

**W. W. Hansen Experimental Physics Laboratory  
STANFORD UNIVERSITY  
STANFORD, CA 94305 – 4085**

**Gravity Probe-B Relativity Mission**

# Precise Orbit Determination From GPS XYZ Position Output

**MOC P0664 Rev D  
03/15/2004**

## Approvals

	Signature	Date
Updated by: Tim Walsh		
Ken Galal OD Manager		
Kelly Burlingham SU, Software Quality Effectiveness		
Marcie Smith SU, MOC Manager		

## Revision Record

REV	DATE	Summary Description
-	03/07/00	Original
A	10/31/01	Major revision
B	4/01/02	Minor updates, including new checklist and delivery form ECO#: 1361
C	04/30/2003	Major revision – replaced much of the procedure w/ new GUI code for od_widget. Added many screen snaps.
D	03/15/2004	Minor updates (new QA form, OD cleanup tool use, current directory locations, OD DB archive naming convention etc.) ; added new section to cover the use of the ACCEL card generator to model thrusting during orbit propagations;

## Acronyms and Abbreviations

Acronym / Abbreviation	Meaning
ATCS	Attitude and Trajectory Control Subsystem
FDF	Flight Dynamics Facility
FTP	File Transfer Protocol
GN	Ground Network
GPS	Global Positioning System
GSFC	Goddard Space Flight Center
IDD	Interface Definition Document
ILRS	International Laser Ranging Service
MOC	Mission Operations Center
OD	Orbit Determination
ODBIOU	Orbit Database Input/Output Utility
ODF	Orbit Determination Function
PCE	MicroCosm orbit element observation data
QA	Quality Assurance
SLR	Satellite Laser Ranging
SN	Space Network
STK	Satellite Tool Kit
TDP	Telemetry Data Processing
UTC	Universal Time Coordinated

ITAR Assessment Performed \_\_\_\_\_  
Tom Langenstein

ITAR Control Req'd? ☐ Yes ☐ No

## 1.0 SCOPE

This procedure	<input type="checkbox"/> Does	<input checked="" type="checkbox"/> Does NOT	provide formal verification of GP-B Requirements.
This procedure	<input type="checkbox"/> Does	<input checked="" type="checkbox"/> Does NOT	include constraints and restrictions for the Payload.

### 1.1 SUMMARY OF CONTENTS

• Section 2	PRE-REQUISITES .....	3
• Section 3	MOC HARDWARE REQUIRED .....	3
• Section 4	MOC SOFTWARE REQUIRED .....	4
• Section 5	PROCEDURES REQUIRED .....	4
• Section 6	REFERENCES, APPLICABLE DOCUMENTS AND NOTES .....	4
• Section 7	OPERATIONS .....	4
• Section 8	PROCEDURE COMPLETION .....	8

## 2.0 PRE-REQUISITES

The operations in Section 7 assume that the user has transferred the GPS data to the MicroCosm home directory.

## 3.0 MOC HARDWARE REQUIRED

### 3.1 WORKSTATION(S) REQUIRED

Type	Label
UNIX on Science Network	Any

### 3.2 OTHER HARDWARE REQUIRED

Type	Quantity
None	

## 4.0 MOC SOFTWARE REQUIRED

### 4.1 CSTOL Scripts

CSTOL Script Name	Version No.
None	

### 4.2 UNIX Scripts/Executables

UNIX Script/Executable Name	Version No.
MicroCosm	v. 2001
ODBIOU	v. 1.2
ORBITget.pl	v. 1.0
od_widget.ptk	v. 3.0
	v. 1.0

### 4.3 Matlab Scripts/m-files

Matlab m-file Name	Version No.
filterC1.m	0
report_mean	0
report_inst	0

## 5.0 PROCEDURES REQUIRED

Procedure Name	Procedure No. & Rev.

## 6.0 REFERENCES, APPLICABLE DOCUMENTS, and NOTES

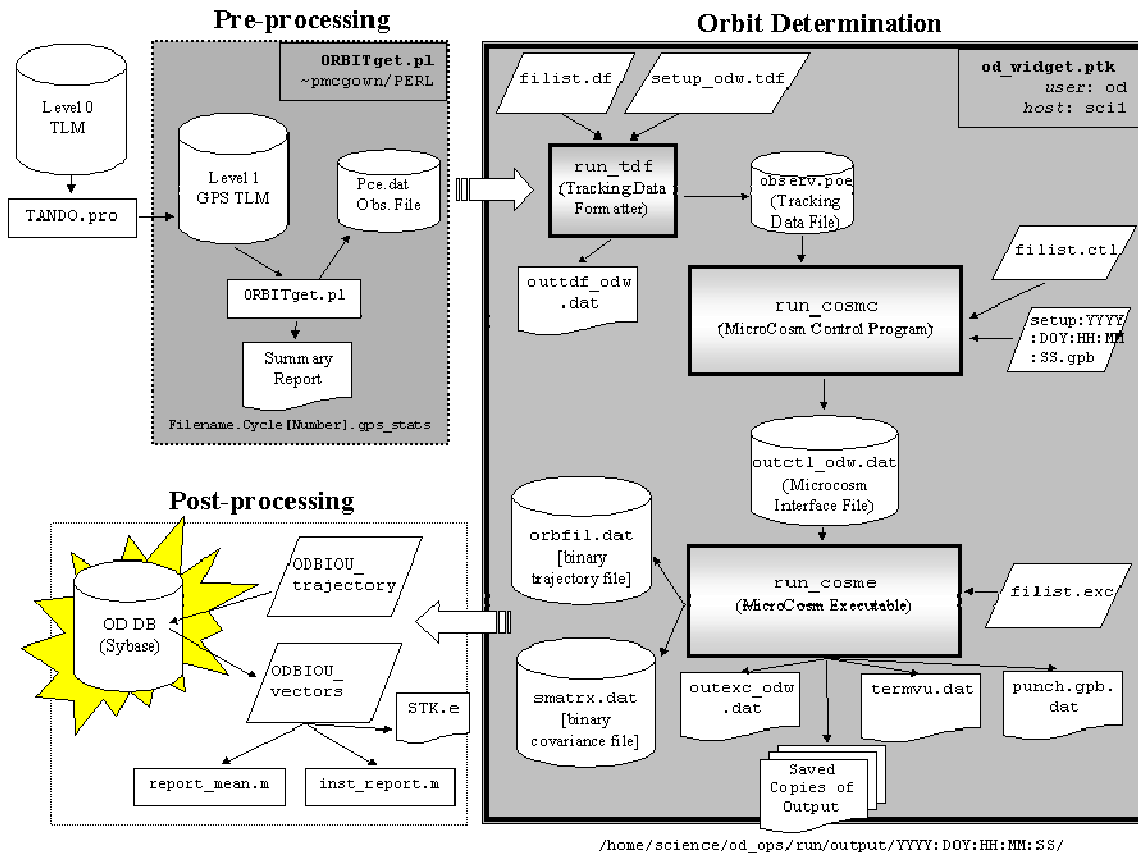
Document	Document No.
MicroCosm Version 2001	Manuals 1-4
Interface Definition Document for GP-B OD	S0505

NOTES:

## 7.0 OPERATIONS

The following operations outline the GPS Orbit Determination function at GP-B, beginning with GPS data collection, through orbit determination, and finally population of the OD database with trajectory data. These three main phases of operations are grouped as Pre-processing, Orbit Determination, and Post-processing, as shown below in Fig. 1. The smaller labelled shapes in Fig. 1 represent the individual ASCII and binary input/output files that are used throughout the orbit determination process, if the complete process were done manually.

To facilitate a streamlined procedure, two PERL scripts have been written. The first, `ORBITget.pl`, performs most of the functions contained within the Pre-processing grey box. Assuming Level0 GPS data has been stored in the telemetry database (`TANDO.pro`), `ORBITget.pl` will extract Level1 GPS data and create a PCE file suitable for use in MicroCosm. The procedure for creating MicroCosm-compliant PCE data files using `ORBITget.pl` is given in the next section.



**Figure 1: GP-B Orbit Determination Process Using GPS Telemetry**

The second PERL script, `od_widget.ptk`, is an interactive GUI utility that streamlines the creation and management of input/output files required for MicroCosm, based upon user input. `od_widget.ptk` performs all functions necessary to run MicroCosm and compute all required orbit determination data products.

The final stage of operations, Post-processing, involves using several programs to load (and recall) data into the OD database. Additionally, MATLAB scripts are used to generate necessary orbit determination products for Mission Planning and the external elements. A working knowledge of MicroCosm and MATLAB is assumed in these operations (In the following procedure, `>` is used to indicate the UNIX prompt and `>>` is used to indicate the MATLAB prompt). Consult MicroCosm manuals 1-4 for information on how to use MicroCosm.

## 7.1 Pre-Processing (`ORBITget.pl`)

- 7.1.1 The `ORBITget.pl` utility assumes that the Level0 database has been populated. To run the utility, you will need to know the following: (1) the desired time span of GPS data in the `YYYY:DOY:HH:MM:SS` format and (2) the cycle number of the data set. While logged into a `scil` UNIX terminal as the user “`od`”, type:

```
> ~pmcgown/PERL/ORBITget.pl [location TBD]
```

- 7.1.2 Enter the desired cycle number, or hit `<RETURN>` to use the default cycle number shown for data retrieval:

```
INTERACTIVE: VTCW Cycle verification ?
Default      [ -4078 ]
? Cycle ?
```

- 7.1.3 Enter the start and stop times for data retrieval. To use the default times (default start - `2000:001:12:00:00.0`, default stop - current time) for the given cycle number, hit `<RETURN>`:

```
INTERACTIVE: MIN TIME [ 2000:001:12:00:00.0 ]
? TIME ?
```

```
INTERACTIVE: MAX TIME [ 2003:101:21:38:38 ]
? TIME ?
```

- 7.1.4 Enter the output filename prefix for the PCE file. In the example below, “`pce-test`” is used:

```
INTERACTIVE: Output filename ?
to be deposited in /projects/od/OFILE/GPS/
? pce-test
GENERATING: /projects/od/OFILE/GPS/pce-test.Cycle-4078.orbit
```

- 7.1.5 Note, the complete PCE file name will be “[prefix from above].Cycle[data cycle number].orbit.pce”. Change to the following directory to view the output files:

```
> cd /projects/od/OFILE/GPS
> ls
```

```
ORBIT_CACA          pce-test.Cycle-4078.orbit
ORBITcln            pce-test.Cycle-4078.orbit.exp
SSS_ORBIT.sql        pce-test.Cycle-4078.orbit.pce
```

The other files in this directory are temporary files used within ORBITget.pl.

#### 7.1.6 Copy the PCE file to the OD “run” directory:

```
> cp pce-test.Cycle-4078.orbit.pce  
/home/science/od_ops/run/pcefiles
```

## 7.2 Orbit Determination

This section details how MicroCosm is used to perform GP-B orbit determination using GPS position data as an input. In previous versions of this document, it was necessary for the user to manually create and edit all required input files for MicroCosm orbit determination. However, most of the manual work required for orbit determination has now been replaced by a GUI PERL script, *od\_widget.ptk*, shown in Fig. 1 as the large, gray process box surrounding the three MicroCosm executables (*run\_tdf*, *run\_cosmc*, *run\_cosme*) and all input/output files.

Once the user has completed GPS data preprocessing outlined in section 7.1, and created a pce-file in the */home/science/od\_ops/run/pcefiles/* directory, orbit determination can be accomplished by executing the PERL script:

```
> od_widget.ptk
```

This script will guide the user through the four main steps required for orbit determination: (1) selecting a PCE file for use as input (2) setting control parameters (drag coefficient, geopotential model size, planetary modeling, etc.) (3) inputting the epoch state vector (4) running the MicroCosm executables (*run\_tdf*, *run\_cosmc*, *run\_cosme*).

Due to the needs of both the Science Team and the Operations Team, it is now necessary to run MicroCosm **twice** when creating standard daily OD products. The first run should create output data in the J2000 reference frame with binary output trajectory every 2 seconds. The second run, using identical input vectors and parameters, should create output data in the TOD reference frame with binary output trajectory every 3600 seconds. The *od\_widget* program has been modified such that the Input/Output window of step (2), allows for easy selection of either Science Team (J2000) output products, or Operations Team (TOD) output products.

### 7.2.1 Getting Ready to Run *od\_widget.ptk*

To perform the orbit determination function via the *od\_widget.ptk* script, you must have the following:

1. Valid GPS data formatted as a PCE file, as outlined in section 7.1 of this document. (Note, MicroCosm can be run with no GPS observations, in which case the trajectory will be simply propagated from the epoch state vector).
2. Be logged into the UNIX workstation, *sci1.private.net*, which is the Sun Ultra 10 located in the Orbit Determination office. Furthermore, you must be logged in as the “od” user, and be the only person currently running MicroCosm.
3. A valid cartesian or Keplerian state vector to use as epoch input.

## 7.2.2 Performing Orbit Determination / Running `od_widget.ptk`

### 7.2.2.1 Execute `od_widget.ptk` from any UNIX directory:

```
> od_widget
```

After a few moments, the main control screen should appear:



Figure 2: `od_widget.ptk` Main Window

Note that the main screen outlines 4 major steps to running MicroCosm. However, steps (2)-(4) are disabled until the user has completed step (1). Once step (1) has been completed, step (2) will become unlocked, and so forth. In this manner, the script guides the user through the four logical steps of running MicroCosm.

To select a PCE file for use as input, or to specify that no GPS data be used for propagation, click on the first button. The PCE file selection window should appear after a few moments:

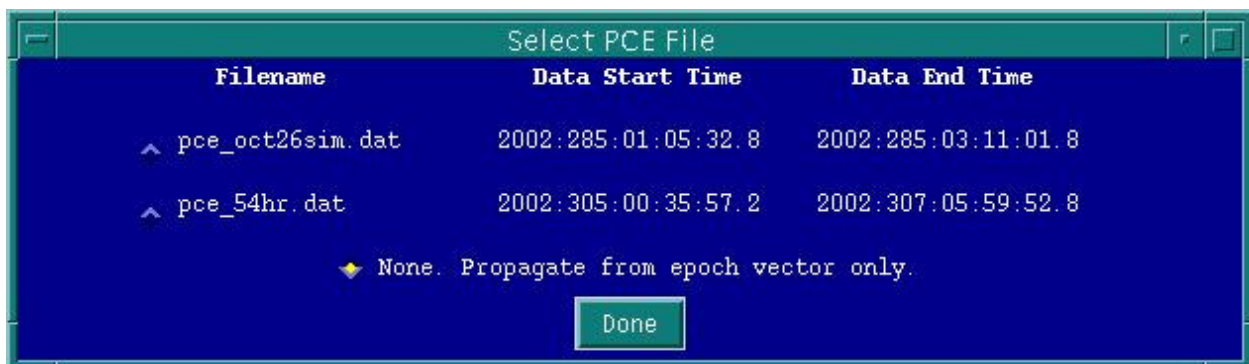


Figure 3: PCE File Selection Window



The PCE file selection window shows all available GPS files that can be used, including each data file's start and end times. The default selection is to use no GPS data, which will setup MicroCosm to propagate a trajectory from the epoch state vector alone.

Once the PCE file (or *none*) has been selected the user will be returned to the main window, as shown below. Click the second button (now unlocked) to enter the MicroCosm parameter editing window.



Figure4: Main Window, step (2) enabled

#### 7.2.2.2 Review default MicroCosm OD parameters from the following submenu:

The user may change any of the default parameters by simply changing the numerical value in the text entry box, or clicking on the appropriate checkbox. There are a few important things to remember while changing the OD parameters:

To compute a drag-free orbit, click the associated checkbox. This effectively makes the satellite a point mass by forcing the cross-sectional area of the spacecraft to zero. This in turn, disables atmospheric drag and solar radiation pressure effects.

The widget has been changed in its last release, such that every single run will always output binaries files (`ORBFIL`) in the **TOD AND J2000** formats. There is no need to specify one or the other. However, the **ASCII** output file (`ORBTUV`) can be selected to the TOD or the J2000 format. Simply enter the desired output step size in the appropriate boxes for each. Also note, when manually changing the integration

step size, be sure to keep the integration step size *less than or equal to* the output step sizes of the ORBFIL and ORBTUV files.

A screen snapshot of the OD parameter entry window is shown below.

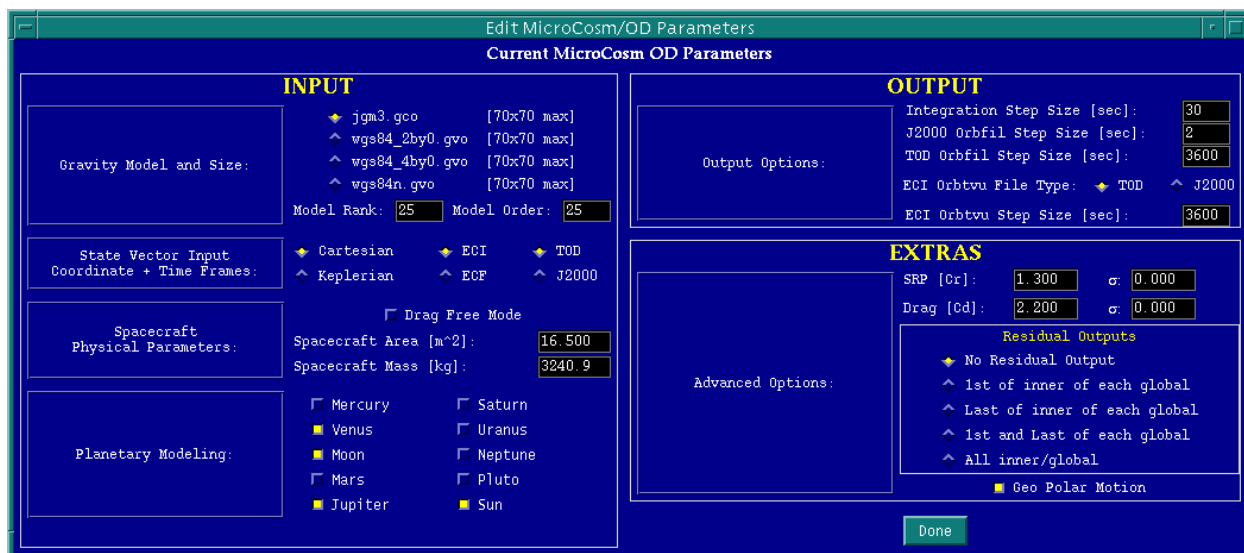


Figure 5: MicroCosm OD Parameter Entry Window

When all OD parameters are satisfactory, click [Done] to go back to the main menu, as shown below. Click the third button (now unlocked) to enter the epoch state vector, as shown below.

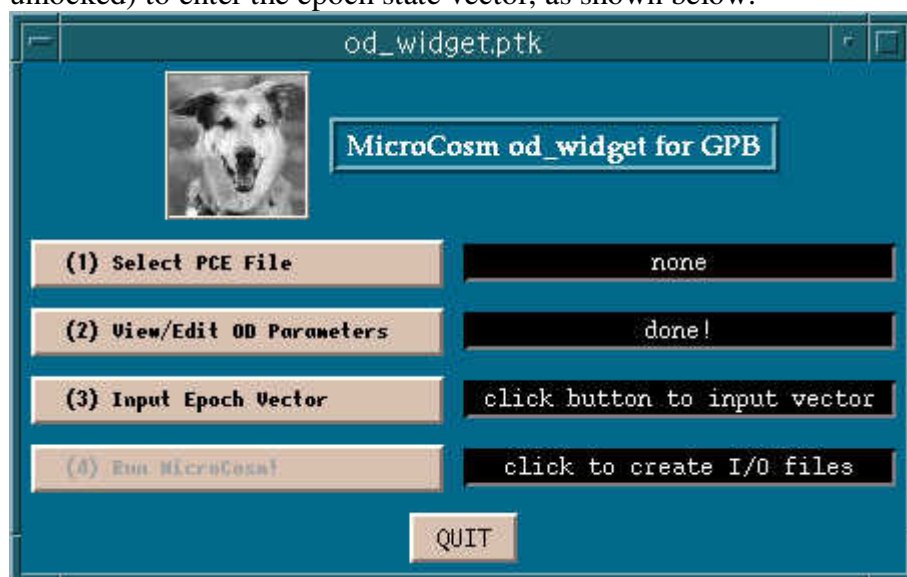


Figure 6: Main Window, step (3) enabled

### 7.2.2.3 Input the epoch state vector by clicking (3) from the main menu:

**Input Epoch State Vector**

**State Vector Input**

Element Type: cartesian

Coord. Sys. Reference: ECI

Time Reference: TOD

Get Epoch from Input File:

Epoch Time: 2003 : 001 : 00 : 00 : 00.0

Run Start Time: 2003 : 001 : 00 : 00 : 00.0

Run End Time: 2003 : 001 : 00 : 00 : 00.0

X [m]: 0.0000000

Y [m]: 0.0000000

Z [m]: 0.0000000

Xdot [m/s]: 0.0000000

Ydot [m/s]: 0.0000000

Zdot [m/s]: 0.0000000

Mean Radius [km]: 0

Mean Velocity [km/s]: 0

Figure 7: State Vector Input Window, Prior to Input

Note, you are about to enter the epoch state vector in the coordinate / reference / time frames set from the default parameters menu in step (2). In the above example, the state vector should be entered in cartesian / ECI / TOD frames.

Once you have entered the epoch / run start / run end times in the YYYY:DOY:HH:MM:SS format, and the 6 components of the cartesian orbit vector, the mean orbit radius and velocity will be computed at the bottom of the input window as a sanity check:

State Vector Input		
Element Type:	cartesian	
Coord. Sys. Reference:	ECI	
Time Reference:	TOD	
Get Epoch from Input File:	Select File	
Epoch Time:	2002 : 335 :	13 : 15 : 00.0
Run Start Time:	2002 : 327 :	01 : 15 : 00.0
Run End Time:	2002 : 348 :	01 : 15 : 00.0
X [m]:	6690663.4600000	
Y [m]:	-2032196.7800000	
Z [m]:	-678090.7200000	
Xdot [m/s]:	-686.9096200	
Ydot [m/s]:	207.7208600	
Zdot [m/s]:	-7499.1982400	
Mean Radius [km]:	7025.2835	
Mean Velocity [km/s]:	7.5335	
Done		

Figure 8: State Vector Input Window, Post Input w/ Mean Radius and Velocity Check

If you have chosen in step (2) to input a Keplerian state vector, a similar window will be shown, but without the sanity check of mean orbit radius and velocity:

State Vector Input		
Element Type:	keplerian	
Coord. Sys. Reference:	ECI	
Time Reference:	TOD	
Get Epoch from Input File: <span>Select File</span>		
Epoch Time:	2003 : 032 :	02 : 00 : 00.0
Run Start Time:	2003 : 032 :	00 : 00 : 00.0
Run End Time:	2003 : 032 :	04 : 00 : 00.0
a [m]:	7790979.5620000	
e:	0.001240775	
i [deg]:	51.997779422	
raan [deg]:	359.757831831	
argp [deg]:	14.115342092	
ma [deg]:	5.144522059	
<span>Done</span>		

Figure 9: State Vector Input Window, Keplerian Vector

od\_widget also includes a feature to import an input state vector from a valid MicroCosm ASCII trajectory file (an Orbtvu file). To select an ASCII input file to use for an epoch state vector, click the “Select File” button. Note you must still select an ASCII file which is of the same coordinate system, reference system, and time system as chosen in step (2). The file selection dialog window is shown below.

Note that the file selection dialog window initially starts in the `/home/science/od_ops/run/output` directory, and only shows filenames that end in the `*.txt` extension.

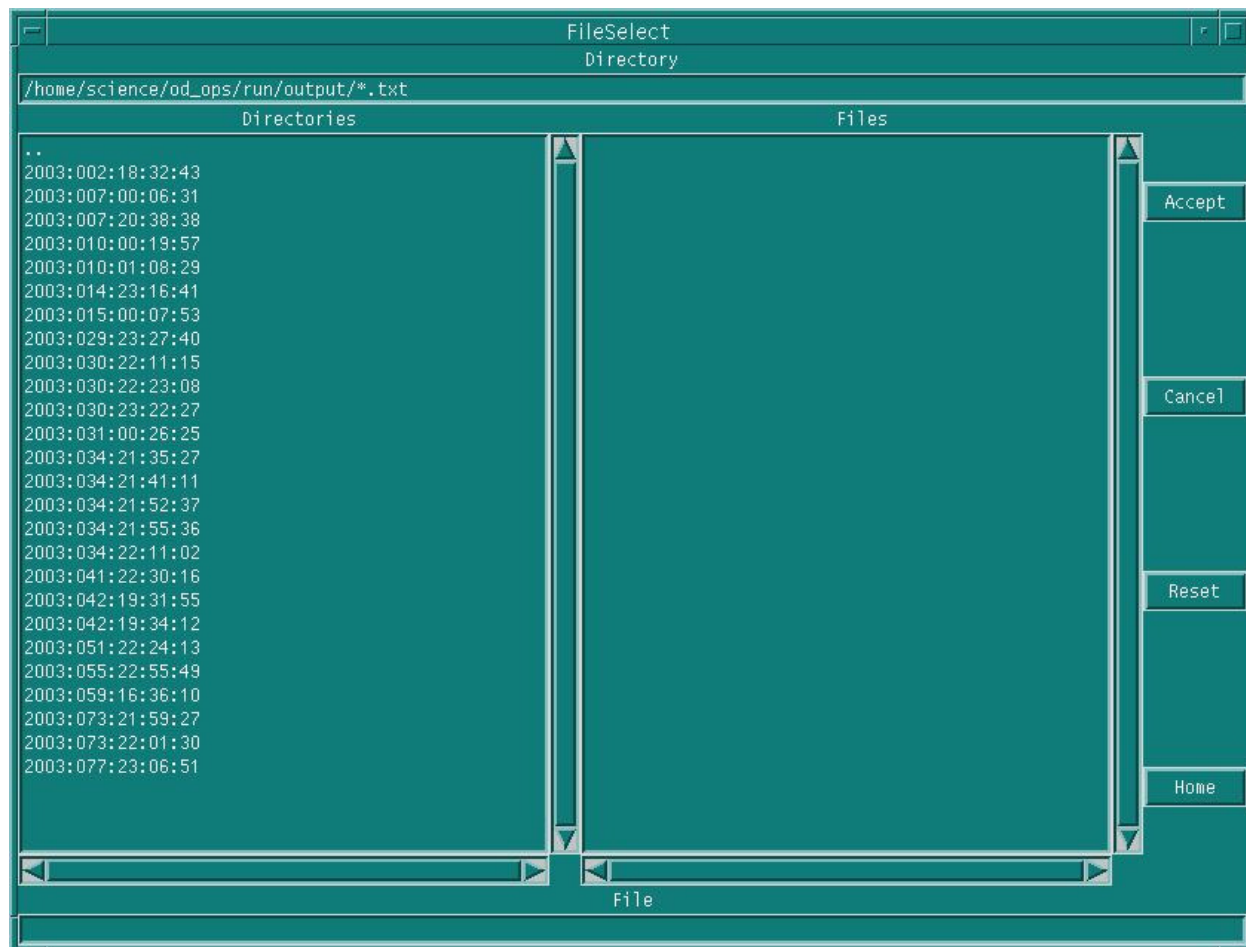


Figure 10: State Vector Input File Select Window

Selecting any directory will take you to a sub-directory, showing you all the text files contained within. Again, you *must* select a file that has the same reference and coordinate frame chosen earlier in step (2) of the `od_widget`. In this case, we had selected a cartesian, TOD, ECI input vector, so we must select the file “`car_tod.txt`” shown below.

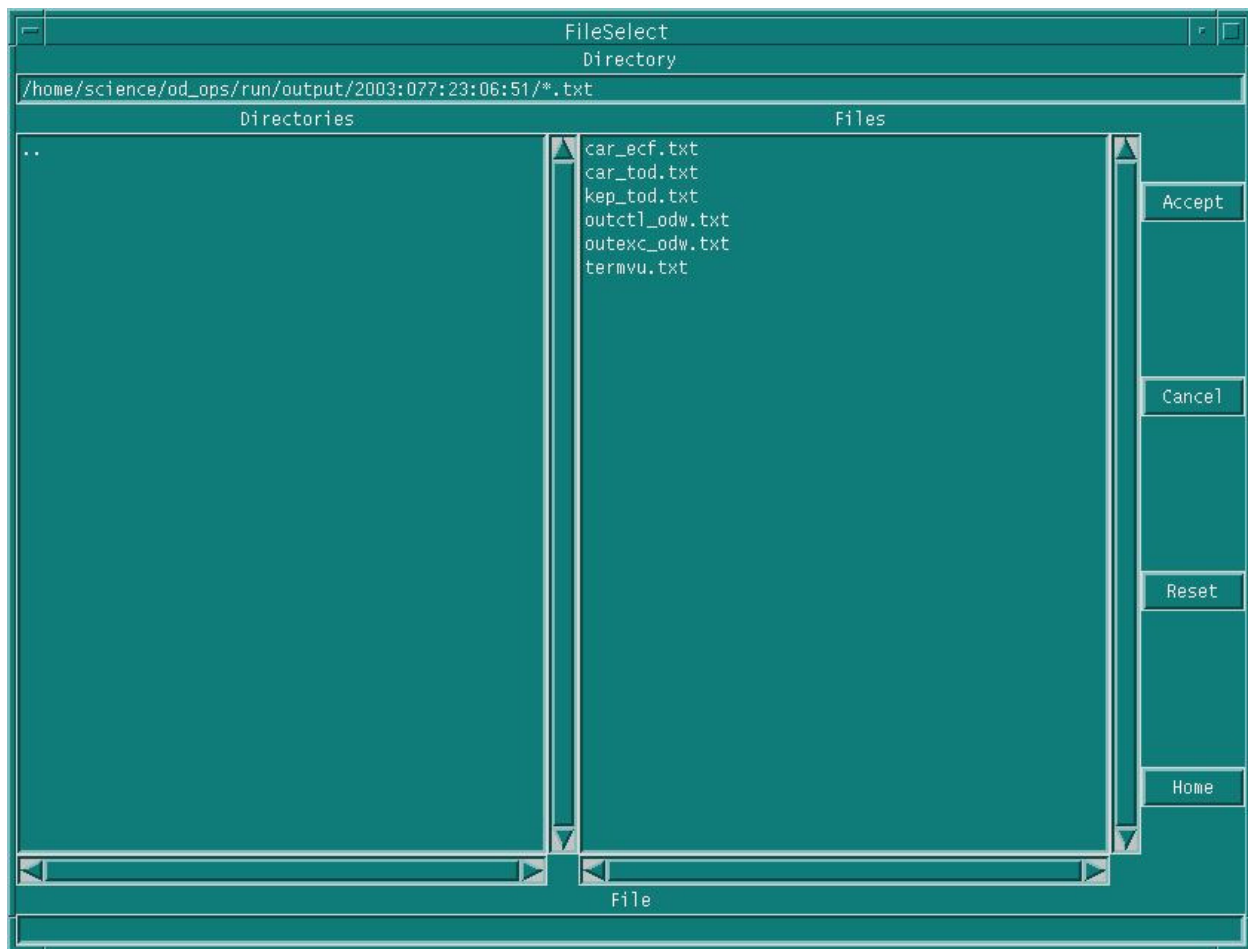


Figure 11: State Vector Input File Select Window

Once you have chosen the input file for use, click “Accept”. If the file chosen is compatible with the input parameters of step (2), you will be prompted to select an individual state vector from all the ones contained within that file. The window that guides you through this process is shown below:



External State Vector

Select Time for External State Vector

Element Type: cartesian

Coord. Sys. Reference: ECI

Time Reference: TOD

File Start Time: 2002:327:01:15:00.000

Desired Epoch Time: 2002:335:13:15:00.000

Epoch Time:

File End Time: 2002:348:01:15:00.000

Groovy

Figure 12: State Vector Selection Window

This selection window will show you the time of the first data point contained within this file, and the last data point contained within this file. You can select any data point between these two times by dragging the slider-bar to the left or to the right. Click “Groovy” when you have chosen the time of the state vector to use as input.

Note that in practice, for daily OD, a 30 hour GPS telemetry span centered at noon of a given processing day will be selected for processing. By convention, a vector at 00:00:00 UTC will be chosen (corresponding to the same date as that of the noon-centered span) as the epoch for the run.

Once you are satisfied with the input state vector, click “Done” to return to the main window again, with the final button now unlocked:





Figure 13: Main Window, step (4) enabled

- 7.2.2.4 Run MicroCosm executables by pressing (4). The script will execute the three MicroCosm executables (run\_tdf, run\_cosmc, run\_cosme) in sequential order. Another subwindow guides you through this process, as shown below:



Figure 14: MicroCosm Runtime Window, step 1

Note that if “No GPS” data was selected, there is no need to run TDF (telemetry data formatter), and you are automatically advanced to the second step of this window, as shown here:



Figure 15: MicroCosm Runtime Window, step 1,  
No GPS data

If you are processing GPS data, clicking on (1) will start TDF. You can monitor the progress of TDF by looking at the UNIX window that started the `od_widget.ptk` script. Normal execution will show information about the version of TDF being run, and nothing else. Below is shown nominal output from TDF displayed in the UNIX shell window:

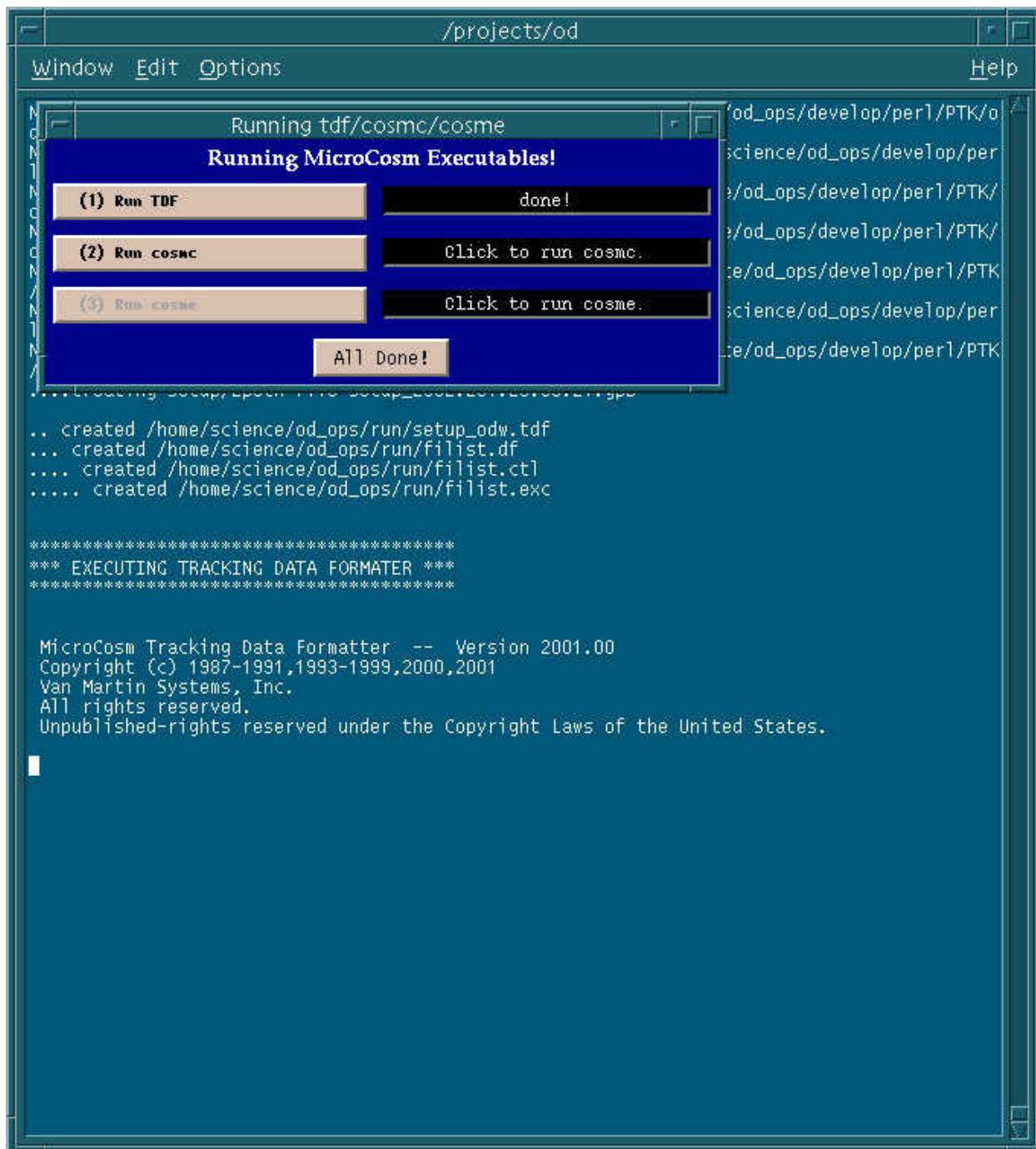


Figure 16: *od\_widget.ptk* results shown in UNIX window for TDF

Once TDF has completed, step (2) will be unlocked (if you have selected “No GPS” data, you will start here). Click the second button to start the MicroCosm control program. Again, you can monitor the progress of `run_cosmc` by examining the UNIX shell window, as shown below. A successful completion will show MicroCosm version information and runtime CPU usage:

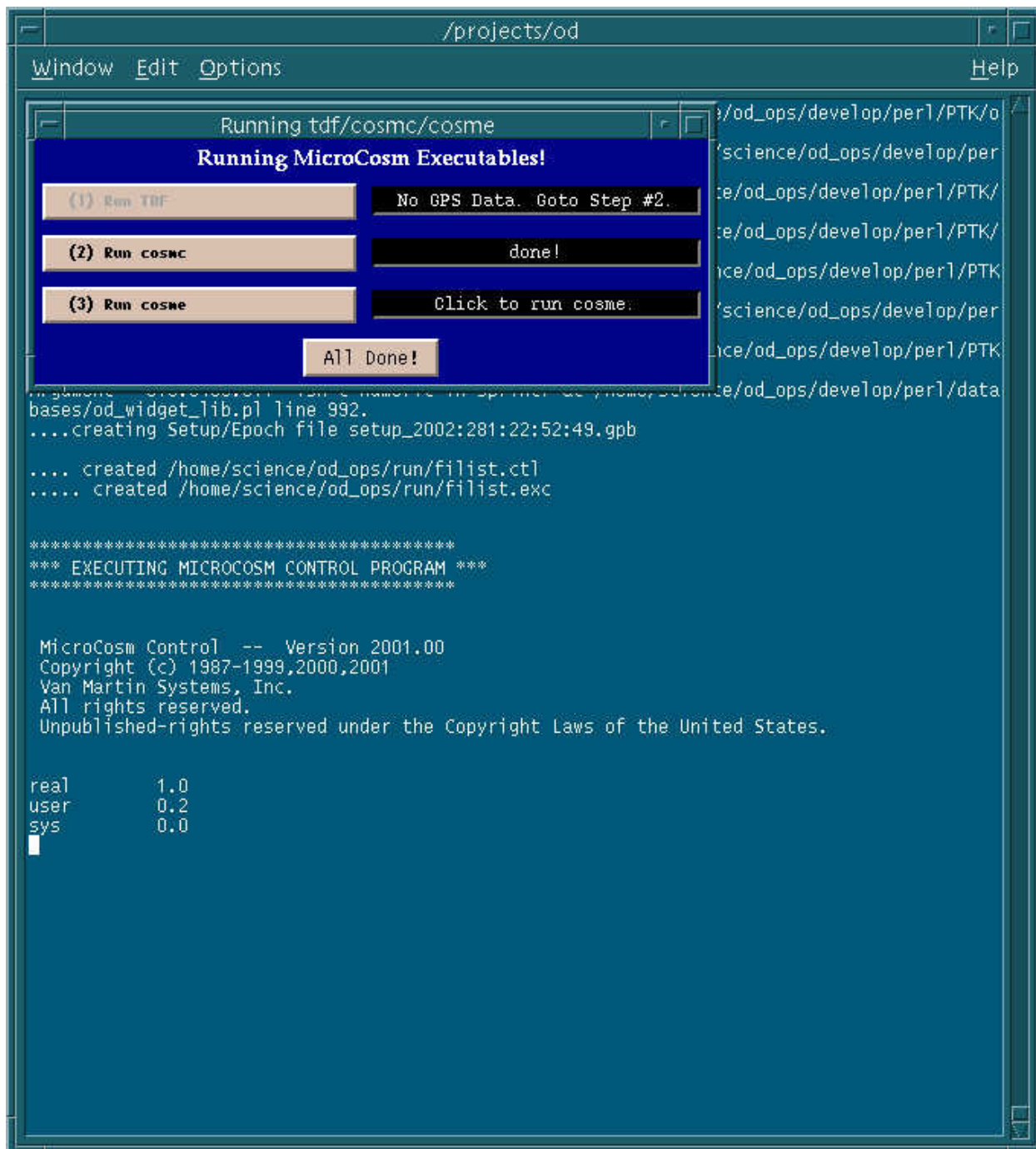


Figure 17: *od\_widget.ptk* results shown in UNIX window for *cosmc*

Once *cosmc* has completed, step (3) will be unlocked. Click the third button to start the MicroCosm executable. Again, you can monitor the progress of *run\_cosme* by examining the UNIX shell window, as shown below. A successful completion will show MicroCosm version information and runtime CPU usage:

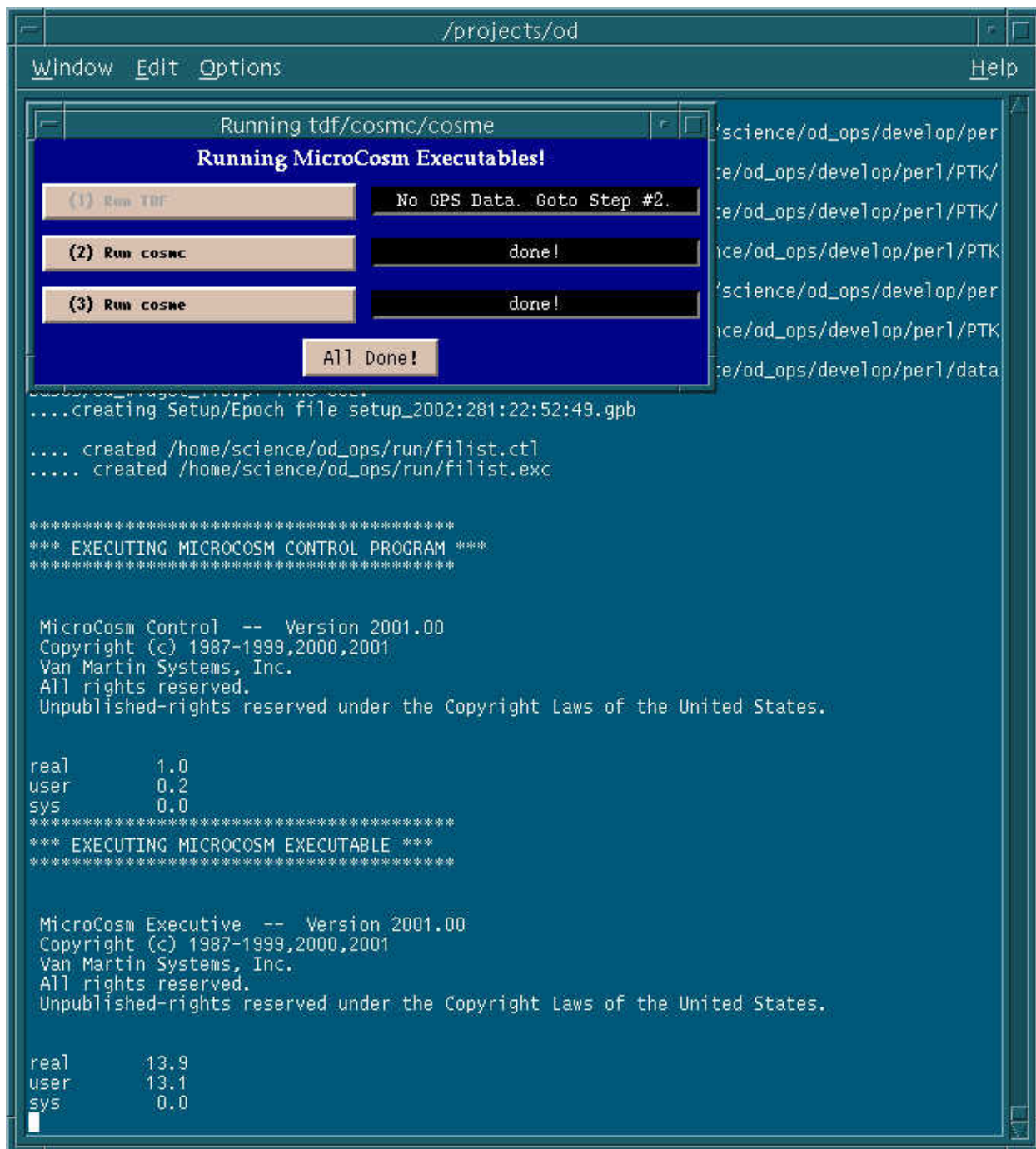


Figure 18: *od\_widget.ptk* results shown in UNIX window for *cosme*

Once *cosme* has successfully completed, click the “All Done!” button to return to the main menu. Clicking this button will also save off all input and output files of MicroCosm to a new directory at:

`/home/science/od_ops/run/output/YYYY:DOY:HH:MM:SS`

where `YYYY:DOY:HH:MM:SS` is the system time at which the MicroCosm run was initiated.



Once back at the main window, click on the “Quit” button to end the `od_widget` script. Otherwise, if changes to the input/output parameters or state vector are required, the OD processing can be repeated by starting over with step (1) or step (2) of the GUI.

#### 7.2.2.5 Review Results of MicroCosm run by changing to the output directory of that run and performing a directory listing:

```
> cd /home/science/od_ops/run/output/YYYY:DOY:HH:MM:SS
> ls -l
```

Individual output files can be examined by performing a UNIX “more” on a filename. A sample directory listing is shown below:

```
sci1:/home/science/od_ops/run/output/2003:077:23:06:51
{od@sci1:31} pwd
/home/science/od_ops/run/output/2003:077:23:06:51
{od@sci1:32} ls -l
total 794
-rw-rw-r-- 1 od od 688 Mar 18 23:31 termvu.txt
-rw-rw-r-- 1 od od 1176 Mar 18 23:31 smatrix_odw.dat
-rw-rw-r-- 1 od od 1391 Mar 18 23:31 setup_2003:077:23:06:51.gpb
-rw-rw-r-- 1 od od 17736 Mar 18 23:31 outexc_odw.txt
-rw-rw-r-- 1 od od 27612 Mar 18 23:31 outctl_odw.txt
-rw-rw-r-- 1 od od 99072 Mar 18 23:31 orbfil_odw.dat
-rw-rw-r-- 1 od od 74647 Mar 18 23:31 kep_tod.txt
-rw-rw-r-- 1 od od 80 Mar 18 23:31 filist.exc
-rw-rw-r-- 1 od od 121 Mar 18 23:31 filist.ct1
-rw-rw-r-- 1 od od 81191 Mar 18 23:31 car_tod.txt
-rw-rw-r-- 1 od od 81213 Mar 18 23:31 car_ecf.txt
{od@sci1:33} 
```

Figure 19: Sample output directory as seen in UNIX shell window

### 7.2.3 Modelling Thrust During Orbit Determination

The MicroCosm software allows for the modeling of thrust during orbit computations. This is particularly challenging to implement because every time the direction of thrust is changed, a new line or “card” must be entered in the master setup file. For Gravity Probe B, the direction of thrust normal to the orbit plane will be switched every half-orbit, normally occurring at the poles.

Since the GP-B orbit period is only approximately 97 minutes, this requires that new cards be added to the setup file every 48.5 minutes. For a run spanning 30 hours, this would require the operator to add 37 lines to the setup file!

To streamline this process, a simple PERL tool, `ACCEL_card_generator.ptk`, has been developed to compute the necessary `ACCEL` cards for a MicroCosm run. User-intervention, however, is required to cut and paste these lines into the master setup file prior to running `cosmc` or `cosme`. This can be best accomplished by running the `od_widget.ptk` tool, and manually intervening

before letting the tool run `cosmc` or `cosme`, described in section 7.2.2.4 above. The process described below is actually a 2-part procedure. First, the user must create a “dummy” ephemeris with NO thrusting over the desired timespan. The result of the dummy ephemeris is an ASCII keplerian output file, from which argument of latitude and north/south pole crossings are computed. The `ACCEL_card_generator.pl` tool reads this file and with GUI inputs from the user, generates the ACCEL cards to be inserted into the subsequent, “with-thrust”, run of MicroCsm.

7.2.3.1 For the desired time span to model thrust, perform an OD run as outlined above in section 7.2.2. Pay special attention to Figure 5 in section 7.2.2.2. Since the purpose of this run is *ONLY* to make an ASCII output file, the step size of the binary output files (`ORBFIL`) can be made rather larger, perhaps 3600 seconds or higher. The ASCII ECI output file (`ORBTUV`), however, **must have a small step size**. It is found that a 60 second step size works well. At the completion of the run, be sure to note which output directory is created to hold the resultant `kep_tod.txt` file.

7.2.3.2 Run the ACCEL card generating utility:

```
> ~twalsh/PERL/ACCEL_CARD/ACCEL_CARD_generator.ptk
```

7.2.3.3 Using the GUI interface shown in Figure 20, select which orbital events (ASC/DSC, N.Pole/S.Pole) shall be reported to the screen. Note that only information regarding north + south pole crossings will be used for generating the ACCEL cards.

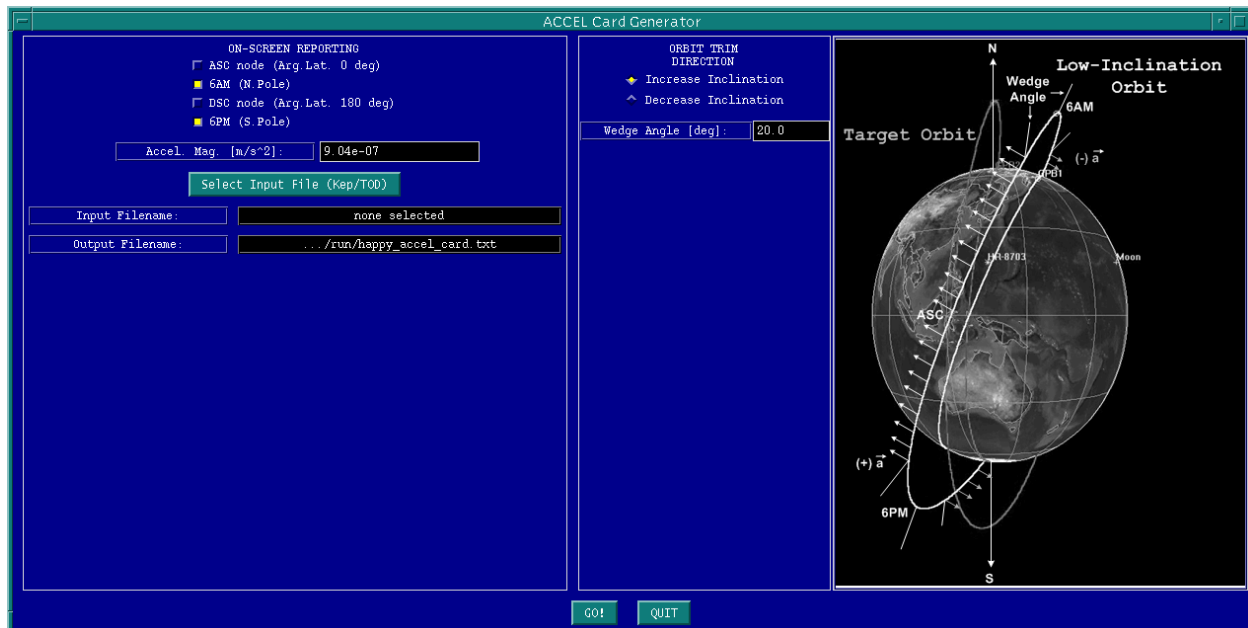


Figure 20: `ACCEL_CARD_generator.ptk` main window

7.2.3.4 Input the thrust/acceleration magnitude in  $\text{m/s}^2$ . It is not possible to vary the thrust during a timespan using this utility.

- 7.2.3.5 Using the file select dialog box, select which ASCII keplerian/TOD file to be used to compute north/south pole crossing times. This should be the same output file created during section 7.2.3.1 above.
- 7.2.3.6 On the right hand side of the GUI, select whether inclination is to be increased or decreased. Figure 20 above shows graphically an orbit that is to have its inclination increased, while also employing a 20 degree wedge angle. Figure 21 below shows an orbit that is to have its inclination decreased while maintaining the same 20 degree wedge angle.

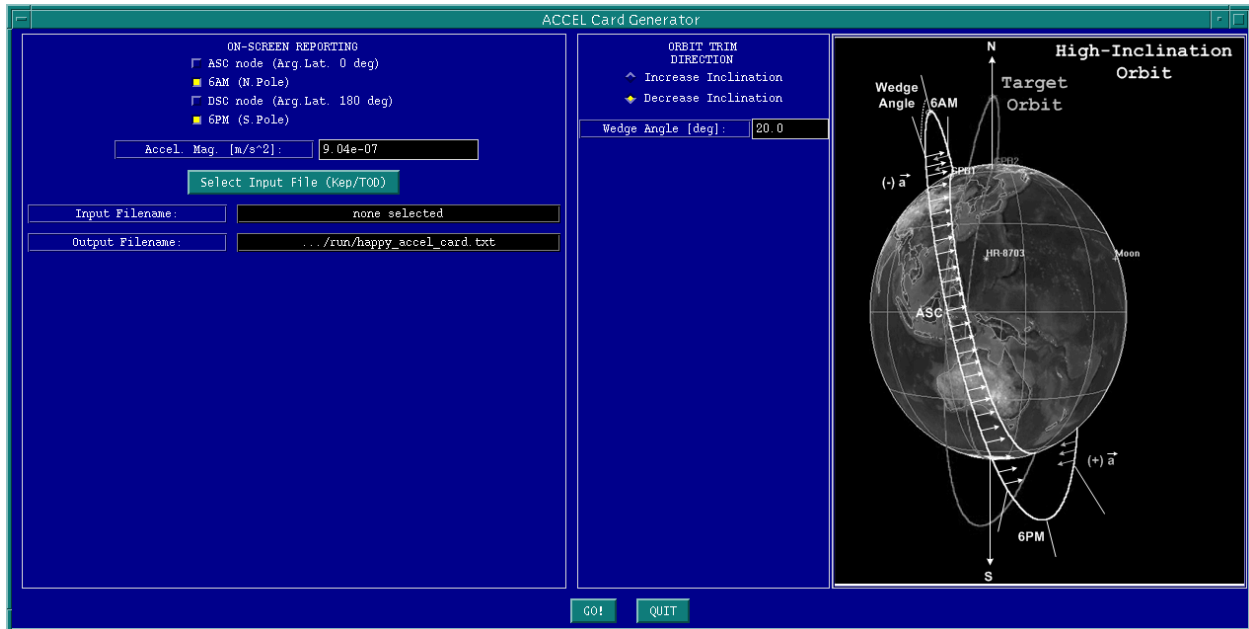


Figure 21: Orbit inclination to be decreased, 20 deg wedge

- 7.2.3.7 The “wedge angle” is essentially the angle on either side of the north and south poles, in which no thrust occurs. As the spacecraft’s thrust is least effective during this portion of the orbit, it has been shown by H. Small that a significant amount of fuel can be saved, while not drastically extending the time for orbit trim, by not thrusting near the poles. Using the input box, any wedge angle can be input into the utility. Should the user decide not to use a wedge angle, simply input “0.0” into the input box. Note, the graphic image changes to reflect no wedge angle:



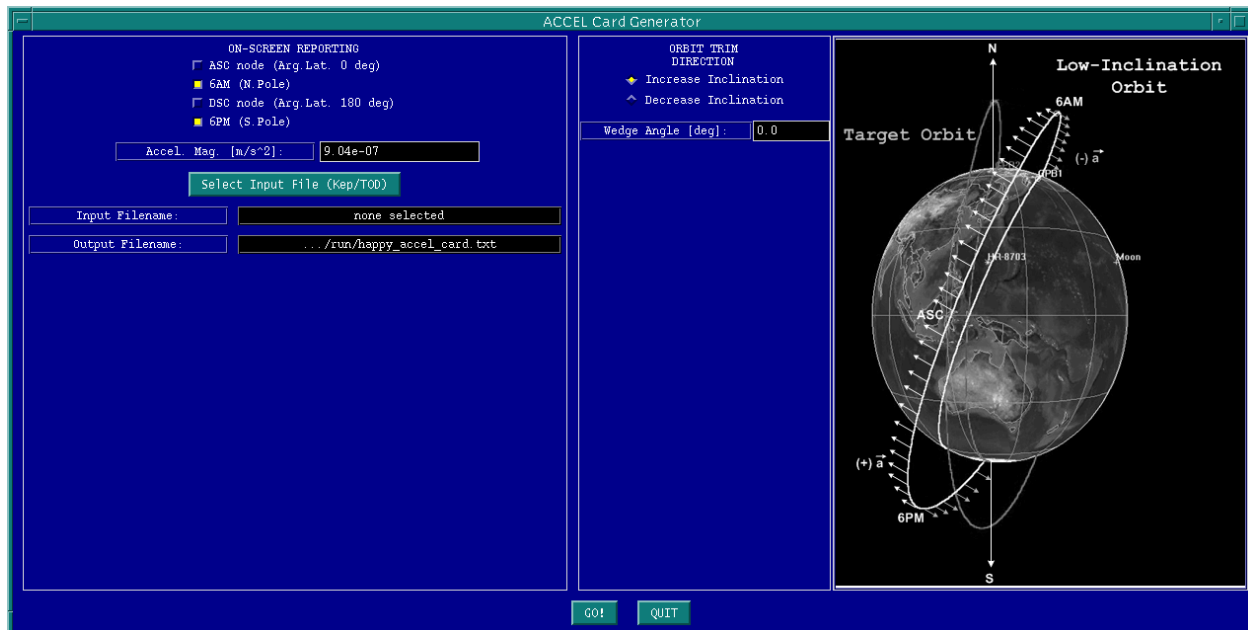


Figure 22: Orbit inclination to be increased, no wedge

Note the other implication of using the wedge angle: increased number of ACCEL cards to input into MicroCosm. In the case of no wedge angle, we need ACCEL cards only when the thrust direction flops at each north/south pole crossing. However, with a non-zero wedge angle, a record is needed to specify 0 thrust at the beginning of the wedge (6AM minus wedge angle, aka “6AMmWA”), another record specifying the start of thrust at the end of the wedge angle (6AM plus wedge angle, aka “6AMpWA”). This process is repeated at the south pole/6PM (“6PMmWA”, “6PMpWA”).

- 7.2.3.8 Click “GO!” to run the utility. Important on-screen info can be seen from the UNIX terminal. The orbital events can be seen in the first column, followed by approximated time of that orbital event (with the MicroCosm time format in parentheses). The final column is the associated argument of latitude in degrees and the sign of the thrust. Figure 23 shows sample on-screen output for the case of a 20 degree wedge. Figure 24 shows the same case but without any wedge.

/home/twalsh/PERL/ACCEL_CARD					
Window Edit Options					Help
6AMmWA	2004:070:13:16:30.000	(040310131630.000)	68.897		
6AM	2004:070:13:22:30.000	(040310132230.000)	90.977	-1	
6AMpWA	2004:070:13:27:30.000	(040310132730.000)	109.377		
6PMmWA	2004:070:14:05:30.000	(040310140530.000)	249.548		
6PM	2004:070:14:11:30.000	(040310141130.000)	271.627	1	
6PMpWA	2004:070:14:16:30.000	(040310141630.000)	290.027		
6AMmWA	2004:070:14:54:30.000	(040310145430.000)	70.197		
6AM	2004:070:14:59:30.000	(040310145930.000)	88.597	-1	
6AMpWA	2004:070:15:05:30.000	(040310150530.000)	110.677		
6PMmWA	2004:070:15:43:30.000	(040310154330.000)	250.845		
6PM	2004:070:15:48:30.000	(040310154830.000)	269.244	1	
6PMpWA	2004:070:15:54:30.000	(040310155430.000)	291.324		
6AMmWA	2004:070:16:32:30.000	(040310163230.000)	71.490		
6AM	2004:070:16:37:30.000	(040310163730.000)	89.889	-1	
6AMpWA	2004:070:16:42:30.000	(040310164230.000)	108.288		
6PMmWA	2004:070:17:20:30.000	(040310172030.000)	248.455		
6PM	2004:070:17:26:30.000	(040310172630.000)	270.535	1	
6PMpWA	2004:070:17:31:30.000	(040310173130.000)	288.934		
6AMmWA	2004:070:18:09:30.000	(040310180930.000)	69.099		
6AM	2004:070:18:15:30.000	(040310181530.000)	91.178	-1	
6AMpWA	2004:070:18:20:30.000	(040310182030.000)	109.578		
6PMmWA	2004:070:18:58:30.000	(040310185830.000)	249.744		
6PM	2004:070:19:04:30.000	(040310190430.000)	271.823	1	
6PMpWA	2004:070:19:09:30.000	(040310190930.000)	290.223		
6AMmWA	2004:070:19:47:30.000	(040310194730.000)	70.391		
6AM	2004:070:19:52:30.000	(040310195230.000)	88.790	-1	
6AMpWA	2004:070:19:58:30.000	(040310195830.000)	110.869		
6PMmWA	2004:070:20:36:30.000	(040310203630.000)	251.036		
6PM	2004:070:20:41:30.000	(040310204130.000)	269.435	1	
6PMpWA	2004:070:20:47:30.000	(040310204730.000)	291.515		
6AMmWA	2004:070:21:25:30.000	(040310212530.000)	71.685		
6AM	2004:070:21:30:30.000	(040310213030.000)	90.084	-1	
6AMpWA	2004:070:21:35:30.000	(040310213530.000)	108.483		
6PMmWA	2004:070:22:13:30.000	(040310221330.000)	248.651		
6PM	2004:070:22:19:30.000	(040310221930.000)	270.731	1	
6PMpWA	2004:070:22:24:30.000	(040310222430.000)	289.131		
6AMmWA	2004:070:23:02:30.000	(040310230230.000)	69.301		
6AM	2004:070:23:08:30.000	(040310230830.000)	91.381	-1	
6AMpWA	2004:070:23:13:30.000	(040310231330.000)	109.781		
6PMmWA	2004:070:23:51:30.000	(040310235130.000)	249.950		
6PM	2004:070:23:56:30.000	(040310235630.000)	268.349	1	
6PMpWA	2004:071:00:02:30.000	(040311000230.000)	290.429		
6AMmWA	2004:071:00:40:30.000	(040311004030.000)	70.600		
6AM	2004:071:00:45:30.000	(040311004530.000)	89.000	-1	
6AMpWA	2004:071:00:51:30.000	(040311005130.000)	111.080		
6PMmWA	2004:071:01:29:30.000	(040311012930.000)	251.250		
6PM	2004:071:01:34:30.000	(040311013430.000)	269.649	1	
6PMpWA	2004:071:01:40:30.000	(040311014030.000)	291.729		
6AMmWA	2004:071:02:17:30.000	(040311021730.000)	68.218		
6AM	2004:071:02:23:30.000	(040311022330.000)	90.299	-1	
6AMpWA	2004:071:02:28:30.000	(040311022830.000)	108.698		
output filename: /home/science/od_ops/run/happy_accel_card.txt					

Figure 23: On-screen output w/ 20 deg wedge

```

/home/twalsh/PERL/ACCEL_CARD
Window Edit Options Help
input filename: /home/science/od_ops/run/output/2004:055:21:58:36/kep_tod.txt
6AM 2004:069:21:05:30.000 (040309210530.000) 89.065 -1
6PM 2004:069:21:54:30.000 (040309215430.000) 269.711 1
6AM 2004:069:22:43:30.000 (040309224330.000) 90.361 -1
6PM 2004:069:23:32:30.000 (040309233230.000) 271.008 1
6AM 2004:070:00:21:30.000 (040310002130.000) 91.659 -1
6PM 2004:070:01:09:30.000 (040310010930.000) 268.628 1
6AM 2004:070:01:58:30.000 (040310015830.000) 89.278 -1
6PM 2004:070:02:47:30.000 (040310024730.000) 269.927 1
6AM 2004:070:03:36:30.000 (040310033630.000) 90.574 -1
6PM 2004:070:04:25:30.000 (040310042530.000) 271.220 1
6AM 2004:070:05:13:30.000 (040310051330.000) 88.187 -1
6PM 2004:070:06:02:30.000 (040310060230.000) 268.831 1
6AM 2004:070:06:51:30.000 (040310065130.000) 89.475 -1
6PM 2004:070:07:40:30.000 (040310074030.000) 270.121 1
6AM 2004:070:08:29:30.000 (040310082930.000) 90.765 -1
6PM 2004:070:09:18:30.000 (040310091830.000) 271.414 1
6AM 2004:070:10:06:30.000 (040310100630.000) 88.381 -1
6PM 2004:070:10:55:30.000 (040310105530.000) 269.031 1
6AM 2004:070:11:44:30.000 (040310114430.000) 89.678 -1
6PM 2004:070:12:33:30.000 (040310123330.000) 270.329 1
6AM 2004:070:13:22:30.000 (040310132230.000) 90.977 -1
6PM 2004:070:14:11:30.000 (040310141130.000) 271.627 1
6AM 2004:070:14:59:30.000 (040310145930.000) 88.597 -1
6PM 2004:070:15:48:30.000 (040310154830.000) 269.244 1
6AM 2004:070:16:37:30.000 (040310163730.000) 89.889 -1
6PM 2004:070:17:26:30.000 (040310172630.000) 270.535 1
6AM 2004:070:18:15:30.000 (040310181530.000) 91.178 -1
6PM 2004:070:19:04:30.000 (040310190430.000) 271.823 1
6AM 2004:070:19:52:30.000 (040310195230.000) 88.790 -1
6PM 2004:070:20:41:30.000 (040310204130.000) 269.435 1
6AM 2004:070:21:30:30.000 (040310213030.000) 90.084 -1
6PM 2004:070:22:19:30.000 (040310221930.000) 270.731 1
6AM 2004:070:23:08:30.000 (040310230830.000) 91.381 -1
6PM 2004:070:23:56:30.000 (040310235630.000) 268.349 1
6AM 2004:071:00:45:30.000 (040311004530.000) 89.000 -1
6PM 2004:071:01:34:30.000 (040311013430.000) 269.649 1
6AM 2004:071:02:23:30.000 (040311022330.000) 90.299 -1
output filename: /home/science/od_ops/run/happy_accel_card.txt

```

Figure 24: On-screen output w/ no wedge

7.2.3.9 The ACCEL cards are written to a file in the od\_ops/run directory. View the file by typing:

```
> more /home/science/od_ops/run/happy_accel_card.txt
```

7.2.3.10 Figure 25 shows what the ACCEL card file looks like. It is also important to understand how MicroCosm interprets accelerations and the time at which they are applied. Instead of the conventional logic of “*from this time on, the thrust shall be*”, MicroCosm ACCEL cards are specified as “*up until this time, the thrust has been*”. Hence the first ACCEL record in the file (with no time associated with it) tells MicroCosm “*until further notice, the thrust is zero*”. The second card, which the user must manually input the first actual on-time of thrusting tells MicroCosm “*up until this time, the thrust was zero.*”

The third record actually tells MicroCosm the amount of thrust and the direction by saying “*from the previous time up until now, the thrust has been  $X.XXXD-XX \text{ m/s}^2$ .*”

```

/home/science/od_ops/run
Window Edit Options Help
[18] more happy_accel_card.txt
** paste the following lines after the DRAG card in MicroCosm setup file **

ACCEL 2      9999999      0.0
ACCEL 2      9999999      0.0      time for first acceleration YYMMDDHHMMSS.ss
ACCEL 2      9999999      9.04000000D-07  40309210530.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40309215430.0 0.0
ACCEL 2      9999999      9.04000000D-07  40309224330.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40309233230.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310002130.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310010930.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310015830.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310024730.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310033630.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310042530.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310051330.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310060230.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310065130.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310074030.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310082930.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310091830.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310100630.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310105530.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310114430.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310123330.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310132230.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310141130.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310145930.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310154830.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310163730.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310172630.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310181530.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310190430.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310195230.0 0.0
ACCEL 2      9999999     -9.04000000D-07  40310204130.0 0.0
ACCEL 2      9999999      9.04000000D-07  40310213030.0 0.0
--More-- (85%)

```

Figure 25: *happy\_accel\_card.txt* file

- 7.2.3.11 Using the procedure outlined in section 7.2.2, re-run the OD over the desired span.
- 7.2.3.12 **BEFORE** running cosmc or cosme in section 7.2.2.4, open a new UNIX window, change to the `od_ops/run` directory and type:
 

```
> ls -ltr setup_2*.gpb
```
- 7.2.3.13 Edit the most recent setup file in the listing using `vi` or similar editor.
- 7.2.3.14 Immediately after the “DRAG” card in the setup file, paste the entire contents of the `happy_accel_card.txt` file. Make sure to manually remove the header from the `happy_accel_card.txt` file, and to manually add the time of the first thrusting in the MicroCosm format of `YYMMDDHHMMSS.ss`.

- 7.2.3.15 With the nominal orbit trim start/stop times properly inserted into the MicroCosm ACCEL card, the user should determine if any interruptions to orbit trim thrusting (beyond those associated with a given wedge angle) occurred during the processing span. Such interruptions should be reflected in the orbit trim Thrust\_wedge.txt file associated with running the Trimgp.m program (see page 9 of the orbit trim procedure P0985 Rev A). Any such interruptions to orbit trim thrusting found in the Thrust\_wedge.txt should be included into the ACCEL card by entering a new line for each start and end time at the appropriate point in the file.
- 7.2.3.16 Save the setup file and exit the editor.
- 7.2.3.17 Resume the OD procedure at section 7.2.2.4.

## 8.0 PROCEDURE COMPLETION

Upon generating binary trajectory and S-matrix files, QA must be performed to verify the validity and accuracy of the run and solution files must be archived. Three QA steps are listed in the following subsections, along with a checklist that outlines the QA process, and archive procedures.

### 8.1 VERIFICATION / SUCCESS CRITERIA

#### 8.1.1 GPS Data statistics

Review the GPS data summary report from running ORBITget.pl in step 7.1 to ensure that sufficient Mode 50 data are available for processing.

```
> more /projects/od/OFILE/GPS/gps-stats.txt
```

#### 8.1.2 Run Conditions and Solution Epoch Validation

- Verify that final line in outtdf.dat, outctl.dat, and outexc.dat indicates successful execution of MicroCosm TDF, cosmc, and cosme steps, respectively.
- Verify that solution at epoch is within expected tolerance and at correct date/time.
- Verify that solution covariance is within expected tolerance (TBS).
- Verify start/stop times of trajectory file and output interval.
- If necessary, dump the output of the trajectory and covariance files using the procedure in step 8.1.4.

#### 8.1.3 Plotting GPS Observation Residuals

To determine how well the computed orbit fit the observational data, it is useful to plot the residuals (both X/Y/Z and R/I/C) over the timespan of the computed trajectory.

- 8.1.3.1 From the directory containing the results of the GPS Microcosm run, type:

```
> ~twalsh/PERL/Residuals/GPS_Residual_Extractor.pl outexc_odw.txt
```

- 8.1.3.2 Using the computed mean vector at the epoch  
(`report_mean.doc`), enter the epoch time (MM/DD/YYYY  
HH:MM:SS), the mean inclination [rad], node [rad], SMA [km], A,  
B, and phi.
- 8.1.3.3 Note that upon completion of the script, three new files will be  
created in this directory: `XYZ_residuals.txt`,  
`RIC_residuals.txt`, and `GPS_Residual_Plotter.m`.
- 8.1.3.4 Run Matlab and execute the `GPS_Residual_Plotter.m` script.  
The script will auto-generate two pages, 3 plots per page,  
suitable for printing. Note the overall magnitude of the RIC  
residuals in determining the overall fit quality of the computed  
trajectory to observational data.

#### 8.1.4 Solution Overlap Compare

Each operational GP-B trajectory file for a given day will include a 3-hour overlap with the previous day and a 3-hour overlap with the next day (i.e. the OD trajectory file for day N will have a start time of 21:00 on day N-1 and a stop time of 03:00 on day N+1). This overlap will permit operators to perform a comparison of the current solution to that of the previous day's solution.

8.1.3.1 Copy the binary trajectory file from the current and previous day's solution into the `/home/science/od_ops/run/orbdiff/` directory.

8.1.3.2 Create a file called `orb_diff.in` with the following default format:

132 35 Name of first orbit file ID of satellite Name of second orbit file ID of satellite Start time End time
---

See the section of "Orb\_Diff INPUT CONTROL FILE" in MicroCosm Manual Orb\_Diff for details.

8.1.3.3 Make sure `filist.odf` contains proper file lists. The default files are as follows:

Filist.odf 01/home/science/od_ops/slp/jplde403.dat 02/home/science/od_ops/tables/tables.dat 05orb_diff.in 06outodf.dat
--

8.1.3.4 Execute:

Type **orb\_diff** at the UNIX command line.  
 Output from this command is an ephemeris of trajectory differences, written to `outodf.dat`, in Cartesian(XYZ) position/velocity and radial, cross track and along track(RCL) position and velocity between 3-hr overlaps on two consecutive orbit determination solutions.

- 8.1.3.5** Using the `outodf.dat` file, a plot of the RCL position and velocity can be created by running the Matlab script, `orbdiff_plot.m`.

## **8.1.5 Post-Processing / Creation of ASCII Trajectory File and Covariance File**

This section describes the post-processing of the MicroCosm binary trajectory and S-matrix files required to generate corresponding ASCII files for QA purposes.

- 8.1.5.1** Open or create `dmptrj.in`.  
 In the body of the file, write out the name of the binary trajectory file. In the example shown in 7.2.1.2, the binary trajectory file name is `orbfil.dat`.
- 8.1.5.2** Execute the command `dmptrj <dmptrj.in> dmptrj.out` at the UNIX command line.  
`dmptrj.out` is the name of the ASCII trajectory file that will be output. This file can be named whatever the user desires.
- 8.1.5.3** The above steps can be repeated for the output covariance file, `smatrix.dat`. The MicroCosm executable for output of an ASCII covariance file is `dmpemt.out`. The input file will be `dmpemt.in`

## **8.2 SOLUTION ARCHIVE AND MEAN ELEMENT COMPUTATION**

### **8.2.1 Archive to OD Database**

The Orbit Determination Database Input/Output Utility (ODBIU) consists of two applications, ODBIU\_trajectory and ODBIU\_vectors. The main use of ODBIU\_trj application is to process binary trajectory and covariance files created by Microcosm, calculate orbit element vectors from this data, and store that data in the Orbit Determination SYBASE Database. The ODBIU\_vectors application provides a means to extract the vectors stored in the OD Database.

#### **ODBIU\_trajectory Application**

[sample execution string ==>

ODBIU\_trajectory -m<username> -p<password> -G -d120

where *username* and *password* are the SYBASE username and password ]

The ODBIOU\_trj application allows the following command line options:

- t <filename> Name of the Microcosm binary trajectory file. If not provided and -i is not used, the user will be queried.
- c <filename> Name of the Microcosm S-Matrix (Covariance) file. If not provided and -i is not used, the user will be queried.
- i <filename> Name of a ASCII input file. Not allowed with either -t or -c options.
- o <basename> Base name for output files i.e. if base is jan1\_2002 the output will be jan1\_2002\_oe.out, jan1\_2002\_avg.out, etc. If this is not provided, the user will be queried.
- d <delta> Number of minutes between vectors for calculating mean elements written to SYBASE. If set to 0, no vectors are written to SYBASE. If not provided, the default of 120 minutes is used.
- M <midpoint> Value in seconds to be used for the midpoint in the average orbit element computation. If not provided the default of 43200 sec (= noon) is used..
- G Data is GPS data. Not allowed with -S option.
- S Data is SLR data. Not allowed with -G option.  
If neither -G or -S are provided, the user will be queried.
- m <username> SYBASE username. If not provided, the application uses the value of the MISSION environment variable. If MISSION is not set, the application exits.
- P <password> SYBASE password. If not provided, the user is prompted.
- O Database output.

There are two types of output for the ODBIOU\_trajectory application. The first type are data records to the SYBASE Orbit Determination database (OD DB). See document S0505 Appendix F for the OD DB layout. The second type are ASCII tab-delimited files.

For the nominal processing of the trajectory/covariance files, the application will output to the OD DB Ephem table every position/velocity vector contained in the input file. Nominally this will be one vector every 2 seconds for 30 hours (54,000 vectors). Other calculations are done to produce mean orbit element vectors for each time point. These mean vectors are written to the OD DB Mean table at an interval defined by the -d option or a default of every 120 minutes. An average orbit element vector is also computed from the instantaneous orbit element vectors. This single vector is also output to the OD DB.

The OD DB Header table contains data found in the trajectory file's header buffer. The items inserted are file start and end time, fractional seconds for the start and end time, inertial coordinate system used, and the maximum degree and order of the gravitational model. The start and end time is stored as double precision value of the form YYMMDDhhmmss.0. For the inertial



coordinate system, 0 = true-of-date, 1= mean of 1950, and 2 = mean of J2000.

Data contained in the covariance (S-matrix) file is also output to the database. The OD DB Covariance table contains the values for the upper triangular elements of the S-Matrix. The Solution table contains label id values and their data values for solutions calculated by Microcosm. See the Microcosm documentation for the meaning of the solution label ids.

The ASCII output of the tool is five files:

```
<basename>.out
<basename>_pv.out
<basename>_oe.out
<basename>_avg.out
<basename>_cov.out
```

The <basename>.out file contains data read from the trajectory file's header and alphanumeric buffers. This includes the input card information.

The <basename>\_pv.out contains the following fields (separated by tabs) for each time point in the trajectory file:

```
Seconds of day
UTC time
Right ascension of Greenwich in radians
X axis position in meters
Y axis position in meters
Z axis position in meters
X axis velocity in m/sec
Y axis velocity in m/sec
Z axis velocity in m/sec
```

The <basename>\_oe.out contains the following fields (separated by tabs) for each time point in the trajectory file (the first 9 fields are identical to those in <basename>\_pv.out):

```
Seconds of day
UTC time
Right ascension of Greenwich in radians
X axis position in meters
Y axis position in meters
Z axis position in meters
X axis velocity in m/sec
Y axis velocity in m/sec
Z axis velocity in m/sec
```

(The next 6 fields are the instantaneous orbit elements)

```
Inclination in radians
Right ascension of ascending node (omega) in radians
A = eccentricity * cos(argument of perigee)
B = eccentricity * sin(argument of perigee)
```

Semi major axis in meters  
 Mean argument of latitude (phi) in radians  
 (The next 6 fields are the mean orbit elements)  
 Inclination in radians  
 Right ascension of ascending node (omega) in radians  
 $A = \text{eccentricity} * \cos(\text{argument of perigee})$   
 $B = \text{eccentricity} * \sin(\text{argument of perigee})$   
 Semi major axis in meters  
 Mean argument of latitude (phi) in radians

The <basename>\_avg.out file contains only a single vector. This is the vector computed from averaging 24 hours instantaneous orbit element vectors nominally around 43200 seconds (12 noon). The fields are:

Seconds of day  
 UTC time  
 Inclination in radians  
 Right ascension of ascending node (omega) in radians  
 $A = \text{eccentricity} * \cos(\text{argument of perigee})$   
 $B = \text{eccentricity} * \sin(\text{argument of perigee})$   
 Semi major axis in meters  
 Mean argument of latitude (phi) in radians

The <basename>\_cov.out file contains the following groups of data:

Parameter group ID  
 Solution IDs and values  
 S-Matrix data

This application also has the capability to read an ASCII file for input as an alternative to the trajectory and covariance files. The input filename for such a run would be provided with the -i command line option. The file would contain a user-determined number of input vector records, with a minimum of two vectors required. Each record would contain the following fields on one line separated by tabs:

Year in the form YY (e.g. 02 = 2002)  
 Date/time in the form MMDDhhmmss (e.g. 1015143500 = Oct 15 2:35:00 PM)  
 Right ascension of Greenwich in radians  
 X Position in meters  
 Y Position in meters  
 Z Position in meters  
 X Velocity in meters  
 Y Velocity in meters  
 Z Velocity in meters

The file outputs for this type of run will be the same, though unless a full 24-hour period of vectors is provided, the average vector is meaningless. The OD DB output is also the same though there is no Solution, Covariance, or Header table data. Note that a default 19x19 degree and order gravity model

is used to compute a mean vector when an ASCII input file is used (-i option).

### ODBIU\_vectors Application

[sample execution string ==>

ODBIU\_vectors -m<username> -p<password> -S

where *username* and *password* are the SYBASE username and password ]

The ODBIU\_vectors application allows the following command line options:

- b <begin\_time> The starting value for the desired time range.  
If this is not provided the user is prompted.
- e <end\_time> The ending value for the desired time range.  
If this is not provided the user is prompted.
- o <filename> The output filename. If this is not provided  
the user is prompted.
- C Extract Cartesian vectors. This is not  
allowed with -M or -S
- M Extract mean orbit element vectors. This is  
not allowed with -C or -S.
- S Extract Cartesian vectors and output data in  
STK format. This is not allowed with -C or -M
- m <mission> User name for SYBASE. If not supplied, the  
value of the MISSION environment variable is  
used. If MISSION is not set, the application exits.
- p <password> SYBASE password. If not provided, the  
user is prompted.

If the C, M, or S options are not provided the user is prompted for the type of output desired. With the C option, an output file is created with all Cartesian vectors within the given range. The output is tab delimited and contains the following fields:

OD DB id value

UTC time in the form MM/DD/YY hh:mm:ss

X position in meters

Y position in meters

Z position in meters

X velocity in m/sec

Y velocity in m/sec

Z velocity in m/sec

With the M option, an output file is created with all mean orbit element vectors within the given range. The output is tab delimited and contains the following fields:

OD DB id value

UTC time in the form MM/DD/YY hh:mm:ss

i = inclination in radians  
omega = right ascension of ascending node in radians  
A = eccentricity \* cos(argument of perigee)  
B = eccentricity \* sin(argument of perigee)  
a = Semimajor axis in meters  
phi = argument of latitude  
Mean or Avg flag – 1=mean orbit element vector, 0 = vector computed from averaging instantaneous orbit elements

With the S option, an output file is created in STK external ephemeris format. Note that when selecting an epoch vector, fractional seconds are not recognized by ODBIOUT.

### 8.2.2 Archive to Trajectory File Directory

Maintain previous 30-hr trajectory files in separate GPS and SLR directories labeled according to the following convention:

[SCI/MP/ANA] [GPS/SLR/LCH] [TOD/J2000] [Start Time] [End Time] [comment field]

... where:

[Start/End Time] format: YYYY:DOY:HH:MM:SS

Note that it is imperative to indicate either “TOD” or “J2000”, as this is how the Science Team scans the database for science solutions. Additionally, the entire length of the database label may not exceed 80 characters.

### 8.2.3 OD Database Maintenance

Periodically, it may be necessary to remove entries from the database or change the label of the database entry. Both of these tasks can be accomplished with the PERL tool, `ODcleanup.pl`. Run the tool from a UNIX prompt:

```
> ~pmcgown/PERL/ODcleanup.pl
```

The tool will give a listing of all entries in the OD database. To edit or delete an entry, type in the 10-digit generation ID number, and follow the directions onscreen.

### 8.3 ORBIT DETERMINATION PRODUCTS

The product delivery form shown in Appendix C will be used to notify interfacing elements of the availability of a routine product update or a special request product. Nominal product delivery schedules are defined for each product in the ODF IDD (S0505).

### 8.3.1 Mission Planning Orbit State Vector (STK e-file)

See Section 8.2 – ODBIOU\_vectors application.

### 8.3.2 GN/SN Backup Acquisition State Vector

Once ODBIU\_trajectory has been run and a <basename>\_oe.out file has been generated, move the oe.out file to the

/home/science/od\_ops/run/mat\_files directory. Run the report\_inst Matlab utility in order to create hard-copy report of the instantaneous OD

solution at noon of the operational day. The name of the instantaneous state vector file is `inst_orbit_report.doc`. This report should be faxed to the FDF as necessary using the product delivery form in Appendix C.

Using Matlab and the same `<basename>_oe.out` file, run the `report_mean` utility to create `orbit_report_mean.doc`. Both the `orbit_report_mean.doc` and the `inst_orbit_report.doc` should be printed and put in the OD log book.

#### 8.3.3 SLR Backup Acquisition State Vector

Once ODBIOU\_trajectory has been run and a `<basename>_oe.out` file has been generated, move the `oe.out` file to the `/home/science/od_ops/run/mat_files` directory. Run the `report_inst` Matlab utility in order to create hard-copy report of the instantaneous OD solution at noon of the operational day. This report should be faxed to the GSFC/ILRS as necessary using the product delivery form in Appendix C.

#### 8.3.4 ATCS Orbit State Vector

After running ODBIOU\_trajectory, a set of mean orbit state vectors will be loaded into the OD database. Move the `oe.out` file to the `/home/science/od_ops/run/mat_files` directory, and run the `report_mean` Matlab utility in order to create a hardcopy report of the OD solution at noon of the operational day. This report should be delivered to the Mission planning group as necessary using the product delivery form in Appendix C.

## Operator Name and Date of Execution

	Name (print)	Signature	Date
Operator			
OD Lead			

## Checklist

Step	Description	Comments
1: <input type="checkbox"/>	Obtain input GPS data	Data processing: TANDO.pro <input type="checkbox"/> GPS_extract <input type="checkbox"/> Start date/time: _____ End date/time: _____
2: <input type="checkbox"/>	Telemetry Statistics	# of data pnts: _____ # of pnts in mode 50: _____ # of non-stale pnts: _____ # of pnts with SQI < .1: _____
3: <input type="checkbox"/>	PCE file generation	Filename / location: _____
4: <input type="checkbox"/>	Run od_widget.ptk	Modeling: <input type="checkbox"/> Orbit Trim Thrust <input type="checkbox"/> Drag-Free Termination: <input type="checkbox"/> Success <input type="checkbox"/> Fail; Error msg: _____
5: <input type="checkbox"/>	QA check of output span	Trajectory file start time: _____ Trajectory file end time: _____
6: <input type="checkbox"/>	QA check of solution statistics	Solution date/time: _____ <input type="checkbox"/> Mean Element Report attached <input type="checkbox"/> Instantaneous Orbit Solution Report attached RMS Residuals: Rad: _____ C-trk: _____ along trk: _____ Solution Uncertainty from Covariance (S-) Matrix: Max Pos Standard Deviation: _____ meter Max Vel Standard Deviation: _____ cm/s
7: <input type="checkbox"/>	Comparison of overlapping trajectory files (using orb_diff)	Comparison File Name: _____ rms pos diff (m): Rad: _____ C-trk: _____ along trk: _____ rms vel diff (mm/s): Rad: _____ C-trk: _____ along trk: _____
8: <input type="checkbox"/>	Archive completion	OD DB entries: _____ _____ Traj file location: _____ STK "e" file location?: _____
9: <input type="checkbox"/>	Product Delivery/Notification	<input type="checkbox"/> Mission Plan/STK? <input type="checkbox"/> FDF? <input type="checkbox"/> Science? <input type="checkbox"/> Mission Plan/ATC? <input type="checkbox"/> SLR? <input type="checkbox"/> Orbit Trim?
10.	Operator Comments	

## Appendix A

### Orbit Determination Setup File

setup\_2002:247:17:10:30.gpb

Microcosm Setup/State Vector File

created by od\_widget.ptk at 2002:247:17:10:30

```

----- START GLOBAL SET CARDS -----
EARTH          25 25          0.398600441500D+15 6378136.300000 298.25640000
OBSVU 50
PLANET 10100
PLANET 00200
PLANET 00301
PLANET 10400
PLANET 00500
PLANET 10600
PLANET 10700
PLANET 10800
PLANET 10900
PLANET 09999
PUNCH 0
SMATRX 2          -1.00000000
PRNTVU60212122 1112112
TERMVU 1212212221
VLIGHT          299792458.0          299792458.0
VOPTIM 361121
ENDGLB 15 3
Begin Mandatory
ARC card
----- START ARC SET CARDS -----
REFSYS2099          020904171030.0
SATPAR 0 1 0259999999 23.00000000          3240.900
EPOCH          020710180000.0          020710180000.0          020712000000.0
ELEMS114          4619118.193          -1405508.272          -5108466.511
ELEMS2          -5230.215645          1587.569998          -5170.785076
STEP 0 0 0 0 09999999          2.0
EDIT 0          3.5          1.0E06
DRAG 0          9999999 2.30000000
SOLRAD          9999999 1.300000
ORBFIL0021          9999999 020710180000.0          020712000000.0          2.0
ORBTVU0021          9999999 020710180000.0          020712000000.0          7200.0
DATA
PREPRO
SELECT          020101000000.0          050101000000.0
ENDARC
ENDALL

```

## Appendix B

### Orbit Propagation Setup File

The MicroCosm control file listed in this Appendix is used to propagate a set of epoch elements on November 1, 2002 at 21:00 UTC to create a 30-hour propagated trajectory file. Input Keplerian elements are contained in the input records **ELEMS1** and **ELEMS2**. When generating a propagated orbit without any observation data (i.e., no pce.dat file used), the Data Selection/Deletion Subgroup should not be present. Therefore, the records **DATA**, **SELECT** and **PREPRO** have all been eliminated from the setup file. Also, as this trajectory file was intended to simulate GP-B "drag-free" operation, solar radiation and atmospheric drag were not modeled. This was accomplished by setting the cross-sectional area in the **SOLRAD** record to zero. Finally, the **ORBTUV** record was included to enable the output of Keplerian elements to an ASCII file called aeiout.dat at an interval of 1800 seconds.

#### Setup.ctlp.gpb

```
GPB orbit propagation, JGM3 <70x70>
Estimate GPB position and velocity
----- START GLOBAL SET CARDS -----
---
EARTH          19 19          0.398600441500D+15 6378136.300000 298.25640000
OBSVU 50
PLANET 10100
PLANET 10200
PLANET 10301
PLANET 10400
PLANET 10500
PLANET 10600
PLANET 10700
PLANET 10800
PLANET 10900
PLANET 19999
PUNCH 0
SMATRX 2          -1.0000
PRNTVU60212122 1112112
TERMVU 1212212221
VLIGHT          299792458.0          299792458.0
VOPTIM 361121
ENDGLB 15 3
Using PCE position data to compute the GPB orbit
Solve for GPB position abd velocity
----- START ARC SET CARDS -----
--
REFSYS2099          021101210000.00000000
SATPAR 0 0 0 709999999 0.00000000000000E+00 1.15290000E+02
EPOCH          021101210000.000 021101210000. 021103030000.
ELEMS124          7027400.00 0.0014 0.00
ELEMS2          163.12 71.3 -71.3
STEP 0 0 0 0 09999999 10. 10.
EDIT 0          3.5 1.0E06
ORBFIL0021 9999999 021101210000.0 021103030000.0 60.0
ORBTUV 23 9999999 021101210000.0 021103030000.0 1800.0
ENDARC
ENDALL
```



---

## Appendix C

---

### Orbit Determination Product Delivery form

## Orbit Determination Product Delivery Form

Date of Delivery: \_\_\_\_\_

File Name: \_\_\_\_\_ Size: \_\_\_\_\_ Date: \_\_\_\_\_

Check Product to Be Delivered:

	Product	Destination
	Mission Planning Orbit State Vector (STK e-file)	Mission Planning (STK)
	ATCS Orbit State Vector	Mission Planning (ParameterGen)
	ATCS Orbit Trim Parameters	Mission Planning (ParameterGen)
	GN/SN Backup Acquisition State Vector	GSFC/FDF
	SLR Backup Acquisition State Vector	GSFC/ILRS
	Special Product: _____	

☐ Attachments: \_\_\_\_\_

Comments:

Created by: \_\_\_\_\_/\_\_\_\_\_  
OD Analyst Date

Approved by: \_\_\_\_\_ / \_\_\_\_\_  
OD Lead Date

Received by: \_\_\_\_\_ / \_\_\_\_\_  
Mission Planing Lead                      Date

Scheduled: \_\_\_\_\_  
Date(s)

Implemented by: \_\_\_\_\_ / \_\_\_\_\_  
Mission Planner Date

SCCS rev: \_\_\_\_\_