Derived Quantities

March 2010

Overview

- What are “derived quantities?”
- A quick tour of some of the routines provided for the computation of derived quantities
  - Vector/Matrix Routines
  - Geometry Routines
  - Coordinate System Routines
- Computing Illumination Angles
- Computing Ring Plane Intercepts
- Computing Occultation Events

What are Derived Quantities?

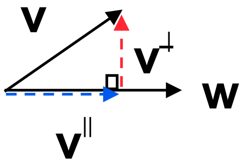
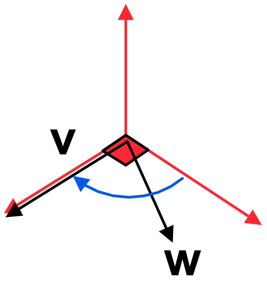
- Derived quantities are data produced from states, C-matrices, frame transformations, physical constants, time conversions, etc.
  - These are the primary reason that SPICE exists!
- Examples are:
  - Angles, Angular Rates
  - Distances, Speeds
  - Directions
  - Lighting conditions
  - Cartographic parameters
  - Time windows of events
- 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.

A Quick Tour

- **Vector/Matrix Routines**
  - Vector and vector derivative arithmetic
  - Matrix arithmetic
- **Geometric "Objects"**
  - Planes
  - Ellipses
  - Ellipsoids
  - Rays
- **Coordinate Systems**
  - Spherical: latitude/longitude, co-latitude/longitude, right ascension/declination; Geodetic, Cylindrical, Rectangular, Planetographic
- **Others**

The lists on the following pages are just
a subset of what's available in the Toolkit.

Vectors

• Function	• Routine
– <**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
– |**v**|	– VNORM
– 	– VPROJ, VPERP
– 	– TWOVEC, FRAME

Matrices

Selected Matrix-Vector Linear Algebra Routines

Function	Routine
$M \times v$	`MXV`
$M \times M$	`MXM`
$M^t \times v$	`MTXV`
$M^t \times M$	`MTXM`
$M \times M^t$	`MXMT`
$v^t \times M \times v$	`VTMV`
M^t	`XPOSE`
M^{-1}	`INVERSE, INVSTM`

`M` = Matrix

`V` = Vector

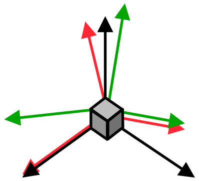
`X` = Multiplication

`T` = Transpose

Matrix Conversions

Function

Routines



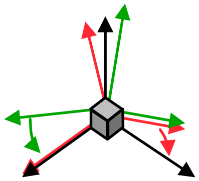
Euler angles

↔ 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



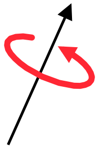
Euler angles and Euler angle rates
or
rotation matrix and angular velocity vector

↔ 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

Geometry

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 Coordinate Transformations

Coordinate Transformation	Routine
– Latitudinal to/from Rectangular	– `LATREC` `RECLAT`
– 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`

Velocity Coordinate Transformations - 1

Coordinate Transformation	Jacobian (Derivative) Matrix Routine
Latitudinal to/from Rectangular	`DRDLAT` `DLATDR`
Planetographic to/from Rectangular	`DRDPGR` `DPGRDR`
R.A. Dec to/from Rectangular	`DRDLAT*` `DLATDR*`
Geodetic to/from Rectangular	`DRDGEO` `DGEODR`
Cylindrical to/from Rectangular	`DRDCYL` `DCYLDR`
Spherical to/from Rectangular	`DRDSPH` `DSPHDR`

* Jacobian matrices for the R.A and Dec to/from rectangular mappings are identical to those for the latitudinal to/from rectangular mappings

Velocity Coordinate Transformations - 2

- Example: transform velocities from rectangular to spherical coordinates using the SPICE Jacobian matrix routines. The SPICE calls that implement this computation are:

```
CALL SPKEZR  (  TARG,  ET,  REF,  CORR,  OBS,  STATE,  LT  )
CALL DSPHDR (  STATE(1), STATE(2),  STATE(3), JACOBI  )
CALL MXV        (  JACOBI,    STATE(4),  SPHVEL  )
```

- After these calls, the vector SPHVEL contains the velocity in spherical coordinates: specifically, the derivatives

  (d (r) / dt,   d (colatitude) / dt,   d (longitude) /dt)

- Caution: coordinate transformations often have singularities, so derivatives may not exist everywhere.
  - Exceptions are described in the headers of the SPICE Jacobian matrix routines.
  - SPICE Jacobian matrix routines signal errors if asked to perform an invalid computation.

Other Derived Geometric Quantities

- Illumination angles (phase, incidence, emission)
  - ILUMIN*
- Subsolar point
  - SUBSLR*
- Subobserver point
  - SUBPNT*
- Surface intercept point
  - SINCPT*
- Longitude of the sun (L_s), an indicator of season
  - LSPCN

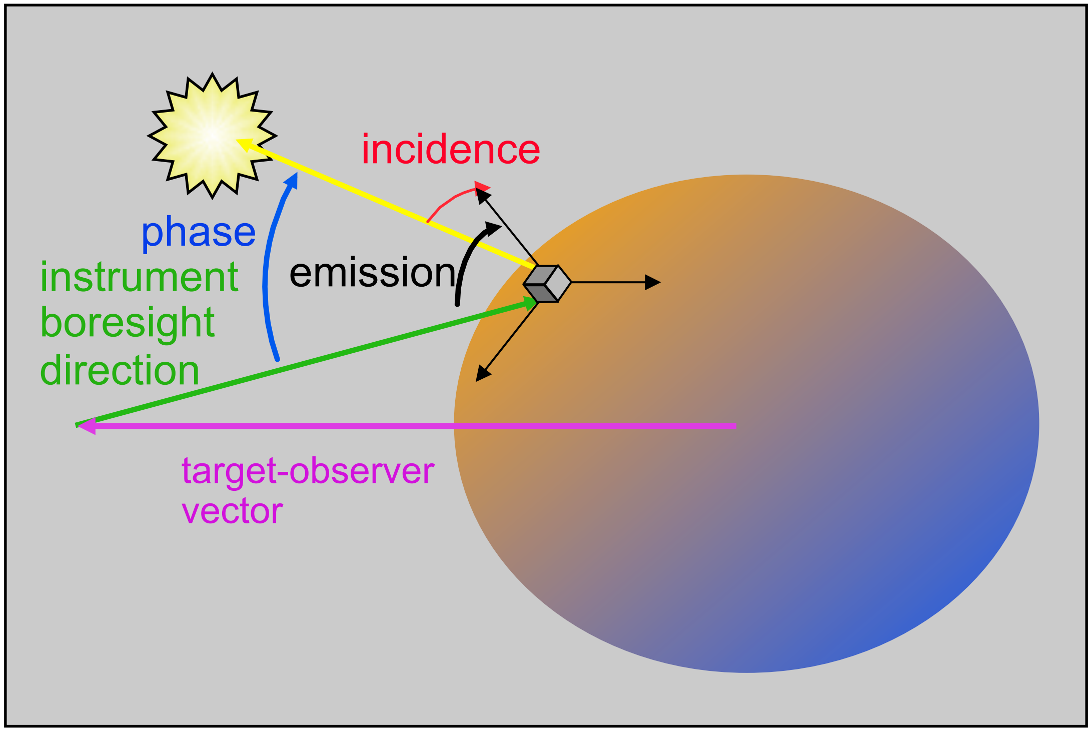
* These routines supercede the now deprecated routines ILLUM, SUBSOL, SUBPT and SRFXPT

Geometric Events

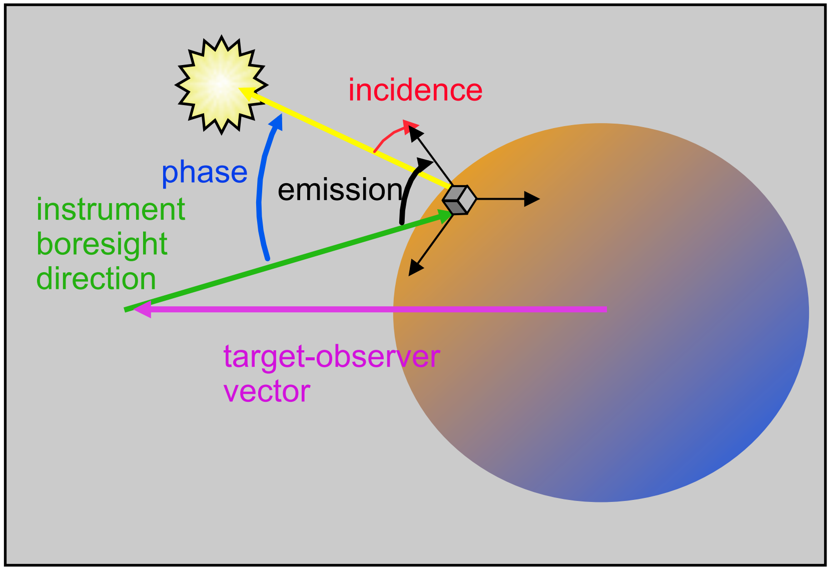
- The SPICE Geometry Finder (GF) subsystem can find times when the following events occur:
  - A specified coordinate of a position vector, sub-observer point, or ray-body surface intercept satisfies a given constraint
  - A specified occultation or transit is in effect
  - A specified target body or ray (e.g. direction to a specified star) is in the field of view (FOV) of a given instrument
  - The angular separation of two specified bodies as seen by a specified observer satisfies a given constraint
  - The distance between a specified target and observer satisfies a given constraint

Computing Illumination Angles

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



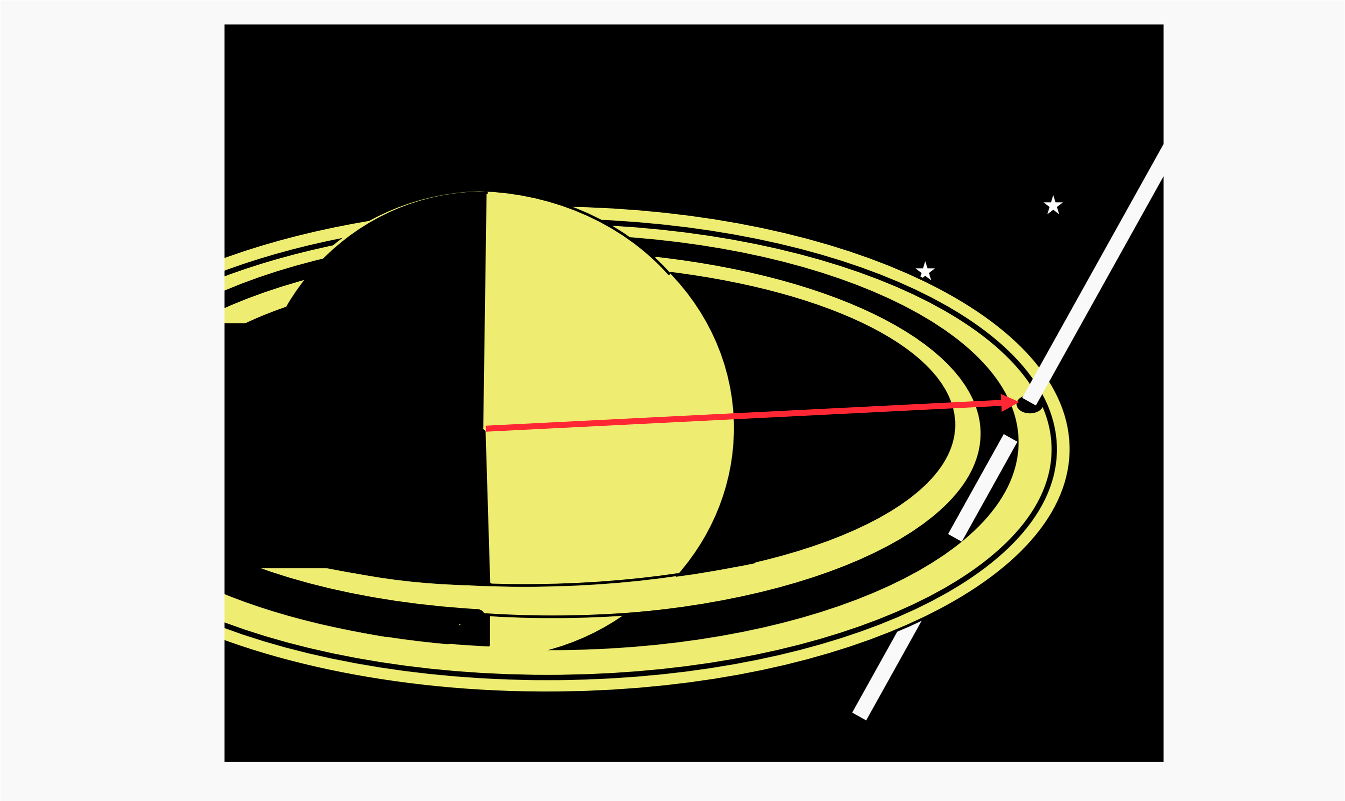
Computing Illumination Angles



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

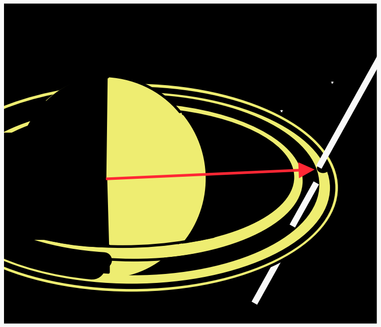
Computing Ring Plane Intercepts

- **Determine the intersection of the apparent line of sight vector between Earth and Cassini with Saturn’s ring plane and determine the distance of this point from the center of Saturn.**



Computing Ring Plane Intercepts-2

Navigation and Ancillary Information Facility



This computation is for the reception case; radiation is received at the earth at a given epoch “ET”.

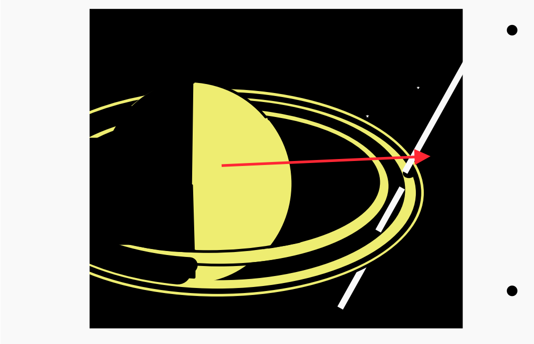
This simplified computation ignores the difference between the light time from Saturn to the observer and the light time from the intercept point to the observer.

The position and orientation of Saturn can be re-computed using the light time from earth to the intercept; the intercept can be re-computed until convergence is attained.

- CALL SPKEZR to get light time corrected position of spacecraft as seen from earth at time ET in J2000 reference frame SCVEC.
- CALL SPKEZR to get light time corrected position position of center of Saturn at time ET as seen from earth in J2000 frame SATCTR.
- CALL PXFORM to get rotation from Saturn body-fixed coordinates to J2000 at light time corrected epoch. The third column of this matrix gives the pole direction of Saturn in the J2000 frame SATPOL.
- CALL NVP2PL and use SATCTR and SATPOL to construct the ring plane RPLANE.
- CALL INRYPL to intersect the Earth-spacecraft vector SCVEC with the Saturn ring plane RPLANE to produce the intercept point X.
- CALL VSUB to get the position of the intercept with respect to Saturn XSAT (subtract SATCTR from X) and use VNORM to get the distance of XSAT from the center of Saturn.

Computing Ring Plane Intercepts-3

Navigation and Ancillary Information Facility



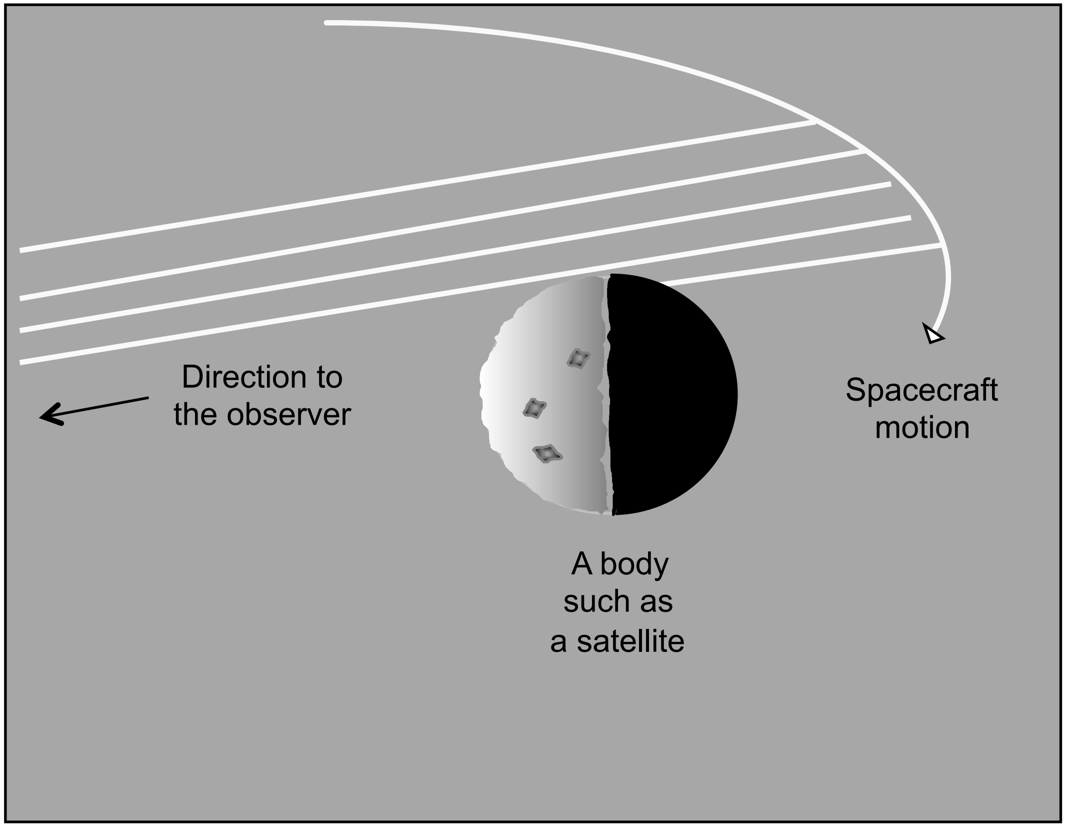
An alternate approach

- Create a dynamic frame with one axis pointing from earth to the light time corrected position of the Cassini orbiter. Use the CN correction for this position vector. (This gives us a frame in which the direction vector of interest is constant.)
- Temporarily change the radii of Saturn to make the polar axis length 1 cm and the equatorial radii 1.e6 km. This can be done either by editing the PCK or by calling BODVCD to fetch the original radii, then calling PDPOOL to set the kernel pool variable containing the radii to the new values. This flat ellipsoid will be used to represent the ring plane.
- Use SINCPT to find the intercept of the earth-Cassini ray with the flat ellipsoid. Use the CN correction. SINCPT returns both the intercept in the IAU_SATURN frame and the earth-intercept vector. Use VNORM to get the distance of the intercept from Saturn's center.
- Restore the original radii of Saturn. If PDPOOL was used to update the radii in the kernel pool, use PDPOOL again to restore the radii fetched by BODVCD.

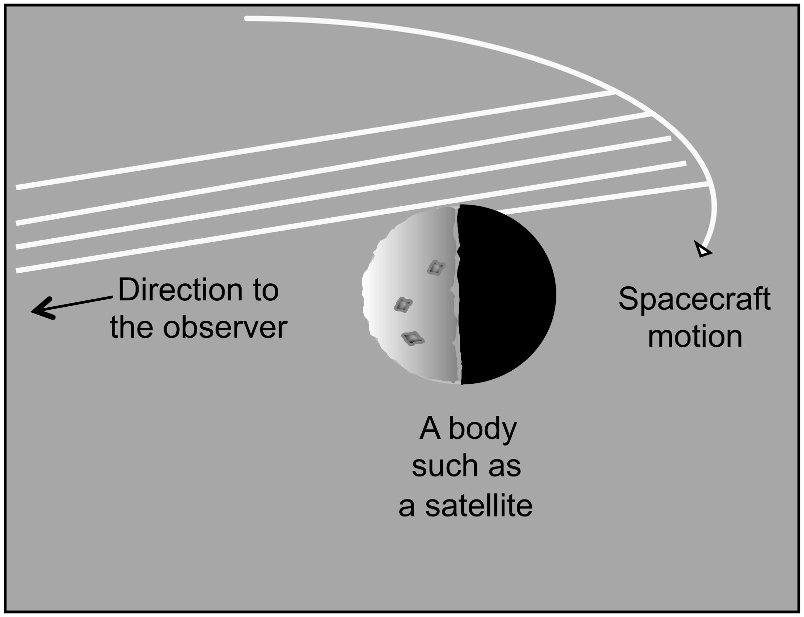
Computing Occultation Events

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



- 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 (for Mice, simply use an array). 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 (for Mice, an array).
  - 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."

GF Enhancements (Fall 2009)

- **NAIF is upgrading the Geometry Finder (GF) subsystem to support searches involving the following geometric quantities:**
  - **Eclipses**
  - **Range rates**
  - **Illumination angles**
  - **Body-centered phase angles**
  - **User-defined binary state quantities**
  - **User-defined scalar quantities**