

# IRIS Flight Dynamics Comparison Software Manual

M. Weber  
for the IRIS Team

April 26, 2013

# Contents

|          |                                                                                                                                                    |          |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>1</b> | <b>Introduction</b>                                                                                                                                | <b>3</b> |
| <b>2</b> | <b>Overview</b>                                                                                                                                    | <b>3</b> |
| <b>3</b> | <b>Using the IRIS FD Comparison software</b>                                                                                                       | <b>5</b> |
| 3.1      | Running <code>iris_fd.compare_files.pro</code> . . . . .                                                                                           | 6        |
| 3.2      | Reading the text report (screen, <i>reportfile</i> ) . . . . .                                                                                     | 7        |
| 3.2.1    | Report: Common structure (all file types) . . . . .                                                                                                | 7        |
| 3.2.2    | Report: Complex structure (file types “view_XXdeg”) . . . . .                                                                                      | 12       |
| 3.3      | Reading the data <code>DELTA_var</code> plots (screen) . . . . .                                                                                   | 15       |
| 3.3.1    | Plots: Common structure (all file types) . . . . .                                                                                                 | 15       |
| 3.3.2    | Plots: Complex structure (file types “view_XXdeg”) . . . . .                                                                                       | 18       |
| 3.4      | Reading the saved data variables ( <i>varfile</i> ) . . . . .                                                                                      | 20       |
| 3.4.1    | Description of saved filenames ( <code>FDF_file</code> , <code>FDT_file</code> ,<br><code>reportfile</code> , and <code>varfile</code> ) . . . . . | 21       |
| 3.4.2    | Description of parsed data ( <code>FDF_parsed</code> and <code>FDT_parsed</code> ) . . . . .                                                       | 22       |
| 3.4.3    | Description of post-matching variables ( <code>matches</code> , <code>FDF_unmatched</code> ,<br>and <code>FDT_unmatched</code> ) . . . . .         | 26       |

## List of Figures

|    |                                                                      |    |
|----|----------------------------------------------------------------------|----|
| 1  | Example of header and data for an FDF file of type “ground”. . . . . | 4  |
| 2  | Example of header and data for an FDT file of type “ground”. . . . . | 4  |
| 3  | Example of basic text report structure. . . . .                      | 8  |
| 4a | Example of complex text report structure (page 1 of 2). . . . .      | 12 |
| 4b | Example of complex text report structure (page 2 of 2). . . . .      | 13 |
| 5  | Example of basic data plot structure. . . . .                        | 17 |
| 6  | Example of multiplot structure . . . . .                             | 19 |
| 7  | Example of basic structure for post-matched variables. . . . .       | 27 |
| 8  | Example of complex structure for post-matched variables. . . . .     | 29 |

## List of Tables

|   |                                                           |    |
|---|-----------------------------------------------------------|----|
| 1 | Types and Data Fields of Flight Dynamics files . . . . .  | 5  |
| 2 | List of downlink stations . . . . .                       | 15 |
| 3 | Contents of IDL savefile ( <i>varfile</i> ) . . . . .     | 20 |
| 4 | Contents of <code>FD*_parsed</code> (“aaz”) . . . . .     | 23 |
| 5 | Contents of <code>FD*_parsed</code> (“eclipse”) . . . . . | 24 |
| 6 | Contents of <code>FD*_parsed</code> (“events”) . . . . .  | 24 |

|    |                                                              |    |
|----|--------------------------------------------------------------|----|
| 7  | Contents of <code>FD*_parsed ("ground")</code> . . . . .     | 24 |
| 8  | Contents of <code>FD*_parsed ("radzones")</code> . . . . .   | 25 |
| 9  | Contents of <code>FD*_parsed ("view_10deg")</code> . . . . . | 25 |
| 10 | Contents of <code>FD*_parsed ("view_5deg")</code> . . . . .  | 25 |

# 1 Introduction

The Lockheed Martin Flight Dynamics Team (FDT), located at the Lockheed Martin Solar and Astrophysics Laboratory (LMSAL), will generate flight dynamics products based on NORAD Two-Line Elements (TLEs) during the Early Orbit and Operations mission phases. The LM FDT will operate in shadow mode initially, comparing products to those generated by the GSFC FDF.

— From the document “Interface Region Imaging Spectrometer (IRIS) Interface Control Document (ICD) Between The Mission Operations Center (MOC) And The Lockheed Martin Flight Dynamics Team (LM-FDT)”.

The IRIS Flight Dynamics Team (“FDT”) was tasked with providing software to generate flight dynamics products. (The specifications are given in the ICD cited above.) Initially, the flight dynamics products will be generated by the GSFC Flight Dynamics Facility (“FDF”). The IRIS Flight Dynamics Comparison software (i.e., the software described in this Manual) is a set of programs that provides a comparison report between a pair of corresponding FDF and FDT product files.

In its present form, the FD Comparison software does ***not*** provide a bottom-line assertion whether two products conform to some level of acceptability. Rather, this software simply generates plots and a text report that describe quantitatively how well the data fields agree. It is up to the user to review the report and discover whether there are any problems to pursue.

## 2 Overview

The set of flight dynamics products are comprised of several different types of ASCII text files, each type focusing on a particular type of flight dynamics event. (See Table 1 for a list of file (i.e., data product) types.) A single file contains a header and a table of event data, spanning some time period. (See Figure 1 for an example.) The events are time-ordered as rows, and the columns are the event’s data fields.

In order to compare an FDT-generated file against a corresponding FDF-generated file, one must first identify which event lines in the FDF file match up with which event lines in the FDT file. Then, for each matching event pair, the differences (or “deltas”) for each data field can be calculated. These data deltas can be compared against given tolerances to ascertain whether the FDT products are acceptably close to the FDF products. Finally, statistics can be generated on the distributions of the data deltas.

|                          |        |        |            |                      |
|--------------------------|--------|--------|------------|----------------------|
| Satellite-IRIS           |        |        |            | 27 Nov 2012 14:08:28 |
| Time (UTCG)              |        |        |            | Lat (deg)            |
|                          |        |        |            | Lon (deg)            |
|                          |        |        |            | Alt (km)             |
| -----                    | -----  | -----  | -----      | -----                |
| 29 Jan 2013 00:00:31.360 | 41.764 | 83.522 | 654.411538 |                      |
| 29 Jan 2013 00:01:31.360 | 45.386 | 82.307 | 653.946304 |                      |
| 29 Jan 2013 00:02:31.360 | 48.996 | 80.959 | 653.480295 |                      |
| 29 Jan 2013 00:03:31.360 | 52.594 | 79.441 | 653.001147 |                      |

Figure 1: Example of header and data for an FDF file of type “ground”.

As an example, Figures 1 and 2 show the first few lines for a pair of corresponding FDF and FDT files, respectively, of the “ground” type of data product. The other types of files are formatted in a similar manner, although the specific data field columns vary. Assuming a tolerance on the “Time” data field of at least 32 seconds, the FD Comparison software would make the following line matches: FDF #1 and FDT #1, FDF #2 and FDT #2, FDF #3 and FDT #3, and FDF #4 and FDT #4. Then the software would report information about the distribution of deltas on all four data fields (“Time”, “Lat”, “Lon”, and “Alt”).

The IRIS FD Comparison suite provides a single user-level routine (`iris_fd_compare_files.pro`) that accepts a pair of FDF/FDT filenames as input, reads the files, performs the event matching, and then produces a text report and a series of distribution plots on the data fields. In addition, the data products generated by the software are saved to an IDL savefile.

|                                     |        |        |             |                      |
|-------------------------------------|--------|--------|-------------|----------------------|
| Using TLE from 29 Jan 2013 00:00:00 |        |        |             | 29 Jan 2013 19:08:16 |
| Satellite-IRIS                      |        |        |             |                      |
| Time (UTCG)                         |        |        |             | Lat (deg)            |
|                                     |        |        |             | Lon (deg)            |
|                                     |        |        |             | Alt (km)             |
| -----                               | -----  | -----  | -----       | -----                |
| 29 Jan 2013 00:00:00.00             | 39.861 | 84.116 | 654.5748209 |                      |
| 29 Jan 2013 00:01:00.00             | 43.487 | 82.957 | 654.0993519 |                      |
| 29 Jan 2013 00:02:00.00             | 47.103 | 81.683 | 653.6293122 |                      |
| 29 Jan 2013 00:03:00.00             | 50.708 | 80.259 | 653.1523750 |                      |

Figure 2: Example of header and data for an FDT file of type “ground”.

Table 1: Types and Data Fields of Flight Dynamics files

| File Type  | Data Headers                                                                                                                       | Data Fields                                                                               | Matching Fields   |
|------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|-------------------|
| aaz        | (Direction), Time, AAZ Angle                                                                                                       | DIRECTION, TIME, ANGLE                                                                    | DIRECTION, TIME   |
| eclipse    | Start Time, Stop Time, Duration, Obstruction, Start Pass, Stop Pass, Current Condition, Worst Condition, Total Duration            | ST_TIME, EN_TIME, DURATION, ST_PASS, EN_PASS, C_COND, W_COND, TOT_DUR                     | ST_TIME, EN_TIME  |
| events     | Pass, Start Time, End Time, Apogee, Time of Apogee, Perigee, Time of Perigee, Period, Right Ascen, Lon Ascen Node, Lon Descen Node | PASS, ST_TIME, EN_TIME, APOGEE, T_APOG, PERIGEE, T_PERIG, PERIOD, RI_ASC, LO_ASC, LO_DESC | PASS              |
| ground     | Time, Lat, Lon, Alt                                                                                                                | TIME, LAT, LON, ALT                                                                       | TIME              |
| radzones   | Start Time, Stop Time, Duration                                                                                                    | ST_TIME, EN_TIME, DURATION                                                                | ST_TIME, DURATION |
| view_10deg | Start Pass, Start Time, End Time, Duration, Max Elevation, Max Elev Time                                                           | PASS, ST_TIME, EN_TIME, DURATION, MAX_EL, MAX_EL_TIME                                     | PASS              |
| view_5deg  | Start Pass, Start Time, End Time, Duration, Max Elevation, Max Elev Time                                                           | PASS, ST_TIME, EN_TIME, DURATION, MAX_EL, MAX_EL_TIME                                     | PASS              |

### 3 Using the IRIS FD Comparison software

The routine `iris_fd_compare_files.pro` is the only user-level program in the IRIS FD Comparison software suite. There are many subroutines that actually handle the work. Here is a list of what `iris_fd_compare_files.pro`

does:

- read the FDF/FDT files into IDL;
- look for matching event pairs;
- calculate deltas between the FDF and FDT data;
- generate a text report;
- printout the text report to the screen;
- write the text report to an ASCII text file (*reportfile*);
- plot all of the data delta distributions to the screen; and
- save the data variables to an IDL savefile (*varfile*).

In the following subsections, we look at:

- how to call the routine `iris_fd_compare_files.pro` (§ 3.1);
- what the contents of the text report are (§ 3.2);
- how to read the data delta plots (§ 3.3); and
- what the contents of the IDL savefile are (§ 3.4).

### 3.1 Running `iris_fd_compare_files.pro`

The IRIS FD Comparison software is distributed through the “iris” package of SolarSoft. Requirements for running this suite include working installations of IDL, SolarSoft, and the “gen” and “iris” SolarSoft packages. Here is how to run the user-level routine `iris_fd_compare_files.pro`:

```
IDL> iris_fd_compare_files, fdf_filename, fdt_filename, $  
IDL> reportfile=reportfilename, varfile=varfilename
```

**`fdf_filename`, `fdt_filename`** These arguments are *mandatory inputs*. They must each be a single string containing the filename of an FDF and FDT flight dynamics file, respectively. These inputs may not be arrays. The FDF and FDT files must be the same type of flight dynamics file. Table 1 lists the possible file types.

**reportfile** This keyword is *optional*, it can serve as an *input* or as an *output*, and it must be a single string that will be interpreted as the filename of an ASCII text file. AS INPUT: the text report will be written to *reportfile*. If the named file already exists, then the report text will be appended to it. If the named file does not yet exist, then the report text will be written to an ASCII file created with the name given by *reportfile*. If the **reportfile** keyword is not invoked, or if the value of *reportfile* is an empty string, then the report text will be written to an existing or new ASCII file with the default name ‘iris\_fd\_report\_file.txt’. AS OUTPUT: the name of the file to which the report text was written. If the keyword is set to an undefined variable name for input, then the default filename will be returned. NOTE: regardless of how the keyword *reportfile* is used or not used, the report will be written to a text file.

**varfile** This keyword is an *optional output*, and it will be a single string equal to the filename of an IDL savefile. The filename is hardset, and is in this format: **iris\_fd\_varfile\_type\_FDFyyyymmdd\_FDTyyyymmdd.sav**, where *type* is one of the file types from Table 1, and *yyyymmdd* are the date codes from the FDF and FDT filenames.

## 3.2 Reading the text report (screen, *reportfile*)

The results of the comparison of a pair of FDF and FDT files are summarized in a text report. This text report is (a) written to the screen, and (b) written to an ASCII text file. The exact same content is written to both devices. (The entire report is generated and stored internally as a string array and then printed as one block.) The name of the report file can be specified with the **reportfile** keyword in the call to **iris\_fd\_compare\_files.pro**. If the file does not pre-exist, then it will be created. If the file does pre-exist, then the report text will be appended under the assumption that the file is an ASCII text file. If the **reportfile** keyword is not used or provided with an undefined variable, then the program will proceed with the default filename of ‘iris\_fd\_report\_file.txt’.

The text report has a structure that follows a basic template that is explained in this subsection. The “view\_10deg” and “view\_5deg” types of files have an extra layer of complexity which is also reflected in the report structure. That extra complexity is explained later in this subsection.

### 3.2.1 Report: Common structure (all file types)

We now examine the basic content structure of the text report by looking in detail at an example. Figure 3 displays the comparison report for an

FDF/FDT pair of files of type “ground”. The different lines of content have been labeled and distinguished by alternating colors. (The colors provide no information except to group like lines of content.)

|                                                                                   |      |
|-----------------------------------------------------------------------------------|------|
| =====                                                                             | (A)  |
| REPORT ON COMPARING FDT VERSUS FDF OF THIS TYPE: ground                           | (B)  |
| FDF filename: FDF_IRIS_ground.20130129.V01.txt                                    | (C)  |
| FDT filename: IRIS_ground.20130129.V01.txt                                        | (D)  |
| Variables saved filename: iris_fd_varfile_ground.FDF20130129_FDT20130129.sav      | (E)  |
| FDF range of start times: 2013-01-29T00:00:31.360 / 2013-02-04T23:59:31.360       | (F)  |
| FDT range of start times: 2013-01-29T00:00:00.000 / 2013-02-04T23:59:00.000       | (G)  |
| Number of FDF/FDT entries: 10080 / 10080                                          | (H)  |
| Number of matches / FDF unmatched / FDT unmatched: 10080 / 0 / 0                  | (I1) |
| Number of unmatched FDT entries inside/outside of FDF time range: 0 / 0           | (I2) |
| DELTA.TIME [sec] mean/min/max/stdev/tol: -31.36/-31.36/-31.36/0.00/60.00          | (I3) |
| DELTA.TIME N_match/N_partial_orbits/N_errors: 10080/0/0                           | (J)  |
| DELTA.TIME was used to make matches, so by definition all DELTA.TIME LE TOL.TIME. |      |
| DELTA.LAT [deg] mean/min/max/stdev/tol: -0.00/-1.94/1.96/1.79/1.00                |      |
| DELTA.LAT N_match/N_partial_orbits/N_errors: 10080/0/0                            |      |
| Number of DELTA.LAT inside/outside TOL.LAT: 555 / 9525                            |      |
| DELTA.LON [deg] mean/min/max/stdev/tol: 2.07/0.39/14.25/3.16/1.00                 | (K)  |
| DELTA.LON N_match/N_partial_orbits/N_errors: 10080/0/0                            |      |
| Number of DELTA.LON inside/outside TOL.LON: 6396 / 3684                           |      |
| DELTA.ALT [km] mean/min/max/stdev/tol: 0.01/-1.72/1.04/0.73/50.00                 | (L1) |
| DELTA.ALT N_match/N_partial_orbits/N_errors: 10080/0/0                            | (L2) |
| Number of DELTA.ALT inside/outside TOL.ALT: 10080 / 0                             | (L3) |

Figure 3: Example of basic text report structure for filetype “ground”. The letter labels refer to explanations in § 3.2.1.

- (A) Every text report begins with one blank line and one separator line of equal signs.
- (B) The format of this line is:  
REPORT ON COMPARING FDT VERSUS FDF OF THIS TYPE: *type*  
where *type* is one of the file types from Table 1. One full text report only ever addresses one pair of FDF and FDT files of one matching type. A text report is not generated if the two files do not match in type.
- (C) The format of these two lines is:  
FDF filename: *FDF\_filename*  
FDT filename: *FDT\_filename*

These are the FDF and FDT filenames provided as inputs to `iris_fd_compare_files.pro`.

- Ⓓ The format of this line is:  
**Variables saved filename:** *varfile*  
This is the name of the IDL savefile which contains the data for this report. The format of the filename is hardset in the program. See § 3.1 for an explanation of the format of *varfile*. See § 3.4 for a discussion of the variables saved to this file.
- Ⓔ The format of these two lines is:  
**FDF range of start times:** *first\_timecode / last\_timecode*  
**FDT range of start times:** *first\_timecode / last\_timecode*  
Every event for every type of file has at least one time field. These two lines of the report indicate the timestamps for the first and last entries for each of the FDF and FDT files. Note that the last event in the file may also have an associated stop-time, which could be after the range indicated here. This part of the report helps to understand how much overlap in time the two files have.
- Ⓕ The format of this line is:  
**Number of FDF/FDT entries:**  $N_{FDF} / N_{FDT}$   
 $N_{FDF}$  and  $N_{FDT}$  are simply the raw number of events in the two files. This is before any matching or comparing between the files. It is possible and legitimate for these two numbers to be different even if the files were generated for the same time period; therefore, an error should not be presumed just because these numbers differ.
- Ⓖ The format of this line is:  
**Number of matches / FDF unmatched / FDT unmatched:**

$$N_{\text{matches}} / N_{FDF\_unmatched} / N_{FDT\_unmatched}$$

The program tries to make unique matching pairs of events between the two files. I.e., under the assumption that the FDT results attempt to recreate the FDF results, the software tries to identify the one-to-one matching. (Table 1 indicates which data fields are used for determining matches, per each file type.)  $N_{\text{matches}}$  is the count of matching pairs.  $N_{FDF\_unmatched}$  and  $N_{FDT\_unmatched}$  are the counts of leftover events, for each file, that could not be matched. Therefore, the following relations should hold (referring to lines Ⓔ and Ⓖ):

$$N_{FDF} = N_{\text{matches}} + N_{FDF\_unmatched}$$

$$N_{FDT} = N_{\text{matches}} + N_{FDT\_unmatched}$$

Ⓕ The format of this line is:

Number of unmatched FDT entries inside/outside of FDF time range:

$$N_{inside} / N_{outside}$$

The FDF and FDT files do not necessarily cover exactly the same time period. Out of the number of FDT entries that could not be matched to FDF entries ( $N_{\text{FDT\_unmatched}}$ , line Ⓖ), some of those might simply be because they did not fall within the timespan of the FDT events.  $N_{\text{inside}}$  and  $N_{\text{outside}}$  are counts to measure this overlap. Therefore, the following relation should hold (referring to lines Ⓖ and Ⓕ):

$$N_{\text{FDT\_unmatched}} = N_{\text{inside}} + N_{\text{outside}}$$

Ⓖ, Ⓗ, Ⓙ, Ⓚ Each of these blocks is focused on the statistics for one of the data fields. The format follows a basic template, which is elucidated in the description of Ⓖ, Ⓗ, Ⓙ, and Ⓚ. In this example for filetype “ground”, the reported data fields are TIME, LAT, LON, and ALT. Other filetypes will have different reported data fields (see Table 1). The distribution statistics are actually for the difference (or “delta”) between the FDF values and the FDT values for all matching pairs; hence, the variable names are labeled as “DELTA\_var”.

Ⓛ The format of the first line in each variable block is:

DELTA\_var [units] mean/min/max/stddev/tol:

*mean/minimum/maximum/standard\_deviation/tolerance*

For every match  $i$  that was found (1 to  $N_{\text{matches}}$ ),

$$\text{DELTA\_var}_i = \text{FDT\_var}_i - \text{FDF\_var}_i$$

This line of the report gives some statistical measures of the distribution of  $\text{DELTA\_var}_i$ . The mean, minimum, maximum, and standard deviation are self-explanatory. Each data field has an assigned tolerance value. These tolerances are arbitrary and were chosen by the IRIS Team as a rough indication of how much difference between the FDF and FDT values is acceptable; i.e., the following inequality relation indicates a good fit:

$$|\text{DELTA\_var}_i| \leq \text{TOL\_var}$$

These tolerances have no impact on any calculations, and are simply reported in conjunction with the  $\text{DELTA\_var}$  statistics for the convenience of the user.

Ⓜ The format of the second line in each variable block is:

DELTA\_var N\_match/N\_partial\_orbits/N\_errors:

$$N_{\text{matches}}/N_{\text{partial\_orbits}}/N_{\text{errors}}$$

$N_{\text{matches}}$  is described in line ⑨. Sometimes, an event’s time field will have a non-standard value, e.g., ‘Not in Pass’. We distinguish between two possibilities here. (a) If this occurs for the first entry of a file, it is presumed that the event spans the beginning of its file’s time range. Such an event is not used as a candidate for matching, but neither is it considered an error. That case is called a “partial orbit”. Since this can only occur for the first entry,  $N_{\text{partial\_orbits}}$  can only have a value of 0 or 1. (b) If an event, which isn’t the first event in the file, has a non-standard value for a time field, then this is considered an error and counts towards the value of  $N_{\text{errors}}$ . (For reasons of homogeneity and software refactoring, currently this information is repeated for every variable.)

- ⑬ The third line of each variable block addresses how often  $|\text{DELTA\_var}_i| \leq \text{TOL\_var}$ . There are two possible cases that apply, depending upon the data field. One case is described here and the other one is described in ⑬. Some of the data fields are used to generate the matches between the FDF and FDT events— which data fields are identified according to file type (see Table 1). Line ⑬ is an example for a variable that is used to make matches. (E.g., TIME is the only matching field for file type “ground”.) To make a match, it is necessary that all of the matching fields agree to within the field’s tolerance. Therefore, by definition,  $|\text{DELTA\_var}_i| \leq \text{TOL\_var}_i$  for all of the matches. In these cases, the third line of the variable block has the format:

**DELTA\_var was used to make matches, so by definition**  
**all DELTA\_var LE TOL\_var.**

- ⑭ Same explanation as ⑪.

- ⑮ Same explanation as ⑫.

- ⑯ The third line of each variable block addresses how often  $|\text{DELTA\_var}_i| \leq \text{TOL\_var}$ . There are two possible cases that apply, depending upon the data field. One case is described here and the other one is described in ⑬. Not all of the data fields are used to generate the matches between the FDF and FDT events. Line ⑬ is an example for a variable that is not used to make matches. (E.g., LAT, LON, and ALT are not matching fields for file type “ground”. See Table 1.) In this case, the third line of the variable block has the format:

**Number of DELTA\_var inside/outside TOL\_var:  $N_{\text{inside\_tol}}/N_{\text{outside\_tol}}$**   
 Since the field was not used to establish matches, it is entirely possible that some of the matches will show a large delta in this field.  $N_{\text{inside\_tol}}$  is the number of matches where  $|\text{DELTA\_var}_i| \leq \text{TOL\_var}$ , and  $N_{\text{outside\_tol}}$

is the number of matches where  $|\text{DELTA\_var}_i| > \text{TOL\_var}$ . Therefore, the following relation should hold:

$$N_{\text{matches}} = N_{\text{inside\_tol}} + N_{\text{outside\_tol}}$$

### 3.2.2 Report: Complex structure (file types “view\_XXdeg”)

We now examine the complex content structure of the text reports for file types “view\_10deg” and “view\_5deg” by looking in detail at an example. The FDF and FDT files contain information about each and every one of the station passes over some time period. The structure of those files is to internally have separate sections for each station/antenna, but then the event list for each station is organized according to a single template. Therefore, the text reports follow the same organization: there are separate subsections for each station, but the subsections are all structured the same way. Figures 4a and 4b display the comparison report for an FDF/FDT pair of files of type “view\_10deg”. The different lines of content have been labeled and distinguished by alternating colors. (The colors provide no information except to group like lines of content.) Much of the content structure is common to the basic format explained in § 3.2.1.

- Ⓜ This is the basic header of any report. These lines are explained in § 3.2.1 under Ⓐ, Ⓑ, and Ⓒ.
- Ⓝ Each station subsection of the report begins with a separator line of periods.
- Ⓞ The format of this line is:  
     Subsection for this station/antenna: *station*  
     where *station* is one of the station ID strings from Table 2.
- Ⓟ, Ⓠ, Ⓡ+Ⓢ+Ⓣ+Ⓤ, and Ⓡ These blocks provide data descriptions for separate stations, and all follow the same organization as lines Ⓔ–Ⓛ in § 3.2.1.
- Ⓢ It sometimes happens that the FDF file contains no pass information for a station. In this case, the string “NO FDF DATA” may appear instead of timecodes. See Ⓣ and Ⓤ for more consequences. (N.B. Sometimes the FDF file does not report pass information for the various Svalbard X-band antennae, with the implication that their pass information is identical to the information provided for the SG1S antenna. The software knows how to handle that situation, and will use the SG1S information for all Svalbard antennae. Therefore, that case does not cause a report of “NO FDF DATA”.)

```

=====
REPORT ON COMPARING FDT VERSUS FDF OF THIS TYPE: view_10deg
FDF filename:  BdP_pkg.20130320/fdf-in/FDF_IRIS.view_10deg.20130129.V01.txt
FDT filename:  BdP_pkg.20130320/fdt-out/IRIS.view_10deg.20130129.V01.txt
Variables saved filename:  iris_fd.varfile.view_10deg-FDF20130129-FDT20130129.sav
.....
Subsection for this station/antenna:  MGS
FDF range of start times:  2013-01-29T00:57:36.846 / 2013-02-04T22:57:29.351
FDT range of start times:  2013-01-29T00:57:37.000 / 2013-02-28T22:58:52.000
Number of FDF/FDT entries:  81 / 349
Number of matches / FDF unmatched / FDT unmatched:  79 / 2 / 270
Number of unmatched FDT entries inside/outside of FDF time range:  0 / 270
DELTA_ST_TIME [sec] mean/min/max/stdev/tol:  0.53/-0.11/1.49/0.33/60.00
DELTA_ST_TIME N_matches/N_partial_orbits/N_errors:  79/0/0
Number of DELTA_ST_TIME inside/outside TOL_ST_TIME:  79 / 0
DELTA_EN_TIME [sec] mean/min/max/stdev/tol:  -0.36/-0.94/0.39/0.35/60.00
DELTA_EN_TIME N_matches/N_partial_orbits/N_errors:  79/0/0
Number of DELTA_EN_TIME inside/outside TOL_EN_TIME:  79 / 0
DELTA_DURATION [sec] mean/min/max/stdev/tol:  -0.89/-1.69/0.07/0.44/120.00
DELTA_DURATION N_matches/N_partial_orbits/N_errors:  79/0/0
Number of DELTA_DURATION inside/outside TOL_DURATION:  79 / 0
DELTA_MAX_EL [deg] mean/min/max/stdev/tol:  0.01/-0.05/0.02/0.01/1.00
DELTA_MAX_EL N_matches/N_partial_orbits/N_errors:  79/0/0
Number of DELTA_MAX_EL inside/outside TOL_MAX_EL:  79 / 0
DELTA_MAX_EL_TIME [sec] mean/min/max/stdev/tol:  0.11/-0.47/1.06/0.36/60.00
DELTA_MAX_EL_TIME N_matches/N_partial_orbits/N_errors:  79/0/0
Number of DELTA_MAX_EL_TIME inside/outside TOL_MAX_EL_TIME:  79 / 0
.....
Subsection for this station/antenna:  WAPS
FDF range of start times:  2013-01-29T00:23:16.890 / 2013-02-04T23:58:54.076
FDT range of start times:  2013-01-29T00:23:17.000 / 2013-02-28T22:24:22.000
Number of FDF/FDT entries:  27 / 108
Number of matches / FDF unmatched / FDT unmatched:  25 / 2 / 83
Number of unmatched FDT entries inside/outside of FDF time range:  0 / 83
DELTA_ST_TIME [sec] mean/min/max/stdev/tol:  0.60/0.07/1.04/0.28/60.00
DELTA_ST_TIME N_matches/N_partial_orbits/N_errors:  25/0/0
Number of DELTA_ST_TIME inside/outside TOL_ST_TIME:  25 / 0
DELTA_EN_TIME [sec] mean/min/max/stdev/tol:  20.32/-0.88/413.00/83.09/60.00
DELTA_EN_TIME N_matches/N_partial_orbits/N_errors:  25/0/0
Number of DELTA_EN_TIME inside/outside TOL_EN_TIME:  24 / 1
DELTA_DURATION [sec] mean/min/max/stdev/tol:  19.72/-1.80/412.08/83.03/120.00
DELTA_DURATION N_matches/N_partial_orbits/N_errors:  25/0/0
Number of DELTA_DURATION inside/outside TOL_DURATION:  24 / 1

```

*Continued in Fig. 4b...*

Figure 4a: Example of complex text report structure for filetype “view\_10deg” (page 1 of 2). The letter labels refer to explanations in § 3.2.2.

...continued from Fig. 4a.

```

DELTA_MAX_EL [deg] mean/min/max/stdev/tol: 0.80/-0.01/20.06/4.01/1.00
DELTA_MAX_EL N_matches/N_partial_orbits/N_errors: 25/0/0
Number of DELTA_MAX_EL inside/outside TOL_MAX_EL: 24 / 1
DELTA_MAX_EL_TIME [sec] mean/min/max/stdev/tol: 6.95/-0.37/174.00/34.80/60.00
DELTA_MAX_EL_TIME N_matches/N_partial_orbits/N_errors: 25/0/0
Number of DELTA_MAX_EL_TIME inside/outside TOL_MAX_EL_TIME: 24 / 1
.....
Subsection for this station/antenna: ASF
FDF range of start times: NO FDF DATA.
FDT range of start times: 2013-01-29T03:31:53.000 / 2013-02-28T17:12:05.000
Number of FDF/FDT entries: 0 / 240
Number of matches / FDF unmatched / FDT unmatched: 0 / 0 / 0
Number of unmatched FDT entries inside/outside of FDF time range: 0 / 0
DELTA_ST_TIME [sec] mean/min/max/stdev/tol: --/--/--/--/--
DELTA_ST_TIME N_matches/N_partial_orbits/N_errors: 0/--/--
Number of DELTA_ST_TIME inside/outside TOL_ST_TIME: 0 / 0
DELTA_EN_TIME [sec] mean/min/max/stdev/tol: --/--/--/--/--
DELTA_EN_TIME N_matches/N_partial_orbits/N_errors: 0/--/--
Number of DELTA_EN_TIME inside/outside TOL_EN_TIME: 0 / 0
DELTA_DURATION [sec] mean/min/max/stdev/tol: --/--/--/--/--
DELTA_DURATION N_matches/N_partial_orbits/N_errors: 0/--/--
Number of DELTA_DURATION inside/outside TOL_DURATION: 0 / 0
DELTA_MAX_EL [deg] mean/min/max/stdev/tol: --/--/--/--/--
DELTA_MAX_EL N_matches/N_partial_orbits/N_errors: 0/--/--
Number of DELTA_MAX_EL inside/outside TOL_MAX_EL: 0 / 0
DELTA_MAX_EL_TIME [sec] mean/min/max/stdev/tol: --/--/--/--/--
DELTA_MAX_EL_TIME N_matches/N_partial_orbits/N_errors: 0/--/--
Number of DELTA_MAX_EL_TIME inside/outside TOL_MAX_EL_TIME: 0 / 0
.....
Subsection for this station/antenna: SG1S
FDF range of start times: 2013-01-29T00:08:09.039 / 2013-02-04T23:44:28.027
FDT range of start times: 2013-01-29T00:08:10.000 / 2013-02-28T23:45:40.000
... Et cetera...

```

Q2

R

S

T

U

V

Figure 4b: Example of complex text report structure for filetype “view\_10deg” (page 2 of 2). The letter labels refer to explanations in § 3.2.2.

- ① These lines are described in lines ⑤ and ⑥ of § 3.2.1. Note that  $N_{\text{FDF}} = 0$  in this example, consistent with line ⑤.
- ② These lines are described in lines ⑧–⑬ of § 3.2.1. In this example,  $N_{\text{FDF}} = 0$ , so therefore  $N_{\text{matches}} = 0$ , and therefore there are no matched pairs for which data statistics can be reported.

- Ⓥ The example is truncated in this block, but the text report would continue on from here in a similar fashion for the remainder of the stations. The complete text report ends with the final station block.

Table 2: List of downlink stations

| Station/Antenna Name | Station ID | Pass data may default to... |
|----------------------|------------|-----------------------------|
| McMurdo-10m          | MGS        | n/a                         |
| Wallops              | WAPS       | n/a                         |
| Alaska-11-3m         | ASF        | n/a                         |
| Svalbard-1S          | SG1S       | n/a                         |
| Svalbard-3X          | SG3X       | SG1S                        |
| Svalbard-6X          | SG6X       | SG1S                        |
| Svalbard-22X         | SG22X      | SG1S                        |

### 3.3 Reading the data *DELTA\_var* plots (screen)

The purpose of this program is to find matching pairs of events between the FDF and FDT files, and report on the differences (or “deltas”) for every data field. The user looks at the outputs of this software to decide whether the FDT results are adequately close to the FDF results. Given the amount of data, the reporting focuses on the distribution statistics of the deltas for each data variable (*DELTA\_var*). Each type of Flight Dynamics file has its own set of data fields (see Table 1).

One of the ways that this software reports on the distributions of the data fields is to plot them. These plots have a structure that follows a basic template that is explained in this subsection. The “view\_10deg” and “view\_5deg” types of files have an extra layer of complexity which multiplies the number of plotted data fields by the number of stations, so multiplotted windows are used for those cases. That extra complexity is also explained later in this subsection.

#### 3.3.1 Plots: Common structure (all file types)

We now examine the basic content structure of a *DELTA\_var* distribution plot by looking in detail at an example. Figure 5 shows the histogram of *DELTA\_LON* values for an FDF/FDT pair of files of type “ground”. Each graphics window that is opened is of size 800×600 pixels.

- Ⓐ The plots are displayed to IDL graphics windows. For all file types except “view\_10deg” and “view\_5deg” there is one plot per window. The IDL

window IDs start at #0 and go up from there as far as necessary. This example is in window #2, so it is the 3rd plot for the “ground” file type.

- ③ The plot title announces the file type and the data field being plotted.
- ④ The x-axis title reidentifies the plotted variable, and gives a reminder that the order of the difference goes like this:

$$\text{DELTA\_var}_i = \text{var\_FDT}_i - \text{var\_FDF}_i$$

over all matches  $i$ . That is because the FDT results are being compared against the presumed-correct FDF results.

- ⑤ The distribution of  $\text{DELTA\_var}_i$  are plotted in histogram-style. Since it is the *difference* of the FDF and FDT results that is being plotted, a value of  $\text{DELTA\_var}_i = 0$  always corresponds to a perfect match. The histogram binsize that is applied varies by data field and file type. In this example, the binsize is 1 degree. However, it is always true that the binning is centered such that the bin containing zero is centered on zero. In this example, therefore, the “zero” bin goes from -0.5 degrees to +0.5 degrees, and that bin contains the number of matches such that  $-0.5 \leq \text{DELTA\_LON}_i < 0.5$ .
- ⑥ Each data field has an assigned tolerance value, and the width of the tolerance window is indicated on the plot by a pair of vertical dotted lines. These tolerances are arbitrary and were chosen by the IRIS Team as a rough indication of how much difference between the FDF and FDT values is acceptable; i.e., the following inequality relation indicates a good fit:

$$|\text{DELTA\_var}_i| \leq \text{TOL\_var}$$

These tolerances have no impact on any calculations, and are simply provided as a suggestion for how good of a fit the distribution is indicating. In this example, we see that  $\text{TOL\_LON} = 1$  degree, and the distribution of  $\text{DELTA\_LON}_i$  has a sizable tail on the positive side. I.e., the FDT results have a significant systematic bias to being too high.

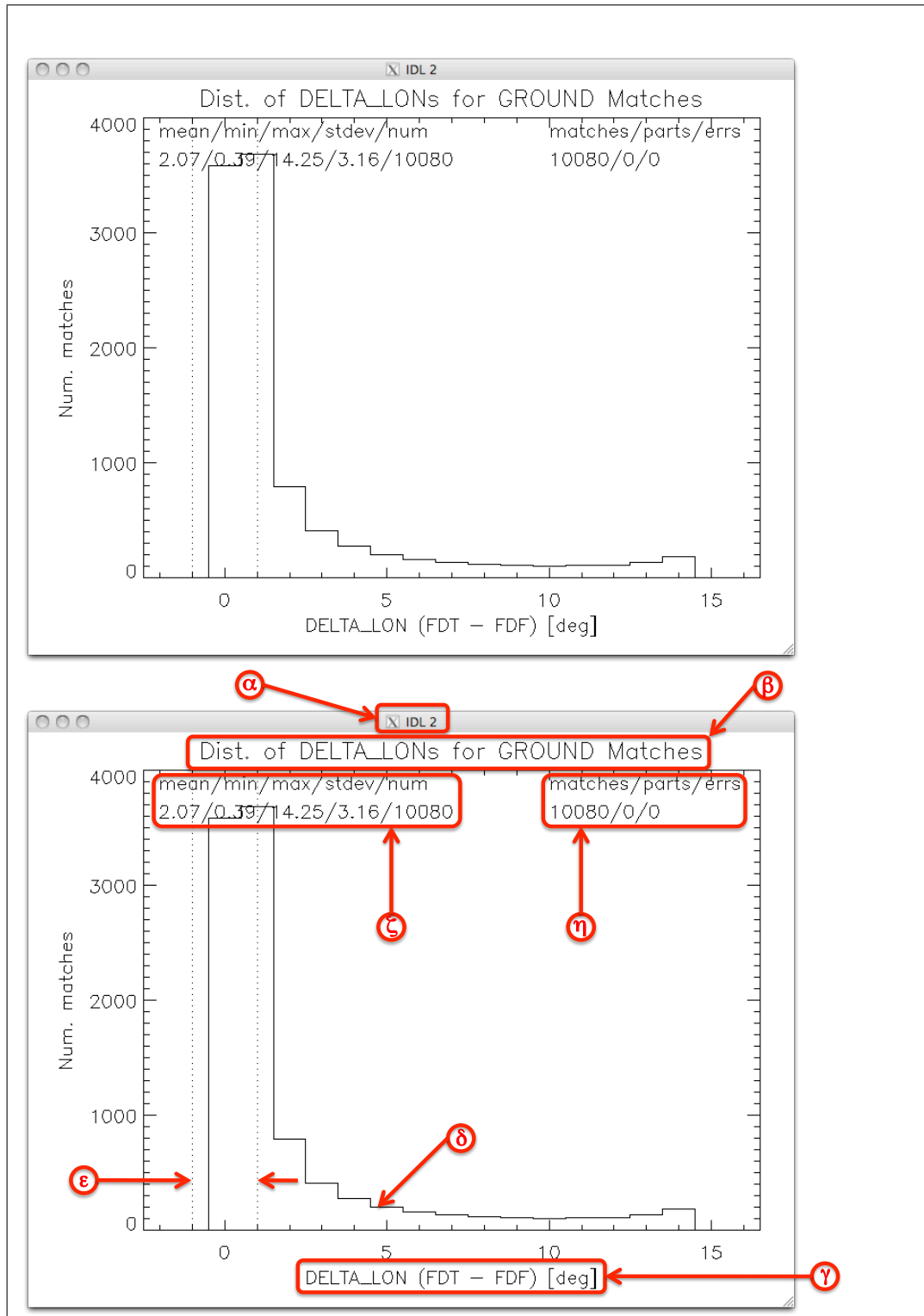


Figure 5: Example of basic data plot structure for the DELTA\_LON data field for filetype “ground”, with labels. The labels refer to explanations in § 3.3.1.

- Ⓒ This text block provides some statistical measures for the distribution. The mean, minimum, maximum, and standard deviation are all self-explanatory. “Num” is simply the total number of matches in the distribution. These values are also to be found in the text report. (See the explanation of lines Ⓖ and Ⓓ in § 3.2.1 about the text report.) In this example, the standard deviation is 3.16 degrees, which is greater than `TOL_LON` = 1 degree (see Ⓔ), thus indicating that the FDT results do not match the FDF results well. In addition, the mean of 2.07 degrees is consistent with the observation that the distribution has a systemic positive bias.
- Ⓙ This text block indicates whether any of the timecode-type data fields have non-standard values. “Matches” refers to the total number of matches; “parts” refers to the number of partial orbit events; and “errs” refers to the number of just plain errors. These values are also to be found in the text report. (See the explanation of line Ⓕ in § 3.2.1 about the text report for more detail.)

### 3.3.2 Plots: Complex structure (file types “view\_XXdeg”)

We now briefly examine the multiplot organization for file types “view\_10deg” and “view\_5deg” by looking at an example. Those types of FDF and FDT files contain information about each and every one of the station passes over some time period. The internal structure of those files is to have separate sections for each station/antenna. Furthermore, there are several data fields to be plotted for these file types (see Table 1). The number of fields times the number of stations is too many to give each plot its own IDL window, so IDL multiplotting is used to group all of the data fields for one station together. Each multiplot window is of size 2400×1200 pixels. **(Warning: The multiplot windows for file types “view\_10deg” and “view\_5deg” are too large for a laptop display and for some workstation monitors.)**

Note that all of the Svalbard stations/antennae are permitted to default their pass information to the data for SG1S (Table 2), so it may happen that the plots for all of the Svalbard stations/antennae are identical.

Figure 6 shows one of the multiplot windows that are opened when the input files are of type “view\_10deg”. The window shown is for the McMurdo (MGS) station. All plots have the same structure as explained in § 3.3.1 and illustrated in Figure 5. The only addition is that the station ID is provided in the plot titles.

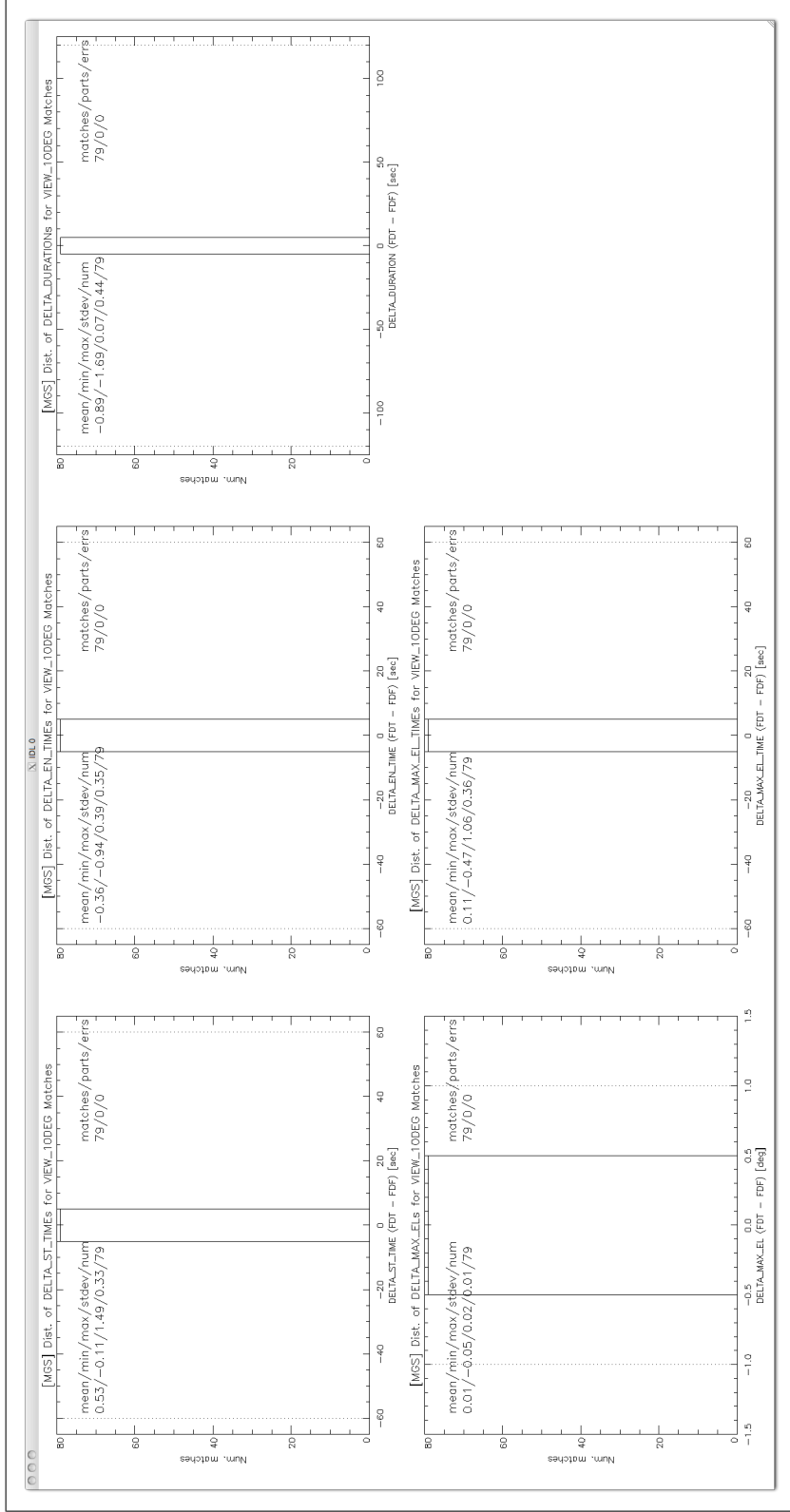


Figure 6: Example of multiplot structure for all of the data fields for station MGS for filetype "view\_10deg". Discussion in § 3.3.2.

### 3.4 Reading the saved data variables (*varfile*)

The main program variables are saved to an IDL savefile, with a filename in this format:

**iris\_fd\_varfile\_type\_FDFyyyymmdd\_FDTyyyymmdd.sav,**

where *type* is one of the file types from Table 1, and *yyyymmdd* are the date codes from the input FDF and FDT filenames. Note that (a) the format of the filename is fixed; (b) a new savefile is written for every run of `iris_fd_compare_files.pro`; and therefore (c) each savefile contains variables for only a specific pair of FDF/FDT files of one type. The filename format is designed to identify the exact files and filetype.

Table 3 lists the variables that are contained in the save file, their basic data type, what information they contain, and where each one is described in this Manual.

Table 3: Contents of IDL savefile (*varfile*)

| Variable Name        | Data Type       | Describes...                                                            | Description in section... |
|----------------------|-----------------|-------------------------------------------------------------------------|---------------------------|
| <b>FDF_file</b>      | string scalar   | name of FDF input file                                                  | § 3.4.1                   |
| <b>FDT_file</b>      | string scalar   | name of FDT input file                                                  | § 3.4.1                   |
| <b>reportfile</b>    | string scalar   | name of text report file                                                | § 3.4.1                   |
| <b>varfile</b>       | string scalar   | name of IDL savefile                                                    | § 3.4.1                   |
| <b>FDF_parsed</b>    | structure array | list of event data in <b>FDF_file</b>                                   | § 3.4.2                   |
| <b>FDT_parsed</b>    | structure array | list of event data in <b>FDT_file</b>                                   | § 3.4.2                   |
| <b>matches</b>       | structure array | list of matched event data from <b>FDF_parsed</b> and <b>FDT_parsed</b> | § 3.4.3                   |
| <b>FDF_unmatched</b> | structure array | list of unmatched event data from <b>FDF_parsed</b>                     | § 3.4.3                   |
| <b>FDT_unmatched</b> | structure array | list of unmatched event data from <b>FDT_parsed</b>                     | § 3.4.3                   |

As Table 3 suggests, there are likenesses among these variables. The filename strings are readily understandable. The **FD\*\_parsed** variables are the raw event data from the input files, immediately after they have been parsed from the ASCII files into IDL variables, and *before* any attempt at matching has been made. The three **\*match\*** variables are the recollected event data *after* they have gone through the matching process, and so the pairing associations (or lack of such) are made clear.

### 3.4.1 Description of saved filenames (`FDF_file`, `FDT_file`, `reportfile`, and `varfile`)

There are four filenames relevant to running the FD Comparison software. (Five if you count the program, `iris_fd_compare_files.pro`.)

**FDF\_file** This variable is a string scalar containing the name of the single FDF file that was used for the comparison. The filename is supplied by the user as an input parameter of the same name when the routine `iris_fd_compare_files.pro` is called. This filename is not only saved in the IDL savefile, but is also recorded in the text report (see § 3.2.1, lines ©). “FDF” is short for Flight Dynamics Facility (i.e., a team at GSFC/NASA), and in this Manual is used to refer to the data products provided by the MOC.

**FDT\_file** This variable is a string scalar containing the name of the single FDT file that was used for the comparison. The filename is supplied by the user as an input parameter of the same name when the routine `iris_fd_compare_files.pro` is called. This filename is not only saved in the IDL savefile, but is also recorded in the text report (see § 3.2.1, lines ©). “FDT” is short for Flight Dynamics Team (i.e., members of the IRIS Team), and in this Manual is used to refer to the data products provided by the IRIS Team. The FD Comparison software was created to help with the validation of the FDT products against the FDF products.

**reportfile** This variable is a string scalar containing the name of a single ASCII text file to which the report is written. The user can optionally specify this filename as an input keyword of the same name when the routine `iris_fd_compare_files.pro` is called. If a name is not supplied, then `reportfile = 'iris_fd_report_file.txt'` by default. At the point that `reportfile` is saved to the IDL savefile, its final value has been resolved. The report text is always appended to the file if it pre-exists; there are two consequences of this. (1) There is no danger of overwriting previous reports in that file. (2) It is possible for the report file to contain multiple reports, for any number of FDF/FDT file pairs and filetypes. So even though the IDL savefile only contains variables regarding one particular comparison, the associated report block may be contained inside the same report file as other report blocks.

**varfile** This variable is a string scalar containing the name of the IDL savefile which contains it. (Recursion, recursion!) This can be useful to identify the last savefile that was restored. The user can also optionally

return this filename from `iris_fd_compare_files.pro` with a keyword of the same name. This provides a simple way for the user to identify the savefile immediately after running the routine. In addition, this filename is also recorded in the text report (see § 3.2.1, line ④). The format of the filename is tightly controlled; more information about the format can be found at the top of § 3.4.

### 3.4.2 Description of parsed data (FDF\_parsed and FDT\_parsed)

The FDF and FDT data product files (of various types) contain several data fields (of various types). Before any matching and comparing can be attempted, the list of event data in the ASCII files must be parsed and read into IDL variables. `FDF_parsed` and `FDT_parsed` contain the event data from this point in the processing. As the files contain a list of time-ordered events with multiple data fields, it is natural to transfer the information into a 1D array of structures, where each array member is a structure with tags that map to the data fields. The size of the array is equal to the number of events, is equal to the number of event lines in the corresponding file. **(N.B. If the number of parsed events is zero, then the variable is an integer with a value of -1.)**

Tables 4–10 describe the organization and contents of both `FDF_parsed` and `FDT_parsed`. There are some data fields which appear in one file and are not relevant to the comparison, so these are not included in the tables. There are also some instances where the parser assigns slightly different names to the same data field. In addition, there is an extra layer of complexity for filetypes “view\_10deg” and “view\_5deg”. As discussed in § 3.2.2 and § 3.3.2, these files contain multiple sets of event lists for all of the stations (Table 2). The parser preserves this organization in `FDF_parsed` and `FDT_parsed` with a middle structure layer of station names.

For example, the “ground” version looks like this:

```
IDL> help, FDF_parsed
FDF_PARSED      STRUCT    = -> <Anonymous> Array[10080]
IDL> help, FDF_parsed[0], /st
** Structure <7873478>, 4 tags, length=40, data length=40, refs=2:
TIME            STRING    '2013-01-29T00:00:31.360'
LAT             DOUBLE     41.764000
LON             DOUBLE     83.522000
ALT             DOUBLE     654.41154
```

`FDF_parsed` is an array of length 10080 because there are that many events, and its tags are the data fields.

The “view\_10deg” version looks like this:

```
IDL> help, FDF_parsed
FDF_PARSED          STRUCT    = -> <Anonymous> Array[1]
IDL> help, FDF_parsed[0], /st
** Structure <428c3b8>, 3 tags, length=13464, data length=11968, refs=2:
MCMURDO_10DEG       STRUCT    -> <Anonymous> Array[81]
SG1S_10DEG          STRUCT    -> <Anonymous> Array[79]
WAPS_10DEG           STRUCT    -> <Anonymous> Array[27]
IDL> help, FDF_parsed[0].MCMURDO_10DEG[0], /st
** Structure <428c028>, 6 tags, length=72, data length=64, refs=2:
PASS                LONG          103
START_TIME          STRING        '2013-01-29T00:57:36.846'
END_TIME            STRING        '2013-01-29T01:06:34.001'
DURATION            DOUBLE         537.15600
MAX_ELEVATION        FLOAT         51.4690
MAX_ELEVATION_TIME  STRING        '2013-01-29T01:02:04.025'
```

FDF\_parsed is an array of length 1, and there is a middle layer of structure tags corresponding to the stations. In this case, the *station layer* is a set of 1D arrays with length equal to the number of events. So in this example there are 81 events at McMurdo, 79 events at Svalbard, and 27 events at Wallops. And then it is the tag set of the station structures that corresponds to the data fields for this filetype.

Table 4: Contents of FD\*\_parsed (“aaz”)

| 1st line tags | 2nd line tags | Units/Format | Description                               |
|---------------|---------------|--------------|-------------------------------------------|
| AAZ_DIRECTION | n/a           | string       | AAZ Angle is 'Increasing' or 'Decreasing' |
| AAZ_TIME      |               | time string  | AAZ Crossing Time                         |
| AAZ_ANGLE     |               | degrees      | AAZ Crossing Angle                        |

Table 5: Contents of FD\*\_parsed (“eclipse”)

| 1st line tags     | 2nd line tags | Units/Format | Description                  |
|-------------------|---------------|--------------|------------------------------|
| START_TIME        | n/a           | time string  | Eclipse segment start time   |
| STOP_TIME         |               | time string  | Eclipse segment stop time    |
| DURATION          |               | sec          | Duration of eclipse segment  |
| OBSTRUCTION       |               | string       | Object (e.g., 'Earth')       |
| START_PASS        |               | integer      | Pass ID at start time        |
| STOP_PASS         |               | integer      | Pass ID at stop time         |
| CURRENT_CONDITION |               | string       | 'Penumbra' or 'Umbra'        |
| WORST_CONDITION   |               | string       | 'Penumbra' or 'Umbra'        |
| TOTAL_DURATION    |               | sec          | Of eclipse over all segments |

Table 6: Contents of FD\*\_parsed (“events”)

| 1st line tags    | 2nd line tags | Units/Format | Description               |
|------------------|---------------|--------------|---------------------------|
| PASS             | n/a           | integer      | Pass ID                   |
| START_TIME       |               | time string  | Orbit start time          |
| APOGEE           |               | km           | Apogee altitude           |
| APOGEE_TIME      |               | time string  | Apogee time               |
| PERIOD           |               | sec          | Orbital period            |
| LON_ASCEN_NODE   |               | degrees      | Long ascension node       |
| ASCEN_NODE_TIME  |               | time string  | Long ascension node time  |
| END_TIME         |               | time string  | Orbit end time            |
| PERIGEE          |               | km           | Perigee altitude          |
| PERIGEE_TIME     |               | time string  | Perigee time              |
| RIGHT_ASCEN      |               | degrees      | Right ascension           |
| LON_DESCEN_NODE  |               | degrees      | Long descension node      |
| DESCEN_NODE_TIME |               | time string  | Long descension node time |

Table 7: Contents of FD\*\_parsed (“ground”)

| 1st line tags | 2nd line tags | Units/Format | Description             |
|---------------|---------------|--------------|-------------------------|
| TIME          | n/a           | time string  | Time s/c is at location |
| LAT           |               | degrees      | Latitude                |
| LON           |               | degrees      | Longitude               |
| ALT           |               | km           | Altitude                |

Table 8: Contents of FD\*\_parsed (“radzones”)

| 1st line tags | 2nd line tags | Units/Format | Description              |
|---------------|---------------|--------------|--------------------------|
| START_TIME    | n/a           | time string  | Zone enter time          |
| STOP_TIME     |               | time string  | Zone exit time           |
| DURATION      |               | sec          | Duration of zone passage |

Table 9: Contents of FD\*\_parsed (“view\_10deg”)

| 1st line tags                       | 2nd line tags      | Units/Format | Description             |
|-------------------------------------|--------------------|--------------|-------------------------|
| MCMURDO_10DEG<br>(or MGS_10M_10DEG) | PASS               | integer      | Pass ID                 |
| WAPS_10DEG                          | START_TIME         | time string  | Visibility start time   |
| ASF_11.3M_10DEG                     | END_TIME           | time string  | Visibility stop time    |
| SG1S_10DEG                          | DURATION           | sec          | Duration of visibility  |
| SG3X_10DEG                          | MAX_ELEVATION      | degrees      | Max. angle of elevation |
| SG6X_10DEG                          | MAX_ELEVATION_TIME | time string  | Time of max. elevation  |
| SG22X_10DEG                         |                    |              |                         |

Table 10: Contents of FD\*\_parsed (“view\_5deg”)

| 1st line tags           | 2nd line tags      | Units/Format | Description             |
|-------------------------|--------------------|--------------|-------------------------|
| MCMURDO<br>(or MGS_10M) | PASS               | integer      | Pass ID                 |
| WAPS                    | START_TIME         | time string  | Visibility start time   |
| ASF_11.3M               | END_TIME           | time string  | Visibility stop time    |
| SG1S                    | DURATION           | sec          | Duration of visibility  |
| SG3X                    | MAX_ELEVATION      | degrees      | Max. angle of elevation |
| SG6X                    | MAX_ELEVATION_TIME | time string  | Time of max. elevation  |
| SG22X                   |                    |              |                         |

### 3.4.3 Description of post-matching variables (`matches`, `FDF_unmatched`, and `FDT_unmatched`)

The variable `matches` is a recollection of some of the event data according to matched FDF/FDT pairs. Sometimes, not all of the FDF or FDT events are associated into a pair by the matching algorithm, and these leftovers are stored in `FDF_unmatched` and `FDT_unmatched`. As you are about to see, these variables are also structure arrays and are organized in a manner very similar to `FDF_parsed` and `FDT_parsed`. Important differences are: (a) the post-matching variables have an organization that accomodates data for one FDF event and one FDT event; and (b) some of the tag names are changed for more consistency.

We now examine the basic organizational structure of the post-matching variables (`matches`, `FDF_unmatched`, and `FDT_unmatched`). Figure 7 displays an example of the `matches` variable for an FDF/FDT pair of files of filetype “ground”. The variables `FDF_unmatched` and `FDT_unmatched` also have this same structure, even though they only contain data for one file of events. The different lines of content have been labeled and distinguished by alternating colors. (The colors provide no information except to group like lines of content.) This basic structure applies to filetypes “aaz”, “eclipse”, “events”, “ground”, and “radzones”.

- Ⓒ For the basic structure, the variable is a 1D array with length equal to the number of matches that were found (`matches`) or equal to the number of unmatched events (`FDF_unmatched`, `FDT_unmatched`). In this example, 10080 matches were found.
- Ⓓ The following filetypes can have this basic structure: “aaz”, “eclipse”, “events”, “ground”, and “radzones”.
- Ⓔ The FDF and FDT input filenames, from which all of the event data originated, are recorded.
- Ⓕ Each matched pair is comprised of one FDF event from the variable `FDF_parsed` and one FDT event from `FDT_parsed`. The tags `FDF_INDEX` and `FDT_INDEX` are addresses to `FDF_parsed` and `FDT_parsed`, respectively. This provides a convenient way to map a matched pair back to the original event lists (although that back-mapping could also be done by searching over all of the data fields). Note that the FDF and FDT files have header lines, so `FDF_INDEX` and `FDT_INDEX` are not the line numbers of the events inside those files.
- Ⓖ, Ⓗ, Ⓖ, and Ⓒ Each of these blocks is focused on one of the data fields, `varN`, and there are four tags associated with every data field. In this

|                                                                       |        |                                    |  |  |    |
|-----------------------------------------------------------------------|--------|------------------------------------|--|--|----|
| IDL> help, matches                                                    |        |                                    |  |  | Ⓐ  |
| MATCHES STRUCT = -> GROUND_MATCH_TYPE Array[10080]                    |        |                                    |  |  |    |
| IDL> help, matches, /st                                               |        |                                    |  |  |    |
| ** Structure GROUND_MATCH_TYPE, 21 tags, length=200, data length=200: |        |                                    |  |  |    |
| FILE_TYPE                                                             | STRING | 'ground'                           |  |  | Ⓘ  |
| FDF_FILENAME                                                          | STRING | 'FDF_IRIS_ground.20130129.V01.txt' |  |  | Ⓚ  |
| FDT_FILENAME                                                          | STRING | 'IRIS_ground.20130129.V01.txt'     |  |  |    |
| FDF_INDEX                                                             | LONG   | 0                                  |  |  | Ⓐ  |
| FDT_INDEX                                                             | LONG   | 0                                  |  |  |    |
| FDF_TIME                                                              | STRING | '2013-01-29T00:00:31.360'          |  |  | μ1 |
| FDT_TIME                                                              | STRING | '2013-01-29T00:00:00.000'          |  |  | μ2 |
| DELTA_TIME                                                            | DOUBLE | -31.360000                         |  |  | μ3 |
| TOL_TIME                                                              | DOUBLE | 60.000000                          |  |  | μ4 |
| FDF_LAT                                                               | DOUBLE | 41.764000                          |  |  |    |
| FDT_LAT                                                               | DOUBLE | 39.861000                          |  |  | Ⓥ  |
| DELTA_LAT                                                             | DOUBLE | -1.9030000                         |  |  |    |
| TOL_LAT                                                               | DOUBLE | 1.0000000                          |  |  |    |
| FDF_LON                                                               | DOUBLE | 83.522000                          |  |  |    |
| FDT_LON                                                               | DOUBLE | 84.116000                          |  |  | Ⓢ  |
| DELTA_LON                                                             | DOUBLE | 0.59400000                         |  |  |    |
| TOL_LON                                                               | DOUBLE | 1.0000000                          |  |  |    |
| FDF_ALT                                                               | DOUBLE | 654.41154                          |  |  |    |
| FDT_ALT                                                               | DOUBLE | 654.57482                          |  |  | Ⓞ  |
| DELTA_ALT                                                             | DOUBLE | 0.16328290                         |  |  |    |
| TOL_ALT                                                               | DOUBLE | 50.000000                          |  |  |    |

Figure 7: Example of basic structure for post-matched variables (`matches`, `FDF_unmatched`, and `FDT_unmatched`), with labels. The labels refer to explanations in § 3.4.3.

example for filetype “ground”, the data fields are TIME, LAT, LON, and ALT. Other filetypes will have different reported data fields (see Table 1).

Ⓘ and Ⓚ The values of the data field from the FDF and FDT versions of this event are recorded here, with tag names in the form of `FDF_varN` and `FDT_varN`. The user is interested in how closely the FDT results recreate the FDF results.

Ⓢ DELTA\_varN is calculated as:

$$\text{DELTA\_varN} = \text{FDT\_varN} - \text{FDF\_varN}$$

(For time strings, the delta is in units of seconds.) The size of `DELTA_varN` is a measure of the error in the FDT results relative to the FDF results. The text report and plots provide information on the distributions of these data deltas.

Ⓞ Each data field has an assigned tolerance value. These tolerances are arbitrary and were chosen by the IRIS Team as a rough indication of

how much difference between the FDF and FDT values is acceptable; i.e., the following inequality relation indicates a good fit:

$$|\text{DELTA\_varN}| \leq \text{TOL\_varN}$$

These tolerances have no impact on any calculations; and are initialized inside the FD Comparison software. The values for all `TOL_varN` are in the text report. Also, the tolerance window is shown on the plots as two vertical dotted lines. The report and plots provide the standard deviation of the distribution of `DELTA_varNi` values across all matches *i*, which is useful for comparing in the aggregate against `TOL_varN`.

Next, we briefly look at the extra complexity of the post-matching variables for the “view\_10deg” and “view\_5deg” filetypes. Those types of FDF and FDT files contain information about each and every one of the station passes over some time period. The internal structure of those files is to have separate sections for each station/antenna, with the same data fields being replicated for each station. The post-matching variables follow the same organization.

Note that all of the Svalbard stations/antennae are permitted to default their pass information to the data for SG1S (Table 2), so it may happen that the FDT data for all of the Svalbard stations/antennae are identical.

Figure 8 displays an example of the `FDF_unmatched` variable for an FDF file of filetype “view\_10deg”. The variables `matches` and `FDT_unmatched` also have this same structure. Note that all of the tags referring to FDT data have “empty” values. The different lines of content have been labeled and distinguished by alternating colors. (The colors provide no information except to group like lines of content.) This complex structure applies to filetypes “view\_10deg” and “view\_5deg”.

- Ⓟ For the complex structure, the top-level of the variable is a one-element array.
- Ⓟ These tags contain information relevant to the comparison of the entire FDF and FDT files, i.e., they are not specific to one station. These lines are explained for the basic example under Ⓛ and Ⓚ.
- Ⓞ The first layer of tags groups the events by station. Each of these station tags is a 1D array with length equal to the number of matches that were found (`matches`) or equal to the number of unmatched events (`FDF_unmatched`, `FDT_unmatched`). In this example, there were 2 left-over, unmatched FDF events for the MGS, WAPS, SG1S, SG3X, and SG22X stations, and zero unmatched events for the ASF and SG6X stations.

|                                                                           |        |      |                                        |   |
|---------------------------------------------------------------------------|--------|------|----------------------------------------|---|
| IDL> help, FDF_unmatched                                                  |        |      |                                        | Ⓟ |
| FDF_UNMATCHED                                                             | STRUCT | = -> | <Anonymous> Array[1]                   |   |
| IDL> help, FDF_unmatched[0], /st                                          |        |      |                                        |   |
| ** Structure <309ae08>, 10 tags, length=3184, data length=3146, refs=2:   |        |      |                                        |   |
| FILE_TYPE                                                                 | STRING |      | 'view_10deg'                           | Ⓟ |
| FDF_FILENAME                                                              | STRING |      | 'FDF_IRIS_view_10deg_20130129.V01.txt' | Ⓟ |
| FDT_FILENAME                                                              | STRING |      | 'IRIS_view_10deg_20130129.V01.txt'     | Ⓟ |
| MGS                                                                       | STRUCT | ->   | VIEW_10DEG_MATCH_TYPE Array[2]         |   |
| WAPS                                                                      | STRUCT | ->   | VIEW_10DEG_MATCH_TYPE Array[2]         |   |
| ASF                                                                       | LONG   |      | -1                                     | Ⓢ |
| SG1S                                                                      | STRUCT | ->   | VIEW_10DEG_MATCH_TYPE Array[2]         |   |
| SG3X                                                                      | STRUCT | ->   | VIEW_10DEG_MATCH_TYPE Array[2]         |   |
| SG6X                                                                      | LONG   |      | -1                                     |   |
| SG22X                                                                     | STRUCT | ->   | VIEW_10DEG_MATCH_TYPE Array[2]         |   |
| IDL> help, FDF_unmatched[0].MGS, /st                                      |        |      |                                        |   |
| ** Structure VIEW_10DEG_MATCH_TYPE, 31 tags, length=312, data length=309: |        |      |                                        |   |
| FILE_TYPE                                                                 | STRING |      | 'view_10deg'                           | Ⓢ |
| FDF_FILENAME                                                              | STRING |      | 'FDF_IRIS_view_10deg_20130129.V01.txt' |   |
| FDT_FILENAME                                                              | STRING |      | 'IRIS_view_10deg_20130129.V01.txt'     |   |
| FDF_STATION                                                               | STRING |      | 'MGS_10m_10deg'                        | Ⓢ |
| FDT_STATION                                                               | STRING |      | 'MGS_10m_10deg'                        | Ⓢ |
| FDF_INDEX                                                                 | LONG   |      | 6                                      | Ⓢ |
| FDT_INDEX                                                                 | LONG   |      | -1                                     | Ⓢ |
| FDF_PASS                                                                  | LONG   |      | 112                                    |   |
| FDT_PASS                                                                  | LONG   |      | 0                                      | Ⓢ |
| DELTA_PASS                                                                | LONG   |      | 0                                      |   |
| TOL_PASS                                                                  | BYTE   |      | 0                                      |   |
| FDF_ST_TIME                                                               | STRING |      | '2013-01-29T15:42:17.675'              |   |
| FDT_ST_TIME                                                               | STRING |      | ' '                                    | Ⓢ |
| DELTA_ST_TIME                                                             | DOUBLE |      | 0.0000000                              |   |
| TOL_ST_TIME                                                               | DOUBLE |      | 60.000000                              |   |
| FDF_EN_TIME                                                               | STRING |      | '2013-01-29T15:44:06.149'              |   |
| FDT_EN_TIME                                                               | STRING |      | ' '                                    | Ⓢ |
| DELTA_EN_TIME                                                             | DOUBLE |      | 0.0000000                              |   |
| TOL_EN_TIME                                                               | DOUBLE |      | 60.000000                              |   |
| FDF_DURATION                                                              | DOUBLE |      | 108.47400                              | Ⓢ |
| FDT_DURATION                                                              | DOUBLE |      | 0.0000000                              |   |
| DELTA_DURATION                                                            | DOUBLE |      | 0.0000000                              |   |
| TOL_DURATION                                                              | DOUBLE |      | 120.00000                              |   |
| FDF_MAX_EL                                                                | DOUBLE |      | 10.468000                              |   |
| FDT_MAX_EL                                                                | DOUBLE |      | 0.0000000                              | Ⓢ |
| DELTA_MAX_EL                                                              | DOUBLE |      | 0.0000000                              |   |
| TOL_MAX_EL                                                                | DOUBLE |      | 1.0000000                              |   |
| FDF_MAX_EL_TIME                                                           | STRING |      | '2013-01-29T15:43:11.895'              |   |
| FDT_MAX_EL_TIME                                                           | STRING |      | ' '                                    | Ⓢ |
| DELTA_MAX_EL_TIME                                                         | DOUBLE |      | 0.0000000                              |   |
| TOL_MAX_EL_TIME                                                           | DOUBLE |      | 60.000000                              |   |

Figure 8: Example of complex structure for post-matched variables (`matches`, `FDF_unmatched`, and `FDT_unmatched`), with labels. The labels refer to explanations in § 3.4.3.

Ⓢ This information is repeated from the the upper level in block Ⓟ, simply

for consistency with the basic template. These lines are explained for the basic example under  $\textcircled{\iota}$  and  $\textcircled{\kappa}$ .

- $\textcircled{\tau}$  For thoroughness, the station identity of the matched events is recorded as a data field. Note that the string indicates also whether this was a 10-degree pass or a 5-degree pass. This information is redundant, but it is convenient to have this string readily available.
- $\textcircled{\upsilon}$  These lines are explained for the basic example under  $\textcircled{\lambda}$ . However, note that `FDF_parsed` and `FDT_parsed` also have a middle layer of station tags, which are the arrays over events, so these indices are applicable at that level.
- $\textcircled{\phi}$ ,  $\textcircled{\chi}$ ,  $\textcircled{\psi}$ ,  $\textcircled{\omega}$ ,  $\textcircled{\vartheta}$ , and  $\textcircled{\mathbb{I}}$  The template for these blocks is explained for the basic example under  $\textcircled{\mu 1}$ ,  $\textcircled{\mu 2}$ ,  $\textcircled{\mu 3}$ , and  $\textcircled{\mu 4}$ .