SPIROS

Documentation of the SPIROS software package for imaging, source location, spectral extraction and timing analysis using observation data from the INTEGRAL spectrometer SPI

P.H.Connell
University of Valencia

Version 5.0 - 8th June 2005

Contents

1 Introduction to SPIROS software

SPIROS is an imaging, spectral and timing analysis software package whose broad aim is to take observation exposure data, derived from event data returned from the spectrometer SPI on board INTEGRAL, and process it to output a catalogue of all sources located, emission images of the field of view observed along with source spectra and light curves.

For the user the main requirement is to have the input observation data of interest along with an Instrument Response Function or **IRF** which describes what SPI "sees" and then prepare a **parameter file** of control parameters for input to **SPIROS** before running it.

These parameters specify where all input datasets are to be found, the names of output datasets along with others to describe the size of the output image field of view and to control just what SPIROS is to search for and what do with it.

Entering the correct values of these parameters will be the most important problem any user will face in using SPIROS but they will be described fully in this document and examples are provided. The simple fact about **SPIROS** is to remember that the user must only prepare an input parameter file, for which a GUI interface is also forseen, and start it running.

New users are advised to first take a look in the **cookbook of examples** which provides examples of **SPIROS** output which they can view and examine. It also provides datasets containing already prepared observation data and input parameters which can be downloaded from the ISDC and used to execute SPIROS and see what it does. The user can change any input parameters to see what effect it has on **SPIROS** output or use them as templates for their own particular tasks.

SPIROS-8.1 and above has two important changes, the main one being in timing analysis mode allowing the user to specify some cyclic models for each source via its parameters in the input source catalogue. The second is the substitution of the NAG subroutine **E04NCF**, to solve a constrained QP problem, by a simpler one from the University of Valencia to provide only a positivity constraint.

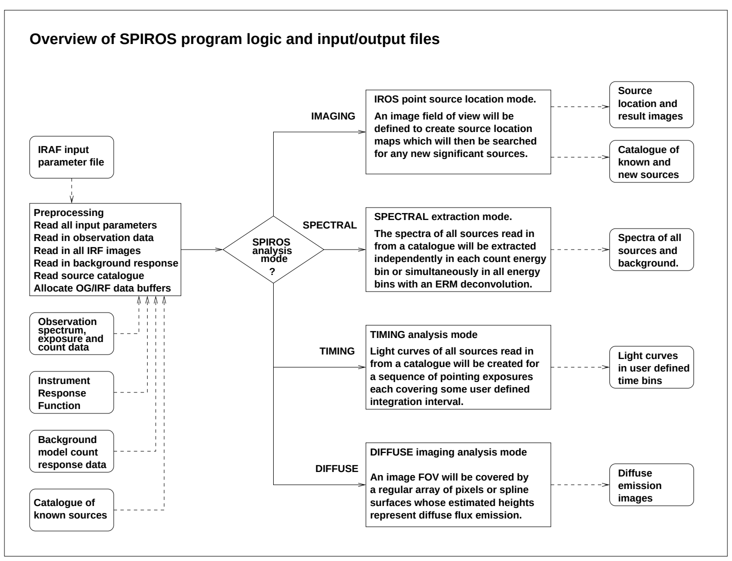


Figure 1: Overview of SPIROS functionality and input/output datsets

2 Overview of SPIROS software

2.1 Overview of the aims of SPIROS software.

The primary purpose of **SPIROS** is to use observation exposure data to locate **point** or **pointlike** sources in the observation field of view and output a catalogue of their parameters along with images of them, their spectra and any flux variablity in time.

The original purpose of SPIROS was only to create images and a catalogue of all point sources located but this was extended to use such a catalogue as input to calculate and output the "raw" spectrum of each source in it for later processing by the spectral deconvolution and modelling package **XSPEC**. It was then later extended to include the possibility of some timing analysis to determine any source flux variability in time.

The result of this history is that SPIROS has three basic operating modes to choose from:

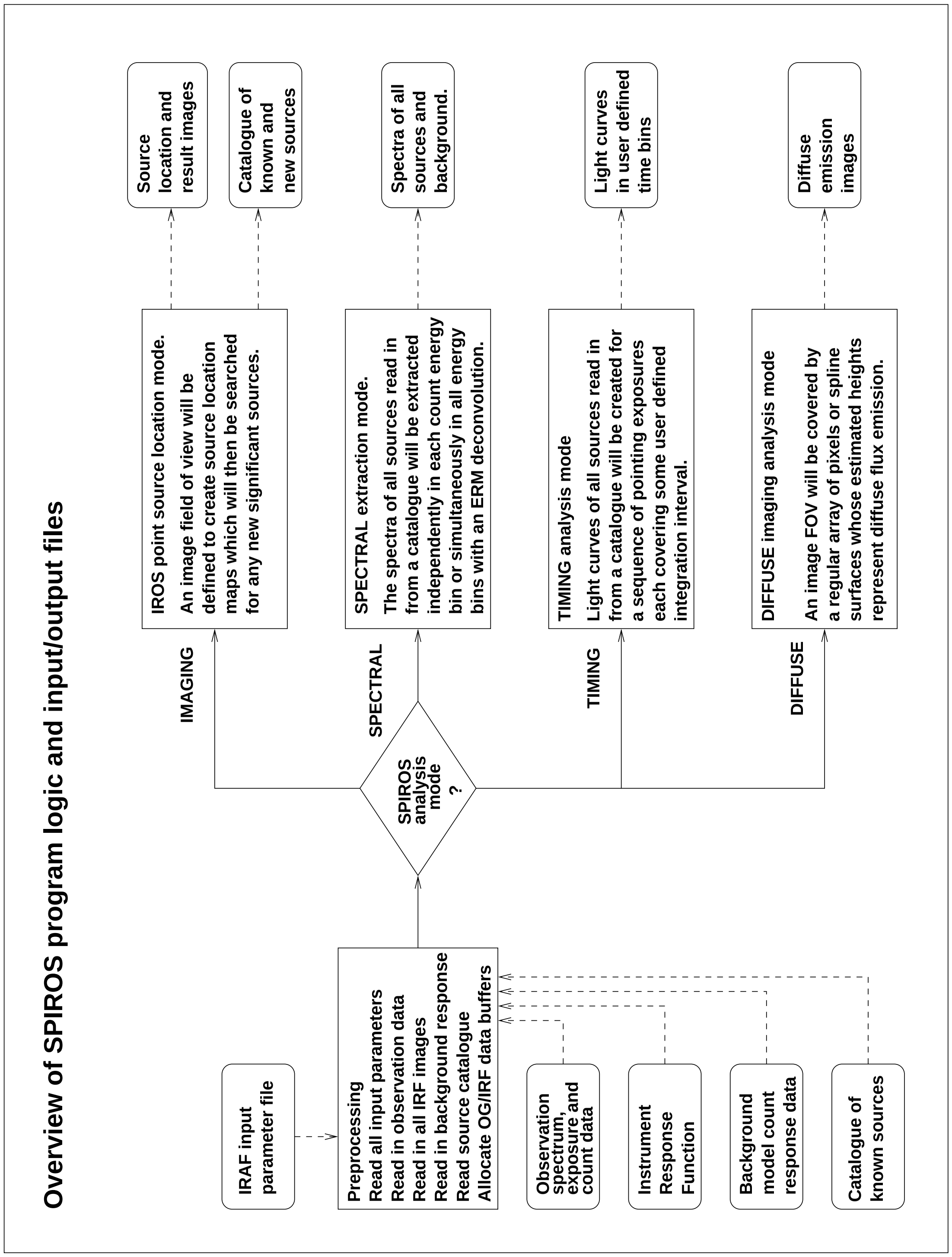
- **IMAGING** or **IROS** mode.
  Here the aim is only to locate all point or pointlike sources in the observation field of view and to output images and a source catalogue of their locations and width parameters.
- **SPECTRAL** mode.
  A catalogue of newly located sources output from **IMAGING** mode or one of known sources compiled by the user is input to calculate the spectrum of each source in it for later processing by the **XSPEC** spectral modelling package.
- **TIMING** mode.
  As in **SPECTRAL** mode a catalogue of known sources is input but to create graphs of the flux variability of each source in as many spectrum energy bins as desired.
- **DIFFUSE** mode.
  Images of diffuse emission only will be created with no identification of source locations or shape. Included for completeness and comparison with **SKYMAX** images.

It can be seen that the aim of SPIROS is to execute **three** basic tasks of interest to observers in first creating images of the observation field of view and locating any sources in it and then extracting the spectra and timing behaviour of each source.

The kind of sources of interest to SPIROS are point or pointlike objects which may also have some small width or spatial extent. SPIROS also has the ability to create images of diffuse emission but this is not its main purpose and is included for completeness and as a backup to the **SKYMAX** imaging package which uses Maximum Entropy imaging methods to create maps of diffuse emission in narrow energy bands.

The main task of the user is to select the observation data required to cover the period and region of interest, then select the analysis mode and its associated parameters in an **input parameter file** before starting **SPIROS** but this is described in Chapter 3 of this documentation.

The user is urged to take a look in the **Cookbook** section which contains several examples of the practical use of SPIROS in creating images, locating sources, extracting their spectra and light curves. There are good examples of imaging and spectral output which the user can examine more closely with display software like **FV** or **DS9**. Observation and IRF data are also supplied with execution scripts which the user can run to get some idea of how well and fast SPIROS will execute on their machine. The scripts do work so the user can use the input parameter files included and alter them for their own particular tasks.



2.2 SPIROS in IMAGING location mode.

Given its main task of locating point sources and creating images of them in the observation field of view the first question is how SPIROS goes about locating them.

One obvious method for this is to create an image of whatever emission is in the field of view and look for point sources in it. This is usually done by

- Creating a rectangular array of image pixels, the height of each representing the flux more or less emanating from its area.
- Solving for these image pixel heights simultaneously.
- Scanning the resulting image and collecting the locations of all local maxima or point source "spikes" which have a sigma significance above some threshold level.
- Take this set of initial locations, assume them to be point sources and start an iterative procedure to let them "float" to more precise locations within some limit of precision.

This location procedure can then be repeated for as many spectrum energy bins desired to calculate a final "sigma weighted" mean location for each which may then be used later in extracting their spectra.

In reconstructing such a location image and solving for the flux values and locations of each source found, their optimal values are determined by the criteria that they are such as when "folded back" through the instrument response the count values which result are in some sense a "best fit" to those derived from the event data returned from actual observations. The result required is the most "probable" such that when simulated it gives the most "probable" counts.

This is important to understand as SPIROS works at two levels

- The **DATA** level of photon counts in each detector of SPI, binned in a number of energy bins covering some spectral range and for each observation exposure.
- The **IMAGING** level of the user to create images of source emission for scientific analysis.

and the validity of the **IMAGING** results is decided by how they produce the best fit at the **DATA** level. This then determines the kind of mathematical methods and algorithms to be used.

2.2.1 The IROS mode of imaging.

The problem with the imaging method of location described above is that the image pixel array can be quite large, taking a long time to construct and solve the imaging equations required to calculate the heights of each pixel. Added to this is the fact that if there is a significant amount of noise in the input count data it will be amplified back into the image during its reconstruction. This can be suppressed somewhat by forcing a simple positivity constraint on the image pixel heights but generally it requires the further complication of "constraining" the image to flatten out any noise which is usually balanced out by blurring the image.

A faster and simpler method results by assuming some **prior knowledge** of the sources expected in the field of view, namely that they are **point** or **pointlike** and that there are only a **small finite** number of them, tens but not hundreds or thousands.

On this assumption the sources are best located using the **Iterative Removal of Sources** or **IROS** method which literally attempts to locate each source in the field of view one after the other from the strongest to the weakest.

This method does not reconstruct image pixel arrays to be searched but works as follows:

- A fast but blurred **location image** is created which shows clearly the most probable location or region of the greatest source emission. This image may be called a "pixel independent correlation map" as it is constructed by scanning the imaging field of view with a "source probe" and calculating the flux it would have assuming all input counts are due to that source and any background events alone. This typically produces a "convolved" image with a global maximum at a good first guess for a probable source location.
- With this **approximate** location find the strength, probably too large, of the source and then the detector counts expected from it.
- Subtract these source counts from the input count data to create **residue** count data more or less due to other sources not yet located.
- With these residue counts the procedure above is repeated creating a new location image, searching it for any new source and adding it to a growing list. Each time a new source is added to the list all sources are allowed to "float" to more precise locations before their counts are subtracted from the input counts.

IROS then repeats the iterative procedure above, locating each new source approximately, always allowing them to float to more optimal locations, creating new residue count data and stopping when nothing significant can be found. As noted earlier it can then repeat this procedure in any number of energy bins to calculate a mean location for each source over the entire spectrum range.

2.2.2 An example of the IROS mode of imaging.

An example showing the **IROS** method of source location can be seen in the sequence of following images. The observation of five sources identified as **1E1740.7-2942, GRS1758-258, GX354-0, GX1+4, TERZAN-2** near the galactic centre was simulated in 5 energy bands covering **70-150 keV**. The observation consisted of a sequence of 1075 exposures covering a 60^o x 40^o rectangular area and which is to be used by SPI for a survey of the galactic centre region. Here each exposure was for 4465 seconds and a total observation time covering 1333 hours.

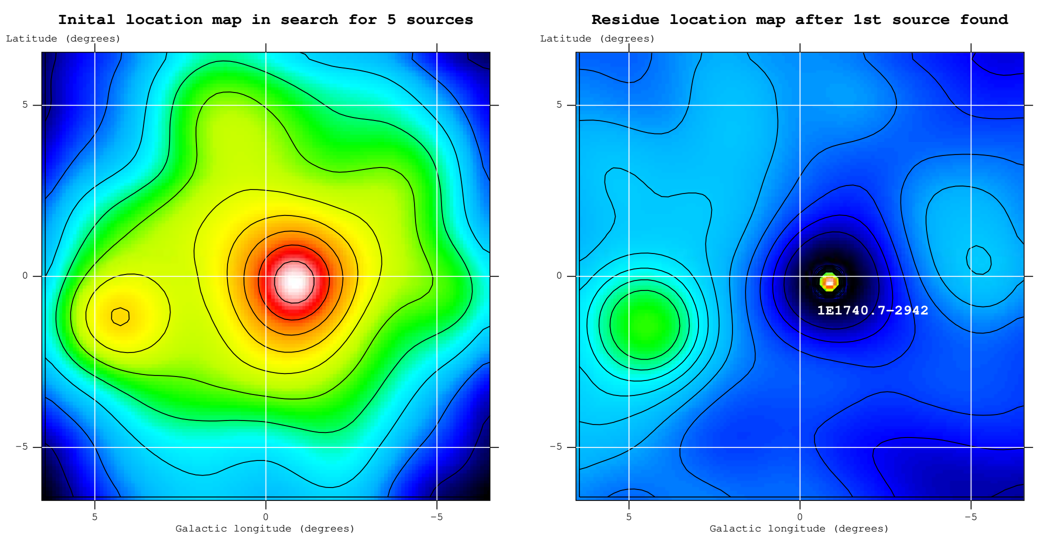


Figure 2: Initial location map to find the 1^{st} and 2^{nd} sources

The left hand image above is the initial **IROS** location map created using the input detector count data and shows the typical blurred character of such images. Knowing the location of all five sources in advance the viewer could more or less see their presence and their effect on the image.

It is clear that there is some maximal source emission existing around $(-0.9, -0.1)$ and SPIROS will locate this maximum, label it, add it to its current source list (If there was also an input catalogue of N sources then this would be source N+1.) and calculate its expected flux. SPIROS then estimates the detector counts from this source, and any others, subtracts them from the input detector count data to create a detector count residue which is then used to create the right hand location map where the first source found has also been superimposed and labelled.

SPIROS will then start a new search of the right hand map and find a significant second source candidate around $(4.5, -1, 4)$, label it, append it to its source list, and then let it "float" with the first source to find more precise locations. Every time SPIROS finds a new source all sources immediately go into a waltz around their current locations to take into account the effect of the new source.

The flux of both sources and their detector counts are again calculated and subtracted from the input detector counts to create a new residue image in the left hand image below with the two sources superimposed.

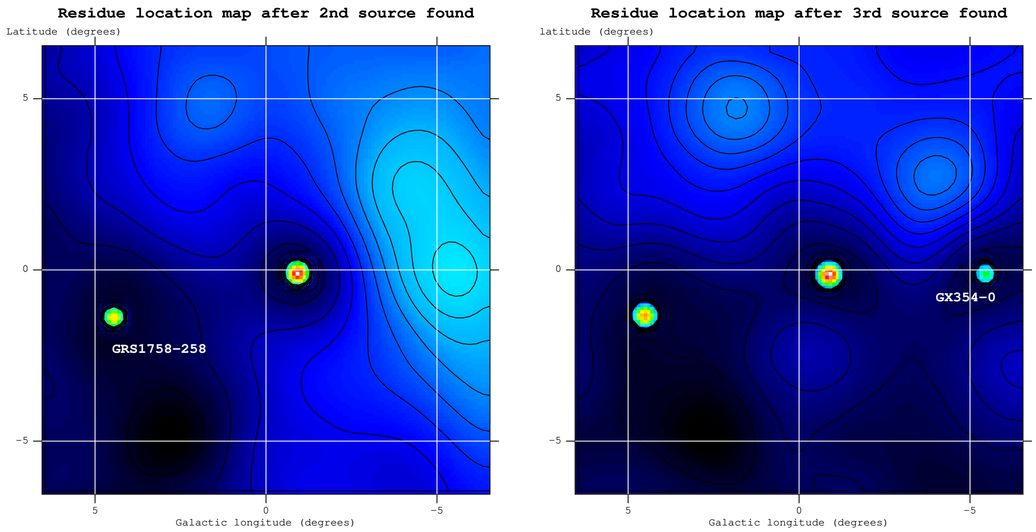


Figure 3: Residue location map to find the 3^{rd} and 4^{th} sources

In these last four images it can be seen how SPIROS keeps on repeating its strategy of search, float and residue imaging operations until it finds no more significant sources. In the example here SPIROS was asked to look for just five sources and stop, producing five location maps and a final sixth in the right hand side below which shows the final locations of all five sources superimposed on a residue map whose maximum flux values are below the 2.0σ level.

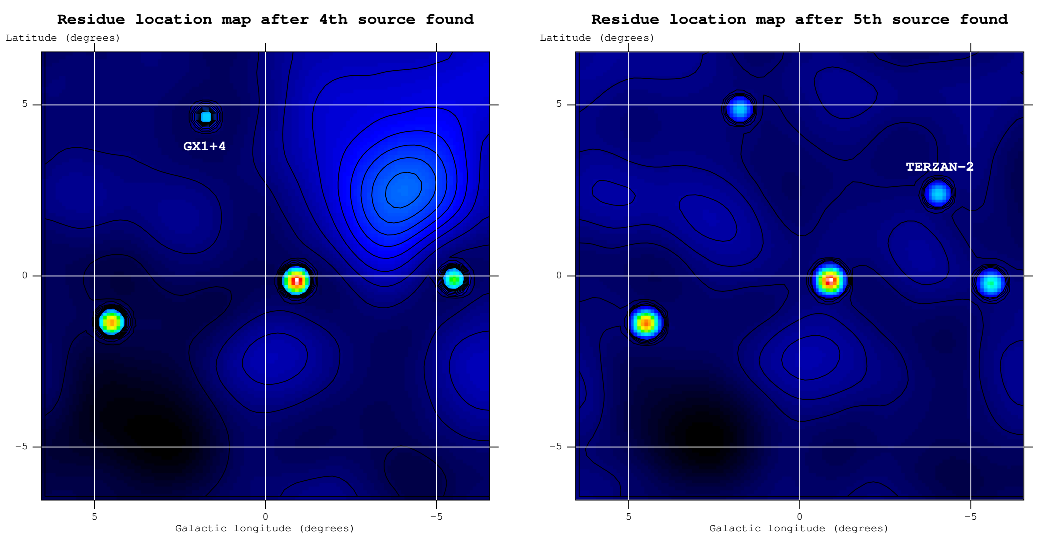


Figure 4: Residue location map to find the 5^{th} and 6^{th} sources

These sources were located in just one energy band but SPIROS can repeat the above search in multiple energy bands and then calculate some mean location for each source over the entire spectrum range. In this case SPIROS will output an additional set of images, like the last one above, for each energy bin containing sources with a common mean location.

2.2.3 Running SPIROS in IMAGING mode.

For the user running **SPIROS** in its **IMAGING** mode the main task is to get the appropriate binned count data output from **SPIHIST**, decide the size of the field of view they wish to search and how many sources they wish to look for. The user may have an input catalogue of known sources to locate them again more precisely or find an extra one or two in addition or just to make a image of them. To use SPIROS in imaging mode a description of all input parameters required can be found in the SPIROS user manual in Chapter 3 of this documentation, especially in the section called **SPIROS in a nutshell**. In the **Cookbook** of Chapter 5 there are further examples and datasets which the user can run to see the relation between input parameters and output.

IROS is a fast method which has its limitations in looking for sources which are within a degree or so of each other. The spectrometer SPI has a Full Width Half Maximum in its Point Spread Function of about 3^o so at low signal-to-noise counts it will have trouble separating weak sources. One way around this is to use an input catalogue of known or previously located sources and create an image array of narrow pixels around the region where multiple sources may be present to see if they can be resolved. But even this may not be conclusive and the user may need to look at imaging results from **IBIS** and **JEMX** to resolve them.

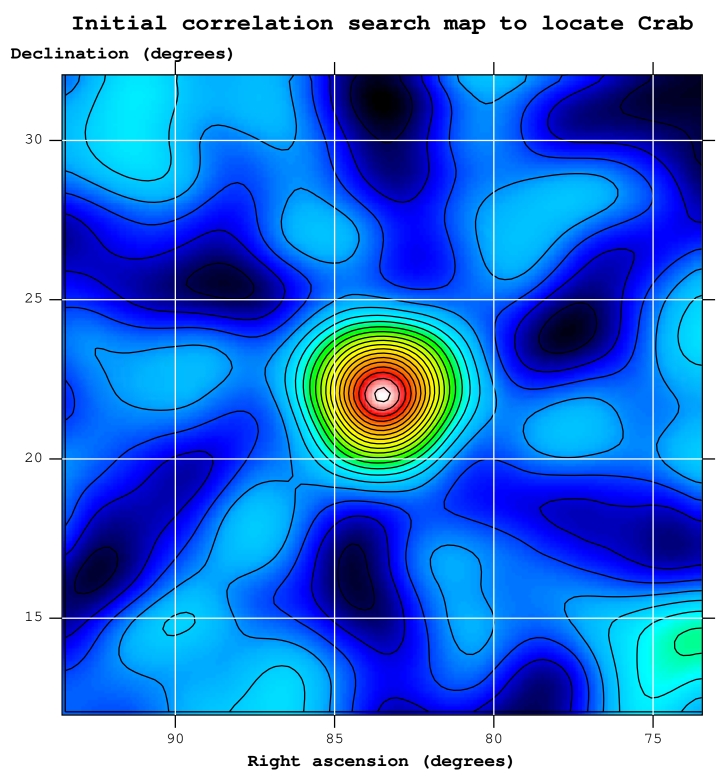


Figure 5: Point spread function map of SPI

2.3 SPIROS in SPECTRAL extraction mode.

The **SPECTRAL** mode of SPIROS is even easier to use as it does not search for sources but gets their locations and width from an input catalogue and then finds the flux values of all of them simultaneously but independently in each energy bin of the input observation count spectrum.

This method of calculating source flux values independently in each energy bin is required as a base line mode of SPIROS to output "raw" or photopeak spectra which are then to be used by the **XSPEC** spectral modelling software to extract their true spectra taking account of the energy convolution effect of off-diagonal terms in the **Energy Response Matrix (ERM)** of the **Instrument Response Function (IRF)**.

The user can also make this **energy deconvolution** in SPIROS by setting an input parameter to calculate all source spectral flux values simultaneously in all energy bins taking account of the off-diagonal terms in the **ERM** of the **IRF**.

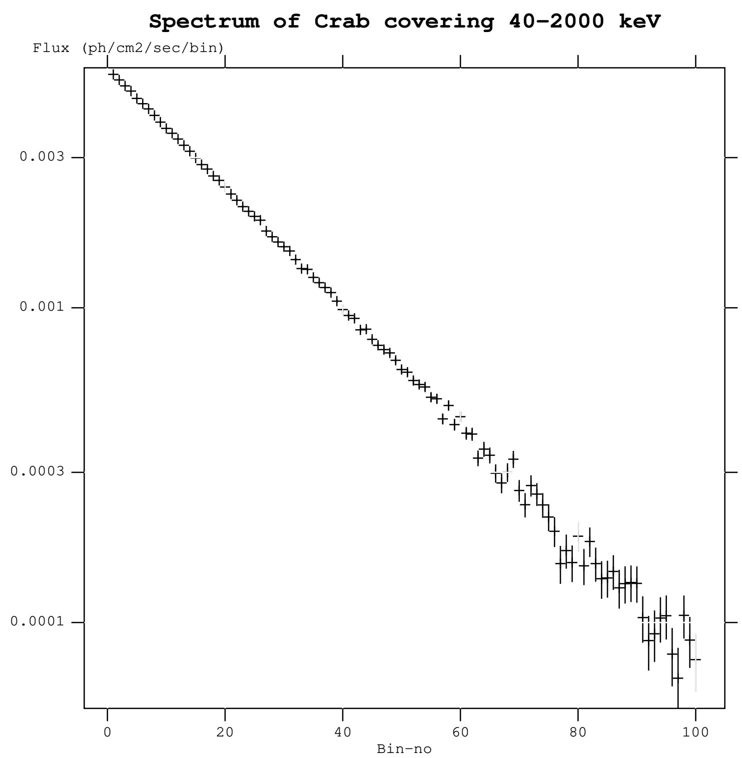


Figure 6: A typical power law spectra extracted by SPIROS

Given observation data output from **SPIHIST** with the spectrum energy binning required and a detector background rate model from **SPIBACK** the user need then only specify the names of output spectra datasets as input parameters to run SPIROS in this mode. A description of the input parameters required for this mode may be found in Chapter 3 of this documentation.

2.4 SPIROS in TIMING analysis mode.

The **TIMING** mode of SPIROS is again easy to use as it does not search for sources but gets their locations from an input catalogue as in **SPECTRAL** mode. The idea is that the user will bin count data over some observation period in as many energy bins as desired and that SPIROS will calculate source flux values at a sequence of MJD times over some **integration time interval** to be given by the user.

At present **TIMING** mode has two forms of analysis

- **QUICKLOOK**
  This is a fast "snapshot" method to calculate flux variability whereby the sequence of observation exposures is broken down into groups each covering the integration time interval required. The flux of all sources along with detector background rates are calculated for each integration interval and stored as a graph, and independently for each energy bin. This is fast but does not force the background rate values calculated to conform to some previously given model function, allowing small fluctuations which translate into compensating increases or decreases in source flux values.

- **WINDOW**
  This is a slower method where detector background rate values have some previously given time model function by the workpackage **SPIBACK** and only variations in source flux values are calculated for the sequence of integration time intervals. It also allows for the output of spectra for each time bin plus a mean spectra covering all time bins.

- **TRANSIENT**
  This is similar to WINDOW mode but allows for light curves to be described by continuous B-splines as well as the usual HAT shaped time bins. It also allows for different time bin scales for individual sources and a range of particular models functions to describe assymmetric flares and cyclical outbursts.

A description of the input parameters required for this mode may be found in Chapter 3 of this documentation.

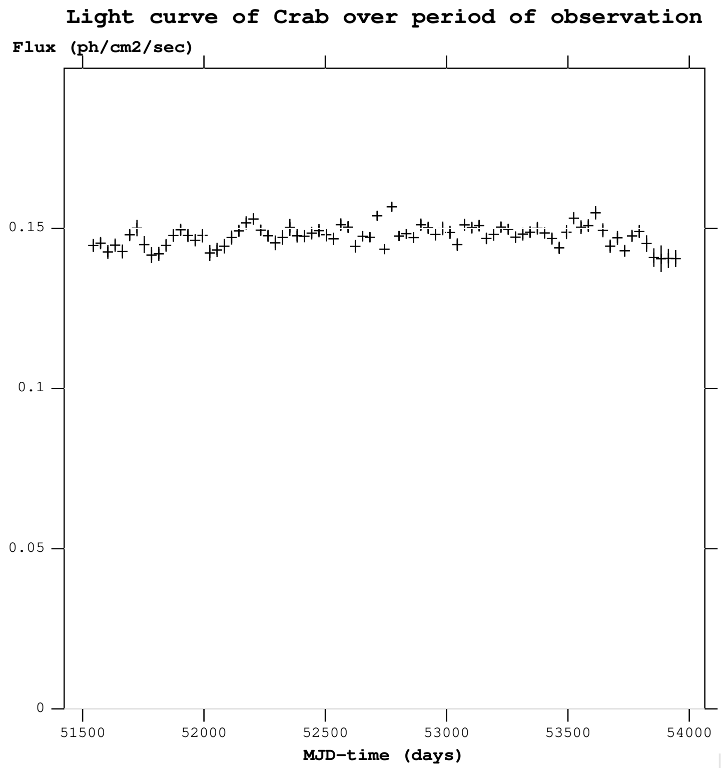


Figure 7: A typical **QUICKLOOK** light curve extracted by SPIROS

2.5 SPIROS error codes.

This section contains a list of errors codes returned by SPIROS, their cause and possible remedy. The following list contains errors codes and their FORTRAN identifiers, in two groups for general errors and those in using specific NAG subroutines.

-210001	SPIROS_ANALYSIS_MODE_INVALID
-210002	SINE_COSINE_BACKGROUND_INVALID
-210003	NO_BACKGROUND_OR_SKY_SOURCES
-210004	NO_SKY_SOURCES_IN_CATALOGUE
-210005	REBINNING_BIN_BAND_INVALID
-210006	REBINNING_ENERGY_BAND_INVALID
-210007	DETECTOR_POINTING_MISMATCH
-210008	ML_PARAMETER_HAS_NO_MINIMUM
-210009	ML_PARAMETER_KEEPS_INCREASING
-210010	ZERO_SOLUTION_RETURNED
-210011	INDETERMINATE_ZERO_SOLUTION
-210012	INDETERMINATE_IMAGING_PROBLEM
-210013	INDETERMINATE_SPECTRA_PROBLEM
-210014	INDETERMINATE_DIFFUSE_PROBLEM
-210015	INDETERMINATE_QUICKLOOK_PROBLEM
-210016	INDETERMINATE_WINDOWING_PROBLEM
-210017	INDETERMINATE_TRANSIENT_PROBLEM
-210018	NO_POINTING_EXPOSURES_FOUND
-210019	TWO_OR_MORE_ZERO_DETECTORS
-210020	NO_POINTER_TO_OG_DATASET

-210021	NO_IRFS_IN_INDEX_FILE
-210022	NOT_ENOUGH_IRF_DETECTORS
-210023	NOT_ENOUGH_PSD_DETECTORS
-210024	NO_INPUT_PSD_EFFICIENCY_DATA
-210025	NO_INPUT_PSD_RESPONSE_DATA
-210026	TABLE_SHORTER_THAN_EXPECTED
-210027	TABLE_LONGER_THAN_EXPECTED
-210028	SOURCE_HAS_NO_NAME
-210029	SOURCE_HAS_IDENTICAL_NAME
-210030	BACKGROUND_COMPONENT_UNKNOWN
-210031	NO_RESPONSE_IN_ENERGY_BIN
-210101	SOLVE_SIMULTANEOUS_EQS_F07MDF
-210102	SOLVE_SIMULTANEOUS_EQS_F07MEF
-210103	SOLVE_SIMULTANEOUS_EQS_F04ATF
-210104	MAXLI_LINEAR_ERRORS_F07MDF
-210105	MAXLI_LINEAR_ERRORS_F07MJF
-210106	MAXLI_IMAGEQ_QPMINIMUM_E04NCF

2.6 SPIROS error code explanation and handling.

-210001 - SPIROS_ANALYSIS_MODE_INVALID

The analysis mode selected in input parameter "mode" is not recognized.
Check its value is **IMAGING, IROS, SPECTRA, TIMING or DIFFUSE**

-210002 - SINE_COSINE_BACKGROUND_INVALID

The SINE or COSINE background components selected in **SPIBACK** are invalid as they have negative counts and cannot be used in **Maximum Likelihood** mode.
Rerun SPIBACK again but select positive SINE+1 or COSINE+1 components.

-210003 - NO_BACKGROUND_OR_SKY_SOURCES

SPIROS found no background model components or any sources selected for analysis in the input catalogue.
Check that the input parameter **background-method** is not zero and edit the column **SEL_FLAG** in the input catalogue to be **1** for sources to be analysed or **2** if they are also to be flagged as **variable** for **TIMING** analysis.

-210004 - NO_SKY_SOURCES_IN_CATALOGUE

There were no sources selected for analysis in the input catalogue.
Check that the column **SEL_FLAG** has been set to **1** for sources to be analysed or **2** if they are also to be flagged as **variable** for **TIMING** analysis.

-210005 - REBINNING_BIN_BAND_INVALID

Counts were to be rebinned in a single energy bin according to the "from-until" parameter **srclocbins** by choosing a range of **bin numbers** which are found to not cover the actual number of bins in the OG count data.
Check the number of bins and make sure the range selected is a subset of them.

-210006 - REBINNING_ENERGY_BAND_INVALID

Counts were to be rebinned in a single energy bin according to the "from-until" parameter **srclocbins** by choosing a range of **energies**, not bins, which are found to not cover the actual energy range in the OG count data.
Check the energy range and make sure the range selected is smaller than it.

-210007 - DETECTOR_POINTING_MISMATCH

In reading in background or count data the number of spectra for all detectors and pointings is not the same as "no-of-detectors * no-of-pointings".
Check the number of detectors and pointings and compare their product with the no of spectra in the background or count data files.

-210008 - ML_PARAMETER_HAS_NO_MINIMUM

SPIROS is using a **Maximum Likelihood** method to find an optimum solution for source flux values in a series of iterations but finds that the ML parameter is decreasing without reaching a minimum after 2000 iterations.
Check that the ML parameter stopping precision specified in input parameter **maxlikprec** is not too small. Try increasing it and start again. If it persists when a large value, greater than 1, is given, get a software developer for SPIROS to put a trace in the subroutine "spibham_maxli_reconstruction" or "spibham_maxli_fixed_solution" to find out what is wrong.

-210009 - ML_PARAMETER_KEEPS_INCREASING

SPIROS is using a **Maximum Likelihood** method to find an optimum solution for source flux values in a series of iterations but for some reason finds that the ML parameter is increasing instead of decreasing. It is fatal.
Check for "abnormalities" in the input count data and that the IRF selected is valid, otherwise get a software developer for SPIROS to put a trace in the subroutine "spibham_maxli_reconstruction" or "spibham_maxli_fixed_solution" to find out what is wrong.

-210010 - ZERO_SOLUTION_RETURNED

SPIROS has returned a "zero solution" when calculating source flux values in some energy bin. It is not fatal and is due to zero counts in the energy bin.

-210011 - INDETERMINATE_ZERO_SOLUTION

SPIROS has has found only one detector in some energy bin to have counts in it and cannot hope to find a sensible solution for flux values and will return a zero solution. It is not fatal.

-210012 - INDETERMINATE_IMAGING_PROBLEM

In estimating the expected number of flux values to be calculated when locating sources in **IMAGING** mode it is found that there are more of them than the number of count data bins available. *i.e.*the product of the number of detectors, pointings and energy bins.
Lower the number of sources to be located in parameter **nofsources** or reduce the number of detector background values to be calculated, either by selecting fewer components in SPIBACK or by changing the input parameter **background-method** to **3** or **4**. Check its meaning in this documentation.

-210013 - INDETERMINATE_SPECTRA_PROBLEM

In calculating the expected number of source spectrum flux values to be calculated in **SPECTRAL** mode it is found there are more of them than the number of independent parameters available.
Lower the number of sources or energy bins to be calculated.
See also error -210012 above.

-210014 - INDETERMINATE_DIFFUSE_PROBLEM

In calculating the expected number of catalogue sources plus pixel values to be calculated in diffuse imaging it finds there are more than the number of independent parameters available.
Lower the number of catalogue sources or image pixels to be calculated.
See also error -210012 above.

-210015 - INDETERMINATE_QUICKLOOK_PROBLEM

In **TIMING** analysis QUICKLOOK mode background values and source flux values are calculated for one or more pointings grouped together. In estimating the expected number of background and source flux values to be calculated it is found there are more of them than count data bins available.

- Increase the integration time in input parameter **source-timing-scale** as this will group more pointings together.
- Reduce the number of detector background values to be calculated, either by selecting fewer components in SPIBACK or by changing parameter **background-method** to **3** or **4**.

-210016 - INDETERMINATE_WINDOWING_PROBLEM

In **TIMING** analysis WINDOWING mode detector background values are calculated over the entire observation period while flux values are calculated in a single "moving" time bin covering successive sequences of pointings lying within the integration time specified in parameter **source-timing-mode**. In estimating the expected number of background and source flux values to be calculated it is found there are more of them than count data bins available.

- Increase the integration time in input parameter **source-timing-scale** as this will group more pointings together.
- Reduce the number of detector background values to be calculated, either by selecting fewer components in SPIBACK or by changing parameter **background-method** to **3** or **4**.

-210017 - INDETERMINATE_TRANSIENT_PROBLEM

In **TIMING** analysis TRANSIENT mode detector background values are calculated over the entire observation period simultaneously with flux values in all time bins. In estimating the expected number of background and time bin values to be calculated it is found there are more of them than the number of count data bins available.

- Increase the integration time in input parameter **source-timing-scale** as this will reduce the no of time bins.
- There might be too few pointings so change the mode in parameter **source-timing-mode** to **WINDOWING** which will then calculate flux values in each time bin separately.

-210018 - NO_POINTING_EXPOSURES_FOUND

In reading in observation data the no of pointing exposures is found to be zero.
Check the SPI.-OBS.-PTG dataset.

-210019 - TWO_OR_MORE_ZERO_DETECTORS

In reading in IRF parameters the detector identity sequence specified should have only one zero detector but more are found.
Check the detector identity sequence obtained from the SPI.-OBS.-GTI dataset.

-210020 - NO_POINTER_TO_OG_DATASET

In reading in IRF parameters there is no address pointer to the Observation Group dataset prepared during parameter input. This should not occur in SPIROS but may occur in other software using the same IRF access subroutine as SPIROS.

-210021 - NO_IRFS_IN_INDEX_FILE

In reading in IRF parameters no IRFs are found in the index file.
Check the IRF index file.

-210022 - NOT_ENOUGH_IRF_DETECTORS

There are more detectors specified in the detector identity sequence obtained from the SPI.-OBS.-GTI dataset than those specified for the IRF.
Find an IRF with enough detectors to cover those in the observation data.

-210023 - NOT_ENOUGH_PSD_DETECTORS

In the detector identity sequence obtained from the SPI.-OBS.-GTI dataset there are PSD detectors [85-103] [104-122] or [123-141] but the detector range found for the IRF is less than the range [0-19] required.
Find an IRF with enough detectors to cover those in the observation data.

-210024 - NO_INPUT_PSD_EFFICIENCY_DATA

In the detector identity sequence obtained from the SPI.-OBS.-GTI dataset there are PSD detectors [85-103] [104-122] or [123-141] but there is no PSD efficiency dataset available with the observation data.
Check the Observation Group datasets for a PSD efficiency dataset.

-210025 - NO_INPUT_PSD_RESPONSE_DATA

In the detector identity sequence obtained from the SPI.-OBS.-GTI dataset there are PSD detectors [85-103] [104-122] or [123-141] but there is no PSD response dataset available with the observation data.
Check the Observation Group datasets for a PSD response dataset.

-210026 - TABLE_SHORTER_THAN_EXPECTED

In reading in GTI, DEADTIME or COUNT data the total no of rows in the dataset is less than the product of the number of detectors and no of pointings from the SPI.-OBS.-PTG dataset.
Check all observation datasets with pointing parameters for a mismatch.

-210027 - TABLE_LONGER_THAN_EXPECTED

In reading in GTI, DEADTIME or COUNT data the total no of rows in the dataset is greater than the product of the number of detectors and no of pointings from the SPI.-OBS.-PTG dataset.
Check all observation datasets with pointing parameters for a mismatch.

-210028 - SOURCE_HAS_NO_NAME

While reading a source catalogue a source selected for analysis has no name.
Check source catalogue names.

-210029 - SOURCE_HAS_IDENTICAL_NAME

While reading a source catalogue two sources selected for analysis have the same name.
Check source catalogue names.

-210101 - SOLVE_SIMULTANEOUS_EQS_F07MDF

-210102 - SOLVE_SIMULTANEOUS_EQS_F07MEF

-210103 - SOLVE_SIMULTANEOUS_EQS_F04ATF

-210104 - MAXLI_LINEAR_ERRORS_F07MDF

-210105 - MAXLI_LINEAR_ERRORS_F07MJF

-210106 - MAXLI_IMAGEQ_QPMINIMUM_E04NCF

3 SPIROS input parameters

Using **SPIROS** is simple. The user must only prepare an text file called **spiros.par** containing input parameters, in so called **IRAF** format, to describe input/output datasets and control what **SPIROS** is required to do and then start it with the command **spiros** preceeded by its directory location pathway.

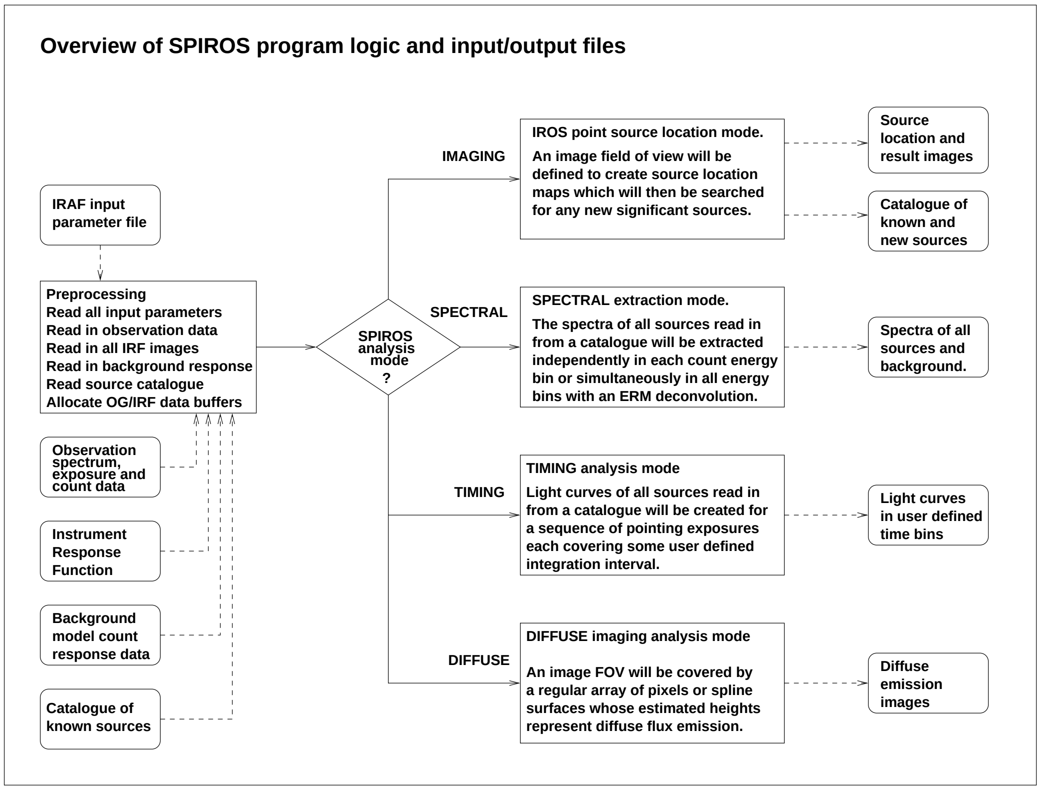


Figure 8: Flowchart overview of SPIROS functionality and IO datasets.

3.1 SPIROS input parameter format.

The **IRAF** format of input parameters is a text file where each line contains a parameter description with the following simple format:

Parameter,Type,Query,Value,From,Until,Prompt

where

- **Parameter** is the parameter description name such as **input-file**.
- **Type** describes whether the parameter is a text string, real or integer number.
  s expects a text string.
  r expects a real number.
  i expects an integer number.
- **Query** is a flag **q** or **h** to prompt the user for input of the parameter value.
  q will display the prompt text in **Prompt** and wait for manual input.
  h means the parameter is hidden from external input with a value given by **Value**.
- **Value** is the actual value of the parameter depending on its type **Type**.
  s expects a text string in quotes such as "filename".
  r expects a real decimal such as 12.345.
  i expects an integer number such as 123.
- **From** is a lower limit on the parameter value in text, real or integer format.
- **Until** is an upper limit on the parameter value in text, real or integer format.
- **Prompt** is a display prompt text used to ask for manual input if **Query** is **q** and should be a text string in quotes such as "Enter name of input observation file: ".

A typical example of an IRAF parameter is the **mode** parameter which describes the IROS, DIFFUSE, SPECTRAL or TIMING analysis mode which the user wishes to select.

mode,s,h,"IROS",,,"Enter analysis mode [IROS,DIFFUSE,SPECTRAL,TIMING,IMAGING]:"

In the following pages each of the IRAF input parameters for **SPIROS** will be described in broad functional groups with their limits, default values and a description of their purpose or effect on the subsequent execution of **SPIROS**.

3.2 SPIROS input parameters.

3.2.1 SPIROS operating mode.

mode,s,h,"IROS",,,"Enter mode for imaging, spectral or timing analysis: "

The **mode** parameter tells SPIROS what sort of analysis it is to make with any input observation data:

- In **IROS** imaging mode (or **IMAGING** mode) SPIROS will search for as many point or pointlike sources the user requests using the method of **Iterative Removal of Sources** to determine their location and any width parameters. It will repeat the search in each spectrum energy bin of the input observation count data and form an average location for each source. Output will be the location and width parameters of each source to a catalogue along with field of view images of them and any residue emission to an image catalogue.
- In **SPECTRAL** extraction mode SPIROS takes an input catalogue of known sources and calculates the flux of all simultaneously in each of the spectrum energy bins of the input observation count data. It will output flux spectra for each source, detector background count spectra and source count spectra for later processing by XSPEC and other spectral modelling software.
- In **TIMING** analysis mode SPIROS again takes an input catalogue of known sources and calculates the flux variations of all simultaneously but for small groups of exposures which all fall within some integration time interval specified by the user. Output is a light curve over the observation period for each source and detector background in each of the spectrum energy bins of the input observation count data.
- In **DIFFUSE** imaging mode SPIROS will create a regular array of image pixels to reconstruct an image of diffuse emssion. It will not at present try to locate any point or compact extended sources in it.

It is expected that in general SPIROS will be used first in IROS mode to locate sources and output source catalogues and images and to get some general idea of what it sees in the SPI field of view. The output catalogues may then be used as input catalogues in SPECTRAL or TIMING mode to output the spectra or light curves of new and known sources. Remember that IROS mode and IMAGING mode are the same.
By appending a **-N** "verbosity level" to this name the logfile output can be increased to display more detailed information not normally of interest to the average user but useful for problem diagnostic purposes.

3.2.2 Observation datasets for input.

From the previous chapter the user should have some idea of the basic way the spectrometer SPI operates and that before using SPIROS a pipeline of preprocessing steps will have converted raw photon event data from the spacecraft into detector count data showing the shadow pattern of the SPI mask cast on the detector array by all sources in the observation field of view for each pointing exposure and in as many spectrum energy bins as the user requires.

This data needs to be accessible to SPIROS via the group of parameters described here.

This observation data comes as a group of FITS files which describe the energy binning, pointing exposure sequence, effective exposure times and detector counts which are grouped together as a list in another file called the Observation Group dataset and it is this which the user must only specify to give SPIROS access to all the observation data and parameters that it needs.

in-og-dol,s,h,"Crab-count-data.fits[1]",,,"Enter name of observation group input DOL: "

This is the Observation Group input dataset which provides access to all the other observation datasets mentioned above and is usually the only parameter in this group that the user must provide. There are no defaults and SPIROS will abort if it is not given and cannot be found. Note also that this is an **input file** or a **Data Object Locator** or **DOL** and requires the [**n**] appendage after the FITS file name to locate the file data correctly. Its value is usually [**1**] and if not added may cause errors or abort SPIROS.

ebounds-dol,s,h,"",,,"Enter energy binning boundaries DOL: "

pointing-dol,s,h,"",,,"Enter observation pointing directions DOL: "

gti-dol,s,h,"",,,"Enter observation good time intervals DOL: "

deadtime-dol,s,h,"",,,"Enter observation dead time intervals DOL: "

evts-det-spec-dol,s,h,"",,,"Enter detector event spectra DOL: "

This group of parameters allows the user to substitute each of the observation input datasets listed in the Observation Group datset defined by parameter **in-og-dol** above with an alternative. This is expected to be seldom used.

back-model-idx,s,h,"Crab-backgr-model-index.fits[1]",,,
"Enter name of detector background model detector spectra index DOL: "

SPIROS must also take account of an additional detector response to background photons at all energies and which will vary with detector location and with time on an orbital scale at least. This parameter is the name of a file which gives access to the response in each detector for each pointing exposure and energy bin according to some background component model which has been previously analysed by the pipeline program **SPIBACK** to produce the file selected here. Again note the [**1**] DOL number which is mandatory for all input files.

3.2.3 Instrument Response Function.

inst-resp-idx,s,h,"/home/phc/INTEGRAL/ISDC/IRFs/SPI-IRF-index.fits[1]",,,
"Enter name of instrument response function: "

This is the name of an **index** file which allows access to the count response in all detectors of SPI due to a point source anywhere in the instrument field of view and at any source photon energy.

3.2.4 Catalogue of known sources for all modes.

source-cat-dol,s,h,"all_stars.fits[1]",,,"Enter name of catalogue of known sources: "

For the SPECTRAL and TIMING analysis modes of SPIROS a catalogue of known sources must be specified for input. In IROS or DIFFUSE imaging mode it is not essential but saves SPIROS from searching the FOV for sources whose locations are more or less known. Again note the [**1**] DOL number which is mandatory for all input files.

3.2.5 Observation Group datasets for output.

out-og-dol,s,h,"Crab-output-og.fits(GNRL-OBSG-GRP.tpl)",,,
"Enter name of Observation Group for output DOL: "

This is the name of a file, similar to the Observation Group input dataset defined by the parameter **in-og-dol** with the same list of datasets extended to included any other datasets for input or output which have been selected above.

This is an **output** file and generally requires a template name **(......tpl)** in brackets to be appended to the FITS file name. For all output files SPIROS calls a subroutine **COMMON_PREPARE_PAR** to create them and requires a template which describes their data format, keywords and array/table format. If the template name is not appended an error will be returned and SPIROS will abort leaving a list of possible templates, which may or may not exist, in its logfile.

This particular **out-og-dol** file is the **only** exception to the template rule and may have just a [**n**] or [**GROUPING**] DOL identifier appended instead.

3.2.6 Input parameters common to all analysis modes.

When SPIROS calculates source flux values, locations or widths it does so by finding their values which optimize a χ^2 or **Maximum Likelihood** objective function to give the most "probable" solution. In solving for such parameters iterative methods are mostly used to solve for better and better values and various **stopping criteria** need to be given to stop the process when they fall within some reasonable precision.

optistat,s,h,"CHI2",,,"Enter ML optimization statistic [CHI2,LIKEH]: "

For the count values expected from SPI exposures the noise or statistics in them can generally be expected to be **Gaussian** requiring the minimization of a χ^2 statistic. In some cases, especially of weak diffuse emission, the count noise can be expected to be **Poisson** requiring the minimization of a Maximum Likelihood parameter such as the **Cash** statistic.

maxlikprec,r,h,0.2,,,"Enter ML optimization precision: "

This is the stopping criteria in optimizing a χ^2 or **ML** statistic and is used in conjunction with the source location precision **srclocprec** or width precision **srcwidprec** to stop improving source location or width values.

solution-constr,s,h,"NONE",,,"Enter image solution constraints [NONE,POSITIVE]: "

Here the user can tell SPIROS to solve its imaging equations linearly or with a positivity constraint. The general imaging problem that SPIROS solves does in fact have a positivity constraint but linear solutions are also useful. The positivity constraint is useful for reconstructing **DIFFUSE** emission maps having large pixel arrays in giving a simple means of suppressing the spread of any large noise oscillations in the image surface.
By appending **-NOERR** or **-LINERR** the user can force nonrobust or robust linear errors to be calculated instead of the more time consuming nonlinear errors calclulated for a positive solution.

nagoptions,s,h," ",,,"Enter NAG options file for minimization subroutines: "

NAG subroutines like **E04NCF** used to solve a classic QP optimization may have its search parameters and stopping criteria altered by entries made in a text file as described in the NAG documentation and this parameter is the name if that text file if required.

background-method,i,h,2,,,"Enter handling of background model response values

SPIROS requires a model of the way detector background varies with time during its ≈ 72 *hour* orbital period and this is prepared by the executable **SPIBACK** with output to a file indexed by the the input parameter **back-model-idx** above. The output in this file is the **background response function** covering the period of observation and is for a number of possible background time variation components, **CONSTANT, LINEAR, QUADRATIC, CUBIC, SIN+1, COS+1, ACS, DFEE, IREM** and is stored as expected counts in a similar way to observation count data for input.

Normally SPIROS calculates detector background rates in each detector simultaneously with all sky source flux values but it might be that the user knows the pattern of background counts across the detector as "relative function values" and wishes only to find a **single** "multiplier" for this count pattern to find a good fit. This parameter allows the user this choice and also to indicate that the background response provided by SPIBACK is indeed the correct background and may be simply subtracted from the input count data to solve for source flux values only.

The most usual value is **2** to calculate background rates in all detectors independently but other allowed values are:

- **0** - This implies there is no background, only good for calibration sources, and SPIROS will calculate sky source flux values only.
- **1** - This implies the background values are given and will be subtracted from input counts first. SPIROS will then calculate sky source flux values only.
- **2** - This implies background rates in **all** detectors are to be calculated simultaneously with sky source flux values.
- **3** - This implies background rates given in each detector are **relative** function values. *i.e.*The relative background rate over all detectors [0-18], [0-60] or [0-84] are known and have been specified in SPIBACK.
- **4** - This is like **3** but more complicated in that if detectors for double events [19-60] or triple events [61-84] are used then SPIROS will assume the **relative** background rates in the detector groups [0-18], 19-60] and [61-84] are known independently of each other and have been specfied in **SPIBACK** and just 1, 2 or 3 "multipliers" will be calculated for the detector groups present and not 19, 61 or 85 as would be the case if **2** would be specified.
- **5** - This is known as the **MCM** or Mean Count Modulation method and was developed for the case where there is no background model, or that from SPIBACK is found to be unsatisfactory, and a solution with some sort of general background variation is required. The method assumes that detector background count rates have a "modulation pattern" across the detector plane which is fixed in time but whose magnitude may vary with time. In this case the background can be "mathematically" removed from the problem by converting the count data to a deviation from the modulation pattern. It takes longer to calculate as it returns a fractional modulation value for each detector plus a magnitude value for each pointing. This is problematic in that it allows for background variations on a time scale of each exposure, or $\frac{1}{50}$ day, when INTEGRAL can expect variations on the scale of a day or longer. This could lead to source variations being interpreted as background variations so light curves of the background should be examined carefully to see if this happens.

- **6** - This is only valid for **TIMING** analysis in **TRANSIENT** mode and will spline the background on a scale given by the new parameter **backgr-timing-scale** or, if this is absent, by parameter **source-timing-scale**. This solves the problem mentioned above for background handling 5 but it will take even longer as it assumes a similar "background modulation pattern" which is the same for each of spline components used. It is however the best general background model available to SPIROS.

energy-response,s,h,"N",,,"Enter if Energy Response Matrix is to be used [Y/N]: "

SPIROS normally does its imaging and spectral analysis for photons at the energy of the source and ignores the fact that a source photon will not always deposit all its energy in the detector it encounters but only some fraction at a lower energy. Thus the response to incoming source photons at a particular energy is mostly at the photopeak or source energy with some fraction spread out more or less continuously over the energy range below the photopeak energy. A matrix relating the fraction of photons at some **source** energy which are dispersed over the **detected** spectrum below the source energy is called the **Energy Response Matrix** or ERM.

This parameter is not really relevant in **IROS** or **DIFFUSE** mode and is normally set to 'N' for **SPECTRAL** mode where the default is to produce "photopeak" spectra which are then passed on for processing by **XSPEC** spectral modelling software where the off-diagonal terms of the ERM are taken into account to reconstruct the true source spectrum. However if the user wishes to set up their own binning scheme to model the continuum part of a spectrum whose lines they already know then by setting the parameter to 'Y' the true source spectra should be returned and may be compared with **XSPEC** output.

maxlikfile,s,h,"residues.fits(SPI.-MAXL-RES.tpl)",,,"Enter residue output file: "

This file will contain the residue counts for each detector, pointing and energy bin and may be used to search for extreme values which may indicate "bad" quality input count data or a wrong background model or an incorrect source catalogue.
The name is a misnomer as the output was originally intended to be χ^2 or **Maximum Likelihood** contributions instead of the more simple count residues.

3.2.7 Output catalogue and images for IROS/DIFFUSE mode.

source-res,s,h,"Crab-output-catalogue.fits(SPI.-SRCL-RES.tpl)",,,
"Enter name of output source catalogue:

This parameter will prompt SPIROS to output a source catalogue containing all known sources and any new ones located. Again note the **template** name which is mandatory.

image-idx,s,h,"Crab-images.fits(SPI.-SKY.-IMA-IDX.tpl)",,,
"Enter name of output sky image index: "

This parameter will prompt SPIROS to output images of emission flux, error and sigma values showing source locations and any emission residue and index them in them filename specified. Note that the filename requires the name of a bf template to be appended which specifies how the output index is to be constructed.

image-int,s,h,"Y",,,"Enter request for flux intensity image [Y,N or image name]: "

This parameter indicates if source flux intensity images are required for output. If only 'Y' is entered for the output dataset name will be made from the image index file name with "**_intensity**" appended to it. If a name is specified it should have a **SPI.-SKY.-IMA.tpl** template name appended but if not this will be constructed by cutting off the **-IDX** part of the index file template name in parameter **image-idx** above.

image-err,s,h,"Y",,,"Enter request for flux error image [Y,N or image name]: "

This parameter indicates if source flux error maps are required for output. If only 'Y' is entered for the output dataset name will be made from the image index file name with "**_error**" appended to it.

image-sig,s,h,"Y",,,"Enter request for flux sigma image [Y,N or image name]: "

This parameter indicates if source flux sigma maps are required for output. If only 'Y' is entered for the output dataset name will be made from the image index file name with "**_sigma**" appended to it.

iteration-output,s,h,"Y",,,"Enter if image output required at each iteration [Y,N]: "

This parameter indicates that in the search procedure which locates sources one by one from the strongest to the weakest there will be output of images at each search iteration showing the sources known or located and the emission residue remaining. It is the residue image that SPIROS will search to see if there are any new sources to be found.

3.2.8 Output image parameters for IROS/DIFFUSE mode.

reference-coord,s,h,"RADEC",,,"Enter output image reference coordinate system:"

This parameter will normally have the value of **RADEC** but if **GALACTIC** is specified the observation pointing directions and imaging frame of reference will be translated into (l, b) galactic coordinates.

image-fov,s,h,"POINTING-CENTRE",,,"Enter extent and location of output image:"

This parameter allows the user to select the imaging field of view or default it to cover the observation pointing field of view.
USER requires the user to specify all imaging array parameters below.
POINTING-CENTRE will set the centrepoint of the image array to that of the observation field of view only. The user must still specify its size.
POINTING does the same as **POINTING-CENTRE** but it will also set the size of the image field of view to a rectangle covering all ponting directions.
POINTING-FCFOV is the same as **POINTING** but the size of the image field of view will be extended by the **FULLY CODED** field of view of SPI, about $\pm 8^o$.
POINTING-ZCFOV is the same as **POINTING** but the size of the image field of view will be extended by the **ZERO CODED** field of view of SPI, about $\pm 16^o$.

center-long,r,h,0.0,,,"Enter longitude of output image centerpoint (degs): "
center-lat,r,h,0.0,,,"Enter latitude of output image centerpoint (degs): "

This parameter pair specifies the centrepoint location of the imaging field of view.

image-dim-long,i,h,101,,,"Enter output image array dimension in longitude: "
image-dim-lat,i,h,101,,,"Enter output image array dimension in latitude: "

This parameter pair specifies the integer dimensions of the image pixel array.

image-pixel-long,r,h,0.1,,,"Enter output image pixel size in longitude (degs): "
image-pixel-lat,r,h,0.1,,,"Enter output image pixel size in latitude (degs): "

This parameter pair specifies the size of the image array pixels and should be set to 0.1^0 or less to allow for the slight blurring of point sources to make them visible in any output image. At present the latitude size is forced to be the same as the longitude size. *i.e.*image pixels will be square.

blur-size,r,h,0.5,,,"Enter FWHM blurring for display of point sources: "

This is a blurring width to output point sources with a nominal width to make them visible in an image. If the output image pixel size is small then the user may not immediate "see" point sources in the image. This is one of the reasons that a pixel size of at least 0.1^o is recommended.

image-proj,s,h,"CAR",,,"Enter output image projection type: "

This parameter has the values **CAR**, **TAN** or **AIT**.

image-orient,s,h,"STANDARD",,,"Enter output image orientation: "

image-pole-long,r,h,0.0,,,"Enter longitude of output image pole (degs): "

image-pole-lat,r,h,90.0,,,"Enter latitude of output image pole (degs): "

These parameters are not used by SPIROS yet and defaulted to the values shown.

3.2.9 Source location parameters for IROS/DIFFUSE mode.

kofsources,s,h,"POINT",,,"Enter kind of sources to be searched for: "

POINT prompts a search for as many point sources as given by parameter **nofsources**. **POINTLIKE** does the same but will try to estimate any width the sources might have. **DIFFUSE** will output image arrays of diffuse emission only.

With this parameter the user can select whether they wish to look for point sources, slightly extended pointlike sources or just a diffuse emission map. It will usually be **POINT** or perhaps **POINTLIKE** if the user is interested in seeing if the sources of interest have some small width.
It is important though to understand that this parameter is also used together with the parameter **pixel-func** mentioned below for imaging diffuse emission.

As mentioned in the overview there are two ways to locate point sources in the observation field of view and the default mode of **SPIROS** is the method of the **Iterative Removal of Sources**, one after the other. This is by far the fastest and simplest method assuming that any sources present can be resolved within the 3^o FWHM of the Point Spread Function of SPI and for this method the parameter **pixel-func** below **must** be set to a blank.

If **pixel-func** is set to anything else then SPIROS will assume the user wishes to create an array of diffuse image pixels and after reconstructing the image will search it for all possible point sources. This may take some time and is only really useful if the user thinks the IROS method cannot separate two sources and wishes to examine the field of view around the two sources on a finer scale to see if two or more sources can in fact be resolved.

Remember, the normal method of using SPIROS is to set **pixel-func** to a blank to invoke the standard IROS method of source location.
If the user only wants a general map of DIFFUSE emission then they must also set the image pixel type and size in **pixel-func** and **pixel-size** below.

nofsources,i,h,1,,,"Enter maximum number of sources to be searched for [0,1,2...]: "

The maximum number of point sources SPIROS will search for unless new source fall below the sigma threshold given by parameter **sigmathres** below in which case it will stop searching.

sigmathres,r,h,3.0,,,"Enter lower sigma threshold to stop source search: "

The sigma threshold to reject new sources located and stop searching for anymore.

In general these three parameters are enough to tell SPIROS just what the user wishes to look for and when it should stop. In most cases the kind of sources will be **POINT**, the sigma rejection threshold will be about **3.0** and the user will only vary the number to be searched for. Even then this can be set to a large number and SPIROS will then keep searching until all possible significant sources have been found.

searchstep,r,h,0.5,,,"Enter source search grid step (degs): "

This parameter sets the search grid step SPIROS will use in scanning the user defined field of view for the location of the next significant point source emission. In general the user should define an output image array with a pixel size of about 0.1^o and a search step of about 0.5^o. SPIROS will then estimate source flux, error and sigma values every 0.5^0 degrees, fill the pixel values in between by linear interpolation and then smooth the whole image a few times to round it out. The **approximate** source location is then estimated by locating the maximum in the smoothed sigma image and then searching on a small 5x5 grid around it for a better solution. The main reason for this is simply to speed up the search for the next source and the user is free to select a search step of the size of the output image pixels if they have a fast machine or time to wait.

srclocbins,s,h,"SUM",,,"Enter count spectrum bins to be used for source location: "

Here also is another attempt to speed up source location. SPIROS will normally try to locate sources in each energy bin of the input count spectrum for an overview. The general idea was that the user should first bin count data via **SPIHIST** in a few coarse energy bins for source location and then again in a large number of bins to extract the spectrum of each source found in two-step approach. With this parameter the user can try a one-step approach by binning count data only once in the large number of energy bins required for spectra then using this parameter to select a smaller subset of bins for source location.

ALL will locate sources in **each** energy bin of the input count spectrum and form a "sigma weighted" mean value to output in the result images.

SUM will locate sources in **one** energy bin covering the input energy spectrum.

A will locate sources in energy bin **A**.

A-B will locate sources in **one** energy bin covering bins **A to B**.

A-BkeV will locate sources in **one** energy bin covering the energy range **A to B keV**.

source-relocation,s,h,"N",,,"Enter if sources are to be relocated over entire spectrum: "

If SPIROS is asked to locate sources in several energy bins it will do so, output images of source locations and residues in each energy bin, then calculate a **sigma weighted** mean location. With this parameter the user can then ask SPIROS to make a further estimate of each source location over the entire count spectrum simultaneously. This may take some time for a large number of sources and a large number of energy bins.

location-max-error,r,h,0.1,,,"Enter maximum allowed input source location error: "

This parameter is only necessary in **IROS/DIFFUSE** imaging mode where an input catalogue of known sources is specified. When SPIROS reads in the sources from the catalogue it will compare their location errors with this error parameter. If they are LESS than it the source will be flagged as having a **FIXED** location and width which will not be changed by SPIROS, otherwise they will be flagged as **VARIABLE** and their locations will be allowed to shift to a more optimum location when new sources are located.

srclocprec,r,h,0.01,,,"Enter source location stopping precision [degrees]: "

In searching for sources SPIROS is always allowing them to simultaneously "float" to more accurate locations and this parameter is the location precision stopping criteria. In iteratively finding better locations SPIROS finds the **largest** change the location of any source. If it falls within the value of **srclocprec** and the change in the ML parameter within **maxlikprec** then the current source locations are returned as optimal.

srcwidprec,r,h,0.1,,,"Enter source width stopping precision [degrees]: "

In searching for **POINTLIKE** sources SPIROS also allows their widths to vary. If the largest change in any source width falls within **srcwidprec** and the change in the ML parameter within **maxlikprec** then the current source widths are returned as optimal.

chilocstep,r,h,0.1,,,"Enter source location sampling step [degrees]: "

chiwidstep,r,h,0.5,,,"Enter source width sampling step [degrees]: "

These two parameters are for any developer and give the so called **sampling** step in finding the local functional form of the ML parameter and the **search** step in finding the next local minimum of the ML parameter in a given search direction. The main point is that they should not be too small " < 0.0001" for speed of calculation and because variations in the ML parameter might not be smooth but discrete due to rounding or other errors.

3.2.10 Image pixel type and size for DIFFUSE imaging mode.

pixel-func,s,h," ",,,"Enter shape of image pixel or spline function: "

As noted above this parameter is important if the user wants a diffuse emission map only or wants to create one so that SPIROS can search it for all possible sources at once and not one after the other in the IROS method. In general **pixel-func** will be set to a blank and the user will be in IROS search mode.

If the user does want a diffuse emission map then they can select here the kind of pixel to be used to build up the image surface shape. Normally such an image is made up of a rectangular array of square **HAT** pixels but the user can select a **BSPLINE** shaped overlapping pixel which will give an image surface cubically smooth and continuous in both its first and second derivatives.
It is also possible to choose **LINEAR** overlapping pixels for a linearly interpolated image surface or a regular array of **POINT** sources.

In general **HAT** pixels are most commonly used and fastest whereas **BSPLINE** pixels can take a long time to reconstruct. It is also to be noted that the parameter **solution-constr** below can be set to force a positivity constraint on diffuse image pixel heights and is the simplest means of suppressing image noise.

pixel-size,r,h,2.0,,,"Enter size of image pixels of FWHM of B-spline function: "

Here the size of the above pixels can be given. It is important to note the difference between this pixel size and that of the output image defined by parameter **image-pixel-long, image-pixel-lat** above and which is only for output display purposes to show whatever size scale the user is interested in.

If **pixel-func** is not blank then SPIROS will create a rectangular array of such pixels to cover the field of view defined for output images above. If **pixel-size** is too small a large number may be required to cover the field of view selected having two main effects, one of which is the long time required to construct the imaging equations required and then to solve them.
The other is the effect of "crosstalk" and "noise" in an image which may result in an image where any significant structure is overwhelmed by large oscillations of amplified noise from the input count data. This can be suppressed but requires the extra complication of choosing an bf image surface constraint with the next group of parameters below.

In general diffuse emission should be reconstructed for a pixel size of $2.0 - 3.0^o$, around the FWHM of the instrument PSF. However if the user thinks the signal to noise ratio is high enough a pixel size of $0.5 - 1.0^o$ used with a positivity constraint is often enough to bring out point source structure.

3.2.11 Image constraint parameters for DIFFUSE imaging mode.

constrtype,s,h," ",,,"Enter image constraint matrix type: "

In reconstructing diffuse emission images the large number of image pixels and a weak signal-to-noise ration can lead to the amplification of noise in the input count data over the image reconstructed. This is sometimes unavoidable and can be often suppressed by the selection of an image positivity constraint via parameter **solution-constr** below.
Otherwise it is required to put a constraint on the surface or topology of the image to be reconstructed by the use of an **image constraint matrix**. The reason this is a matrix is that it is generally used to modify the Hessian matrix of second derivatives used in a 2nd order Taylor expansion to locally describe the behaviour of the ML statistic or objective function.
The simplest of such constraints is either a Wiener diagonal or smoothing matrix which try to flatten the image or any ripples in its surface. Such constraint matrices are quite effective in suppressing the spread of large noise oscillations in the image surface but always do so at the expense of blurring the image having the effect similar to putting an ever increasing surface tension in the image surface.

constrmult,r,h,0.1,,,"Enter image constraint matrix multiplier: "

If a constraint matrix is chosen the next question is how much it should constrain the surface. For a high signal-to-noise ratio little constraint is required and for a very weak signal-to-noise ratio the constraint might need to large enough to leave only a few scattered "blobs" visible. For the constraint matrix chosen this is done by a "constraint matrix multiplier" to multiply the values in the matrix uniformly. For the methods SPIROS currently uses this multiplier should be roughly the same as the **noise-to-signal** ratio of the image though changes are underway to improve this estimate to a more robust optimal value.
One way to use this parameter is to give it a range of values and let the user choose the image which balances the noise against the structure they wish to see in it and this can be done with the following parameters **constriter** and **constrincr**.

constriter,i,h,0,,,"Enter number of image constraint iterations: "

If larger than one then SPIROS will reconstruct **constriter** images each having a different value of the "constraint matrix multiplier" above and which are caclulated from the parameter bf constrincr below. The idea behind this is to give the user an overview of the noise created in each image and how they might choose their optimal image.

constrincr,r,h,0.0,,,"Enter image constraint matrix multiplier increment: "

The parameter bf constrincr will be used to construct a range of multiplier values from **1.0/constrincr** to **constrincr** in **logarithmic** divisions for bf constriter different values and an image for each will be output. If it is set to zero then the range of **constriter** multiplier values will be from 0.01 to 100 and this is often sufficient for a overview of how much noise is to be expected in images.

3.2.12 Output datasets for SPECTRAL mode.

SPIROS in **SPECTRAL** mode allows for the output of three different kinds of spectra.

source-spec-idx,s,h,"Crab-spectra-index.fits(SPI.-SRC.-SPE-IDX.tpl)",,,
"Enter name of output source spectra index: "

source-spec,s,h,"Crab-spectra-data.fits",,,"Enter output source spectra dataset: "

This parameter pair specifies the output index file for all **source flux spectra** along with the name of the dataset file in which their spectra are stored.
The index construction template appended to its name is mandatory.
SPIROS will abort if an index filename is not specified.

back-det-spec-idx,s,h,"Crab-backg-spectra-index.fits(SPI.-BACK-DSP-IDX.tpl)",,,
"Enter name of output detector background spectra index: "

back-det-spec,s,h,"Crab-backg-spectra-data.fits",,,
"Enter name of output detector background spectra dataset: "

This parameter pair specifies the output index file for **background detector count spectra** along with the name of the dataset file in which their count spectra are stored.
The index construction template appended to its name is mandatory.
If "NO" is entered there will no output of this kind of spectrum.

source-det-spec-idx,s,h,"NO",,,"Enter output source detector spectra index: "

source-det-spec,s,h," ",,,"Enter output source detector spectra dataset: "

This parameter pair specifies the output index file for all **source detector count spectra** along with the name of the dataset file in which their count spectra are stored.
The index construction template appended to its name is mandatory.
If "NO" is entered there will no output of this kind of spectrum.

3.2.13 Parameters and output datasets for TIMING mode.

SPIROS in **TIMING** mode will group pointing exposures into groups spanning a user defined integration time interval to compute any significant flux variation within it.

source-timing-mode,s,h,"QUICKLOOK",,,"Enter timing mode analysis method: "

This parameter specifies the analysis mode to compute timing light curves for output.

- **QUICKLOOK**
  This computes the flux for all sources and background in small groups of pointing exposures covering the integration time interval, in parameter **source-timing-scale** below, and in each count spectrum energy bin.
- **WINDOW**
  This mode assumes a MCM or known background variations and computes source flux values simultaneously in a sequence of integration time intervals spanning the period of observation. For a large number of integration intervals and energy bins this might take some time to calculate.
- **TRANSIENT**
  This upgraded mode is similar to **WINDOW** mode but allows the removal of the discontinous nature of the **HAT** like timebin functions to return an implicitly smoothed light curve. The integration time intervals spanning the period of observation are used here to create a sequence of **splining nodes**, or knotpoints, which will define a sequence of piecewise or smoothly continuous **B-spline functions**.
  Select either TRANSIENT-HAT, TRANSIENT-LINEAR, TRANSIENT-QUADR, or TRANSIENT-CUBIC. If no **-XXXX** function is appended a simple **HAT** type is selected and the results should be the same as in **WINDOW** mode.
  These functions are actually created by taking a HAT function covering each time interval and smoothing it with a HAT filter several times over to create LINEAR, QUADRATIC or CUBIC B-spline functions, a sequence which will tend towards a Guassian shape with each further smoothing.

source-timing-scale,r,h,0.0,,,"Enter timing mode integration time interval: "

This parameter specifies the integration time interval used to group observation exposures in computing transient source flux variations with time over the observation period.

source-timing-idx,s,h,"Crab-timing-index.fits(SPI.-SRC.-LCR-IDX.tpl)",,,
"Enter name of output source timing index: "

source-timing,s,h,"Crab-timing-data.fits",,,"Enter output source timing dataset: "

This parameter pair specifies the output index file for all source **light curves** along with the name of the dataset file in which their light curves are stored.
The index construction template appended to its name is mandatory.

3.2.14 TIMING TRANSIENT option catalogue parameters.

This mode is useful in that it allows the user to specify a different integration interval and some cyclic transient function model for each source via its parameters in the input catalogue. It might be that some sources need to be analysed on a finer time scale than most or that they are known to have a cyclical behaviour with a known period and phase, thus removing the need to define a large number of integration bins to extract the mean or variable cyclical amplitude.

The catalogue parameters for cyclic models and a different time scale are as follows:

VAR_MODL
Entering a name here will cause SPIROS to overide the normal TIMING mode parameters and allow the user to model specific sources. The text has two independent components *"transient-type,model-type"*, separated by a comma, the first being the name of a spline function to describe the long term transient behaviour and the second being an additional, though not necessary, type of model to describe any special cyclical or flaring component.

The first option *"transient-type"* can have the values **CONSTANT**, **LINEAR**, **QUADR** or **CUBIC** and overides the type common to all sources appended after the **TRANSIENT-XXXX** option described above.

The second option *"model-type"*, (don't forget the comma!), has the possibilities

- **SINE**
  This appends a single **SINE** function whose period and MJD-phase at minimum are known, and described in **VAR_PARS** below, for SPIROS to determine its amplitude.

- **SINPLO**
  The **SINE** function above may introduce negative counts when added to the underlying long term transient component so **SINPLO** will append a modified positive "SIN+1" model to ensure that the total amplitude can be forced to be positive. Otherwise it should return the same result as the **SINE** model.

- **SINCOS**
  This appends both a **SINE** and **COSINE** function whose period only has to be known, and described in **VAR_PARS** below, for SPIROS to determine both the amplitude and implicit phase of a cyclic component. It returns the amplitude of two extra components and might first be used to determine the MJD-phase before repeating the analysis using a single **SINE** or **SINPLO** model.

- **XXXXX-CYCLE**
  This may be used for a cyclic component which is not perfectly cyclic but skewed or flattened out to determine the shape or **phase curve** of the cyclic behaviour. Instead of just one perfectly sinusoidal component here the user can specify the **no of spline components** required to describe any non perfect behaviour covering the period of the cycle using the catalogue parameter **VAR_NPAR**. The type of component **XXXXXX** can have the values **CONSTANT, LINEAR, QUADRATIC or CUBIC** with **QUADR** being sufficient to describe a smooth phase curve. **Three** components will allow for a cyclic central component plus one either side of it to pick up skewed or flattened behaviour. **Five** components will allow for two extra

components on either side if any flattening is wider. If **-SYMM** is also appended the actual no of components required will be less as the side components added will be symmetric about the cycle centrepoint and force the phase curve to be symmetric. It is important to note that this mode does not normally require an MJD-phase time as the cyclic nature of the components used will implicitly determine the MJD-phase in a similar way to the **SINCOS** component described above. Only if **-SYMM** is appended is the MJD-phase of the minimum required.

- **XXXXX-FLARE**
  This is similar to XXXXX-CYCLE but describes a cyclic flare whose period and MJD-phase of its minimum must be known. Again catalogue parameter **VAR_NPAR** must be used to specify the no of components requiredto describe the flaring shape, but they should not be too many, 3,4,5 or 6 but not much more unless the source is strong.

It is important to note that by appending ***TRANS** to the above model options SPIROS will additionally calculate the variation in the amplitude of the cyclic variation otherwise the amplitude returned will be the mean over the observation period.

VAR_NPAR
This is the no of components required to for the cyclic model and the model will not be implemented if this is zero. It is a form of switch to turn the model on. Again don't make it too large.

VAR_PARS
These are a series of eight parameters to specify a cyclic or flaring model or to change the integration time scale taken from the parameter **source-timing-scale** above.

- **VAR_PARS(1) + VAR_PARS(2)**
  The sum of these two is the MJD or IJD at which the **minimum** of a cyclic component occurs. MJD times are too long for one parameter so two are used, the first for the MJD day and the second for the day fraction. SPIROS will always add the two together.

- **VAR_PARS(3)**
  This is the **flaring time** in days of an outburst and is expected to be less than or equal to any cyclic period the flaring might have. If this value is zero it will be reset to the length of the observation period.

- **VAR_PARS(4)**
  This is the **cyclic period** in days of any outburst and is expected to be greater than or equal to any flaring time. If this value is zero it will be reset to the flaring time in **VAR_PARS(3)**.

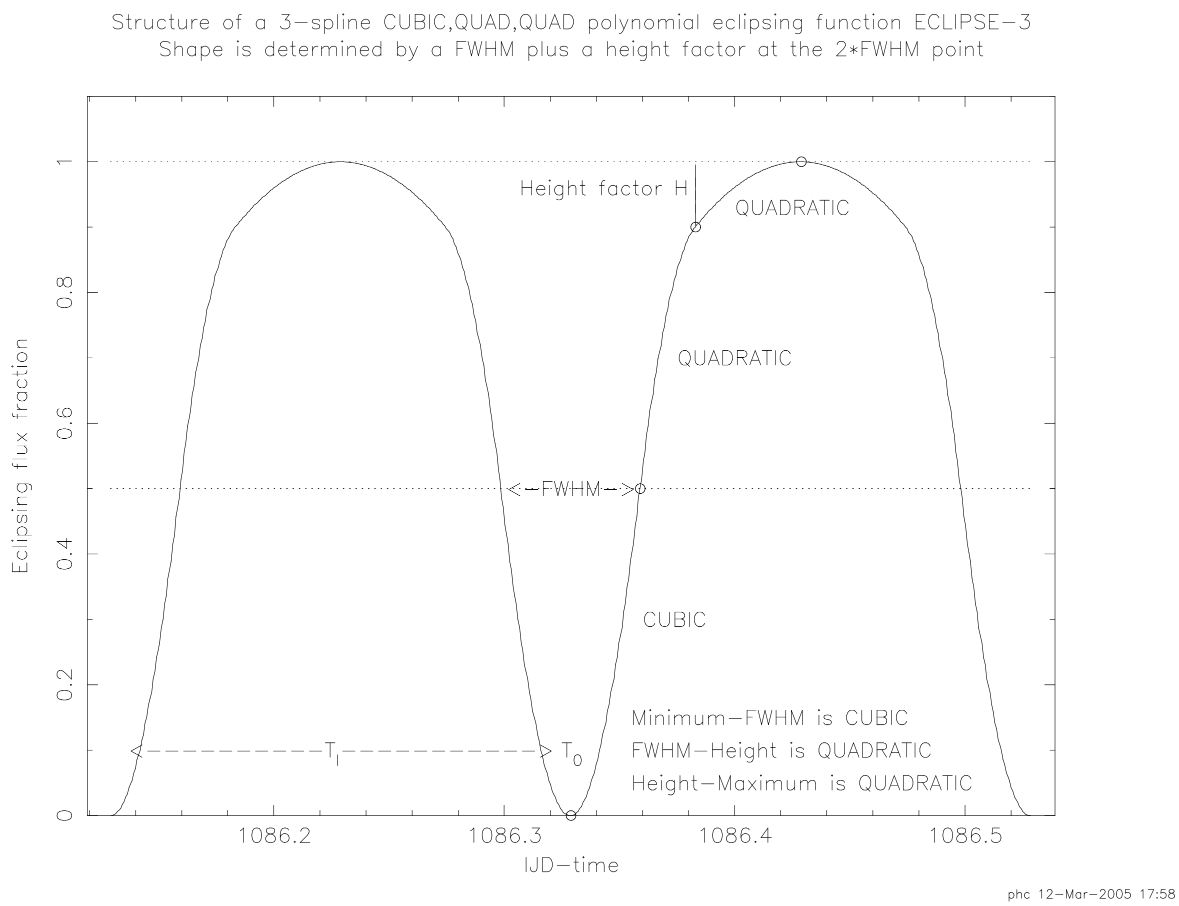
- **VAR_PARS(5)**
  This is the new time-scale, in days, for the long term transient component covering the period of observation and overides the parameter **source-timing-scale** above. This will happen quite independently of the other **VAR_PARS** values, so if it is only the time scale that needs to be changed, this is the only parameter you need to specify.

3.3 Cyclic eclipsing functions for orbital timing analysis.

The light curves of some sources, such as **Cygnus-X3**, show a regular cyclical variation of flux at all energies, a kind of "eclipsing" behaviour, and where the amplitude of the minimum and maxima of the eclipsing flux may also be variable. SPIROS allows the user to specify some simple eclipsing models with a few parameters to provide a more detailed light curve description for further modelling of the geometry of the source system.

In the diagram below a SPIROS eclipsing function is shown which is constructed by dividing the interval between eclipse minimum and maximum into three spline segments to support a CUBIC and two QUADRATIC polynomials and described by four parameters:

- The MJD or time of the eclipse minimum $\mathbf{T_0}$.
- The length of the eclipse from minimum to minimum $\mathbf{T_l}$.
- The "full width half maximum" **FWHM** around the eclipse minimum.
- A "height" or "flatness" factor **H** at the **2*FWHM** point which indicates the degree of "flatness" around the maximum.



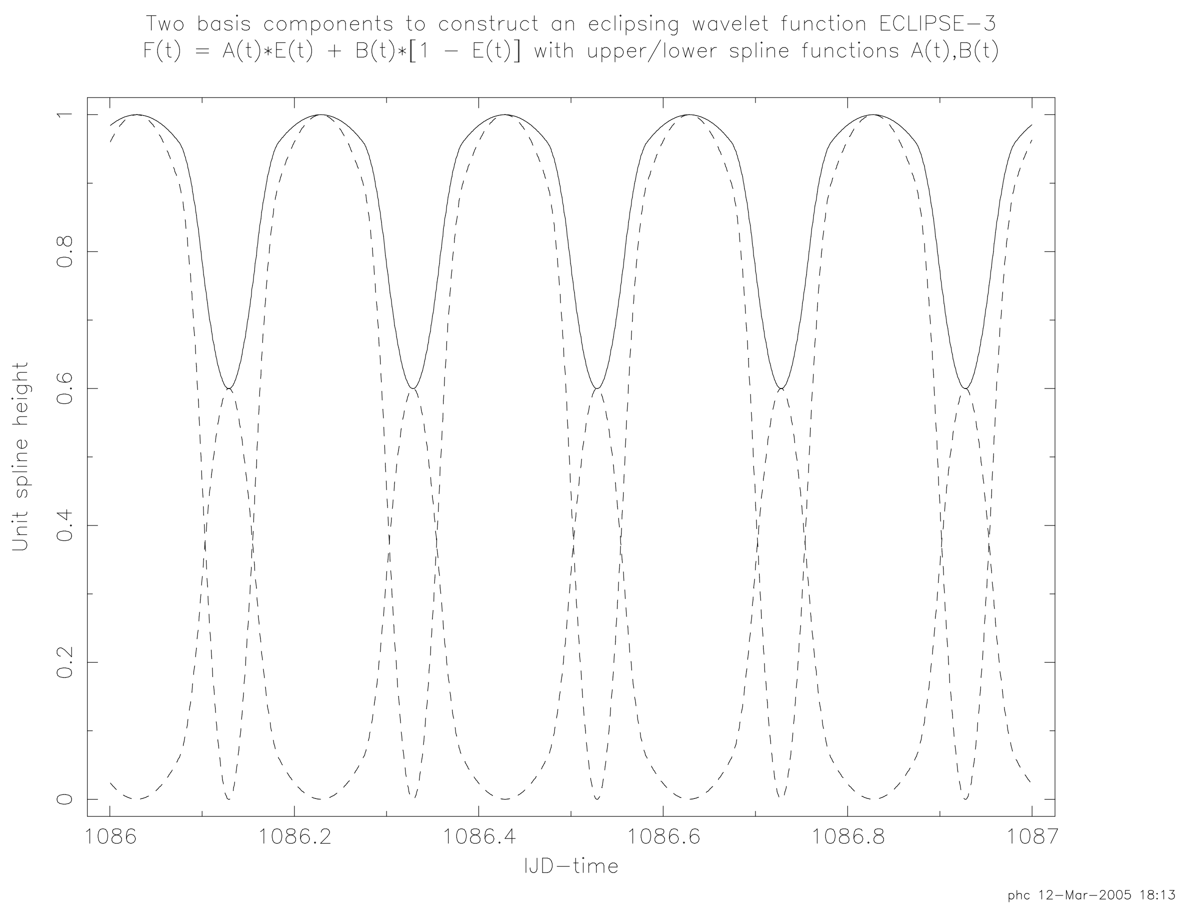
This kind of light curve is most useful for an eclipsing function where the minimum MJD $\mathbf{T_0}$ and period $\mathbf{T_l}$ are known and where the signal to noise ratio in the observation data is not high, and where just a FHWM and a height parameter are sufficient to describe the form of the eclipse from minimum to maximum and not much more detail can really be expected. The FWHM will be the dominant parameter while the height factor merely "rounds out" the description and is not always expected to have a high sigma significance.

The catalogue parameters for this eclipsing model are as follows:

VAR_MODL
This is the name of any time model and has two forms:

- **ECLIPSE**
  SPIROS will create the cyclic function $f(t)$ described above and assume the eclipsing flux is defined by $A * f(t) \; + \; B * [1 - f(t)]$ and will find optimum values for the multipliers A, B. When $A = B$ then the flux will have a constant value A.

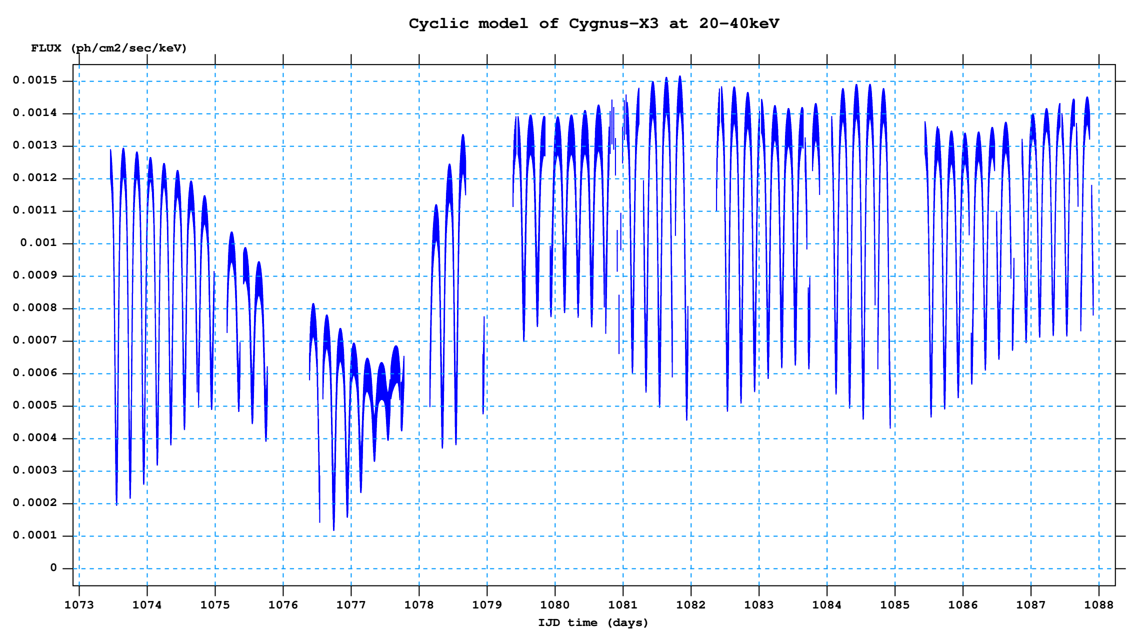
- **ECLIPSE*XXXXX**
  where **XXXXX** is **CONSTANT** , **LINEAR** , **QUADRATIC** or **CUBIC**.
  This is similar to the default case above but SPIROS will assume the eclipsing flux is defined by $A(t) * f(t) \; + \; B(t) * [1 - f(t)]$ where the multipliers $A(t), B(t)$ may vary in time and each will be made up of a number of splines covering a time scale defined in parameter **source-timing-scale** or **VAR_PARS(5)** defined below.



VAR_PARS
These are a series of eight parameters to specify a cyclic or flaring model or to change the integration time scale taken from the parameter **source-timing-scale** above.

- **VAR_PARS(1) + VAR_PARS(2)**
  The sum of these two is the MJD or IJD at which the **minimum** occurs. MJD times are too long for one parameter so two are used, the first for the MJD day and the second for the day fraction. SPIROS will always add the two together.
- **VAR_PARS(3)**
  This is the **flaring time** or eclipse length in days.
- **VAR_PARS(4)**
  This is the **cyclic period** in days of any outburst and here is expected to be the same as the flaring time in **VAR_PARS(3)**.
- **VAR_PARS(5)**
  The time scale of any CONSTANT, LINEAR, QUADRATIC or CUBIC spline time bins or nodes and is a substitute for the parameter **source-timing-scale**.
- **VAR_PARS(6)**
  This is the **FWHM** factor.
- **VAR_PARS(7)**
  This is the height or **flatness** factor.

This is a good model for observations of **Cygnus-X3** where each pointing exposure lasts about 1800 seconds and therefore can only provide an incomplete sampling of the 0.2 day period of each "eclipse" with about 10 exposures. Using a model of type **ECLIPSE*CUBIC** in the 20-40 keV range gives the light curve below and the parameter estimates:



Estimates returned for the model parameters are:

- Errors in the magnitude of the eclipse which give a significance of about 10 sigma.
- A FWHM of the order of 0.31 ± 0.023 of the eclipse period (0.2 days).
- A not quite flat height factor at 0.20 ± 0.11 of the eclipse magnitude.

4 SPIROS parameters in a nutshell

4.1 SPIROS parameters in ALL modes.

First comes the SPIROS analysis mode

mode,s,h,"IROS",,,"Enter mode for imaging, spectral or timing analysis: "

then all observation datasets containing energy binning, exposure times, pointing and count data, most of which are listed in the input observation group defined by **in-og-dol**:

in-og-dol,s,h,"Crab-count-data.fits[1]",,,"Enter name of observation group input DOL: "
ebounds-dol,s,h,"",,,"Enter energy binning boundaries DOL: "
pointing-dol,s,h,"",,,"Enter observation pointing directions DOL: "
gti-dol,s,h,"",,,"Enter observation good time intervals DOL: "
deadtime-dol,s,h,"",,,"Enter observation dead time intervals DOL: "
evts-det-spec-dol,s,h,"",,,"Enter detector event spectra DOL: "
out-og-dol,s,h,"Crab-output-og.fits(GNRL-OBSG-GRP.tpl)",,,

An **Instrument Response Function** is always required and whether the **ERM** part of it is also to be used (for **XSPEC** spectra output it is not):

inst-resp-idx,s,h,"/home/phc/INTEGRAL/ISDC/IRFs/SPI-IRF-index.fits[1]",,,

A **Background Response Function** is always required and how it is to be used:

back-model-idx,s,h,"Crab-backgr-model-index.fits[1]",,,

The ML optimizing statistic and other solution control parameters are required:

optistat,s,h,"CHI2",,,"Enter ML optimization statistic [CHI2,LIKEH]: "
maxlikprec,r,h,0.2,,,"Enter ML optimization precision: "
solution-constr,s,h,"NONE",,,"Enter image solution constraints [NONE,POSITIVE]: "
nagoptions,s,h," ",,,"Enter NAG options file for minimization subroutines: "
background-method,i,h,3,,,"Enter handling of background model response values
energy-response,s,h,"N",,,"Enter if Energy Response Matrix is to be used [Y/N]: "
maxlikfile,s,h,"CHI2.fits(SPI.-MAXL-RES.tpl)",,,"Enter ML residue output file: "

4.2 Additional parameters for IROS or DIFFUSE imaging mode.

An input source catalogue is not essential but probable in IROS/DIFFUSE imaging mode in which case sources must be flagged as having either FIXED or VARIABLE locations:

source-cat-dol,s,h,"",,,"Enter name of catalogue of known sources: "
location-max-error,r,h,0.1,,,"Enter maximum allowed input source location error: "

An output catalogue for source locations and parameters is necessary:

source-res,s,h,"Crab-output-catalogue.fits(SPI.-SRCL-RES.tpl)",,,

The names of output images:

image-idx,s,h,"Crab-images.fits(SPI.-SKY.-IMA-IDX.tpl)",,,
image-int,s,h,"Y",,,"Enter request for flux intensity image [Y,N or image name]: "
image-err,s,h,"Y",,,"Enter request for flux error image [Y,N or image name]: "
image-sig,s,h,"Y",,,"Enter request for flux sigma image [Y,N or image name]: "

The output image field of view, resolution and point source blurring:

image-fov,s,h,"POINTING-CENTRE",,,"Enter extent and location of output image:"
center-long,r,h,0.0,,,"Enter longitude of output image centerpoint (degs): "
center-lat,r,h,0.0,,,"Enter latitude of output image centerpoint (degs): "
image-dim-long,i,h,101,,,"Enter output image array dimension in longitude: "
image-dim-lat,i,h,101,,,"Enter output image array dimension in latitude: "
image-pixel-long,r,h,0.1,,,"Enter output image pixel size in longitude (degs): "
image-pixel-lat,r,h,0.1,,,"Enter output image pixel size in latitude (degs): "
blur-size,r,h,0.5,,,"Enter FWHM blurring for display of point sources: "

What is the imaging coordinate system and its orientation:

reference-coord,s,h,"RADEC",,,"Enter output image reference coordinate system:"
image-proj,s,h,"AIT",,,"Enter output image projection type: "
image-orient,s,h,"STANDARD",,,"Enter output image orientation: "
image-pole-long,r,h,0.0,,,"Enter longitude of output image pole (degs): "
image-pole-lat,r,h,90.0,,,"Enter latitude of output image pole (degs): "

4.3 Additional parameters for IROS/DIFFUSE location imaging mode.

Most important - what are you looking for and how should you search. Remember you are most likely to use the **IROS** search mode so set **pixel-func** below to a blank:

kofsources,s,h,"POINT",,,"Enter kind of sources to be searched for: "

nofsources,i,h,1,,,"Enter maximum number of sources to be searched for [0,1,2...]: "

srclocbins,s,h,"SUM",,,"Enter count spectrum bins to be used for source location: "

searchstep,r,h,0.5,,,"Enter source search grid step (degs): "

What are the criteria to stop searching for sources?

sigmathres,r,h,3.0,,,"Enter lower sigma threshold to stop source search: "

srclocprec,r,h,0.1,,,"Enter source location stopping precision [degrees]: "

srcwidprec,r,h,0.1,,,"Enter source width stopping precision [degrees]: "

chilocstep,r,h,0.1,,,"Enter source location sampling step [degrees]: "

chiwidstep,r,h,0.5,,,"Enter source width sampling step [degrees]: "

Are location maps required to be output at each search iteration and should all sources be finally located using all spectrum energy bins simultaneously?

iteration-output,s,h,"Y",,,"Enter if image output required at each iteration [Y,N]: "

source-relocation,s,h,"N",,,"Enter if sources are to be relocated over entire spectrum: "

These parameters are only useful if diffuse emission imaging is required otherwise set **pixel-func** below to a blank:

pixel-func,s,h," ",,,"Enter shape of image pixel or spline function:

pixel-size,r,h,2.0,,,"Enter size of image pixels of FWHM of B-spline function: "

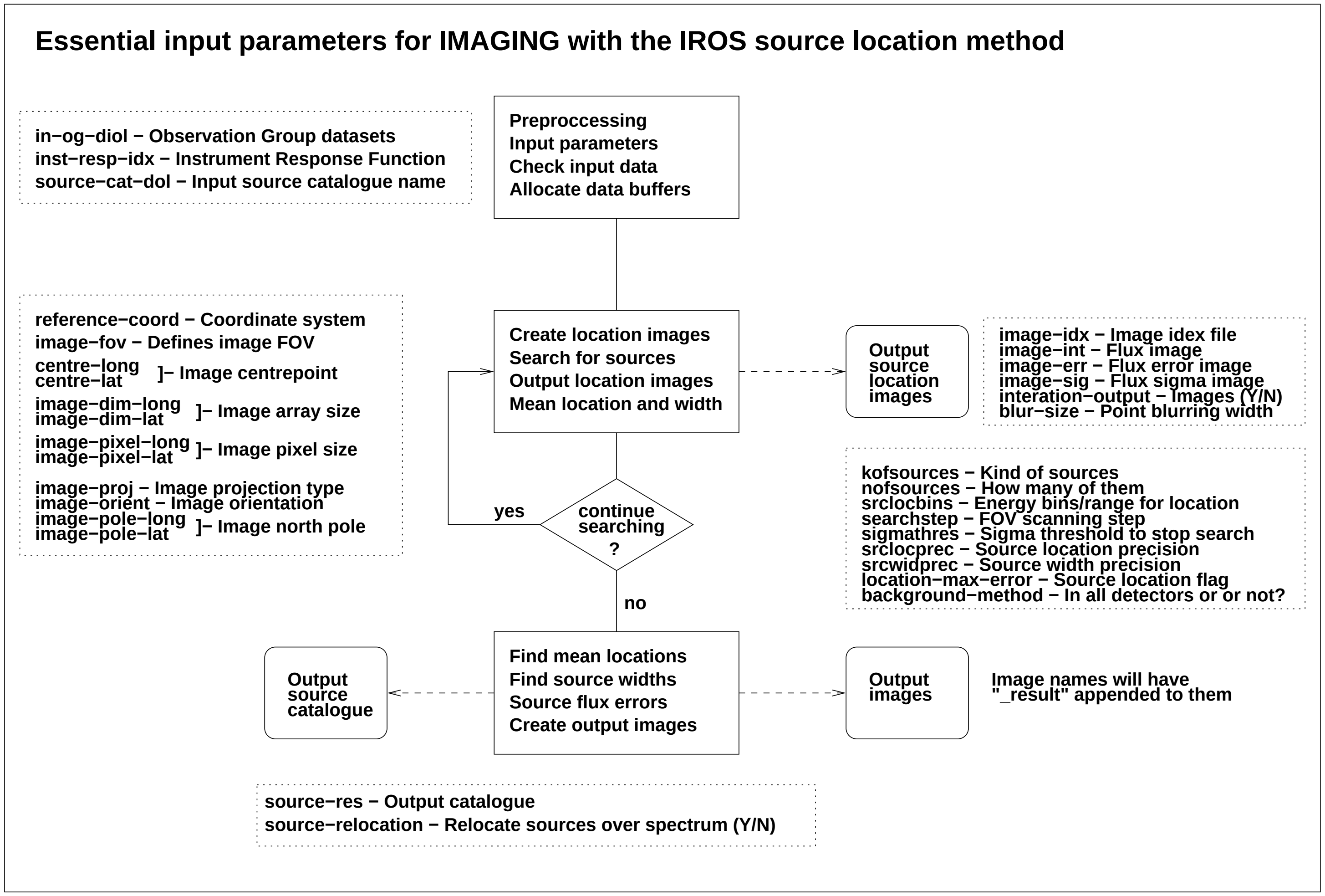
If you want diffuse emission maps but have a low signal-to-noise ration two should the image surfaces be constrained to reduce noise amplification in them?

constrtype,s,h," ",,,"Enter image constraint matrix type:

constrmult,r,h,0.1,,,"Enter image constraint matrix multiplier: "

constriter,i,h,0,,,"Enter number of image constraint iterations: "

constrincr,r,h,0.0,,,"Enter image constraint matrix multiplier increment: "


Essential input parameters for IMAGING with the IROS source location method
in−og−diol − Observation Group datasets
inst−resp−idx − Instrument Response Function
source−cat−dol − Input source catalogue name
Preproccessing
Input parameters
Check input data
Allocate data buffers
reference−coord − Coordinate system
image−fov − Defines image FOV
centre−long
centre−lat]− Image centrepoint
image−dim−long
image−dim−lat]− Image array size
image−pixel−long
image−pixel−lat]− Image pixel size
image−proj − Image projection type
image−orient − Image orientation
image−pole−long
image−pole−lat]− Image north pole
Create location images
Search for sources
Output location images
Mean location and width
Output source location images
image−idx − Image idex file
image−int − Flux image
image−err − Flux error image
image−sig − Flux sigma image
interation−output − Images (Y/N)
blur−size − Point blurring width
yes
continue searching ?
no
kofsources − Kind of sources
nofsources − How many of them
srclocbins − Energy bins/range for location
searchstep − FOV scanning step
sigmathres − Sigma threshold to stop search
srclocprec − Source location precision
srcwidprec − Source width precision
location−max−error − Source location flag
background−method − In all detectors or or not?
Output source catalogue
Find mean locations
Find source widths
Source flux errors
Create output images
Output images
Image names will have
"_result" appended to them
source−res − Output catalogue
source−relocation − Relocate sources over spectrum (Y/N)

4.4 Additional parameters for SPECTRAL mode.

Here you only have to specify a catalogue of known sources

source-cat-dol,s,h,"",,,"Enter name of catalogue of known sources: "

whether you want source spectra output for XSPEC etc

source-spec-idx,s,h,"Crab-spectra-index.fits(SPI.-SRC.-SPE-IDX.tpl)",,,
source-spec,s,h,"Crab-spectra-data.fits",,,"Enter output source spectra dataset: "

or detector background count spectra output

back-det-spec-idx,s,h,"Crab-backg-spectra-index.fits(SPI.-BACK-DSP-IDX.tpl)",,,
back-det-spec,s,h,"Crab-backg-spectra-data.fits",,,

or even source detector count spectra (???)

source-det-spec-idx,s,h,"NO",,,"Enter output source detector spectra index: "
source-det-spec,s,h," ",,,"Enter output source detector spectra dataset: "

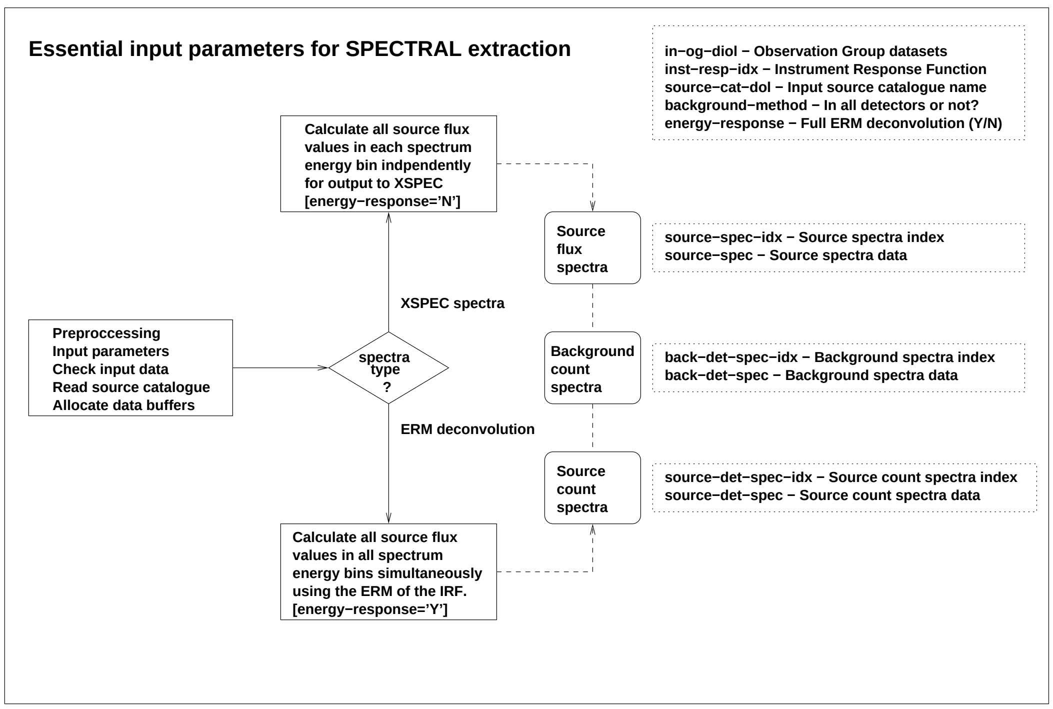


Figure 9: Essential parameters for SPECTRAL extraction mode.

4.5 Additional parameters for TIMING mode.

As in **SPECTRAL** mode an input catalogue of known sources is necessary:

source-cat-dol,s,h,"",,,"Enter name of catalogue of known sources: "

What sort of timing analysis is required and on what integration time scale

source-timing-mode,s,h,"QUICKLOOK",,,"Enter timing mode analysis method: "
source-timing-scale,r,h,0.0,,,"Enter timing mode integration time interval: "

and where do you want to put it?

source-timing-idx,s,h,"Crab-timing-index.fits(SPI.-SRC.-LCR-IDX.tpl)",,,
source-timing,s,h,"Crab-timing-data.fits",,,"Enter output source timing dataset: "

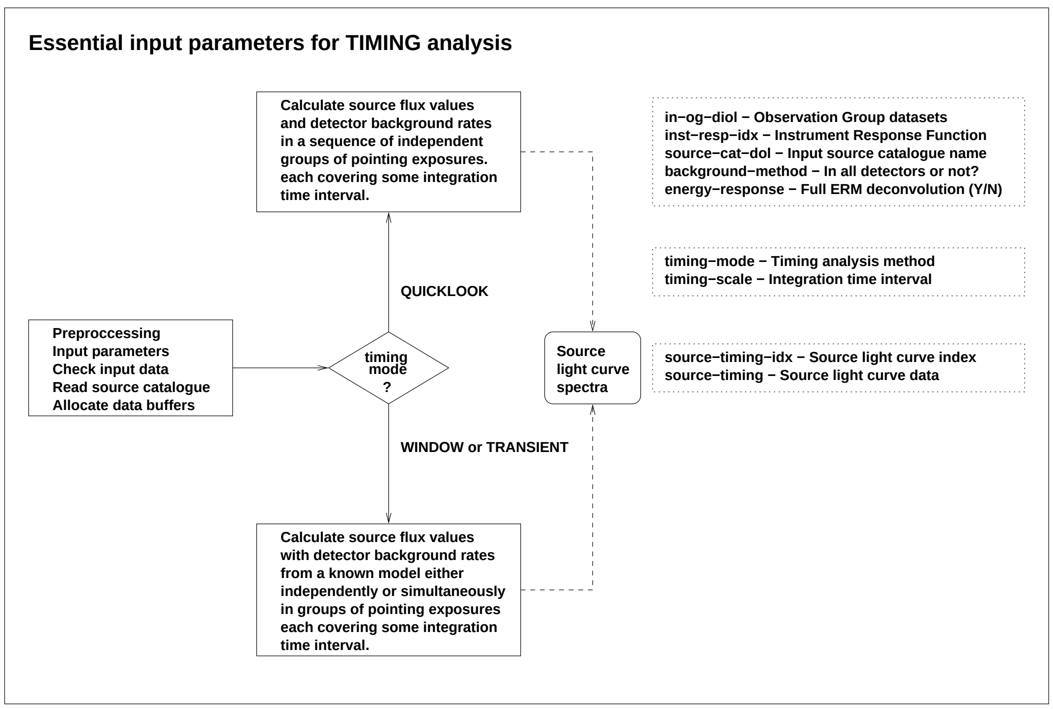


Figure 10: Essential parameters for TIMING analysis mode.

5 Tips and tricks for new users

Running SPIROS in **SPECTRAL** and **TIMING** mode is not very difficult as there are few parameters to specify, mostly output file names. It is **IROS** or **DIFFUSE** imaging mode that has most parameters and some advice is needed so that they are not set to extreme values that will require a long execution time or massive amounts of memory for which there is no room.

5.1 Essential parameters required to do anything at all.

Remember that no matter what mode is run SPIROS always requires an **Observation Group** dataset whose name is specified by parameter **in-og-dol** and which contains only a list of datasets each containing information about the observation exposures, pointing directions, and detector count spectra returned from the spacecraft. SPIROS allows alternative names to these datasets to be specified but generally this will not be the case as the observation data will mostly be created by a pipeline of preprocessing programs which will ensure all the observation datasets required are grouped togther correctly.

The other main common component is the **Instrument Response Function** specified by parameter **inst-resp-idx** if it is not also listed in the **Observation Group**.

The corresponding **Background Response Function** is specified by the parameter **back-model-idx** but in most cases this will also be listed in the **Observation Group**.

So you have to start by specifying parameter **in-og-dol** and you may have to specify parameter **inst-resp-idx** but then you will have observation data and be able to simulate how SPI detects and registers source and background photons.

5.2 Essential parameters for IROS imaging mode.

There are two essential steps to imaging, first deciding on the celestial field of view to be imaged and then selecting what sort of sources to look for and how many.

The user might know in advance exactly where and how large a FOV they wish to image but SPIROS does make life a little easier if they are uncertain. The image to be output is always a rectangular array of pixels whose heights represent source flux intensity and its FOV is specified by its centrepoint, array dimensions and pixel size. The user will always need to specify the pixel size via parameter **image-pixel-long** and **image-pixel-lat** which are currently made equal.

In general the image pixel size **image-pixel-long** should be about $\mathbf{0.1^o}$ or small enough to accomodate a point source blurred by the factor given in parameter **blur-size** which will normally be about $\mathbf{0.5^o}$ wide. Point sources can get "lost" in an image if they occupy just one tiny pixel and SPIROS allows this blurring factor to spread them out so they may be made to stand out.

5.3 Essential parameters to define the imaging field of view.

The next essential parameter is **image-fov** to request the FOV required. If it begins with **POINTING** the pointing directions of each observation exposure will be used to generate a FOV which covers them. If **POINTING** only is specified the image centrepoint will be set to the mean direction of all observation exposures and the rectangular width of the FOV will be calculated to include all exposure directions. From this the dimensions of the image pixel array required will be calculated.

If **POINTING-FCOV** is specified the FOV will be the extended outward to include the **Fully Coded Field of View** of SPI by an angle (**F** $\approx$ **8°**) measured from the longitudinal axis of SPI where the detector can still "see" inside the mask pattern boundary.

If **POINTING-ZCOV** is specified the FOV will be the extended outward to include the **Zero Coded Field of View** of SPI by an angle (**Z** $\approx$ **16°**) measured from the longitudinal axis of SPI where the detector only "sees" outside the mask pattern boundary and has zero coding information. This will generally be the largest field of view where any source can be expected to be imaged.

If **POINTING-CENTRE** or **POINTING-CENTER** is specified the FOV will be NOT be calculated, only the image FOV centrepoint will be set to the mean direction of all observation exposures and the user must then define its width by the size the image pixel array in **image-dim-long** and **image-dim-lat**.

If **USER** is specified **ALL** image dimension parameters must be specified.

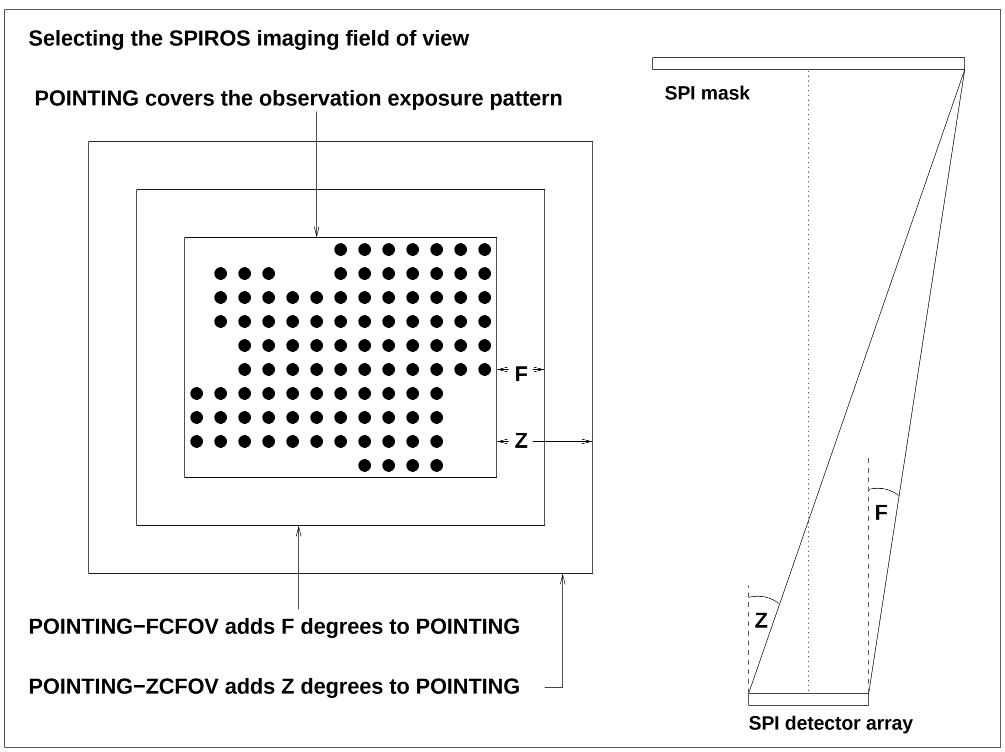


Figure 11: How parameter "image-fov" selects the output image field of view.

So be careful not to specify a tiny pixel size and a large width of view which may result in an extremely large image pixel array and not enough memory to accomodate it. It may help to look first into the observation pointing dataset to find the width of the observed field. There is a program called **spiobs** which allows the user to enter the name of the **Observation Group** dataset and display basic information about the observation such as the pointing directions, exposure times and detector count patterns for each energy bin.

5.4 Essential parameters to search for sources in IROS mode.

The essential parameters to search for sources are **kofsources** and **nofsources** to select what you wish to look for and how many at most you wish to locate. Normally **kofsources** is set to **POINT** to look for point sources but if you wish to find out if these sources have any width then make it **POINTLIKE**. This will take longer as SPIROS will first execute exactly as for **POINT**, and when finished, add the extra step of finding any width they may have.

If **nofsources** is set to zero no sources will be searched for and SPIROS will output convolved **correlation images** which give a quick overall view of what might be in the field of view selected. If you want SPIROS to look for all sources that are significant above the level set by parameter **sigmathres** then set **nofsources** to some large value or the largest number of sources you can imagine might be visible.

Another point to note in searching for sources is that the larger the field of view the longer it will take SPIROS to scan it to build up its source location maps. SPIROS also has a parameter **searchstep** which defines the size of a search grid and this is defaulted to a $\mathbf{0.5^o}$ so a $10\text{x}10^o$ FOV width will have a 20x20 grid of search points laid over it to be scanned.

In fact when first imaging it is a good idea to define a small FOV such as a 51x51 array of 0.1^o pixels to cover a $5\text{x}5^o$ field of view and not ask to search for any sources. This will return a single image which can give the user an initial idea of what may be in the FOV, if it should be larger and how long it will take to execute when SPIROS searches for sources in it. Don't start a large size problem and have to wait for a long time to find a parameter has been overestimated.

6 SPIROS mathematics and logic

6.1 The SPIROS mathematical model.

SPIROS uses a method of image reconstruction resulting in source locations and their spectral flux intensity values based on the following mathematical model of the effect of any number of celestial and detector background sources on the actual detector counts recorded by SPI.

$$\mu_{e,d,p} \;=\; \sum_{\epsilon,s} M_{e,d,p:\epsilon,s}\;\alpha_{\epsilon,s} \tag{1}$$

or in matrix/vector notation

$$\underline{\mu}(\underline{\alpha}) \;=\; [M]\,\underline{\alpha} \tag{1a}$$

where its components are

$\boxed{\mu_{e,d,p}}$

The expected counts in detector ($d = 1, N_d$) during pointing exposure ($p = 1, N_p$) in count spectrum energy bin ($e = 1, N_e$) due to all sources. The basic problem SPIROS will try to solve is to determine the location and spectrum flux values of sources which will produce such count data that "best fit" the real observation count data $\nu_{e,d,p}$ which is input to SPIROS and is derived from photon event data processed by the workpackage **SPIHIST**.

$\boxed{M_{e,d,p:\epsilon,s}}$

The **Instrument Response Function** or count response expected in detector (d) during pointing exposure (p) in count energy bin (e) from source ($s = 1, N_s$) at location (x_s, y_s) in source spectrum energy bin (ϵ) with unit flux. This is quite general and represents both detector background and celestial sources. In fact the algorithms SPIROS uses do not distinguish between these sources except to flag them as different and call up one of two kinds of **IRF**. So this component represents two different kinds of function. The count spectrum bins (e) may be quite different from the source spectrum bins ($\epsilon = 1, N_\epsilon$) but often will be the same and because of energy convolution results in source photons in any energy bin being distributed in count energy bins below it. Its units are those of exposure ($cm^2 sec$).

$\boxed{\alpha_{\epsilon,s}}$

The source flux from celestial or detector background source ($s = 1, N_s$) at location (x_s, y_s) in source energy bin (ϵ). Its units are $ph/cm^2/sec/bin$ for sky sources or $ph/sec/det/bin$ for detector background. The summation accumulates the effect of all sources over their spectral range.

6.2 Finding an optimal solution.

As noted above the solution for all source and detector background flux values $\alpha_{\epsilon,n}$ is that which produces a "best fit" to the actual observation count data $\nu_{d,p,e}$ and this is found by optimizing an objective function or **Maximum Likelihood** parameter. This is found by noting that any source flux values $\alpha_{\epsilon,n}$ will produce expected counts $\mu_{d,p,e}$ that have a certain probability of occuring and the ML parameter is some function of this probability which is to be maximized.

If the observation counts $\nu_{d,p,e}$ are large enough their statistical fluctuations are assumed to be **Gaussian** relative to $\mu_{d,p,e}$ with the probability

$$Pr\{\ \underline{\nu}\mid\underline{\mu}(\underline{\alpha})\ \} \ = \ \prod_{d,p,e}\left[\frac{1.0}{\sqrt{2\pi\sigma^2_{d,p,e}}}\ e^{-[\nu_{d,p,e}-\mu_{d,p,e}]^2/2\sigma^2_{d,p,e}}\right] \tag{2}$$

which is to be **maximized** and by taking $-2\ln[Pr\{\ \underline{\nu}\mid\underline{\mu}(\underline{\alpha})\ \}]$ we can extract the well known ML parameter χ^2 which is then to be **minimized**

$$L(\underline{\alpha}) \ = \ \sum_{d,p,e}\left[\frac{[\nu_{d,p,e}-\mu_{d,p,e}]^2}{\sigma^2_{d,p,e}}\right] \ = \ \chi^2(\underline{\alpha}) \tag{3}$$

If the observation counts $\nu_{d,p,e}$ are small then their statistical fluctuations are assumed to be **Poisson** with the probability

$$Pr\{\ \underline{\nu}\mid\underline{\mu}(\underline{\alpha})\ \} \ = \ \prod_{d,p,e}\left[e^{-\mu_{d,p,e}}\frac{\mu_{d,p,e}^{\nu_{d,p,e}}}{\nu_{d,p,e}!}\right] \tag{4}$$

Taking $-2\ln[Pr\{\ \underline{\nu}\mid\underline{\mu}(\underline{\alpha})\ \}]$ we now extract the ML **Cash** statistic which to be minimized

$$L(\underline{\alpha}) \ = \ 2\sum_{d,p,e}[\mu_{d,p,e}-\nu_{d,p,e}\ln(\mu_{d,p,e})] \tag{5}$$

In both cases an optimal solution $\underline{\alpha}^{opt}$ is that which results in counts $\mu_{d,p,e}=\mu^{opt}_{d,p,e}$ such that either ML parameter is minimized. For both parameters the **Likelihood Ratio**

$$\mathcal{L}(\underline{\alpha}^{opt}) \ = \ -2\ln\left(\frac{Pr\{\ \underline{\nu}\mid\underline{\mu}^{opt}\ \}}{Pr\{\ \underline{\nu}=\underline{\mu}^{opt}\ \}}\right)$$

has a mean value and variance equal to the **Degrees of Freedom**.

6.3 Optimizing the ML parameter.

To find the solution $\underline{\alpha}$ that minimizes the ML parameter an iterative scheme is used whereby a second order Taylor expansion of $L(\underline{\alpha})$ around a current solution is created and used to find a more optimal solution in its neighbourhood.

For simplicity the solution vector $\alpha_{\epsilon,s}$ is changed to α_n where $n = 1, N_\epsilon N_s$ numbers all source spectrum bins sequentially starting at $s = 1$ and the Taylor expansion is written as

$$L(\underline{\alpha} + \delta\underline{\alpha}) \;=\; L(\underline{\alpha}) \;+\; \sum_n \left(\frac{\partial L}{\partial \alpha_n}\right)_{\underline{\alpha}} \delta\alpha_n \;+\; \frac{1}{2}\sum_n \sum_m \left(\frac{\partial L}{\partial \alpha_n \alpha_m}\right)_{\underline{\alpha}} \delta\alpha_n \delta\alpha_m \tag{6}$$

or

$$L(\underline{\alpha} + \delta\underline{\alpha}) \;=\; L(\underline{\alpha}) \;+\; \sum_n \nabla L_n(\underline{\alpha})\; \delta\alpha_n \;+\; \frac{1}{2}\sum_n \sum_m \nabla^2 L_{n,m}(\underline{\alpha})\; \delta\alpha_n \delta\alpha_m \tag{6a}$$

or

$$L(\underline{\alpha} + \delta\underline{\alpha}) \;=\; L(\underline{\alpha}) \;+\; \sum_n G_n(\underline{\alpha})\; \delta\alpha_n \;+\; \frac{1}{2}\sum_n \sum_m H_{n,m}(\underline{\alpha})\; \delta\alpha_n \delta\alpha_m \tag{6b}$$

or

$$L(\underline{\alpha} + \delta\underline{\alpha}) \;=\; L(\underline{\alpha}) \;+\; G(\underline{\alpha})\; \delta\underline{\alpha} \;+\; \frac{1}{2}\delta\underline{\alpha}^t\; [H(\underline{\alpha})]\; \delta\underline{\alpha} \tag{6c}$$

in matrix/vector notation where

- $\delta\underline{\alpha}$ is a change in the current solution $\underline{\alpha}$
- $G(\underline{\alpha})$ is the ML **Gradient** vector
- $[H(\underline{\alpha})]$ is the symmetric positive definite ML **Hessian** matrix of 2^{nd} derivatives

If the count noise is **Gaussian** then it can be shown (section 6.4) that the **Hessian** matrix is independent of $\underline{\alpha}$ and the optimal change $\delta\underline{\alpha}$ in $\underline{\alpha}$ to minimize equation (6) is the solution to the set of simultaneous equations

$$G(\underline{\alpha}) \;=\; [H]\; \delta\underline{\alpha} \tag{7}$$

If the iteration required by equation (6) starts from a null vector a solution $\underline{\hat{\alpha}}$ satifies

$$G(\underline{0}) \;=\; [H]\; \underline{\hat{\alpha}} \tag{7a}$$

giving the equivalent matrix inverse solution

$$\underline{\hat{\alpha}} \;=\; [H]^{-1}\; G(\underline{0}) \tag{7b}$$

If there is a positivity constraint on $\underline{\alpha}$ or the count noise is **Poisson** then equation (6) must be solved iteratively and SPIROS does this via the NAG subroutine **E04NCF** used to solve a classic **Quadratic Programming** problem with solution constraints.

6.4 Optimizing the χ^2 ML parameter.

Using equation (3) and assuming the count variance $\sigma^2_{d,p,e} = \nu_{d,p,e}$ the χ^2 ML parameter for any solution $\underline{\alpha}$ which results in the simulated or "expected" counts $\mu_{d,p,e}(\underline{\alpha})$ is

$$L(\underline{\alpha}) = \sum_{d,p,e} \left[\frac{[\nu_{d,p,e} - \mu_{d,p,e}(\underline{\alpha})]^2}{\nu_{d,p,e}} \right] = \chi^2(\underline{\alpha})$$

The **Gradient** vector of 1^{st} derivatives is therefore

$$\frac{\partial L}{\partial \alpha_n} = \sum_{d,p,e} \left[\frac{\nu_{d,p,e} - \mu_{d,p,e}(\underline{\alpha})}{\nu_{d,p,e}} \right] \frac{\partial \mu_{d,p,e}}{\partial \alpha_n} \tag{8}$$

or by evaluating the derivative from equation (1)

$$\frac{\partial L}{\partial \alpha_n} = -2 \sum_{d,p,e} \left[\frac{\nu_{d,p,e} - \mu_{d,p,e}(\underline{\alpha})}{\nu_{d,p,e}} \right] M_{d,p,e,n} \tag{8a}$$

or by introducing the **count residue** vector $R_{d,p,e}(\underline{\alpha})$.

$$G_n(\underline{\alpha}) = -2 \sum_{d,p,e} \frac{M_{d,p,e,n} R_{d,p,e}(\underline{\alpha})}{\nu_{d,p,e}} \tag{8b}$$

or in vector/matrix notation

$$\underline{G}(\underline{\alpha}) = -2 \, [\mathcal{M}]^T \, \underline{\mathcal{R}}(\underline{\alpha}) \tag{8c}$$

where $\mathcal{M}_{d,p,e,n} = \frac{M_{d,p,e,n}}{\sqrt{\nu_{d,p,e}}}$ and $\mathcal{R}_{d,p,e} = \frac{R_{d,p,e}}{\sqrt{\nu_{d,p,e}}}$.

The **Hessian** matrix of 2^{nd} derivatives is derived from equation (8a)

$$\frac{\partial L}{\partial \alpha_n \alpha_m} = 2 \sum_{d,p,e} \left[\frac{\partial \mu_{d,p,e}}{\partial \alpha_m} \right] \frac{M_{d,p,e,n}}{\nu_{d,p,e}} \tag{9}$$

or

$$H_{n,m} = 2 \sum_{d,p,e} \frac{M_{d,p,e,n} M_{d,p,e,m}}{\nu_{d,p,e}} \tag{9a}$$

or in matrix notation

$$[H] = 2 \, [\mathcal{M}]^T [\mathcal{M}] \tag{9b}$$

which is independent of any solution vector $\underline{\alpha}$.

6.5 Optimizing the Cash ML parameter.

The **Cash** ML parameter for any solution $\underline{\alpha}$ resulting in simulated counts $\mu_{d,p,e}(\underline{\alpha})$ is

$$L(\underline{\alpha}) \;=\; 2 \sum_{d,p,e} \Big[\mu_{d,p,e}(\underline{\alpha}) - \nu_{d,p,e} \ln[\mu_{d,p,e}(\underline{\alpha})]\Big]$$

The **Gradient** vector of 1^{st} derivatives is therefore

$$\frac{\partial L}{\partial \alpha_n} \;=\; -\,2 \sum_{d,p,e} \left[\frac{\nu_{d,p,e} - \mu_{d,p,e}(\underline{\alpha})}{\mu_{d,p,e}(\underline{\alpha})}\right] \frac{\partial \mu_{d,p,e}}{\partial \alpha_n} \qquad (10)$$

or by evaluating the derivative from equation (1)

$$\frac{\partial L}{\partial \alpha_n} \;=\; -\,2 \sum_{d,p,e} \left[\frac{\nu_{d,p,e} - \mu_{d,p,e}(\underline{\alpha})}{\mu_{d,p,e}(\underline{\alpha})}\right] M_{d,p,e,n} \qquad (10a)$$

or by introducing the **count residue** vector $R_{d,p,e}(\underline{\alpha})$

$$G_n(\underline{\alpha}) \;=\; -\,2 \sum_{d,p,e} \left[\frac{\nu_{d,p,e}}{\mu_{d,p,e}(\underline{\alpha})}\right] \frac{M_{d,p,e,n} R_{d,p,e}(\underline{\alpha})}{\nu_{d,p,e}} \qquad (10b)$$

or in vector/matrix notation using the diagonal matrix $[\mathcal{C}(\underline{\alpha})]_{d,p,e} = \nu_{d,p,e}/\mu_{d,p,e}(\underline{\alpha})$

$$\underline{G}(\underline{\alpha}) \;=\; -\,2\; [\mathcal{M}]^T\; [\mathcal{C}(\underline{\alpha})]\; \underline{\mathcal{R}}(\underline{\alpha}) \qquad (10c)$$

The **Hessian** matrix of 2^{nd} derivatives is derived from equation (10a)

$$\frac{\partial L}{\partial \alpha_n \alpha_m} \;=\; 2 \sum_{d,p,e} \left[\frac{\nu_{d,p,e}}{\mu^2_{d,p,e}}\right] \left[\frac{\partial \mu_{d,p,e}}{\partial \alpha_m}\right] M_{d,p,e,n} \qquad (11)$$

or

$$H_{n,m}(\underline{\alpha}) \;=\; 2 \sum_{d,p,e} \left[\frac{\nu_{d,p,e}}{\mu_{d,p,e}}\right]^2 \frac{M_{d,p,e,n} M_{d,p,e,m}}{\nu_{d,p,e}} \qquad (11a)$$

or in matrix notation

$$[H(\underline{\alpha})] \;=\; 2\; [\mathcal{M}]^T\; [\mathcal{C}(\underline{\alpha})]^2\; [\mathcal{M}] \qquad (11b)$$

and which is NOT independent of any solution vector $\underline{\alpha}$. Note that these equations for $G_n(\underline{\alpha})$ and $H_{n,m}(\underline{\alpha})$ are similar in form to those for **Gaussian** statistics in section 6.4 except for the "count ratio" scaling factor $\frac{\nu_{d,p,e}}{\mu_{d,p,e}}$ which multiplies $\mathcal{M}_{d,p,e,n}$ and $\mathcal{M}_{d,p,e,m}$.

SPIROS optimizes $L(\underline{\alpha})$ iteratively after getting an initial solution assuming **Gaussian** statistics then using the current source counts in $\mu_{d,p,e}$ to solve as a **QP** problem using **E04NCF** for a better solution. This continues until $L(\underline{\alpha})$ is within its stopping precision.

6.6 Optimizing a constrained ML parameter.

In section 6.3 equations (6) specified the second order Taylor expansion used to locally define how the ML parameter varies with any imaging solution vector $\underline{\alpha}$ notably in vector/matrix notation as

$$L(\underline{\alpha} + \delta\underline{\alpha}) \;=\; L(\underline{\alpha}) \;+\; G(\underline{\alpha})\;\delta\underline{\alpha} \;+\; \frac{1}{2}\delta\underline{\alpha}^t\;[H(\underline{\alpha})]\;\delta\underline{\alpha} \qquad (6c)$$

In solving this equation for large imaging pixel arrays any solution contains the side effect of **amplifying** any noise in the count data. This is well known in imaging mathematics and can be lessened by constraining the pixel heights or surface topology of the resulting image. There are simple methods for acheiving this but most result in a trade off between lessening or removing image noise with a blurring or other distortion of the image.

The most straightforward methods result in modifying the **Hessian** matrix $[\mathbf{H}(\underline{\alpha})]$ in equation (6) by the addition of a image constraint matrix $\gamma[\mathbf{C}(\underline{\alpha})]$ where γ is a constant with an optimum value which balances removal of noise with blurring of image details.

The constrained image ML parameter to be optimized then becomes

$$L(\underline{\alpha} + \delta\underline{\alpha}) \;=\; L(\underline{\alpha}) \;+\; G(\underline{\alpha})\;\delta\underline{\alpha} \;+\; \frac{1}{2}\delta\underline{\alpha}^t\;[H(\underline{\alpha}) \;+\; \gamma C(\underline{\alpha})]\;\delta\underline{\alpha} \qquad (6d)$$

The optimum value of γ is directly related to the **noise-to-signal** ratio in the image and may be estimated or given a range of values around some nominal optimum to allow the user to choose heuristically which image they prefer. It is to be noted that for display purposes an image with significant residual noise but still showing important details is sometimes preferable to one with no noise but smoothed and spread out to a blob-like form containing little of the information which is really of interest to the user.

In SPIROS $[\mathbf{C}(\underline{\alpha})]$ is referred to as the "imaging constraint matrix" and γ as the "imaging constraint multiplier" and in reading the user manual in Chapter 3 it can be seen that for the imaging of diffuse emission γ can take on a range of values to allow the user to see the effect and reduction of noise is a sequence of output images.

6.7 Calculating errors in the optimal solution.

Gaussian count noise.

For **Gaussian** count noise the equations (6) can be solved as a set of simultaneous equations using the gradient vector and Hessian matrix in equations (8) and (9) with linear errors in the solution vector $\underline{\alpha}$ given by the vector

$$\Delta\alpha \;=\; Diagonal\Big\{ \; \Big[\,[H] + \gamma[C]\,\Big]^{-1} \;\; [\mathcal{M}]^T[\mathcal{M}] \;\; \Big[\,[H] + \gamma[C]\,\Big]^{-1} \; \Big\} \tag{12}$$

or

$$\Delta\alpha \;=\; Diagonal\Big\{ \; \Big[\,[\mathcal{M}][\mathcal{M}]^T + \gamma[C]\,\Big]^{-1} \;\; [\mathcal{M}]^T[\mathcal{M}] \;\; \Big[\,[\mathcal{M}][\mathcal{M}]^T + \gamma[C]\,\Big]^{-1} \; \Big\} \tag{12a}$$

and with no imaging constraint matrix or $\gamma = 0$ simplifies to

$$\Delta\alpha \;=\; Diagonal\Big\{ \; \Big[\,[\mathcal{M}]^T[\mathcal{M}]\,\Big]^{-1} \; \Big\} \tag{12b}$$

If a **positivity** constraint is chosen equations (6) will then be solved iteratively by **NAG** subroutine **E04NCF** but the errors will currently be calculated as if linear.

Poisson count noise.

For **Poisson** count noise the equations (6) will be solved iteratively and with a **positivity** constraint using the gradient vector and solution dependent Hessian matrix in equations (10) and (11). The errors however must be calculated robustly by a procedure which for each element α_n of the solution vector $\underline{\alpha}$:

- Allows α_n to vary at several sampling points around its optimal value.
- For each of these values α_n is **fixed** and a new optimal ML parameter and solution vector found.
- Uses a cubic interpolation to calculate the deviation of α_n around its optimal value to give a change in the **Likelihood Ratio** defined in section 6.2 of just 1.0.

7 SPIROS software structure

In this section a number of flowcharts and data descriptions will be given to explain the data flow and algorithms in the most important subroutines that **SPIROS** calls.

7.1 Overview of main calling subroutines.

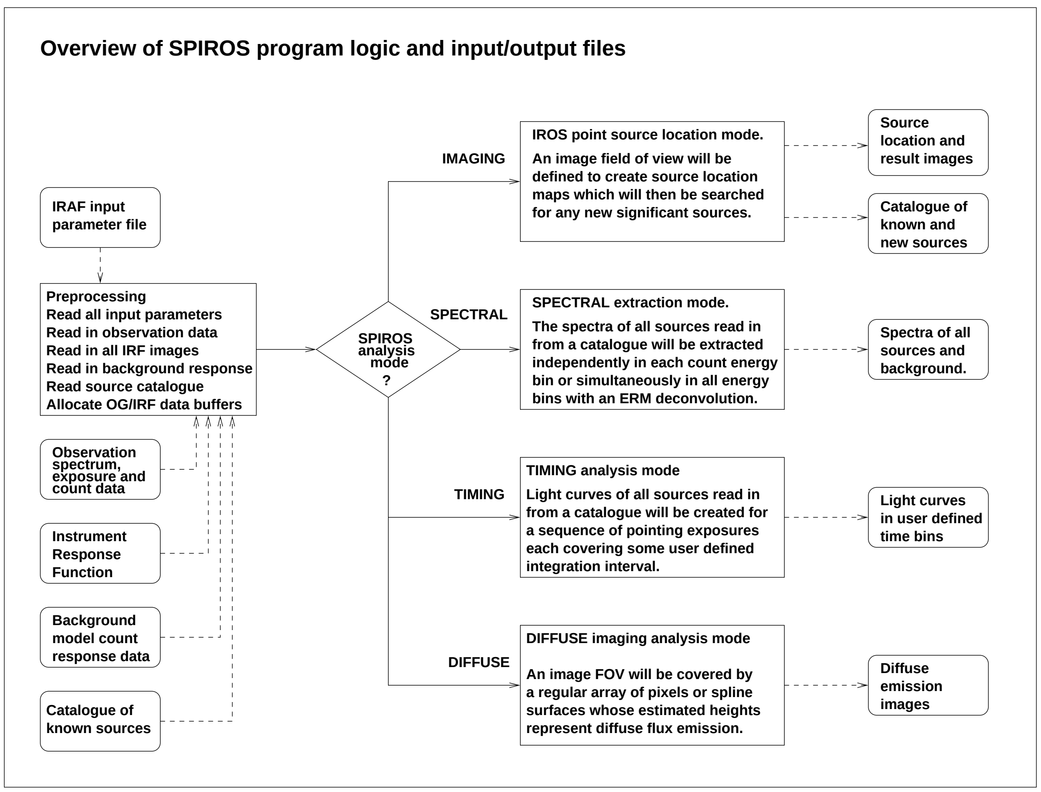
The main calling subroutine of **SPIROS** is a simple program **spiros** whose main task is to call the following three subroutines:

COMMON_INIT('spiros',versiono)
to initialize a common work area for **spiros** with the version-no in **versiono**.

spiros_main(phas)
to call up the subroutines to execute all **SPIROS** modes.

COMMON_EXIT(irec%stat)
to exit with the status returned in **irec%stat**.

The main subroutine is therefore **spiros_main** which has the tasks outlined below



7.2 Overview of main subroutine "spiros_main".

The tasks outlined above are executed in **spiros_main** as follows:

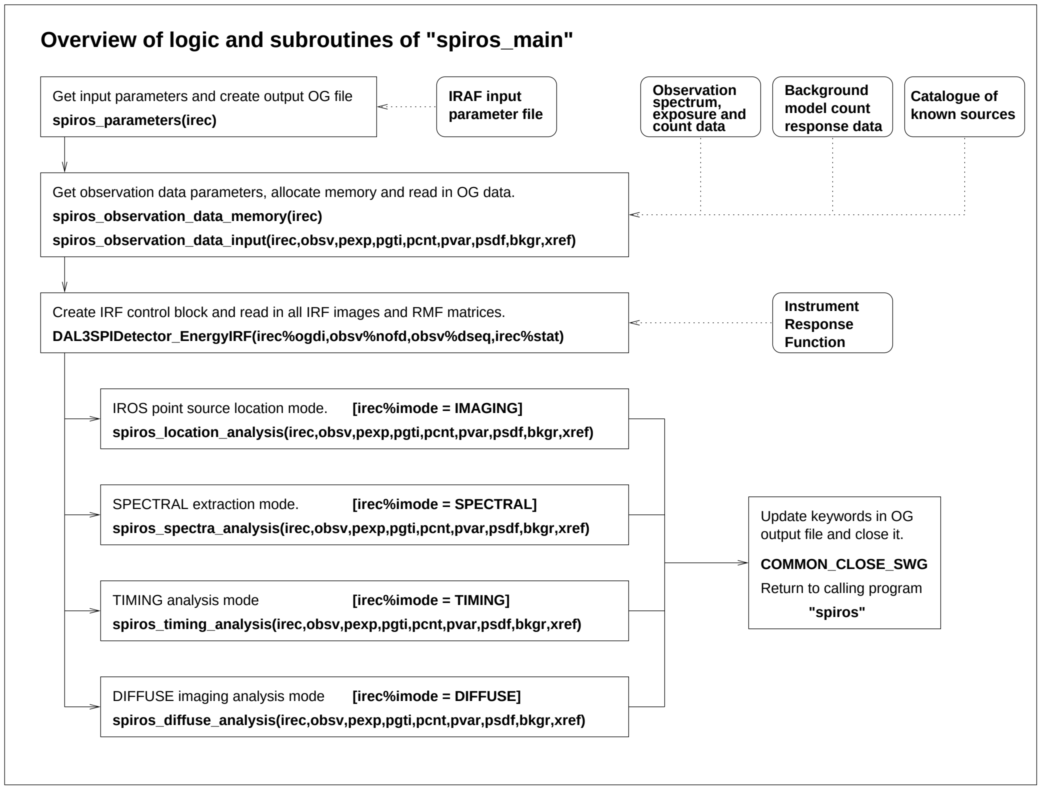


Figure 12: Summary of subroutines called in **spiros_main**

What **spiros_main** does is to read in all the parameters required to select one of the four **SPIROS** modes, read in all relevant observation data after creating buffers for each component and then pass them over to the subroutine that executes the mode selected. At this point nothing is done about creating the image, source, pixel and energy bin buffers required for the analysis mode as this will be done in the subroutine called.

The buffers described above for all input parameters and observation data are as follows with names that are kept the same in all subroutines using them:

irec	**IMAGREC** data structure for input and reconstruction parameters
obsv	**POINTING_OBS** data structure for observation exposure parameters
pexp(P)	**POINTING_EXP** array of spacecraft axis directions and pointing times
pgti(D,P)	**REAL** array of goodtime interval LIVETIMES for each detector and pointing
pcnt(D,P,E)	**REAL** array of bservation counts for each detector, pointing, energy bin
pvar(D,P,E)	**REAL** array of variance values for each count bin of **pcnt**
psdf(19,P,2,E)	**REAL** PSD efficiency and response for 19 detectors, all pointings, energy bins
bkgr(D,P,B,E)	**REAL** background response for each detector, pointing, component, energy bin
xref(D+P+E)	**INTEGER** subset cross reference for each detector, pointing, energy bin

MODULE spibham_f90_observ_parm

```
integer,parameter :: MAXNOFBKG = 8        !Maximum no of time variable background components
integer,parameter :: MAXNOFDET = 512      !Maximum no of detectors
integer,parameter :: MAXNOFCHA = 8000     !Maximum no of count binning energy channels
```

Observation parameters describing exposure count and pointing bins

type pointing_obs

```
character*256  :: file                 !Observation Group file name
integer        :: stat                 !Observation Group file status [0 = OK]
integer        :: addr                 !Observation Group file address [0 = NOT OPEN]
character*128  :: desc                 !Observation count file description
character*128  :: observer             !Observer identity
character*128  :: obsid                !Observation identifier
character*128  :: groupid              !Observation group identifier
character*32   :: datamode             !Observation data mode
character*32   :: timesys              !Time frame system []
character*32   :: timeunit             !Time units [days]
character*32   :: timeref              !Time reference frame [LOCAL etc]
character*32   :: UTCstart,UTCend      !Start/end time in UTC [YYYY.MM.DDTHH:MM:SS]
real(kind=8)   :: Mjdref               !Observation MJD reference time [51544.0 days]
real(kind=8)   :: Start,End            !Start and end time after ¡Mjdref¿ [seconds]
real(kind=8)   :: Elapsed              !Total elapsed exposure time [seconds]
real(kind=8)   :: Ontime               !Total ontime or good time interval [seconds]
real(kind=8)   :: Deadtime             !Mean dead time RATIO for all live detectors

character*16   :: RADECsys             !Julian coordinate system [FK5 etc]
character*16   :: equinox              !Coordinate system equinox [2000.0 etc]
real(kind=8)   :: RAmin,DECmin         !Minimum RA/DEC of observation [degrees]
real(kind=8)   :: RAmid,DECmid         !Mean RA/DEC of observation [degrees]
real(kind=8)   :: RAmax,DECmax         !Maximum RA/DEC of observation [degrees]
character*32   :: RAmint,DECmint       !Minimum RA/DEC of observation as text
character*32   :: RAmidt,DECmidt       !Mean RA/DEC of observation as text
character*32   :: RAmaxt,DECmaxt       !Maximum RA/DEC of observation as text
real(kind=8)   :: L2min,B2min          !Minimum L2/B2 of observation [degrees]
real(kind=8)   :: L2mid,B2mid          !Mean L2/B2 of observation [degrees]
real(kind=8)   :: L2max,B2max          !Maximum L2/B2 of observation [degrees]
character*16   :: coor                 !Observation data coordinate system [RADEC, GAL]

character*32   :: detref               !Reference for pseudo-detector definition
integer        :: nofd                 !Number of detectors (including pseudos)
integer        :: nofp                 !Number of pointing exposure intervals
integer        :: nofe                 !Number of count spectrum energy bins
integer        :: nofi                 !Number of good time intervals per exposure
integer        :: nofw                 !Number of science windows (including slews)
integer        :: nofb                 !Number of background model components
integer        :: nofv                 !Number of background values per model component
character*32   :: bnam(MAXNOFBKG)      !Names of background model components
integer        :: dseq(MAXNOFDET)      !Detector-no identity sequence in count response buffer
```

real(kind=4)	:: **emin**	!Minimum spectrum energy [keV]
real(kind=4)	:: **emax**	!Maximum spectrum energy [keV]
character	:: **esca**	!Spectrum energy scaling [L=LOG10]
real(kind=4)	:: **elow**(MAXNOFCHA)	!Spectrum energy bin lower bound [keV]
real(kind=4)	:: **eupp**(MAXNOFCHA)	!Spectrum energy bin upper bound [keV]

end type pointing_obs

Observation exposure and attitude data for **nofp** pointing sequences

type pointing_exp

real(kind=4)	:: **axis**(2,3)	!Spacecraft X,Y,Z axis directions [degrees]
real(kind=4)	:: **roll**	!Roll angle (Clockwise looking up) [degrees]
real(kind=8)	:: **Expstart**	!Exposure start time after ¡Mjdref¿ [days]
real(kind=8)	:: **Expend**	!Exposure end time after ¡Mjdref¿ [days]
real(kind=4)	:: **exptime**	!Exposure time interval [seconds]

end type pointing_exp

7.3 SPIROS input parameter data structure.

This structure contains all input parameters plus some for logfile messages and results of image or spectral reconstruction which are usually passed to all SPIROS subroutines as a common block of control data with the name **"irec"**. [IN] refers to parameters which may be required as input to image reconstruction analysis while [OUT] refers to parameters returned as output during the analysis.

MODULE spibham_f90_imaging_parm

type imagrec

Program control parameters

character*64	:: root	!Root program name - INITIALIZEME
character*32	:: vers	!Root program version-no - INITIALIZEME
character*8	:: runo	!NOT USED
character*128	:: logf	!Program logfile - INITIALIZEME
integer	:: logu	!Program logfile unit-no - INITIALIZE to zero
character*512	:: mess	!Message output buffer
character*256	:: prot	!Additional message buffer - use as needed
integer	:: stat	!Subroutine status code
character*32	:: task	!Subroutine task buffer
integer	:: disp	!Message display channel - 6 = display messages on screen also
integer	:: OK	!Subroutine code for OK - INITIALIZE to zero or ISDC_OK
integer	:: EOF	!NOT USED
integer	:: ALERT	!NOT USED
integer	:: ABORT	!NOT USED

Parameters common to all imaging executables

integer	:: ogdi	![IN] Address of output OG dataset if opened [0 = NOT OPENED]
character*16	:: coor	![IN] **[reference-coord]** Image coordinate system [RADEC, GALACTIC]
character*16	:: proj	![IN] **[image-proj]** Image projection type [CAR,TAN,AIT]
character*16	:: tofv	![IN] **[image-fov]** Image FOV type [USER,POINTING,POINTING-C/F/Z]
real(kind=4)	:: icen(2)	![IN] **[center-long , center-lat]** Image centrepoint [degrees]
integer	:: idim(2)	![IN] **[image-dim-long,image-dim-lat]** Image array dimensions
real(kind=4)	:: idel(2)	![IN] **[image-pixel-lon,image-pixel-lat]** Image pixel width [degrees]
real(kind=4)	:: ihwd(2)	![IN] **0.5*idim(n)*idel(n)** Image halfwidth FOV
character*16	:: iori	![IN] **[image-orient]** Image orientation [USER, STANDARD]
real(kind=4)	:: npol(2)	![IN] **[pole-long,pole-lat]** Image north pole coordinates [degrees]
character*16	:: phun	![IN] **ph/cm2/sec** Image pixel height units [ph/cm2/sec etc]
character*16	:: pwun	![IN] **degrees** Image pixel width units [degrees etc]
character*16	:: maxlifun	![IN] **[optistat]** Optimization statistic type [CHI2, LIKELIHOOD]
character*16	:: solutype	![IN] **[solution-constr]** Matrix equation solution type [POS,NEG,LIN]
character	:: restart	![IN] **C** Restart iterative solution [Y/N/C default is C=Cold]
character	:: errobust	![IN] **Y** Calculate robust errors [Y/N default is Y]
character	:: iterout	![IN] **[iteration-output]** Output images after each iteration [Y/N]
character	:: srcreloc	![IN] **[source-relocation]** Relocate sources using entire spectrum [Y/N]
character	:: erspmat	![IN] **[energy-response]** Include energy response matrix [Y/N]
character	:: loghess	![IN] **N** Log progress of Hessian matrix construction
integer	:: verbose	![IN] **[mode]** Log verbose details [N in mode parameter XXXX-N]
integer	:: backhand	![IN] **[background-method]** Background handling [1,2,3,4]
character*16	:: locbins	![IN] **[srclocbins]** Source location bins [ALL,SUM,a-b,a-bkeV]

Parameters for source location and diffuse imaging

character*16	:: imode	![IN] **[mode]** Analysis mode [IMAGING,SPECTRA,TIMING,DIFFUSE]
integer	:: nofi	![IN] **[nofsources]** No of source search iterations
character*16	:: kofs	![IN] **[kofsources]** Kind of sources to search for [POINT,POINTLIKE]
real(kind=4)	:: scanstep	![IN] **[searchstep]** Source correlation scanning step [0.0 = ¡idel¿ above]
real(kind=4)	:: sigmathr	![IN] **[sigmathres]** Lower sigma threshold level to stop
real(kind=4)	:: mliprec	![IN] **[maxlikprec]** ML parameter precision [Default is 0.1]
real(kind=4)	:: locprec	![IN] **[srclocprec]** Source location precision [degrees - default is 0.1]
real(kind=4)	:: locstep	![IN] **[chilocstep]** Source location sampling step [degrees - default is 0.1]
real(kind=4)	:: widprec	![IN] **[srcwidprec]** Source width precision [degrees - default is 0.1]
real(kind=4)	:: widstep	![IN] **[chiwidstep]** Source width sampling step [degrees - default is 0.5]
real(kind=4)	:: locerr	![IN] **[location-max-error]** Maximum admissable source error[degrees]
real(kind=4)	:: psfwhm	![IN] **3.0** Point spread function FWHM [degrees - FWHM of central peak]
real(kind=4)	:: psblurr	![IN] **[blur-size]** Point source blurring FWHM [degrees - blurrs point sources]
character*16	:: pfun	![IN] **[pixel-func]** Image pixel function [POINT,HAT,LINEAR,BSPLINE]
real(kind=4)	:: psiz	![IN] **[pixel-size]** Image pixel FWHM/width [degrees]
real(kind=4)	:: psam	![IN] **0.5** Image pixel sampling step [degrees - default is 0.5]
character*16	:: imco	![IN] **[constrtype]** Image constraint matrix [WIENER,DIAGONAL,RIPPLE]
real(kind=4)	:: imcoparm	![IN] **NOT USED** Image constraint matrix parameter[FWHM of smoothing function etc]
integer	:: lmdanofi	![IN] **[constriter]** Image constraint multiplier iterations wanted
real(kind=4)	:: lmdaincr	![IN] **[constrincr]** Image constraint multiplier increment for optimum search
real(kind=4)	:: lmdacrit	![OUT] Image constraint multiplier critical or Wiener optimum value
real(kind=4)	:: lmda	![IN] **[constrmult]** Image constraint multiplier of critical value
integer	:: lmdaiter	![IN] **0** Image constraint multiplier iteration counter
character*512	:: nagf	![IN] **[nagoptions]** Name of NAG options file [Default is "spiros.nag"]
integer	:: nofebins	![IN] **0** No of continuum energy bins [0 = default to COUNT binning]
character*16	:: ebinfunc	![IN] **COUNT** Energy bin function [COUNT,POINT,HAT,LIN,BSPLINE,POW]
real(kind=4)	:: ebinparm	![IN] **NOT USED** Energy bin parameter [Default is 0.0 otherwise power law index]
character	:: ebinscal	![IN] **NOT USED** Energy bin scaling [L = LOG10 otherwise LINEAR]
real(kind=4)	:: ebinsamp	![IN] **NOT USED** Energy bin sampling step [Default is 0.1 for LOG10, 10.0 for LINEAR]
character	:: ebinmaps	![IN] **NOT USED** Energy bin image output [YES/NO or output filename]
integer	:: noflines	![IN] **NOT USED** No of lines to search for
character*16	:: koflines	![IN] **NOT USED** Shape of ine features [POINT,HAT,LINEAR,BSPLINE]
real(kind=4)	:: linelocp	![IN] **NOT USED** Line location precision [keV - default is 0.1]
real(kind=4)	:: linelocs	![IN] **NOT USED** Line location sampling step [keV - default is 1.0]
real(kind=4)	:: linewidp	![IN] **NOT USED** Line width precision [keV - default is 0.1]
real(kind=4)	:: linewids	![IN] **NOT USED** Line width sampling step [keV - default is 1.0]

Input/output file names and templates

```
character*512 :: obsvinpf   ![IN] [in-og-dol] Input observation group file
character*512 :: obsvptgf   ![IN] [pointing-dol] Input observation pointing file
character*512 :: obsvgtif   ![IN] [gti-dol] Input observation goodtime interval file
character*512 :: obsvdtif   ![IN] [deadtime-dol] Input observation deadtime interval file
character*512 :: obsvbinf   ![IN] [ebounds-dol] Input observation energy binning file
character*512 :: obsvcntf   ![IN] [evts-det-spec-dol] Input observation count spectra
character*512 :: obsvpsde   ![IN] [psd-efficiency-dol] Input observation PSD efficiency file
character*512 :: obsvpsdr   ![IN] [psd-response-dol] Input observation PSD response file
character*512 :: instrspf   ![IN] [inst-resp-idx] Input instrument response file
character*512 :: backmodf   ![IN] [back-model-idx] Input background model file
character*512 :: scatlogf   ![IN] [source-cat-dol] Input source catalogue file
character*512 :: obsvoutf   ![IN] [out-og-dol] Output observation group datasets

character*512 :: maxlfile   ![IN] [maxlikfile] Output ML parameter residue file [YES/NO or output filename]
character*32  :: maxltemp   ![IN] SPI.-MAXL-RES.tpl Output ML parameter residue file template
character*512 :: scatfile   ![IN] [source-res] Output source catalogue file [YES/NO or output filename]
character*32  :: scattemp   ![IN] SPI.-SRCL-RES.tpl Output source catalogue template
character*512 :: iidxfile   ![IN] [image-idx] Output image index file [YES/NO or output filename]
character*32  :: iidxtemp   ![IN] SPI.-SKY.-IMA-IDX.tpl Output image index file template
character*32  :: idattemp   ![IN] SPI.-SKY.-IMA.tpl Output image data file template

character*512 :: fmap       ![IN] [image-int] Output flux intensity map [YES/NO or output filename]
character*512 :: smap       ![IN] [image-sig] Output flux sigma map [YES/NO or output filename]
character*512 :: emap       ![IN] [image-err] Output flux error map [YES/NO or output filename]
character*512 :: omap       ![IN] NOT USED Output observation FOV map [YES/NO or output filename]
character*512 :: lmap       ![IN] NOT USED Output ML parameter map [YES/NO or output filename]

character*512 :: bidxfile   ![IN] [back-det-spec-idx] Output background count spectra index file [YES/NO or output filename]
character*32  :: bidxtemp   ![IN] SPI.-BACK-DSP-IDX.tpl Output background count spectra index template name
character*512 :: bgrp       ![IN] [back-det-spec] Output background count spectra [YES/NO or output filename]
character*32  :: bgrptemp   ![IN] SPI.-BACK-DSP.tpl Output background count spectra file template name

character*512 :: cidxfile   ![IN] [source-det-spec-idx] Output source count spectra index file [YES/NO or output filename]
character*32  :: cidxtemp   ![IN] SPI.-SDET-SPE-IDX.tpl Output source count spectra index template
character*512 :: cgrp       ![IN] [source-det-spec] Output source count spectra [YES/NO or output filename]
character*32  :: cgrptemp   ![IN] SPI.-SDET-SPE.tpl Output source count spectra file template

character*512 :: sidxfile   ![IN] [source-spec-idx] Output source flux spectra index file [YES/NO or output filename]
character*32  :: sidxtemp   ![IN] SPI.-PHA1-SPE-IDX.tpl Output source flux spectra index template
character*512 :: sgrp       ![IN] [source-spec] Output source flux spectra [YES/NO or output filename]
character*32  :: sgrptemp   ![IN] SPI.-PHA1-SPE.tpl Output source flux spectra file template

character*512 :: tidxfile   ![IN] [source-timing-idx] Output transient data index filename [YES/NO or output filename]
character*32  :: tidxtemp   ![IN] SPI.-SRC.-LCR-IDX.tpl Output transient data index template
character*512 :: tranfile   ![IN] [source-timing] Output transient data filename [YES/NO or output filename]
character*32  :: trantemp   ![IN] SPI.-SRC.-LCR.tpl Output transient data filename template
character*16  :: timode     ![IN] [source-timing-mode] Timing mode or method [QUICKLOOK, WINDOW or SIMULTAN]
real(kind=4)  :: timescal   ![IN] [source-timing-scale] No of pointing exposures per transient exposure
```

Reconstruction control and output parameters

character*8	:: imaging	![OUT] Type of imaging [COMPACT, ARRAYS, DIFFUSE]
integer	:: nofixed	![OUT] Total no of fixed sources (detectors and pixel arrays)
integer	:: nofcats	![OUT] Total no of catalogue sources
integer	:: nofknown	![OUT] Total no of known sources (detectors, pixel arrays and catalogued)
integer	:: nofs	![OUT] Total no of known and new sources
real(kind=4)	:: adel(2)	![OUT] Spline node array lattice size [degrees]
integer	:: adim(2)	![OUT] Spline node array dimensions [Calculated with ¡fdel¿ to cover image FOV]
integer	:: rank	![OUT] Current rank of matrices in CHI2/CASH/etc optimization function.
real(kind=8)	:: Chi0	![OUT] Image reconstruction ML parameter at Taylor expansion point
real(kind=8)	:: Chi2	![OUT] Image reconstruction ML parameter
real(kind=8)	:: Chim	![OUT] Image reconstruction modified ML parameter
real(kind=8)	:: Chid	![OUT] Image reconstruction ML parameter difference from optimum
real(kind=8)	:: Lmdacurr	![OUT] Image reconstruction current constraint matrix multiplier
real(kind=8)	:: Cmin	![OUT] Minimum no of counts per detector for ML cutoff [1.0E-12]
real(kind=4)	:: fluxminm	![OUT] Minimum flux for ML cutoff [1.0E-12]
integer	:: bino	![OUT] Current energy bin being processed
integer	:: iter	![OUT] Current ML solution iteration-no
integer	:: itermaxm	![OUT] Maximum number of ML solution iterations
integer	:: maxnofs	![OUT] Maximum no of sources
integer	:: maxnofp	![OUT] Maximum no of source pixels
integer	:: maxnofe	![OUT] Maximum no of source spectra energy bins
integer	:: maxnofb	![OUT] Maximum no of background model components
integer	:: maxnofi	![OUT] Maximum no of output image pixels
integer	:: imagtime	![OUT] Image creation time in seconds
integer	:: exestime	![OUT] Estimated total execution time in seconds
integer	:: exectime	![OUT] Actual total execution time in seconds
integer	:: byts	![OUT] Total no of bytes allocated to memory for internal buffers
integer	:: NAGstat	![OUT] Status returned from any NAG subroutines [0 = OK]
character*256	:: esubset	![IN] **energy-subset** ”a-b,c-d,e,f” subset of energy bins to be selected
character*256	:: psubset	![IN] **pointing-subset** ”a-b,c-d,e,f” subset of pointing bins to be selected
character*256	:: dsubset	![IN] **detector-subset** ”a-b,c-d,e,f” subset of detector bins to be selected

end type
END MODULE spibham_f90_imaging_parm

7.4 Source, pixel, energy bin system "srce,spix,sbin".

Most subroutines called for each analysis mode require a description of each source, their pixels and spectrum energy bins and for this arrays of three specific data structures are used to store their particular parameters. Each **source** is either detector background, a point source or pixel image group and is described by the data structure **source** which is usually implemented in SPIROS as an array named "**srce**".

MODULE **spibham_f90_source_parm**

```
type source
    character*32    :: iden       !Source identifier in catalogue
    character*32    :: name       !Source name

    character       :: type       !Source type
                                  !D = Detector background (may have multiple pixels)
                                  !S = Single sky source (may have multiple pixels)

    character       :: grup       !Source pixel grouping flag
                                  !F = Pixel heights represent a spatial function
                                  !I = Pixel heights are independent or an image array

    character       :: loca       !Source location and width flag
                                  !F = Source has a FIXED location and width
                                  !V = Source has a VARIABLE location and width

    character       :: flar       !Source flaring above expected behaviour [Y/N]

    integer         :: nofp       !No of source pixels
    integer         :: pptr       !Pointer to first source pixel [0,1...]
    real(kind=4)    :: psam       !Pixel response sampling step [degrees]
    character*8     :: pfun       !Pixel function [POINT,HAT,LINEAR,BSPLINE]

    integer         :: nofe       !No of source spectrum energy bins
    integer         :: eptr       !Pointer to first spectrum energy bin [0,1...]
    real(kind=4)    :: esam       !Energy bin sampling step [keV or LOG10(keV)]
    real(kind=4)    :: erng(2)    !Spectrum energy range [keV or LOG10(keV)]
    character       :: esca       !Spectrum energy scale [L = LOG10(keV) otherwise linear]
    character*8     :: efun       !Spectrum energy bin [POINT,HAT,LINEAR,BSPLINE]

    integer         :: comp       !Time variation component-no if response is stored
    integer         :: noft       !No of time function parameters - NOT USED
    real(kind=8)    :: Tpar(4)    !Time variation function parameters - NOT USED
    character*64    :: tfun       !Time variation function description - NOT USED

    real(kind=4)    :: posn(2)    !Source centrepoint longitude/latitude [degrees]
    real(kind=4)    :: erra       !Source position error radius [degrees]
end type
```

Each **pixel** of a detector background or sky source is described by the following data structure **source_pixel** and is usually an array named "**spix**".

MODULE **spibham_f90_pixel_parm**

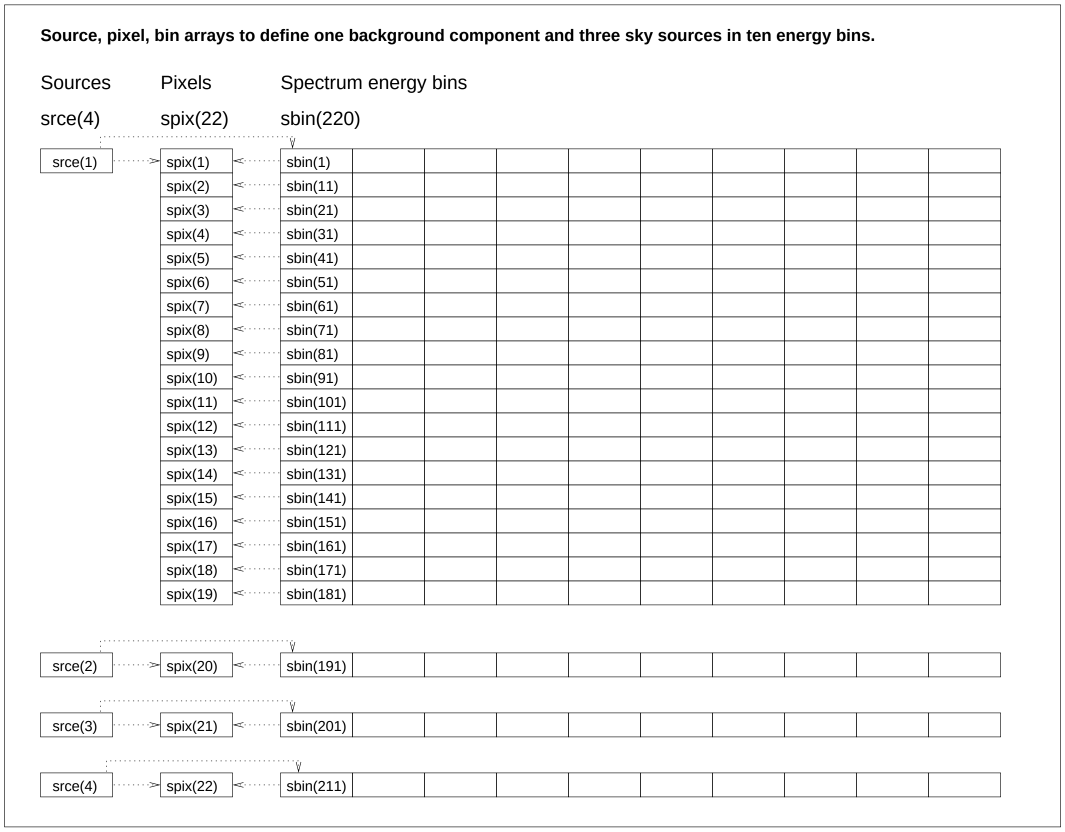
```
type source_pixel
    integer       :: srcn      !Source-no containing pixel
    integer       :: pino      !Source pixel-no or detector-no for background)
    real(kind=4)  :: hght      !Pixel function height
    real(kind=4)  :: posx(2)   !Pixel centrepoint longitude/error [degrees]
    real(kind=4)  :: posy(2)   !Pixel centrepoint latitude/error [degrees]
    real(kind=4)  :: wdth(2)   !Pixel function width/error [degrees]
    character     :: type      !Pixel type: NOT USED ANYMORE
                               !F = Fixed location and width
                               !V = Variable location and width allowed
end type
```

Each spectrum **energy bin** of a detector background or sky source is described by the following data structure **energy_bin** and is usually an array named "**sbin**".

MODULE **spibham_f90_ebin_parm**

```
type energy_bin
    integer       :: srcn      !Source-no of energy bin
    integer       :: pixn      !Energy bin pixel-no [0 = all pixels]
    real(kind=4)  :: ergy      !Energy bin function centrepoint [keV/LOG10(keV)]
    real(kind=4)  :: hght(2)   !Energy bin height/error
    real(kind=4)  :: wdth(2)   !Energy bin width/error [keV/LOG10(keV)]
    character     :: type      !Energy bin type: NOT USED YET
                               !L = Line or compact feature
                               !B = Boundary node of continuum spectrum
                               !otherwise internal node of a continuum spectrum
    integer       :: pntg(2)   !Pointing exposure range for which energy bin is "visible"
    character     :: mark      !Marker for energy bin having been changed [y/n]
    integer       :: stat      !ISDC solution status or error code
    integer       :: NAGstat   !Status or error code returned from NAG routines
end type
```

A typical example of the use of these data structures is in reconstructing one detector background component, which may have a different flux in each of 19 detectors, and three catalogue sources, all with a spectrum of 10 energy bins. In this case the first source is for a CONSTANT detector background component and has 19 detector "pixels" each having a spectrum with 10 energy bins. The second to fourth sources are sky sources each with only one point source pixel and 10 energy bins. In this case the "srce,spix,sbin" arrays have the following configuration with the contents of the source description array **srce(4)** given.



Parameter	srce(1)	srce(2)	srce(3)	srce(4)
iden	BACKGROUND	J195821.7+351206	J203225.8+405728	J205707.2+413112
name	CONSTANT	Cygnus X-1	Cygnus X-3	GRO J2058+42
type	D	S	S	S
grup	I	I	I	I
loca	F	V	V	F
flar	N	N	N	N
nofp	19	1	1	1
pptr	0	19	20	21
psam	0.0	0.0	0.0	0.0
pfun	POINT	POINT	POINT	POINT
nofe	190	10	10	10
eptr	0	190	200	210
esam	0.0	0.0	0.0	0.0
erng(2)	40,200	40,200	40,200	40,200
esca				
efun	HAT	HAT	HAT	HAT
comp	1	0	0	0
noft	0	0	0	0
Tpar(4)	NOT USED	NOT USED	NOT USED	NOT USED
tfun	NOT USED	NOT USED	NOT USED	NOT USED
posn(2)	0.0,0.0	299.59,35.20	308.11,40.96	314.28,41.52
erra	0.0	0.01	0.01	0.01

The contents of the source pixels in array **spix(22)** are as follows:

Parameter	spix(1)-spix(19)	spix(20)	spix(21)	spix(22)
srcn	1	2	3	4
pino	1-19	1	1	1
hght	1.0	1.0	1.0	1.0
posx(2)	0.0,0.0	299.59,0.01	308.11,0.06	314.28,0.15
posy(2)	0.0,0.0	35.20,0.01	40.96,0.05	41.52,0.13
wdth(2)	0.0,0.0	0.0,0.0	0.0,0.0	0.0,0.0
type	D	V	V	F

The contents of energy bins $\mathbf{3, 191, 202, 215}$ in array **sbin(220)** for spectra covering $\mathbf{40 - 200}$ **keV** in 10 **16.0 keV** wide bins and observed on a **9x9** grid of **81** pointing exposures are as follows:

Parameter	sbin(3)	sbin(191)	sbin(202)	sbin(215)
srcn	1	2	3	4
pixn	3	1	1	1
ergy	40.0	8.0	24.0	72.0
hght(2)	0.0,0.0	0.0,0.0	0.0,0.0	0.0,0.0
wdth(2)	16.0,0.0	16.0,0.0	16.0,0.0	16.0,0.0
type	D	V	V	F
pntg(2)	1,81	1,81	1,81	1,81
mark	n	n	n	y
stat	0	0	0	-210106
NAGstat	0	0	0	1

with an error $-\mathbf{210106}$ in the **5th** energy bin of source-no **5** indicating a "weak solution" returned with status **1** from NAG subroutine E04NCF, used to find an optimal positive or ML solution.

Some important parameters in each **source** data structure above are a "pixel pointer" **pptr** to the **nofp** pixels a source has in the pixel array **spix** and an "energy bin pointer" **eptr** to the **nofe** energy bins a source has in array **sbin**. These are used to navigate through any **srce,spix,sbin** configuration to obtain the pixel and spectrum information of each source. In F90 source-no **s** has **srce**(**s**)%**nofp** pixels and the parameters of its **nth** pixel are found in **spix**(**srce**(**s**)%**pptr** + **n**). Similarly it has **srce**(**s**)%**nofe** energy bins and the parameters of its **nth** energy bin are found in **sbin**(**srce**(**s**)%**eptr** + **n**).

From the diagram above and the data structures described it can be seen that each pixel knows the source to which it belongs and each energy bin knows both the source and pixel to which it belongs.

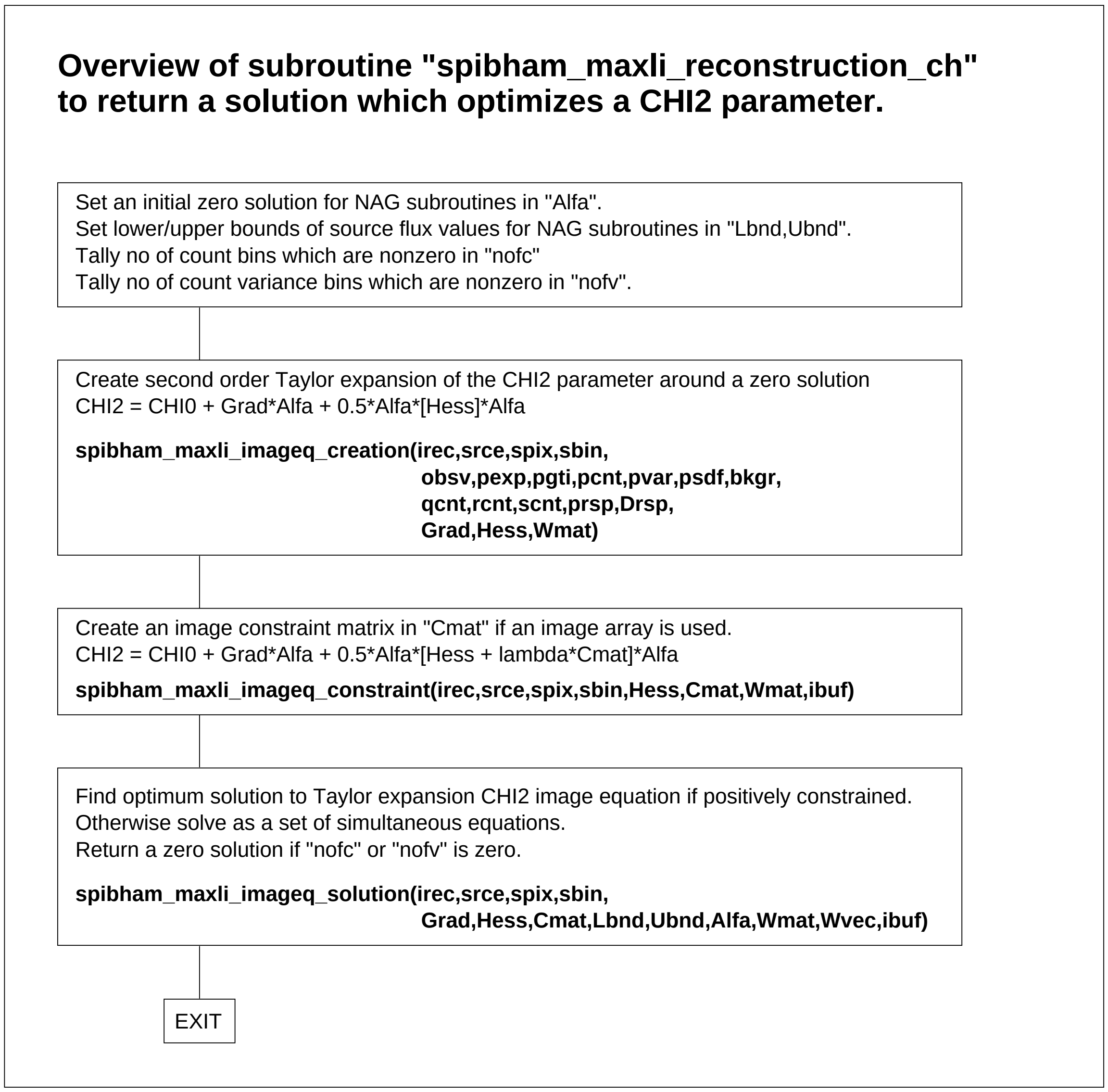
Note also that each energy bin contains a **pntg(2)** pair of pointing numbers for which the bin is "visible" during an observation. This pair normally covers all pointing exposures used but in **TIMING** analysis mode it will contain subsets of pointings to calculate flux values in sequences covering some integration interval to look for any flaring behaviour over the observation period. In this sense the energy bins are actually energy-time bins.

7.5 Solving for source flux values with spibham_maxli_reconstruction

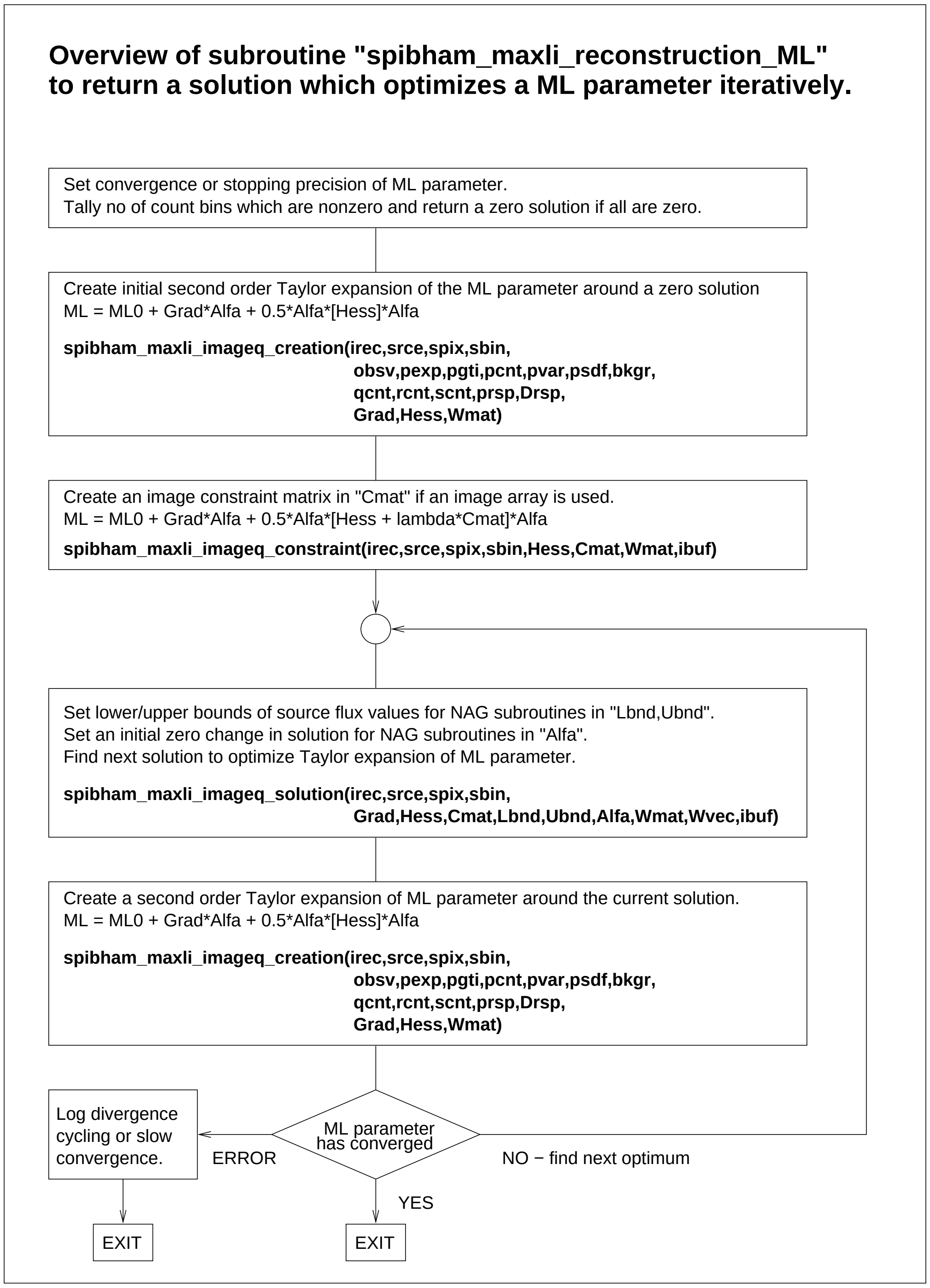
This **srce,spix,sbin** configuration system is used along with the input observation data **obsv,pexp,pgti,pcnt,pvar,psdf,bkgr** as input to a general reconstruction subroutine called **spibham_maxli_reconstruction** which will return an optimal solution for the flux in all source spectrum bins in **sbin** and found as a flux and its error in the field **sbin(n)%hght**.

The main point to understand in using **SPIROS** is to set up the **srce,spix,sbin** configuration for the background components and sky sources selected for analysis and call the reconstruction subroutine to estimate the solution which gives an optimal fit to the observation data in **pcnt**. In general the main data passed to the subroutines used by **SPIROS** is the group **irec,srce,spix,sbin,obsv,pexp,pgti,pcnt,pvar,psdf,bkgr**.

The subroutine **spibham_maxli_reconstruction** which is used to calculate all detector background and any number of source spectra simultaneously contains two options to return a solution by optimizing either a χ^2 or **Maximum Likelihood** parameter. The first option calls **spibham_maxli_reconstruction_CH** which has the structure:

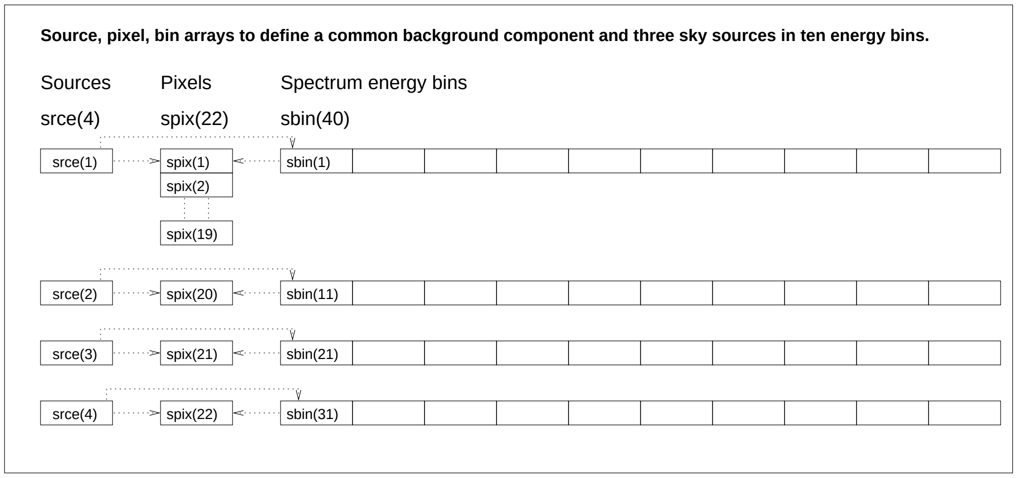


The second option calls the subroutine **spibham_maxli_reconstruction_ML** to calculate all detector background and any number of source spectra simultaneously by optimizing a **Maximum Likelihood** parameter and has the structure:

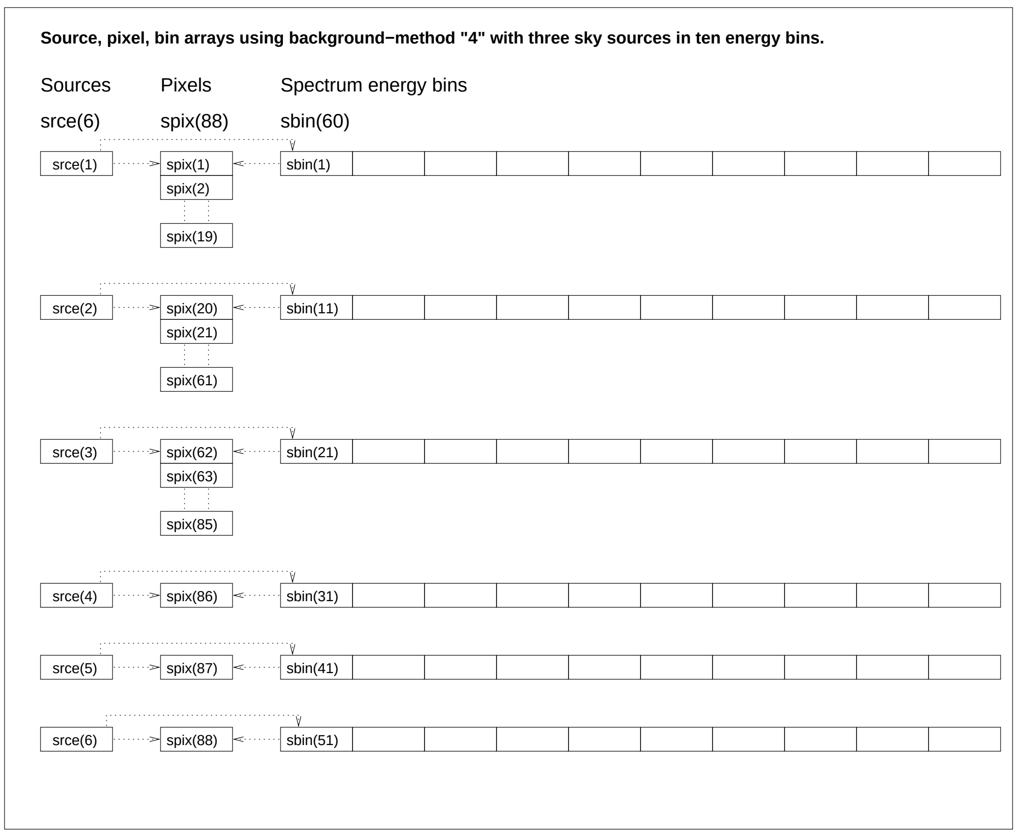


7.6 Configuration for a common relative background spectrum.

In the configuration example above background rates are calculated in all detectors independently and this will be the result of setting the input parameter **background-method** to **2**. If this parameter was set to **3** then it would be assumed the background rates were given in **bkgr** as relative values over the detector plane and only one spectra for all detectors would be returned. This would have a **srce,spix,sbin** configuration as follows:

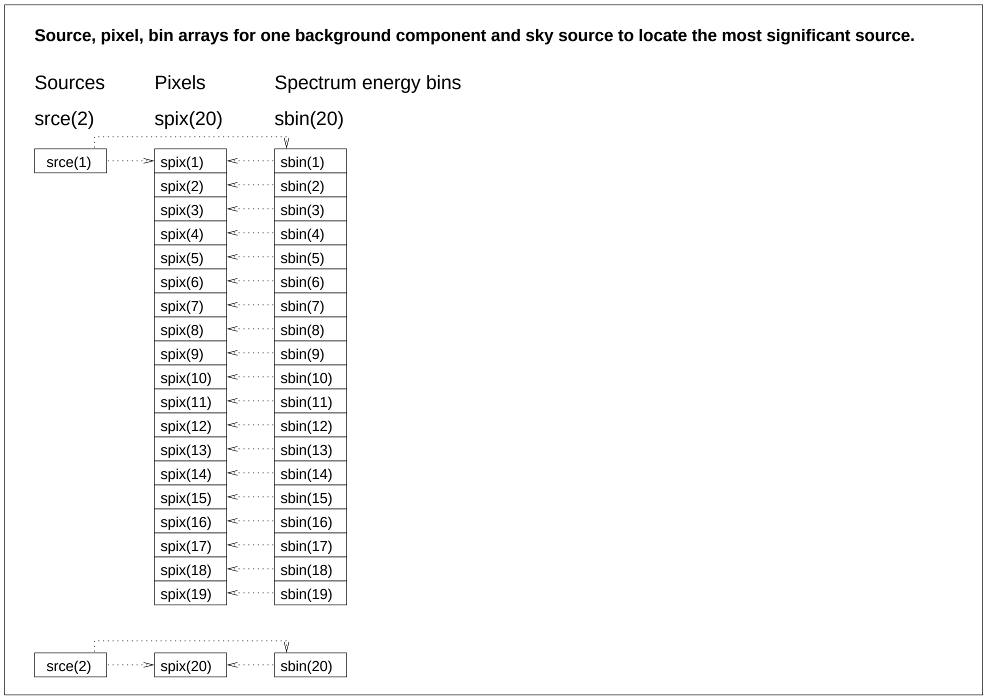


If **background-method** would be set to **4** then background rates would be assumed to be relative values in the detector groups [0-18] [19-60] and [61-84] resulting in the following configuration:



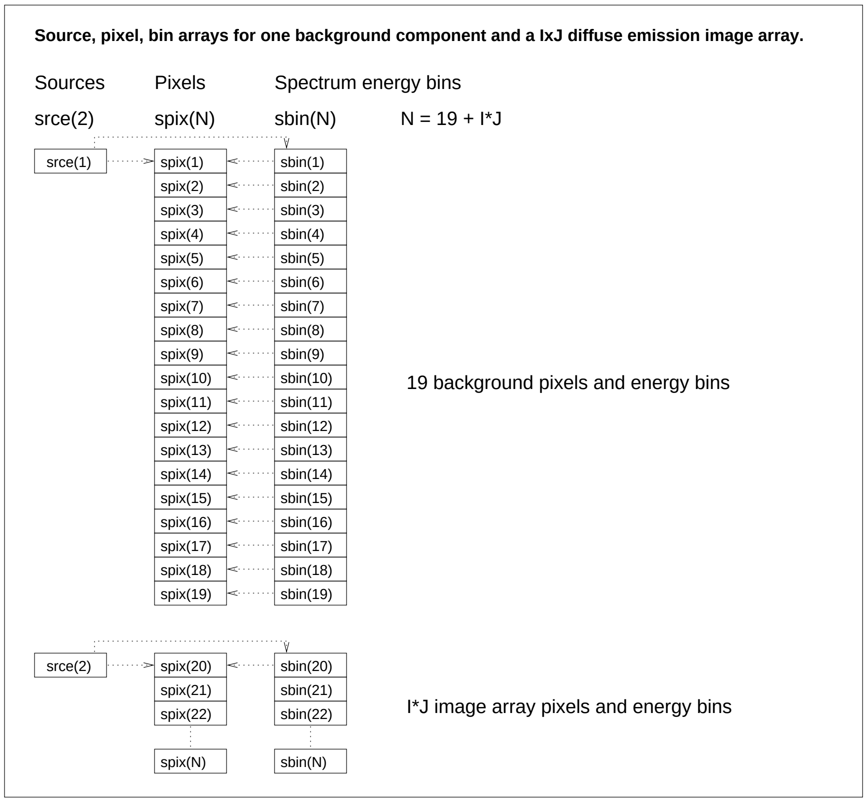
7.7 Configuration to scan for point sources.

When locating sources in **IMAGING** mode SPIROS will set up sources, pixels and energy bins for the background components then append a single point source whose location in **posx,posy** may be varied over the observation field of view to find the location which returns the most optimal and significant fit to the observation data. For this the **srce,spix,sbin** configuration simplifies to:



7.8 Configuration for a diffuse emission image array.

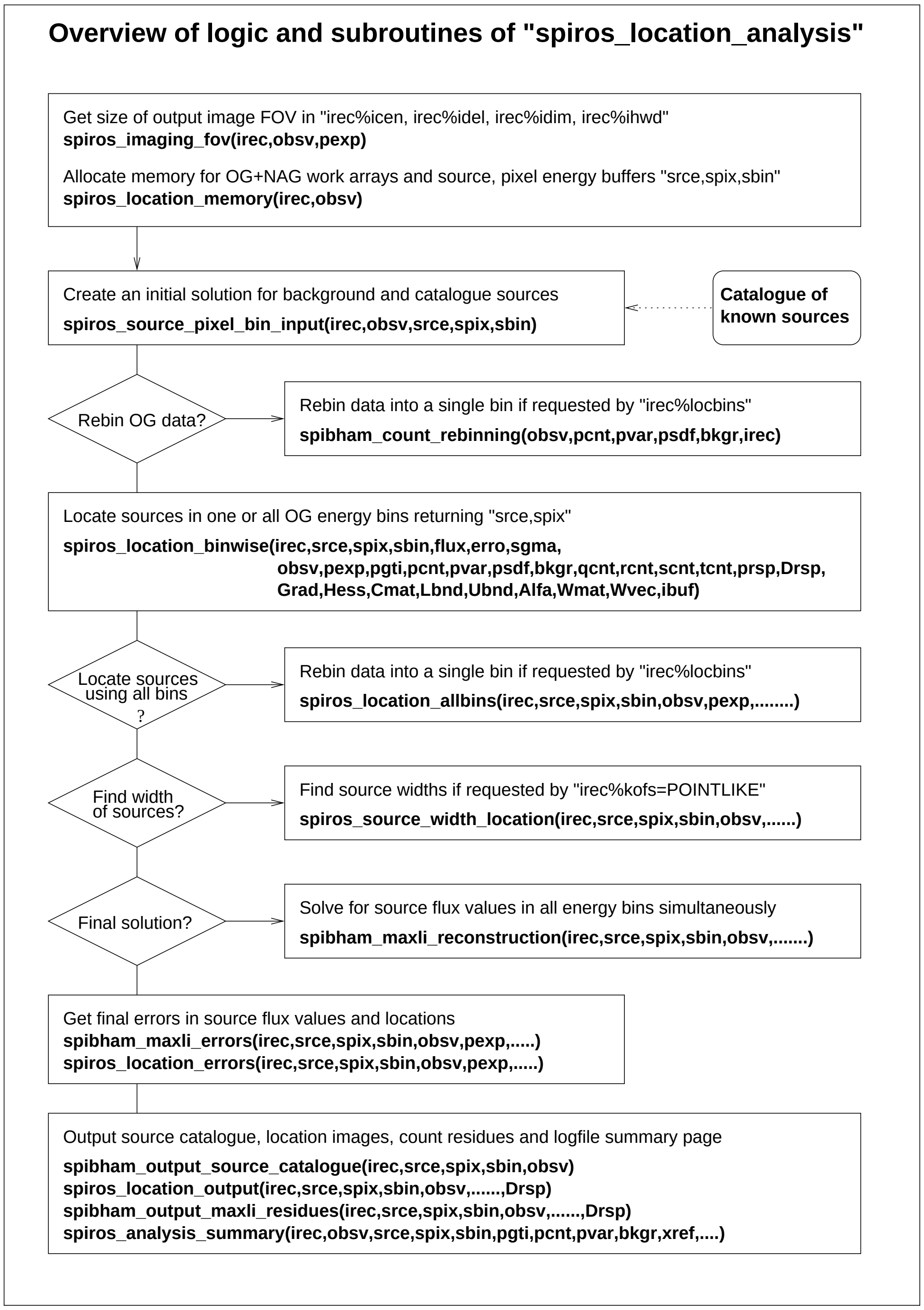
In **DIFFUSE** imaging mode a diffuse emission image array is reconstructed in each count energy bin with a typical **srce,spix,sbin** configuration as follows:



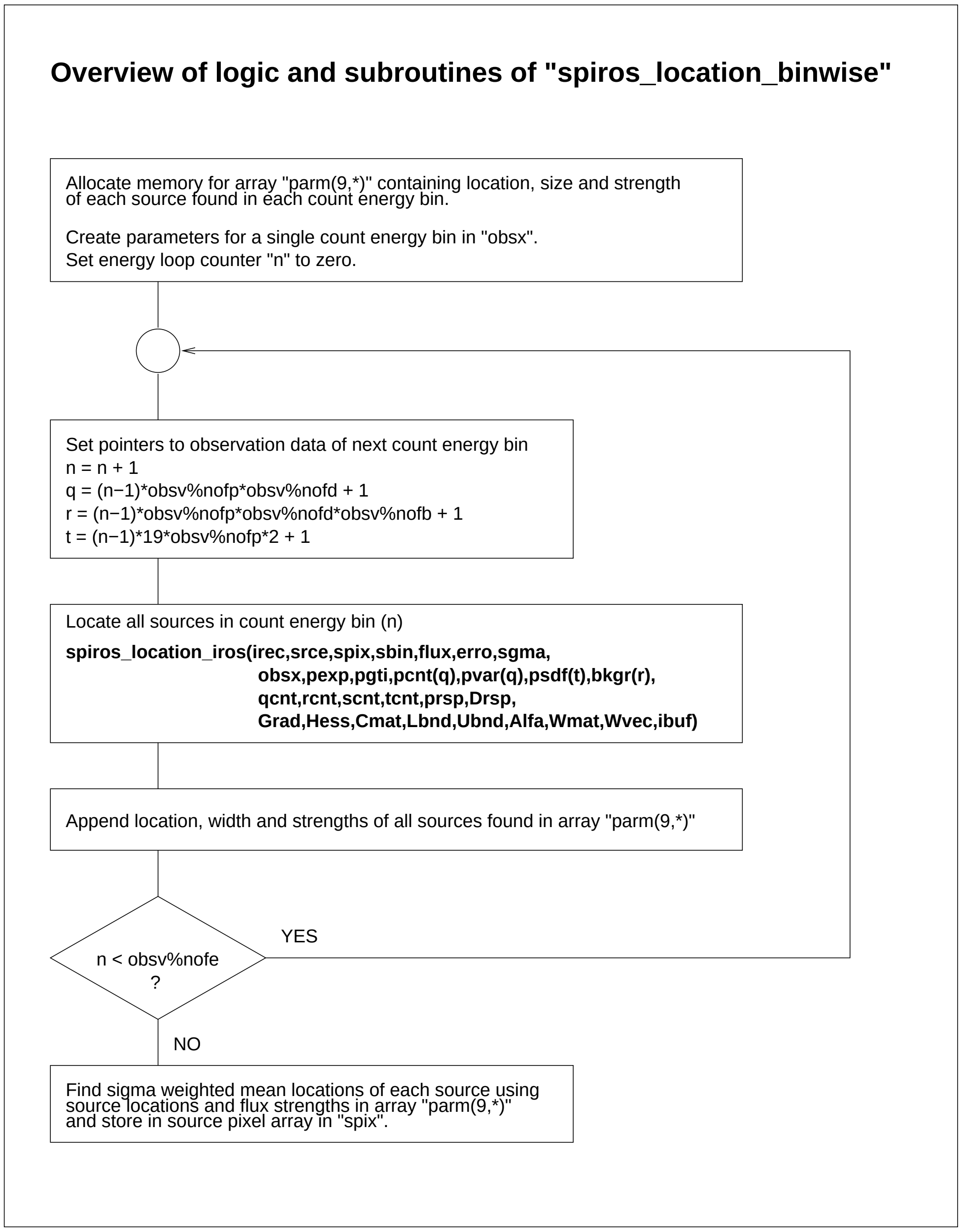
In this case the source is a rectangular image array but it could also be a group of adjacent pixels with an irregular boundary, or a multiplicity of such groups if several locally diffuse sources are observed whose spatial extent is already known.

7.9 Source location software for IMAGING mode.

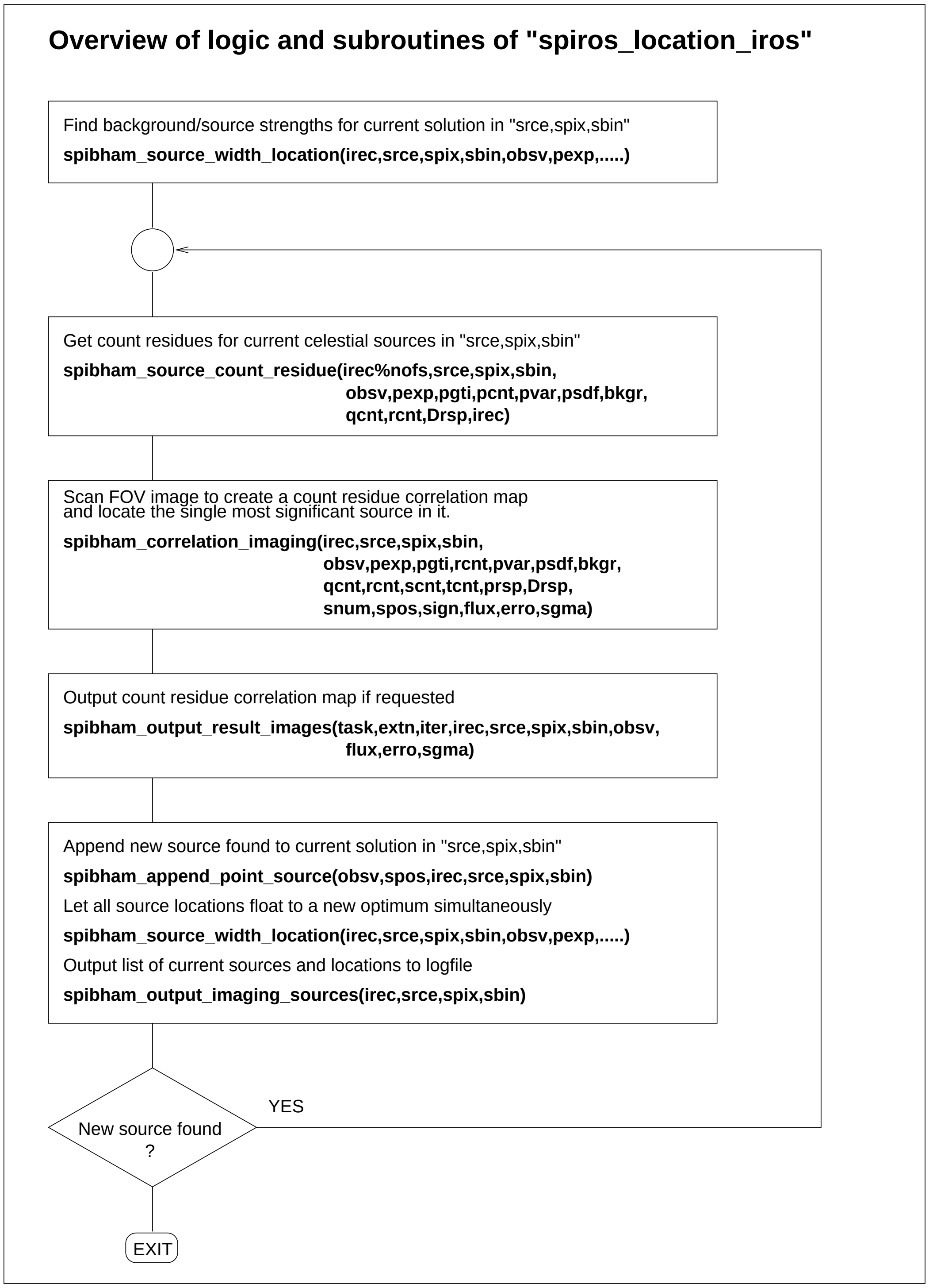
The main execution subroutine for this mode is **spiros_location_analysis** whose main tasks are called as follows:



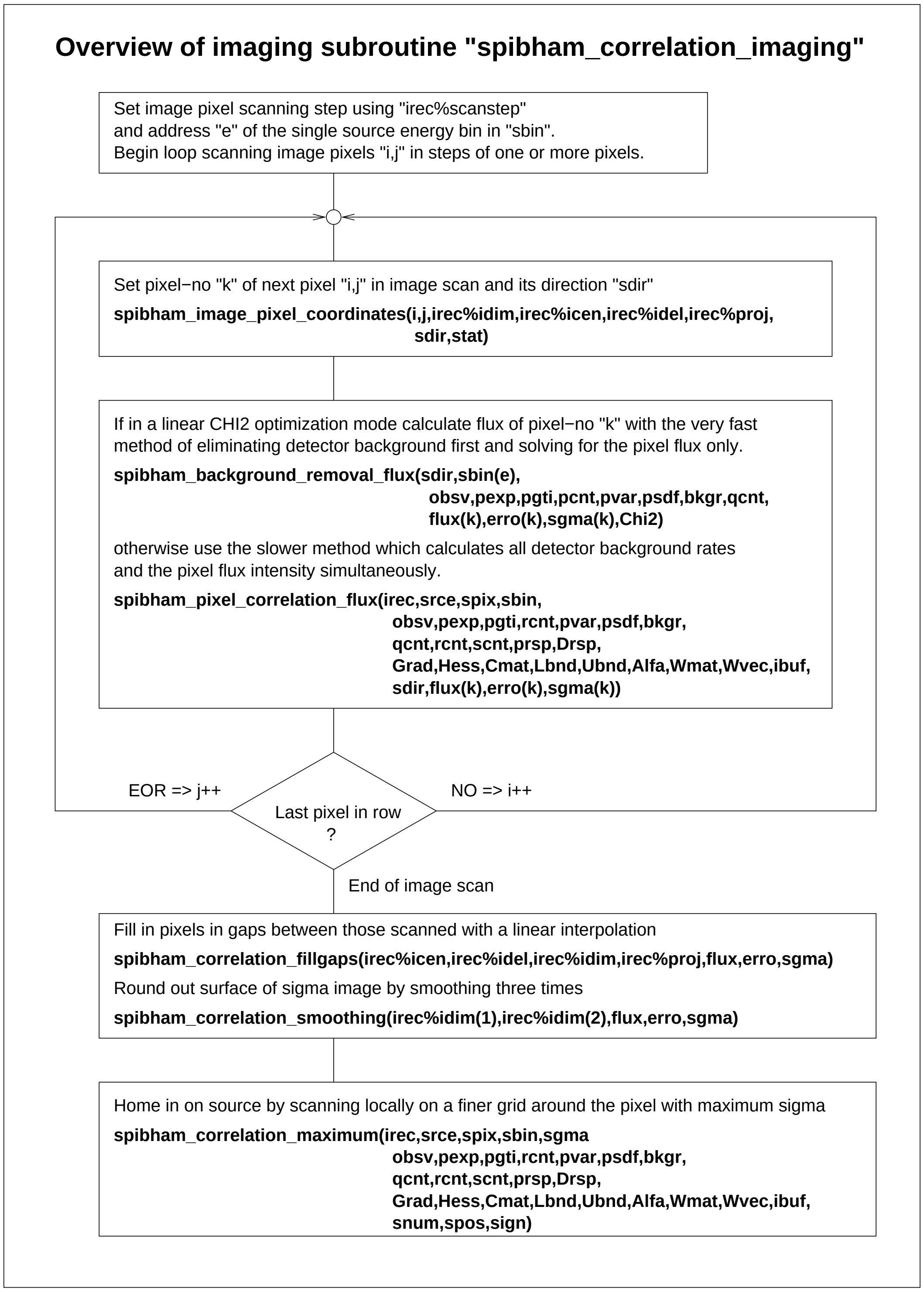
The subroutine **spiros_location_binwise** is used to locate all sources in each energy bin and has the following structure:



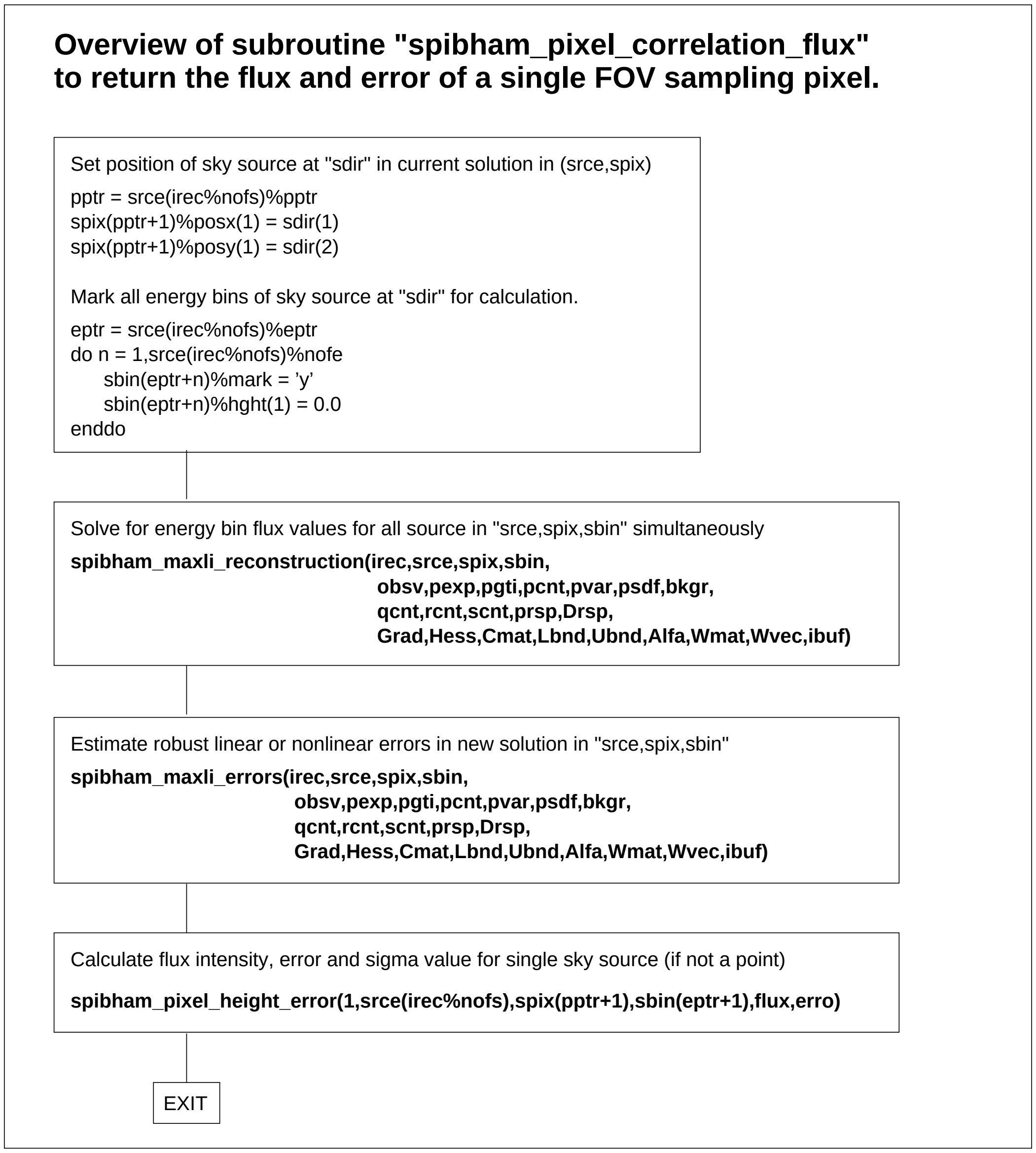
The subroutine **spiros_location_iros** is used to locate all sources in just one energy bin and has the following structure:



The subroutine **spibham_correlation_imaging** is used to locate just one source in one energy bin and has the following structure:



The subroutine **spibham_pixel_correlation_flux** is used to calculate all detector background rates and a single source flux simultaneously and has the following structure:



The subroutine **spibham_maxli_reconstruction** is used to calculate all detector background and any number of source spectra simultaneously and will return a solution by optimizing either a χ^2 parameter using subroutine **spibham_maxli_reconstruction_CH** or a **Maximum Likelihood** parameter using subroutine **spibham_maxli_reconstruction_ML** both of which are described under the **srce,spix,sbin** source, pixel, spectrum imaging system.

7.10 Spectral reconstruction software for SPECTRAL mode.

The main execution subroutine for this mode is **spiros_spectra_analysis** whose main tasks are called as follows:

Overview of logic and subroutines of "spiros_spectra_analysis"

Allocate memory for OG+NAG work arrays and source, pixel energy buffers "srce,spix,sbin"
spiros_spectra_memory(irec,obsv)

Create an initial solution for background and catalogue sources
spiros_source_pixel_bin_input(irec,obsv,srce,spix,sbin)

Catalogue of known sources

If parameter "irec%erspmat" is set to "N" calculate the background and source flux in each count energy bin independently using the IRF photopeak response only.

spiros_location_binwise(irec,srce,spix,sbin,flux,erro,sgma,
obsv,pexp,pgti,pcnt,pvar,psdf,bkgr,
qcnt,rcnt,scnt,prsp,Drsp,
Grad,Hess,Cmat,Lbnd,Ubnd,Alfa,Wmat,Wvec,ibuf)

otherwise if "irec%erspmat" is set to "Y" calculate the flux in all background and source energy bins simultaneously using the IRF Energy Response Matrix.

spiros_location_allbins(irec,srce,spix,sbin,flux,erro,sgma,
obsv,pexp,pgti,pcnt,pvar,psdf,bkgr,
qcnt,rcnt,scnt,prsp,Drsp,
Grad,Hess,Cmat,Lbnd,Ubnd,Alfa,Wmat,Wvec,ibuf)

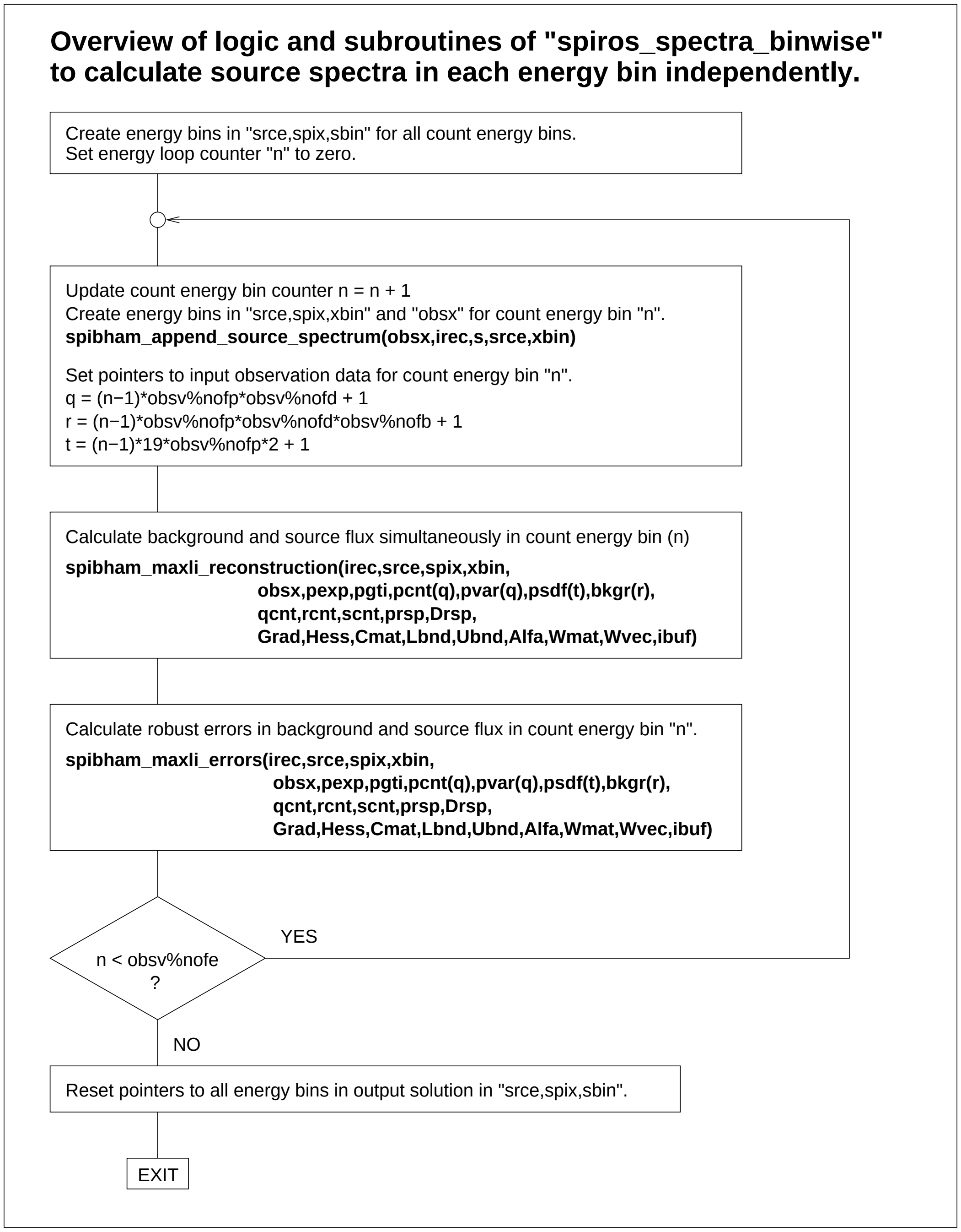
Output source, background and source detector spectra.
spibham_output_source_spectra(irec,srce,spix,sbin,obsv)
spibham_output_backgr_spectra(irec,srce,spix,sbin,obsv)
spibham_output_srcdet_spectra(irec,srce,spix,sbin,obsv,pexp,pgti,psdf,bkgr,qcnt,Drsp)

Output count residues and a logfile summary of the results.
spibham_output_maxli_residues(irec,srce,spix,sbin,obsv,......,Drsp)
spiros_analysis_summary(irec,obsv,srce,spix,sbin,pgti,pcnt,pvar,bkgr,xref,qcnt,scnt,rcnt)

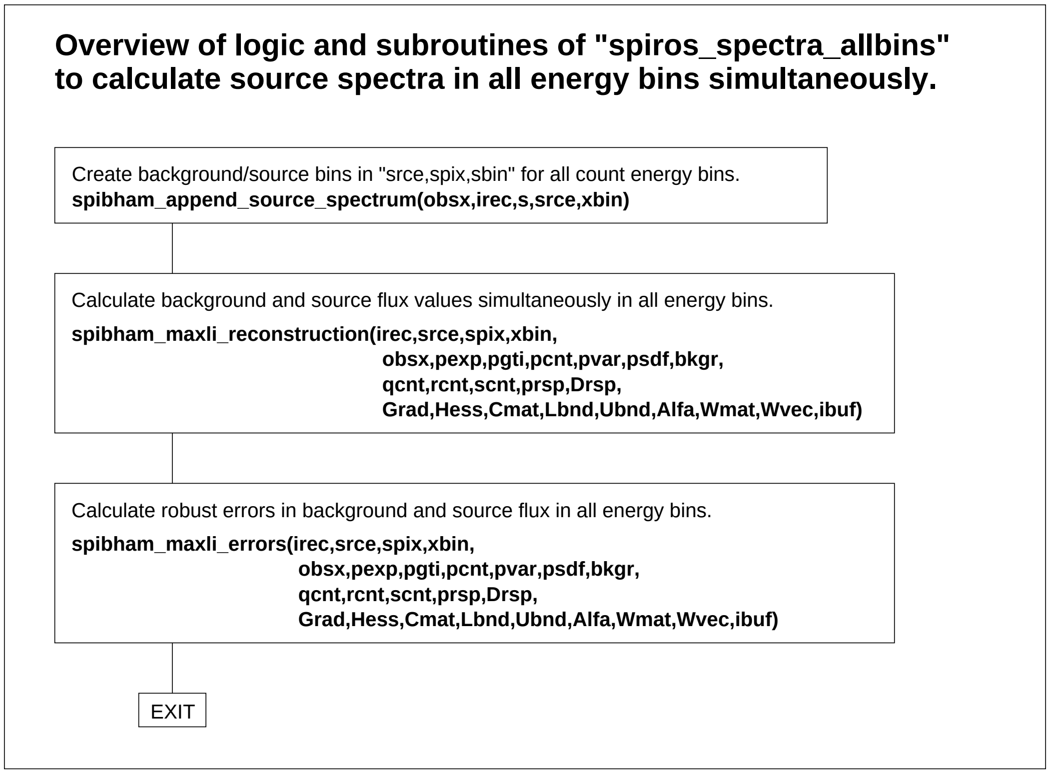
List any error messages returned calculating the flux in any energy bin.
spiros_error_message_output(irec,srce,sbin)

EXIT

The subroutine **spiros_spectra_binwise** is used to calculate detector background rates and source flux values together in each count energy bin independently with only an IRF photopeak response as follows:

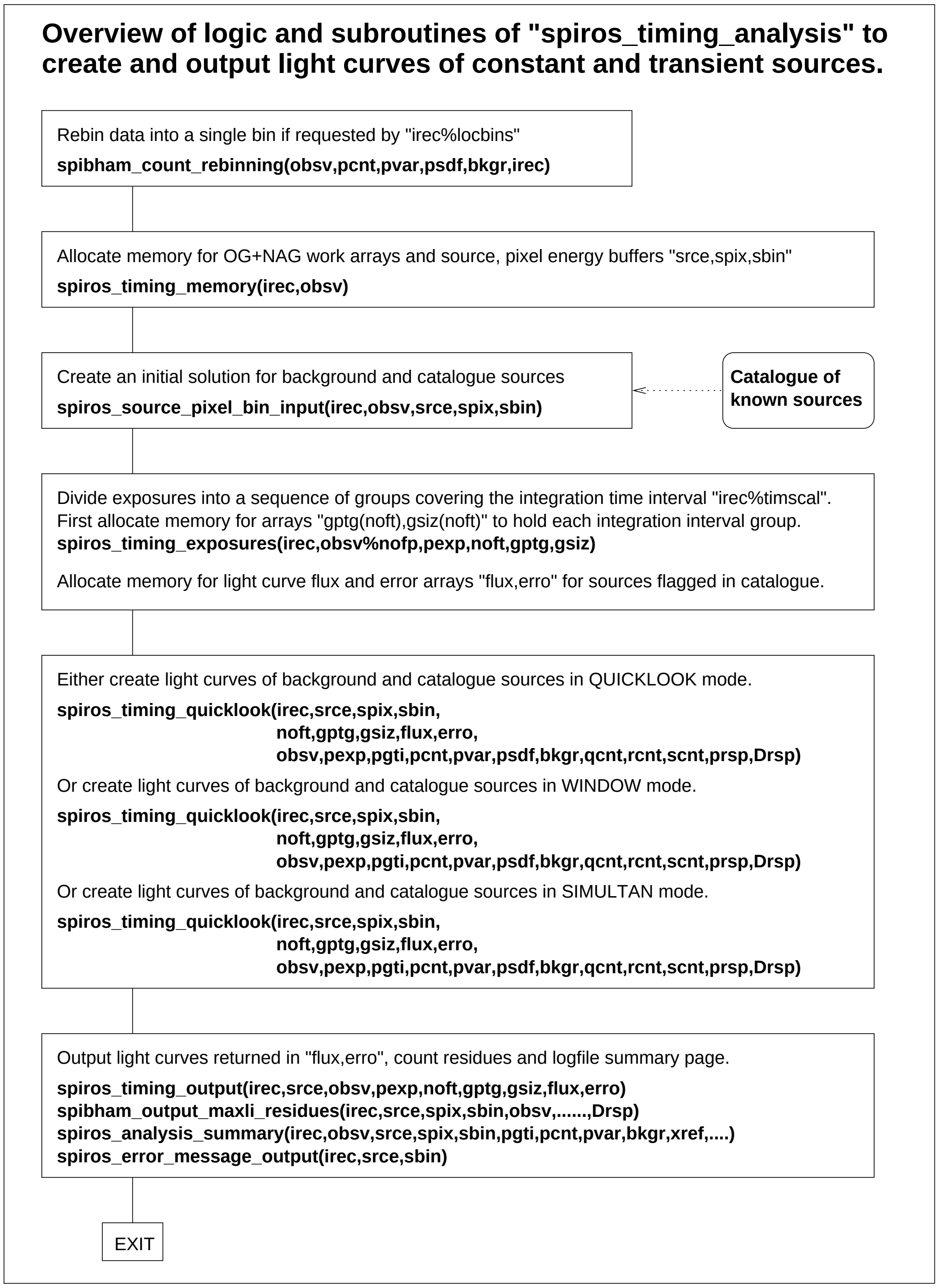


The subroutine **spiros_spectra_allbins** is used to calculate all detector background rates and source flux values in all count energy bins simultaneously using the IRF photopeak response and its Energy Response Matrix. I has a simple structure as follows:



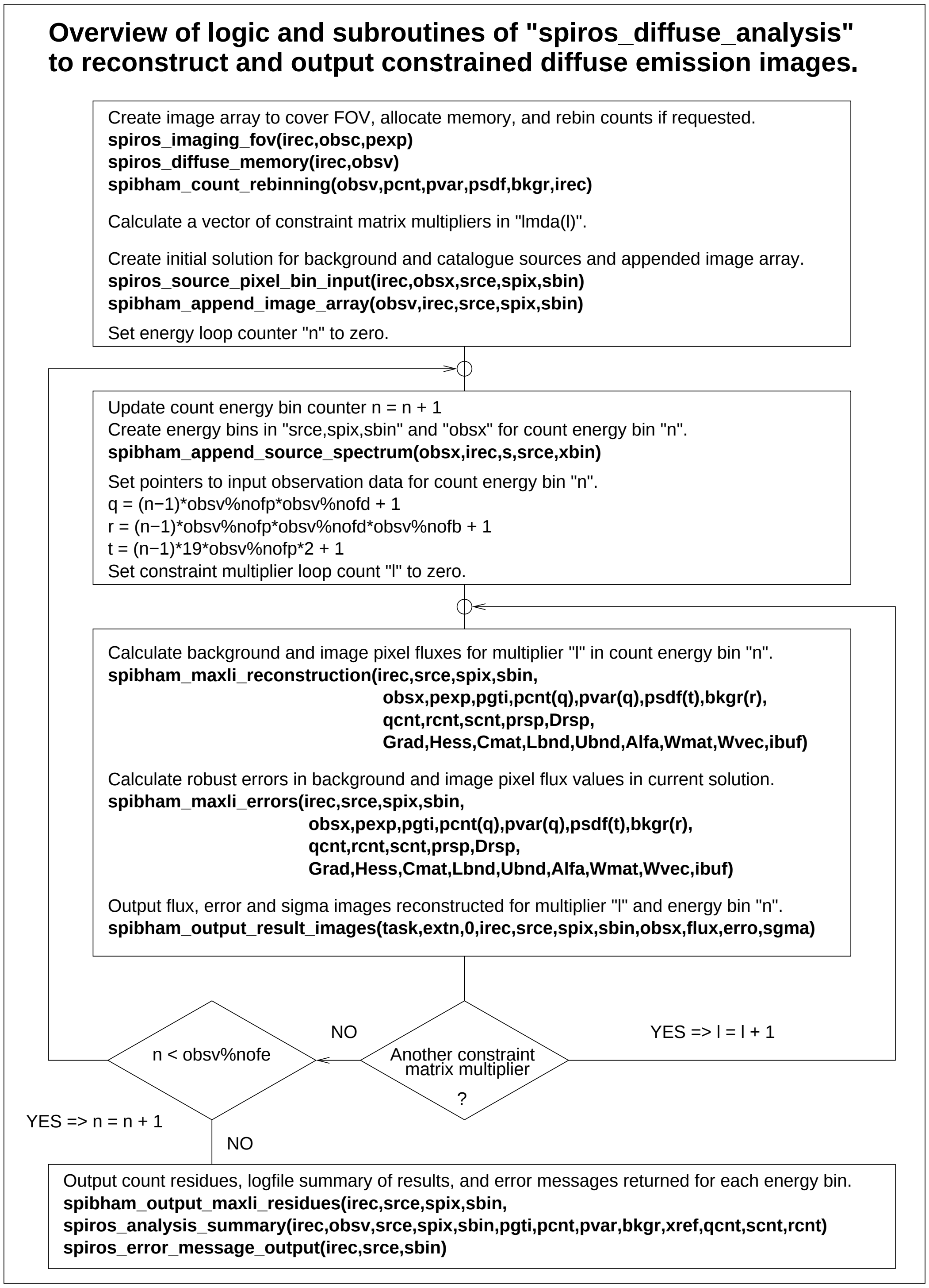
7.11 Transient source analysis software for TIMING mode.

The main execution subroutine for this mode is **spiros_timing_analysis** to select one of three methods of creating light curves as follows:



7.12 Diffuse emission imaging software for DIFFUSE mode.

The main execution subroutine for this mode is **spiros_diffuse_analysis** whose main tasks are as follows:



8 Cookbook of examples

In this chapter a number of practical examples are presented showing the use of SPIROS and the relation of its input parameters to its output of images, spectra and light curves.

With each example a zipped up TAR file is also provided which can be downloaded from an ISDC website then unzipped and expanded to yield a group of data directories containing input observation datasets, parameter files and a script which can be executed immediately to show the user examples of what SPIROS can do.

Typically such a script will use the input data to run SPIROS in two or more of its analysis modes **IMAGING**, **SPECTRAL** and **TIMING** and place the output in separate directories named after each mode.

It is to be noted that the point of these scripts is to provide the new user with examples of SPIROS which do actually work along with examples of the output of each. The idea is that the user should take the opportunity to use the images and spectra provided to both examine them and familiarize themselves with software display tools such as

- **DS9** from the **SAO** High Energy Astrophysics Division and available via their website **http://hea-www.harvard.edu/RD/ds9**
- **FV** from **HEASARC - GSFC** to display and annotate images and spectra and available via **http://heasarc.gsfc.nasa.gov/docs/corp/software.html**.
- **GAIA** for imaging analysis from **Rutherford Appelton Laboratories** available via **http://www.starlink.rl.ac.uk/star/docs/sun214.htx/sun214.html**

These examples are useful for the user to change some input parameters to see the effect on the output or to use the parameter files as templates for their own particular observation data and analysis required.

The examples given here are:

- Locating Crab then reconstructing its spectra and light curve
- Imaging five sources near the galactic centre
- Resolving three point sources separated by one degree
- Imaging 511 keV diffuse emission in the galactic plane

8.1 Prepared data and parameters to test SPIROS.

Each SPIROS test package can be downloaded from its ISDC website then unzipped and expanded with typical UNIX commands like

gunzip spirosexample.tar.gz

tar xvf spirosexample.tar

This will result in the appearance of a SPIROS execution script **run_spiros** and the following data directories:

- **OBSDATA**
  Contains datasets describing observation count data, exposure times and pointing directions.
- **IRFDATA**
  Contains Instrument Response Functions at various energies and an index file to access them.
- **INPUT**
  Contains input parameter files required for each analysis mode SPIROS will execute plus an input catalogue of located sources for **SPECTRAL** and **TIMING** mode.
- **IMAGING,SPECTRA,TIMING**
  These are output directories and will contain the images, spectra, light curves and logfiles output during execution of each SPIROS analysis mode.
- **IMAGING_examples,**
  SPECTRA_examples,
  TIMING_examples
  These will contain examples of the output images, spectra, logfiles, etc to be expected after execution of the SPIROS script provided and should be viewed by the new user to get some idea of the kind of output from SPIROS.

To execute the SPIROS test package the user must do only two things

- Set an environment variable **SPIROS_EXEC** to tell the execution script where the current SPIROS executable is to be found, typically with a UNIX command like **setenv SPIROS_EXEC "pathway/spiros"**
- Type in the execution script name **run_spiros** and wait for SPIROS to finish.
  The user can keep track of what is happening by viewing the logfile **spiros.log** during execution and then displaying the ***.fits** images and spectra and ***.log** logfiles in directories **IMAGING**, **SPECTRA** and **TIMING** when finished.

8.2 The location, spectra and light curve of Crab.

This is an example of the location and spectral extraction of a strong source like **Crab** using simulated observation data in 1000 energy bins covering **40-2000 keV**. The observation consisted of a sequence of 81 exposures of 1800 seconds each on a 9x9 pointing grid covering a 16^o x 16^o rectangular area which is roughly the size of the fully coded field of view of SPI.

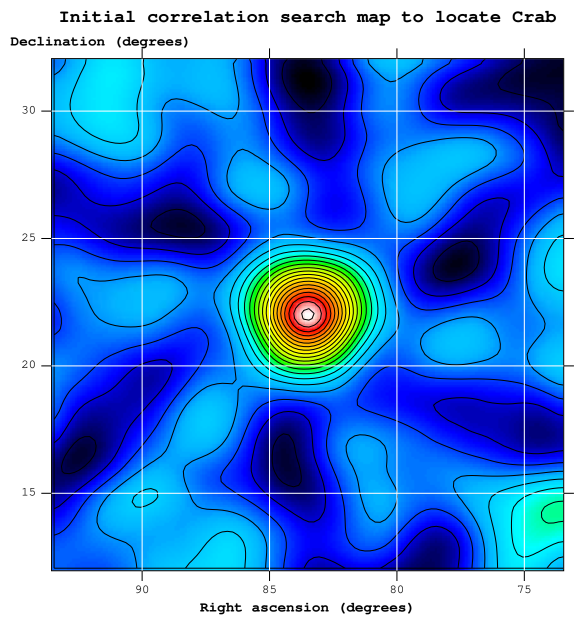


Figure 13: Point spread function of a single source like Crab.

All observation datasets and input parameters for this example can be found in

spiros_CRAB_40_2000_powl.tar.gz

In this example the script **run_spiros** will run **SPIROS** three times using observation datasets stored in directory **OBSDATA**:

- First in **IMAGING** mode to locate Crab and output images of it using input parameters stored in **INPUT/CRAB_40_2000_powl_imaging.par**.
- Second in **SPECTRAL** mode to extract the photopeak spectrum of Crab using its source location found in the output catalogue from **IMAGING** mode. It will use input parameters in **INPUT/CRAB_40_2000_powl_spectra.par**.
- Third in **TIMING** mode to extract a light curve showing any flux variability. It will use input parameters in **INPUT/CRAB_40_2000_powl_timing.par**.

The location of **Crab** found in the image above is stored in a catalogue which is used in **SPECTRAL** mode to extract the photopeak spectrum below. Note that the flux values are per bin and not per keV.

\includegraphics[scale=0.4,angle=0.0]{SPIROS_cookbook_fig2.ps}

Figure 14: Spectrum extracted after locating Crab in the above image.

Similarly the catalogue location of **Crab** is used to extract the light curve below in **TIMING** mode showing the estimated total flux at the midpoint of each exposure interval, indicating an apparent variability that is in fact not present.

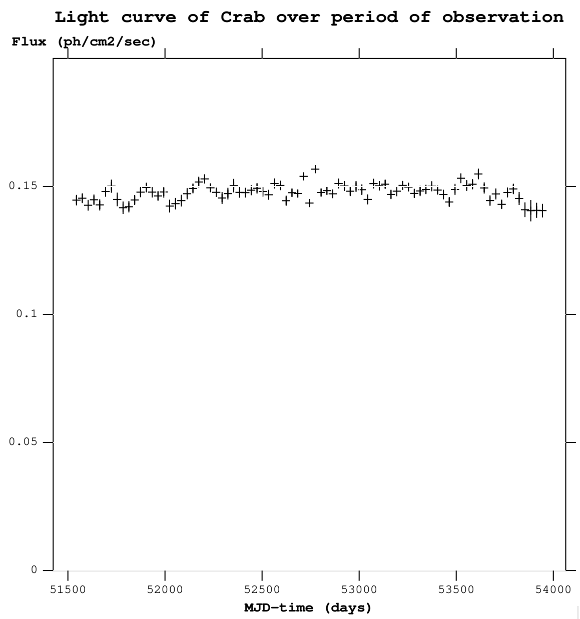


Figure 15: Light curve of Crab extracted over period of observation.

8.3 Locating five sources near the galactic centre.

This is an example of source location and spectral extraction using simulated observation data in 5 energy bands covering **70-150 keV** of five sources near the galactic centre identified as **1E1740.7-2942, GRS1758-258, GX354-0, GX1+4** and **TERZAN-2** The observation consisted of a sequence of 1075 exposures of 4465 seconds each covering a 60^o x 40^o rectangular area and which is to be used by SPI for a deep survey of the galactic centre region for a total observation time covering 1333 hours. This observation sequence is called the **Galactic Centre Deep Exposure** survey or **GCDE**.

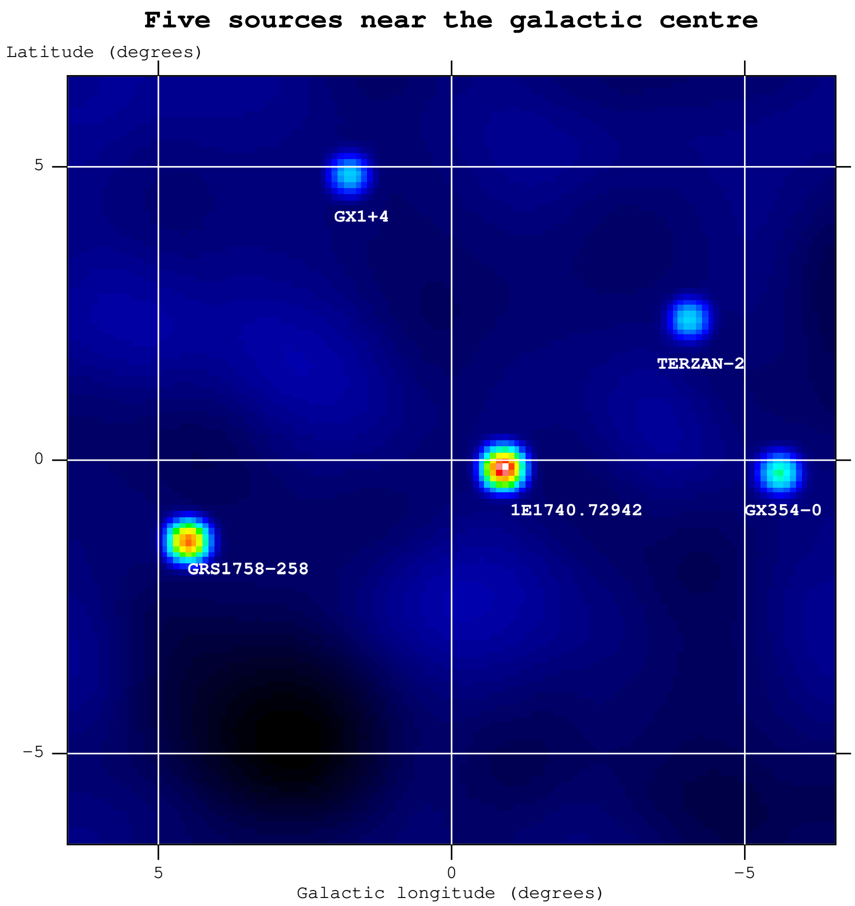


Figure 16: Five sources simulated with the GCDE observation sequence.

All observation datasets and input parameters for this example can be found in

spiros_GCDE_five_sources.tar.gz

In this example the script **run_spiros** will run **SPIROS** twice using observation datasets stored in directory **OBSDATA**:

- First in **IMAGING** mode to locate and image the five sources observed using input parameters stored in **INPUT/GCDE_lund_70_150kev_point_images_spiros.par**.
- Secondly in **SPECTRAL** mode to extract the photopeak spectrum of each source found in the output catalogue from **IMAGING** mode. It will use input parameters in **INPUT/GCDE_lund_70_150kev_point_spectra_spiros.par**.

8.4 Resolution of three sources a degree apart.

In this example the user can begin to see how SPIROS goes about "floating" all sources found to find their most accurate location. It is an interesting case because the three sources observed are all quite close to each other at the corners of a triangle with a side of 1.0^o, well within the $\approx 3.0^o$ source resolution of SPI. The sources all had the same strength of 100 mCrab and flux sigma values of ≈ 90 when located and under these circumstances may be resolved but for the much weaker sources expected this will not be the case and the output from the **IBIS** or **JEMX** instruments will be required to resolve them.

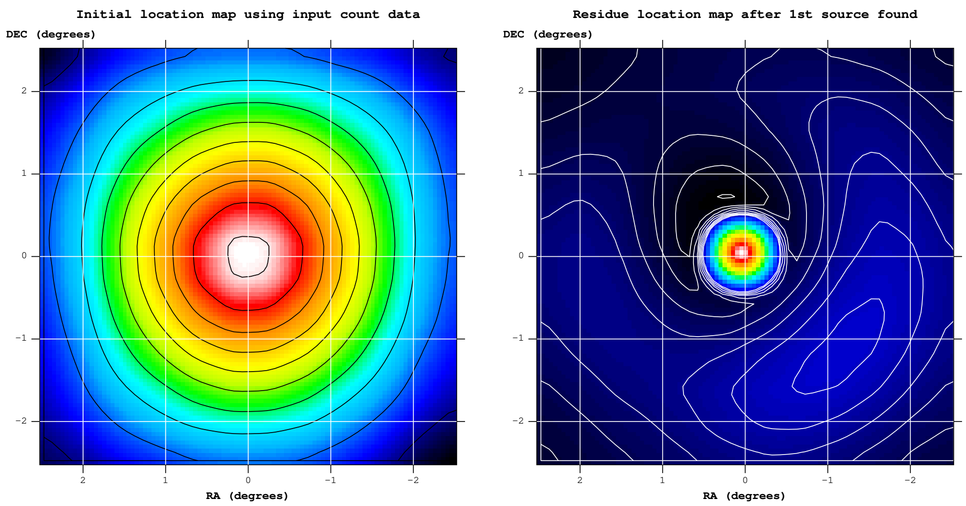


Figure 17: Location maps used to search for 1st and 2nd sources.

In the figures above it can be seen that the three sources can not be distinguished from a single blob of diffuse emission. It is interesting however to note from contours in the location image on the right that another source candidate is indeed possible.

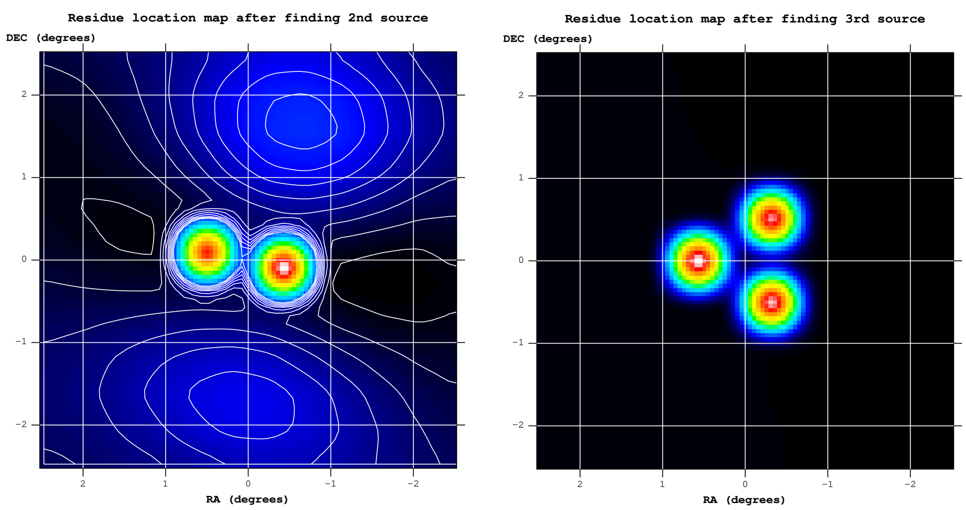


Figure 18: Location map and output image after searching for 3rd source.

In the lefthand image above it can be seen that a second source has been located but that it has then moved a considerable distance, along with the 1^{st} source, to a more accurate location. Again from the contours in the image a 3^{rd} candidate can be identified and in the righthand image the 2^{nd} and 3^{rd} sources have again moved a considerable distance to their true locations at the corners of an equilateral triangle.

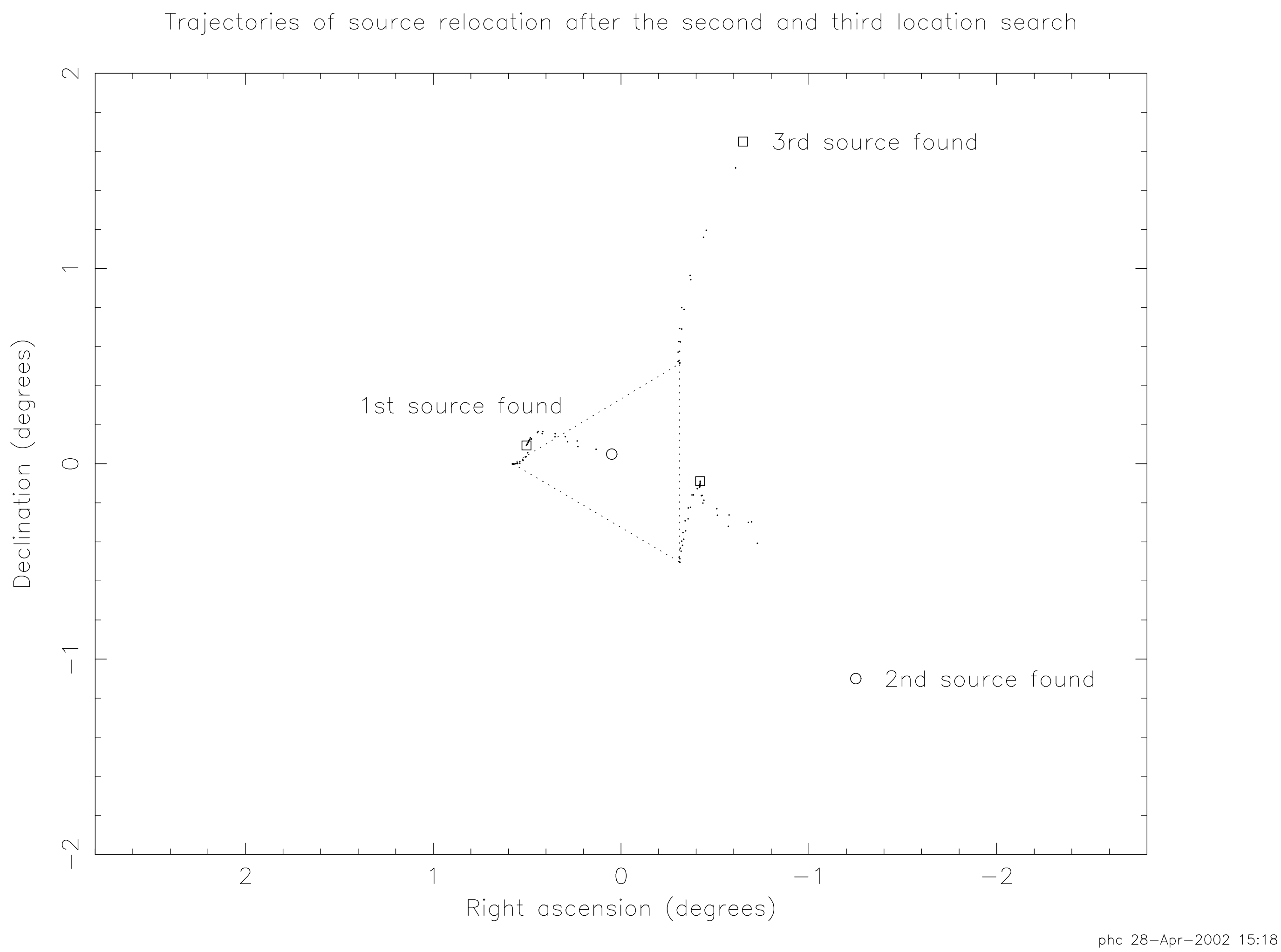


Figure 19: Source migration trajectories after each location search.

The relocation trajectories of each source after a new source has been identified in the residue location maps can be seen in the diagram above which shows the intermediate steps SPIROS calculates in getting all sources to migrate to more accurate locations. The first source found is initially at the centre of the triangle and after the second search the two sources migrate from the locations given by a circle to those given by a square. After the 3^{rd} source is found at the top they all make a last migration to the corners of the triangle shown where they really are. For sources $2-3^{o}$ or more apart sources are initially located quite close to their true positions and do not have to migrate far but when close together this becomes considerably more complicated and slower.

All observation datasets and input parameters for this example can be found in

spiros_triple_point_resolution.tar.gz

In this example the script **run_spiros** will run **SPIROS** in **IMAGING** mode to locate the three sources using observation datasets stored in directory **OBSDATA** and input parameters stored in **INPUT/spiros_triple_point_resolution_imaging.par**