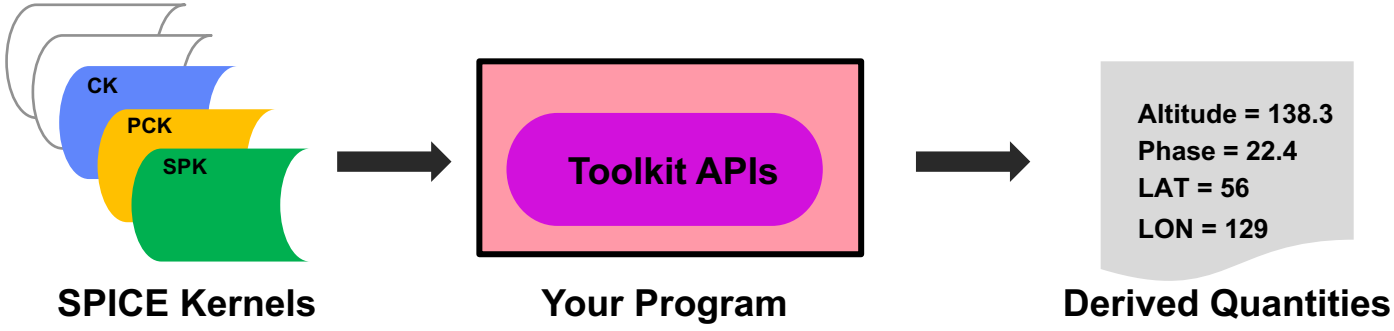
What are Derived Quantities?

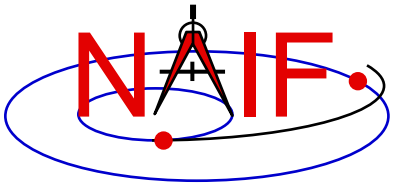
Navigation and Ancillary Information Facility

- **Derived quantities, what we often call "observation geometry," are produced using data from kernels.**
  - **These are the primary reason that SPICE exists!**

- **The SPICE Toolkit contains many routines that assist with the computations of derived quantities.**
  - **Some are fairly low level, some are quite high level.**
  - **More are being added as time permits.**



- **Examples follow on the next several pages.**



High-level Geometric Computations

- **Geometric Parameter or Condition**
  - **Determine a quantity or a condition at a specified time.**

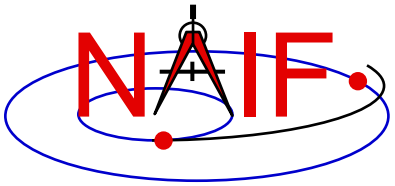
- **Geometry Finder (GF)**
  - **Find times, or time spans, when a specified "geometric event" occurs, or when a specified "geometric condition" exists.**
    - » **This is such a large topic that a separate tutorial ("geometry_finder") has been written for it.**

Examples of Geometric Parameters

Navigation and Ancillary Information Facility

- Illumination angles (phase, incidence, emission)
  - ILLUMF, ILLUMG, ILUMIN*
- Sub-solar point
  - SUBSLR*
- Sub-observer point
  - SUBPNT*
- Surface intercept point
  - SINCPT*, DSKXV, DSKXSI
- Longitude of the sun (Ls), an indicator of season
  - LSPCN
- Phase angle between body centers
  - PHASEQ
- Limb and terminator points on an ellipsoid or DSK
  - LIMBPT, TERMPT
- Surface points at specified longitude, latitude coordinates
  - LATSRF
- Outward surface normal on extended object
  - SRFNRM

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



Examples of Geometric Conditions

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

Examples of Geometric Searches

- **Find times when:**
  - ray is in field-of-view
    - » GFRFOV
  - ephemeris object is within field-of-view
    - » GFTFOV
  - object is in occultation or transit
    - » GFOCLT
  - object is at periapse
    - » GFDIST
  - latitude and longitude are in specified ranges
    - » GFPOSC
  - solar incidence angle is below a specified limit
    - » GFILUM
- **Far more GF functionality is available; see the GF tutorial.**

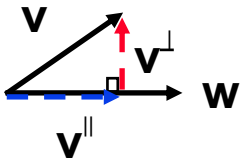
Position and State Coordinate Transformations

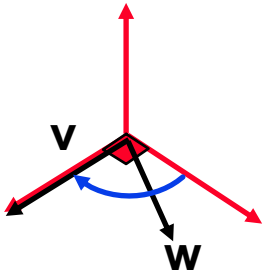
Navigation and Ancillary Information Facility

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**
  - **<v,w>**
  - **v** x **w**
  - **v**/||**v**||
  - **v** x **w** / || **v** x **w**||
  - **v** + **w**
  - **v** - **w**
  - a**v** [+ b**w** [+ c**u**]]
  - angle between **v** and **w**
  - ||**v**||
  - 

  - 

- **Routine**
  - `VDOT,  DVDOT`
  - `VCROSS,  DVCRSS`
  - `VHAT,  DVHAT`
  - `UCROSS,  DUCRSS`
  - `VADD,  VADDG`
  - `VSUB,  VSUBG`
  - `VSCL,  [VLCOM,  [VLCOM3]]`
  - `VSEP`
  - `VNORM`
  - `VPROJ,  VPERP`
  - `TWOVEC,  FRAME`

Matrices

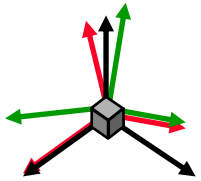
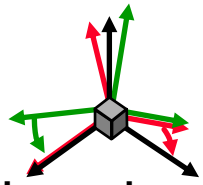
Selected Matrix-Vector Linear Algebra Routines

• 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}	– `INVERT, INVSTM`

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

Matrix Conversions

Navigation and Ancillary Information Facility

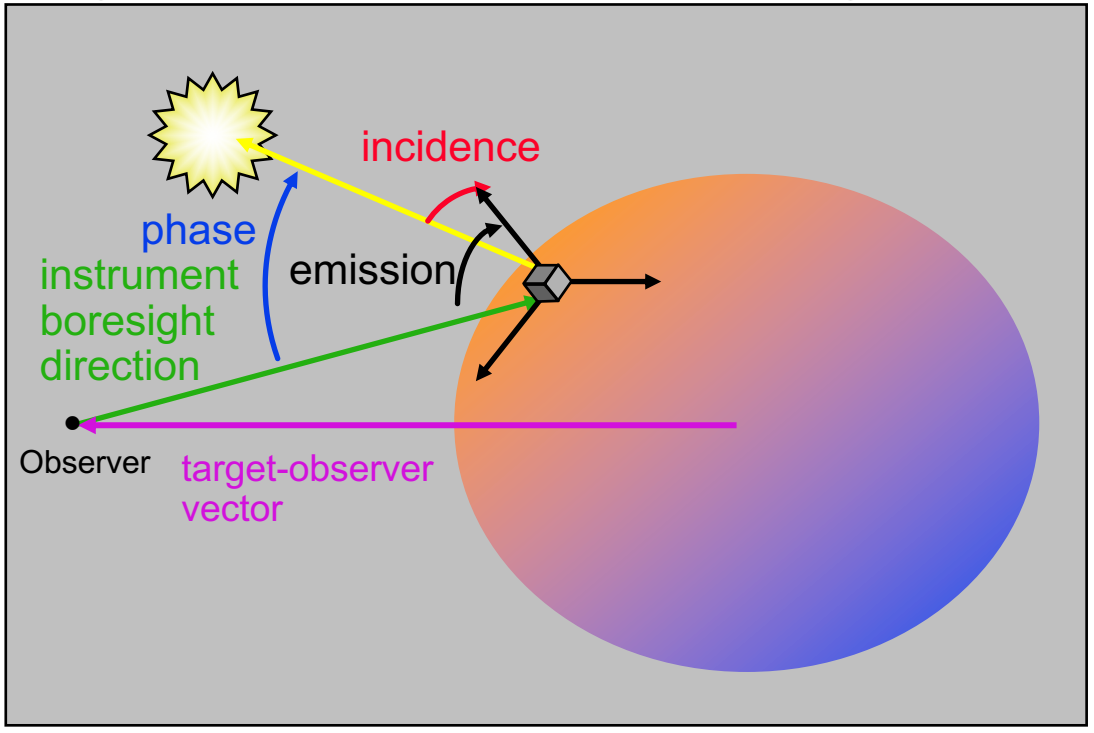
Function			Routines
 Euler angles	⟷ Transform between	$\begin{matrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{matrix}$ 3x3 rotation matrix	– EUL2M, M2EUL
 Euler angles and Euler angle rates *or* rotation matrix and angular velocity vector	⟷ Transform between	$\begin{matrix} a_x & a_y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

Examples of Computing Derived Quantities

- On the next several pages we present examples of using some of the “derived quantity” APIs.
- Explore the “Most Used SPICE APIs” document to learn more.

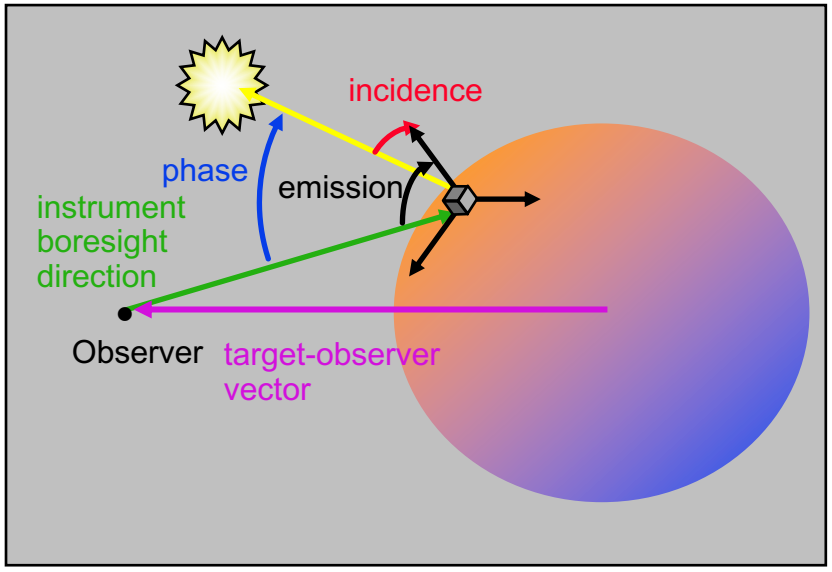
Computing Illumination Angles

- **Given the direction of an instrument boresight in a body-fixed frame, return the illumination angles (incidence, phase, emission) at the boresight's surface intercept on an object, with the object's shape modeled by a tri-axial ellipsoid or by DSK data.**



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