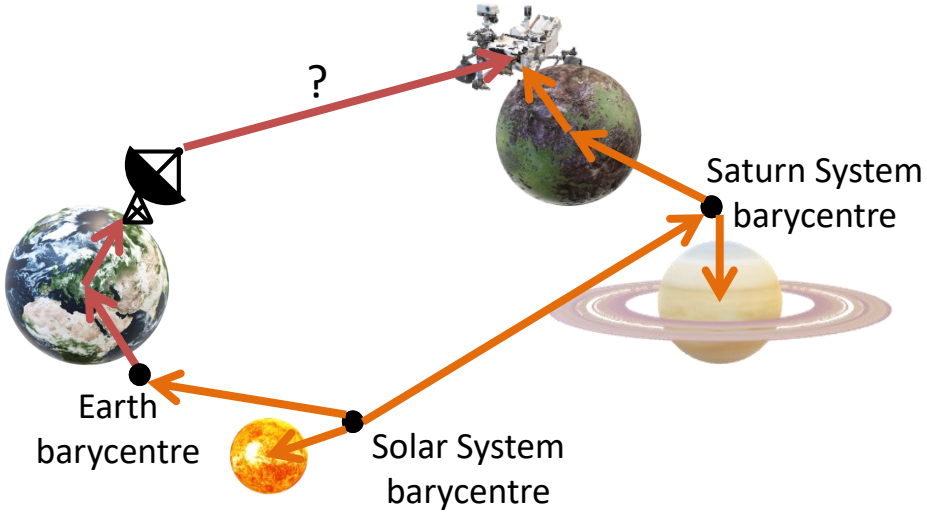
06_naif_ids.pdf

Some notes on SPK kernels

- SPK kernels are composed of **segments**
  - Header with meta-data → load reference frame definition in SPICE!
  - List of epoch + state vectors
  - State in-between epochs can be accessed → **interpolation**
- Each SPK can contain **multiple segments** from **multiple targets**
  - SPK segments **do not need to be ordered** (chronologically, by frame, by target, …)
  - Within a SPK file, segments whose **physical location** is **later** in the file have **higher priority**
- When loading more than one kernel with info for an object, the one loaded later have higher priority



SPICE routines **automatically** compute the chain of rotations/translations to compute the desired states/vectors



18_spk.pdf

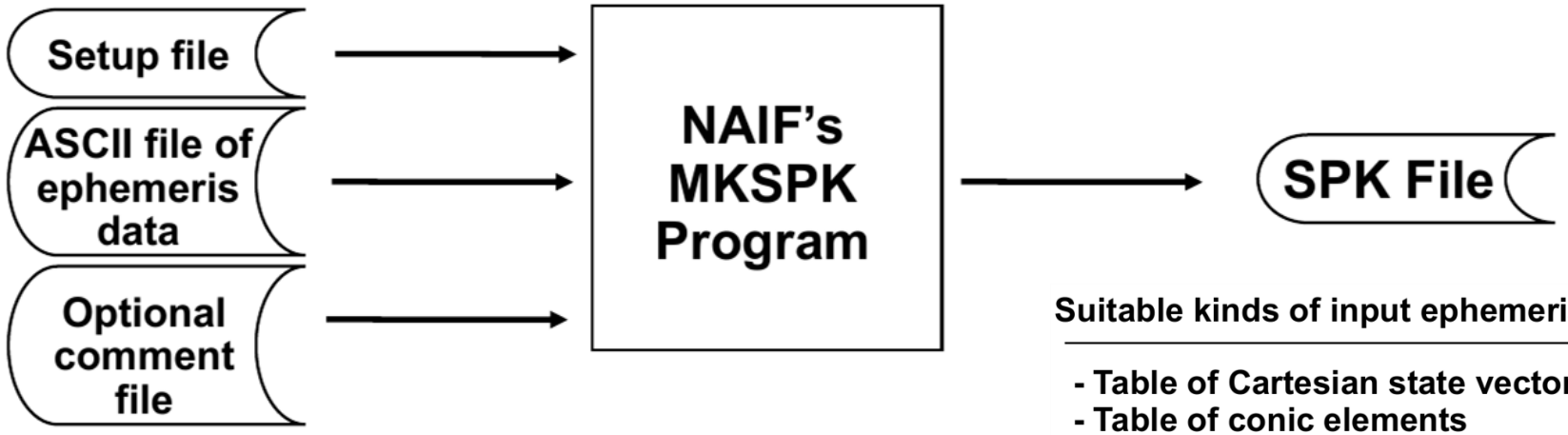
HowTo Make a SPK

- Using the **mkspk** utility
- Using SPK writing subroutines

Then ...

Validate your SPK!

Consider using:
- **brief** utility
- **spacit** utility
- comparison w/ **source**



Credits *NAIF Team.*

Suitable kinds of input ephemeris data are:

- Table of Cartesian state vectors
- Table of conic elements
- One or more sets of equinoctial elements
- One or more sets of Space Command two-line elements

42_making_an_spk.pdf

“Camera-Matrix” Kernel

CK Kernels are Attitude/Orientation Kernels

To obtain orientation one needs at least three SPICE kernel types
[One may also need an **FK** in some cases]

- **CK**
- **SCLK** [spacecraft clock]
- **LSK** [leapseconds]

Types of CKs

- **Reconstruction/Definitive**
- **Predict**

Check the Data Coverage! → **SPKs have continuous coverage. CKs often have gaps.**

HowTo Make a CK

Too many procedures ...

Already Existing
Orientation Data
[only **Packaging**]

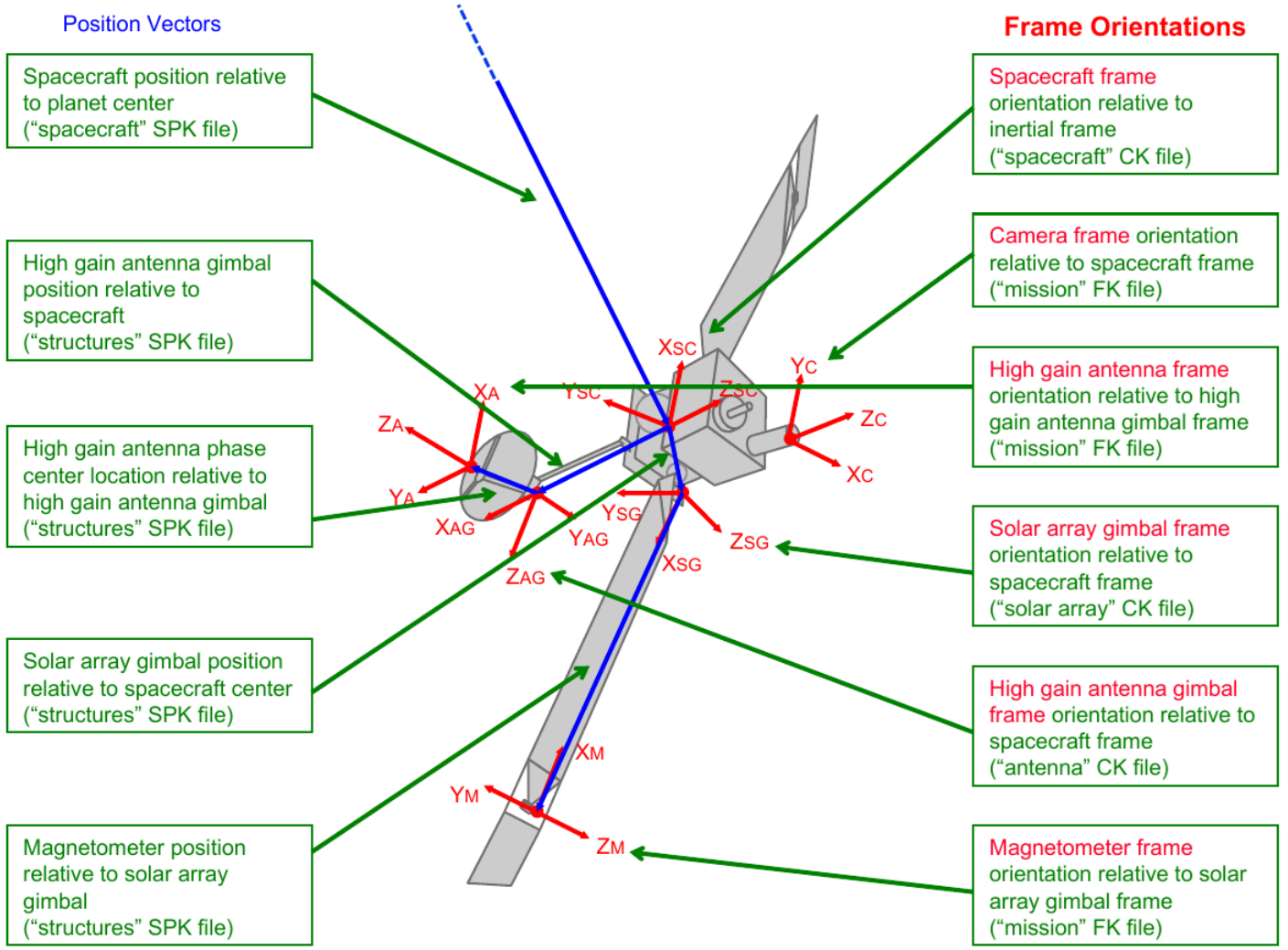
- Using the **msopck** for packaging
- Using CK packaging subroutines

Computing & Packaging
Orientation Data

- Using the **prediCkt** utility
- Using CK computing subroutines

Consider to read the related documentations, the procedures are **strongly different** and **specific** for each method.

Example



Credits *NAIF Team.*

Conclusions

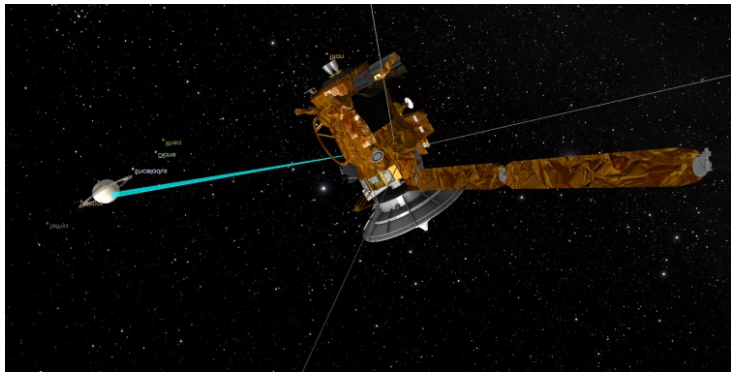
- This is just an overview of the capabilities of SPICE…there is **a lot** of information available
  - Documentation: naif.jpl.nasa.gov/pub/naif/toolkit_docs/MATLAB/index.html
  - NAIF tutorials: naif.jpl.nasa.gov/naif/tutorials.html
  - NAIF SPICE lessons: naif.jpl.nasa.gov/naif/lessons.html
  - … **and also the MATLAB help**

- Kernel location
  - Generic kernels: naif.jpl.nasa.gov/pub/naif/generic_kernels/
    - Look at the page here ↑ to download the SPK, PCK, FK, LSK kernels you might need
  - ESA/NASA missions kernels: naif.jpl.nasa.gov/naif/data_archived.html
  - IERS data: www.iers.org/IERS/EN/DataProducts/data.html

Visualization tool - Cosmographia 4.0

Cosmographia is a visualization program rendering the solar system and its bodies in 3D to create a freely navigable map of the solar system.

- **Time** and **Camera** Position **Manipulation**
- Compatible with **SPICE Data**
- Support **Mission Analysis**
- Support **Public Outreach**



Credits *cosmoguide.org*

Download

1. Page: https://naif.jpl.nasa.gov/naif/cosmographia_components.html
2. Download and install it.

Then….

Write your own *".json"* files and upload them in Cosmographia!