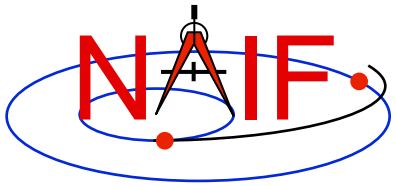


Derived Quantities

April 2016

What are Derived Quantities?

Navigation and Ancillary Information Facility

- "Derived quantities" are computed using data from kernels
  - These are the primary reason that SPICE exists!

- Examples follow on the next several pages.

High-level Geometric Quantities

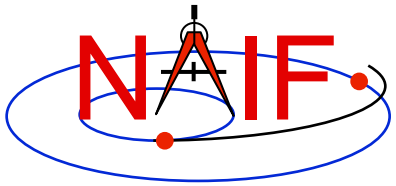
Navigation and Ancillary Information Facility

- Illumination angles (phase, incidence, emission)
  - ILUMIN*
- Illumination angles with any ephemeris object as the source
  - ILLUMG
- Subsolar point
  - SUBSLR*
- Subobserver point
  - SUBPNT*
- Surface intercept point
  - SINCPT*
- Longitude of the sun (L_s), an indicator of season
  - LSPCN
- Phase angle between body centers
  - PHASEQ
- Terminator on an ellipsoid
  - EDTERM

* These routines supercede the now deprecated routines ILLUM, SUBSOL, SUBPT and SRFXPT

Geometric Condition

- Ray in field-of-view?
  - FOVRAY
- Ephemeris object within field-of-view?
  - FOVTRG
- Determine occultation or transit condition
  - OCCULT



Geometric Events Finder

- **Most SPICE routines are used to determine a quantity at a specified time.**

- **The SPICE Geometry Finder (GF) subsystem takes the opposite approach: find times, or time spans, when a specified geometric condition or event occurs.**
  - **This is such a large topic that a separate tutorial ("geometry_finder") has been written for it.**

Geometric Objects

Navigation and Ancillary Information Facility

Function	Routine
• **Ellipsoids**	
– nearest point	– `NEARPT, SUBPNT, DNEARP`
– surface ray intercept	– `SURFPT, SINCPT`
– surface normal	– `SURFNM`
– limb	– `EDLIMB`
– slice with a plane	– `INELPL`
– altitude of ray w.r.t. to ellipsoid	– `NPEDLN`
• **Planes**	
– intersect ray and plane	– `INRYPL`
• **Ellipses**	
– project onto a plane	– `PJELPL`
– find semi-axes of an ellipse	– `SAELGV`

Position and State Coordinate Transformations

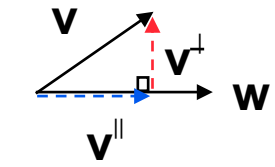
Navigation and Ancillary Information Facility

Coordinate Transformation	Routine	
– Transform state vector between two coordinate systems	– `XFMSTA`	General purpose API
– Latitudinal to/from Rectangular	– `LATREC` `RECLAT`	Single purpose APIs
– Planetographic to/from Rectangular	– `PGRREC` `RECPGR`	
– R.A. Dec to/from Rectangular	– `RADREC` `RECRAD`	
– Geodetic to/from Rectangular	– `GEOREC` `RECGEO`	
– Cylindrical to/from Rectangular	– `CYLREC` `RECCYL`	
– Spherical to/from Rectangular	– `SPHREC` `RECSPH`	

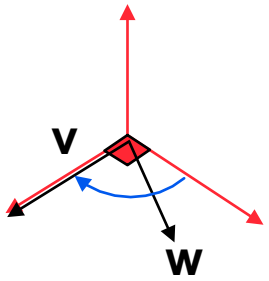
Vectors

Navigation and Ancillary Information Facility

Function	Routines
– **<v,w>**	– `VDOT, DVDOT`
– **v** x **w**	– `VCROSS, DVCRSS`
– **v**/|**v**|	– `VHAT, DVHAT`
– **v** x **w** / | **v** x **w**|	– `UCROSS, DUCRSS`
– **v** + **w**	– `VADD, VADDG`
– **v** - **w**	– `VSUB, VSUBG`
– a**v** [+ b**w** [+ c**u**]]	– `VSCL, [VLCOM, [VLCOM3]]`
– angle between **v** and **w**	– `VSEP, DVSEP`
– |**v**|	– `VNORM`



– `VPROJ and VPERP`



– `TWOVEC, FRAME and TWOVXF`

Matrices

Navigation and Ancillary Information Facility

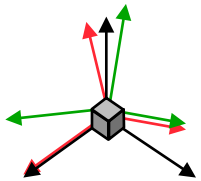
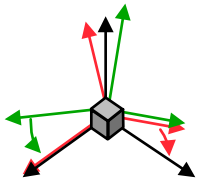
Selected Matrix-Vector Linear Algebra Routines

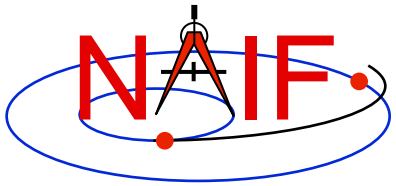
• Function	• Routine
– M x v	– `MXV`
– M x M	– `MXM`
– M^t x v	– `MTXV`
– M^t x M	– `MTXM`
– M x M^t	– `MXMT`
– v^t x M x v	– `VTMV`
– M^t	– `XPOSE`
– M^{-1}	– `INVERSE, INVSTM`

`M` = Matrix
`V` = Vector
`X` = Multiplication
`T` = Transpose

Matrix Conversions

Navigation and Ancillary Information Facility

Transform between…			Routines
 Euler angles	⟷	$\begin{matrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{matrix}$ 3x3 rotation matrix	– EUL2M, M2EUL
 Euler angles and Euler angle rates *or* rotation matrix and angular velocity vector	⟷	$\begin{matrix} a_x & a_y & a_z & \\ b_x & b_y & b_z & 0 \\ c_x & c_y & c_z & \\ \alpha_x & \alpha_y & \alpha_z & a_x \; a_y \; a_z \\ \beta_x & \beta_y & \beta_z & b_x \; b_y \; b_z \\ \gamma_x & \gamma_y & \gamma_z & c_x \; c_y \; c_z \end{matrix}$ 6x6 state transformation matrix	– EUL2XF, XF2EUL RAV2XF, XF2RAV
 Rotation axis and angle	⟷	$\begin{matrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{matrix}$ 3x3 rotation matrix	– RAXISA, AXISAR ROTATE, ROTMAT
(Q_0,Q_1,Q_2,Q_3) SPICE Style Quaternion	⟷	$\begin{matrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{matrix}$ 3x3 rotation matrix	– Q2M, M2Q



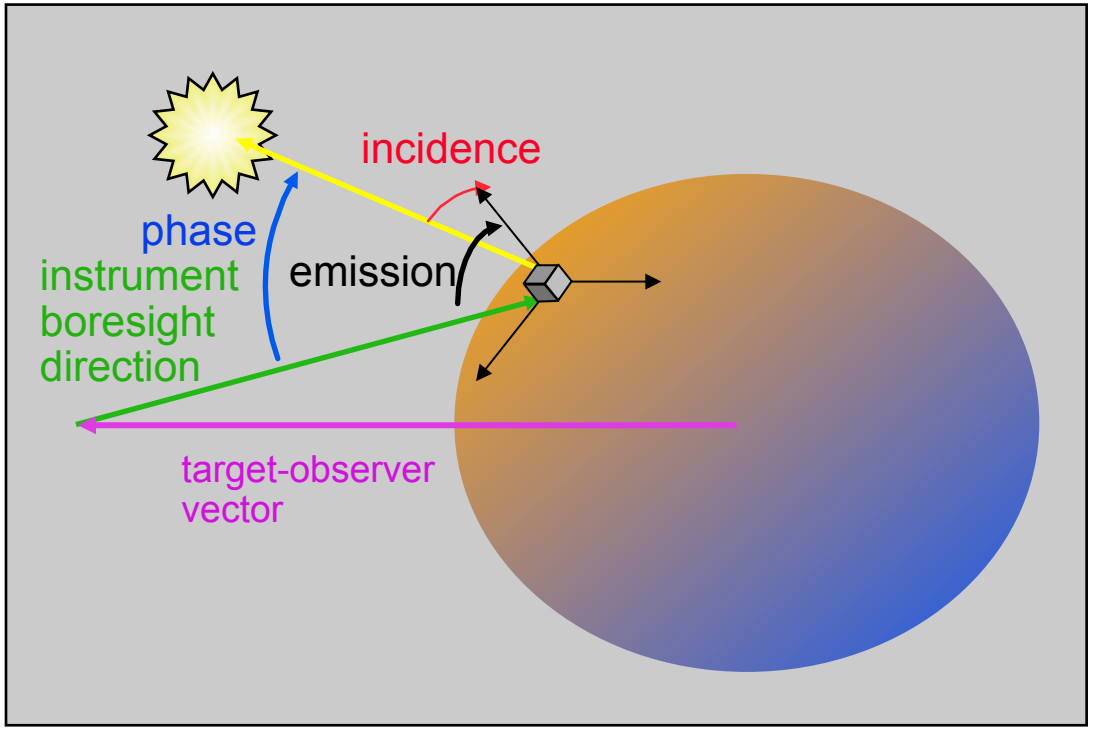
Examples

- On the next several pages we present examples of using some of the “derived quantity” APIs.

Computing Illumination Angles

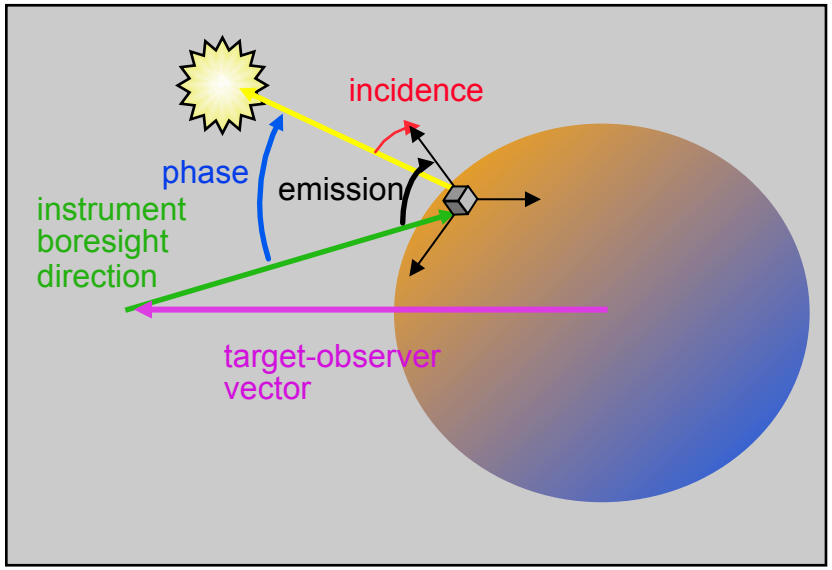
Navigation and Ancillary Information Facility

- **Given the direction of an instrument boresight in a body-fixed frame, return the illumination angles (incidence, phase, emission) at the surface intercept on a tri-axial ellipsoid**



Computing Illumination Angles

Navigation and Ancillary Information Facility

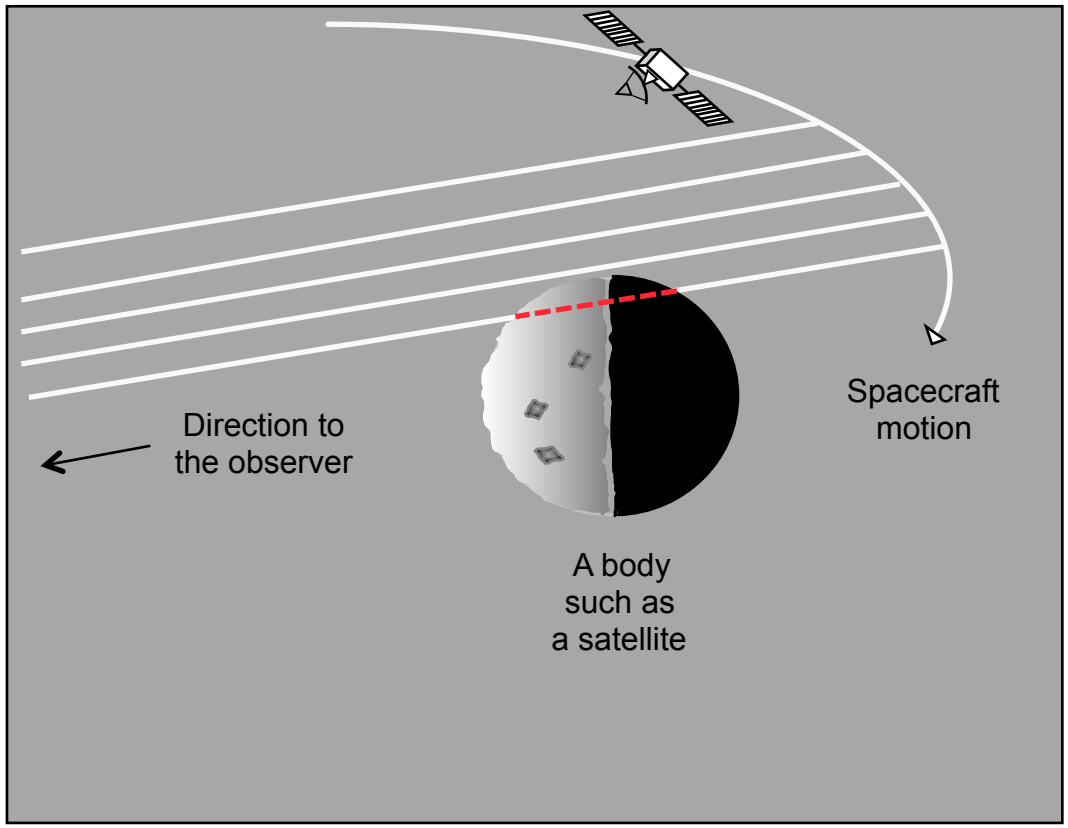


- CALL GETFOV to obtain boresight direction vector
- CALL SINCPT to find intersection of boresight direction vector with surface
- CALL ILUMIN to determine illumination angles

Computing Occultation Events

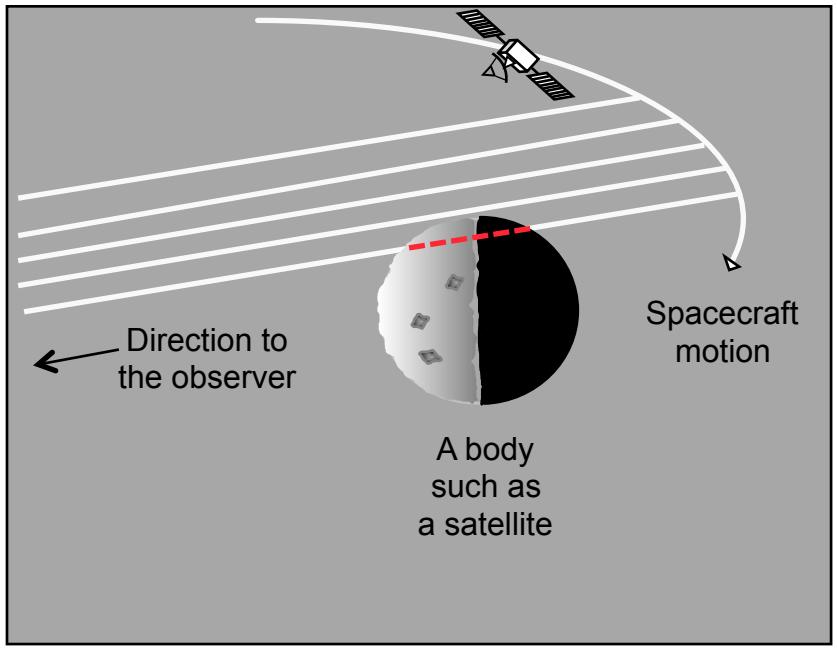
Navigation and Ancillary Information Facility

- **Determine when the spacecraft will be occulted by an object (such as a natural satellite) as seen from an observer (such as earth).**



Find Occultation Ingress/Egress

Navigation and Ancillary Information Facility



- **Select a start epoch, stop epoch and step size.**
  - **Start and stop epochs can bracket multiple occultation events**
  - **Step size should be smaller than the shortest occultation duration of interest, and shorter than the minimum interval between occultation events that are to be distinguished, but large enough to solve problem with reasonable speed.**
  - **Insert search interval into a SPICE window. This is the "confinement window."**
- **CALL GFOCLT to find occultations, if any. The time intervals, within the confinement window, over which occultations occur will be returned in a SPICE window.**
  - **GFOCLT can treat targets as ellipsoids or points (but at least one must be an ellipsoid).**
  - **GFOCLT can search for different occultation or transit geometries: full, partial, annular, or "any."**