CIRS Spectral Utility Programs

Conor A Nixon, Ground Data System Working Group, GSFC

April 16, 2003

Contents

1 Introduction

This document describes the utility programs which are currently available on the `CIRSDATA` data server machine at GSFC. These programs are designed to help users extract text versions of the spectra from the binary database.

The `Vanilla` program itself, which is the primary low-level method for interogating the CIRS database, is not described herein. Whereas only certain information can be extracted using the programs `cirs_spec` and `cirs_avspec`, any or all information can be extracted using `Vanilla`, albeit in a less user-friendly way. For documentation on `Vanilla` check out the website at: `vanilla.asu.edu`. The `Vanilla` users guide can also be downloaded by anonymous FTP from `east.la.asu.edu` in the directory `pub/software/vanilla`. There is also an on-line user guide to the CIRS inplementation of `Vanilla` by Monte Kaelberer at `http://cirswww.gsfc.nasa.gov`.

For users who require more information on the processing of CIRS data and the software used at the base level to construct the database, pleas refer to the document 'CIRS Ground Data System'. The CVS version is to be found in `/data/cvs/docs` on `CIRSDATA`.

2 Running programs on `CIRSDATA`

In order to run any of the following programs when logged into `CIRSDATA`, you must have the directory `/usr/local/bin` in your command path. If you are unsure whether your path is set correctly, try either of the following:

1. Type '`which cirs_time`' and hit 'ENTER'. You should get the result: '`/usr/local/bin/cirs_time`'.
2. Type '`printenv PATH`' and hit 'ENTER'. You should see a list of colon-separated directory names. If '`/usr/local/bin`' is in the list then your path is set correctly.

If your path is not set correctly, you can still run the programs, but will need to put the full path in front of the command name. Hence, you would type '`/usr/local/bin/cirs_time`' instead '`cirs_time`' and similarly for the other programs. Contact either of the following `CIRSDATA` sysadmins for help in setting your path.

Matt Elliott (`Matthew.H.Elliott@gsfc.nasa.gov`)
Jim Tingley (`James.S.Tingley@gsfc.nasa.gov`)

3 Basic Programs

3.1 cirs_time

Summary: This program is designed to help users convert between the integer number 'Spacecraft Event Time' (SCET) and text time. SCET is used as the primary key field in the database: all spectra, interferograms and other information are stored in sequential order of SCET time. However, a user will more commonly know the date of the observation they wish to find.

Syntax: `cirs_time` **scet**
or
Syntax: `cirs_time` **mm/dd/yy hh:mm:ss**

Usage: If the program is given a single argument, it assumes that it has has been given SCET and wil print out the text time, including day-of-year (DOY). If two arguments are given, it assumes that it has been given text time in the format above, and will convert to SCET.

`cirs_time` with a '-h' or '-help' argument will print a usage summary.

3.2 cirs_spec

Summary: `cirs_spec` allows the user to extract a single interferogram from the database and print to the screen. Unlike the actual `Vanilla` program however, `cirs_spec` prints the spectrum in a column, i.e. on sucessive lines, rather than on a single line. Also, in the database, only the start wavenumber and wavenumber interval are stored for a particular spectrum. `cirs_spec` will automatically extract the wavenumber scale and prepend each radiance by its wavenumber. Hence, `cirs_spec` is a useful program for quickly extracting a spectrum into a usable text form for plotting, retrievals etc.

Syntax: `cirs_spec` **scet**

`Usage:` By default the program will read apodised spectra from the `/data/currentdb/apod_spectra` directory on `CIRSDATA`. Also, the default detector to extract is 0 (FP1). If no spectrum for the given time/detector combination exists, an error message will be printed. The following are optional arguments, which if present, must be given after the scet time (however multiple options are allowed):

`-v`	verbose option
`-apod`	read from `/data/curentdb/apod_spectra`
`-unapod`	read from `/data/currentdb/unapod_spectra`
`-det d`	read spectra for detector number `<d>`
`-dir <dir>`	read from directory `<dir>`
`-regrid`	read re-gridded/interpolated spectra (ISPM type)

`-noregrid` read non-regridded/non-interpolated spectra (SPM type)

The default is to read interpolated spectra (ISPM type). Note that the directory `/data/currentdb/` is a symbolic link which points to the most recent version of the database. The following is an example of the current output format (subject to change):

```
header 4
spts 4096
emission_angle[5] 90.000000
longitude_zpd[5] 120.595795
latitude_zpd[5] -40.646229
  566.48157     -9.9948e-07
  566.61987     -9.8729e-07
  566.75818     -9.9644e-07
  566.89648     -1.0175e-06
  567.03479     -1.0022e-06
```

Only the first 10 of 4101 lines have been shown here. Although more information lines may be added to the header, it is guaranteed that the first line of the file will be '`header n`' giving the number of lines of information immediately following the first, but before the actual spectral data.

In the above example, the '[5]' indicates that the information is for the 5th Q-point of the detector. See figure 1.

Note that the output may be conveniently re-directed from the screen to a file by appending the following to the command string: '`> myfile.txt`'. The text will then be put in the file '`myfile.txt`' in the current working directory. You must make sure you have write permission for the working directory.

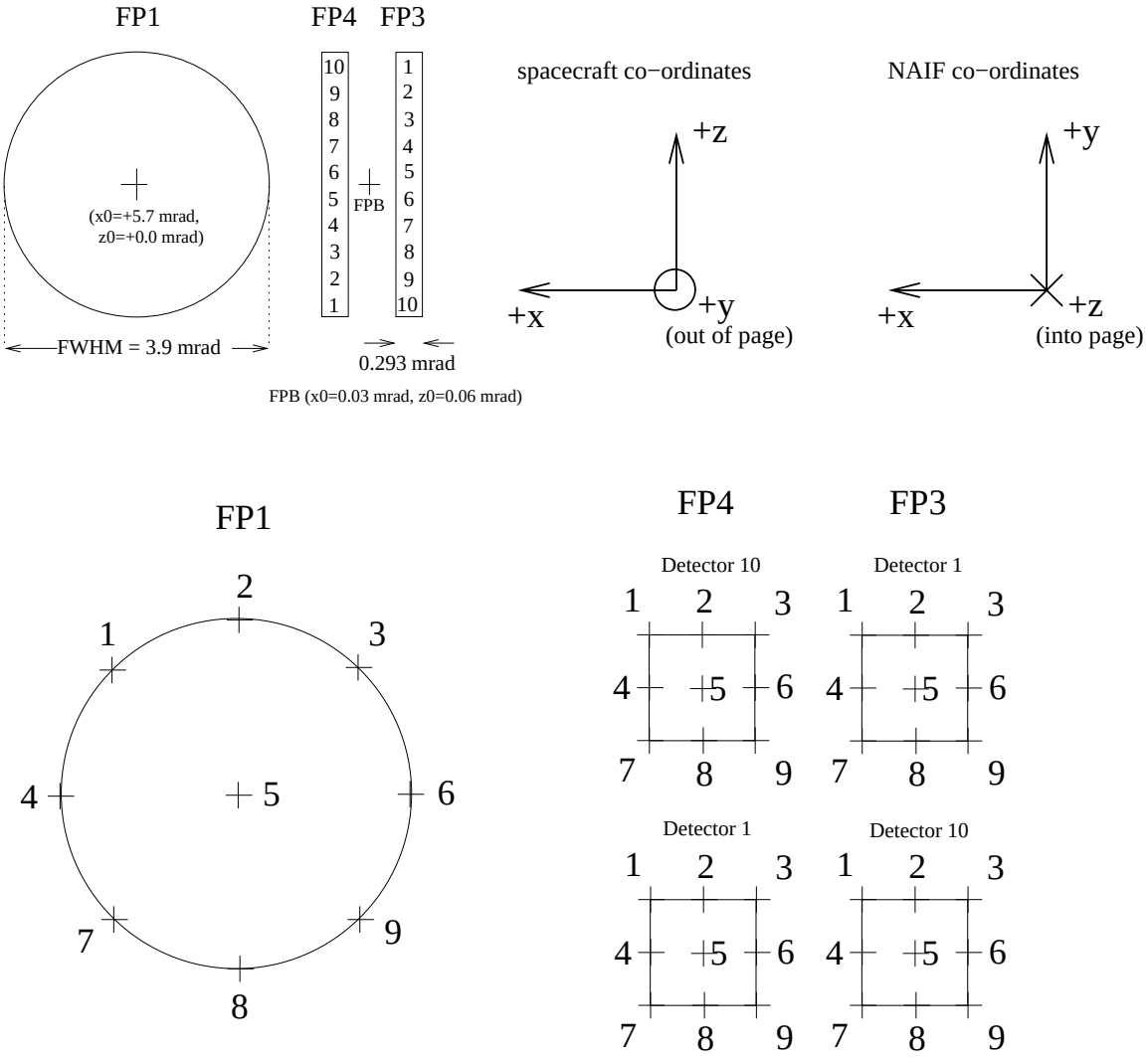

Figure 1: Q-points for CIRS detectors.

4 Averaging Programs

The programs described in this section can all be used to take the average, or co-add, CIRS spectra, in different ways. All programs share a common format of *task* file, which specifies exactly which spectra are to be extracted and averaged, and what form the output should take.

4.1 Averaging Task Files

Each task file consists of a series of data lines, which provide information to the averaging program.[1] Each data line normally consists of a **keyword** followed by one or more data items. Data lines can now occur in any order, unlike earlier versions.

Data line keywords:

The following compulsory data lines must be set:

dir <`dirname`> sets the target Vanilla directory to `dirname`.

rti <`n`> **shutter** <`s`> **deep_space** <`d`> selects spectra of RTI length `n` (scan's length measured in 1/8 sec, in the range 32 to 404), shutter status `s` (O or C), and deep_space field `d` (1 or 0). See the data system description document for the current definition of 'deep_space'.

lat <`lt1`> <`lt2`> <`dlt`> <`ltstep`> indicates that the program should restrict the spectra being averaged to the latitude range `lt1`–`lt2` (compulsory). If gridding is used, then `dlt` sets the latitude bin width and is compulsory. `ltstep` may be used to set the latitude step size, which defaults to the bin width if not present. For the non-gridding, these two parameters are ignored if present.

lon <`ln1`> <`ln2`> <`dln`> <`lnstep`> indicates that the program should restrict the spectra being averaged to the longitude range `ln1`–`ln2` (compulsory). If gridding is used, then `dln` sets the longitude bin width and is compulsory. `lnstep` may be used to set the longitude step size, which defaults to the bin width if not present. For the non-gridding, these two parameters are ignored if present. **Note:** all averaging programs provide an 'intelligent' longitude interface to Vanilla. Longitudes in the range -360 to $+360$ are accepted, however, the overall range specified must satisfy `ln2`$-$`ln1`≤ 360.

and one of the following:

fp1det `0 0` indicates that the program should average FP1 spectra.

fp3det <`d1`> <`d2`> indicates that the program should average FP3 spectra, for detectors numbered `d1` to `d2` (in the range 1–10 or 21–25).

[1] Users of earlier versions of the programs should note that some of these data were previously supplied either on the command line, or in the task file but in a different format

fp4det <d1> <d2> indicates that the program should average FP4 spectra, for detectors numbered d1 to d2 (in the range 11-20 or 26–30).

The following lines are optional:

fp1range <w1> <w2> indicates that the program should restrict the wavenumber range of FP1 spectra during output, to the range w1 to w2.

fp3range <w1> <w2> indicates that the program should restrict the wavenumber range of FP3 spectra during output, to the range w1 to w2.

fp4range <w1> <w2> indicates that the program should restrict the wavenumber range of FP4 spectra during output, to the range w1 to w2.

fp1errfil <filename> indicates that the program should read FP1 NESRs from the file filename and add a column of spectral NESRs to the output spectra.

fp3errfil <filename> indicates that the program should read FP3 NESRs from the file filename and add a column of spectral NESRs to the output spectra.

fp4errfil <filename> indicates that the program should read FP4 NESRs from the file filename and add a column of spectral NESRs to the output spectra.

fp1errpc <ppp.pp> indicates that the program should output a column of spectral NESRs for FP1, calculated as a percentage ppp.pp of the spectrum.

fp3errpc <ppp.pp> indicates that the program should output a column of spectral NESRs for FP3, calculated as a percentage ppp.pp of the spectrum.

fp4errpc <ppp.pp> indicates that the program should output a column of spectral NESRs for FP4, calculated as a percentage ppp.pp of the spectrum.

fp1errsd indicates that the program should output the standard deviation of the averaged data as a third spectral column.

fp3errsd indicates that the program should output the standard deviation of the averaged data as a third spectral column.

fp4errsd indicates that the program should output the standard deviation of the averaged data as a third spectral column.

If the program encounters a line which does not begin with a recognised keyword, this is assumed to be an extra Vanilla modifier. Lines of this type are not parsed, but copied verbatim to the Vanilla query used to extract the data. See example below. *It is recommended that the user always sets at least the target_id as an optional modifier, otherwise all available targets will be co-added!.*

Notes:

1. Only keywords relating to one focal plane should be used,[2] and this should match the focal plane set with the `fp*det` keyword.

2. Only `fp*errfil` or `fp*errpc` should be set, not both. I.e., the NESR can only be calculated by one method.

3. If multiple focal planes are specified, or conflicting keywords, in general the last one to occur in the file will supercede ones which occur earlier.

4. Care must be taken to ensure that the spectral range, if specified via the `fp*range` keyword, does not conflict with other considerations, i.e. the detectors/ focal plane in the averaging.

NESRs: If the `fp*errpc` or `fp*errfil` keyword is used then a third column of spectral errors is printed, after the the wavenumber and radiance. This is very useful for quickly setting up retrievals. Spectral NESRs, either calculated as a fixed percentage of the average spectrum, or read from a file, are then divided by the square root of the number of spectra in the average, for correct weighting.

If the `errfil` method is used, it is not necessary that the NESR spectrum and the spectra being averaged are defined on the same wavenumber grid. The NESRs will be linearly interpolated to the positions of the spectral data. Note that if the NESRs do not cover the entire spectral range which is being averaged, then the final spectrum is trimmed at either or both ends. I.e. only points in the spectrum within the NESR range will be output.

Example:

```
dir /data/ver_0.5/apod_spectra
fp1det 00 00
lon 0 360 360
lat -80 80 10
rti 400 shutter 0 deep_space 0
emission_angle[5] 0.0 45.0
fp1errfil /data/nesr/nesr_fp1_p5cm-1.txt
scet 975628800 978307199
target_id 599 599
```

4.2 cirs_avspec

Summary: `cirs_avspec` produces averaged spectra for specified time periods, detector numbers, and other ranges of database fields. A single average spectrum is produced and different detectors are co-added if present.

Syntax: `cirs_avspec` **[filename] [-v] [-h] [-nohead] [-regrid] [-noregrid] [-tb] [-nw] [-fovfill]**

[2] This restriction does not apply to `cirs_microwin`.

Usage: To run the program, simply typing the command name alone is sufficient. The program will default to reading information about the averaging 'task' from the file `avspec_task.lst`, unless another filename is given as an argument. Using '-' as the filename (no quotes) will read the task from the standard input stream. See description of the task file format below. The `-h` option prints help and exits without executing.

The `-v` argument enables verbose output as usual, including the actual SCET times which are selected to go into the average spectrum.

The `-nohead` option prevents the standard three-line header from being prefixed to the spectral data in the output file (see below).

Other options are `-regrid/-noregrid`, which determine whether the code attempts to read the ISPM (interpolated, re-gridded spectra) or SPM (non-interpolated spectra) respectively, from the target directory. The default is now to read interpolated data.

The `-tb` argument enables brightness temperature output instead of radiance, and the `-nw` argument rescales the radiance by a factor $\times 10^9$ (short for nanowatts).

The `-fovfill` argument causes individual spectra to be divided by the FOV filling factor (as stored in the pointing data). This may be useful for including partially-filled FOVs in the average, which should otherwise be excluded by the user (using the 'all_q_on' field in Vanilla). However, caution is urged when using this feature: because the SNR of partially-filled fields may be lower, and so, very low filling-factors will contribute lareg amounts of noise after re-scaling.

Output files: The results of the averaging are dumped into a single text file with the name: `avspec_detD1--D2.txt`, where D1 and D2 are the first and last detectors contributing to the average. If the file exists already it is overwritten.

All output spectral files now begin with a three-line header. The header information is:

```
ave_fwhm ave_lat ave_lon nspec
npoints
incident emission azimuth
```

The first item is the average FWHM. Remember that even spectra which have nominally the same RTI length may have slight variations in the actual number of IFM points. Hence, the FWHM varies slightly and is averaged. The next two numbers are the mean latitude and longitude of spectra which were averaged. The number `nspec`, the number of output spectra, is 1.

`npoints` is the number of output spectral points (wavenumbers). On the third line is the average incident, emission and azimuth angle; although currently only the average emission angle is calculated. The other angles are set to zero.

Following the header is the spectral information:

```
566.481567 2.39581e-07
567.587952 2.21918e-07
568.694336 2.03812e-07
569.800720 2.04339e-07
570.907104 1.72582e-07
572.013489 1.63318e-07
573.119873 1.48168e-07
574.226257 1.73843e-07
575.332642 1.85276e-07
576.439026 1.7367e-07
```

etc

4.3 cirs_avspec_old

Summary: cirs_avspec_old produces averaged spectra for specified time periods, detector numbers, and other ranges of database fields. This is the old version of cirs_avspec, and produces a separate averaged spectrum for each detector.

Syntax: cirs_avspec_old [**filename**] [**-v**] [**-h**] [**-nohead**] [**-regrid**] [**-noregrid**] [**-tb**] [**-nw**] [**-fovfill**]

Usage and task file: Identical to cirs_avspec.

Output files: The only difference from cirs_avspec is that separate output files are produced for each detector. This program may now be more useful when non-regridded/interpolated spectra are being averaged, since these have different wavenumber axes for different detectors within the same focal plane, due to off-axis effects. Note that the restriction on detector range which pertains to cirs_avspec is lifted for cirs_avspec_old.

4.4 cirs_avspec_grid

Summary: cirs_avspec_grid produces averaged spectra for specified time periods, detector numbers, latitudes and longitudes, and other ranges of database fields. This is essentially the same as cirs_avspec, except that a unique average is produced for each lat/lon bin.

Syntax: cirs_avspec_grid [**filename**] [**-v**] [**-h**] [**-nohead**] [**-regrid**] [**-noregrid**] [**-tb**] [**-nw**] [**-fovfill**]

Usage: Identical to cirs_avspec.

Task file: Basically identical to cirs_avspec. The default filename is avspec_grid_task.lst. The only important differences are that the third and fourth latitude and longitude parameters (spatial bin width and step size) become active. The bin width is compulsory, and the bin spacing is optional. The bin spacing defaults to equal the bin width if not present.

Output files: A single output file: `cirs_avspec_grid.spe` is produced, which contains all the average spectra concatenated one after another. This is a suitable format for input to the Oxford `Oxmultiret` code for example.

4.5 `cirs_microwin`

Summary: `cirs_microwin` is more sophisticated version of `cirs_avspec_grid`, which allows for multiple spectral microwindows to be specified from one, two or three focal planes. In this usage, a microwindow simply refers to a continuous spectral region. For example, using `cirs_microwin` you could specify windows at 100–200 cm^{-1} and 300–400 cm^{-1} on FP1 and 800–900 cm^{-1} on FP3. Only these and no other spectral regions would appear in the final averaged spectrum.

Syntax: `cirs_microwin` [**filename**] [**-v**] [**-h**] [**-regrid**] [**-noregrid**] [**-tb**] [**-nw**] [**fovfill**]

Usage: Identical to `cirs_avspec`, except there is no `-nohead` flag.

Task file: Similar to `cirs_avspec_grid`. The the default filename is `microwin_task.lst`. The `fp*det` and `fp*range` keywords are disallowed, and replaced by the keywords `fp1win`, `fp3win` and `fp4win`. Each of these keywords must start a new line, and is followed by one or more pairs of wavenumber delimiters, specifying a microwindow. Each of these three keywords may be used multiple times in any order, and may be mixed.

```
Example 1:


fp1win 100.0 200.0
fp3win 850.0 890.0 1100.0 1150.0


Example 2:


fp3win 1100.0 1150.0
fp1win 100.0 200.0
fp3win 850.0 890.0
```

Example 1 and example 2 are identical specifiers of the three windows.

When FP1 is specified, detector 0 is included in the averaging. When FP3 is specified, detectors 1–10 are included. When FP4 is specified, detectors 11–20 are included. It is not possible to restrict the detector range within a focal plane at this time, nor is it possible to use the PAIR mode detectors (21–30).

If either NESR method is specified, then the NESR is computed separately for different spectral microwindows, depending on which focal plane the microwindow is in, and how many spectra were averaged for that focal plane.

Mean longitudes, latitudes, emission angles etc are computed by giving equal weight to each focal plane contributing to the final spectrum, regardless of actual numbers of spectra averaged in each window or the size of the microwindow.

Output files: A single output file: `cirs_microwin.spe` is produced, which contains all the average spectra concatenated one after another, preceded by the three-line header. This is a suitable format for input to the Oxford `Oxmultiret` code for example. Note that wavenumbers within a given spectrum will be discontinuous if more than one microwindow is specified.