Yohkoh Analysis Guide
User Guide

Ver 1.06 Ed: M.D. Morrison 6-Nov-92
Ver 2.00 Ed: M.D. Morrison 25-Jan-94
Ver 2.9x Ed: R.D. Bentley – Date below

This Copy Produced: March 31, 1999

Prepared at
Mullard Space Science Laboratory
University College London

Contents

List of Figures

1 Introduction to the Analysis of Yohkoh Data

This chapter is meant as an introduction, explaining basic terms and concepts about the Yohkoh mission and data. A newcomer to Yohkoh should read this before trying to use the rest of the Yohkoh Analysis Guide (**YAG**). The **YAG** is a comprehensive guide describing the data and the software; it is self-contained document that explains how to access the data, carry out analysis tasks and even install the software. The **YAG** may be fairly intimidating at first glance because of its bulk, but the reader should note that the bulk results mostly from the great breadth of the software, and that the basic principles are quite straightforward.

1.1 Yohkoh Instruments and Spacecraft

Yohkoh ("Sunbeam" in Japanese) is a satellite dedicated to high-energy observations of the Sun, specifically of flares and other coronal disturbances. The Yohkoh mission was launched on August 30, 1991, from the Kagoshima Space Centre in southern Japan. The spacecraft carries a payload of four scientific instruments: the Soft X-ray Telescope (SXT), the Hard X-ray Telescope (HXT), the Bragg Crystal Spectrometer (BCS) and the Wide Band Spectrometer (WBS). The SXT (which is sensitive in the range 1-2 KeV) takes images in various wavebands (selected by filters) using a CCD - either the full CCD frame, or a selected part of the CCD frame is returned in telemetry - these are known as full frame, and partial frame images (FFI and PFI); the HXT (which is sensitive in the range 10-100 KeV) measures Fourier components in 4 channels through a set of 64 pairs of grids - the images are reconstructed on the ground; the BCS observes the line complexes of Fe XXVI, Fe XXV, Ca XIX and S XV using bent germanium crystals; and the WBS observes the overall energy release between soft X-rays and gamma-rays using three separate instrument packages. Together these instruments provide the most detailed record yet obtained of high-energy processes in solar flares and in other forms of solar coronal activity.

The spacecraft is in a slightly elliptical, low-Earth orbit (approx. 570 x 730 km), with an orbital period of $\sim$ 96 minutes. As the orbit precesses, the time the spacecraft spends in sunlight on each revolution varies between 65 and 75 minutes. During 5 or 6 orbits each day, the spacecraft passes through the radiation belts of the South Atlantic Anomaly; at these times, all detectors that use high voltages are turned off (i.e. all BCS and HXT channels, and most WBS channel; the SXT is not affected).

Observations made by the instruments are stored in the spacecraft Bubble Data Recorder (BDR). The BDR has a capacity of 10 Mbytes, and in order to optimise the use of the recorder, the spacecraft can operate at several bit-rates: high, medium and low; switching between these rates is partly automatic (as a function of the spacecraft mode) and is partly controlled by on-board deferred commands. In high bit-rate, the BDR can only hold 42 minutes of observations and some switching of bit-rates is therefore necessary; some overwriting of the data in the BDR is allowed, but the rules for this are beyond the scope of this document.

There are a large number of spacecraft modes, including several subsystem specific modes (e.g. ACS mode for the Attitude Control System specific activities), but the ones of principal interest are Quiet mode and Flare mode; switching between these modes is normally controlled by a flare flag generated by the WBS instrument (its threshold is usually set to GOES X-ray class C2). The allocation of telemetry to the four instruments and other subsystems differs in the different modes, but the main difference between the two principal modes is that there is more HXT data in Flare Mode than in Quiet Mode (where the extra space is used by the SXT).

For 5 or 6 orbits each day, the spacecraft passes over the Kagoshima Space Centre (KSC).

During these passes, all spacecraft commanding is performed; commanding at other times is through the on-board deferred command store. The BDR is dumped at each KSC pass, and is in addition dumped at a number of passes (up to 10/day) over the ground-stations of the NASA Deep Space Network (DSN). After each KSC contact, the data received from Yohkoh is reformatted so that quick-look analysis can be performed. Once per week, when all the data has been transmitted to ISAS, the Yohkoh Data Archive is produced by running the reformatter once more.

See the special edition of **Solar Physics (1991, Vol 136, No. 1)** for more information on the instruments and spacecraft. The **Instrument Guide** of the **YAG** also contains information about instruments; in some cases it is more up to date than the special edition.

1.2 Yohkoh Data Files

1.2.1 Instrument Data Files

The Yohkoh reformatter produces a file for each instrument for every orbit, with a start time corresponding to the (predicted) first minute of Sun of the orbit (spacecraft dawn); these files have prefixes of BDA for the BCS instrument, HDA for the HXT, and WDA for the WBS; for the SXT, two files corresponding to full and partial frame images are produced, the SFR and SPR files. All the files are simple byte streams that are interpreted into IDL structures when the files are read in. Each file is broken into a number of sections, the ones of principal interest are the roadmap, index and data sections. For each instrument mode (image, spectral or other integration), there is an index and a data record; the roadmap is a short summary of all the index records. The contents of each section varies slightly between the instruments (as defined by the structures), but there are a number of fields that are always present (e.g. time, spacecraft mode, etc); each file is intended to stand by itself, so some information is duplicated between the files. Details of the format of the Yohkoh data files can be found in Appendix B.1 of this volume of the **YAG**.

The names of these files are of the form **"pppyymmdd.hhmm"**, where *"ppp"* is the three-letter prefix, *"yymmdd"* is the date (year, month, day) and *"hhmm"* is the time (hours and minutes). For example, "spr911115.1712", would contain SXT partial-frame images for the orbit starting at 17:12 UT on 15 November 1991. Details of the Yohkoh naming conventions can be found in Appendix B.2 of this volume of the **YAG**.

1.2.2 Secondary Database files

From the reformatted instrument files, a number of secondary files are produced. These include the observing logs and other synoptic files, and also a number of files relating to instrument performance and calibration. One secondary file per type per week is produced, with names of form **"pppyy_wwa.nn"**, where *"ppp"* is the prefix, *"yy_ww"* is the year and week, and *"nn"* is a revision number; the *"a"* was allocated in case the files had to be split - this has not happened. Examples of files of this type include the Yohkoh event log (EVN), the Yohkoh orbital ephemeris (FEM), the instrument observing logs (OBD, OSF, OSP and OWH), the SXT Dark Current (SDC), etc. For example, the BCS observing log for week 30 of 1993 would be "obd93_30a.01".

There are also a few files produced on a monthly basis, e.g. the SXT Full Frame Images (FFI) monthly file (SFM); these have names of the form **"pppyymm00.0000"**, where *"ppp"* is the prefix and *"yymm"* is the year and month of the file. For example, "sfm930600.0000" would contain an SXT FFI image for each day of the month of June 1993.

A list of all file prefixes can be found in Appendix B.2 of this volume of the **YAG**, and may also be obtained by typing HELP_PREFIX in IDL.

1.2.3 Other Database Files

There are a number of other ancillary datasets that are included in the framework of the Yohkoh Data Archive. These include observations from other spacecraft, and from ground-based observatories. These have names of the form of the secondary files ("pppyy_wwa.nn", see above), and examples include the GOES event log (GEV), GOES lightcurves (GXT), CGRO Batse lightcurves (GBL), Nobeyama radio lightcurves (NTS), etc.

Some image data from ground-based observatories are also stored, usually in FITS format. Datasets of this type include various images from Big Bear, He 10,830 Åimages and magnetograms from Kitt Peak, and H-alpha images from a number of sites around the world. Ground-based data files have a prefix of the form Gsx, where the "G" stands for ground-based, "s" is the site and "x" is the data type. Thus, all Big Bear images have prefixes of the form "GBx", and Kitt Peak images prefixes are "GKx"; all H-alpha images have prefixes of the form "GsH".

Again, a list of all file prefixes can be found in Appendix B.2 of this volume of the **YAG**, and may also be obtained by typing HELP_PREFIX in IDL.

1.2.4 Location of Available Files

The volume of data produced by Yohkoh is several hundred Gbytes, without considering all the supporting datasets. This clearly exceeds the data storage capabilities of the computers most people use and many sites only have a fraction of the files available.

At Institute of Space and Astronautical Science (ISAS) in Japan, the Solar Data Archive Centre (SDAC) in the USA, the UK Yohkoh Data Archive Centre (YDAC), and other principal sites for Yohkoh analysis, all the catalogue files are held on-line, as are all the secondary instrument files and all pertinent subsystem files needed for data analysis. These files are generally stored under subdirectories and are accessed by the software using environment variables (or logicals under VMS) of the form **"$DIR_sss_ppp"**, where *"sss"* is a subsystem and *"ppp"* is usually a prefix. Thus, the SXT dark current files would be found under a directory addressed by the environment variable "DIR_SXT_SDC"; this would probably (but not necessarily) correspond to the directory /ydb/sdc. A list of all defined environment variables can be found in Appendix B.2 of this volume of the **YAG**.

If they are held, files from ground based observatories are stored under directories of the form **"/gboyy_ww"**, where *"yy_ww"* is the year and week.

A number of the Yohkoh instrument data files are usually held on-line. How many, and the organization of where the files are held, differs from site to site. For example, there are a number of large flares held at YDAC under /ydb/flares. Within IDL, all public directories holding data are determined by the software through routines like DATA_PATHS.

1.2.5 What files are needed for Analysis

When analysing Yohkoh data, the only data files needed are those for instruments and periods of interest. Other secondary data files may be needed for detailed analysis; in most cases, only secondary files relating to the weeks of interest will be needed. It is therefore possible to analyse Yohkoh data with only a partial installation of the data and secondary files, changing which ones are kept as the period of interest changes. At the YDAC, all necessary secondary files are stored on-line and these remarks relate mainly to people who wish to continue analysis work at their home institution after first selecting the data through YDAC.

1.3 The Analysis Software

Almost all the Yohkoh analysis software is written in IDL and is generally optimized for computer workstation use. The BCS, HXT and WBS have a few routines in Fortran and C, but because of difficulties in the transportability between platforms and operating systems, and because of the difficulties that occur if the structures are modified, these are being phased out.

Quite detailed descriptions of general user software for each instrument are given in later parts of this volume of the **YAG** (the **User Guide**). Nevertheless, a few comments on simple access are given below.

While using IDL, more information on any individual routine may be found using the DOC_LIBRARY and XDOC routines; information on the calling sequence of any procedure or function may be found using the routine CHKARG. For example, doc_library,'help_prefix' would return information contained in the documentation header of the routine HELP_PREFIX, while chkarg,'help_prefix' would return information on the calling sequence of the routine.

1.3.1 Examining the Catalogue Files

The initial access route into the Yohkoh Data Archive should be through the catalogue files. For Yohkoh itself, these include the weekly observing logs, the event log, and orbital ephemeris, and the monthly SXT FFI image files. It is possible to do many preliminary searches and statistical surveys using the abstracted data of the observing logs, which for example contains time series of counting rates from each instrument. Additional information can be found in the GOES event log, the files containing the GOES 1-minute lightcurves, and the list of NOAA active regions.

There are many routines that can be of help: Listings of the Yohkoh event log, the orbital ephemeris, the GOES event log, and list of NOAA active regions may be accessed through routines PR_ENV, PR_FEM, PR_GEV, PR_NAR, etc.; the format of the calls for this type of routine is usually PR_ppp,start_time,end_time. Similarly, plots of the GOES lightcurves and the BATSE lightcurves may be produced with PLOT_GOES and PLOT_GBL; again the call is often of the form PLOT_ppp,start_time,end_time. For example, plot_goes,'1-Jun-93' would plot the GOES data for the June 1st, 1993.

Plots from the observing log may be produced with calls to PLOTY, OBS_PLOT and BCS_24HR_PLOT (the later routine is BCS specific).

1.3.2 Extracting Instrument Data

All of the reformatted instrument files, and many other types of files can be accessed in IDL using the routine YODAT. Through a series of simple questions, the user may select one or more data files of a specified type, and select the desired time interval. The programme then reads in the file, producing a roadmap structure, and if requested, index and data structures. These can then be accessed and displayed by many routines (see later sections of the **YAG**), but a couple of routines to initially play with are STEPPER for the SXT and BCS_SPMOVIE for the BCS; STEPPER will produce a movie of images, while BCS_SPMOVIE does the same for BCS spectra. LIST_BDA,INDEX allows the user to list the BCS modes, while PRINT,GET_INFO(INDEX) will produce similar following the reading of an SXT data file.

2 General Yohkoh Software (Getting Started)

Welcome to the analysis of Yohkoh data. This document is intended to help you use the Yohkoh software and to understand the Yohkoh databases and some of the instrument calibration issues. The Yohkoh software has been written mostly in IDL, or Interactive Data Language, which is a licensed software program available from Research Systems, Inc. In order to run the Yohkoh analysis software, you must have IDL Version 3.0 or later installed on your computer or workstation. The Yohkoh Guide assumes that the reader has some familiarity with IDL already. For specific IDL questions, you are encouraged to consult the IDL User's and Reference Guides.

The Yohkoh software is known to work currently on several Unix-based platforms (e.g., HP, Sun, Mips, DEC, SGI). It is also available for VMS computers, however, we recommend that Unix be selected for analysis of Yohkoh data if possible. The installation and upgrade procedures described in the Reference Guide support only Unix systems.

Before beginning, it may be helpful to refer to Chapters 1 and 2 of the Reference Guide (Volume 2) which briefly describe the Yohkoh instruments and also provide an introduction to the Yohkoh data and software organization. More details about the instruments can be found in the Instrument Guide (Volume 3). You will want to make sure the Yohkoh software is set up on your computer. Talk to your local manager about this or follow the installation procedure described in the appendix of the Reference Guide for Unix-based machines. The installation can be accomplished easily by those possessing a moderate amount of Unix experience. Next, check that you have a `.yslogin` file on your home directory. Then type the following command:

% **source ~/.yslogin**

You must type this command at the beginning of each new login session or see the Yohkoh Reference Guide about how to customize your `.login` or `.cshrc` file.

Next, if you have Yohkoh data files on your disk, you should be able to start looking at data after typing:

% **idl**

to start the IDL software system. YODAT (described in the next section) is the most common way to read Yohkoh data files and display various information or, in the case of SXT, to display images. If you have not yet copied any Yohkoh data files to your computer's disk, you will probably want begin with sections C.1 and C.2 which describe how to read the Yohkoh data archive tapes.

2.1 How to Access the Yohkoh Observing Logs and Event Logs

The Yohkoh observing log contains a wealth of information which is readily accessible for searches and quick comparisons between instruments. In the following example the format of the plots may not be exactly what you need, but it is important to realize how simple the PLOTY program was to write since the observing log data files already exist. There is a lot of potential in the observing logs which is not being tapped as of yet, but the following sections touch on a few capabilities which currently exist.

2.1.1 Performing Searches Using the Observing Log (XSEARCH_OBS)

XSEARCH_OBS provides graphical front end access to the Yohkoh Observing Log. The user can select search criteria and define various options for output [1]. The calling sequence is:

IDL> **xsearch_obs**

The following features are anticipated.

- graphical definition of instrument parameters initiate search of observing log (for example, sswhere-derived interface is used for SXT searches).
- save defined search parameters in a configuration file which can be used for a repeat search on different times or as starting point for future parameter modifications.
- read in existing search parameter configuration file to set initial (or final) parameters – this permits archiving of useful search definitions and promotes repeatability.
- provide access to low level routines which translate observing log records to archive tape names or MO disk names and data sets and ultimately allow reformatted data extraction.

2.1.2 Displaying a Light Curve Set of all Instruments (PLOTY)

PLOTY is a crude method to plot the light curves from all of the Yohkoh instruments on the same plot and the same time axis. If you have the observing log on-line and wish to use it to make the plots, you can specify the plot times. It is recommended not to plot more than approximately 24 hours of data at a time. An example is:

IDL> **ploty, '8-may-92 15:00', '8-may-92 18:40'**
IDL> **ploty, '8-may-92 15:00', '8-may-92 18:40', /zoom, /big**

Fig. 2.1 is an example of the plot which can be produced. Note that the BCS counting rates drop to zero at around 22:37 because they become saturated. The format for the plots of this program might change in the future, but the capability to plot data from several instruments with the same time axes will remain.

2.1.3 Displaying a Light Curve Set of all Instruments (OBS_PLOT)

TODO Needs text

2.1.4 Listing of the SXT Images and Heliocentric Locations

There are a variety of routines which will produce listings of the images and pointing locations of the SXT partial frame images.

The routine PR_SXTOBS will read the observing log and list every FFI and PFI image taken for the times requested. The format of the output is simply the output of GET_INFO which is described in detail in the Reference Guide (PR_SXTOBS simply reads the observing

[1] Preliminary at the time of publication — expected working prototype for SXT searches in Feb, 1994

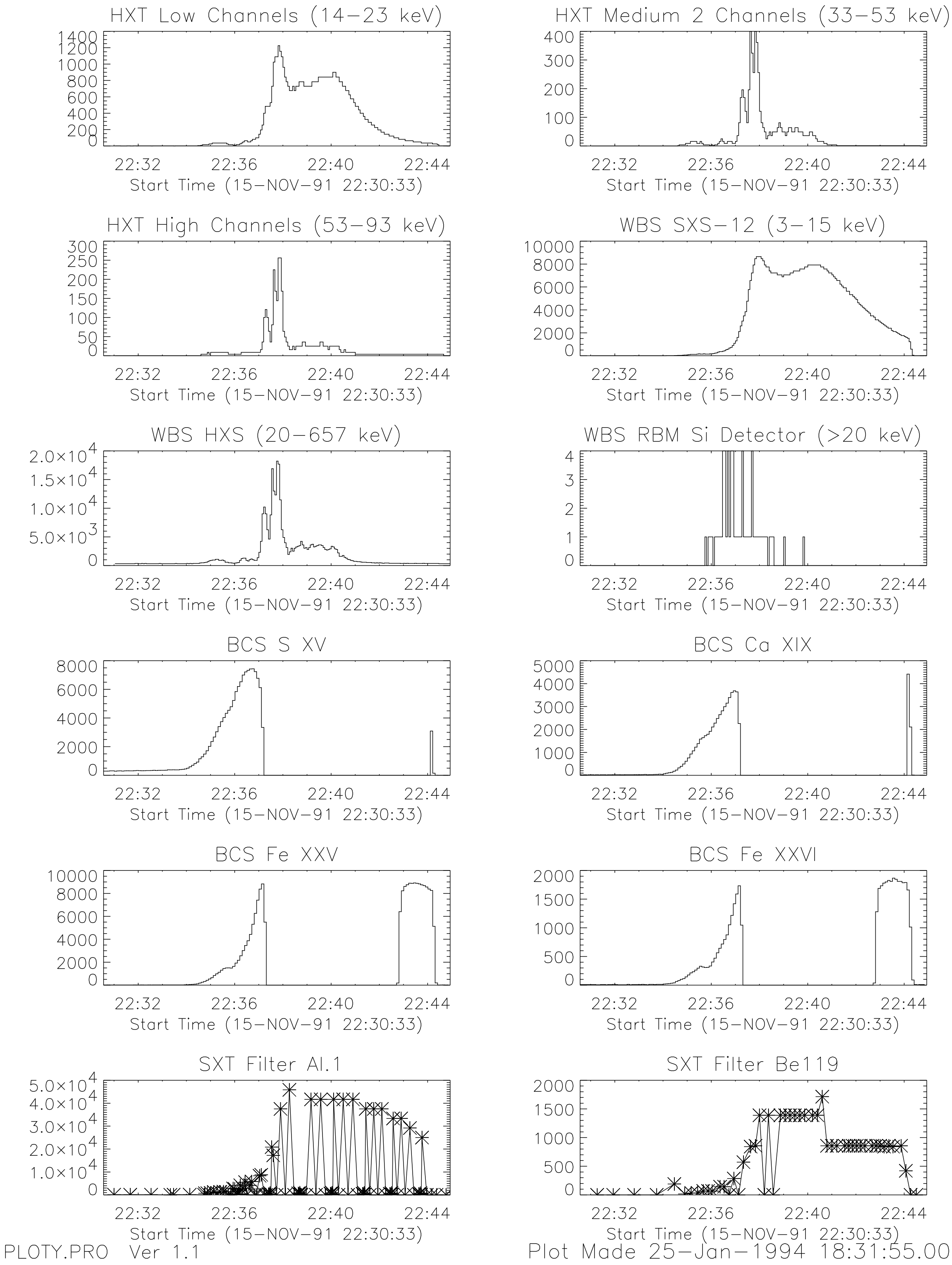

Figure 2.1: ploty, '15-nov-91 22:30','15-nov-91 23:50'

log and then calls GET_INFO). The command:

IDL> **pr_sxtobs,'15-nov-91 22:00','15-nov-91 22:05',/long**

will produce the output:

```
PR_SXTOBS.PRO Run: 20-Dec-1993 21:06:29.00
SXT Observing Log Search from: 15-NOV-91  22:00:00 to 15-NOV-91  22:05:00

   0 15-NOV-91  22:03:40  QT/H  Open /Mg3   Half Norm C  17    338.0   64x 64  100%  S15W11   0%     41.3
   1 15-NOV-91  22:03:48  QT/H  Open /Al.1  Half Norm C  17    338.0   64x 64  100%  S15W11   0%     41.2
   2 15-NOV-91  22:03:56  QT/H  Open /AlMg  Half Norm C  19    668.0   64x 64  100%  S15W11   0%     41.0
   3 15-NOV-91  22:04:04  QT/H  Open /Be119 Half Norm C  27 10678.0   64x 64  100%  S15W11   0%     40.9
   4 15-NOV-91  22:04:18  QT/H  Open /Al12  Half Norm C  27 10678.0   64x 64  100%  S15W11   0%     40.7
   5 15-NOV-91  22:04:32  QT/H  Open /Mg3   Half Norm C  19    668.0   64x 64  100%  S15W11   0%     40.4
   6 15-NOV-91  22:04:36  QT/H  Open /Al.1  Half Norm C  23   2668.0  512x512  100%  Full D   2%     40.4
   7 15-NOV-91  22:04:44  QT/H  Open /Al.1  Half Norm C  17    338.0   64x 64  100%  S15W11   0%     40.2
   8 15-NOV-91  22:04:52  QT/H  Open /AlMg  Half Norm C  19    668.0   64x 64  100%  S15W11   0%     40.1
```

The routine SXTPNT_SUM will read the observing log and give a summary of the unique PFI pointing for a given period. It groups the summary listing based on each unique pointing location over time. The command:

IDL> **sxtpnt_sum, '5-oct-93 13:00', '6-oct-93 1:00'**

will produce a file in your home directory with the name "sxtpnt_sum.txt" (unless you specify the output file name with the *outfil* keyword option). The beginning of this file is:

```
OBS_SUMMARY  Ver 3.0
Program Run: 20-Dec-1993 21:12:40.00

     Date          Times         #Img  Mode    Table#    Filter A/B  Res  Exp  C DPE  msec    ImgSize  Loc   PntChange  NOAA

----------------------------------------------------------------------------------------------------------------------------
  5-OCT-93  13:04:24 - 14:58:24    BCS: 441    SXTP: 146    SXTF:  19    W_H: 677     HXT-L:    1 SXS2:   16  S11E18
     5-OCT-93  13:04:24 - 14:56:50   39  QT/Hi    733/1  = Open /Al.1  Full Norm C  16   238.0  128x128 S11E18  (  0.2) [ 7592]
     5-OCT-93  13:04:42 - 14:57:52   37  QT/Hi    733/1  = Open /Al12  Full Norm C  24  3778.0  128x128 S11E18  (  0.2) [ 7592]
     5-OCT-93  13:05:14 - 14:58:24   36  QT/Hi    733/1  = Open /Al.1  Full Norm C  16   238.0  128x128 S11E18  (  0.2) [ 7592]
     5-OCT-93  13:05:54 - 14:56:18   34  QT/Hi    733/1  = Open /Al12  Full Norm C  24  3778.0  128x128 S11E19  (  0.2) [ 7592]
----------------------------------------------------------------------------------------------------------------------------
  5-OCT-93  15:16:24 - 15:16:24    BCS:   0    SXTP:   1    SXTF:   0    W_H:   0     HXT-L:    0 SXS2:    0  S13E27
     5-OCT-93  15:16:24 - 15:16:24    1  QT/Hi    733/1  = Open /Al12  Full Norm C  24  3778.0  128x128 S13E27  (  0.0) [ 7592]
----------------------------------------------------------------------------------------------------------------------------
  5-OCT-93  15:16:40 - 16:40:56    BCS: 190    SXTP: 133    SXTF:  15    W_H: 606     HXT-L:    1 SXS2:   16  S11E15      *
     5-OCT-93  15:16:40 - 16:40:56   33  QT/Hi    733/1  = Open /Al12  Full Norm C  24  3778.0  128x128 S13E17  (  0.7) [ 7592]
     5-OCT-93  15:16:56 - 16:39:20   35  QT/Hi    733/1  = Open /Al.1  Full Norm C  16   238.0  128x128 S13E17  (  0.7) [ 7592]
     5-OCT-93  15:17:28 - 16:39:52   34  QT/Hi    733/1  = Open /Al12  Full Norm C  26  7548.0  128x128 S13E17  (  0.7) [ 7592]
     5-OCT-93  15:18:00 - 16:40:24   31  QT/Hi    733/1  = Open /Al.1  Full Norm C  16   238.0  128x128 S11E15  (  0.7)
```

It can be seen that the listing contains information on the number of images of a given filter for each time span, the heliocentric coordinates, the NOAA active region number(s) if applicable, the peak flux in HXT-LOW and SXS2, the GOES flare event level, and a few other things. A "*" at the end of the summary line says that the time of the peak of the GOES event is contained within the time span shown.

2.1.5 Light Curves from BCS, WBS and HXT

The routine BCS_24HR_PLOT (discussed on page 19) uses the data in the observing log by default, and the routines PLOTT_HDA (discussed on page 27) and PLOTT_WDA (discussed on page 53) use the observing log if the calling sequence specifies the starting and ending time. See the description and sample plots of those routines described Sections 3.1.3, 4.2.1, and 6.1.1.

2.1.6 Creating a List of Yohkoh Events (PR_EVN)

PR_EVN helps you find times for which Yohkoh data are available. An event is driven by a change between QUIET and FLARE mode, or when there is a data gap of more than 60 seconds. By typing:

IDL> **pr_evn, '23-jun-92'**

a list of the times that Yohkoh data are available and the number of datasets available for each instrument is listed for 24 hours starting at 23-jun-92 00:00. By typing:

IDL> **pr_evn, '15-nov-91 20:00', '17-nov-91 15:00', /flare**

all times that Yohkoh was in FLARE mode between those times are listed. By typing:

IDL> **pr_evn, '1-jan-92', '9-jan-92', /flare, /counts, mindur=5, outfil='pr_evn.results'**

the FLARE modes for 1992, and since the /COUNTS option was used, the maximum counting rate for certain WBS, HXT, and BCS channels are printed instead of the number of datasets available. It also prints the GOES classification when it is available. The command shown above results in the following listing:

```
PR_EVN.PRO Run on  9-Dec-1993 12:07:26.00
Search Start Time:  1-JAN-92  00:00:00
Search End Time:    8-JAN-92  00:00:00
Minimum event duration:     5.00 minutes

NOTE: "*" means Yohkoh missed the beginning or end of the flare duration
      "+" means Yohkoh has a sub portion of data within the flare duration

     Start         End        Duration   DP          HXT     /         WBS                 /          BCS          / GOES
     (UT)         (UT)          (min)  Mode    Low  Med2 High /   SXS2   HXS   GRS1   RBMSD / Fe26 Fe25 Ca19  S15 /

 1-JAN-92  03:29:21-03:34:21   5.00  Flare     49    9    9 /    121    900    729      1 /  866 9676 5590 7586 / M2.8
 1-JAN-92  22:53:37-23:05:40  12.05  Flare      1    4    9 /    196   4624   7744      4 /  240  396  430 4083 /
 2-JAN-92  21:05:29-21:15:28   9.98  Flare     36    4    9 /     64    625    900      1 /  796 4210 9593 4340 / C9.3
 2-JAN-92  21:36:11-21:56:04  19.88  Flare     16    9   36 /    676  10000   7396      9 /  513 2210 3013 7290 / M1.9
 3-JAN-92  22:02:59-22:15:32  12.55  Flare      4    4   25 /    169   4489   7744     16 / 2083 3790 4573 9610 /
 4-JAN-92  08:48:43-08:58:40   9.95  Flare      4    9   16 /    256  24336   7396   4900 /   70  120  196 1276 /
 4-JAN-92  11:11:29-11:21:24   9.92  Flare     36    1    4 /     64    400    576      0 /  486 3150 2953 7210 /
 4-JAN-92  13:43:13-13:53:12   9.98  Flare     36    9    9 /     64   2809    576     25 /   80  490  883 3783 / C6.4
 5-JAN-92  13:15:33-13:25:14   9.68  Flare     25    4    9 /     64    576    784      0 /  626 4286 3930 6960 /
 5-JAN-92  15:14:55-15:24:20   9.42  Flare    576  961  196 /      4    441    841      0 /    0    0    0    0 /
 6-JAN-92  15:41:46-15:50:36   8.83  Flare    625  961  196 /      4    484    961      0 /    0    0    0    0 /
 6-JAN-92  19:55:06-20:09:39  14.55  Flare      4   16   36 /    400   7569   7225     16 / 3953  963  493 3903 /
 7-JAN-92  04:05:40-04:17:55  12.25  Flare     25    4    4 /     64    441    441      0 /  573 4196 4346 6420 / M1.5
 7-JAN-92  20:21:52-20:31:47   9.92  Flare     16    9   16 /     64   1764   2025     64 / 2116 1596 2000 4690 / C8.9
```

2.1.7 Direct Reading of the Observing Log

A sample standard calling sequence for reading the observing log data files for all instruments for a 12-hour period starting at 00:00 on 8-May-92 is:

IDL> **rd_obs, '8-may-92', '8-may-92 12:00', bcs, sxtf, sxtp, w_h, fid**

An example for reading only SXT full frame entries is:

IDL> **rd_obs, '1-jan-92', '1-jan-93', bcs, sxtf, /nobcs**

Be aware that these datasets can get quite large so the time spans should probably not exceed 10 days unless you are only accessing SXT full frame entries. See the Reference Guide and the documentation header for RD_OBS for more details.

2.2 Summary SXT Full-frame image data

2.2.1 The Monthly SXT FFI files (SFM)

XMOVIE_SFM, DISP_MONTH, GET_SFM

2.2.2 Browsing the Laser Movie (at ISAS, LPARL or MSU)

A laser disk movie exists at both ISAS and LPARL which contains SXT images for the whole Yohkoh mission. These full frame images are all of the SFD images which are available, which means 50–100 images a day for over two years so far. It is possible to browse through the video recordings using a remote control to look for events of a specific type or a specific time.

The routine VIDEO_MENU will allow a user to access the "special laser movie" disk which has individual movies of particular events. See the description in the Reference Guide for more details.

2.3 Accessing the Yohkoh Data

2.3.1 Interactive Access (YODAT)

YODAT will access any data from BCS, HXT, SXT, or WBS. It will also read some FITS files which have been renamed to use the Yohkoh convention. This procedure:

1. Shows what data are on line
2. Lets you select specific files to read and display
3. Reads the ROADMAP for the selected files
4. If you select data sets to be read, it creates the variables *data* and *index* which you can then manipulate using IDL or other higher-level Yohkoh routines.

This procedure is run by typing (make sure that yodat is in lower case if you are running on a Unix or Ultrix machine):

IDL> **.run yodat**

The prompt that you will receive will look something like this:

```
% Compiled module: $MAIN$.
               *******  YODAT V9.2 (7-Jul-93)  *******

It is possible to have YODAT extract every "n"th dataset by setting
the variable QYODAT_NSAMP to 1.  You will be asked one extra question

It is possible to read the Ground Based Observation (GBO) FITS files
by using a command like: MENU g*
```

```
          gb_ files are from Big Bear, gk_ are from Kitt Peak
RFITS will be called with /SCALE option if QYODAT_SCALE is set to 1

Enter MENU if you want to use the filenames menu option
Enter SAME if you want to access the same fileID for a different instrument
Enter MANY if you want to use menu option and extract many files
Enter TIME if you want to enter the start/end time to extract
Enter QUIT to abort out of YODAT

Enter file name (or wild cards)
```

The first step is to select the data files to be read. The name of the file(s) selected is saved in the variable *infil*. When one or more files are selected, the details of the datasets in those files is read into the variable *roadmap*. There are several different techniques for selecting files.

- You can type the full file name, including the directory if the data file does not exist in the default directory. Wild cards are acceptable (e.g., **spr9301*.***).
- You can use the MENU option to get a listing of all of the files that are on-line (this option uses the output of DATA_PATHS to get a list of the data directories). For example, if you type **menu spr***, you will get a list of all SPR files that are on-line. Or if you type **menu bda9207*** you get a list of all BDA files for Jul-92 that are on-line. You can only extract one file with the MENU option so click on the one file you wish to read.
- You can use the MANY option to select several files. This option is also menu driven, and is similar to the MENU option but you can select multiple files. After you have selected all the files you want to read, click on QUIT/EXIT.
- If you have already read a file, and wish to get the data for the same time but from a different experiment, you can use the SAME option. For example, if you have just read spr911115.2141, and want to get the BDA file for that orbit, type **same bda**.
- Once you have selected a file, you can re-run YODAT and simply hit RETURN and the same file is selected again so you can select a different set of down-line options.
- If you type **quit**, then YODAT will be stopped.
- If you are running YODAT over the network, then it might not be practical to run options which create an X-window. You can specify the file and directory using wild cards. The list of selected files will then appear with a number next to each file. You can then enter the starting and ending file number to select a set of files. For example, you could type "/yd*/*/spr930623*" or "/yd6/flares/spr91*".

The "MENU" and "MANY" options search all data directories which are returned by the function DATA_PATHS. At ISAS and LPARL this routine automatically checks for new data directories that are on-line, in addition to some "hardwired" directories which it always checks. It is possible for a user to add directories to the list which DATA_PATHS returns by using the routine NEW_DPATH. In the following example, the user adds the directory "/yd3/morrison/agu_paper" to the list of directories which will be checked by YODAT.

IDL> **new_dpath, '/yd3/morrison/agu_paper'**

The second step in accessing any data set is to perform a quick review of the data available, and select the data sets to be read. The options for selecting the data are listed below.

```
Enter the number of data sets to extract
   * If you enter 0, all datasets will be extracted
   * If you enter -99, then it uses the datasets specified in variable "SS"
   * If you enter -888, then the file is not read
   * For SXT, enter a negative # (from -1 to -13) to access only that seq#
   * For SXT, enter -777 for sequence menu option
              Enter -776 to use "show_obs3" and select
              Enter -775 to use "plot_fov" and select
              Enter -774 to list the sequence summary
              Enter -773 to use "show_obs4" and select
              Enter -772 to use SSWHERE to select
   * For HXT, enter -666 to plot and select on SUM_L light curve
              Enter -665 to plot and select on SUM_M1 light curve
              Enter -664 to plot and select on SUM_M2 light curve
              Enter -663 to plot and select on SUM_H light curve
   * For WBS, enter -555 to plot and select on SXS1 light curve
              Enter -554 to plot and select on SXS2 light curve
              Enter -553 to plot and select on HXS light curve
   * For BCS, enter -444 to plot and select on S XV light curve
              Enter -443 to plot and select on Ca XIX light curve
              Enter -442 to plot and select on Fe XXV light curve
              Enter -441 to plot and select on FE XXVI light curve
   * For any, enter -333 to extract only flare mode data
```

If you selected data to be read, then the results are saved in the following variables:

infil	the input file(s)
index	the index structure for datasets extracted
data	the data (array or structure) for datasets extracted
roadmap	the complete roadmap for all datasets in *infil*
dset_arr	the list of dataset numbers selected
info_array	for SXT only, a text array describing each image

For SXT options 'SHOW_OBS3', 'SHOW_OBS4', and 'SSWHERE', see the description in the Reference Guide. The light curve options (all of the −400, −500, and −600 series) will show a light curve of the selected channel. YODAT will prompt you with the instructions at each step. It is possible to select the range in time you wish to look at by:

1. Click with the left button at the starting time
2. Click with the right button at the ending time
3. Click with the middle button to exit

An example of how one would use the −99 option is:

1. Run YODAT to select the file and to read the roadmap.
2. Exit out of YODAT with the −888 option.

3. Select the data to be extract using some high-level routine or simple IDL 'where' commands. Put the results in the variable 'ss'.
4. Run YODAT again selecting the same file by just hitting RETURN.
5. Selecting option −99.

2.3.2 Advanced YODAT Options

It is possible to use the −777 option to select SXT data by the sequence table entry (all images taken with the same filter, resolution and field of view). If you wish to abort from this option, click on the title (the heading of the table). The most common technique is to just select one sequence entry, but it is possible to select many entries. The procedure is to select option −777, and then:

1. Click on "Enable option to select multiple sequences (reset)"
2. Click on all of the sequence entries you wish to extract
3. Click on "If enabled selecting multiple options, now extract data"

Another option is to use the −772 option to call SSWHERE which allows the selection of SXT data based on filter, image resolution, DP mode and rate, exposure duration, and several other options. See the SSWHERE description in the Reference Guide for more details.

2.3.3 Lowlevel Access Routine (READ_XDA)

2.3.4 Most Common BCS Display Routines

The BCS can provide detailed light curves and spectra. After reading the BCS data with YODAT, you may wish to use the following commands:

```
IDL> plott_bda, index                    ;light curve of all 4 chan
IDL> plott_bda, roadmap, psym=10         ;light curve of all 4 chan
IDL> wbda, index, data                   ;light curves and spectral plots
IDL> plots_bda, index, data              ;spectral plot of all 4 chan
```

2.3.5 Most Common HXT Display Routines

Raw HXT data can provide light curves and spectra of the non-theraml emission in flares. After reading the HXT data with YODAT, you may wish to use the following commands:

```
IDL> plott_hda, index                    ;light curve of all 4 chan
IDL> plott_hda, roadmap, psym=10         ;light curve of all 4 chan
```

2.3.6 Most Common SXT Display Routines

The SXT provides images of the thermal plasma in flare, active regions, and the quiet sun. After reading the SXT data with YODAT, you may wish to use the following commands:

```
IDL> stepper, data                       ;movie of data
IDL> stepper, data, xsiz=512             ;movie of data enlarged to 512x512
IDL> stepper, data, info=info_array      ;movie of data with times...
```

```
IDL> xy_raster, index, data                        ;raster display of data
IDL> sxt_prep, index, data, index2, data2, /reg    ;calibrate data
IDL> show_obs3, index                              ;time line of data
```

2.3.7 Most Common WBS Display Routines

The WBS provides light curves and spectra from soft X-ray to gamma rays. After reading the WBS data with YODAT, you may wish to use the following commands:

```
IDL> plott_wda, index                ;light curve of 7 sub-instruments
IDL> plott_wda, roadmap, psym=10     ;light curve of 7 sub-instruments
IDL> plots_wda, index, data          ;simple spectral plots
```

2.4 How to Perform Spectral Fitting

It is possible to perform spectral fitting on the HXT, WBS and SXT data by using several routines written by Jim McTiernan. You can fit three or more filters, and thermal and non-thermal spectra. There are programs called SXTHXT_FSP and SXTHXTBOX_FSP, which will fit combined spectra for the two instruments (SXT and HXT). See the reference manual for details on how to run the program.

2.5 Selection and Extraction of an Arbitrary Shaped Sub-Image (WDEFROI) [*]

The WDEFROI function allows the user to select and cut-out an arbitrary shaped region-of-interest (ROI) from a displayed image via point and click with the mouse. With WDEFROI one can isolate a small intricately shaped active region from a larger image for further analysis. This can be particularly useful for reducing the computation time for calculating temperatures and emission measures of large data cubes. WDEFROI is not SXT specific, so it can be used on ground-based data or other derived data such as temperatures and emission measure. One can obtain light curves of the selected ROIs by passing normalized data and the times of data.

Example call to WDEFROI to obtain a sub-image via the keyword image:

```
IDL> sub_img = wdefroi(data, /image)
```

The program starts by bringing up the 1st image and running STEPPER to allow the user to pick a working image frame from the variable data for selecting ROIs (One can specify the working image frame by using the *sub=image_number* keyword and avoid running STEPPER). Once the working frame has been selected (say image 5), the user should exit STEPPER, and a widget window will appear entitled WDEFROI.

The top row of buttons of this widget is used to pick the mode of ROI selection and to change the color table (via XLOADCT). Below these buttons is the online instruction box which displays useful general information and specific information on how to make box, polygon, and contour selections with the mouse. This is followed by some contour controls, namely a slider widget which can be used to set the contour level at a specific value. Then there is a row of control buttons: a select button to blacken-out all but the selected area of the image; reset button to re-display the original image; light curve to display a light curve of the selected area; exit to quit the widget and to return to IDL command line. The final row has a display box for the total number of pixels selected by box, polygon, and contour.

Example call to also plot light curves from normalized data:

IDL> **sub_img = wdefroi(norm_data, /image, zoom=6, sub=5, /lc, time=index)**

Where, *norm_data* is a data cube of small images say 64 by 64 pixels, *zoom* specifies a zoom factor of 6, *lc* keyword requests light curve plot, and the times for the light curve are passed via the *time* keyword.

Note: If the input data cube contains full frame images and the area of interest is very small it may be best to call WDEFROI twice: 1st to isolate the general area from the full context image and a second time with some zoom factor to make contouring and extracting the smaller features easier.

3 Bragg Crystal Spectrometer (BCS)

The purpose of this section is to help the user access the BCS data using the various pieces of BCS software that have been written. All the routines discussed use the reformatted database; IDL code assumes IDL Version 3 (or greater). Note that some of the widget-based IDL routines only work with more recent releases of IDL.

This document only presents an outline of how to use the routines. Please use the IDL routines DOC_LIBRARY, XDOC and CHKARG, together with the Reference Guide chapter for further information on any particular routine.

There are several additional documents that describe different parts of the BCS software. All are kept on $DIR_BCS_DOC; in particular:

- Please refer to the document BSC_SOFTWARE.TXT for the most recent information on the BSC routines that Dominic Zarro has been writing.
- Please also refer to Dave Pike's document GS_PROCS.TEX for further information on the "GS" (grey scale) suite of software.
- There are also several documents relating to the BSD FORTRAN software.

Notes:

1. BCS data files have the prefix "bda" – the rest of the name represents the start time of the file in the form "yymmdd.hhmm".
2. In the BCS section of the manual, data and index arrays relating to the BDA file are called *index* and *data*. Those related to other files and structures will will have some type of prefix in the example calls, e.g. "bsd_data", "bsd_index"
3. Often in procedure calls there are additional, optional parameters than can be used to further define what the routine will do – these are indicated by the use of [ ] within the example of a routine calling sequence; [,etc] means that there are several additional parameters and the user should refer to the Reference Manual.
4. For most applications, the DP_SYNC[2] data are not needed — they are however required for deadtime correction within MK_BSC.
5. On the VMS machines, the BDA data should be stored under several directories that can be accessed with the logical BDA_DAT: (e.g. bda_dat::bda911024.*)

Section 3.1 details how to first access the BCS data. To do detailed analysis work on the spectra, see sections 3.3 and E.1; otherwise, use the routines in section 3.2. Several other useful BCS routines are summarized in section A.2.

3.1 Entry Point Into the Data

Basically here we are asking why are you wanting to look at BCS data. This is because the approach used to access the data if you are interested in a particular time is different from that used if you are trying to find the occurrence of a particular type of event without any *a priori* knowledge of its occurrence.

If you are interested in a particular time interval, you have probably found out that there is an event from the book of BCS Ca XIX light curves, SA plots, PLOTY plots, GOES light curves, the event lists, the occurrence of a flare flag or from something seen in another instrument. In this case, the way to proceed is to use YODAT or WBDA.

[2] The BCS dp_sync data contain instrument data, including the detector event counters, in a form that is synchronous with the rest of the frames of the data from the spacecraft DP. The spectral data, because of the queue memory, are asynchronous to this data.

3.1.1 Accessing a Specific Time Interval Using YODAT

YODAT is a general purpose routine that will access all types of Yohkoh data, together with certain datasets from ground-based observatories. The detailed operation of YODAT is described on page 10, but here we present a short summary.

After starting YODAT (within IDL) the user must select a BDA file (or files). YODAT reads in the roadmap of the selected file and the user must then decide on which further selection procedure to use. The user can:

1. Use the −44x options within YODAT to select a time interval using a BCS light curve and read in the data for that time interval – the best channel to use for this is probably channel 3 (Ca XIX). YODAT returns the data and index arrays, and (for the BCS) optionally the *dp_sync* array.
2. Exit YODAT (using the −888 option), make a selection of a part of the data from the roadmap by constructing an *ss*[3] vector, and then read this data by re-running YODAT using option −99. Again, YODAT will return the *data* and *index* arrays, and optionally the *dp_sync* array. This non-YODAT selection can be done in various ways, two are:
   (a) use LIST_BDA to list the mode header information from the roadmap, then find the time interval and use the *ss=ss* option to produce the *ss* array that can be used with YODAT.

   IDL> **list_bda,roadmap,start_rec,nrec,ss=ss**

   (b) use SELECT_BDA to plot the roadmap light curve of a selected channel and select the period of interest using the cursors. (Note: there is an option to read the data in with select_bda if required).

   IDL> **.run select_bda**

3.1.2 Accessing a Specific Time Interval Using WBDA

A widget-driven interface that allows the user to examine any BDA file on the data directories (or to display and accumulate the contents of a BDA file already read in using YODAT) is the procedure WBDA. This routine allows the user to interactively work with the data. If the data and index array have not already been read in using YODAT, it is recommended that a specific date be requested when running WBDA; a question concerning the required date is asked if WBDA is called without any input parameters.

IDL> **wbda,index,data**
IDL> **wbda**

3.1.3 Browsing

[3]See the documentation of YODAT for details on the *ss* vector.

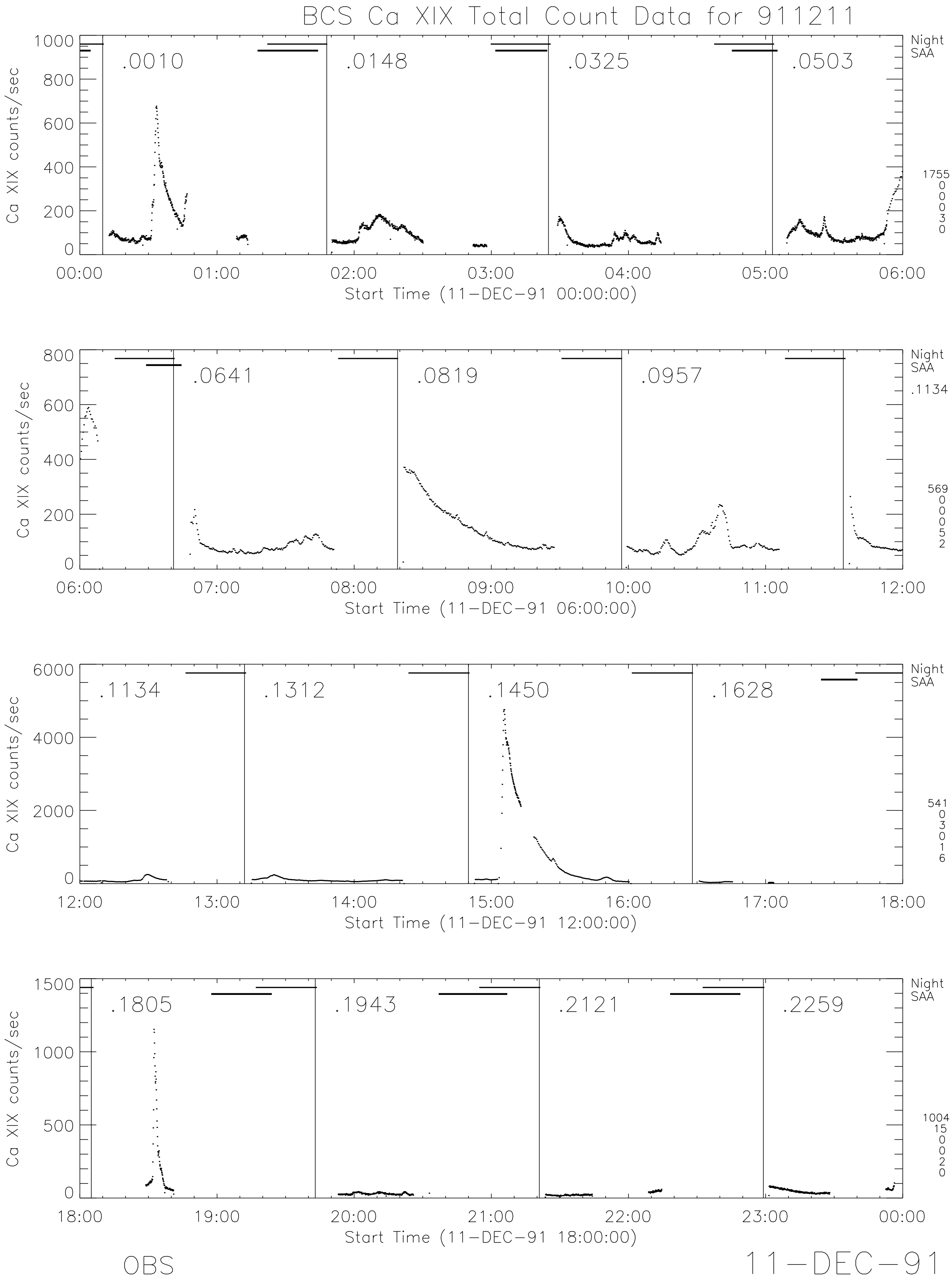

Figure 3.2: bcs_24hr_plot, '11-dec-91'

If you do not know exactly which period you are interested, you can get an idea of what the BCS has seen either by producing light curves using BCS_24HR_PLOT, using WBDA to examine files, or by producing a grey-scale image of spectra against time within a selected file using GS. Fig. 3.2 shows a sample output for the routine BCS_24HR_PLOT.

If you interested only in whether the BCS has seen anything, this can be determined using BCS_24HR_PLOT. This routine works using either the BDA roadmaps or the Observing Log (OBD files in the case of the BCS) and allows plots of a selected day to be made. The start of each (per-orbit) BDA file is marked, as are the times of SAAs and nights. By default, the observing log is used and channel 3 (Ca XIX) is selected. Note: YODAT does NOT need to have been run before BCS_24HR_PLOT.

IDL> **bcs_24hr_plot,'dd-mmm-yy'** [,**chan=n**]
IDL> **bcs_24hr_plot,'dd-mmm-yy',/bda** [,**chan=n**]

The widget-driven interface WBDA allows the user to examine any BDA file on the data directories. This routine allows the user to interactive work with the data, plotting the light curve and also plotting spectra for selected times with the data.

IDL> **wbda**

If you are interested in more detail of what spectra the BCS has observed, use GS — this routine must be run after the data has been read in with YODAT. There is a limit of how much data can be handled at a time (about 900 spectra), and some selection of the required time interval may be needed using the methods described above, but for the per-orbit files, this limit may not be a problem. The advantage of GS is that you can get a good idea of what has been seen in the spectra — a particularly useful tool if you are trawling for data. Remember, all the routines that run from GS (those that have names starting with GS_) require that a time-period be selected first using the cursor routine, GS_CUR.

IDL> **.run gs**
IDL> **.run gs_cur**

Warning: If there is a gap in the time coverage in a data file, the time axis on the GS greyscale plot may not be uniform!!

3.2 Examining Data in the BDA Files

These routines (except where stated) assume that the required spectral data has already been read in (e.g. using YODAT).

The data extracted from the BDA file is in compressed form and is not normalized for the integration time. Also, spectra from all channels are included in the data array and extraction of a channel may be needed before the data can be properly displayed. A single channel can be extracted from the BDA data array, and then decompressed and normalized for integration time using BCS_DSPEC. By default, channel 3 data are returned.

IDL> **chan_data = bcs_dspec(index,data)**

IDL> **chan_data = bcs_dspec(index,data,chan=n** [,**etc.**])

Essentially, this is the same as the following expression, but BCS_DSPEC also does some additional things.

IDL> **chan_data = bcs_norm(index,bcs_decomp(ext_bcschan(index,data,channel)))**

3.2.1 Listing the BDA Data

If you are interested in what modes the BCS was executing, or what the count rate in a particular channel was at a particular time, this can be determined by using LIST_BDA. The procedure will work on either the roadmap or the index structure (which must have been read in with YODAT). By default, the count rate for channel 3 (Ca XIX) is given in the listing.

IDL> **list_bda,roadmap,start_rec,nrec** [,**chan=n, etc.**]
IDL> **list_bda,index,start_rec,nrec** [,**chan=n, etc.**]

3.2.2 Light Curve (only) Plot Routines

These routines only plot light curves. Examples of calls to available routines are:

IDL> **lcbda,index** [,**chans=[2,3,4],psym=psym,/log**]
IDL> **plott_bda,index**
IDL> **plott_bda,roadmap,psym=psym**

3.2.3 Spectra and Light Curve Plot Routines

The light curve, and spectra at times selected from the light curve, may be displayed using WBDA. This is an interactive, Widget-driven program. Notes: (i) WBDA can work either from extracted data, or can extract its data directly from the BDA file; (ii) the spectral plots are versus wavelength.

IDL> **wbda,index,data**

Times can be selected by the cursor from a light curve and summations overplotted together with a light curve using BCS_SURVEY.

IDL> **bcs_survey,index,data** [,**chan=n** [,**/norm , etc**]]

Spectra summed over times selected by the cursor from the GS greyscale can be plotted with GS_SP (which displays four spectra).

IDL> **.run gs_sp**

There are several variants of a routine that plots spectra from several channels and a light curve on a page. In due course, these will be combined into one program. These routines display a light curve and three spectra)

IDL> **.run vg**
IDL> **.run gs_vg**

3.2.4 Spectral (only) Plot Routines

A plot with a spectrum for each channel stacked one above the other may be produced with PLOTS_BDA; the user can page through the modes if more than one is passed in the arrays to the routine.

IDL> **plots_bda,index,data**

Many spectra from the same channel may be plotted on a page as a matrix with the routine BCS_MULTI.

IDL> **bcs_multi,index,data** [,**chan=n, etc.**]

MPLOT allows the user to overplot many spectra from the same channel, optionally displacing each one in the x- and/or y-direction. Note that the BCS channel must have been extracted before calling this routine (see earlier in this section).

IDL> **mplot,xvals,chan_data,x_spacing,y_spacing**

3.2.5 2-d Plots of Spectra Against Time

The evolution of spectra against time can be displayed as a contour map by BCS_CONT, or as a greyscale image using DISP_BDA and GS. Note: The time axis of BCS_CONT is uniform, but those of DISP_BDA and GS are not. Example calls of these routines are:

IDL> **bcs_cont,index,data** [,**chan=n, range=**[**n1,n2**]**, etc.**]
IDL> **disp_bda,index,data**
IDL> **.run gs**

3.2.6 Spectral Movie

After selecting a time interval with YODAT, a movie of the changing spectra for a given channel may be displayed by BCS_SPMOVIE. Note: this program will only run on an X-windows workstation.

IDL> **bcs_spmovie,index,data** [,**chan=n, etc.**]

3.2.7 Overplotting Spectra

Overplot several spectra to show evolution of blue wing and line width with PLOT_REF.

IDL> **plot_ref,index,data,channel,dset_arr** [, **etc.**]

3.2.8 Light Curves of Selected Bins

Plot light curves of different regions of the spectra selected by cursors. Note: With GS_LC, the time period must first have been selected from the GS greyscale display with GS_CUR.

IDL> **.run gs_lc**

3.3 Calibrating and Fitting Spectra (using the BSC software)

BCS data that have had instrumental corrections and calibrations applied to them are referred to as BSC data. BSC stands for BCS Spectrally Calibrated data which has had crystal curvature corrections applied and has been converted to photons cm^{-2} s^{-1}. For quantitative work with BCS spectra the user should always begin by constructing the BSC data structures from the raw data files. This IDL-based software replaces that FORTRAN code that was previously used (see the section on BSD Analysis Software); hopefully this will eliminate some of the difficulties encountered in the installation of the FORTRAN code on the various Unix and VMS platforms. The routines are outlined here, but more information can be found in the BCS part of the reference manual.

The routines work with structures that are compatible with all the other instrument structures in the Yohkoh database. An outline structure (the BSA structure) containing a specification of what data are to be extracted from the BDA file is first created by MK_BSA, and then the BSC structure (and optionally a BSC file) is assembled by the routine MK_BSC.

First run YODAT to extract the index and data arrays from the BDA file for the selected time interval. Make sure to ask for the DP_SYNC data. The routine MK_BSC is then called as follows:

IDL> **mk_bsc,index,data,bsc_index,bsc_data,dp_sync=dp_sync**
IDL> **mk_bsc,index,data,dp_sync=dp_sync,/file**

The second case will write the corrected data to a BSC file. The following routines are used in the assembly of the BSC file:

```
BCS_ACCUM     accumulates BDA data into a BSC structure.
BSC_XCORR     applies curvature corrections to BSC data.
BSC_FLUXCAL   applies flux calibrations to BSC data.
BSC_ERRCAL    computes uncertainties for the BSC count spectra.
```

If the /file option is not used when calling MK_BSC, a BSC file may be written using SAV_BSC. An existing BSC file may be read in with RD_BSC and, after modification, again a new file may be written with SAV_BSC — see the Reference Manual for more details.

3.3.1 Listing the BSC Data

The contents of the BSC file or structure may be examined using LIST_BSC.

IDL> **list_bsc,bsc_index,chan**

3.3.2 Fitting the BSC Data

BCS spectra that are held in a BSC structure my be fit using the routine FIT_BSC.[4] In the following example of a call to the routine, the results of the fit are returned in the parameters *fit_index* and *fit_data*:

IDL> **fit_bsc,bsc_index,bsc_data,fit_index,fit_data,chan=n**
IDL> **fit_bsc,bsc_index,bsc_data,fit_index,fit_data,ss=ss**

All channels and all spectra are fit if no selection is specified. In addition to the *chan* and *ss* keywords, there are many other keyword parameters that specify of the range of the fit, allow the input of starting guesses of electron temperature and other parameters. Note: FIT_BSC is relatively new and is still under development.

3.3.3 Temperature Fit (by Comparison)

Compare a measured spectrum (selected using an *ss* vector from several contained in *bsc_index* and *bsc_data*) to spectra calculated using BCS_SPEC (for channel *chan*, at a temperature of *Te6* MK) using the routine BCS_SPEC_PLOT. An example of the sequence of calls required to do this is given below:

IDL> **bcs_spec_plot,bsc_index,bsc_data,ss=ss**
IDL> **spec=bcs_spec(chan,Te6,wave=wave [,etc])**
IDL> **bcs_spec_plot,wave,spec,/over**

3.3.4 Extraction from the BSC structures

Because the index and data records in a BSC file represent interleaved integrations of individual channels, if the user wishes to extract a single channel from the index structure they must use SEL_BSC:

IDL> **sel_bsc,bsc_index,bsc_data,new_index,new_data,wave,flux,eflux,chan=chan**

The selected index and data structures are returned in *new_index* and *new_data*. The wavelength, flux, and uncertainties in the flux are returned in the arrays *wave, flux* and *eflux*. The following keywords are also accepted: *TSTART, TEND*.

3.3.5 Displaying the BSC Data

The light curve and spectra selected from the light curve (from the BSC file or structure) may be plotted with PLOT_BSC. This program may be used interactively if desired, and can also be used to overplot fitted spectra (if FIT_BSC has been called with the data). If there are

[4] It is also possible at this point to use the BSD suite of FORTRAN-77 and IDL software. By using BSC2BSD, a BSC structure can be converted to a BSD structure (and optionally a BSD file) and following this, any of the routines that normally work off the BSD file (including the programs BSDCAL and BSDFIT) can be used. See section E.1 for details on BSD routines.

IDL> bsc2bsd,bsc_index,bsc_data,index=bsd_index,data=bsd_data

IDL> bsc2bsd,bsc_index,bsc_data,filename=filename

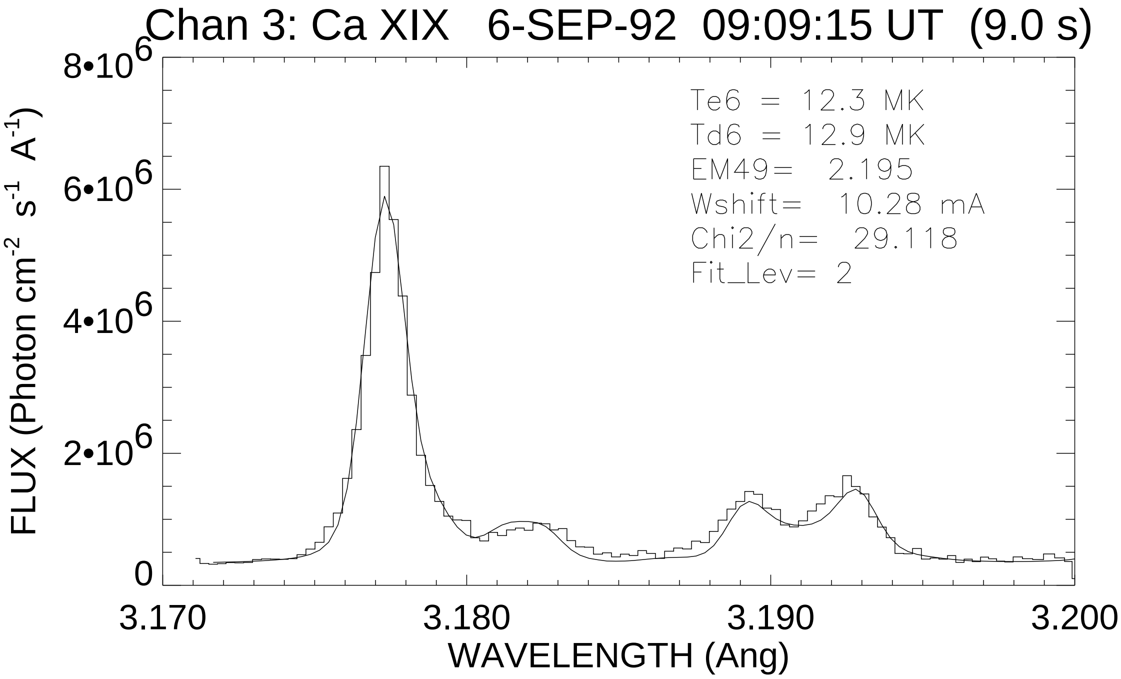

Figure 3.3: An example of a one-component fit of BCS channel 3 data by FIT_BSC.

multiple spectra, PLOT_BSC will provide a widget interface to select the spectrum to plot. An example is shown in Figure 3.3 which plots data which has been fit using FIT_BSC.

IDL> **plot_bsc,bsc_index,bsc_data,chan=n**
IDL> **plot_bsc,bsc_index,bsc_data,chan=n,fit_index,fit_data**

A light curve of FIT_BSC results can be produced with the FIT_BSC_PLOT. An example plot is shown in figure 3.4.

IDL> **fit_bsc_plot, bsc_index**

Note that FIT_BSC_PLOT displays the non-thermal broadening, or turbulent velocity as a function of time. The routine BCS_BROAD is used to make the conversion between line broadening and velocity. FIT_BSC will optionally fit a second component with an upflow velocity as one of the degrees of freedom. A light curve of selected wavelength regions of a set of spectra can be produced with LC_BSC.

IDL> **lc_bsc,bsc_index,chan=n**

A psuedo-map of stacked BSC spectra can be produced with MAP_BSC.

IDL> **map_bsc,bsc_index,chan=n**

3.4 Other Useful BCS Routines

See the routines listed in the Appendix of this User's Guide on page 73.

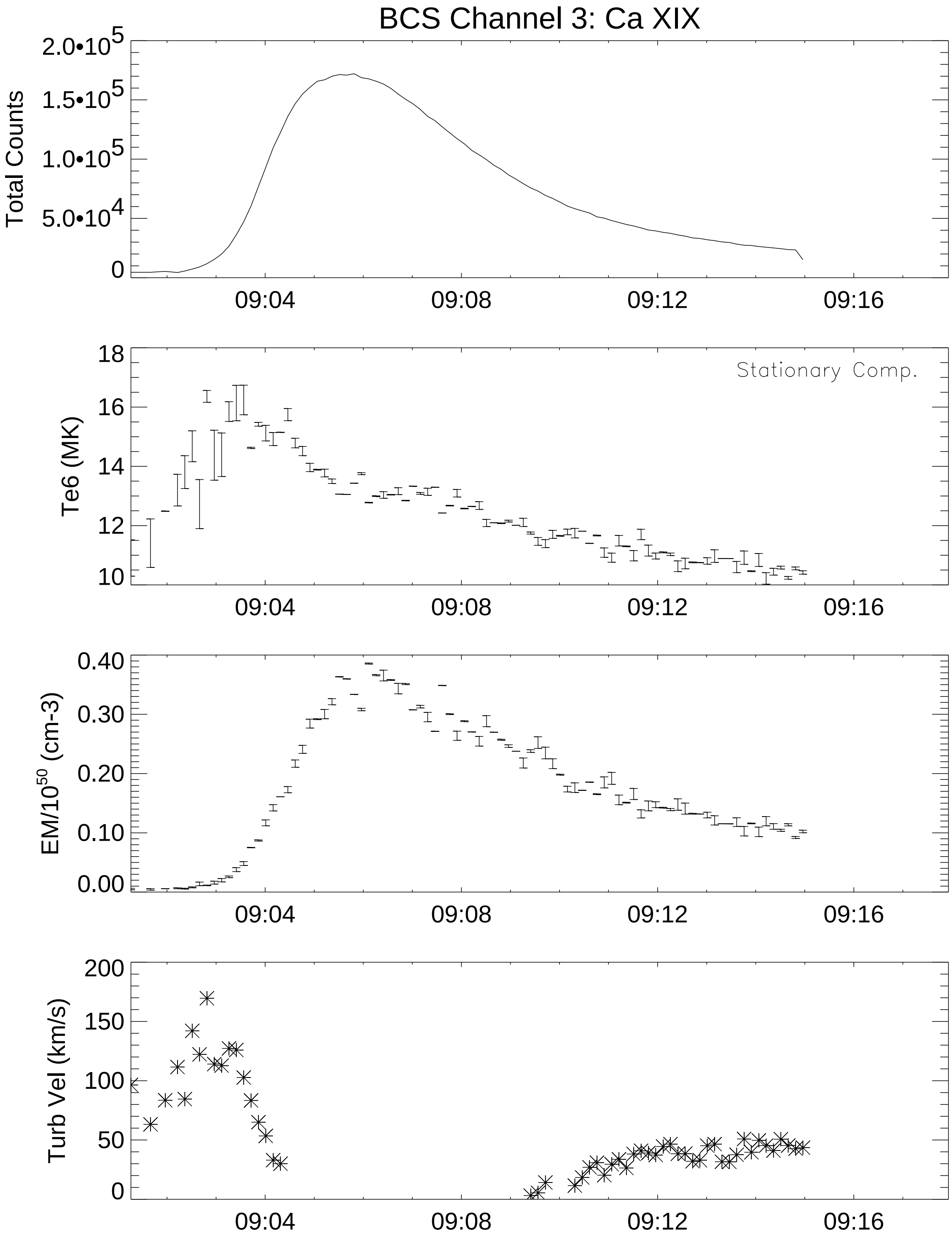

Figure 3.4: Time series plot of results from FIT_BSC using FIT_BSC_PLOT.

4 Hard X-Ray Telescope (HXT)

4.1 Preparations

Use YODAT to extract HXT data from the reformatted database. There are four key variables that are necessary for HXT data analyses described below: *index, data, ssss,* and *infil.* Here *index, ssss* and *infil* have similar contents as those in SXT data. The variable *data* contains compressed count rate data from HXT in the four energy bands. It has the form *data = bytarr(*4,64,4,*m)*, where m is the number of major frames extracted from the reformatted database. The first "4" indicates the four energy bands of HXT, the next "64" indicates the 64 subcollimators (SC's), and the second "4" corresponds to the fact that HXT data appears four times in one major frame.

4.2 Time profiles

4.2.1 How to Plot Light Curves of HXT Data

It is possible to get light curves of the four HXT channels on a single page by using the routine PLOTT_HDA. A sample plot is shown in Figure 4.5. If the HXT index or roadmap has already been read in using YODAT or another routine, then you can use the command:

IDL> **plott_hda, roadmap**

You can also get HXT light curve plots by specifying a date and time. When you use this calling option with PLOTT_HDA it will use the HXT data stored in the Yohkoh observing log files, which is lower time and count resolution than accessing the data file directly. In the first example shown below, the data between 22:00 and 24:00 for 15-Nov-91 are plotted. In the second example, it will plot 30 minutes of data starting at 2-Dec-91 4:30.

IDL> **plott_hda, '15-nov-91 22:00', '15-nov-91 24:00'**
IDL> **plott_hda,'2-dec-91 4:30',30**

4.2.2 Displaying or printing time profiles

After extracting HXT data using YODAT, time profiles of HXT data in the four energy bands with the highest time resolution (e.g. 0.5 s for high-bit rate case) are displayed on the screen by typing:

IDL> **hxt_4chplot,index,data**

with pre-storage of HXT data in DP taken into account (a four or eight second delay in the time tag for the data). Print-outs of HXT time profiles are available by adding an option */psout* to the above command:

IDL> **hxt_4chplot,index,data,/psout**

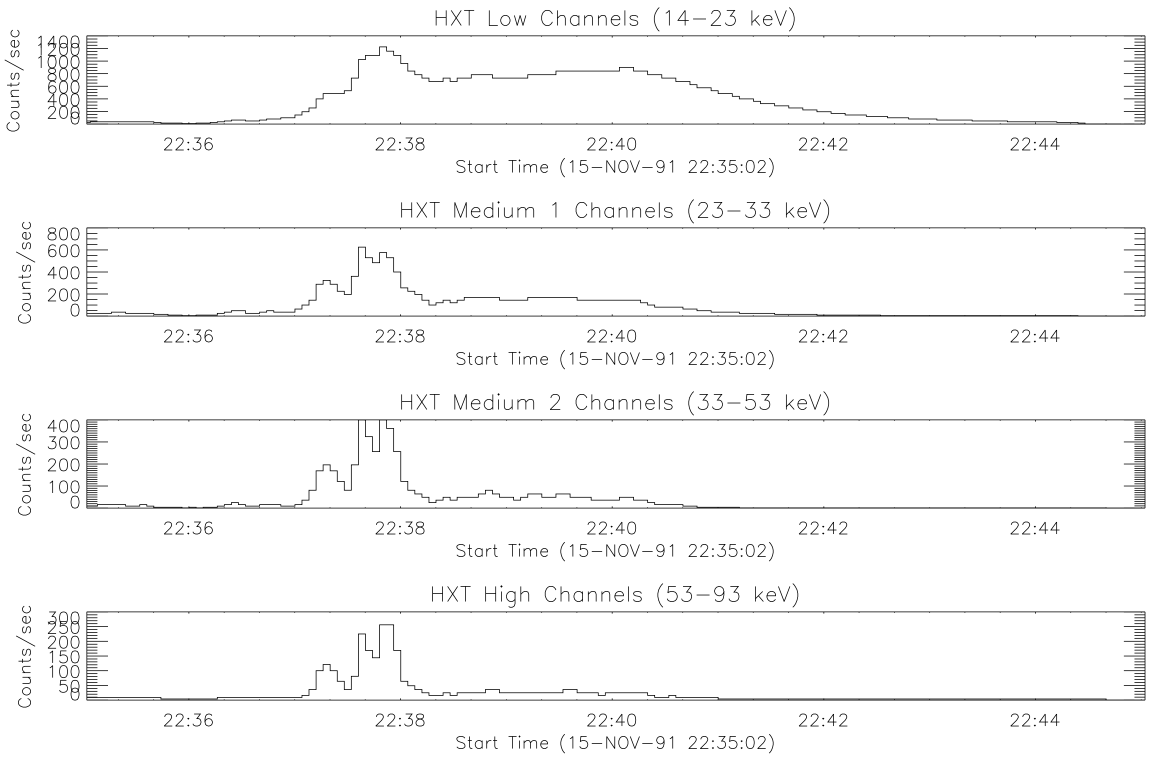

Figure 4.5: plott_hda, '15-nov-91 22:35', 10,psym=10

which produces a PostScript file "idl.ps" containing the time profiles on the current directory. The print-outs are easily obtained on Unix systems (which have the default printer set up properly) by typing

% **lpr idl.ps**

4.2.3 Obtaining count rate data

HXT count rate data (in units of cts/s/SC) in the four energy bands are obtained by the following command:

IDL> **ave = ave_cts(index,data,time=time)**

The count rate data are derived as an average of count rates from the 64 subcollimator outputs. The variable *ave* itself has no time tag in it. Instead, the variable *time* contains necessary information on time tag (pre-storage of HXT data taken into account). Time profiles of count rates in e.g. the M1 band are displayed by:

IDL> **utplot,time,ave(1,*,*),index(0)**

This is just what is done in HXT_4CHPLOT.

4.3 Synthesizing Images from HXT Data

4.3.1 Determining the Flare Location on the Sun (HXT Address)

Before starting HXT image synthesis, we need to know the flare location on the Sun, in the HXT coordinates (whose origin is the HXT optical axis and unit is the HXT pitch (= 126″). East and north are positive). The most convenient way to do this is to run the following program:

IDL> **xy0 = get_hxt_pos('15-Nov-91 22:37')**

Here '15-Nov-91 22:37' is the (approximate) date and time of the flare for which you want to make images. Values contained in *xy0* (= fltarr(2)) are used in the next image synthesis procedure.

The routine GET_HXT_POS will perform the following checks in this order until it finds a match.

1. Read the $DIR_HXT_CAL/hxt_flare_pos.txt file to see if the time is in that list.
2. Use the GOES heliocentric location to determine the address
3. Use an SXT image index to define the address
4. Pass in heliocentric coordinates and convert them
5. Ask the user to specify the HXT address directly

See the Reference Guide for more details on GET_HXT_POS.

4.3.2 Quicklook of Synthesized Image (HXT_QLOOK)

HXT_QLOOK makes a series of HXT quick-look images (i.e. a quick-look movie). This program is much faster than the MEM code, but should be used only for quick-look movies. It is very useful for verifying positions before running a long MEM job. The HXT address of the flare location is needed even for this quick look synthesis, and it helps to use this routine to adjust the HXT address until a proper value is found.

IDL> **hxt_qlook, index, data, movie, hxi_index**
IDL> **hxt_qlook, index, data, movie, hxi_index, patterns**

See the Reference Guide for more details.

4.3.3 HXT_IMG (McTiernan) [*]

Jim McTiernan converted the FORTRAN program which was running on the mainframes to IDL. Currently the criteria used by HXT_IMG to stop the iterations is different from the FORTRAN program. HXT_IMG generally does not iterate as far. It takes a very long time to create a single image (between 1 and 30 minutes depending on the intensity) so we recommend running in batch mode when creating more than a couple of images.

1. The HDA data must be read into the variables *data* and *index*. A common way to do this is to use YODAT.

2. Now start running HXT_IMG by typing

   IDL> **.run hxt_img**

3. Answer 'No' to the first question ("Have the image synthesis parameters been loaded and modulation patterns been calculated?")

4. Now comes the hard part. It is necessary to know the coordinates of the flare in HXT units (126 arcsec units). Please see the routine HXT_IMG_POS to see how to determine the HXT coordinates. For example, the 15-Nov-91 flare the coordinates were (-2.1, -4.8)

5. Answer 'Yes' to the next two questions (use the default parameter files)

6. Select the channel you wish to have an image synthesized for

7. Answer 'YES' for background subtracted

8. Locate a portion of the light curve where the signal is low (just background). Click with the left button on the plot at the start time, click with the right button at the end time, click the middle button to select the portion you have marked.

9. Locate a portion of the light curve where the signal is high where you want to have the HXT image synthesized. Click with the left button on the plot at the start time, click with the right button at the end time, click the middle button to select the portion you have marked.

10. The generation of that image will begin.

11. You will be asked if you want to save the results in an output file. That output HXI file can be read with YODAT or directly with RD_HXI.

12. When it is finished, the variables *index_out* and *data_out* will hold the results.

4.3.4 AUTO_HXI [*]

A driver for HXT_IMG was written which will automatically figure out how to integrate the HXT signal for each of the channels to accumulate 200 counts. The routine is very sensitive to the location of the flare and will not converge if the location is not correct. The flare is specified by selecting a time range and this means that the data files have to be in the directories that are returned by the DATA_PATHS routine (it figures out which files to use from the input times).

It figures out the location of the flare by (1) reading the GOES event log for the flare and converting the heliocentric coordinates, (2) converting the location of the SXT partial-frame images into HXT coordinates, or (3) to pass the location of the flare in the call to AUTO_HXI. The default is to do all channels, to use the GOES event location, to not subtract background, and to use a variable integration time to achieve 200 counts.

Some sample calls are:

IDL> **auto_hxi, sttim, entim**
IDL> **auto_hxi, sttim, entim, chan=0**
IDL> **auto_hxi, sttim, entim, chan=0, /sxt_pfi**
IDL> **auto_hxi, sttim, entim, acc_cnts=[100,200,300,200]**

IDL> **auto_hxi, sttim, entim, loc=[-2.1,-4.8]**
IDL> **auto_hxi, sttim, entim, outdir='/yd8/scratch/morrison'**

where *sttim* is the start date/time in any of the three standard formats (described in the Reference Guide), and *entim* is the end time. At this time, the routine performs no background subtraction before image synthesis.

4.4 Using HXT for Broad-Band Hard X-Ray Spectrophotometry

HXT has a large effective area. Accordingly, its four energy channels can be used for ordinary spectrophotometry via direct summation of the rates from the 64 independent detectors, independently of any application of these data to imaging. The data are well–calibrated and are normally free of any effects of pulse pile–up, owing to the small effective areas of the individual detectors. The four channels can be used to fit power–law or thermal spectra. The presence of spatial structure in the image affects the precision of the total–count photometry, but because there are 64 detectors, this effect is probably small.

4.4.1 Determining Temperatures

The program HXT_TEEM can be used to determine thermal fits to the data. It returns pairwise fits of the four channels, i.e., three independent values for temperature and emission measure. The instrument is sensitive to temperatures as low as 15 MK, but at the time of writing no systematic comparison of these temperatures with, say, those of the GOES photometers, had been carried out. The HXT_TEEM program has an interactive background determination feature that may make it easier to generate consistent temperature fits (Med1/Low and Med2/Med1) in the presence of the temporal variations of the background rates.

4.5 How to Fit HXT Spectra: HXTBOX_FSP (McTiernan) [*]

A general discussion of the McTiernan spectral fitting programs is given on page 59 in the description of fitting WBS spectra.

A routine which will do spatially resolved HXT spectra is also available, called HXTBOX_FSP. To use this, the input index and data structures must come from an HXI file, instead of index and data from HDA files used by HXT_FSP. The region for the spectral fit is chosen using the routine LCUR_IMAGE, before the time interval is chosen. Here are some sample calls.

IDL> **hxtbox_fsp, index, data, fit_pars**
IDL> **hxtbox_fsp, index, data, fit_pars, boxq=boxq**
IDL> **hxtbox_fsp, index, data, countfile = 'test.dat'**
IDL> **hxtbox_fsp, index, data, fit_pars, same_bx=boxq**
IDL> **hxtbox_fsp, index, data, fit_pars, boxq=boxq, /same_bx**

where *boxq* is an array containing the subscripts of the chosen regions, (corresponding to the keyword *marks* in LCUR_IMAGE). If *same_bx* is set, the routine will use the array of subscripts contained in *boxq* (or if *boxq* is not set, than the array in *same_bx* is used). Note that the final two examples will accomplish the same thing. The output structure *fit_pars* is an array of (n_intervals × n_boxes).

4.6 Spectral analysis with HXT (Sakao)

As HXT has four energy bands between 14 keV and 93 keV, we can estimate incident hard X-ray spectra under certain assumptions about the spectral form (such as power law or thermal Bremsstrahlung emission) using count data pairs from different energy bands.

4.6.1 Background selection

Before starting spectral analysis, prepare background data using the following command:

IDL> **bkgd = mk_hxtdata(index,data,ssss,infil,channel=0)**

In the above example, the time profile in the L band (channel=0) is displayed on the screen. Choose the background data interval in the same way as described in E.2 (4). Note that BGD data in all the four energy bands, which have the same start and end times specified from the L band time profile, are stored in the structure variable *bkgd*. The variable *bkgd* has the same structure as the one in E.2 (4); if you have already have the BGD data in your IDL memory, you need not run the above MK_HXTDATA.

4.6.2 Power Law Spectrum

If we assume an incident X-ray energy spectrum of a power law form: I(E) = $A * (E/20keV)^G$ (photons/s/cm^2/keV), we can estimate hard X-ray flux at 20 keV (A) and photon index (G) by the following command:

IDL> **p = hxt_powerlaw(index,data,1,bkgd,time=time)**

which provides us with time profiles of A *(p(0,*))* and G *(p(1,*))* with the highest temporal resolution (e.g. 0.5 s in high-bit rate). The number '1' in the above command specifies the lower band of a pair of two adjacent energy bands from which A and G are calculated; you can specify '0' (M1/L), '1' (M2/M1), or '2' (H/M2). The variable *bkgd* is the BGD data obtained in section 4.6.1. For example, a time profile of photon index is displayed as follows:

IDL> **utplot,time,p(0,*),index(0)**

Values of *p* for each major frame are obtained by adding an option */sf* to the arguments of HXT_POWERLAW:

IDL> **p = hxt_powerlaw(index,data,1,bkgd,/sf,time=time)**

which is useful when the flare is not very intense and you need to accumulate HXT data to increase count statistics.

4.6.3 Thermal spectrum

If we assume that observed hard X-rays are emitted by single-temperature thermal Bremsstrahlung, then temperature (T (K)) and emission measure ($EM \times 10^{48}$ cm^{-3}) of the thermal plasma is obtained by

IDL> **t = hxt_thermal(index,data,0,bkgd,time=time)**

The variables *t(0,*)* and *t(1,*)* provide T and EM (in units of 10^{48} cm^{-3}), respectively, with the highest temporal resolution. The arguments '0' and *bkgd* have the same meaning as in section 4.6.2. Option */sf* is also available for this program.

4.7 Useful Routines for HXT

See the routines listed in the Appendix of this User's Guide on page 75.

5 Soft X-Ray Telescope (SXT)

5.1 How to View SXT Images

Once the SXT data are read into *index* and *data* using YODAT or RD_SDA, it is very simple to display images. A general purpose routine called STEPPER is the most common method used.

STEPPER takes a data cube (*data*) and allows a user to step through the images and to display them as a movie. If the *index* is passed into STEPPER (as shown in 2 of the examples below) it is possible to chose an option to display light curves. Some sample calls are:

IDL> **stepper, data**
IDL> **stepper, data, info_array**
IDL> **stepper, index, data, xsiz=512**
IDL> **stepper, index, data, info_array**

where *info_array* and *xsiz* are optional parameters. *xsiz* indicates the size of the displayed image you want (must be an integer multiple of the original array). It is a window driven program.

Some other keyword options are *start=start* to specify the starting image to display; */noscale* to not perform automatic scaling on each individual image (uses TV instead of TVSCL); *subscript=ss* to specify the image numbers to display (default is to display all images in the data cube); */filter_panels* to display each unique filter in a different location on the window; */sequence_panels* to display each unique image sequence in a different location on the window.

When STEPPER begins, the following menu is displayed.

```
-------- STEPPER Options --------
There are 259 images in the array
There are 259 images selected
Enter "b" to step backwards              "h" to call HARDCOPY
      "s" to select new start index      "z" for ZOOM
      "c" to call LOADCT                 "x" for XLOADCT
      "p" to call the IDL PROFILES routine  "l" to call light curve routine
      "g" to overlay SXT solar grid      "o" for observing region location
      "q" to quit                        "m" for movie mode
      "?" to display this help menu      "anything else" to step forward
```

The user only needs to press the key to select the option (no <CR> required). It is possible to overlay a grid showing the solar limb and latitude/longitude lines for either full frame or partial frame images. The “l” option calls LCUR_IMAGE and the user should read section 5.3 for more details.

The routine XSTEPPER is a widget driven routine which operates in a manner similar to STEPPER, but has some other capabilities. A sample call is:

IDL> **xstepper, data, info_array**

It is possible to load a data cube into a widget to be “animated” using the routine XMOVIE (which is simply an interface for IDL’s XINTERANIMATE). The speed which the frames can be displayed is quite fast. There are some memory restrictions which do not allow large data cubes. Some sample calls are:

IDL> **xmovie, data**
IDL> **xmovie, data, ss=ss** ; only displaying images listed in ss

A final popular routine for displaying images is the routine XY_RASTER. This routine will make a raster pattern of a time series of images to display time evolution. The images start at upper left and go to lower right, and the time is optionally written at the bottom of each image. Some sample calls are:

IDL> **xy_raster, index, data**
IDL> **xy_raster, index, data, 3, ss=ss**

The second example blows the image up to three times its input size and only displays the images listed in the array *ss*.

5.2 Preparing the SXT Data for Analysis (SXT_PREP)

It is the desire of the SXT team to have the routine SXT_PREP used for most of the preparation of the SXT data. The SXT_PREP routine performs a variety of tasks, some of which are described in the following list.

1. Flag the saturated pixels
2. Calculate the decompression error uncertainties for each pixel
3. Perform standard corrections to the data
   - Decompress the data
   - Subtract the dark current
   - Subtract the pin-hole visible stray light
4. Align the images and build the observing region (or mosaic of PFI images). The routine can handle PFI, FFI, or a mix of images. It can extract subportions out of FFI images. The alignment can be done
   - in heliocentric coordinates
   - relative to sun center
   - relative to CCD pixels
5. Optionally fill in the horizontal gaps in the observing region (or)
6. Optionally exposure normalize to a one second exposure

Some important elements of SXT_PREP are:

- The default when registering image is to convert all images to full resolution. This can be overridden by the optional keyword *outres=outres*.
- The signal is rescaled if the resolution is changed. This occurs whether the exposure normalize option is selected or not. The signal is divided by 4 when going from HR to FR, or from QR to HR (and divided by 16 when going from QR to FR).
- The optical images are shifted to be aligned with the X-ray images starting with SXT_PREP version 1.20 (4-Jan-94). The shift is less than 1.5 FR pixels.

- The S/C roll and SXT roll offset can be corrected so that solar north is straight up by using the keyword option */roll*
- The */sfd* keyword should be set when using SXT_PREP on SFD files
- The units out of SXT_PREP are data number (DN) unless the */normalize* options is used, in which case they are DN/s.

The steps to using SXT_PREP can be somewhat involved. The basic steps are described below, but the routine XSXT_PREP is an easy way for a user to specify what processing should be performed, and XSXT_PREP is described at the end of this section.

1. Select the files which have the data of interest. You can use YODAT to do this, or some other method.

2. Select the images that you wish to process. Again, YODAT can do this, or you can use SSWHERE or another method for selecting the images. If you are not using YODAT, then one option is to read the roadmaps of the files, and pass the roadmap to SSWHERE. The following example assumes the list of file names (as a string vector) is in the variable *infil.*

   IDL> **rd_roadmap, infil, rmap**
   IDL> **ss = sswhere(rmap)**

3. To register images, it might be necessary to determine the coordinates for the center of the aligned images. Some methods for specifying the coordinates are:

   - Allow the routine to find the center of all images, and take the average. This works fine if the selection has been done carefully and the images selected are from one location. This also works fine if you are registering FFI images.
   - Specify the heliocentric coordinates specifically
     helio=[-20.2,50.3], helio_date='8-dec-93 4:43'
   - specify the sun coordinates specifically
     suncoord=[400,600]
   - Use a PFI image as a reference image
     image_ref = pfi_index(40)
   - Use a PFI image as a reference image and select heliocentric tracking.
     image_ref = pfi_index(40), /helio

   If you only selected PFI images, and the PFI images you selected are all for the same active region, then you do not need to specify the coordinates of an aligned image (it will automatically derive the optimal coordinates by averaging all PFI center locations). If they are not for the same region, then you should either modify your selection so that it only includes one location on the sun, or select a image which all of the data should be aligned to and use the *ref_image=index(ialign)* option, or you can specify the absolute coordinates (the procedure for establishing those coordinates is described below).

   If you have mixed PFI and FFI images, then you can specify one of the partial frame images, and then all images will be registered to that image, and the appropriate portion of the FFI image will be extracted and aligned.

If you selected only FFI images, but you want to extract a portion to make a registered movie, then you will need to determine the coordinates for the registered image. A simple method to do this is to use SXT_GRID. If you want to simply remove the S/C jitter and the SXT pointing changes (register relative to sun center), then display a full frame image and call SXT_GRID. An example is:

```
IDL> tvscl, data(*,*,5)
IDL> sxt_grid, index(5), /read_out, last=loc, /angle
```

As you move the cursor around, you will see that the values in the small window are changing. When the cursor is located at the location where you want the center of your image, press the right button. If you want to register images and track on a heliocentric coordinate, then you can use SXT_GRID again to get those coordinates. An example is:

```
IDL> tvscl, data(*,*,5)
IDL> sxt_grid, index(5), /read_out, last=loc
```

It is necessary to specify the size of the output image when extracting a partial frame out of a full frame.

4. Run SXT_PREP. If you are reading the data and passing the index and data to SXT_PREP, then the observing regions should not be assembled (answer "No" to this question in YODAT). This is important because jitter and drift can cause pointing changes between the different exposures that comprise of one PFI observing region.

   It is possible to pass *index* and *data* into SXT_PREP, or to pass the file name array (*infil*) and the data sets to extract (*ss*). The results can be returned in the third and fourth parameters (*index2* and *data2*) or they can be written directly to an output file (*outfil=outfil*).

   Below are some examples of how to call SXT_PREP.

```
IDL> sxt_prep, infil, ss, index2, data2, /reg
IDL> sxt_prep, infil, ss, index2, data2, uncert, satpix, /reg
IDL> sxt_prep, infil, ss, index2, data2, helio=loc, date_helio=date_helio
IDL> sxt_prep, infil, ss, index2, data2, helio=[-9.8,-20.3], $
IDL>                date_helio='14-JUN-92 02:37:41')
IDL> sxt_prep, index, data, index2, data2, /reg, ref_image=index(55), outsiz=100
IDL> sxt_prep, index, data, index2, data2, /reg, ref_image=index(55), /helio
IDL> sxt_prep, infil, ss, index2, data2, /reg, /sfd, outres=1, $
IDL>                outsiz=[100,200], suncoord=suncoord
IDL> sxt_prep, infil, ss, /reg, outfil=outfil, /dc_interpolate, /normalize
```

XSXT_PREP provides a graphical user interface to SXT_PREP. The XSXT_PREP routine simplifies the access to the many SXT_PREP options and varied calling sequences. In addition, XSXT_PREP provides a front end to some additional data selection and filtering tools which may be used to fine tune the data set and allows graphical definition of all SXT_PREP input parameters. The following graphical tools are provided:

- selection of alignment coordinates and definition of output data size by display and mouse selection [optionally overlay on SXT Full Frame Image].
- filtering and re-filtering of input data (can spawn call to SSWHERE).
- ability to step through input data to select alignment reference image.
- given a time, user may graphically select AR from FFI (supplied or from SFM data) to identify FFI and PFI images which contain that AR – process if data are available online or setup background job.
- selection and display of all SXT_PREP parameters and display associated SXT_PREP calling sequence.
- spawn SXT_PREP on request or spawn a background task (file output).
- maintain log of SXT_PREP input parameters and calling sequence so that work can be repeated or reviewed at a later date.

The combination of graphical interface and online help make this a good routine to learn through use. It is a main level routine. For starters, try the following calling sequence:

IDL> **.run yodat** ; select SXT data cube to process
IDL> **.run xsxt_prep** ; play around

5.3 Making a Light Curve with SXT Images (LCUR_IMAGE)

It is possible to display a light curve of the SXT image intensities using the routine LCUR_IMAGE. This routine is available from within STEPPER so the user can select the option by entering "l".

Some examples for calling LCUR_IMAGE directly are:

IDL> **lcur_image, index, data**
IDL> **lcur_image, index, data, lcur**
IDL> **lcur_image, index, data, lcur, uncert, satpix**
IDL> **lcur_image, index, data, xsiz=512**
IDL> **lcur_image, index, data, xsiz=512, /poly**
IDL> **lcur_image, index, data, xsiz=512, /nonorm, subscript=subscript**

index and *data* are input, and *lcur* is the output – an array with the image intensities. *uncert* and satpix are optional inputs which are the uncertainty and saturated pixel map which is returned from SXT_PREP. If these two variables are passed, the routine will mark points which are saturated or have high uncertainty. The *xsiz* option will blow up the image to be 512 pixel wide (an 512 pixels tall if it is square) so that the selection of a region is simple.

```
-------- LCUR_IMAGE Options --------
Enter "p" to mark and plot light curves (loops until exited)
      "m" to overplot all of the selected light curves
      "r" to display the regions selected ("R" to make a hardcopy)
      "s" to change the method to mark regions to plot.  Currently: RECT
```

```
      "d" to call "stepper" to refresh/display a new image
      "z" to reset (zero out) all of the regions saved so far
      "o" to enable/disable over plotting of new curves. Currently: DISABLED
      "6" to enable/disable making 6 window plots.  Currently:      DISABLED
      "h" to produce a hard copy of the most recent light curve
      "H" to produce a hard copy of all selected light curves
      "q" to exit LCUR_IMAGE ("x" will work too)
Enter your option (no return required).  It is case sensitive.
```

The user can select several different areas on the image and overplot the light curves for each region. The light curves and regions are labeled so that they can be differentiated. Hardcopy options exist, along with several other capabilities. See the documentation header for the program and the Reference Guide for more details.

5.4 Making a Mission Long SXT Light Curve

The following command will display a light curve which uses the integrated total of the full disk SFD images. There is one plot for thin Al and another one for AlMg.

IDL> **.run plot_sxl**

The SXL database must be available on your system for this program to work.

5.5 Jitter and Assembly of PFIs from Multiple Exposures

See the section "Preparing the SXT Data for Analysis (SXT_PREP)" on page 35 for a description of how to align images.

5.6 Displaying SXT and Ground-Based Synoptic Maps

The SXT synoptic maps are constructed from strips extracted from full disk desaturated SXT images made using the thin aluminum filter. Each map depicts a single Carrington rotation. There are three sets of maps, corresponding to strips centered on the central meridian of the solar disk, 45 E longitude, and 45 W longitude. The widths of the strips are determined by the time differences between images. For a given map, as many full disk images are used as are available provided the time difference between contiguous images exceeds 1.5 hours. Due to occasional gaps in the coverage of full disk imagery, some of the synoptic maps show corresponding gaps. The maps constructed from strips centered at the $\pm$ degree longitudes are distorted at high latitudes because vertical strips are extracted which do not follow the longitude line. The images from which the strips are extracted have been registered to remove the effects of spacecraft pointing changes. Each strip is shifted along the vertical axis so that heliographic latitude 0 is always centered in the assembled map. However, projection effects in the vertical axis have not been removed. This means that the latitude scale is not linear but reflects the cosine projection effect of a spherical structure (the corona) projected on to a flat image.

The following are some examples of how to display the synoptic images which are in the Yohkoh database. The first example just displays a single synoptic map for the central meridian for carrington rotation number 1860. The second example displays three SXT synoptic maps, central, +45 and −45. The third example has a single synoptic image which is the Kitt Peak magnetogram for carrington rotation number *carrington_rot_num*. The fourth example shows SXT central meridian and the Kitt Peak magnetogram and He 10830 synoptic maps.

IDL> **disp_synop, 1860**
IDL> **disp_synop, 1860, /allsxt**
IDL> **disp_synop, carrington_rot_num, /kpmag**
IDL> **disp_synop, carrington_rot_num, /gbo**

An example of how to read one of the SXT central meridian synoptic map images is shown below, along with an example of how to read a Kitt Peak magnetogram synoptic map for carrington rotation number 1862.

IDL> **rd_xda, '$DIR_GEN_SYNOPTIC/ssc_cr1862a.01', -1, index, data**
IDL> **img = rfits('$DIR_GEN_SYNOPTIC/gkm_cr1862a.01', header=header)**

5.7 Alignment of Optical and X-Ray Images

When SXT_PREP is called with the */register* switch, the optical images are automatically aligned to the X-ray images. See the Instrument Guide and the routine GT_SXT_AXIS in the Reference Guide for more details.

5.8 Aligning Ground-Based Images with SXT

The software tools for aligning SXT images and other images is in a very early stage of development. The routine COAL_IMAGE along with some other tools can be used in a fairly manual method to co-align a pair of images.

The alignment between SXT images and other images can be performed by the following steps. The SXT images do not require steps 1 or 2 since the absolute pointing and roll are known for SXT images.

1. For the non-SXT image, determine the distance that the lower left pixel is from the center of the sun in pixel units (see Fig. 5.6). This can be accomplished in one of several methods.
   - If the image contains enough of the solar limb, then a fitting routine can be used to determine the coordinates of the center of the sun.
   - The spacecraft or observation site might know the absolute coordinates of the image, so the pixel address of the lower left corner can be easily derived.
   - Solar features can be manually co-registered with a full disk image, and the coordinates for the center of the sun can be derived from that.
2. Determine the roll of the image relative to solar north. Again there are several techniques to determine this value
   - The most common method is simple knowledge based on spacecraft attitude, star trackers, or ephemeris information.
   - It is possible to manually co-register the image with another image which has a known roll.
3. Run COAL_IMAGE to register one image to another image. The program will correct for:

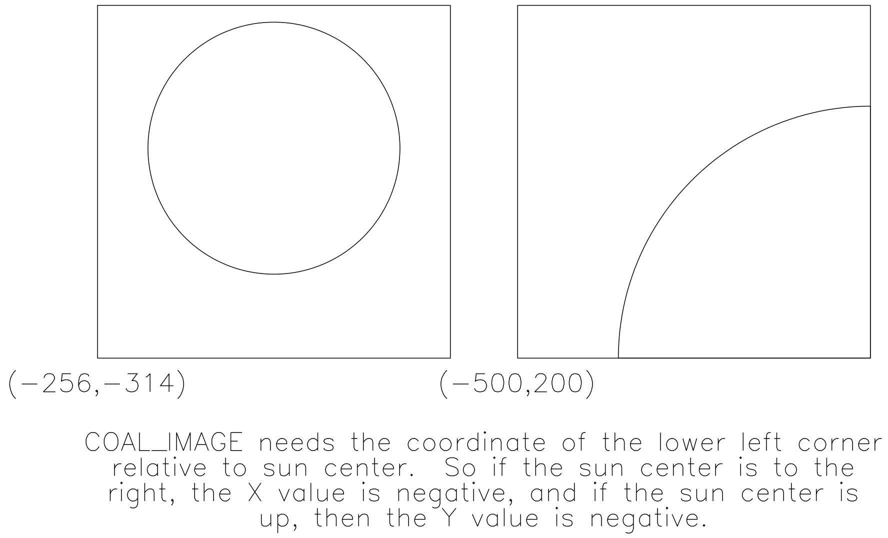

Figure 5.6: Coordinate definition for COAL_IMAGE

- translational offsets
- roll
- different pixel resolutions

5.8.1 Determining the Coordinates Using FIT_LIMB

FIT_LIMB is a modified version of SXT_CENTER. It currently requires that you know the pixel size of the image which is being processed (although the pixel size can be derived by calling FIT_LIMB multiple times with different pixel sizes until the diameter of the solar limb outline matches the image). It allows fits to a partial solar limb. A sample call to FIT_LIMB is

IDL> **fit_limb,img,x0,y0,/int,pixel_size=3.97**

where it assumes that you have the image to be fit in the variable *img*. The */int* enables the interactive selection of the initial guess of the fit. It is very important that the first guess be very close. *pixel_size=3.97* specifies the pixel size in arcseconds. The E/W and N/S sun center coordinates are returned in *x0* and *y0*. These values must be inverted since we want the corner coordinate relative to sun center.

IDL> **corner2 = [-x0, -y0]**

5.8.2 Determining the Coordinates Using DSK_LOCB

If you have a full disk image with well a defined limb, you can use the routine DSK_LOCB. A sample call is:

```
IDL> dsk_locb,img,horzcnt=x0,vertcnt=y0,horzrad=rz,vertrad=ry
IDL> corner2 = [-x0, -y0]
```

See the Reference Guide or the documentation header for more details on DSK_LOCB.

5.8.3 Determining the Coordinates and/or Roll by Feature Correlation

It is also possible to determine the coordinates of the corner by matching features to another image for which the sun center location is know. This would be necessary for images that do not include the solar limb. No standard piece of software was available when this guide was written so send a request for information to the e-mail address shown in the preface of this guide.

A crude routine called FLASH_COMP is available to take two images and flash them using XINTERANIMATE. The images are scaled independently and do not need to be put into an data cube. A sample call is

```
IDL> flash_comp, img1, img2
```

5.8.4 Registering One Image to Another Image

Once the image coordinates are known for the non-SXT image(s), you can call the routine COAL_IMAGE which will co-align one image to another. The following assumptions are made for the sample calling sequences below.

```
IDL> pix2 = 3.97                              ; pixel size of non-SXT image
IDL> pix1 = gt_pix_size(index)                ; SXT image pixel size
IDL> corner1 = gt_corner(index, /from_sc)     ; the SXT corner
```

To convert the non-SXT image to the resolution and align it to the SXT image you would use the first command below, and you can optionally compare the newly aligned images with the second command.

```
IDL> out1 = coal_image(img, corner2, corner1, [256,256], mag=pix2/pix1)
IDL> flash_comp, out1, data
```

The following are the commands to convert the SXT image to the resolution of the non-SXT image, and align it to the non-SXT image:

```
IDL> out2 = coal_image(data, index, corner2, [512,512], mag=pix1/pix2)
IDL> flash_comp, out2, img
```

There is an optional keyword *angle* which will roll one image relative to the other if the angle is known. See the description of COAL_IMAGE in the reference guide and the program header for more information.

5.9 How to Correct for Vignetting

The SXT X-ray effective area varies across the 42×42 arcmin field-of-view. The shape of the vignette function can be examined with the IDL program HELP_VIGNETTE. At 21 arcmin from the optical axis, the low-energy response (< 1.49 keV) falls to approximately 75% and the high-energy response (2.98 keV) falls to approximately 50% of the on-axis value (see the Instrument Guide for more details). Because of this, some care should be taken when, for example, making temperature analyses with thin and thick filters of flares that occur near the solar limb.

There are no fully approved routines to remove or correct for the SXT vignette function. There are, however, two preliminary routines. SXT_VIGNETTE will compute the vignette function as it is currently understood. The form and parameters of the shape may be changed as additional data and analysis results become available. The routine SXT_OFF_AXIS will correct for the vignette function by calling SXT_VIGNETTE. Users are warned that these routines are preliminary and may not be the most appropriate technique to use, depending on the scientific application.

5.10 Determining the Pointing Coordinates for SXT PFI

5.10.1 Simple Plots of Location Relative to the Solar Limb

It is possible to make a simple display of the sun and to overlay the box showing the field of view size and location by using PLOT_FOV if you have the index, roadmap or observing log structures. A sample call is:

IDL> **plot_fov, index, /box**

If you don't have a PFI structure, then you can use the routine PFI_LOC. It can accept a date and time and will display a simple plot of the sun and the SXT PFI location (PFI_LOC simply reads the observing log and then calls PLOT_FOV). Some sample calls are

IDL> **pfi_loc, '23-jun-93 1:00'**
IDL> **pfi_loc, timstr, hours=2**

The second example will display all of the different PFI locations for the time stored in the string variable *timstr* $\pm$ 2 hours.

5.10.2 Overlaying the PFI Location on an Image (PFI_LOC)

If you have displayed a full frame image and you wish to see where the SXT PFI region was located, you can use the routine PFI_LOC. Assuming that the index of the image which is displayed is *index*, then a sample calling sequence is:

IDL> **pfi_loc, index**

It is possible to use the *hours* keyword option to specify the number of hours plus and minus the full frame time for which you wish to have PFI locations displayed. When using STEPPER, the "o" option will call PFI_LOC automatically for you, taking into account any rebinning which was performed on the image.

5.10.3 Listing of the Images and Heliocentric Locations

See section 2.1.4 for some details on routines for listing the SXT partial frame pointing information.

5.10.4 Extracting the PFI Location (GT_CENTER)

The routine GT_CENTER will allow a user to determine the coordinates of the center of a PFI image. The output from this routine can be the location of the center of the field of view in (a) full resolution IDL pixels coordinates [5], (b) heliocentric, (c) the angle E/W and N/S from the center of the sun in arcseconds, or (d) the HXT equivalent pixel coordinates. If the user has the index available in the variable *index*, then he can use the following commands (these commands will also work for the roadmap structure)

IDL> **pix = gt_center(index)**
IDL> **helio = gt_center(index, /helio)**
IDL> **helio = gt_center(index, /helio, /string)**
IDL> **helio = gt_center(index, /helio, /cmd)**
IDL> **ang = gt_center(index)**

The default is to read the ATT S/C pointing database for the true sun center location. It is possible to use the commanded S/C pointing information by using the /CMD option, which is much faster and only of slightly reduced accuracy. The */STRING* option will print the heliocentric coordinates in the format "N23E45".

5.11 Filter Responses

See the Instrument Guide for details on the SXT telescope and filter responses.

5.12 Determining Temperatures for SXT Images (SXT_TEEM)

The SXT X-ray filters were chosen to provide a different effective area for each filter as a function of wavelength (Fig. 5.7). The SXT response functions are computed by folding a thermal spectrum with the SXT effective area through each filter. The result is shown in Fig. 5.8, for the Mewe 1985 spectrum using coronal abundances. The ratios of the fluxes observed through the various filters is temperature sensitive. Three filter ratio combinations are shown in Fig. 5.9. These ratios are actually "mean" temperatures and are only correct for the case of an isothermal plasma.

[5]Not SXT **CCD** pixel coordinate system, which is reversed E/W from IDL pixel coordinate system. See the Appendix of the Instrument Guide

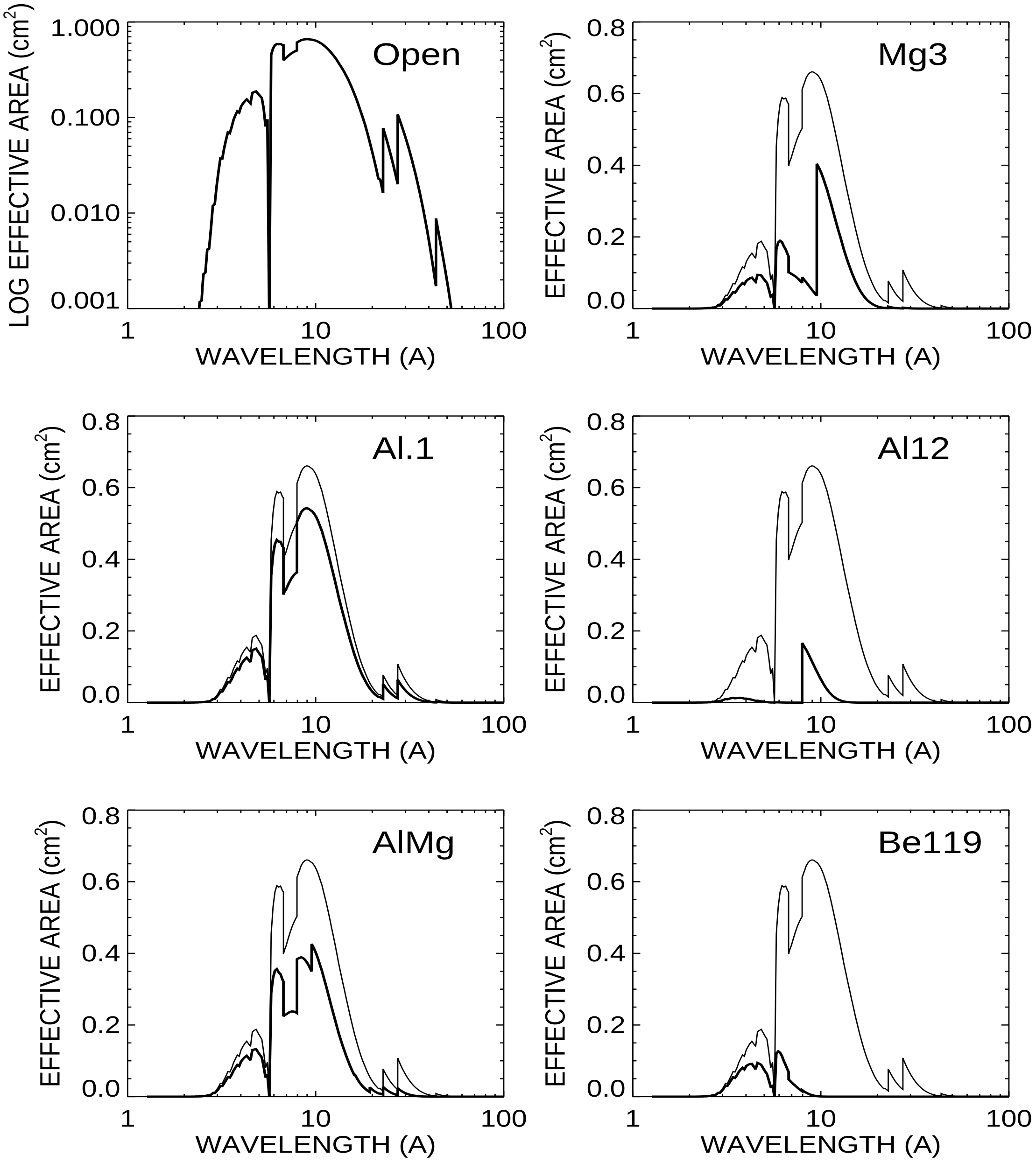

Figure 5.7: The effective area of the SXT: The thick curve shows gives the effective area for the indicated filter. The thin curve is the open filter case (no analysis filter) effective area over-plotted for comparison.

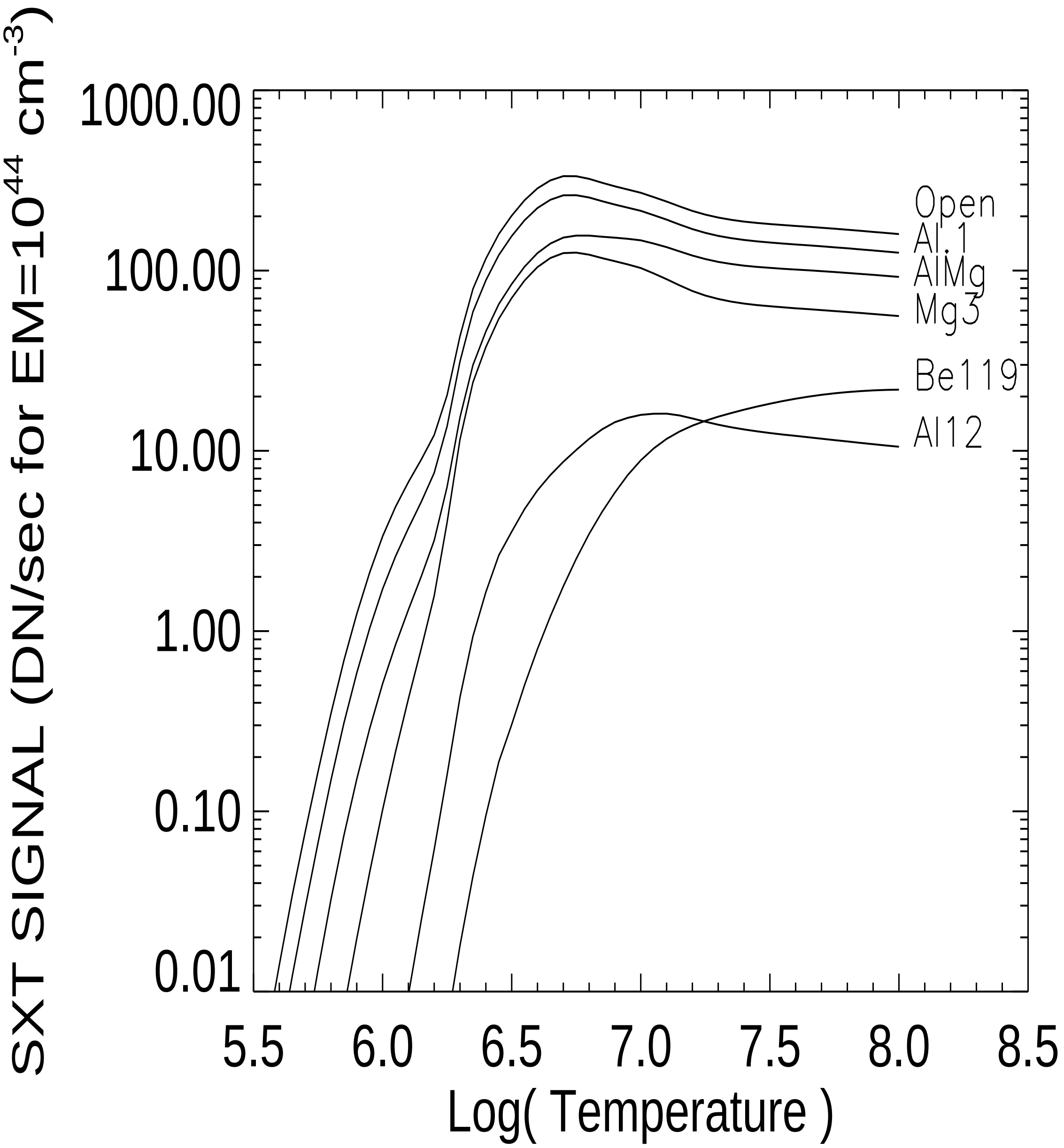

Figure 5.8: The total SXT signal as a function of $\log_{10}(T_e)$ for the open filter and the analysis filters as indicated. NOTE: Be and Al12 are mistakenly reversed in Tsuneta *et al.*, (1991).

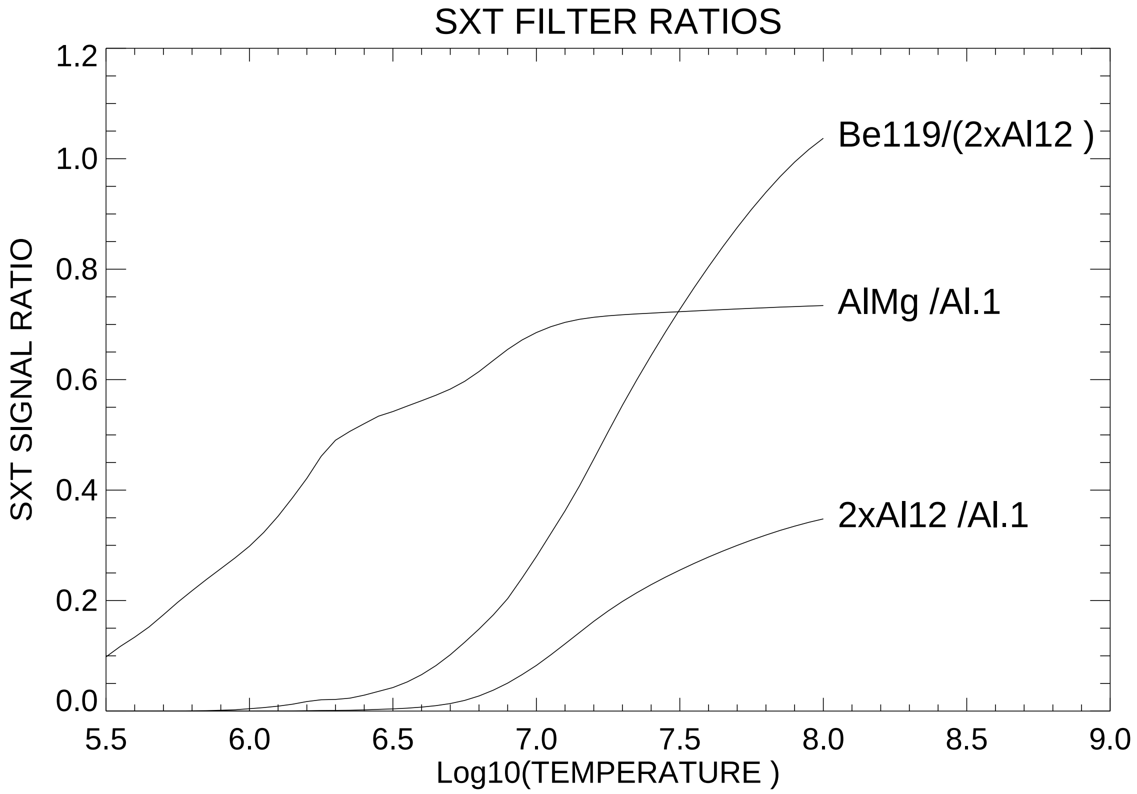

Figure 5.9: Ratios of the SXT response functions. For the curve marked "Be119/(2×Al12)" the effective observing time with the Al12 filter has been doubled with respect to the Be119 filter. For the curve marked "2×Al12/Al.1" the effective observing time with the Al12 filter has again been doubled, this time with respect to the Al.1 filter.

The SXT images may be used to compute temperatures after the data have been properly prepared. Normally, SXT_PREP should be called to prepare the data. The steps to follow are given in the next sections.

5.12.1 Read in the data

This can be done using YODAT or by directly reading the data file. If the data consist of observing regions (OR), then do not assemble the data when reading it with YODAT (the data will be assembled by SXT_PREP).

5.12.2 Prepare the data: Part 1 (background subtraction, registration)

Normally, you should use SXT_PREP to do the decompression, background subtraction, and stray light subtraction (pinhole leak). For example:

IDL> **SXT_PREP,index,data,new_index,new_data,unc,sat**
IDL> **SXT_PREP,index,data,new_index,new_data,unc,sat,/register**

The variables *unc* and *sat* are arrays that contain, respectively, the decompression error and information about the saturated pixels. It is not necessary to exposure normalize the data.

See the Reference Guide or Section 5.2 of the User's Guide for more information about running SXT_PREP.

5.12.3 Prepare the data: Part 2 (Filter selection)

It is necessary to select the filters which will be used for the analysis. The are many ways to do this. An example is given below for the Al.1 and Be119 filters:

```
IDL> i1 = where(GT_FILTB(new_index) eq 4)      ; Get Be119 indices
IDL> i2 = where(GT_FILTB(new_index) eq 2)      ; Get Al.1 indices

IDL> data1 = new_data(*,*,i1)
IDL> index1 = new_index(i1)
IDL> sat1 = sat(*,*,i1)
IDL> unc1 = unc(*,*,i1)

IDL> data2 = new_data(*,*,i2)
IDL> index2 = new_index(i2)
IDL> sat2 = sat(*,*,i2)
IDL> unc2 = unc(*,*,i2)
```

5.12.4 Prepare the data: Part 3 (Interpolation)

Because of the design of the SXT and the spacecraft telemetry system, images are never obtained in two different filters simultaneously. If the solar fluxes vary rapidly compared to the observing cadence, it is recommended that you interpolate the intensities of one filter to match the time at which the data in the other filter were acquired. This can be done using SXT_INTERP, which will do a linear interpolation on a pixel by pixel basis. An example calling sequence is:

```
IDL> SXT_INTERP,index1,data1,ntimes,unc1,sat1
```

where *ntimes* is an array of times to which the interpolation should be made. For example, in the case given here, it would be appropriate to set

```
IDL> ntimes = index2
```

in order to interpolate the filter "1" data to the times of filter "2" data.

Note that SXT_INTERP "over-writes" the input variables (to save memory).

5.12.5 Compute SXT temperatures

The routine SXT_TEEM should be used to compute temperatures from SXT filter data. This routine will compute the filter ratio and then determine the temperature and emission measure and uncertainties. Some example calling sequences are:

```
IDL> sxt_teem, index1, data1, index2, data2, Te
```

```
IDL> sxt_teem, index1, data1, index2, data2, Te, EM, dTe, dEM,subs=subs
IDL> sxt_teem, index1, data1, index2, data2, Te, EM, dTe, dEM,$
IDL>                  sat1=sat1,sat2=sat2,unc1=unc1,unc2=unc2
IDL> sxt_teem, index1, data1, index2, data2, Te, EM, dTe, dEM,$
IDL>                  sat1=sat1,sat2=sat2,unc1=unc1,unc2=unc2,/average
```

Note that *Te* (K) and *EM* (cm^{-3}) are returned as log10 values as are their uncertainties. Some ratios may have invalid temperatures and these are flagged with *Te=0*. *data*1 and *data*2 do not need to be exposure normalized if *index*1 and *index*2 are SXT index structures (since the *index*1 and *index*2 parameters can optionally be scalar values which are the X-ray filter numbers).

The results may be displayed using different techniques. One way is to use a user contributed routine:

```
IDL> tv_teem,Te,dTe,level=.3
```

where *level=* specifies the maximum value of *dTe* which is displayed. This is a convenient way to suppress the display of pixels that have low statistical significance.

5.12.6 Additional Hints and Notes

- Saturated Pixels

  There are two ways in which SXT images might be saturated (see the Instrument Guide). In the case of full-resolution images, saturation most likely occurs in the individual CCD pixels. In this case, the charge that is read out of the device is conserved, but it is distributed incorrectly. In the case of half- or quarter-resolution images, the saturation is most likely in the on-chip summing well. Pixels that are saturated will not produce a meaningful temperature and emission measure estimate. Because of the nature of this problem, there is no means to "restore" a saturated image to what it should have looked liked when the photons were initially absorbed in the CCD.

  SXT_PREP optionally returns an *sat* array which will flag saturated pixels. It calls SXT_SATPIX to do this and this will only work if data passed to SXT_PREP have not been decompressed or background subtracted. If the *sat* array is passed into SXT_TEEM, then the routine will sum all the saturated pixels together and treat them as a "macro" pixel. The output *Te* array will contain the resulting temperature and the *EM* array will contain the corresponding emission measure per pixel. For flare data which are acquired in full-resolution mode, this approach will give a mean estimate for the temperature and emission measure.

- How to improve signal-to-noise

  Because of the low gain of the SXT CCD camera, the ratio of DN/photon (depending on the filter) is between approximately 2 (Al.1) and 4 (Be119) for $log10(T_e) = 6.5$. In full-resolution images, the CCD full-well corresponds approximately to DN = 3313. Normally, the exposure time is automatically adjusted so as to keep the peak intensity at approximately half the full-well capacity. So it is difficult to obtain accurate temperature and emission measure estimates in the faint portions of flare images which have inherent dynamic ranges exceeding 100.

One way to improve the signal-to-noise ratio is to trade spatial resolution for photon flux, for example, by rebining the image. A procedure that does this is SXT_SUMXY. The calling sequence is:

IDL> **SXT_SUMXY,index,data,sumX,sumY,unc,sat**

Another method is to use a user contributed routine called GO_TEEM which allows the user to specify an arbitrarily shaped polygon region.

- Effect of Vignetting

  The vignetting function of the SXT is discussed in the Instrument Guide. Because the vignetting function is steeper for the thick filters, the temperature determination from the filter ratio will be effected somewhat by the position in the field-of-view of the SXT, especially when the filter ratio involves a thick/thin filter pair at the limb or near the edge of the SXT field-of-view. In this case, the vignetting effect should be corrected before calling SXT_TEEM.

- Effect of X-Ray Scattering

  SXT_TEEM does not make any corrections for scattering or the effects of the point-spread function. The X-ray scattering is a function which increases with X-ray energy (see the Instrument Guide). As a result, scattering is expected to be a bigger problem for the thicker filters. The user of SXT_TEEM should realize that this will sometimes result in higher than expected temperatures at the edges of bright flare kernels.

- Effect of Neutral Density filter

  As mentioned in the Instrument Guide, the Neutral Density filter increases X-ray scattering. Even though the amount is small (about 0.7%), and therefore, does not affect the results in the bright portions of a flare, it can significantly alter the level of the faint emission outside the flare region. One should be careful, therefore, about the interpretation of temperatures in faint regions when one of the filter pairs includes the neutral density filter.

5.13 Looking at Time-Profiles of Temperatures (GO_TEEM)

The GO_TEEM procedure computes the time-profile of the spatially averaged electron temperature (Te) and emission measure (EM) for a processed data cube. The results are plotted to the screen and are also saved to the return variables te, em, dte, and dem so they can be saved for later reference. The program prompts the user to select a pair of thin and thick filters to be used in the computation, an interpolation method (none is an option) and a reference image from the data cube from which the user can view in detail. Once the user has reviewed the selected reference image and decided which parts of the image to study, the user then uses various tools to specify small or large arbitrarily shaped regions to study the time-profiles.

Processed data cubes can be made in the following way:

- Run YODAT selecting the data of interest. While running YODAT use the SSWHERE option to reduce these data being careful to select only the normal mode data (get rid of dark frames), all the possible X-ray filters you wish to use, and all the exposures. Also use the FOV option to get only those data from the general area of interest.

- Run SXT_PREP to decompress, dark current subtract, and align the data. If your input data were taken in full and half resolution or were of different sizes be sure to specify the output resolution and size. Most importantly specify the output variables for uncertainty and saturated pixels as these are required in the GO_TEEM procedure.

A sample call for SXT_PREP is:

IDL> **sxt_prep, in_index, in_data, out_index, out_data, unc_data, satpix, /reg**

The *out_index* and *out_data* (in units of data number (DN)) can then be saved to a user created "sda" file with a simple call to SAV_SDA:

IDL> **sav_sda, 'my_processed_data', out_index, out_data**

Example call to GO_TEEM:

```
IDL> go_teem, out_index, out_data, te, em, dte, dem, fl1, fl2, time=time, $
IDL>                  comments=comments, unc=unc, satpix=satpix, /interp
```

	INPUT:	
	out_index	index record. from SXT_PREP
	out_data	data from SXT_PREP.
	unc	(optional) data for computing error bars
	satpix	(optional) data for flagging saturated pixels
	/interp	flag requesting that SPLINE interpolation of the database values is used (default is linear interp.)
Where:	OUTPUT:	
	te	list of log(Te K)
	em	list of log(EM cm^{-3})
	dte	1σ errors of Te
	dem	1σ errors of EM
	fl1	dn/sec light curves for selected thick filter
	fl2	dn/sec light curves for selected thin filter
	time	list of interpolated times
	comments	list of user entered comments for plot titles

5.14 How to Fit SXT Spectra (McTiernan) [*]

A general discussion of the McTiernan spectral fitting programs is given on page 59 in the description of fitting WBS spectra.

Also available are spectrometer-type fits for SXT data. In order to do this, the regular SXT filter response functions are rewritten as if they are spectrometer channel response curves.

SXT_FSP calls SXT_PREP and SXT_INTERP to get the data in the appropriate form. Then the routine ANY_FSP is called, which does the spectral fit in the same manner as the HXS or HXT. Only two types of fits can be used for SXT_FSP, the thermal SXR lines fit (tyspec = 10) or the single power law fit (tyspec = 1). Thus it is possible to get single temperature fits using three or more filters. SXTBOX_FSP does the same thing, except for regions chosen by the user, as in HXTBOX_FSP.

5.15 Useful Routines for SXT

See the routines listed in the Appendix of this User's Guide on page 76.

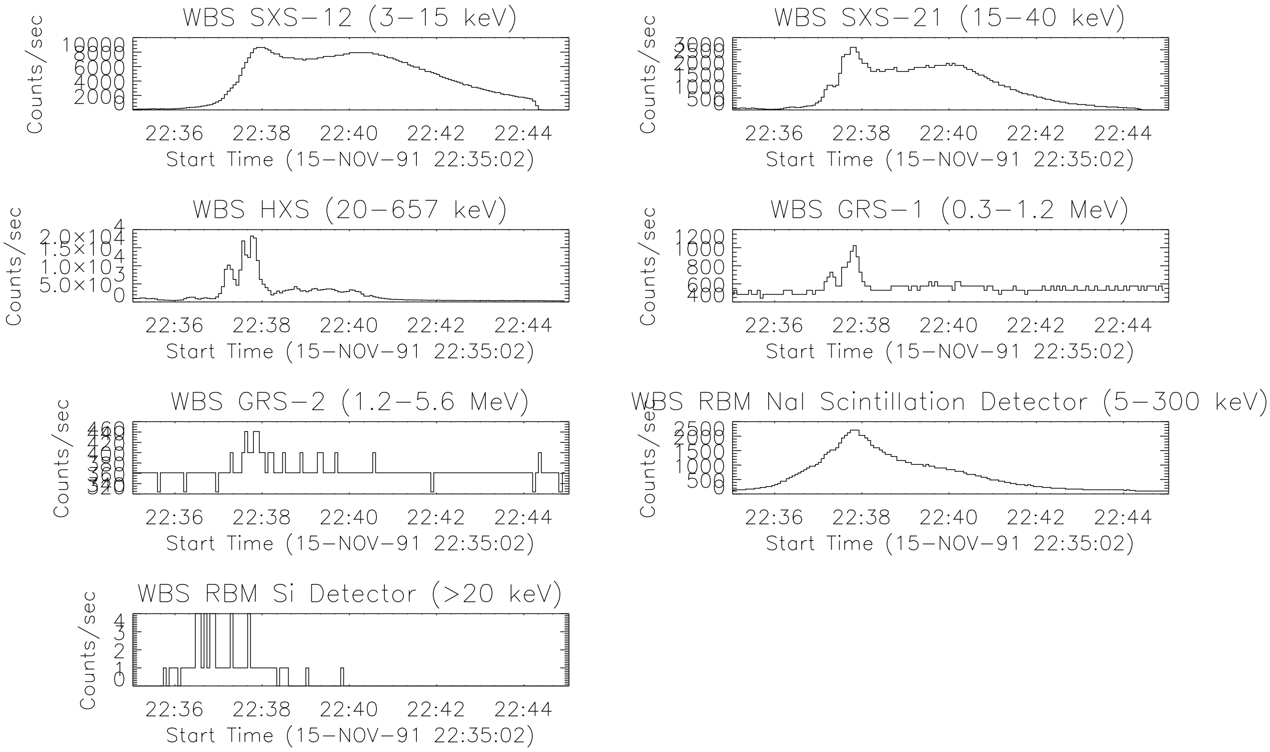

Figure 6.10: plott_wda, '15-nov-91 22:35', 10,psym=10

6 Wide Band Spectrometer (WBS)

6.1 How to Plot Light Curves of WBS Data

6.1.1 Plotting Seven WBS Sub-Instruments on One Page

It is possible to get light curves of several of the WBS channels on a single page by using the routine PLOTT_WDA. A sample plot is shown in Fig. 6.10. The routine will plot the SXS12, SXS21, HXS, GRS1, GRS2, radiation belt monitor (RBM) NaI Scintillation Detector, and RBM Si Detector. If the WBS index or roadmap have already been read in using YODAT or another routine, then you can use the command:

IDL> **plott_wda, roadmap**

You can also get WBS light curve plots by specifying a date and time. When you use this calling option with PLOTT_WDA it will use the WBS data stored in the Yohkoh observing log files, which is lower time and count resolution than accessing the data file directly. In the first example shown below, the data between 22:30 and 24:00 for the 15-Nov-91. In the second example, it will plot 30 minutes of data starting at 2-Dec-91 4:30.

IDL> **plott_wda, '15-nov-91 22:30', '15-nov-91 24:00'**
IDL> **plott_wda,'2-dec-91 4:30',30**

6.1.2 Light Curves of SXS and HXS-PC

First, run YODAT to extract all the relevant WDA data for the flare. After running YODAT, when you return to the IDL prompt, you can execute the following WBS time plotting routine.

IDL> **wbspc,data,index**

WBSPC plots the SXS-PC (2ch), and HXS-PC (2ch) time histories on a UT time axis. The following are the steps required to do the time plotting:

1. First, a rough light curve of HXS of the whole event will pop-up in which you can select a time interval that you want as the complete 4-channel time histories.
2. When selecting the time interval, follow the instructions which appear in the IDL session window.
3. After obtaining the result, a menu window will pop-up, in which you can select a print-out to the local printer or exit from this routine to the main YODAT routine.

NOTE: You must be careful of the limitation due to dead time correction (see DTCF_PC_HXS.PRO in the Reference Guide).

The following time plotting routines from Sections 6.1.3 through 6.1.12 have executing steps similar to those described above. To obtain more detailed information about the execution, you can access the header section of each procedure by using the DOC_LIBRARY IDL procedure.

6.1.3 Light Curves of SXS-PC

After having read the WBS data with YODAT, you can get two light curves of the count rate (in unit of counts/sec) of SXS-PC11 and PC12, and PC21 and PC22 with UTPLOT. The routine SXSCV extracts the SXS-PC data in full time resolution and with the deadtime corrected. You can plot these data with the UTPLOT routine. With the capability of the PLOT_LCUR routine, you can plot in the SXSPC routine the light curves more easily using the mouse. Some examples are:

IDL> **pc_data=sxscv(data,index,/plot)**
IDL> **pc_data=sxscv(data,index)**
IDL> **sxspc,pc_data,index**

6.1.4 Light Curve of HXS-PC

After having read the WBS data with YODAT, you can get a light curve of the count rate (counts/0.125 sec) of HXS-PC1, HXS-PC2 or HXS- PC1+PC2 with UTPLOT. There are some optional inputs such as error bars, integration time, time range and summation of adjacent pulse count (PC) data etc.

IDL> **plot_hxspc,index,data,pc**
IDL> **plot_hxspc,index,data,2**
IDL> **plot_hxspc,index,data,[1,2]**

where *pc* is 1 to have PC1 plotted, 2 for PC2, and [1,2] to have the sum of PC1 and PC2 plotted. NOTE: You must be careful of the limitation due to dead time correction (see DTCF_PC_HXS.PRO in the Reference Guide).

6.1.5 Light Curve of Ratio of HXS-PC2 to HXS-PC1

After having read the WBS data with YODAT, you can get a light curve of ratio of HXS-PC2 to HXS-PC1 counting rates with UTPLOT. There are similar optional inputs to PLOT_HXSPC.

IDL> **hxspc =mk_hxspc(index,data,xtime=xtime)**

MK_HXSPC is a function for making a float array of HXS-PC1 and PC2 data with UTPLOT. hxspc=fltarr(2,16*N), where N is the number of major frames. It takes two major frames to assemble a full set of WBS data. *hxspc(0,*)* is pc1 and *hxspc(1,*)* is pc2.

IDL> **utplot, xtime, hxspc(1,*)/hxspc(0,*), index(0)**

NOTE: You must be careful of the limitation due to dead time correction (see DTCF_PC_HXS.PRO in the Reference Guide).

6.1.6 Light Curve of HXS-PH

After having read the WBS data with YODAT, you can get a light curve of the count rate (counts/sec) of selected HXS-PH channel or energy ranges with UTPLOT.

IDL> **plot_hxsph,index,data,range=[a,b]**
IDL> **plot_hxsph,index,data,range=[234.5,345.6]**
IDL> **plot_hxsph,index,data,channel=[i,j]**
IDL> **plot_hxsph,index,data,channel=[15,20]**

where *range* is the energy range in keV to plot. The second example plot the total hxs_ph counts between 234.5 and 345.6 keV. The optional keyword *channel* can specify which channels to sum. The fourth example above bins all counts between 15th and 20th channels. There are some optional inputs such as error bar, time range and summation of adjacent PH data etc. NOTE: You must be careful of the limitation due to dead time correction (see DTCF_PH_HXS.PRO in the Reference Guide).

6.1.7 Light Curve of Ratio of HXS-PHi-j to HXS-PHk-l

After having read the WBS data with YODAT, you can get a light curve of the count rate of the ratio of HXS-PHi-j to HXS-PHk-l with UTPLOT.

IDL> **hxsph1=mk_hxsph(index,data,xtime=xtime,channel=[i,j])**
IDL> **hxsph2=mk_hxsph(index,data,xtime=xtime,channel=[k,l])**

MK_HXSPH is a function for making a float array of HXS-PH data. Next you extract background data.

IDL> **bgph=out_hxsph(index,data,/disp)**
IDL> **bgph=out_hxsph(index,data,stime=stime)**

OUT_HXSPH is a function for output of HXS_PH data structure. You subtract the background data and plot it.

IDL> **subph1=hxsph1-total(bgph.signal(i:j))**
IDL> **subph2=hxsph2-total(bgph.signal(k:l))**
IDL> **utplot,xtime,subph2/subph1,index(0)**

NOTE: You must be careful of the limitation due to dead time correction (see DTCF_PH_HXS.PRO in the Reference Guide).

6.1.8 Light Curve of GRS-PCL

After having read the WBS data with YODAT, you can get a light curve of the count rate (counts/0.25 sec) of GRS-PC11, PC12, PC21 or PC22 with UTPLOT. There are similar optional inputs to PLOT_HXSPC.

IDL> **plot_grspcl,index,data,pc=[11,12,21,22]**
IDL> **plot_grspcl,index,data,pc=[11,12]**
IDL> **plot_grspcl,index,data,pc=[21,22]**

6.1.9 Light Curve of GRS-PCH

After having read the WBS data with YODAT, you can get a light curve of the count rate (counts/0.5 sec) of GRS-PC13, PC14, PC15, PC16, PC23, PC24, PC25 or PC26 with UTPLOT. This works in a similar way to PLOT_GRSPCL.

IDL> **plot_grspch,index,data,pc=[13,14,15,16,23,24,25,26]**

6.1.10 Light Curve of Ratio of GRS-PCi to GRS-PCj

After having read the WBS data with YODAT, you can get a light curve of the ratio of GRS-PCi to GRS-PCj with UTPLOT. For GRS-PCL1, you do the following:

IDL> **grspcl=mk_grspcl(index,data,xtime=xtime)**

MK_GRSPCL is a function for making a float array of GRS-PC11 and GRS-PC12 data. grspcl=fltarr(4,8*N), where N is the number of major frames. It takes two major frames to assemble a full set of WBS data. *grspcl(0,*)* is pc11 and *grspcl(1,*)* is pc12. For example, to ratio GRS-PC12 to GRS-PC11:

IDL> **utplot,xtime,grspcl(1,*)/grspcl(0,*),index(0)**

For GRS-PCL2, you can do a similar ratio. *grspcl(2,*)* is pc21 and *grspcl(3,*)* is pc22.

6.1.11 Light Curve of GRS-PHL and GRS-PHH

After having read the WBS data with YODAT, you can get a light curve of the count rate (counts/4 sec) of GRS-PHL or GRS-PHH using UTPLOT. This works in a similar way to PLOT_HXSPH. The units of a and b are MeV.

IDL> **plot_grsphl,index,data,ph=1,range=[a,b]**
IDL> **plot_grsphl,index,data,ph=2,channel=[i,j]**
IDL> **plot_grsphh,index,data,ph=1,range=[a,b]**
IDL> **plot_grsphh,index,data,ph=2,channel=[i,j]**

6.1.12 Light Curve of Ratio of GRS-PHLi-j to GRS-PHLk-l

After having read the WBS data with YODAT, you can get a light curve of the count rate of the ratio of GRS-PHLi-j to GRS-PHLk-l with UTPLOT. This works in a similar way to light curve of ratio of HXS-PHi-j to HXS- PHk-l. For GRS-PHL1:

IDL> **grsphl1=mk_grsphl1(index,data,xtime=xtime,channel=[i,j])**
IDL> **grsphl2=mk_grsphl1(index,data,xtime=xtime,channel=[k,l])**

MK_GRSPHL1 is a function for making a float array of each of 128 GRS-PHL1 data. Next you extract background data.

IDL> **bgph=out_grsphl1(index,data,/disp)**
IDL> **bgph=out_grsphl1(index,data,stime=stime)**

OUT_GRSPHL1 is a function for output of the grs_phl1 data structure. Now you subtract the background data and plot it.

IDL> **subph1=grsphl1/4.0-total(bgph.signal(i:j))**
IDL> **subph2=grsphl2/4.0-total(bgph.signal(k:l))**
IDL> **utplot,xtime,subph2/subph1,index(0)**

For GRS-PHL2, you can do a similar calculation using MK_GRSPHL2 and OUT_GRSPHL2.

6.1.13 Light Curve of RBM-SC-PC and RBM-SD

After having read the WBS data with YODAT, you can get a light curve of the count rate (counts/0.25 sec) of RBM-SC-PC1, PC-2 or RBM-SD with UTPLOT.

IDL> **rbmpc=mk_rbmpc(index,data,xtime=xtime)**

MK_RBMPC is a function for making a float array of RBM-SC-PC1, PC2 and RBM-SD data with UTPLOT. rbmpc=flarr(3,8*N), where N is the number of major flames. It takes two major frames to assemble a full set of WBS data. *rbmpc(0,*)* is RBM-SC-PC1, *rbmpc(0,*)* is RBM-SC-PC1, and *rbmpc(2,*)* is RBM-SD. For example, to plot RBM-SC-PC1.

IDL> **utplot,xtime,rbmpc(0,*), index(0)**

NOTE: We do not take account of the deadtime correction.

6.2 How to Display WBS Spectra

6.2.1 Displaying the Spectra in Image Form

An image representation of the SXS, HXS, RBM-SC, and GRS spectra over time can be displayed using the routine DISP_WDA. After reading the data with YODAT, a sample call is:

IDL> **disp_wda, index, data**

6.2.2 HXS Spectral Display (single power law fitting)

A single power law fit to the HXS data can be obtained by executing:

IDL> **hxs_sp1,data,index**

The steps you should follow are:

1. First, a small menu window will pop-up on the screen.
2. There are four menu items
   - Select background time
   - Select flare time
   - Print out this result
   - Exit
3. You select the background time first. A rough HXS light curve will pop-up where you can determine the start and end time of the background necessary for the spectral analysis.
4. Next, when you click the second menu item, then the rough HXS light curve will pop-up again where the time interval in the flare needed for the spectral fitting analysis can be selected by left and right button on the mouse.
5. If the result is not satisfactory or several spectra are needed, you can repeat the first or second menu item.
6. When hard copies are necessary, you can click the third menu item.

NOTE: You must be careful of the limitations of the instrument. Spectral distortion occurs when the deadtime-corrected counts of total PH exceeds about 21,000 counts/sec. It is difficult to correct the spectral distortion (see DTCF_PH_HXS.PRO in the Reference Guide).

6.2.3 HXS Spectral Display (double power law fitting)

It is also possible to perform a double power-lwa fit to the HXS data. Just type:

IDL> **hxs_sp2,data,index**

NOTE: You must be careful of the limitations of the instrument outlined above.

6.2.4 GRS Spectral Display

You can plot a preliminary gamma-ray spectrum with

IDL> **grs_speff,index,data,ph=1**
IDL> **grs_speff,index,data,ph=2**

In this plot the flux of each channel is calculated by dividing counts by the full energy peak efficiency. The full energy peak efficiency at each channel is given by dividing "Efficiency" in grs1_40.rel and grs2_40.rel by 1000. NOTE: We do not plot the first 3 channels of data of GRS-PHL1 and the first four channels of data of GRS-PHL2 because their channel widths are not exactly determined.

6.3 How to Fit WBS Spectra

These sections describes the routines for detailed spectral analysis. You must be careful of the limitations of the instrument. HXS spectral distortion occurs when the deadtime-corrected counts of total PH exceeds about 21,000 counts/sec. It is difficult to correct the spectral distortion.

6.3.1 HXS_FSP (McTiernan)

Spectra can be obtained for HXS data using the program HXS_FSP, which takes the WBS index and data structures, and returns the parameters for whatever type of spectrum is desired. HXS_FSP is interactive. You choose the data interval, and the type of fit. The routines HXSGRS_FSP, GRS32_FSP and HXT_FSP work in the same way, for the HXS-GRS, GRS first 32 channels, and HXT data. The examples shown here will apply to all of them.

First, you must read the WBS data. The most simple method is to use YODAT, but you can read a WBS data file directly using RD_WDA. Given *index* and *data*, some sample calls to HXS_FSP are:

IDL> **HXS_FSP, index, data, fit_pars**
IDL> **HXS_FSP, index, data, fit_pars, sc_par, ch_dta**
IDL> **HXS_FSP, index, data, outfile=outfile**
IDL> **HXS_FSP, index, data, pfile=pfile**
IDL> **HXS_FSP, index, data, countfile=countfile**
IDL> **HXS_FSP, index, data, fit_pars, sc_par, ch_dta, sdel=sdel**

The input parameters, *index* and *data*, must always be present. All of the other parameters are optional. The structure fit_pars contains the results of the fit, including the type of fit, the values of the parameters, labels for the different parameters, the value of χ^2, and the interval and background times. (Times are given in the standard 7-element array, [hr,min,sec,msec,day,mon,yr].) The structure sc_par contains spectrometer channel information, such as the channel edges and the background count rate; ch_dta is a structure containing the data for each channel. The keyword *pfile* is the name for an output plot PostScript file, *outfile* is a name for a file for a print-out of the results, *sdel* is an array of channels you'd like to delete. The keyword *countfile* is the name for a file, with the format given in /ys/ucon/soft/mctiernan/spectral_data_format, which can be fit by the instrument-independent routine FSP_PROC.

The following are the steps required to fit HXS spectra.

1. First the program prompts for an HXS channel to be used for the choice of background and fitting intervals.
2. Next you choose the background interval using the routine PLOT_LCUR. After the choice is made, the program asks you to confirm that you are satisfied with the chosen interval, if so, then you choose the fitting interval using another call to PLOT_LCUR.
3. Once the choice of interval is confirmed, the program asks for the time resolution, in number of minor frames, which are 1 s apiece for HXS in flare mode, and 8 s apiece in quiet mode. (And 0.5 s in flare mode for HXT; 4 s in quiet mode, etc...) The resolution chosen need not be divisible into the total; the program will ignore extra data.
4. Next the program prompts for an integer for the type of fit desired, tyspec. The array a is the field of fit_pars which contains the fit parameters.
   - tyspec = 1, single power law, $a(0) = K$, $a(1) = \gamma$.
   - tyspec = 2, broken power law, $a(0) = K_1$, $a(1) = \gamma_1$, $a(2) = \gamma_2$, $a(3) = E_{br}$.
   - tyspec = 3, Thermal fit, $a(0) = EM/10^{47}$, $a(1) = T/10^6$.
   - tyspec = 4, Thermal plus P.L., $a(0) = EM/10^{47}$, $a(1) = T/10^6$, $a(2) = K$, $a(3) = \gamma$.
   - tyspec = 5, Double Thermal, $a(0) = EM_1/10^{47}$, $a(1) = T_1/10^6$, $a(2) = EM_2/10^{47}$, $a(3) = T_2/10^6$.
   - tyspec = 6, P.L. plus P.L., $a(0) = K_1$, $a(1) = \gamma_1$, $a(2) = K_2$, $a(3) = \gamma_2$
   - tyspec = 7, Thermal below E_{br}, P.L. above E_{br}, $a(0) = K$, $a(1) = \gamma$, $a(2) = T/10^6$, $a(3) = E_{br}$.
   - tyspec = 10, Thermal including SXR line emission, $a(0) = EM/10^{47}$, $a(1) = T/10^6$.
   - tyspec = 11, Thermal including SXR lines plus P.L., $a(0) = EM/10^{47}$, $a(1) = T/10^6$, $a(2) = K$, $a(3) = \gamma$.

   Tyspec = 8 and 9, thermal plus broken power law, and triple power law fits are available, although not yet recommended. K always denotes the photon flux at 1 keV in photons/(cm^2/sec keV) , γ is the spectral index, T is temperature in degrees K, and EM is emission measure in cm^{-3}; E_{br} is the break energy in keV.
5. The program then asks if you want to delete channels, if you respond "Y" then you'll be prompted for the number of channels to delete, and the channel numbers. If the keyword "sdel" is set this step is skipped; the channels specified in sdel are not included in the fit, and if sdel is set to -1 then all channels are included. Steps 3 and 4 will have associated keywords in future versions.

That's all there is to it. The program will plot each fit on the screen as it is completed, and if the keyword "pfile" is set, these plots go into the specified file.

6.3.2 FSP_PROC (McTiernan)

If you have set the keyword *countfile* in a call to one of the *_FSP routines, the output is a file which can be used as input to the routine FSP_PROC. You may want to do this if you want to save the count rates from a given instrument, and run multiple fits, using different kinds of spectra, and different sets of channels, for example,

IDL> **hxs_fsp, index, data, countfile='test.dat'**
IDL> **fsp_proc, 'HXS', 'test.dat', 'test.otp', fit_pars, sc_par, ch_dta**

will result in the fit of the file "test.dat" and put the output into the file "test.otp". The structures are the same, except that the interval and background times are in a structure which is not compatible with the Yohkoh time structures; you can change the times to the seven element array using the routine TFSPTOT7 (it is possible that the default time structures will be changed to the Yohkoh format in the future).

IDL> **n_fits = N_ELEMENTS(fit_pars)**
IDL> **t_new = intarr(n_fits, 7)**
IDL> **FOR j = 0, n_fits - 1 DO t_new(j, *) = tfsptot7(fit_pars(j).t)**

This will not be necessary in future versions.

6.3.3 SPFD HXS

SPFDHXS performs spectral fitting to the HXS data. The routine SPFDHXS runs on the only SUN workstation. After reading the data with YODAT, you choose a time interval for the flare data.

IDL> **flph=out_hxsph(index,data,/disp)**
IDL> **flph=out_hxsph(index,data,stime=stime)**

where */disp* allows you to choose by mouse operation, and *stime* is the signal time to be chosen. Next you choose the background data in a similar way.

IDL> **bgph=out_hxsph(index,data,/disp)**
IDL> **bgph=out_hxsph(index,data,stime=stime)**

You get a count spectrum (plot of counts/s keV vs. energy) with

IDL> **spplot_hxs,flph,bgph**

The text file of the spectral data for SPFD (routine of spectral fitting double precision) is written with

IDL> **write_hxsph,flph,bgph**

The name of the text file (default) is HXS_SPECTRUM.DATA.

We have a simplified routine called TEST_OUTH. This routine consists of three functions, OUT_HXSPH.PRO, SPPLOT_HXS.PRO and WRITE_HXSPH.PRO which are mentioned above. You can do the same procedure shown above by using TEST_OUTH.

IDL> **test_outh,index,data,/disp**
IDL> **test_outh,index,data,fltime=fltime,bgtime=bgtime**

Next you set a spectral function (single power law, broken power law, thermal spectrum, Gaussian spectrum and their combination) and the initial values of spectral parameters in the file `hxs.model`.

This file must be in the executing directory of SPFDHXS. Maximum numbers of fitting models and parameters ('PAR', 'XPAR' and 'YPAR') are 10 and 20, respectively.

The following is a sample parameter file for power law analysis:

```
'REM'    SAMPLE FILE FOR POWER LAW ANALYSIS
'SER'    0.01   /SYSTEMATIC ERROR
'CND'    300     3.00      0.001  / MAXSTP, CHISQ-MIN, VARIATION
'BIN'    0111111111111111111111111111110
'MDL'    POWL(1,2)
'PAR'    1          1          0.1          10
'PAR'    2          3.0       2.0          4.0
'OUT'    './HXS_RESULTS.DATA'
'FILE'   './HXS_SPECTRUM.DATA'
```

The following is sample parameter file for broken power law + Gauss analysis

```
'REM'      SAMPLE FILE FOR COUTOUR MAP (XAXIS INDEX1,YAXIS INDEX2)
'TRACE'    1                     / 1 OFF   0    ON
'SER'      0.01   /SYSTEMATIC ERROR
'CND'      300     3.00      0.001  / MAXSTP, CHISQ-MIN, VARIATION
'BIN'      0111111111111111111111111111110
'MDL'      BRPW(1,2,3,4) + GAUS (5,6,7)
'PAR'      1          1           0.1          10
'XPAR'     2          3.0        4.0          20
'YPAR'     3          4.0        5.0          20
'PAR'      4           70         50         100
'PAR'      5          3.0        0.3          30
'PAR'      6          511         0            0
'PAR'      7           2          0.2          20
'XRMK'     'INDEX1  ( <Eb)'
'YRMK'     'INDEX2  ( >Eb)'
'TITLE'    'CONTOUR MAP INDEX 1/2'
'CHI2'     5.0    / MAXIMUM ALLOWED RED_CHI2 VALUE.
'LEVEL'    99.0     95.0      90.0     68.0    3.0   /MAX:(10)
'OUT'      './HXS_RESULTS.DATA'
'FILE'     './HXS_SPECTRUM.DATA'
```

For details on the parameters for these files, see the file $DIR_WBS_CAL/spfd_hxs.example or /ys/ucon/soft/sato/hxs.model.

You carry out the spectral fitting on Sun machines with the command:

% **/ys/ucon/soft/sato/spfdhxs**

or from IDL with the command:

IDL> **spfdhxs**

You read the output file with the command

IDL> **spfd=read_spfdhxs()**

The name of file (default) is HXS_RESULTS.DATA. You plot the spectrum with

IDL> **spfdplot_hxs,spfd**
IDL> **spfdplot_hxs,spfd,/chimap**

You can plot a chi-square map with the second example above.

You can plot the time profile of time sequential data of spectral parameters (spectral index, spectral coefficient, temperature etc.) with

IDL> **utplot,spfd.xtime,spfd.parafinal(1,*),spfd(0).fltime(0)+spfd(0).date**

For details on the routines for this spectral fitting, use DOC_LIBRARY.

6.3.4 SPFD GRS

SPFDHXS performs spectral fitting to the GRS data. The routine SPFDGRS runs on the only SUN workstation. First you choose a time interval of the flare data.

The steps are identical to those described for SPFDHXS in the previous section, just use OUT_GRSPHL1 instead of OUT_HXSPH (and OUT_GRSPHL2 for GRS-2). Use SSPLOT_GRS and WRITE_GRSPH instead of SSPLOT_HXS and WRITE_HXSPH. There is a simplified routine called TEST_OUTG which is similar to TEST_OUTH. The default name of the text file with the data to be fit is GRS_SPECTRUM.DATA

Next you set a spectral function out of single power law, broken power law and power law plus Gaussian in a file of `grs.model`. This file must be in the executing directory of spfdgrs. You can set the parameters in a similar way to `hxs.model`. NOTE: Maximum numbers of fitting parameters and fitting models are 62 and 30, respectively.

The Unix and IDL routine to perform the spectral fitting is spfdgrs, and the results are read with READ_SPFDGRS. The default name of the results file is GRS_RESULTS.DATA. You plot the spectrum with SPFDPLOT_GRS.

6.4 Useful Routines for WBS

See the routines listed in the Appendix of this User's Guide on page 80.

7 Spacecraft Attitude and Ephemeris

7.1 Routines for Determining Spacecraft Pointing

7.1.1 RD_PNT_HIST

The routine RD_PNT_HIST will read the file `$DIR_GEN_STATUS/pointing.history` to determine all of the spacecraft commanded off-points. The structure returned has the .TIME and .DAY when the pointing change has occurred, and the offset (.OFFSET) for that pointing change. The S/C pointing bias is taken into account (the pointing bias has not been changed since 28-Oct-91). A sample call is:

IDL> **rd_pnt_hist, pntdata**

PR_PNT_HIST provides a simple method to display a summary of the commanded spacecraft pointing changes. Sample calls are:

IDL> **pr_pnt_hist**
IDL> **pr_pnt_hist, outfil='pr_pnt_hist.txt'**

The routine SXT_CMD_PNT will return the SXT pixel coordinates of the center of the sun using the spacecraft commanded pointing history database file (calls to SXT_CEN with the */cmd* switch simply call SXT_CMD_PNT). A sample call is:

IDL> **xy = sxt_cmd_pnt(index)**

7.1.2 SXT_CEN

The routine SXT_CEN should be used by all routines that wish to determine the S/C pointing. The function will return the pointing in SXT full resolution pixels for the set of times passed. The following logic is used if no switches are used

- Check to see if the ATT database files are available. The processed pointing value has already been calculated and is saved in those files.
- If the ATT step above fails, check to see if the ATR database file exist (or old PNT files). If they do, call GET_SUNCENTER to reduce the raw IRU and HXA data stored in the PNT files and return the actual S/C pointing.
- If the ATT and PNT steps fail, then read the commanded S/C file. This result gives the coarse pointing and will not correct for S/C drift.

Some sample calls are:

IDL> **cen = sxt_cen(times)**
IDL> **cen = sxt_cen(roadmap, /cmd)**
IDL> **cen = sxt_cen(index, roll=roll)**

The S/C roll value can optionally be returned with the keyword parameter *roll*. The roll values are only valid if the ATT database is available. If the */cmd* option is used, it will use the

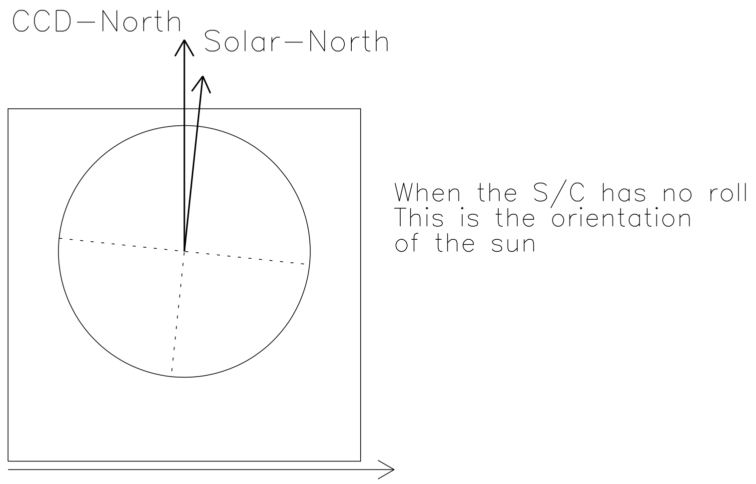

Figure 7.11: Roll Angle Offset for the SXT

commanded S/C data file which is much faster than reading the ATT database files. See the discussion on page 67 for more details on the ATT database.

7.1.3 GET_ROLL

The spacecraft is generally aligned to solar north, although it oscillates around slightly. The actual S/C roll value is determined from the star tracker and is stored in the ATT database file. There is also a one degree offset between the spacecraft and the SXT CCD-north. The orientation of the offset is shown in Fig. 7.11.

The sign convention of the roll value which is returned by GET_ROLL is shown in Fig. 7.12.

Figs. 7.11 and 7.12 can be displayed on the screen by using the IDL routine HELP_ROLL. Some sample calls to GET_ROLL are (it only accesses the date and time in *index*):

IDL> **roll = get_roll(index)**
IDL> **print, get_roll('23-jun-93 1:00')**
IDL> **print, get_roll('23-jun-93 1:00', /predict)**

The */predict* option tells it to not access the ATT database but to use the predicted roll based on seasonal variations. This value can be very incorrect for certain times. The */notsxt* option will return the S/C roll value without the SXT offset, and */sxt_offset* will just return the fixed

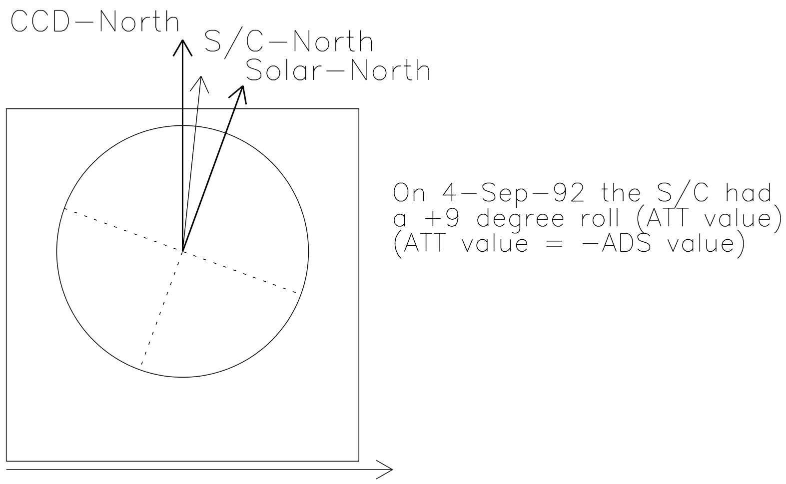

Figure 7.12: Sign Convention for the SXT Roll Angle

offset value between S/C north and SXT north. The routine SXT_CEN calls GET_ROLL automatically and you can get the roll using the keyword *roll* with SXT_CEN.

7.1.4 HXA_SUNCENTER

Calculate the suncenter position (in SXT pixel coordinates) from the HXA info in the ATR (or PNT) files. Tries to reconstruct hidden limbs.

IDL> **sunc = hxa_suncenter(atr)**
IDL> **sunc = hxa_suncenter(index=index)**

If the input *atr* or *index* have N elements, then the result will be a floating array of 4xN elements. *sunc(0,*)* is the SXT column number in 'IDL coordinates'[6], *sunc(1,*)* is the SXT line number, *sunc(2,*)* is the milliseconds of day for the input time, and *sunc(3,*)* is the days since 1-Jan-79 number for the input time.

7.2 Routines to Access the Yohkoh Attitude Database

7.2.1 RD_PNT

The PNT database was broken up into two parts in mid-1993. See RD_ATR and RD_ATT. The PNT database is no longer used.

[6] See the Appendix of the Instrument Guide for a discussion of SXT pixel coordinates

7.2.2 RD_ATR and RD_ATT

The RD_ATR routine allows a user to read the raw S/C pointing information which is stored in secondary database files. The ATR files have the IRU, HXA, and TFSS information for every major frame. The RD_ATT routine allows a user to read the processed S/C pointing information. Sample calls would be:

IDL> **rd_atr, sttim, entim, atr**
IDL> **rd_att, sttim, entim, att**
IDL> **rd_atr, '1-nov-91', '3-nov-91', atr**

The ATR structure that is returned has the following tags:

`.TIME`	The major frame time for the data
`.DAY`	The major frame day for the data
`.DP_MODE`	The S/C DP mode (see GT_DP_MODE)
`.DP_RATE`	The S/C telemetry rate (see GT_DP_RATE)
`.IRU`	A three element array with the raw IRU X,Y,Z values
`.TFSS`	A two element array with the raw TFSS values
`.HXA`	A four element array with the addresses of the limbs as detected by the HXA

The ATT database is derived from the HXA and IRU data using the routine IRUHXA2SXT which was written by Jean-Pierre Wuelser. The ATT structure that is returned has the following tags:

`.TIME`	The major frame time for the data
`.DAY`	The major frame day for the data
`.DP_MODE`	The S/C DP mode (see GT_DP_MODE)
`.DP_RATE`	The S/C telemetry rate (see GT_DP_RATE)
`.PNT`	A three element array with the derived center for the sun in SXT full resolution pixels. (0) = E/W (East has smaller values) in 1/100 SXT FR pixel units. (1) = N/S (South has smaller values) in 1/100 SXT FR pixel units. (2) = Roll in 0.1 arcsecond units - Negative values are S/C rotating counter-clockwise relative to solar north
`.STATUS1`	Indicates the reliability of each sun center position. 1: plain S/C commanded value used, no dejittering with IRU. 2: S/C commanded value dejittered with IRU. 4: HXA value dejittered with IRU. Standard result, good. Note: S/C commanded values can be several pixels off.
`.STATUS2`	- Reserved
`.ADS`	- If set, then the S/C roll value from the ADS database was inserted into the structure

7.2.3 GET_ATT

For a given set of input times, GET_ATT will read the relevant ATT records (if 100 input times are passed in, then 100 ATT records are returned). This routine is useful when trying to access ATT data that covers a set of times over a long time span. A sample call would be:

IDL> **get_att, index, att**

7.2.4 GT_HXA

GT_HXA will extract the HXA data from a ATR structure or from an ADA data structure. Some sample calls are:

IDL> **hxa = gt_hxa(atr)**
IDL> **sxtcen = gt_hxa(atr_data, /sxtpix)**
IDL> **sxtcen = gt_hxa(atr_data, /sxtpix, /x)**
IDL> **hxacen = gt_hxa(atr_data, /hxacen)**
IDL> **hxacen = gt_hxa(atr_data, /hxacen, /y)**
IDL> **x1 = gt_hxa(atr_data, 0)**
IDL> **x2 = gt_hxa(atr_data, 1)**
IDL> **y1 = gt_hxa(ada_data, 2)**

The *hxa* result would be 4xN where there are four addresses for HXA limbs. The */x* or */y* switches will result in a single address being returned, and the */sxtpix* will return the results in SXT pixel coordinates.

7.2.5 GT_IRU

It is possible to extract the IRU information from the ATR or ADA structures using GT_IRU. The output is in arcseconds. It is also possible to have the drift in the IRU removed for short time periods by using the /RESID switch (it fits a line to the drift and subtracts that drift). Sample calls are:

IDL> **iru = gt_iru(atr)**
IDL> **iru = gt_iru(atr, /resid)**
IDL> **iru = gt_iru(ada)**
IDL> **iru = gt_iru(ada, index, /resid)**

7.3 Spacecraft Ephemeris

7.3.1 ORB files

7.3.2 FEM Files

some text?

A Tables of Useful Yohkoh Routines

This appendix contains a list of the most useful routines used by Yohkoh personnel. The user should consult the Reference Manual and/or use the IDL routine DOC_LIBRARY for more information. (M) is main program, (P) is procedure, (F) is function, and (S) is script.

A.1 General Useful Routines

The following table is a list of the routines useful for General Yohkoh analysis.

		GENERAL HELP
help_prefix	P	Gives information about Yohkoh file names
pr_path	P	Print the !path variable
pr_path_lib	P	Print the full path of a given routine(s) (* is ok)
doc_library	P	Print the documentation header to a given routine
fastdoc	P	Same as DOC_LIBRARY but is much faster
		GENERAL YOHKOH ROUTINES
yodat	M	Select and Read any Yohkoh data files
stepper	P	Display images one-by-one or as a movie
show_pix	P	Display PR and noted Yohkoh images
xstepper	P	Display images in rapid fashion
		DATA PATH AND FINDING FILES ROUTINES
data_paths	F	Obtain the paths to reformatted and databases files
new_dpath	P	Add personal directories to on-line Yohkoh data path
concat_dir	F	Builds a path given file name and logical or environment var
get_subdirs	F	Return all the subdirectories under a given dir
file_list	F	Get a list of files that fulfill given specifications
file_menu	F	Menu driven FILE_LIST but only returns single files
files_search	P	Search for a given string within the give directory(s)
ydb_exist	F	Checks status of needed Yohkoh database files
		YOHKOH INFORMATION ROUTINES
get_info	F	Convert index data into a readable information. (/longform)
pr_fem	P	Print day and night events given a date
pr_evn	P	Print flare-event summary for a given date(s)
plot_evn	P	Plot station contacts, SAA, S/C D/N, Yohkoh data periods & modes
rd_fem	P	Read FEM structures
rd_evn	P	Read EVN structures
rd_gev	P	Read GEV structures
pr_pnt_hist	P	List a summary of the spacecraft pointing history

		GENERAL PLOTTING ROUTINES
utplot	P	Line plots with time axis
eutplot	P	Overplots error bars on plot previously drawn by UTPLOT
outplot	P	Overplots on plot previously drawn by UTPLOT
clear_utplot	P	Resets the UTPLOT plotting variables
clearplot	P	Reset plot parameters
ploty	P	Plot light curves from all Yohkoh instruments
ocontour	P	Contours overlaid on images
plot_arc	P	Plot the intensity along a path of a given width
plot_lcur	P	Time plots with zoom abcissa
box_latlon	P	Draw a box from specified latitude/longitude values
		YOHKOH TIME ROUTINES
tim2dset	F	Return data set numbers from roadmap given time-date
sel_timrange	F	Return indices of roadmap or index which agrees to input times
fmt_tim	F	Returns formatted date and time from roadmap or index
fmt_timer	P	Print 1st and last formatted times from an index file
anytim2ints	F	Converts date and time to internal-format
anytim2ex	F	Converts date and time to external-format
int2secarr	F	Converts roadmap or index times to seconds from a ref. time
anytim2doy	F	Converts date and time to day of year number
ex2fid	F	Converts external-format to Yohkoh file ID: yymmdd.hhmm
tim2orbit	P	Print or return the FID, WID, D/N, SAA for given date and time
ex2dow	F	Converts external-format to day of the week number
gt_day	F	Returns date from roadmap or index
gt_time	F	Returns UT time from roadmap or index
doy2ex	F	Convert from day of year (DOY) to internal-format
doy2date	P	Convert from DOY and year to month and day or yymmdd
doytim2ex	F	Convert DOY and time to external-format, year is optional
get_yo_dates	F	Return significant Yohkoh dates (e.g., launch)
anytim2weekid	F	Convert a Yohkoh time to a week ID
pr_tim2week	P	Print week ID on screen for given time
weekid2ex	F	Convert a Yohkoh week ID to external time
pr_week2tim	P	Print time on screen for given Yohkoh week ID
		YOHKOH ATTITUDE AND SOLAR EPHEMERIS ROUTINES
help_roll	P	Display Yohkoh roll offset and sign conventions
get_roll	F	Return S/C roll value
hxa_suncenter	F	Return sun center position (SXT pix coord.) from HXA data
conv_p2a	F	Return angle in arcsec relative to sun center given pixel address
conv_a2p	F	Return pixel address given angle in arcsec relative to sun center
conv_a2h	F	Return heliocentric coordinates from sun center angle in arcsec
conv_h2a	F	Return sun center angle in arcsec from heliocentric coordinates

conv_p2h	F	Return heliocentric coordinates from SXT-pixel address
conv_h2p	F	Return SXT-pixel address from heliocentric coordinates
conv_p2hxt	F	Convert SXT-pixel address to HXT-HXA sun sensor address
conv_hxt2p	F	Convert HXT-HXA sun sensor address to SXT-pixel address
conv_p2c	F	Converts SXT pixel locations to Carrington coordinates
conv_c2p	F	Converts Carrington coordinates to SXT pixel location
get_rb0p	F	Return solar radius, b0 angle, p angle for given time
rd_atr	P	Read raw S/C pointing information for specified times
rd_att	P	Read processed S/C pointing information for specified times
get_att	P	Read processed S/C pointing information for index-times
c2s	F	Covert between Cartesian and spherical coordinates
get_carr	F	Return geocentric physical ephemeris of the Sun
gt_hxa	F	Read HXA data from either ATR or ADA data structures
gt_iru	F	Read IRU data from either ATR or ADA data structures
		YOHKOH I/O ROUTINES
rdtap	P	Read data off a Yohkoh archive tape
def_tapd	F	Return the full device name of the specified tape drive #
go_rdtap	P	Read data off a Yohkoh archive tape via simple user interface
rd_ydbtap	P	Read data off a Yohkoh database (ydb) tape
rd_xda	P	Read any Yohkoh file data section
rd_roadmap	P	Read any Yohkoh file roadmap section
rd_pointer	P	Read any Yohkoh file pointer seciton
rd_fheader	P	Read any Yohkoh file header section
rd_qs	P	Read any Yohkoh file quasi-static section
sav_sda	P	Write SXT data to Yohkoh-format data file
sav_bda	P	Write BCS data to Yohkoh-format data file
sav_hxi	P	Write HXT data to Yohkoh-format data file
		OTHER I/O AND CONVERSION ROUTINES
savegen	P	Save data to a GENX (GENeral Xdr) file
restgen	P	Restore data from GENX (GENeral Xdr) file
rfits	F	Read FITS format files
wrt_fits	P	Write a data-array into a standard FITS file
sxt2fits	P	Write SXT index and data to standard FITS file
mk_tiffb	P	Write (SXT) image to a full color (24 bit) TIFF file
mk_tiffp	P	Write (SXT) image to a TIFF-Palette Color (256 levels) file
rd_tfile	F	Read and return contents of text file
file_append	P	Appends text to text file
file_exist	F	Boolean check for finding files
file_delete	P	Deletes or removes files
file_purge	P	Purge file that have similar file names (wild cards OK)
file_info	F	Return standard file info. for a given file
file_compress	P	Compress a file using the standard Unix compress utility
file_uncompress	P	Decompress a file using the standard Unix compress utility

		OPERATING SYSTEM ROUTINES
set_logenv	P	Set Unix environmentals/VMS logicals
get_logenv	F	Get Unix environmentals/VMS logicals (* wildcard ok)
		OBSERVING LOG ROUTINES
rd_obs	P	Read the observing log
pr_sxtobs	P	Print get_info2 output from observing log
bcs_24hr_plot	P	Plots BCS light curve for 24 hour period
pfi_loc	P	Display the location of the the SXT PFI on the sun
		DEVICE ROUTINES
wdef	P	Create IDL-graphics window on an X device
wshow	P	Pops an IDL-graphics window to the forground
rcolor	P	Capture and print a window on Misubishi Color Printer (D-tou)
hardcopy	P	Generate color or B&W PostScript file & output to PS printer
pprint	P	Output an idl.ps file to printer
		USEFUL MISC. ROUTINES
str_concat_tags	F	Concatenate two similar data structures
str_diff	F	Determine the differences in contents between two structures
show_struct	S	Script to print data structures. Example, show_struct sxt
prstr	P	Print a string-array one element per line
pmm	P	Print the min and max of an array
str2arr	F	Parse a string using a given delimiter into a string array
arr2str	F	Combine a string array into a string with "," delimeter
str_replace	F	Replace all occurances of a given string with another string
wmenu_sel	F	Select from a string-array list via mouse
wc_where	F	Perform wild card searches on string arrays
yo_file_check	P	Compare actual file size and file size in pointer rec.
mk_mapfile	P	Create/update GENX file which maps files into pathnames e.g. fastdoc
		NOAA AND GOES ROUTINES
plot_nar	P	Plot a grid with NOAA Active Region positions
pr_nar	P	Print NOAA Active Region positions
plot_goes	P	Plot 1-minute averaged or 3 second GOES fluxes
pr_gev	P	Print GOES events
goes_reducer	P	Compute Te, EM, L_X from GOES 1 min or 3 sec fluxes
rd_gxd	P	Read 3-sec or 1-min data from GOES-6 or GOES-7
rd_gxt	P	Read 1-min averaged GOES fluxes
		GRO/BATSE ROUTINES
pr_gbe	P	List GRO/BATSE events for given time period or event
plot_gbl	P	Plot GRO/BATSE light curve

A.2 BCS Useful Routines

The following table is a list of the routines useful for BCS analysis.

		GENERAL BCS ROUTINES
yodat	M	Select and Read any Yohkoh data files
bcs_decomp	F	Decompress (8 bit) BDA spectra data
bcs_comp	F	Compresses BCS spectral data
ext_bcschan	F	Extracts spectra for a single channel from BDA data
bcs_norm	F	Normalizes extracted data for integration time; extracts fast queue data
bcs_dspec	F	Extracts, decompresses and normalizes BDA data
bcs_tvec	F	Creates a intergration-time corrected time vector
sum_bda	F	Sums a number of BDA spectra
bcs_point	F	Source position on disk from resonance line bin number
bcs_dtfac	F	Calculates deadtime corrections
		BCS GT ROUTINES
gt_total_counts	F	Extracts detector total counts
gt_blockid	F	Extracts data block ID
gt_bsc_bincal	F	Returns a structure with miscellaneous calibration info
		BCS DISPLAY ROUTINES
plott_bda	P	Plot a light curve of the 4 BCS channels
wbda	P	Browse light curves and spectra in BDA files
plot_bsc	P	Plot BSC light curves and spectra
map_bsc	P	Stack plots of successive BSC spectra
lc_bsc	P	Plot and extract BSC light curves of selected spectral regions
fit_bsc_plot	P	Plot time series results from FIT_BSC
bcs_spec_plot	P	Plot BSC spectrum (landscape)
mk_fit_bsc_eps	P	Plot BSC spectrum (portrait and eps)
wbsc	P	Browse light curves and spectra in BSC files
		BCS SPECTRALLY CALIBRATED ROUTINES (BSC)
mk_bsc	P	Create instrumentally corrected BCS data (BSC data)
fit_bsc	P	Fit synthetic spectra to observed BSC spectra
get_bsc	F	Extract BSC data fields
sel_bsc	P	Extracts wavelength and flux fields from BSC_DATA
get_bsc_anno	F	Return plot annotation from BSC index structure
bcs_broad	F	Convert between line width and Doppler velocity
gt_bsc_wave	F	Generate a standard wavelength vector
gt_bsc_chan	F	Extract BCS channel number from BSC index

gt_bsc_bincal	F	Get BCS crystal calibration data (calls RD_BCSCAL)
list_bsc	P	List BSC header information
bsc2bsd	P	Convert a BSC file to BSD format
help_bsc	P	Print a list of BSC routines
		BCS THEORETICAL SPECTRAL ROUTINES
bcs_synthetic	M	Main program to compute BCS synthetic spectra
bcs_line	P	Compute theoretical line intensities
conflx	F	Compute theoretical continuum
collexc	P	Compute effective excitation rate coefficients
drexc	P	Compute effective dielectronic rate coefficients
wvoigt	F	Compute a Voigt profile
bcs_spec	P	Compute a BCS synthetic spectrum
get_atomic	F	Read atomic data files
get_elemabun	F	Return Yohkoh adopted coronal abundances
get_ionbal	F	Return ionization balance fractions
		BCS READ/WRITE
rd_bcscal	P	Reads calibration data for a single BCS channel
rd_bda	P	Reads in a BDA structures (Reformatted data)
sav_bda	P	Saves new or modified BDA structures
sav_bsc	P	Save a BSC index and data array to a file
rd_bsc	P	Read a BSC index and data array from a file
get_bsd	P	Reads in a BSD structures (F77 and IDL code)
wr_bsd	P	Saves new or modified BSD structures

A.3 HXT Useful Routines

The following table is a list of the routines useful for HXT analysis.

		HXT LIGHT CURVE ROUTINES
yodat	M	Select and Read any Yohkoh data files
plott_hda	P	Time series plot of 4 channels
hxt_4chplot	P	Time series plot of 4 channels
		HXT DATA EXTRACTION ROUTINES
ave_cts	F	HXT count rate data (in units of cts/s/SC)
mk_hxtdata	F	Extract a portion of the HXT data
		HXT SPECTRAL FITTING ROUTINES
hxtbox_fsp	P	Peform spectral fit to HXT data
		HXT IMAGE SYNTHESIS ROUTINES
hxt_qlook	P	Quick image synthesis
hxtpro	M	Image synthesis program (only runs on SUN machine)
hxt_img	M	Image synthesis program (McTiernan)
		HXT ANALYSIS ROUTINES
get_hxt_pos	F	Returns the HXT flare position address
hxt_powerlaw	F	Estimate hard X-ray flux and photon index
hxt_thermal	F	Calculate temperature and emission measure

A.4 SXT Useful Routines

The following table is a list of the routines useful for SXT analysis.

		GENERAL SXT ROUTINES
yodat	M	Select and Read any Yohkoh data files
stepper	P	Display images one-by-one or as a movie
sswhere	F	Select SXT images by filter, exp., mode, etc.
show_obs3	P	Plot the SXT observing table sequence
show_obs4	P	Enhanced verison of SHOW_OBS3
plot_fov	P	Plot SXT PFI fields of view (FOV)
lcur_image	P	Plot a light curve from a selected area of SXT data cube
plot_arc	P	Plot the intensity along a path of a given width.
xmovie_sfm	P	Reviews the online SFM database via widget
get_sfm	P	Returns SFM index and image closest to requested times
disp_month	P	Display a month of SXT images in calendar format
		SXT I/O ROUTINES
rd_sot	P	Read the SOT files: rd_sot,sttim, entim, index, data_structure
rd_xda	P	Read any of the Yohkoh reformatted data files
		SXT INFORMATION ROUTINES
get_info	F	Return formatted information from SXT index or roadmap
pr_sxtobs	P	Print GET_INFO output from observing log
sxtpnt_sum	P	Summary line for each unique pointing
pfi_loc	P	Display the location of the the SXT PFI on the sun
		SXT DATA REDUCTION ROUTINES
sxt_prep	P	Prepares SXT data for further processing (LEAK_SUB, aligns,...)
xsxt_prep	M	Adds a graphical user interface to SXT_PREP
exp_norm	F	Perform exposure normalization
		PLASMA PARAMETER ROUTINES
go_teem	P	Compute Te and EM timeseries from SXT image ratios
sxt_teem	P	Compute Te and EM from SXT image ratios
tv_teem	P	Display Te's and tolerances in image format
sxt_interp	P	Do a linear time interpolation of SXT images
sxt_flux	F	Compute the Flux (DN/s or ph) given Te and filter
sxt_erg_per_dn	F	Compute erg/DN at the sun given Te and filter
sxt_dn2ph	F	Converts DN to photons counts given DN, ph energy or wavelength

		ALIGNMENT/POINTING ROUTINES
sxt_prep	P	Aligns and dejitters SXT images
sxt_center	P	Returns row, column, and radius of the solar disk from images
sxt_cen	F	Return SXT sun center coordinates
sxtpnt_sum	P	Return unique PFI pointings
get_rb0p	P	Calculate Sun's radius, B0 angle, P angle
coal_image	P	A low level routine to coalign images (shift, rotate, stretch)
get_gbo_pfi	F	Given a SXT PFI image and GBO image, extract and align the portion of the GBO image which corresponds to the PFI
		SXT PLOTTING ROUTINES
sxt_grid	P	Draws heliocenteric grid over an image
profiles	P	Plot line or column intensity profiles
tprofiles	P	Plot light curve for any pixel in a data cube
xy_raster	P	Display a raster of images
plot_clon	P	Overlay Carrington longitude or grid on SXT image
		GENERAL IMAGE PROCESSING ROUTINES
unsharp_mask	F	Improve image contrast using unsharp mask method
sxt_merge	P	Merge two overlaping off-point images into a single image
compst	F	Interleaves two images with each using half the color table
sxt_composite	F	Make desaturated composite image from several images
wdefroi	F	Widget to extract sub-image
ext_subset	P	Extracts a subset (in rows and columns) from a data cube
de_spiker	F	Cleans images by removing SAA spikes
spikes	F	Locates single pixel spikes in an image array
rm_darklimb	F	Removes the limb darkening from optical images
sxt_off_axis	F	Correct an SXT image for the X-ray vignette function
sxt_vignette	F	Return SXT X-ray vignette function determined from WSMR1 data
		MOVIE MAKING AND VIEWING ROUTINES
go_nvs5	P	Writes SXT data to SONY laser disk
go_per2opt	P	Writes SXT data to Panasonic laser disk
mk_vcr	P	Controls and commands the SONY laser disk for output to VCR
video_menu	P	Lists the all movie segments recorded on the SONY laser disk
video_title	P	Make a title page image (for laser disk)
		DATA REDUCTION ROUTINES
leak_sub	F	Decompress and subtract dark current/stray light from SXT image
dark_sub	F	Decompress and subtract dark current from SXT image
sxt_decomp	F	Decompress SXT data
sxt_comp	F	Compress SXT data
sfd_decomp	F	Decompress an SFD image
sfd_comp	F	Compress an SFD image

restore_low8	F	Restore low-8 (bit) optical images to 12 bit representation
get_dc_image	P	Return appropriate dark frame and matching index
sxt_satpix	F	Returns map of saturated pixels from index and data
dprate2sec	F	Return the major frame rate in seconds given a DP_RATE code
pr_image	P	Print the average DN level of a box
		LOW LEVEL SXT ACCESS ROUTINES
gt_filta	F	Return filter A (optical) setting from index or roadmap
gt_filtb	F	Return filter B (X-ray) setting from index or roadmap
gt_res	F	Return image pixel-resolution from index or roadmap
gt_expdur	F	Return effective exposure duration from index
gt_time	F	Return time exposure was taken
gt_day	F	Return date exposure was taken
gt_explat	F	Return Exposure latency time from indexed time
gt_dpe	F	Return command exposure level
gt_mbe	F	Return mail box exposure level
gt_dp_mode	F	Return Yohkoh modes of operation
gt_dp_rate	F	Return SXT data telemetry rate
gt_corner	F	Return corner pixel address relative to sun center
gt_center	F	Return center of FOV in full res pix, helio coord, angle, HXT pix
gt_temp_ccd	F	Return the temperature of the CCD in degrees celsius
gt_temp_hk	F	Return the monitor temperatures for the S/C and SXT
gt_comp	F	Return the compression mode for an image
gt_pfi_ffi	F	Return the image-type PFI, FFI, PFI-observing region, FFI-patrol
gt_sxt_cen	F	Return sun center value from history record
gt_sxt_roll	F	Return the roll of the SXT image
gt_sxt_axis	F	Returns boresight axis location for SXT
		ISAS OPERATIONS ROUTINES
contacts	P	Show Yohkoh KSC contact times
first_light	M	Provides quick look at SXT conditions
lastsfd	P	Read and displays the most recent sfd file
goes_summary	P	Plot GOES summary (shows flares separately)
plot_fem	P	Plot the Yohkoh day/night, SAA, Pass schedules
pr_fem	P	Print the Yohkoh day/night, SAA, Pass schedules
pr_evn	P	Print a summary of Yohkoh data times (flare/quiet)
pr_gev	P	Print GOES events

		SXT CHIEF OBSERVER ROUTINES
gtab_entry	F	List SXT entry tables
gtab_pfi	F	List SXT PFI tables
gtab_ffi	F	List SXT FFI tables
gtab_comm	F	List SXT common tables
gtab_roi	F	List SXT observing region tables
gtab_summary	P	List a summary of SXT sequence tables
seq_summary	F	List SXT table (-777 in yodat) ⇒ print,seq_summary(roadmap)
get_sunset	F	Get the time of Yohkoh sunsets (Ultrix and SGI only)
plot_sls	M	Plot the SXT Leak Scatter Log
plot_sld	M	Plot the SXT Leak Dark Current Log
sld_sls_vstrips	P	Run after plot_sls as: IDL> sld_sls_vstrips,index_s,sls
sxt_obs_coord	P	To find the coordinates for ARS2 pointing
plot_sot	P	Plot the aspect sensor degradation
plot_temps	P	Plot SXT instrument temperatures
plot_ssl	P	Plots SXT summaries (filter use, exposure, ...)

A.5 WBS Useful Routines

The following table is a list of the routines useful for WBS analysis.

		WBS LIGHT CURVE ROUTINES
yodat	M	Select and Read any Yohkoh data files
plott_wda	P	Time plot series of seven sub-instruments
wbspc	P	Plots the SXS-PC and HXS-PC time histories
sxspc	P	Plots the SXS-PC data
plot_hxspc	P	Plots the HXS-PC data
plot_hxsph	P	Plots the HXS-PH data
plot_grspcl	P	Plots the GRS-PCL data
plot_grspch	P	Plots the GRS-PCH data
plot_grsphl	P	Plots the GRS-PHL data
plot_grsphh	P	Plots the GRS-PHH data
		WBS SPECTRAL DISPLAY ROUTINES
disp_wda	P	Image representation of the WBS spectra is displayed
hxs_sp1	P	Spectral plot obtained a single power law fitting
hxs_sp2	P	Spectral plot obtained a double power law fitting
grs_speff	P	Plot a preliminary gamma-ray spectrum
		WBS DATA EXTRACTION ROUTINES
mk_rbmpc	F	Extract the RBM data into arrays
		WBS SPECTRAL FITTING ROUTINES
hxs_fsp	P	Perform spectral fit to HXS data
spfdhxs	P	Fit HXS data (Runs on SUN machines only)
spfdgrs	P	Fit GRS data (Runs on SUN machines only)
		WBS CALIBRATION ROUTINES
dtcf_pc_hxs	P	Perform dead time correction for HXS-PC
dtcf_ph_hxs	P	Perform dead time correction for HXS-PH
dtcf_ph_grs1l	P	Perform dead time correction for GRS-PH1-L
dtcf_ph_grs2l	P	Perform dead time correction for GRS-PH2-L

B Format and Naming of Yohkoh Data Files

B.1 Yohkoh Data File Format

Each file is logically divided into the following six sections. Some files will not use all the sections described below, but all will have a Pointer and File Header Section.

1. File Information and Pointer Section
2. File Header Section
3. Quasi-Static Index Section
4. Index and Data Section
5. Optional Data Section
6. Road Map Section

The program which reads the file learns from the Pointer Section how to read the rest of the file and where to go to get certain data.

The File Header Section provides information on what data are contained in the file, generally the extent of the time covered by the contents.

The Quasi-Static Section of the file contains index information that does not vary during the course of an orbit, or varies slowly.

The Index and Data section contains 'data sets'. A data set is a single image for SXT, single spectrum for BCS, a single major frame of data for HXT, and two major frames of data for WBS (it takes two major frames for a complete set of WBS data). For each data set there is an index which describes the date and time that the data were taken, the mode and position of the instrument's peripherals (e.g., filters, HV), and information on temperature and gain information.

The Optional Data Section is only used by the BCS data (BDA) files and the spacecraft attitude (ADA) files. The BDA files hold the 'DP_SYNC' information, which is information that is coming down every major frame. Since the BCS spectra are asynchronous to the major frames, it is stored separately. The ADA file holds the full 2048 point HXA scans.

The Roadmap Section allows a user to access a brief summary of the contents of the file and to perform searches on that summary to select what data should be extracted.

For a full description of the Yohkoh database and all the different data structures, see the **File Control Document**.

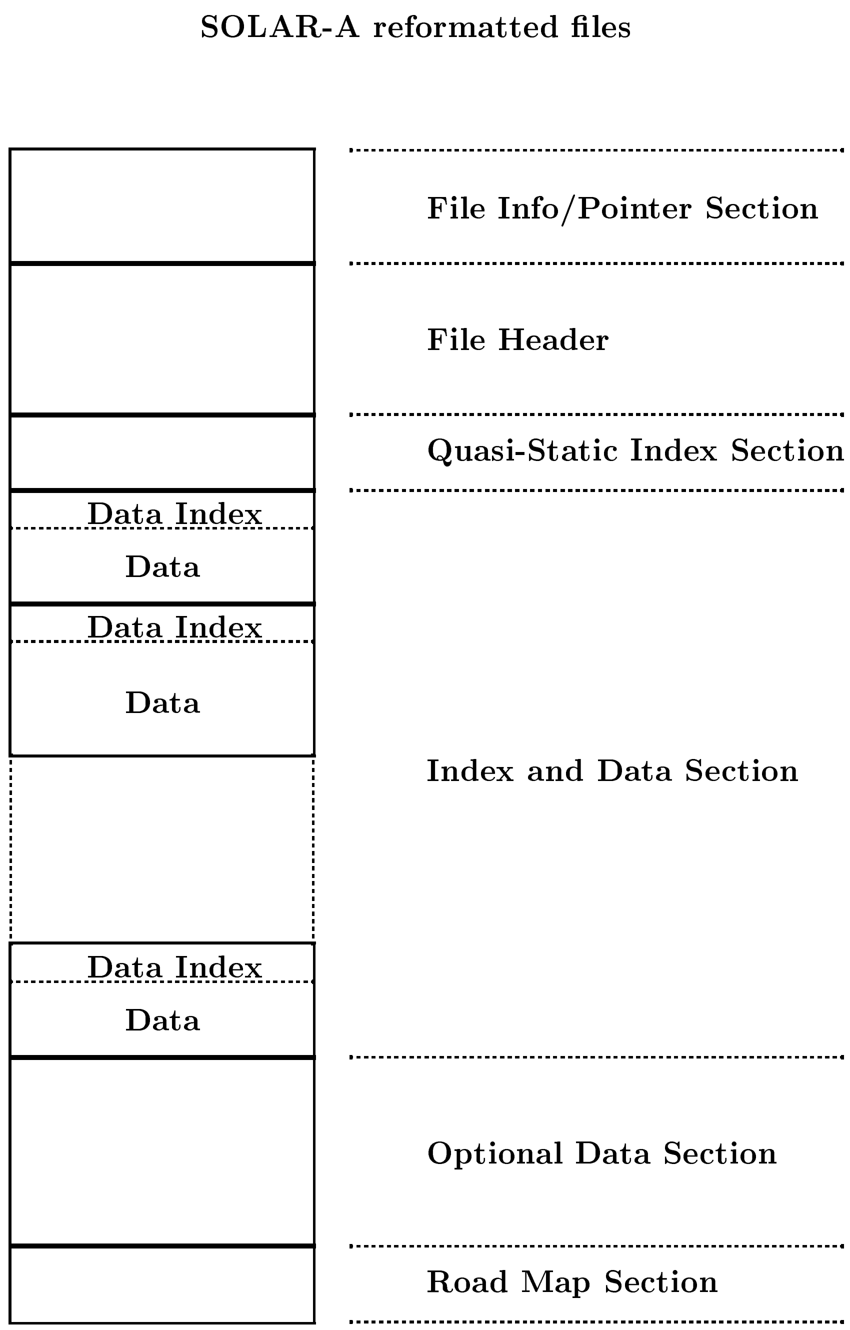

Figure B.13: Yohkoh Data File Format

B.2 Yohkoh Data File Names

B.2.1 Yohkoh Directory Logicals/Environment Variables

There is a set of VMS logicals and Unix environment variables which point to the database directories. By using these logicals in all of the access routines (along with the routine CONCAT_DIR) it is possible for software to be directly portable between VMS and Unix machines. A list of the logicals that exist for the Yohkoh database is given in Table B.1.

B.2.2 Yohkoh Prefixes

Yohkoh filesnames start with a three letter prefix. A list of currently defined prefixes is given in the Table B.2. More information about a prefix (*"ppp"*) can be found by using the IDL command:.

IDL> **help_prefix,'ppp',/full**

B.2.3 Yohkoh File IDs

The orbit file ID's are an 11 characters date format following the prefix as shown below:

```
pppYYMMDD.HHMM

where ppp  - Prefix of the file
      YY   - Year of data
      MM   - Month of data
      DD   - Day of data
      HH   - Hours
      MM   - Minutes
```

B.2.4 Yohkoh Week IDs

The Yohkoh week starts on Sunday and ends on Saturday. CAUTION: Yohkoh data taken at 00:01 (one minute after midnight) on Sunday morning is most likely in a file and tape for the previous week since the orbit started on Saturday and the first full orbit for Sunday has not started. Up to the first 90 minutes of data taken on Sunday could be stored in the previous week, depending on when the spacecraft day begins.

The weekly IDs are of the form:

```
pppYY_WWa.NN

where ppp  - Prefix of the file
      YY   - Year of data
      WW   - Week number of the data (1 to 53)
       a   - is fixed (reserved for future use)
      NN   - is the program version number which created the file
```

A list of the dates covered by each of the weeks is given in Table B.3.

B.2.5 Yohkoh Carrington IDs

The Carrington IDs are of the form:

```
ppp_crRRRRa.NN

where ppp  - Prefix of the file
      cr   - signifies a Carrington Rotation ID
      RRRR - is the rotation number
       a   - is fixed (reserved for future use)
      NN   - is the program version number which created the file
```

The starting dates for Carrington Rotations are shown in Table B.4.

Table B.1: Defined Yohkoh Enviroment Variables

DIR_GEN_ADS	DIR_SXT_ATABLES	DIR_BCS_ATODAT
DIR_GEN_ATR	DIR_SXT_CAL	DIR_BCS_BALDAT
DIR_GEN_ATT	DIR_SXT_CALIBRATE	DIR_BCS_CAL
DIR_GEN_DATA	DIR_SXT_DOC	DIR_BCS_CALDAT
DIR_GEN_DOC	DIR_SXT_ENGIN	DIR_BCS_CAT
DIR_GEN_EVN	DIR_SXT_SDC	DIR_BCS_DOC
DIR_GEN_FEM	DIR_SXT_SDL	DIR_BCS_EXE
DIR_GEN_GBE	DIR_SXT_SDW	DIR_BCS_LOGS
DIR_GEN_GBL	DIR_SXT_SENSITIVE	DIR_BCS_MICRO
DIR_GEN_GEV	DIR_SXT_SFC	DIR_BCS_SYNSPEC
DIR_GEN_GOL	DIR_SXT_SFM	
DIR_GEN_GXT	DIR_SXT_SFS	DIR_GBO_TEMP
DIR_GEN_MO	DIR_SXT_SLD	
DIR_GEN_MOVIE	DIR_SXT_SLS	
DIR_GEN_NAR	DIR_SXT_SOT	
DIR_GEN_NEL	DIR_SXT_SSL	
DIR_GEN_NTS	DIR_SXT_SSX	
DIR_GEN_OBD	DIR_SXT_SXA	
DIR_GEN_OBS	DIR_SXT_SXC	
DIR_GEN_ORBIT	DIR_SXT_SXL	
DIR_GEN_ORBIT_RAW	DIR_SXT_SXL2	
DIR_GEN_ORBIT_SOL	DIR_SXT_TABLES	
DIR_GEN_ORBIT_SW		
DIR_GEN_OSF	DIR_WBS_CAL	
DIR_GEN_OSP		
DIR_GEN_OWH	DIR_HXT_CAL	
DIR_GEN_PAN_LASER	DIR_HXT_EXE	
DIR_GEN_PNT		
DIR_GEN_SCRIPT		
DIR_GEN_SETUP		
DIR_GEN_SETUPD		
DIR_GEN_SETUPDBACK		
DIR_GEN_SPECTRA		
DIR_GEN_STATUS		
DIR_GEN_SYNOPTIC		
DIR_GEN_TAPECOPY		
DIR_GEN_TBB		
DIR_GEN_XAD		
DIR_GEN_XBD		

Table B.2: Defined Yohkoh Prefixes

Prefix	*Description*	*File Type*
ADA	S/C Attitude Raw Reformatted Data	per Orbit
ATR	Raw S/C Attitude Database	Weekly
ATT	Processed S/C Attitude Database	Weekly
BDA	BCS Raw Reformatted Data	per Orbit
BSD	*OLD* BCS Instrument Calibrated Spectra	User Specified
BPC	*OLD* BSDCAL Output (parameters)	User Specified
BTH	*OLD* BSDFIT Output (Fitted Spectra)	User Specified
BFT	*OLD* BSDFIT Output (parameters)	User Specified
BSC	*NEW* BCS Instrument Calibrated Spectra	User Specified
BSA	*NEW* Answer File for BSC Generation	User Specified
BSF	*NEW* Output from Spectral Fitting	User Specified
CBA	S/C Common Basic Raw Reformatted Data	per Orbit
EVN	Yohkoh Event Log	Weekly
FEM	Yohkoh Orbital Ephemeris	Weekly
GEV	GOES Event Log	Weekly
GXT	GOES One Minute Light Curve Data	Weekly
G6D	GOES 3 Sec Light Curve for S/C 6	Weekly
G7D	GOES 3 Sec Light Curve for S/C 7	Weekly
GXD	GOES Derived One Minute Light Curve Data	Weekly
GBE	GRO BATSE Event Log	Mission
GBL	GRO BATSE Light Curves	Weekly
G_x	Ground-Based Observation (GBO) Images (Sites)	per Image (FITS)
Gx_	Ground-Based Observation (GBO) Images (Data Type)	per Image (FITS)
GBC	GBO Big Bear Large Scale H-alpha PFI FITS Image	per Image (FITS)
GBH	GBO Big Bear H-alpha FITS Image	per Image (FITS)
GBK	GBO Big Bear Calcium FITS Image	per Image (FITS)
GBW	GBO Big Bear White Light FITS Image	per Image (FITS)
GCH	GBO Boulder Colorado H-alpha FITS Image	per Image (FITS)
GGH	GBO GSFC H-alpha FITS Image	per Image (FITS)
GHH	GBO Holloman H-alpha FITS image	per Image (FITS)
GLH	GBO Leamonth Australia H-Alpha FITS Image	per Image (FITS)
GNH	GBO NAOJ Japan H-alpha FITS Image	per Image (FITS)
GJH	GBO CRL Hiraiso H-alpha FITS Image	per Image (FITS)
GKI	GBO Kitt Peak He 10830 FITS Image	per Image (FITS)
GKM	GBO Kitt Peak Magnetogram FITS Image	per Image (FITS)
GNP	GBO Nobeyama Polarized Radio Image (17 GHz)	per Image (FITS)
GNT	GBO Nobeyama Total Intensity Radio Image (17 GHz)	per Image (FITS)
GOL	GBO Observing Log	Weekly

Prefix	*Description*	*File Type*
HDA	HXT Raw Reformatted Data	per Orbit
HXI	HXT Synthesized Image	User Specified
MDM	Magneto-Optical (MO) Map	Mission
NAR	NOAA Active Region	Weekly
NEL	Nobeyama Radio Observatory Event Log	Mission
NTS	Nobeyama Radio Observatory Time Series	Weekly
OBS	*OLD* Yohkoh Observing Log	Weekly
OBD	BCS Observing Log	Weekly
OSF	SXT Full Frame Observing Log	Weekly
OSP	SXT Partial Frame Observing Log	Weekly
OWH	WBS/HXT Observing Log	Weekly
PNT	*OLD* S/C Pointing File	Weekly
SDL	SXT Dark Current Log	Weekly
SDC	SXT Dark Current Images	Weekly (SDA)
SDW	SXT Dark Current Images for Warm CCD	Weekly (SDA)
SDP	SXT Dark Current PFI Images	Weekly (SDA)
SFD	SXT FFI Desaturated Composite Images	Weekly (SDA)
SFR	SXT FFI Raw Reformatted Data	per Orbit (SDA)
SFS	SXT FFI Special Images (diffuser, flood...)	Weekly (SDA)
SFT	SXT FFI Terminator Images	Weekly (SDA)
SFW	SXT FFI White Light Images	Weekly (SDA)
SFC	SXT FFI Calibration (Leak) Image	per Image (SDA)
SFM	SXT FFI Monthly File (daily SFD image)	Monthly (SDA)
SLS	SXT Leak Scatter Log	Weekly
SLD	SXT Leak Dark Current Log	Weekly
SPR	SXT PFI Raw Reformatted Data	per Orbit (SDA)
SOT	SXT Optical Telescope Log	Weekly
SSC	SXT Synoptic Images (centered)	Weekly (SDA)
SSE	SXT Synoptic Images (east of center)	Weekly (SDA)
SSW	SXT Synoptic Images (west of center)	Weekly (SDA)
SSL	SXT Summary Log	Weekly
SXC	SXT X-Ray Sun Center Log	Weekly
SXL	SXT X-Ray Histogram Log	Weekly
SXA	SXT X-Ray Average	Weekly
WDA	WBS Raw Reformatted Data	per Orbit
XAD	Exabyte ASCII Directory for Archive Tape	Weekly per Tape
XBD	Exabyte Binary Directory for Archive Tape	Weekly per Tape

Table B.3: Dates covered by Week IDs

Week	1991		1992		1993		1994		1995	
01	1-Jan	5-Jan	1-Jan	4-Jan	1-Jan	2-Jan	1-Jan	1-Jan	1-Jan	7-Jan
02	6-Jan	12-Jan	5-Jan	11-Jan	3-Jan	9-Jan	2-Jan	8-Jan	8-Jan	14-Jan
03	13-Jan	19-Jan	12-Jan	18-Jan	10-Jan	16-Jan	9-Jan	15-Jan	15-Jan	21-Jan
04	20-Jan	26-Jan	19-Jan	25-Jan	17-Jan	23-Jan	16-Jan	22-Jan	22-Jan	28-Jan
05	27-Jan	2-Feb	26-Jan	1-Feb	24-Jan	30-Jan	23-Jan	29-Jan	29-Jan	4-Feb
06	3-Feb	9-Feb	2-Feb	8-Feb	31-Jan	6-Feb	30-Jan	5-Feb	5-Feb	11-Feb
07	10-Feb	16-Feb	9-Feb	15-Feb	7-Feb	13-Feb	6-Feb	12-Feb	12-Feb	18-Feb
08	17-Feb	23-Feb	16-Feb	22-Feb	14-Feb	20-Feb	13-Feb	19-Feb	19-Feb	25-Feb
09	24-Feb	2-Mar	23-Feb	29-Feb	21-Feb	27-Feb	20-Feb	26-Feb	26-Feb	4-Mar
10	3-Mar	9-Mar	1-Mar	7-Mar	28-Feb	6-Mar	27-Feb	5-Mar	5-Mar	11-Mar
11	10-Mar	16-Mar	8-Mar	14-Mar	7-Mar	13-Mar	6-Mar	12-Mar	12-Mar	18-Mar
12	17-Mar	23-Mar	15-Mar	21-Mar	14-Mar	20-Mar	13-Mar	19-Mar	19-Mar	25-Mar
13	24-Mar	30-Mar	22-Mar	28-Mar	21-Mar	27-Mar	20-Mar	26-Mar	26-Mar	1-Apr
14	31-Mar	6-Apr	29-Mar	4-Apr	28-Mar	3-Apr	27-Mar	2-Apr	2-Apr	8-Apr
15	7-Apr	13-Apr	5-Apr	11-Apr	4-Apr	10-Apr	3-Apr	9-Apr	9-Apr	15-Apr
16	14-Apr	20-Apr	12-Apr	18-Apr	11-Apr	17-Apr	10-Apr	16-Apr	16-Apr	22-Apr
17	21-Apr	27-Apr	19-Apr	25-Apr	18-Apr	24-Apr	17-Apr	23-Apr	23-Apr	29-Apr
18	28-Apr	4-May	26-Apr	2-May	25-Apr	1-May	24-Apr	30-Apr	30-Apr	6-May
19	5-May	11-May	3-May	9-May	2-May	8-May	1-May	7-May	7-May	13-May
20	12-May	18-May	10-May	16-May	9-May	15-May	8-May	14-May	14-May	20-May
21	19-May	25-May	17-May	23-May	16-May	22-May	15-May	21-May	21-May	27-May
22	26-May	1-Jun	24-May	30-May	23-May	29-May	22-May	28-May	28-May	3-Jun
23	2-Jun	8-Jun	31-May	6-Jun	30-May	5-Jun	29-May	4-Jun	4-Jun	10-Jun
24	9-Jun	15-Jun	7-Jun	13-Jun	6-Jun	12-Jun	5-Jun	11-Jun	11-Jun	17-Jun
25	16-Jun	22-Jun	14-Jun	20-Jun	13-Jun	19-Jun	12-Jun	18-Jun	18-Jun	24-Jun
26	23-Jun	29-Jun	21-Jun	27-Jun	20-Jun	26-Jun	19-Jun	25-Jun	25-Jun	1-Jul
27	30-Jun	6-Jul	28-Jun	4-Jul	27-Jun	3-Jul	26-Jun	2-Jul	2-Jul	8-Jul
28	7-Jul	13-Jul	5-Jul	11-Jul	4-Jul	10-Jul	3-Jul	9-Jul	9-Jul	15-Jul
29	14-Jul	20-Jul	12-Jul	18-Jul	11-Jul	17-Jul	10-Jul	16-Jul	16-Jul	22-Jul
30	21-Jul	27-Jul	19-Jul	25-Jul	18-Jul	24-Jul	17-Jul	23-Jul	23-Jul	29-Jul
31	28-Jul	3-Aug	26-Jul	1-Aug	25-Jul	31-Jul	24-Jul	30-Jul	30-Jul	5-Aug
32	4-Aug	10-Aug	2-Aug	8-Aug	1-Aug	7-Aug	31-Jul	6-Aug	6-Aug	12-Aug
33	11-Aug	17-Aug	9-Aug	15-Aug	8-Aug	14-Aug	7-Aug	13-Aug	13-Aug	19-Aug
34	18-Aug	24-Aug	16-Aug	22-Aug	15-Aug	21-Aug	14-Aug	20-Aug	20-Aug	26-Aug
35	25-Aug	31-Aug	23-Aug	29-Aug	22-Aug	28-Aug	21-Aug	27-Aug	27-Aug	2-Sep
36	1-Sep	7-Sep	30-Aug	5-Sep	29-Aug	4-Sep	28-Aug	3-Sep	3-Sep	9-Sep
37	8-Sep	14-Sep	6-Sep	12-Sep	5-Sep	11-Sep	4-Sep	10-Sep	10-Sep	16-Sep
38	15-Sep	21-Sep	13-Sep	19-Sep	12-Sep	18-Sep	11-Sep	17-Sep	17-Sep	23-Sep
39	22-Sep	28-Sep	20-Sep	26-Sep	19-Sep	25-Sep	18-Sep	24-Sep	24-Sep	30-Sep
40	29-Sep	5-Oct	27-Sep	3-Oct	26-Sep	2-Oct	25-Sep	1-Oct	1-Oct	7-Oct
41	6-Oct	12-Oct	4-Oct	10-Oct	3-Oct	9-Oct	2-Oct	8-Oct	8-Oct	14-Oct
42	13-Oct	19-Oct	11-Oct	17-Oct	10-Oct	16-Oct	9-Oct	15-Oct	15-Oct	21-Oct
43	20-Oct	26-Oct	18-Oct	24-Oct	17-Oct	23-Oct	16-Oct	22-Oct	22-Oct	28-Oct
44	27-Oct	2-Nov	25-Oct	31-Oct	24-Oct	30-Oct	23-Oct	29-Oct	29-Oct	4-Nov
45	3-Nov	9-Nov	1-Nov	7-Nov	31-Oct	6-Nov	30-Oct	5-Nov	5-Nov	11-Nov
46	10-Nov	16-Nov	8-Nov	14-Nov	7-Nov	13-Nov	6-Nov	12-Nov	12-Nov	18-Nov
47	17-Nov	23-Nov	15-Nov	21-Nov	14-Nov	20-Nov	13-Nov	19-Nov	19-Nov	25-Nov
48	24-Nov	30-Nov	22-Nov	28-Nov	21-Nov	27-Nov	20-Nov	26-Nov	26-Nov	2-Dec
49	1-Dec	7-Dec	29-Nov	5-Dec	28-Nov	4-Dec	27-Nov	3-Dec	3-Dec	9-Dec
50	8-Dec	14-Dec	6-Dec	12-Dec	5-Dec	11-Dec	4-Dec	10-Dec	10-Dec	16-Dec
51	15-Dec	21-Dec	13-Dec	19-Dec	12-Dec	18-Dec	11-Dec	17-Dec	17-Dec	23-Dec
52	22-Dec	28-Dec	20-Dec	26-Dec	19-Dec	25-Dec	18-Dec	24-Dec	24-Dec	30-Dec
53	29-Dec	31-Dec	27-Dec	31-Dec	26-Dec	31-Dec	25-Dec	31-Dec	31-Dec	31-Dec

Week	*1996*		*1997*		*1998*		*1999*		*2000*	
01	1-Jan	6-Jan	1-Jan	4-Jan	1-Jan	3-Jan	1-Jan	2-Jan	1-Jan	6-Jan
02	7-Jan	13-Jan	5-Jan	11-Jan	4-Jan	10-Jan	3-Jan	9-Jan	7-Jan	13-Jan
03	14-Jan	20-Jan	12-Jan	18-Jan	11-Jan	17-Jan	10-Jan	16-Jan	14-Jan	20-Jan
04	21-Jan	27-Jan	19-Jan	25-Jan	18-Jan	24-Jan	17-Jan	23-Jan	21-Jan	27-Jan
05	28-Jan	3-Feb	26-Jan	1-Feb	25-Jan	31-Jan	24-Jan	30-Jan	28-Jan	3-Feb
06	4-Feb	10-Feb	2-Feb	8-Feb	1-Feb	7-Feb	31-Jan	6-Feb	4-Feb	10-Feb
07	11-Feb	17-Feb	9-Feb	15-Feb	8-Feb	14-Feb	7-Feb	13-Feb	11-Feb	17-Feb
08	18-Feb	24-Feb	16-Feb	22-Feb	15-Feb	21-Feb	14-Feb	20-Feb	18-Feb	24-Feb
09	25-Feb	2-Mar	23-Feb	1-Mar	22-Feb	28-Feb	21-Feb	27-Feb	25-Feb	2-Mar
10	3-Mar	9-Mar	2-Mar	8-Mar	1-Mar	7-Mar	28-Feb	6-Mar	3-Mar	9-Mar
11	10-Mar	16-Mar	9-Mar	15-Mar	8-Mar	14-Mar	7-Mar	13-Mar	10-Mar	16-Mar
12	17-Mar	23-Mar	16-Mar	22-Mar	15-Mar	21-Mar	14-Mar	20-Mar	17-Mar	23-Mar
13	24-Mar	30-Mar	23-Mar	29-Mar	22-Mar	28-Mar	21-Mar	27-Mar	24-Mar	30-Mar
14	31-Mar	6-Apr	30-Mar	5-Apr	29-Mar	4-Apr	28-Mar	3-Apr	31-Mar	6-Apr
15	7-Apr	13-Apr	6-Apr	12-Apr	5-Apr	11-Apr	4-Apr	10-Apr	7-Apr	13-Apr
16	14-Apr	20-Apr	13-Apr	19-Apr	12-Apr	18-Apr	11-Apr	17-Apr	14-Apr	20-Apr
17	21-Apr	27-Apr	20-Apr	26-Apr	19-Apr	25-Apr	18-Apr	24-Apr	21-Apr	27-Apr
18	28-Apr	4-May	27-Apr	3-May	26-Apr	2-May	25-Apr	1-May	28-Apr	4-May
19	5-May	11-May	4-May	10-May	3-May	9-May	2-May	8-May	5-May	11-May
20	12-May	18-May	11-May	17-May	10-May	16-May	9-May	15-May	12-May	18-May
21	19-May	25-May	18-May	24-May	17-May	23-May	16-May	22-May	19-May	25-May
22	26-May	1-Jun	25-May	31-May	24-May	30-May	23-May	29-May	26-May	1-Jun
23	2-Jun	8-Jun	1-Jun	7-Jun	31-May	6-Jun	30-May	5-Jun	2-Jun	8-Jun
24	9-Jun	15-Jun	8-Jun	14-Jun	7-Jun	13-Jun	6-Jun	12-Jun	9-Jun	15-Jun
25	16-Jun	22-Jun	15-Jun	21-Jun	14-Jun	20-Jun	13-Jun	19-Jun	16-Jun	22-Jun
26	23-Jun	29-Jun	22-Jun	28-Jun	21-Jun	27-Jun	20-Jun	26-Jun	23-Jun	29-Jun
27	30-Jun	6-Jul	29-Jun	5-Jul	28-Jun	4-Jul	27-Jun	3-Jul	30-Jun	6-Jul
28	7-Jul	13-Jul	6-Jul	12-Jul	5-Jul	11-Jul	4-Jul	10-Jul	7-Jul	13-Jul
29	14-Jul	20-Jul	13-Jul	19-Jul	12-Jul	18-Jul	11-Jul	17-Jul	14-Jul	20-Jul
30	21-Jul	27-Jul	20-Jul	26-Jul	19-Jul	25-Jul	18-Jul	24-Jul	21-Jul	27-Jul
31	28-Jul	3-Aug	27-Jul	2-Aug	26-Jul	1-Aug	25-Jul	31-Jul	28-Jul	3-Aug
32	4-Aug	10-Aug	3-Aug	9-Aug	2-Aug	8-Aug	1-Aug	7-Aug	4-Aug	10-Aug
33	11-Aug	17-Aug	10-Aug	16-Aug	9-Aug	15-Aug	8-Aug	14-Aug	11-Aug	17-Aug
34	18-Aug	24-Aug	17-Aug	23-Aug	16-Aug	22-Aug	15-Aug	21-Aug	18-Aug	24-Aug
35	25-Aug	31-Aug	24-Aug	30-Aug	23-Aug	29-Aug	22-Aug	28-Aug	25-Aug	31-Aug
36	1-Sep	7-Sep	31-Aug	6-Sep	30-Aug	5-Sep	29-Aug	4-Sep	1-Sep	7-Sep
37	8-Sep	14-Sep	7-Sep	13-Sep	6-Sep	12-Sep	5-Sep	11-Sep	8-Sep	14-Sep
38	15-Sep	21-Sep	14-Sep	20-Sep	13-Sep	19-Sep	12-Sep	18-Sep	15-Sep	21-Sep
39	22-Sep	28-Sep	21-Sep	27-Sep	20-Sep	26-Sep	19-Sep	25-Sep	22-Sep	28-Sep
40	29-Sep	5-Oct	28-Sep	4-Oct	27-Sep	3-Oct	26-Sep	2-Oct	29-Sep	5-Oct
41	6-Oct	12-Oct	5-Oct	11-Oct	4-Oct	10-Oct	3-Oct	9-Oct	6-Oct	12-Oct
42	13-Oct	19-Oct	12-Oct	18-Oct	11-Oct	17-Oct	10-Oct	16-Oct	13-Oct	19-Oct
43	20-Oct	26-Oct	19-Oct	25-Oct	18-Oct	24-Oct	17-Oct	23-Oct	20-Oct	26-Oct
44	27-Oct	2-Nov	26-Oct	1-Nov	25-Oct	31-Oct	24-Oct	30-Oct	27-Oct	2-Nov
45	3-Nov	9-Nov	2-Nov	8-Nov	1-Nov	7-Nov	31-Oct	6-Nov	3-Nov	9-Nov
46	10-Nov	16-Nov	9-Nov	15-Nov	8-Nov	14-Nov	7-Nov	13-Nov	10-Nov	16-Nov
47	17-Nov	23-Nov	16-Nov	22-Nov	15-Nov	21-Nov	14-Nov	20-Nov	17-Nov	23-Nov
48	24-Nov	30-Nov	23-Nov	29-Nov	22-Nov	28-Nov	21-Nov	27-Nov	24-Nov	30-Nov
49	1-Dec	7-Dec	30-Nov	6-Dec	29-Nov	5-Dec	28-Nov	4-Dec	1-Dec	7-Dec
50	8-Dec	14-Dec	7-Dec	13-Dec	6-Dec	12-Dec	5-Dec	11-Dec	8-Dec	14-Dec
51	15-Dec	21-Dec	14-Dec	20-Dec	13-Dec	19-Dec	12-Dec	18-Dec	15-Dec	21-Dec
52	22-Dec	28-Dec	21-Dec	27-Dec	20-Dec	26-Dec	19-Dec	25-Dec	22-Dec	28-Dec
53	29-Dec	31-Dec	28-Dec	31-Dec	27-Dec	31-Dec	26-Dec	31-Dec	29-Dec	31-Dec

Table B.4: Start Dates of Carrington Rotations

Carrington Rotation	*Starting Date*	*Carrington Rotation*	*Starting Date*	*Carrington Rotation*	*Starting Date*
1846	21-Aug-91	1888	10-Oct-94	1930	29-Nov-97
1847	18-Sep-91	1889	6-Nov-94	1931	26-Dec-97
1848	15-Oct-91	1890	3-Dec-94	1932	22-Jan-98
1849	11-Nov-91	1891	31-Dec-94	1933	18-Feb-98
1850	8-Dec-91	1892	27-Jan-95	1934	18-Mar-98
1851	5-Jan-92	1893	23-Feb-95	1935	14-Apr-98
1852	1-Feb-92	1894	23-Mar-95	1936	11-May-98
1853	28-Feb-92	1895	19-Apr-95	1937	7-Jun-98
1854	26-Mar-92	1896	16-May-95	1938	5-Jul-98
1855	23-Apr-92	1897	12-Jun-95	1939	1-Aug-98
1856	20-May-92	1898	10-Jul-95	1940	28-Aug-98
1857	16-Jun-92	1899	6-Aug-95	1941	25-Sep-98
1858	14-Jul-92	1900	2-Sep-95	1942	22-Oct-98
1859	10-Aug-92	1901	29-Sep-95	1943	18-Nov-98
1860	6-Sep-92	1902	27-Oct-95	1944	15-Dec-98
1861	3-Oct-92	1903	23-Nov-95	1945	12-Jan-99
1862	31-Oct-92	1904	20-Dec-95	1946	8-Feb-99
1863	27-Nov-92	1905	17-Jan-96	1947	7-Mar-99
1864	24-Dec-92	1906	13-Feb-96	1948	3-Apr-99
1865	20-Jan-93	1907	11-Mar-96	1949	1-May-99
1866	17-Feb-93	1908	7-Apr-96	1950	28-May-99
1867	16-Mar-93	1909	5-May-96	1951	24-Jun-99
1868	12-Apr-93	1910	1-Jun-96	1952	22-Jul-99
1869	10-May-93	1911	28-Jun-96	1953	18-Aug-99
1870	6-Jun-93	1912	26-Jul-96	1954	14-Sep-99
1871	3-Jul-93	1913	22-Aug-96	1955	11-Oct-99
1872	30-Jul-93	1914	18-Sep-96	1956	8-Nov-99
1873	27-Aug-93	1915	15-Oct-96	1957	5-Dec-99
1874	23-Sep-93	1916	12-Nov-96	1958	1-Jan-00
1875	20-Oct-93	1917	9-Dec-96	1959	29-Jan-00
1876	17-Nov-93	1918	5-Jan-97	1960	25-Feb-00
1877	14-Dec-93	1919	1-Feb-97	1961	23-Mar-00
1878	10-Jan-94	1920	1-Mar-97	1962	19-Apr-00
1879	6-Feb-94	1921	28-Mar-97	1963	17-May-00
1880	6-Mar-94	1922	24-Apr-97	1964	13-Jun-00
1881	2-Apr-94	1923	22-May-97	1965	10-Jul-00
1882	29-Apr-94	1924	18-Jun-97	1966	6-Aug-00
1883	26-May-94	1925	15-Jul-97	1967	3-Sep-00
1884	23-Jun-94	1926	11-Aug-97	1968	30-Sep-00
1885	20-Jul-94	1927	8-Sep-97	1969	27-Oct-00
1886	16-Aug-94	1928	5-Oct-97	1970	24-Nov-00
1887	13-Sep-94	1929	1-Nov-97	1971	21-Dec-00

C Accessing Yohkoh files

C.1 How to Read Yohkoh Archive Tapes (GO_RDTAP)

The routine GO_RDTAP allows access to the Yohkoh archive tapes. Normally the archive data are written on Exabyte (8 mm) or DAT (4 mm) tapes. There is one archive tape for each week of data. The first step to accessing archive data are to determine which tape it is on. Use the WeekID table shown in the Reference Guide to figure out what tape you need. For example, if you wish to look at data on 27-Mar-92 you first determine that it is week 13 in 1992, so the weekID is 92_13. You can use the routine TIM2ORBIT with the */print* option to get the week number. Put that tape into the tape drive.

You need to move to the directory where you want the data to be written.

GO_RDTAP is a menu driven program to retrieve data from the archive tapes via a menu driven program. The ISAS and LPARL machines have a copy of the directory listing for each tape saved on the disk. This means that you can simply type:

IDL> **go_rdtap**

and it will list all of the tapes that have been made. You need to select the tape that you are reading. If your system does not have the XBD files on-line, then type:

IDL> **go_rdtap, /tape**

after you have loaded the tape, and the program will read the directory listing from the tape.

1. A menu will appear with a series of weekly log files. If you do not need any of the weekly files, then just select QUIT/EXIT.
2. The next menu will be the file prefixes that are available. Click on each file prefix that you wish to copy to the disk.
3. When you have selected all of the prefixes you need, click on QUIT/EXIT.
4. The final menu is the FileIDs that are on the tape. Select the fileIDs you need, and then click on QUIT/EXIT.
5. You will be asked whether you are ready to read the tape. GO_RDTAP will also display the number of kilobytes that you have selected to read, so check the available space on the disk selected before continuing.

If the */ydb* switch is used on the call to GO_RDTAP, then the weekly data files will be copied to their proper directories (the default is to copy them to the current directory).

See the Reference Guide appendix for more detailed information on the archive tapes.

C.2 How to Read Yohkoh Database Tapes (RD_YDBTAP)

RD_YDBTAP reads and installs Yohkoh Data Base (ydb) files from the distributed "ydb" tapes. A "ydb" tape contains data files which are used by the Yohkoh analysis system. These may include both Yohkoh specific data (e.g., Yohkoh pointing, ephemeris, observing logs, etc.) and supporting data sets (GOES, BATSE). Although some of the smaller database (ydb) sets are available as compressed tar[7] files and are included in the installation and upgrade procedure,

[7] *tar* is the Unix backup utility

the ydb tapes are the recommended method for distribution of large database sets (such as pointing data and observing log files).

RD_YDBTAP Functions:

- Provide graphical interface which allows user to specify database sets for extraction; selection may be by ydb file type and/or times.
- Using ydb tape directory file, compare directory contents with user requests and flag any discrepancies.
- If output directory is not specified, use site specified Yokhoh environment to place files in proper locations.
- Optionally create required directories which do not exist.
- Using directory file information on file sizes and local output directory status, warn user if there is insufficient disk space to satisfy requests.

The following example will allow the user how to interactively select times and data sets for extraction and installation

IDL> **rd_ydbtap**

Several options are available on input – if files are fully specified, no additional user interaction is required. The next example would extract and install all data base files for the specified week using tape drive one.

IDL> **rd_ydbtap, dev=1, weekid='92_45', /all**

RD_YDBTAP may function to "boot up" a remote system prior to implementation of a ydb network upgrade system. Due to the potential of adversely impacting the network, a network-based ydb upgrade system should be coordinated with the Yohkoh SW managers. They can provide some Unix scripts which assist in this task and allow some scheduling control. For more information on network upgrades of the Yohkoh data base, send e-mail to:

```
software@isass0.solar.iasa.ac.jp, SUBJ="ydb via network"
```

C.3 FTP Access to Yohkoh Archive, etc...

D Using the Yohkoh Team Bulletin Board (TBB)

D.1 Overview

The Team Bulletin Board (TBB) was introduced as a mechanism to encourage the Yohkoh data users to openly exchange information on their research in the form of proposals. Once scientists think up a research plan using the Yohkoh data, they are requested to write a short proposal. The TBB keeps all the on-going research projects that utilize the Yohkoh data. By accessing the TBB, one finds what projects are now vigorously undertaken by who, and may either think of a project nobody is working on, or one that is similar to existing proposals. In the latter case, ideally, the Data Use Coordinators (DUCs, see below) may make certain suggestions in order not to jeopardize the proposals that have already been started. It is important that an on-going project initiated by a graduate student in particular should not be taken away by a more experienced scientist.

The lead authors of proposals are requested to report on what they have found every time major progress is made. Without such a report, the proposal will be regarded as not on-going and will automatically disappear six months after the submission. Now, as the data are put into public domain, the Yohkoh team invites every scientist who uses the Yohkoh data to participate in this system.

Every project that will become part of somebody's master's or PhD thesis is included in one sub-directory, that is separate from the general projects. The TBB also offers addresses of the scientists who have submitted proposals and meeting and travel reports.

D.2 Access to the TBB

The TBB may be accessed by typing:

IDL> **teambb**

The banner page appears, showing the most recent proposals, updated and deleted ones. Then in the main menu, one can choose either scientific proposals or other information such as addresses. If "sci" is selected, there are three ways of getting a proposal on the screen — search on the authors, on the category or on the summary file that shows the titles and lead authors.

One can also type:

IDL> **teambb,'sci'**

to skip the banner page and the main menu and to enter the science menu. Such flexibilities will be enhanced in future updates of TEAMBB.

D.3 How to submit a new proposal

Although proposals in the past have had loose guidelines as to the format, it is encouraged that a new proposal conform to the following for a more efficient use of the TBB. A proposal should be concise but at the same time should contain the following information:

1. Title
2. Collaboration (list of scientists working on the problem)
3. Primary science question being addressed

4. Method
5. Required data (i.e. date and time of the phenomenon)

All these should fit in to no more than a half page (roughly 80 characters and 20 lines). Once completed, the proposal should be sent to the relevant DUCs. Currently, they are:

- Takeo Kosugi (kosugi@laputa.mtk.nao.ac.jp) for HXT,
- Saku Tsuneta (tsuneta@flare2.solar.isas.ac.jp) and
  Keith Strong (strong@sag.space.lockheed.com) for SXT,
- Tetsuya Watanabe (watanabe@uvlab.mtk.nao.ac.jp) and
  Andrzej Fludra (af@41887.dnet.nasa.gov) for BCS,
- Masato Yoshimori (yosimori@ax251.rikkyo.ac.jp) for WBS

It is advisable to send the proposal to multiple DUCs in case of network trouble. Once the DUCs find that the proposal has no problem in terms of jeopardizing on-going proposals, it will be forwarded to TBB administrators (currently Kazunarai Shibata and Nariaki Nitta) who will post it on the TBB in a timely fashion and keep it there for six months.

D.4 How to Update a Proposal

Send the progress report to `tbb_adm@isass0.solar.isas.ac.jp`. Do not send it to the DUCs. The reports will be posted in the TBB. However, if the proposal has expired before the update, send it to DUCs. They will decide whether the TBB administrators can reinstate the original proposal.

D.5 Where to Find TeamBB Files

All the TBB files are kept under $DIR_GEN_TBB (which is /ys/gen/tbb on Unix machines) and are FTP-able. There are four subdirectories. The most important to scientists is the proposal subdirectory, which in turn has several subdirectories. The current proposals are kept under current. They have file names like ar210.hudson33. This means that the proposal is the 210th one in the category of active regions and the 33rd by Hudson. Other subdirectories under proposal contain, among other things, information on completed projects, summary files and thesis topics.

In addition to the proposal subdirectory, there are three others, address, help and news. The last one has various information such as reports on meetings and campaign announcements.

E Old Fortran 77 Analysis Software

E.1 Calibrating and Fitting BCS Spectra (the BSD software)

This software consists of three programs written in standard FORTRAN-77 that are used to reduce the BDA data to fitted spectral parameters, together with a number of IDL display routines. This software has been made largely obsolete by the development of the IDL-based BSC software discussed earlier. It is no longer being actively supported by BCS personnel. The FORTRAN programs are MKBSD, BSDCAL and BSDFIT:

MKBSD extracts the spectra, applies instrument corrections and writes a new file, the BSD file.

BSDCAL fits selected lines in the spectra of the BSD file with Voigt profiles to produce a calibrated wavelength scale and writes this fit information to a BPC file.

BSDFIT fits a full theory spectrum to the calibrated BSD data to give electron temperature, emission measure and plasma velocity; the BPC file provides the first guess for the fitting program. The resulting fit parameters are written to a BFT file; the output theory spectra are written to the BTH file.

These programs can be run at the operating-system level using control files created either by using the setup sections of the programs (in the case of MKBSD and BSDCAL), or with the editor. It should be noted, however, that the differences between VMS, Ultrix, SunOS, and the other Unix-based operating systems used at Yohkoh sites complicate matters and the user may find it easier to run the programs from IDL, using BCSPRO. To do this:

1. Select a time interval by either:
   (a) Using the cursors in a -44x option in YODAT
   (b) Running SELECT_24HR_PLOT after running BCS_24HR_PLOT.

   IDL> **.run select_24hr_plot**

2. Execute BCSPRO to create the control files, and run the programs.

   IDL> **.run bcspro**

3. It is sometimes necessary to edit the BPC file because the fits made by BSDCAL sometimes "fail":
   (a) Enter IDL and run ED_BPC to edit the required parts of the BPC file.

   IDL> **.run ed_bpc**

   (b) Again run the BSDFIT program

If more subtle variations of the code options are required, after creation the control files may be modified either using the setup sections of the programs or by using the editor — the programs should then be re-run. For more information on these programs, see the various documents on $DIR_BCS_DOC.

Under the different operating systems, different executables are needed. The BCSPRO routine takes care of this, but the user should be aware of the different names if running outside of IDL. The task names under the different operating systems are:

```
Name            VMS             Unix
MKBSD           MKBSD           $DIR_BCS_EXE/mkbsd_x
BSDCAL          BSDCAL          $DIR_BCS_EXE/bsdcal_x
BSDFIT          BSDFIT          $DIR_BCS_EXE/bsdfit_x
```

Note: Within Unix: the "_x" should be replaced by "_u" under Ultrix; "_s" under Sun/OS; and "_m" on the MIPS. Since MKBSD needs certain environment variables, this is normally run by executing "$DIR_BCS_EXE/run_mkbsd" which temporarily creates these, and automatically selects the correct task for the current machine.

A BSD file may be read into IDL with GET_BSD. After modification a new file may be written with WR_BSD – see section A.2.

IT SHOULD BE NOTED that the BSD files are not transferrable between machines and operating systems. Even though it would be possible (using IDL) to make them transferrable, since the normal route for generating the files using the FORTRAN-77 code does not produce transferrable files, it has been decided that the IDL code would follow the same restriction. The BSC files (see section 3.3), which are generated under IDL, are transferrable.

E.1.1 Displaying Data from the BSD File

The light curve from the BSD file may be plotted with LCBSD.

IDL> **lcbsd,filename**

The light curve and spectra from the BSD file may be plotted by PLOTBSD.

IDL> **plotbsd,filename**

E.1.2 Displaying Intermediate Results (from BSDCAL)

The fit results written to the BPC file by BSDCAL, can be over-plotted on the BSD spectra using PLOTBPC.

IDL> **plotbpc [,bpc_file,chan,option,start]**

E.1.3 Displaying Fitted Spectral Data (from BSDFIT)

The theoretical spectra written to the BTH file by BSDFIT, can be over-plotted on the BSD spectra using PLOTBTH (was PLOTSPC). This program is interactive.

IDL> **plotbth,filename**

E.1.4 Displaying Processed Parameters (from BSDFIT)

The parameters stored in the BFT file (created by BSDFIT), such as emission measures and electron temperatures, may be plotted against time by PLOTBFT.

IDL> **plotbft,filename [,chan=n [,/plot_page, etc.]]**

E.2 Synthesizing HXT Images (HXTPRO; Sakao)

Following is a description on the HXT image synthesis using the FORTRAN programs with an IDL user interface. The programs, which make use of the Maximum Entropy Image restoration procedure, were originally developed on FACOM mainframe computers at ISAS. At present, the image synthesis is available on SUN machines. Neither MIPS nor DEC machines are supported yet.

1. After extracting HXT data using YODAT and obtaining the flare location, type in the following command:

   IDL> **.run hxtpro**

2. `* Execute HXT MOVIE setup code:  [Default:  Y ] ?`

   Choose 'Y' for the above prompt. This step enables you to choose

   - HXT energy band in which images are synthesized (step (3)),
   - background data interval (step (4)), and
   - flare data interval (step (5))

   on your CRT display. If you choose 'N', then steps (3), (4), and (5) are skipped.

3. `* Enter HXT Channel to use in Image synthesis:  [Default:  0 ]`

   Enter either of 0,1,2, or 3 (0=L, 1=M1, 2=M2, 3=H bands, respectively).

4. There appears an HXT time profile in the energy band specified in the step (3). Choose background a data interval by clicking mouse buttons. See messages which appear in your IDL window for details. After choosing BGD interval, there is a message:

   ```
   Write Comments for this data structure >
   ```

   Simply press return if you do not have anything special to comment for the selected BGD data, which is stored in a variable named *bkgd*. The BGD data thus selected are also contained in a file named "flare.bgd" on the current directory. This file can be read using the following command even if you exit from IDL and re-enter:

   IDL> **rd_hxtdat,bkgd,'flare.bgd'**

5. There appears another HXT time profile. Choose flare data interval of which you want to make an image. The procedure is just the same as in step (4). The selected flare data are stored in a variable named *flare*. Background subtraction is conducted in this step. The BGD-subtracted data are stored in a variable *hxt* and is also stored in a file "hxt_trn.dat" on the current directory. This file can be read by the same command as above.

   IMPORTANT NOTICE:
   When choosing a flare data interval, it is recommended not to accumulate too many X-ray counts; usually accumulated hard X-ray counts of e.g. 100–500 cts/SC are enough for

synthesizing an image (depending on profiles of hard X-ray sources etc.). Suppose you are analyzing flare data in flare-high mode. Each data point in the time profile corresponds to 2-s of data. As the unit of Y axis is cts/s/SC, if you choose one data point which has a Y value of 100, it means that you have accumulated 100 cts/s/SC $\times$ 2 s = 200 cts/SC, which is already enough for making an image. Even if you accumulate larger counts than the above values, the image synthesis program can somehow make images, but with increased possibility that an iteration procedure in the program would not converge sufficiently. If the accumulated counts are high, there appears a warning message saying:

```
% \$MAIN\$: ** Warning: Recommended count level is 100 < total , 200
* Is this O.K.:  [Default: N ] ?
```

6. The Image Synthesis program uses a JCL-style control file

   ```
   * Create JCL file  [Default: Y ] ?
   ```

   Simply type-in 'Y' if it is the first time for you to synthesize an image of the flare which you are now working on.

7. `* Enter HXT Flare Coordinates, FOV Exp.  Fact.:  [Default:  (None) ]`

   Type-in HXT flare location (x0,y0) and the size of the HXT image synthesis field-of-view. The location (x0,y0) is given in section 4.3.1. The default size of the HXT FOV is 1.0, which provides us with a resultant image array of 64×64 pixels with the same pixel size as SXT, thus making it easy to compare HXT images with SXT images. If you type 2.0 for the HXT FOV size, it means that you expand the HXT image synthesis FOV by a factor of 2.0; you will obtain an HXT image of 64×64 pixels with its pixel size twice as large as the SXT one. For example, in the case of 15 November, 1991 flare, *xy0* obtained in section 4.3.1 is xy0 = (−2.09,−4.84) and in the nominal case type

   ```
   -2.09, -4.84, 1.0
   ```

   for the above prompt.

8. `* Enter Comment:`

   Write comments (if any) for the current image synthesis. After this step, a control file (JCL file) for the FORTRAN image synthesis programs "hxtmve.jcl" is written on the current directory.

9. `* Execute HXT MOVIE task:  [Default:  Y ] ?`

   Type 'Y' for starting image synthesis procedure. The procedure typically takes 30 s - 5 min to make an image. The synthesized HXT image as well as some other information such as start and end times of data accumulation are contained in a file named "hxt_img.dat" on the current directory. The image file can be read-in and the synthesized image is displayed by the following commands:

IDL> **rd_hxtimg,img,hdr,inf,'hxt_img.dat'** ; (read-in image file)
IDL> **disp_hxtimg,img,hdr,inf** ; (display)

10. `* Convert output to HXI file: [Default: N ] ?`

    An HXI file is produced after typing-in 'Y' for the above prompt.

11. `* Read and Display Image: [Default: Y ] ?`

    If you type-in 'Y', the synthesized image is displayed using DISP_HXTIMG.

F Web Version of the YAG

The **User Guide** and **Instrument Guide** of the **Yohkoh Analysis Guide (YAG)** may be viewed as **Web documents** on the URL's:

- **http://diapason.lmsal.com/~bentley/guides/yag/**
- **http://umbra.nascom.nasa.gov/~bentley/guides/yag/**
- **http://ydac.mssl.ucl.ac.uk/guides/yag/**

PostScript versions of the **User Guide** and **Instrument Guide** of the **YAG** are available at each site - they are designed to be printed double-sided. If you want a printed copy of the YAG, the PostScript version is recommended since the screen representation of special characters in the Hypertext version does not produce good printed output.

This version of the **YAG** was prepared at the **YDAC** (located at the Mullard Space Science Laboratory, University College London) by **Bob Bentley**. The Hypertext version was translated from LaTeX using **TtH**, and further formatted using **IDL**.