

# **Wide-field Infrared Survey Explorer (WISE)**

## **Solar System Object Identification (SSOID) Subsystem Design Document**

**Version 2.0**

**28 May 2009**

**Prepared by: John W. Fowler**



**Infrared Processing and Analysis Center  
California Institute of Technology**

**WSDC D-D017**

**Concurred By:**

---

Roc Cutri, WISE Science Data Center Manager

---

Tim Conrow, WISE Science Data Center System Architect

---

John Fowler, WISE Science Data Center PRex Cognizant Engineer

---

Dave Tholen, WISE Solar-System Object Prediction Cognizant Engineer

## **Revision History**

<b>Date</b>	<b>Version</b>	<b>Author</b>	<b>Description</b>
9 October 2008	0.1	J. W. Fowler	Preliminary Draft
15 December 2008	1.0	J.W. Fowler	Initial Version
20 January 2009	1.1	J.W. Fowler	Added frame coordinates for testing unmatched SSOs to determine whether they are truly “missed”
12 March 2009	1.2-1.3	J.W. Fowler	Added option to skip one-way light time correction (for supporting tests using simulation), "-xL"; added array-limit specifications, "-cn", "-cx", "-rn", "-rx"; added output meta-data option, "-om"
24 March 2009	1.4	J.W. Fowler	Implemented R. Walker's derived-parameter algorithms
24 March 2009	1.5	J.W. Fowler	Implemented nulls in table files
8 May 2009	1.6	J.W. Fowler	Updated SIS numbers; added ssothmod; expanded testing discussion
28 May 2009	2.0	J.W. Fowler	Added planetary satellites and SSO “Type” codes

## Table of Contents

- 1 Introduction**
  - 1.1 Subsystem Overview**
    - 1.1.1 Requirements
    - 1.1.2 Liens
  - 1.2 Applicable Documents**
  - 1.3 Acronyms**
- 2 Input**
  - 2.1 SSOINIT Input**
    - 2.1.1 Control Input
    - 2.1.2 Orbital Elements Input
    - 2.1.3 FITS Input
  - 2.2 SSOID Input**
    - 2.2.1 Control Input
    - 2.2.2 FITS Input
    - 2.2.3 WISE Source Input
    - 2.2.4 Orbital Elements Input
    - 2.2.5 Planetary-Satellite Data Input
    - 2.2.6 NAMELIST Input
  - 2.3 SSOTHMODO Input**
    - 2.3.1 Control Input
    - 2.3.2 SSOID Input
- 3 Processing**
  - 3.1 SSOINIT Processing**
  - 3.2 SSOID Processing**
    - 3.2.1 SSO Input
    - 3.2.2 SSO Position Computation and WISE Association
    - 3.2.3 Thermal Model Processing
  - 3.3 SSOTHMODO Processing**
- 4 Output**
  - 4.1 SSOINIT Output**
    - 4.1.1 Orbital Element Subset Output
    - 4.1.2 Three-Epoch Ephemerides File
  - 4.2 SSOID Output**
    - 4.2.1 SSO/WISE Associations Output File
    - 4.2.2 SSO Meta-Data Output File
  - 4.3 SSOTHMODO Output**
    - 4.3.1 SSO/WISE Associations Output File
- 5 Testing**
  
- Appendix A SSO Orbit and Apparent Position Computation**
- Appendix B Mapping Position Error Covariance Matrices**

# 1 Introduction

## 1.1 Subsystem Overview

This document presents the requirements, design, algorithms, and state of implementation of the SSOID (Solar System Object Identification) subsystem of the WSDC data processing system. Two SSOID modules run in the scan pipeline with different processing scopes: SSOINIT processes a half-orbit scan to produce a subset of the SSO data base appropriate for the sky covered in that scan, and SSOID runs on individual framesets. The third module, SSOTHMOD, runs offline as needed to update thermal-model results.

### 1.1.1 Requirements

The SSOID subsystem is required to predict the positions of known solar system objects in WISE framesets and to associate these predictions with WISE point sources. The Level 4 requirements supported by this processing are as follows.

L4WSDC-027: The WSDC shall identify and compile a listing of known solar system objects that are positionally associated with source extractions in the WISE single -epoch image frames.

L4WSDC-028: The solar system objects associated with WISE single-epoch extractions shall include asteroids, comets, planets, and planetary satellites.

### 1.1.2 Liens

- Processing of WMOPS files by ssothmod has not yet been implemented.
- The SIS for the planetary satellite data file (sso06) has not been written.

## 1.2 Applicable Documents

This subsystem conforms to the specifications in the following project documents:

- WISE Science Data Center Functional Requirements Document, WSDC D-R001
- WISE Science Data System Functional Design, WSDC D-D001
- Software Management Plan, WSDC D-M003
- SIS sso01: SSO Orbital Elements File, WSDC D-I145
- SIS sso02: SSO ssoinit/ssoid Intermediate File, WSDC D-I146
- SIS sso03: SSO Three-Epoch Ephemerides File, WSDC D-I147
- SIS sso04: SSO/WISE Associations File, WSDC D-I148
- SIS sso05: SSO Meta-Data File, WSDC D-I149
- SIS sso06: SSO Planetary-Satellite Data File, WSDC D-I151
- SIS pht01: WPHot Output Photometry Table, WSDC D-I107
- Bowell, E., Harris, A.W., & Lumme, K., 1988, A two-parameter magnitude system for asteroids, *Icarus*
- Harris, A.W. 1998. A thermal model for near-Earth asteroids. *Icarus* 131, 291-301
- Everhart, E., 1984. An efficient integrator that uses Gauss-Radau spacings, *Dynamics of Comets: Their Origin and Evolution*, Proceedings of IAU Colloq. 83

## 1.3 Acronyms

2MASS	Two-Micron All-Sky Survey
FRD	Functional Requirements Document
SDS	Subsystem Design Specification
SIS	Software Interface Specification
SSO	Solar System Object
SSOID	Solar System Object Identification (subsystem and processor)
SSOINIT	Solar System Object Initialization processor
SSOTHMOD	Solar System Object Thermal Model processor
W1	WISE wavelength channel 1, 3.3 microns
W2	WISE wavelength channel 2, 4.7 microns
W3	WISE wavelength channel 3, 12 microns
W4	WISE wavelength channel 4, 23 microns
WISE	Wide-field Infrared Survey Explorer
WMOPS	WISE Moving-Object Processing System
WSDC	WISE Science Data Center
WSDS	WISE Science Data System

## 2 Input

### 2.1 SSOINIT Input

#### 2.1.1 Control Input

The `ssoinit` module reads control input in the form of command-line parameters which are shown in its tutorial display:

```
ssoinit: Solar-System Object Initialization Utility vsn 1.2 A90401

usage:  ssoinit <flags> <specifications>

where <flags> <specifications> must be:

    -i InputFNam   (Input orbital elements file name)
    -o OutputFNam  (Output orbital elements subset file name)
    -f1 FITSnam1   (Name of first-frame FITS file)
    -f2 FITSnam2   (Name of ~mid-frame FITS file)
    -f3 FITSnam3   (Name of last-frame FITS file)
    -w Window      (Half width of angle window perpendicular
                   to the scan axis, deg; default = 2)
    -i2 SSolist    (Name of file containing a list of names
                   of SSOs to be output independently of
                   whether they are in the scan swath)
    -o2 SSOephem   (Name of output ephemeris file for SSOs
                   specified via "-i2")
```

The first five specifications above are required.

The input orbital elements file contains the orbital elements in the format used by the 2MASS DTPGM program. The output orbital elements subset file contains these same elements for the subset of SSOs that fall within the scan swath defined by the epochs of the first and third FITS files (swath endpoints) and the `Window` parameter. In addition to these SSOs' orbital elements, this file also contains six intermediate parameters per SSO to save time in the `ssoid` module. These six parameters are evaluated at the epoch of the second FITS file. Thus for every SSO in the subset file, there are two lines of ASCII text: the line from the input orbital elements file and a line containing the three components of the spacecraft-centered unit vector to the SSO, the mean motion, the topocentric distance, and the eccentric anomaly.

The second FITS file should be chosen to have an epoch near the center of the scan. This is because its epoch is used as the representative approximate time of the scan and because the cross products of its image-center unit vector with those of the other two FITS files are used to establish the scan axis unambiguously. This depends on the mission design employing survey scans that are always closer to 180 degrees long than to 270 degrees (there is no assumption that scans are not much shorter than 180 degrees).



### 2.1.2 Orbital Elements Input

The `ssoinit` module reads orbital elements input in the format used by the 2MASS DTPGM program (SIS sso01: SSO Orbital Elements File, WSDC D-I145)

### 2.1.3 FITS Input

The `ssoinit` module reads three FITS files, which must be in the correct order chronologically. The first and third must correspond to the scan endpoints, and the second should be close to the middle of the scan. The three J2000 spacecraft vectors will be obtained from the corresponding three FITS files. The keywords are SUN2SCX, SUN2SCY, and SUN2SCZ.

## **2.2 SSOID Input**

### 2.2.1 Control Input

#### 2.2.1.1 Primary Command-Line Parameters

The `ssoid` module reads control input in the form of primary command-line parameters which are shown in its tutorial display:

ssoid: Solar-System Object Identification Program vsn 2.0 A90528

usage: ssoid <flags> <specifications>

where <flags> <specifications> must be:

-i1 InputFNam1 (Input full orbital elements file name)  
-i2 InputFNam2 (Input subset orbital elements file name)  
-i3 InputFNam3 (Input planetary-satellite data file name)  
-w WISEnam (Name of WISE source file)  
-f FITSnam (Name of FITS file for WISE frame)  
-o OutputFNam (Output WISE/SSOID associations file name)  
-w3 W3Tnam (Name of W3vsT FITS file)  
-w4 W4Tnam (Name of W4vsT FITS file)  
-z3 ZpW3 (Photometric zero point for W3 to convert mag to w/cm<sup>2</sup>; default = -38.24; optional)  
-z4 ZpW4 (Photometric zero point for W4 to convert mag to w/cm<sup>2</sup>; default = -41.75; optional)  
-om MDnam (Meta-Data output file name; optional)  
-n NLnam (Namelist input file name; optional)  
-c ChiSqMax (Maximum 2-D chi-square value to accept association; optional; default = 16)  
-d DistMax (Distance for coarse search for matches, arcsec; optional; default = 10)  
-m MaxAng (Maximum angle from frame center to SSO to consider, deg; optional; default = 2)  
-h HalfDiag (Half-diagonal of frame, arcsec; optional; default = 2000)  
-xL (Omit one-way light time correction [to support testing with simulation data]; optional; default = F)  
-xD (Omit distortion when transforming from U-scan to band-frame coordinates; optional; default = F)  
-cn ColMin (Minimum SSO band-frame column coordinate to be considered inside array; optional; default = 1)  
-cx ColMax (Maximum SSO band-frame column coordinate to be considered inside array; optional; default = 1016)  
-rn RowMin (Minimum SSO band-frame row coordinate to be considered inside array; optional; default = 1)  
-rx RowMax (Maximum SSO band-frame row coordinate to be considered inside array; optional; default = 1016)  
-cr ChSqRS (Chi-square maximum for W3/W4 diameter average above which uncertainty will be inflated; optional; default = 9)  
-t0 TssBase (Base Temperature for Tss tables; optional; default = 120)  
-mu MaxUnc (Maximum Uncertainty of WISE source position for SSO matching, 1-sigma, arcsec, either axis; optional; default = 5)

Either "-i1" or "-i2" must be specified but not both; "-w", "-f", "-o", "-w3", and "-w4", are required.

The `ssoid` module can operate with either the full orbital-elements file or the subset file prepared by the `ssoinit` module; one or the other of these, but not both, must be specified via the “-i1” or “-i2” options, respectively. The latter allows a reduction in CPU time of typically a factor of 30. Planetary satellites may optionally be specified by giving the name of their data file via the “-i3” flag. This module processes a single frameset, and so a single FITS file is needed, and the corresponding list of point sources is also needed. The former is used to establish the mapping between celestial coordinates and WISE U-scan and band-frame coordinates. The latter file supplies the WISE point sources to be sought for matches to the SSO positions. The “-o” option specifies the name of the main output file, which has entries for all SSOs that fall into the band-frame area and includes WISE information and derived parameters for SSOs with WISE matches (SIS sso04, WSDC D-I148).

The components of the sun-to-spacecraft position and velocity vectors are obtained from the FITS header. The keywords are `SUN2SCX`, `SUN2SCY`, and `SUN2SCZ` for position, and `SCVELX`, `SCVELY`, and `SCVELZ` for velocity. These are J2000 vectors in units of AU and AU/day, respectively, at the frame epoch, which is also the epoch of the FITS file. The latter may be for any WISE band; whatever band is used determines the band-frame mapping of SSO position coordinates for deciding whether each tested SSO really fell inside the array area.

The `w3vst` and `w4vst` files are lookup tables for the thermal model in FITS format specified via the “-w3” and “-w4” flags. The use of these tables requires converting WISE magnitudes to watts per square centimeter; photometric zero points are used for this purpose and may optionally specified via the “-z3” and “-z4” flags for W3 and W4, respectively. The lookup tables employ two independent variables, phase angle horizontally and sub-solar temperature vertically. The lowest phase angle is always zero degrees, and the spacing is always 1 degree; the number of entries is not constrained. The lowest temperature is currently 120 kelvins, with a spacing of 1 kelvin; the number of entries is not constrained. If a different base temperature is used, it may be specified via the “-t0” flag.

All other command-line specifications are optional. The “-om” option allows an output meta-data file to be generated; this contains information regarding the number of objects processed, match rate, etc. (SIS sso05, WSDC D-I149). An example is shown below.

```

\ ssoid Meta-Data (SSMD) file - WISE/SSO association parameters
\ Generated by ssoid vsn 1.51 A90331 on 27-04-09 at 9:01:02
| name | band | hdrname | type | value | comment |
| c | i | c | c | c | c |
ssoid:ChiSqMx 0 CHISQMAX r 1.600000E+01 Max 2-D position chi-square to accept WISE/SSO match
ssoid:DistMax 0 DISTMAX r 1.000000E+01 Coarse search distance in arcsec
ssoid:Nin 0 NIN i 4171 No. of orbital element sets read
ssoid:Nsso 0 NSSO i 37 No. of SSOs in WISE frame
ssoid:Nmch 0 NMCH i 5 No. of WISE/SSO matches
ssoid:NCnfzd 0 NCNFZD i 0 No. of SSO-WISE confused matches
ssoid:Nmiss 0 NMISSE i 32 No. of unmatched SSOs
ssoid:RatMch 0 RATMCH r 1.351351E-01 Match rate, NMch/Nsso
ssoid:AvgDX 0 AVGDY r -3.349664878845215E+00 Average X offset in arcsec
ssoid:AvgDY 0 AVGDY r -1.279651689529419E+00 Average Y offset in arcsec
ssoid:SigDX 0 SIGDX r 4.853681981853966E+00 Sigma of X offset in arcsec
ssoid:SigDY 0 SIGDY r 3.825556837101503E+00 Sigma of Y offset in arcsec
ssoid:AvgChSq 0 AVGCHSQ r 5.106672966480255E+00 Reduced 2-D chi-square
ssoid:Nstat 0 NSTAT i 5 No. of clean matches for stats
ssoid:nDC 0 NDC i 0 No. of disconnected SSOs
ssoid:NnoW3W4 0 NNOW3W4 i 0 No. of fruitless matches

```

The “-n” option allows a namelist file to be specified (see section 2.2.6). The “-mu” option allows a maximum position uncertainty for WISE sources to be specified; WISE sources with an uncertainty larger on either axis will be “penalized”, which means that if more than one SSO/WISE match is available, the best non-penalized WISE source will take precedence over any penalized WISE source. This prevents sources with excessive position uncertainties from gaining unfair advantage in the position chi-square test. Penalized-source matches are re-assigned chi-square scores greater than the nominal threshold; a penalized source may participate in a match only if there is no non-penalized source within threshold. The other options are as documented in the tutorial display.

### 2.2.1.2 Auxiliary Command-Line Parameters

In addition to the command-line parameters shown in the tutorial display, `ssoid` accepts several other command-line specifications associated with deprecated options and support for offline analysis. These are listed below.

<u>Flag</u>	<u>Specification</u>
-o2	Name of “Detection Diagram” output file
-s3	Name of uncertainty file for W3vsT (“-w3”)
-s4	Name of uncertainty file for W4vsT (“-w4”)
-s	Three components of the J2000 sun-to-spacecraft vector (AU)
-v	Three components of the J2000 spacecraft velocity vector (AU/day)

The Detection Diagram file is an ASCII text file containing information for use in offline analysis; a program named `SSODD` that runs in Microsoft Windows can use this file to produce diagrams of each SSO in the frame against a 21x21-pixel postage stamp from the image specified via the “-f” option, along with all WISE sources in the postage stamp. Each SSO and WISE source is represented by a set of three concentric ellipses that show the position uncertainty contours at levels of 1, 2, and 3 sigma. WISE sourceID numbers are also displayed at or near the center of the corresponding ellipses. Examples of such diagrams are shown in section 5 (Testing).

The “-s3” and “-s4” options were used in a previous version to specify uncertainties in the thermal-model tables of W3 and W4 flux. The current version derives the uncertainties in thermal-model parameters exclusively from the WISE photometric uncertainties, treating the thermal model as a context within which the only uncertainty pertains to how well the WISE photometry has been mapped into that context. Specifying either of these options produces a warning message that they are no longer needed or used.

The “-s” and “-v” options are also no longer needed, but if they are specified, they *are* used. These override the corresponding information in the FITS header of the “-f” file.

### 2.2.2 FITS Input

The ssoid module reads FITS input to obtain coordinate mapping information, the J2000 spacecraft position and velocity vectors, and the epoch of the observation.

### 2.2.3 WISE Source Input

The ssoid module reads a table file of WISE point sources which are to be matched with the SSO predictions (SIS pht01: WPHot Output Photometry Table, WSDC D-I107 ).

### 2.2.4 Orbital Elements Input

The ssoid module reads orbital elements input in one of two optional ways: (a.) the format used by the ssoinit module (SIS sso01: SSO Orbital Elements File, WSDC D-I145), which covers the full set of known SSOs; (b.) the subset file generated by the ssoinit module (SIS sso02: ssoinit/ssoid Intermediate File, WSDC D-I146).

### 2.2.5 Planetary-Satellite Data Input

The ssoid module optionally reads a file containing data needed to predict the apparent positions of planetary satellites (SIS sso06: SSO Planetary-Satellite Data File, WSDC D-I151).

### 2.2.6 NAMELIST Input

The ssoid module optionally reads NAMELIST input to control table-file column names in the input WISE source table file. The name of the NAMELIST is `ssoidin`. The parameters defined in the NAMELIST are as follows. Default values are shown in nominal case, but the processing is case-insensitive.

Name	Description	Dim	Type	Units	Default
<code>hdrNsrcs</code>	Name of the header parameter that specifies the number of sources	1	C*25	-	'Nsrc'
<code>NamType</code>	Array of 1-character "Type" IDs for SSOs (see SIS sso04, WSDC-I148)	50	C*1	-	'A','C', 'P','T', 'U',45*''
<code>NType</code>	Number of each SSO "Type" in the current Orbital Elements file	50	I*4	-	435634, 2593,8,3, 9999999,45*0
<code>tblRA</code>	Name of the column parameter that specifies RA	1	C*25	-	'ra'

Name	Description	Dim	Type	Units	Default
tblDec	Name of the column parameter that specifies Dec	1	C*25	-	'dec'
tblSigX	Name of the column parameter that specifies 1-sigma RA uncertainty	1	C*25	-	'sigra'
tblSigY	Name of the column parameter that specifies 1-sigma Dec uncertainty	1	C*25	-	'sigdec'
tblSigXY	Name of the column parameter that specifies RA/Dec co-sigma uncertainty	1	C*25	-	'sigradec'
tblW1nois	Name of the column parameter that specifies W1 noise	1	C*25	-	'w1sigsk'
tblW2nois	Name of the column parameter that specifies W2 noise	1	C*25	-	'w2sigsk'
tblW3nois	Name of the column parameter that specifies W3 noise	1	C*25	-	'w3sigsk'
tblW4nois	Name of the column parameter that specifies W4 noise	1	C*25	-	'w4sigsk'
tblW1m	Name of the column parameter that specifies W1 magnitude	1	C*25	-	'w1mpro'
tblW1sgm	Name of the column parameter that specifies W1 magnitude uncertainty	1	C*25	-	'w1sigmpro'
tblW2m	Name of the column parameter that specifies W2 magnitude	1	C*25	-	'w2mpro'
tblW2sgm	Name of the column parameter that specifies W2 magnitude uncertainty	1	C*25	-	'w2sigmpro'
tblW3m	Name of the column parameter that specifies W3 magnitude	1	C*25	-	'w3mpro'
tblW3sgm	Name of the column parameter that specifies W3 magnitude uncertainty	1	C*25	-	'w3sigmpro'
tblW4m	Name of the column parameter that specifies W4 magnitude	1	C*25	-	'w4mpro'
tblW4sgm	Name of the column parameter that specifies W4 magnitude uncertainty	1	C*25	-	'w4sigmpro'

## 2.3 SSOTHMODO Input

### 2.2.1 Control Input

The `ssoinit` module reads control input in the form of command-line parameters which are shown in its tutorial display:

```
ssothmod: Solar-System Object Thermal Model Update Program vsn 1.0  A90501

usage:  ssothmod <flags> <specifications>

where <flags> <specifications> must be:

    -i1 InputFNam1  (Input SSOID file name)
    -i2 InputFNam2  (Input WMOPS file name)
    -o  OutputFNam  (Output file name)
    -w3 W3Tnam      (Name of W3vsT FITS file)
    -w4 W4Tnam      (Name of W4vsT FITS file)
    -z3 ZpW3        (Photometric zero point for W3 to convert
                    mag to w/cm^2; default = -38.24; optional)
    -z4 ZpW4        (Photometric zero point for W4 to convert
                    mag to w/cm^2; default = -41.75; optional)
    -cr ChSqRS      (Chi-square maximum for W3/W4 diameter
                    average above which uncertainty will be
                    inflated; optional; default = 9)
    -t0 TssBase     (Base Temperature for Tss tables; optional;
                    default = 120)

Either "-i1" or "-i2" must be specified but not both; "-o",
"-w3", and "-w4", are required.
```

Except for the first two, these command-line specifications use the same flags and have the same meanings as in the `ssoid` module (section 2.2.1.1). The “-i1” option specifies an `ssoid` output file (“-o” in `ssoid`) as input to `ssothmod`, and the “-i2” option specifies a WMOPS output file (SIS TBD) as input. One and only one of these two is required. The output file has the same format as the input `ssoid` file, SIS sso04, WSDC D-I148. (Note: currently the WMOPS option is not implemented.)

## 3 Processing

### 3.1 SSOINIT Processing

The `ssoinit` module begins by reading the command-line input, verifying that all required specifications have been made, and verifying that all input files exist. If meta-data output was

requested, then that file is opened and initialized. If any unrecognized specifications were made, an error message is generated and processing terminates with a 64 return code.

The three FITS input files are opened, and the following header parameters are read for each and stored in three-element arrays: `CRVAL1`, `CRVAL2`, `SUN2SCX`, `SUN2SCY`, `SUN2SCZ`, and `DATE_OBS`. The last parameter string is converted to Julian Date and used as the corresponding frame epoch. The other parameters are used to compute the scan axis and azimuthal scan swath; since these values are the RA and Dec of each frame center, corresponding unit vectors can be constructed:

$$\begin{aligned} X_i &= \cos \alpha_i \cos \delta_i \\ Y_i &= \sin \alpha_i \cos \delta_i \\ Z_i &= \sin \delta_i \end{aligned} \tag{1}$$

where  $i = 1$  to 3 for the three epochs  $T_i$ .

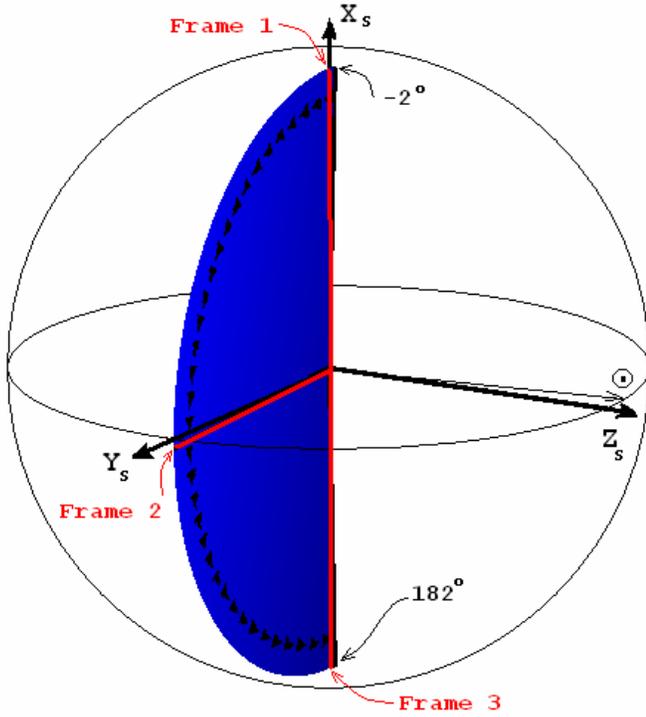
If three-epoch ephemerides were requested for a list of SSOs (see section 2.1.1, command-line options “-i2” and “-o2”), then the input file containing the object names is opened, the number of lines is counted, memory is allocated for an array of the 35-character names and for a logical flag for each, `GotEphem`, that indicates whether the object has been found in the orbital-elements input and processed, the file is rewound, the names are read into memory, and `GotEphem` is set to `F` for each. The output file is then opened, and the header is written.

The scan axis is computed by averaging two estimates, each obtained via the cross product of two unit vectors shown in Equation 1:

$$\begin{aligned} \hat{U}_i &\equiv (X_i, Y_i, Z_i) \\ \hat{S}_1 &= \frac{\hat{U}_1 \times \hat{U}_2}{|\hat{U}_1 \times \hat{U}_2|} \\ \hat{S}_2 &= \frac{\hat{U}_2 \times \hat{U}_3}{|\hat{U}_2 \times \hat{U}_3|} \\ \hat{S} &= \frac{\hat{S}_1 + \hat{S}_2}{|\hat{S}_1 + \hat{S}_2|} \\ \Delta\theta &= \sin^{-1}(\hat{S}_1 \times \hat{S}_2) \end{aligned} \tag{2}$$

If  $\Delta\theta > 3$  arcsec, a warning message is issued, and the scan-swath half-width  $W$  (default 2 degrees; see “-w” in section 2.1.1) is increased by  $\Delta\theta$ .

The scan axis is defined as the pole of the “scan” coordinate system, denoted by a subscript “s”:



$$\begin{aligned}
 \hat{Z}_s &= \hat{S} \\
 \hat{X}_s &= \hat{U}_1 \\
 \hat{Y}_s &= \hat{Z}_s \times \hat{X}_s
 \end{aligned} \tag{3}$$

The  $X_s$  axis points to the center of Frame 1, which is thus the zero point of the azimuthal angle. The “scan swath” is the area whose azimuthal center line is in the  $X_s Y_s$  plane (hence perpendicular to the scan axis) with a width of  $\pm W$  and extending from  $-W$  (i.e., below the  $X_s$  axis, preceding Frame 1 along the scan) to  $+W$  past Frame 3. The illustration on the left shows a scan center line from the north ecliptic pole to the south ecliptic pole; this is the scan-system azimuthal range from  $0^\circ$  to  $180^\circ$ . The “scan swath” therefore extends from  $-2^\circ$  to  $182^\circ$  in azimuth and is  $4^\circ$  full width in elevation. For a morning

launch, the  $Z_s$  scan axis should be generally very close to the spacecraft-sun vector as shown in the illustration, but it may be a few degrees off to allow some variation in which ecliptic meridian is chosen to be scanned. For an evening launch, the two vectors should be close to anti-parallel. In either case, the scan axis should lie very close to the ecliptic plane.

The unit vectors in Equation 3 can be used to form a direction-cosine matrix which is a transformation matrix for computing scan-system coordinates corresponding to any RA and Dec,  $(\alpha, \delta)$ , as follows, where unit vectors will be denoted  $u$ , subscript “c” indicates “celestial” (J2000 equatorial) coordinates, and subscript “s” indicates “scan-system” coordinates:

$$\begin{aligned}
 u_{c1} &= \cos \alpha \cos \delta \\
 u_{c2} &= \sin \alpha \cos \delta \\
 u_{c3} &= \sin \delta \\
 \begin{pmatrix} u_{s1} \\ u_{s2} \\ u_{s3} \end{pmatrix} &= \begin{pmatrix} X_{s1} & X_{s2} & X_{s3} \\ Y_{s1} & Y_{s2} & Y_{s3} \\ Z_{s1} & Z_{s2} & Z_{s3} \end{pmatrix} \begin{pmatrix} u_{c1} \\ u_{c2} \\ u_{c3} \end{pmatrix} \\
 \theta &= \tan^{-1} \left( \frac{u_{s2}}{u_{s1}} \right) \\
 \phi &= \sin^{-1}(u_{s3})
 \end{aligned} \tag{4}$$

where the azimuthal and elevation angles are  $\theta$  and  $\phi$ , respectively. The scan swath is the set of all points satisfying the inequalities:

$$\begin{aligned} -W &\leq \phi \leq W \\ -W &\leq \theta \leq \theta_3 + W \end{aligned} \tag{5}$$

where  $\theta_3$  indicates the azimuthal coordinate of Frame 3. The criterion defining whether any given point on the sky is inside the scan swath is simply whether its azimuthal and elevation angles satisfy these two inequalities.

Once the transformation matrix in Equation 4 has been set up, the processing involves computing  $(\alpha, \delta)$  for each SSO at each of the three epochs, computing the corresponding  $(\theta, \phi)$ , and applying the test in Equation 5. Objects satisfying that test at any of the three epochs are written to the main output file (specified via “-o” on the command line; see section 2.1.1). In addition, objects that straddle the scan swath in either angle over any two epochs are also written to the main output file. Such output involves the data line from the input orbital elements file, followed by a line containing the following six numbers evaluated at the second epoch: the three components of the spacecraft-centered unit vector to the SSO, the mean motion, the topocentric distance, and the eccentric anomaly.

If output three-epoch ephemerides were selected (the “-i2” and “-o2” options in section 2.1.1), then each object name is checked against the input list of names, and if it is found, then the information shown at the end of section 2.1.1 is written to the output ephemerides file.

All that remains is the computation of  $(\alpha, \delta)$  for each SSO at each epoch. This is performed in the same way for each object/epoch combination, and it is performed in both modules, `ssoinit` and `ssoid`, so we will describe the case for a single object and a single epoch in Appendix A. There is actually one slight difference between epoch 1 and the other two epochs: once the eccentric anomaly has been found iteratively for epoch 1, it becomes the first estimate for the iteration at epoch 2; similarly, epoch 2 provides the first estimate for epoch 3. This is done to save CPU time by converging the iterative solution for the eccentric anomaly faster, but since it involves merely a better starting estimate for iterations 2 and 3, its interest is strictly computational efficiency and will not be discussed further.

### 3.2 SSOID Processing

The `ssoid` module begins by reading the command-line input, verifying that all required specifications have been made, and verifying that all input files exist. If meta-data output was requested, then that file is opened and initialized. If any unrecognized specifications were made, an error message is generated and processing terminates with a 64 return code.

The FITS input file is opened, and the following header parameters are read: `CRVAL1`, `CRVAL2`, `CRPIX1`, `CRPIX2`, `CD1_1`, `CD1_2`, `CD2_1`, `CD2_2`, `SUN2SCX`, `SUN2SCY`, `SUN2SCZ`, `SCVELX`,

SCVELY, SCVELZ, and DATE\_OBS. The last parameter string is converted to Julian Date and used as the frame epoch; the parameters with names beginning with “c” are used in the mapping of RA and Dec to U-Scan coordinates and band-frame coordinates via the PRex subroutines j2k2us, sete2u, setus, and u2fnd2; the parameters with names beginning with “s” are used for topocentric corrections for apparent SSO positions on the sky.

A file containing orbital elements is required; this may be the “full” set (“-o1”), or the “subset” generated by ssoinit (“-o2”). The subset contains only objects in the scan swath, and for each such object, an additional data line exists which contains the spacecraft-centered J2000 unit vector to the object, the mean motion, the topocentric distance, and the eccentric anomaly, all evaluated for the epoch of a frame near the center of the scan being processed (or at least between the first and last frames of the scan).

### 3.2.1 SSO Input

Memory is allocated for arrays that will hold SSO information. The size necessary for these arrays to hold all relevant SSOs is not known in advance, and so an initial size of 1000 is used, and as SSOs are found to be close to the frame area and added to these arrays, if more than 1000 are found (extremely unlikely), the memory is released, the array size is doubled (this may be performed as many times as necessary), and execution reverts to the memory allocation step. Any SSO processing that had been performed is abandoned, and execution begins anew. It is not expected that this recovery procedure will ever be needed in practice, but a warning message will be issued if it ever is used. The arrays store the following information for each SSO found to be inside the array area of the WISE band indicated in the FITS file:

<i>X, Y:</i>	U-scan coordinates (arcsec)
<i>VarX, VarY, VarXY:</i>	Position error covariance matrix elements
<i>ObjID:</i>	35-character object name
<i>ObjType:</i>	1-character “Type” name
<i>RA, Dec:</i>	Apparent J2000 RA and Dec (deg)
<i>Rhelio:</i>	Heliocentric distance (AU)
<i>Rtopo:</i>	Topocentric distance (AU)
<i>Phase:</i>	Phase angle (deg)
<i>Vmag:</i>	Visual magnitude
<i>Mu:</i>	Proper motion (arcsec/s)
<i>Theta:</i>	Direction of proper motion (deg E of N)
<i>ErrMaj, ErrMin:</i>	Semimajor and semiminor axes of uncertainty ellipse (arcsec, 1-sigma)
<i>ErrAng:</i>	Direction of ErrMaj axis (deg E of N)
<i>ChiSq:</i>	WISE Position-match 2-D chi-square
<i>Nmatch:</i>	No. WISE sources with acceptable position-match chi-squares
<i>WISEnum:</i>	WISE source ID for best position match
<i>P(3):</i>	Orbital P vector
<i>Q(3):</i>	Orbital Q vector

<i>PerD:</i>	Orbital perihelion distance (AU)
<i>Ecc:</i>	Orbital eccentricity
<i>PerT:</i>	Time of perihelion passage (JD)
<i>Err:</i>	Orbital element quality estimate
<i>H:</i>	Absolute magnitude
<i>G:</i>	Slope parameter

The P vector is the unit vector from the Sun toward perihelion in J2000; the Q unit vector is orthogonal to the P vector in the orbital plane, right-handed about the orbital angular momentum vector. H and G, the absolute magnitude and slope parameter, are used in computing the phase-dependent visual magnitude using the asteroid phase function developed by Bowell, Harris, and Lumme. The following processing is performed for each SSO in the input file.

### 3.2.2 SSO Position Computation and WISE Association

If the subset orbital input is being used, then the apparent-position unit vector for the SSO is read and used in a dot product with the array-center unit vector, and if the result is less than  $\cos(W)$ , the SSO is immediately discarded.  $W$  is the scan-swath half-width (default 2 degrees; see “-w” in section 2.1.1, or “-m” in section 2.2.1, where it is called `MaxAng`). The mean motion is also read from the subset input or computed from the full input.

If the command-line specifications included “-xL”, then no one-way light-time corrections are made to the SSO orbital solution (this supports testing with simulation data for which this correction was not made). Otherwise, if the subset orbital input is being used, then the topocentric distance is already known to sufficient accuracy, and the one-way light-time correction can be made on the first orbital solution, making the second solution slightly more accurate (this accuracy boost has never been observed to be larger than a milli-arcsec, but it is used since it is available).

The standard orbital solution, including topocentric corrections, is then obtained as described in Appendix A. If the “-i3” command-line option was used, then planetary satellites are processed in their own specific manner to arrive at the same information. This provides the  $(\alpha, \delta)$  for the SSO at the frame epoch, along with the corresponding unit vector. The latter is used in a dot product with the frame-center, and if the result is less than  $\cos(W)$ , the SSO is discarded as described two paragraphs back. Otherwise, the U-scan coordinates of the SSO are obtained via the `PRex` subroutine `j2k2us`. The U-scan system is Cartesian in arcsecond units, right-handed, and with the Y axis pointing north. If either coordinate is greater than `HalfDiag` (default 2000 arcsec; see “-h” in section 2.2.1). Otherwise the SSO is recorded in the arrays described above, with the `ChiSq`, `NMatch`, and `WISEnum` entries set to -9.9, 0, and 0, respectively.

After reading and processing the input file(s) completely, the number of SSOs retained in memory is checked; if this is zero, only the number of WISE sources is read from the header of the WISE source file, and the rest of the WISE source I/O is skipped. Otherwise this latter

number is used to allocate memory for WISE sources in arrays containing the following information for each source.

<i>RA, Dec:</i>	J2000 RA and Dec (deg)
<i>X, Y:</i>	U-scan coordinates (arcsec)
<i>VarX, VarY, VarXY:</i>	Position error covariance matrix elements
<i>Noise(4):</i>	Noise in the four WISE bands
<i>Wmag(4):</i>	Magnitudes in the four WISE bands
<i>SigWmag(4):</i>	Magnitude uncertainties in the four WISE bands, 1-sigma
<i>SSOnum:</i>	Index of SSO with which this WISE source is matched

SSOnum is initialized to zero. The average noise in each WISE band is computed for use in output SSO records for which no WISE association was found. Then a loop over all SSOs in memory is performed; for each SSO the following match processing done.

A loop over all WISE sources in memory is performed; for each WISE source, a position comparison to the SSO is made in U-scan coordinates. The SSO position uncertainty parameters are mapped to a U-scan error covariance matrix as described in Appendix B. Subscripts “w” and “o” will be used to indicate WISE and SSO objects, respectively. First a coarse test is made:

$$\begin{aligned}
 \Delta X &= X_o - X_w \\
 \text{if } |\Delta X| > D_{\max} &\text{ then discard} \\
 \Delta Y &= Y_o - Y_w \\
 \text{if } |\Delta Y| > D_{\max} &\text{ then discard}
 \end{aligned} \tag{6}$$

where  $D_{\max}$  defaults to 10 arcsec (see “-d” in section 2.2.1). SSO/WISE pairs passing this test proceed to the chi-square test:

$$\begin{aligned}
 V_x &= \text{Var}X_o + \text{Var}X_w \\
 V_y &= \text{Var}Y_o + \text{Var}Y_w \\
 V_{xy} &= \text{Var}XY_o + \text{Var}XY_w \\
 \chi^2 &= \frac{V_y \Delta X^2 + V_x \Delta Y^2 - 2V_{xy} \Delta X \Delta Y}{V_x V_y - V_{xy}^2} \\
 \text{if } \chi^2 > \chi_{\max}^2 &\text{ then reject match}
 \end{aligned} \tag{7}$$

where  $\chi_{\max}^2$  is the position-match threshold (see “-c” in section 2.2.1), with a default value of 16. This value for a 2-D chi-square random variable implies that 99.966% of all true matches should be accepted, i.e., 1 out of every 2981 true matches will be sacrificed in the attempt to avoid false matches.

If the match is acceptable, then the following logic is executed, where N indicates the array index of the SSO, K indicates the array index of the WISE source, Ntotal is a counter for the total number of SSOs that are matched, and Ncnfzd is a counter for the number of SSOs that have more than one match.

```

If Nmatch(N) = 0 then                                { first match for this SSO }
  Nmatch(N) = 1
  Ntotal ← Ntotal + 1
  if SSONum(K) = 0 then                               { first match for this WISE source }
    WISEnum(N) = K                                   { store match parameters }
    ChiSq(N) =  $\chi^2$ 
    SSONum(K) = N
  else if ChiSq(SSONum(K)) >  $\chi^2$  then              { new match better than previous }
    WISEnum(N) = K                                   { store match parameters }
    ChiSq(N) =  $\chi^2$ 
    SSONum(K) = N
    WISEnum(SSONum(K)) = 0                           { detach other SSO's connection }
    ChiSq(SSONum(K)) = 9.9d9
  endif
else                                                  { not this SSO's first match }
  Nmatch(N) ← Nmatch(N) + 1
  if (Nmatch(N) .eq. 2) Ncnfzd = Ncnfzd + 1          { count confused SSOs }
  if ChiSq(N) >  $\chi^2$  then                            { new match better than previous }
    if SSONum(K) = 0 then                             { first match for this WISE source }
      WISEnum(N) = K                                   { store match parameters }
      ChiSq(N) =  $\chi^2$ 
      SSONum(K) = N
    else if ChiSq(SSONum(K)) >  $\chi^2$  then            { new match better than previous }
      WISEnum(N) = K                                   { store match parameters }
      ChiSq(N) =  $\chi^2$ 
      SSONum(K) = N
      WISEnum(SSONum(K)) = 0                           { detach other SSO's connection }
      ChiSq(SSONum(K)) = 9.9d9
    endif
  endif
endif

```

This is the same algorithm that was used by 2MASS. In unconfused cases, it provides correct associations. In confused cases, it does not attempt to unravel every possible association. For example, if SSO A has a best-match involving WISE source B, but WISE source B already has a better match to SSO C, no search for a second choice for SSO A is made. Similarly, if SSO A has a better match to WISE source B than SSO C, then SSO C is disconnected from WISE source B and no search for a second choice for it is made. Only best-match associations are kept. The fact that confusion occurred is seen in the Nmatch values. A matched SSO may have  $Nmatch > 1$ , and an unmatched SSO may have  $Nmatch > 0$ .

It can also happen that a WISE source has two acceptable matches to SSOs, loses its best match to another WISE source, and does not fall back to its second choice SSO.

One extra ingredient omitted from the pseudo-code above to reduce the complexity level is as follows: any WISE source with a position uncertainty greater than a threshold (default 5 arcsec) on either axis is tagged as “penalized”, and its position error covariance matrix elements are set to 1% greater than the maximum with zero covariance. As Equation 7 shows, large position uncertainties enlarge the denominator of chi-square, which is appropriate in cases of normal variation in estimation accuracy. When many faint and/or deblended sources are included for the sake of completeness, however, a large fraction of the sources have unacceptably large position uncertainties and dominate the chi-square test unrealistically. The maximum uncertainty limit for penalizing a WISE source can be specified via the “-mu” command-line option. When a matched source has been marked as penalized, its chi-square score is changed to the maximum allowed (“-c” option) plus an integer counter for the number of matches within threshold for the given SSO. For example, with a threshold of 16, if the first match is a penalized source, the match chi-square will be set to 17. If no other matches are found, this match is retained, and the value of 17 in the output file shows that this was a penalized match that was not overridden by an unpenalized source. In most cases, however, several other matches are available (i.e., the match is diagnosed as confused), and one of these usually involves an unpenalized source. This approach allows the fact of confusion to be tracked but prohibits the highly uncertain source from taking over. Testing has shown that this tends to promote matches with photometric information usable in the thermal model (i.e., matches with WISE sources having usable W3 and/or W4 flux estimates).

After all SSOs have been processed for association with WISE sources, another loop over all SSOs is performed to verify that unmatched SSOs had positions inside the real band-frame area, which is generally rotated relative to the U-scan system. The association processing is performed in U-scan because it is the most convenient system for that purpose, but since it is aligned with the local RA and Dec directions, the coarse position test must use a rectangular area whose width is the hypotenuse of the actual array. An unmatched SSO should not be considered “missed” if it fell into a part of the U-scan area not overlapped by the rotated band-frame. So the positions of all unmatched SSOs are transformed to band-frame coordinates (using the PRex subroutine `u2fnd2`), and to be considered truly “missed”, the X coordinate must lie between `ColMin` and `ColMax` (defaults 1 and 1016, resp.; see “-cn” and “-cx” in section 2.2.1), and the Y coordinate must lie between `RowMin` and `RowMax` (defaults 1, and 1016, resp.; see “-rn” and “-rx” in section 2.2.1). Only matched and truly “missed” SSOs are written to the output table file.

Missed SSOs are output with null values for WISE fields and derived parameters, with the exception of the WISE noise values, which are taken as the average over all WISE source noise values, unless there are no WISE sources in the field, in which case these too are null. Matched SSOs have their WISE parameters filled in from the WISE source input, and in addition the derived parameters are computed from the thermal model.

### 3.2.3 Thermal Model Processing

The thermal model is the NEATM model (Harris, 1998) employing the WISE bandpasses in W3 and W4, which allow lookup tables to be computed for flux in watts per square centimeter for an SSO one kilometer in diameter and one AU in topocentric distance as a function of phase angle  $Phase$  and sub-solar temperature  $Tss$ ,  $W3(Phase, Tss)$  and  $W4(Phase, Tss)$ . These lookup tables are read into memory, and another table is computed by dividing each W3 flux by its corresponding W4 flux to obtain a lookup table of flux ratios as a function of phase angle and sub-solar temperature,  $R(Phase, Tss) = W3(Phase, Tss)/W4(Phase, Tss)$ . All values obtained from the tables are interpolated bilinearly in  $Phase$  and  $Tss$ . All uncertainties in the thermal model parameters are propagated from the WISE photometric uncertainties. The thermal model itself is an approximation with its own errors relative to physical reality, but absolute calibration of the thermal model is an intractable problem at this time, and so instead it is taken as a standard against which the WISE data are calibrated, similar to the spirit in which survey photometry is often calibrated against the survey's own internal standard. The WISE photometric errors are assumed to be small enough to use the usual truncated Taylor-series approach to propagating uncertainties (e.g., as in Equation 11 below). The derived parameters are the following.

<i>Alb</i> :	Bond Albedo
<i>SigAlb</i> :	One-sigma uncertainty in Bond albedo
<i>Diam</i> :	Diameter (km)
<i>SigDiam</i> :	One-sigma uncertainty in diameter (km)
<i>Beam</i> :	Beaming parameter
<i>SigBeam</i> :	One-sigma uncertainty in beaming parameter
<i>Tss</i> :	Sub-solar temperature (k)
<i>SigTss</i> :	One-sigma uncertainty in sub-solar temperature (k)

#### 3.2.3.1 No WISE Association

If the SSO has no WISE association, then the derived parameters are set to null, as are the WISE photometric parameters  $Wmag$  and  $SigWmag$ .

#### 3.2.3.2 No W3 or W4 Flux

If the SSO has a WISE association, but the WISE source has no usable flux in W3 or W4, then the derived parameters are set to null, but the WISE photometric parameters  $Wmag$  and  $SigWmag$  are passed to the output table file.

### 3.2.3.3 W3 or W4 Flux But Not Both

If the SSO has a WISE association whose photometric information includes a usable W3 flux or a usable W4 flux but not both, then the following iterative algorithm is employed to estimate the derived parameters. The emissivity  $eps$  is set to 0.9, and the WISE flux and uncertainty,  $Wflux_n$  and  $sigWflux_n$ , in watts per square centimeter are computed for the available band by applying the appropriate photometric zero point to  $Wmag(n)$  and  $sigWmag(n)$ , where  $n = 3$  or  $4$  for W3 or W4, respectively.

If the perihelion distance is less than 1.3 AU, the SSO is considered an NEA (Near-Earth Asteroid), otherwise it is considered an MBA (Main-Belt Asteroid). The beaming parameter is estimated as follows.

$$\begin{aligned} Beam &= 0.910 + 0.013 \times Phase && (NEA) \\ SigBeam &= 0.017 + 0.004 \times Phase && (NEA) \\ Beam &= 0.781 && (MBA) \\ SigBeam &= 0.150 && (MBA) \end{aligned} \tag{8}$$

The following parameters are initialized, where  $q$  is the factor that converts geometric albedo  $pv$  to Bond albedo  $Alb$ , and  $N_{iter}$  is the iteration number.

$$\begin{aligned} N_{iter} &= 0 \\ q &= 0.290 + 0.684G \\ pv_0 &= 2 \\ pv &= 0.1 \end{aligned} \tag{9}$$

The iteration is performed as follows.

$$\begin{aligned}
N_{iter} &\Leftarrow N_{iter} + 1 \\
Alb &= q \times pv \\
T_{ss} &= 394.48 \sqrt{\frac{\sqrt{\frac{|1 - Alb|}{Beam \times eps}}}{Rhelio}} \\
F &= Interp(W_n(Phase, T_{ss})), \quad n = 3 \text{ or } 4 \text{ for } W3 \text{ or } W4, \text{ resp.} \\
Diam &= R_{topo} \sqrt{\frac{Wflux_n}{F}} \\
SigDiam &= \left| \frac{\partial Diam}{\partial Wflux_n} \right| sigWflux_n = \frac{R_{topo} \times sigWflux_n}{2 \sqrt{F \times Wflux_n}} \\
pv &= \frac{(1329 \times 10^{-H/5})^2}{Diam^2} \\
SigAlb &= \frac{|q \times (1329 \times 10^{-H/5})^2|}{Diam^3} SigDiam \\
&\text{if } |pv - pv_0| < 0.0005 \text{ then iteration converged} \\
&\text{else } pv_0 = pv, \text{ iterate again}
\end{aligned} \tag{10}$$

If the iteration is not converged after  $N_{iter} = 1000$ , a warning message is written, and iteration ceases with the current values passed forward. The only parameter not computed inside the iteration loop is the uncertainty in the sub-solar temperature, which is computed after the iteration ceases.

$$\begin{aligned}
\frac{\partial T_{ss}}{\partial Alb} &= \frac{394.48}{\sqrt{Rhelio} \sqrt{eps}} \frac{1}{4 Beam \left( \frac{|1 - Alb|}{Beam} \right)^{3/4}} \equiv K \\
\frac{\partial T_{ss}}{\partial Beam} &= \frac{K (1 - Alb)}{Beam} \\
SigT_{ss} &= \sqrt{\left( \frac{\partial T_{ss}}{\partial Alb} SigAlb \right)^2 + \left( \frac{\partial T_{ss}}{\partial Beam} SigBeam \right)^2}
\end{aligned} \tag{11}$$

### 3.2.3.3 W3 and W4 Flux

If the SSO has a WISE association whose photometric information includes both a usable W3 flux and a usable W4 flux, then the following algorithm is employed to estimate the derived parameters. The emissivity  $eps$  is set to 0.9, and the WISE fluxes and uncertainties,  $Wflux_n$  and  $sigWflux_n$ , in watts per square centimeter are computed for both bands by applying the

photometric zero points to  $Wmag(n)$  and  $sigWmag(n)$ , where  $n = 3$  and  $4$  for W3 and W4, respectively.

The *Phase* value is known, so the *Tss* value can be found by computing the ratio of the observed fluxes,  $R_{34} = Wflux_3/Wflux_4$  and locating this in the  $R(Phase, Tss)$  table by reverse interpolation in *Tss*. Denoting the phase-interpolated *R* value at *Tss* row *K* as  $R_K$ , the interpolation fraction in *Tss* is  $f_T = (R_{34} - R_K)/(R_{K+1} - R_K)$ , and  $Tss = TssBase - 1 + K + f_T$  (since the spacing in the *Tss* rows is one degree; *TssBase* is the base temperature of the flux tables, defaults to 120, and may be specified on the command line via the “- $\tau 0$ ” flag). The error in this value depends on the error in the WISE fluxes. The uncertainty  $sigTss$  is therefore given by:

$$\begin{aligned} \frac{\partial Tss}{\partial Wflux_3} &= \frac{\partial}{\partial Wflux_3} \left( \frac{\frac{Wflux_3}{Wflux_4} - R_K}{R_{K+1} - R_K} \right) = \frac{1}{R_{K+1} - R_K} \\ \frac{\partial Tss}{\partial Wflux_4} &= \frac{\partial}{\partial Wflux_4} \left( \frac{\frac{Wflux_3}{Wflux_4} - R_K}{R_{K+1} - R_K} \right) = \frac{-\frac{Wflux_3}{(Wflux_4)^2}}{R_{K+1} - R_K} \\ sigTss &= \sqrt{\left( \frac{\partial Tss}{\partial Wflux_3} sigWflux_3 \right)^2 + \left( \frac{\partial Tss}{\partial Wflux_4} sigWflux_4 \right)^2} \end{aligned} \quad (12)$$

With both *Phase* and *Tss* known, the values in the W3 and W4 flux tables can be interpolated to obtain  $F_3$  and  $F_4$ , respectively. From these, two estimates of the diameter are possible,  $D_3$  and  $D_4$ , respectively, with uncertainty variances computed from the products of  $sigWflux_n$  and the derivatives of the dependence on  $Wflux_n$ .

$$\begin{aligned} D_3 &= Rtopo \sqrt{\frac{Wflux_3}{F_3}} \\ var(D_3) &= \frac{(Rtopo \times sigWflux_3)^2}{4 \times F_3 \times Wflux_3} \\ D_4 &= Rtopo \sqrt{\frac{Wflux_4}{F_4}} \\ var(D_4) &= \frac{(Rtopo \times sigWflux_4)^2}{4 \times F_4 \times Wflux_4} \end{aligned} \quad (13)$$

These two diameter estimates are averaged with inverse-variance weights, and a chi-square is computed to check whether they were compatible.

$$\begin{aligned}
\text{var}(Diam) &= \frac{\text{var}(D_3) \times \text{var}(D_4)}{\text{var}(D_3) + \text{var}(D_4)} \\
Diam &= \text{var}(Diam) \left( \frac{D_3}{\text{var}(D_3)} + \frac{D_4}{\text{var}(D_4)} \right) \\
\chi^2 &= \frac{(D_3 - D_4)^2}{\text{var}(D_3) + \text{var}(D_4)} \\
\text{if } \chi^2 > ChSqRS \text{ then } \text{var}(Diam) &\Leftarrow \chi^2 \times \text{var}(Diam) \\
sigDiam &= \sqrt{\text{var}(Diam)}
\end{aligned} \tag{14}$$

The uncertainty is inflated if  $\chi^2$  is greater than the threshold  $ChSqRS$ , which may be specified on the command line and defaults to 9. Since  $\chi^2$  here has two degrees of freedom, the threshold of 9 implies that one out of every 90 cases will have uncertainty inflation without outliers playing any role; if the two diameter estimates are discordant, the uncertainty will almost certainly be enlarged to express that fact.

With  $Diam$  evaluated the Bond albedo  $Alb$  and its uncertainty can be computed as follows.

$$\begin{aligned}
pv &= \left( \frac{1329 \times 10^{-H/5}}{Diam} \right)^2 \\
q &= 0.290 + 0.684 G \\
Alb &= q \times pv \\
sigAlb &= \frac{q \times 1329 \times 10^{-H/5} \times sigDiam}{Diam^3}
\end{aligned} \tag{15}$$

The beaming parameter and its uncertainty are computed as follows.

$$\begin{aligned}
C &\equiv \frac{(394.48)^4}{eps \times Rhelio^2} \\
Beam &= \frac{C \times (1 - Alb)}{Tss^4} \\
sigBeam &= \sqrt{\left( \frac{4C \times (1 - Alb) \times sigTss}{Tss^5} \right)^2 + \left( \frac{C \times sigAlb}{Tss^4} \right)^2}
\end{aligned} \tag{16}$$

### **3.3 SSOTHMODO Processing**

The ssothmod module reads the SSO/WISE Associations File (SIS sso04, WSDC D-I148) *or* a WMOPS source source file (SIS TBD). The thermal model is then evaluated for each input SSO for which WISE photometry in W3 and/or W4 is usable.

When processing a file generated by ssoid, ssothmod simply repeats the same thermal-model computations described in section 3.2.3.

Processing of WMOPS files is not yet implemented. Some design issues regarding how to handle a possible lack of absolute visual magnitudes need to be worked out, and the file format has not been finalized. The ssothmod output file is expected to be the same format for either input.

## **4 Output**

### **4.1 SSOINIT Output**

#### 4.1.1 Orbital Element Subset Output

The orbital-element information for SSOs inside the scan swath is written to the file whose name is specified on the command line via the “-o” option (SIS sso02: SSO ssoinit/ssoid Intermediate File, WSDC D-I146).

#### 4.1.2 Three-Epoch Ephemerides File

If three-epoch ephemerides were requested (command-line options “-i2” and “-o2”), then these are written to the specified output file (SIS sso03: SSO Three-Epoch Ephemerides File, WSDC D-I147).

### **4.2 SSOID Output**

#### 4.2.1 SSO/WISE Associations Output File

The ssoid module always generates a file for SSO/WISE associations, even if there are no SSOs and/or WISE sources in the array area (SIS sso04: SSO/WISE Associations File, WSDC D-I148).

#### 4.2.2 SSO Meta-Data Output File

If a meta-data file was requested via the command-line option “-om”, then the ssoid module always generates one, even if there are no SSOs and/or WISE sources in the array area (SIS sso05: SSO Meta-Data File, WSDC D-I149).

### **4.3 SSOTHMOD Output**

#### 4.3.1 SSO/WISE Associations Output File

The ssothmod module generates the same output file as ssoid (section 4.2.1, SIS sso04, WSDC D-I148) except for the addition of a single header line identifying the ssothmod version and run date/time.

## 5 Testing

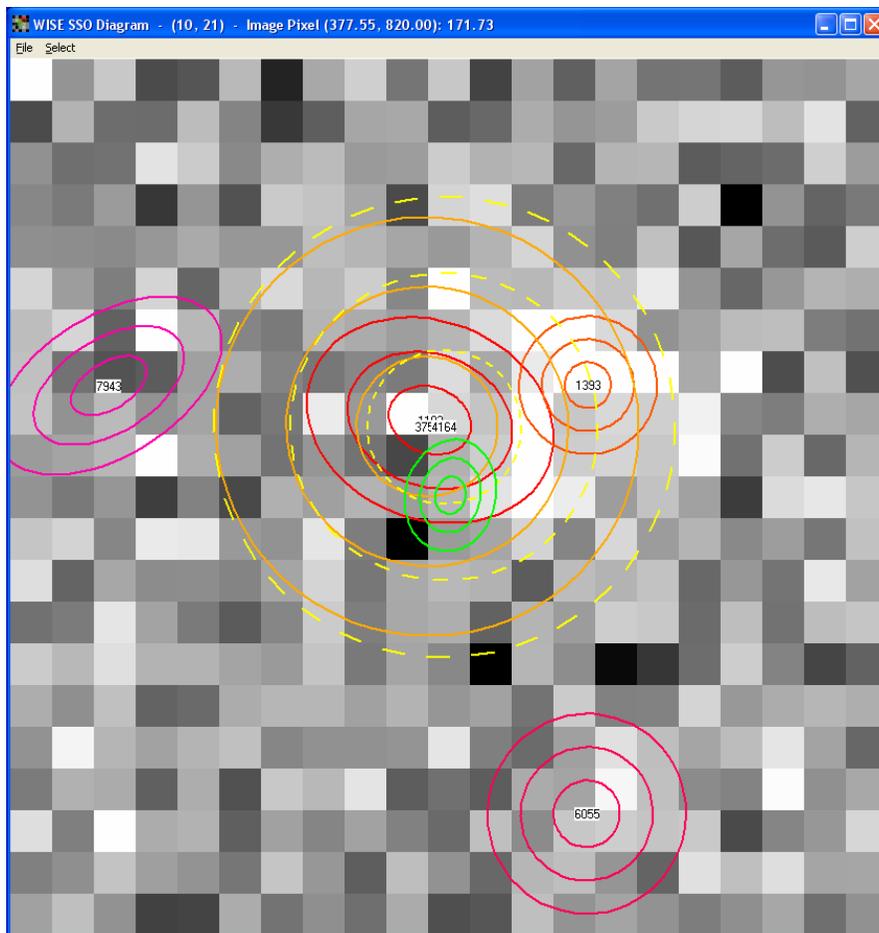
SSOID testing has been performed with the simulation data provided by N. Wright and by comparison of results using D. Tholen's orbital data to D. Tholen's own results. The latter testing employed the ssoinit Three-Epoch Ephemerides File to obtain positions for comparison to independent calculations by D. Tholen. The former testing required the use of N. Wright's orbital elements and the use of the "-xL" option in ssoid; this produced a fairly high match rate between predicted SSOs and WISE sources detected and extracted via standard WSDC processing of simulated image files. Detailed examination of processing results for each SSO in the standard test frameset was performed with the SSODD offline analysis program using "-o2" output from ssoid. The standard test frameset is characterized by the following FITS header lines (from the W3 image):

```
RAWFILE = '/wise/data/sky_sim/mar25_09/09328075108-3.fits'  
L0FILE  = './5a/99105a/fr/001/99105a001-w3-int-0.fits'  
BAND    = 3  
DATE_OBS= '2009-11-24T07:51:12'  
SCAN    = '99105a '  
SCANGRP = '5a      '  
FRNUM   = 1  
DATIME  = '091124T075112Z'  
EPHEMT  = 312364272  
UTC     = '2009-328T07:51:12.000'  
SUN2SCX = 0.462968884770082  
SUN2SCY = 0.7999991208995145  
SUN2SCZ = 0.346831529608879  
SCVELX  = -0.0150917238950426  
SCVELY  = 0.00659417246213268  
SCVELZ  = 0.00745522741259859
```

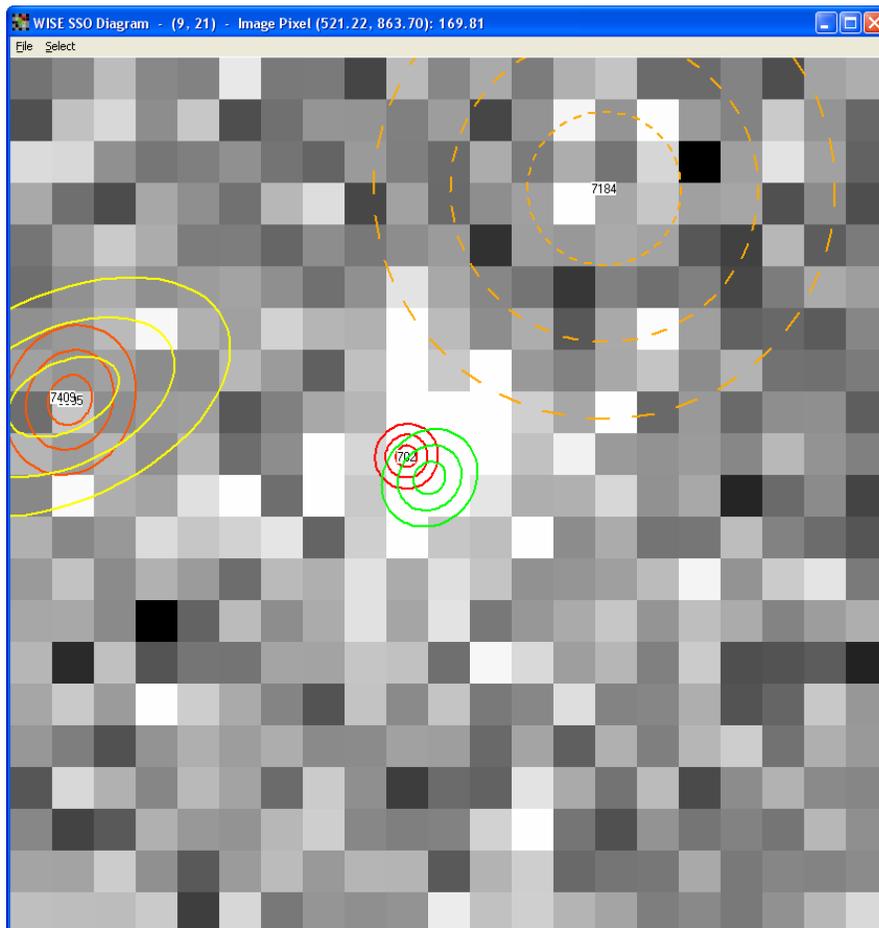
Examples of clean unconfused matches were examined, as were cases of considerable confusion. Alignments of WISE position uncertainty ellipses with image features were examined in each wavelength (which requires four runs of ssoid, which uses only one FITS file per execution).

Two examples are given below using W3, one a very confused match that was recovered by using the "penalizing" feature for sources with inordinately large position uncertainties, and one a clean match. Penalized sources are shown with their error ellipses drawn with dashed lines. The postage stamp is centered on the SSO of interest, which is drawn in green; all other colors indicate WISE sources except for blue, which is used only for neighboring SSOs, if any.

```
WISE SSO Diagram Information
DD file: W:\SSOID\test\ddtest2
Image File: 99105a001-w3-int-lb.fits
Image l% - 99% range: 167.06 to 177.21
-----
Main SSO: (8035) 1992 TB
X, Y (pix): 378.53 809.53
SigX, SigY (pix): 0.45 0.36; Angle (deg): 77.4
Image postage-stamp min and max: -2.34 and 180.29
Linear stretch from 167.06 to 177.21
WISE source no. 1192; Usage code 1
X, Y (pix): 378.04 811.33
SigX, SigY (pix): 1.01 0.79; Angle (deg): 158.7
WISE source no. 1393; Usage code 1
X, Y (pix): 381.82 812.17
SigX, SigY (pix): 0.55 0.55; Angle (deg): 158.7
WISE source no. 3758; Usage code 1
X, Y (pix): 377.97 811.18
SigX, SigY (pix): 1.69 1.67; Angle (deg): 158.7
WISE source no. 4164; Usage code 0
X, Y (pix): 378.38 811.17
SigX, SigY (pix): 1.84 1.84; Angle (deg): 68.7
WISE source no. 6055; Usage code 1
X, Y (pix): 381.79 801.90
SigX, SigY (pix): 0.80 0.79; Angle (deg): 68.7
WISE source no. 7943; Usage code 1
X, Y (pix): 370.35 812.16
SigX, SigY (pix): 1.00 0.56; Angle (deg): 31.8
```



```
WISE SSO Diagram Information
Main SSO: (53928)                2000 GT39
X, Y (pix):          523.03   853.91
SigX, SigY (pix):   0.41  0.36; Angle (deg): 49.3
Image postage-stamp min and max: -1.78 and 183.85
Linear stretch from 167.06 to 177.21
WISE source no. 702; Usage code 1
X, Y (pix):          522.48   854.43
SigX, SigY (pix):   0.26  0.25; Angle (deg): 68.7
WISE source no. 6805; Usage code 1
X, Y (pix):          514.42   855.78
SigX, SigY (pix):   0.61  0.51; Angle (deg): 68.7
WISE source no. 7184; Usage code 0
X, Y (pix):          527.21   860.85
SigX, SigY (pix):   1.84  1.84; Angle (deg): 68.7
WISE source no. 7409; Usage code 1
X, Y (pix):          514.26   855.85
SigX, SigY (pix):   1.42  0.82; Angle (deg): 25.3
```



## Appendix A SSO Orbit and Apparent Position Computation

Except for planetary satellites, the orbit computation is performed by the same algorithm as the 2MASS DTPGM program. For a given SSO the following computations are performed.

The input orbital information consists of:

<i>NAME:</i>	35-character ID of object (standard name, number, and designation, as available)
<i>Q:</i>	Perihelion distance (AU)
<i>ECC:</i>	Eccentricity
<i>PV:</i>	Unit vector from the Sun toward perihelion in J2000
<i>QV:</i>	Unit vector orthogonal to the P vector in the orbital plane, right-handed about the orbital angular momentum vector
<i>TPERI:</i>	Time of perihelion passage (Julian Date)
<i>EPOCH:</i>	Epoch of osculation (Julian Date)
<i>H:</i>	Absolute magnitude
<i>G:</i>	Slope parameter used in computing the phase-dependent term in the visual magnitude using the asteroid phase function developed by Bowell, Harris, and Lumme
<i>ERR:</i>	Ephemeris uncertainty code

All numeric parameters are double-precision reals except for *G*, which is single precision. All computations are done in double precision.

The mean motion *DTN* is computed from

$$DTN = G_k \sqrt{\left(\frac{|1 - ECC|}{Q}\right)^3} \quad (A.1)$$

where  $G_k$  is the Gauss constant, 0.01720209895. The mean anomaly *MA* is computed from

$$MA = DTN \times (ET - TPERI - DELTA / c) \quad (A.2)$$

where *ET* is the ephemeris time of the observation (the observation epoch in UT corrected to ephemeris time by application of standard corrections), *DELTA* is the topocentric distance, initially set to zero because it is not yet known, and *c* is the speed of light in AU/day. This last term is the correction for one-way light time, which will be made by recomputing the mean anomaly on a second pass after the topocentric distance has been computed without this correction. In principle, this should be iterated more than twice, but in practice the second-pass solution is accurate at the milli-arcsec level, since distant objects (with large one-way light times) have low angular motions, while nearby objects with high angular motion have small one-way light times.

Given the mean anomaly  $MA$  and the eccentricity  $ECC$ , Kepler's equation can be solved iteratively for the eccentric anomaly,  $EA$ . The subroutine that solves Kepler's equation can handle any positive eccentricities, but only eccentricities between 0 and 1 are encountered in SSOID, so only that solution will be discussed. A Newton-Raphson iteration is employed to solve the equation

$$\begin{aligned} f(EA) &= EA - ECC \times \sin(EA) - MA \\ \frac{dF(EA)}{dEA} &= 1 - ECC \times \cos(EA) \end{aligned} \quad (\text{A.3})$$

The second line above is the derivative, which is used in the denominator of the Newton-Raphson iteration. The initial guess for  $EA$  is just the mean anomaly,  $MA$ . A subscript  $n$  will denote the iteration number, with the initial guess having  $n = 0$ .

$$\begin{aligned} EA_0 &= MA \\ EA_{n+1} &= EA_n - \frac{EA_n - ECC \times \sin(EA_n) - MA}{1 - ECC \times \cos(EA_n)} \end{aligned} \quad (\text{A.4})$$

When either  $f(EA_n)$  or  $f(EA_n)/EA_n$  drops below  $10^{-12}$  in absolute value, the iteration is terminated, and the value of  $EA$  is taken to be  $EA_{n+1}$  on the last iteration. Then the true anomaly,  $TA$ , is computed, which allows the heliocentric distance  $R$  to be computed.

$$\begin{aligned} TA &= 2 \tan^{-1} \left( \sqrt{\frac{1 + ECC}{1 - ECC}} \tan \left( \frac{EA}{2} \right) \right) \\ R &= \frac{Q(1 + ECC)}{1 + ECC \times \cos(TA)} \end{aligned} \quad (\text{A.5})$$

This completes the solution for the SSO coordinates in the orbital plane, but that plane is generally rotated with respect to J2000 celestial coordinates, so a transformation into the latter system is required and accomplished as follows to obtain the position vector  $POS$  and velocity vector  $VEL$ .

$$\begin{aligned}
X &= R \cos(TA) \\
Y &= R \sin(TA) \\
POS(1) &= PV(1) \times X + QV(1) \times Y \\
POS(2) &= PV(2) \times X + QV(2) \times Y \\
POS(3) &= PV(3) \times X + QV(3) \times Y \\
C &\equiv G_k \sqrt{\frac{1}{Q(1 + ECC)}} \\
\dot{X} &= -C \sin(TA) \\
\dot{Y} &= C(\cos(TA) + ECC) \\
VEL(1) &= PV(1) \times \dot{X} + QV(1) \times \dot{Y} \\
VEL(2) &= PV(2) \times \dot{X} + QV(2) \times \dot{Y} \\
VEL(3) &= PV(3) \times \dot{X} + QV(3) \times \dot{Y}
\end{aligned} \tag{A.6}$$

The heliocentric distance,  $Hdist$ , is the magnitude of the  $POS$  vector. The spacecraft vectors provided on the command line are then subtracted from these heliocentric SSO vectors to obtain the topocentric vectors. Once the topocentric distance  $DELTA$  is known, the one-way light time can be computed and subtracted off of the epoch as shown in Equation A.2; then the computation is repeated. The apparent  $RA$  and  $Dec$  of the SSO are obtained straightforwardly from the topocentric vector, and a finer test can be performed to determine whether the SSO is actually in the field of interest defined by the FITS file. In the *ssoid* module, such SSOs need proper motion and uncertainty parameters, which are computed as follows, where  $VEL2$  is the SSO's topocentric velocity vector.

$$\begin{aligned}
RUNOFF &= \frac{ERR \times Hdist}{DELTA} + 1 \\
\begin{pmatrix} RATES(1) \\ RATES(2) \\ RATES(3) \end{pmatrix} &= \begin{pmatrix} -\sin(Dec) \cos(RA) & -\sin(Dec) \sin(RA) & \cos(Dec) \\ -\sin(RA) & \cos(RA) & 0 \\ -\cos(Dec) \cos(RA) & -\cos(Dec) \sin(RA) & -\sin(Dec) \end{pmatrix} \begin{pmatrix} VEL2(1) \\ VEL2(2) \\ VEL2(3) \end{pmatrix} \\
MOTION &= \frac{2.387324146 \sqrt{RATES(1)^2 + RATES(2)^2}}{DELTA} \\
ANGLE &= \tan^{-1} \left( \frac{RATES(2)}{RATES(1)} \right)
\end{aligned} \tag{A.7}$$

Here  $RUNOFF$  is the 1-sigma semimajor axis of the uncertainty ellipse in arcseconds; it is clipped if necessary at 99.9. The semiminor axis of uncertainty is always taken to be 1 arcsecond.  $MOTION$  is the proper motion in arcseconds/second, and  $ANGLE$  is the direction of

proper motion measured east of north. The same computation using VEL instead of VEL2 yields the orientation of the “line of variation”, which is the angle of the major uncertainty axis.

The phase angle, *Phase*, is the angle between the heliocentric and topocentric vectors. This, together with the topocentric distance, permits the visual magnitude to be computed from the Bowell, Harris, and Lumme model,  $B(G, Phase)$ .

$$Vmag = H + 5 \log_{10}(Hdist \times DELTA) + B(G, Phase) \quad (A.8)$$

Processing of planetary satellites involves a completely different solution for the apparent position, proper motion, and line of variation. Each object’s orbital position about its parent planet must be integrated from a known orbital state at a chronologically close epoch. The method used is implemented in Fortran code supplied by D. Tholen based on a 15<sup>th</sup>-order Gauss-Radau integrator as discussed by E. Everhart (An efficient integrator that uses Gauss-Radau spacings, Dynamics of Comets: Their Origin and Evolution, Proceedings of IAU Colloq. 83).

## Appendix B Mapping Position Error Covariance Matrices

Section 3.2.2 above describes the position matching for SSOs and WISE sources. A chi-square test is used, and the position error covariance matrices are needed for each pairwise SSO/WISE test. Neither the SSO nor the WISE input files explicitly contain the elements of a position error covariance matrix; the latter must be computed from parameters that do exist in the input files. Positions themselves are in J2000 RA and Dec; these are converted to U-scan coordinates in a straightforward manner (U-scan coordinates are Cartesian in arcsecond units, right-handed with the Y axis pointing north and the origin at the center of the frame described by the input FITS file, which may be for any WISE band).

After converting to U-scan coordinates, the position information is as follows. For WISE sources, the positions and uncertainties are:

$X_w, Y_w$ : U-scan coordinates (arcsec)  
 $SigX_w, SigY_w, SigXY_w$ : 1-sigma position uncertainties (arcsec)

where  $SigXY_w$  is the “co-sigma” parameter that carries the covariance information using the same units as the other two uncertainties. Positive values of  $SigXY_w$  indicate positive correlation in the errors in RA and Dec; because of the left-handedness of RA/Dec coordinates, however (when viewed from inside the celestial sphere), such positive correlation implies negative correlation in U-scan X and Y coordinates. The elements of the WISE error covariance matrix are therefore:

$$\begin{aligned} VarX_w &= SigX_w^2 \\ VarY_w &= SigY_w^2 \\ VarXY_w &= -SigXY_w | SigXY_w | \end{aligned} \tag{B.1}$$

where the minus sign on the third line reverses the covariance so that positive error correlation in RA and Dec becomes negative correlation in U-scan.

For SSOs, the position parameter and uncertainties are:

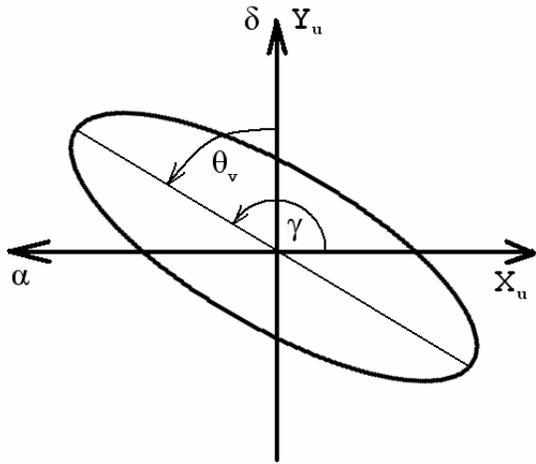
$X_o, Y_o$ : U-scan coordinates (arcsec)  
 $ErrMaj_o, ErrMin_o$ : Semimajor and semiminor axes of uncertainty ellipse (arcsec, 1-sigma)  
 $ErrAng_o$ : Direction of ErrMaj axis (deg E of N)

The angle  $ErrAng_o$  is the amount of rotation from the north direction to the “line of variation”, which is the line on the sky along which the SSO moves in its orbit as seen from the observatory as though the observatory were stationary in heliocentric coordinates; this is generally not the same as the direction of proper motion, which is typically strongly influenced by the motion of the observatory. The largest component of SSO position uncertainty is along the line of variation, since the orientation of the plane of the orbit is typically much better determined than the angular location of the SSO within that plane. The major axis of the uncertainty ellipse is assigned the

label  $X$  arbitrarily; this means that we need the angle from the U-scan  $X$  axis to the line of variation, which we will label  $\gamma$ , and  $ErrAng_o$  will be labeled  $\theta_v$ . The conversion from standard deviation (sigma) to variance is the usual:

$$\begin{aligned} VarX_{co} &= ErrMaj_o^2 \\ VarY_{co} &= ErrMin_o^2 \end{aligned} \quad (B.2)$$

where the  $c$  subscripts indicate “celestial”. These variances then need to be rotated through the angle  $\gamma$  by a similarity transformation. The RA/Dec and U-scan coordinate systems are diagrammed in the figure below, where positive correlation in RA and Dec is used for illustration. It is clear that  $\gamma = \theta_v + 90^\circ$  and that this produces the negative correlation in U-scan.



The similarity transformation provides the error covariance matrix elements needed in U-scan; this simple 2-dimensional transformation amounts to:

$$\begin{aligned} VarX_o &= VarX_{co} \cos^2 \gamma + VarY_{co} \sin^2 \gamma \\ VarY_o &= VarX_{co} \sin^2 \gamma + VarY_{co} \cos^2 \gamma \\ VarXY_o &= (VarX_{co} - VarY_{co}) \sin \gamma \cos \gamma \end{aligned} \quad (B.3)$$