Format, access and visualization of solar radio spectrograms

André Csillaghy
Institute of Astronomy, ETH-Zentrum,
CH-8092 Zurich, Switzerland*

June 6, 1998

The Zurich archive of solar radio spectrograms contains observations recorded by two spectrometers, Ikarus (Perrenoud, 1982) and Phoenix (Benz et al., 1991). The instruments—and the data access system— are operated by the Radio Astronomy Group (RAG) at ETH Zurich. Both instruments were able to observe up to 500 programmable channels, in the range 0.1 to 3 GHz, the world's largest bandwidth in this domain. An upgrade will eventually increase the frequency agility and the sensitivity of the instrument (Stehling, 1995).

Solar radio observations from these instruments are usually visualized with "images" that represent measured values in the frequency-time plane. In general, the frequencies observed are not equidistant, and a lookup table associates frequencies with rows of the image. Similarly, the recording time of a spectrogram may not be continuous and a lookup table associates observation times with columns of the image. When storing the image data in a file, the lookup tables must also be stored. The Flexible Image Transport System, or shortly FITS (Wells et al., 1981; Grosbøl et al., 1988; FITS, 1997) is able to store both the image and the lookup tables in the same file. Moreover, this format is machine-independent and can be read on any hardware or operating system.

FITS has the official denomination of standard (NOST, 1995). It is the common format for many major instruments, for instance the NASA-funded observation satellites, as the Hubble Space Telescope, SOHO, etc. It is used by a wide astronomical community, and hence basic read and write operations are supported by a number of public domain software tools (FITS, 1997) on usual operating systems. Section 1 describes the FITS structure designed for the management of solar radio data, named RAG-FITS. Section 2.1 describes where and how to access the data, and Section 3 describes a number of tools to read and write RAG-FITS. Note that this is not an exhaustive description

*now at Space Sciences Laboratory, University of California, Berkeley CA 94720-7450, csillag@ssl.berkeley.edu

of the FITS standard, which is described in the NOST draft standard (NOST, 1995).

1 Components of the format

FITS files associate an ASCII header of metadata with the actual data, that is, information about the stored data. In this way, the file contents can be deduced without information sources external to the file itself, such as instruments, methods, observers, data type, etc. This section describes the structure specific to RAG-FITS files.

1.1 Overall structure

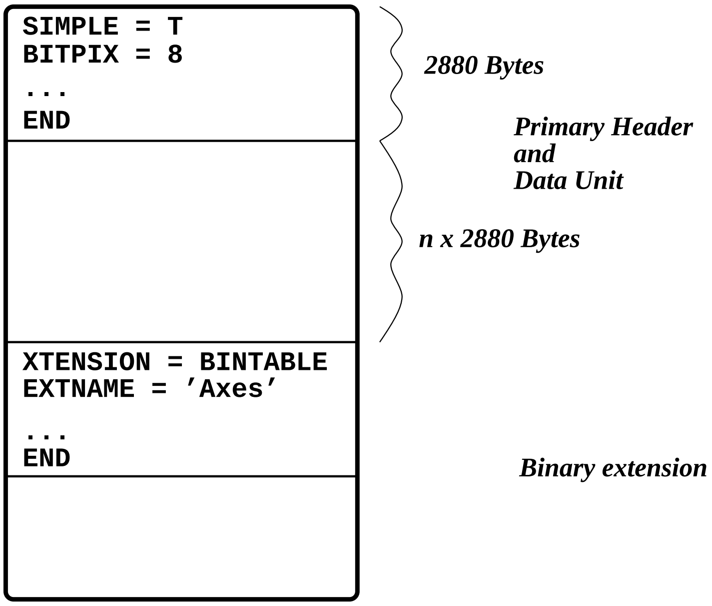

Figure 1: *The structure of a RAG-FITS file. Measurements are described in the primary header and stored in the primary data unit; the binary extension contains the lookup tables associating image coordinates with frequency and time information.*

The RAG-FITS structure is illustrated in Figure 1. It consists of the following components:

- A *primary Header and Data Unit* (HDU), consisting of an ASCII primary header and a data array:
  - The *primary header* gives general information about the spectrogram: number of bits per pixel, number of pixels in x- and y-direction, scaling factors, minimum and maximum values, observation date and time, frequency program used, etc.
  - The *primary data array* stores the actual image in form of a two-dimensional array. Its elements, or pixels, are written either with 8-, 16- or 32- bits, integers, or in 32-bits IEEE-standard real numbers.
- A *binary table extension* is optionally written if one or both of the axes have non-equidistant values. The binary table follows the standard defined by Cotton et al. (1995), and consists also of an ASCII header and a data unit.
  - The *binary table extension header*, named `Axes`, defines the scaling parameters, physical units, minimum and maximum values of both axes.
  - The *binary extension table data* stores the two lookup tables associated with the frequency and the time axes. The first table describes the frequency axis, whereby the i-th axis value corresponds to the frequency associated with the i-th row of the image. Similarly, the second table describes the time axis, whereby the j-th axis value of the axis corresponds to the time associated with the j-th column of the image.

1.2 RAG-FITS specific keywords

The primary header and the binary header of a RAG-FITS file contain *required* and *optional* keywords. The primary header and the binary extension are independent structures and therefore their corresponding keywords are presented separately hereafter. The default values given are those written by the PV-WAVE program `ragfitswrite` (see Section 3.1.1), and correspond to the values occurring in files of the RAG archive.

1.2.1 Primary Header

The keywords used in the primary array are defined in the standard FITS documents previously cited. No additional, site-specific keywords has been required for RAG-FITS. The information about axes given in the primary header allow to reconstruct straightforwardly regular axes.

`BITPIX` (required) means *bits per pixels* and takes the value 8 (byte), 16 (integer), 32 (long integers) or -32 (IEEE-standard reals). The default is determined by the data type of the input array, when the file is created.

BSCALE (required) is a scaling factor used to convert the internal data value to the actual values stored, with the formula

$$\mathsf{actual\ data} = \mathsf{data\ stored} \times \mathtt{BSCALE} + \mathtt{BZERO}. \tag{1}$$

BUNIT (optional) contains the physical units of the pixels. The default is `45*log10( SFU + 10 )`, the unit used by the Zurich spectrometers.

BZERO (required) is a scaling offset used to convert the internal data format to the actual values stored, using the formula (1).

CDELT1, CDELT2 (optional) are constants indicating the steps between measurements in frequency and time, respectively, if axes have equidistant values. If axis values are not equidistant, that is, if the AXES extension is present, the CDELTx keyword just indicate the difference between the first axis element and the second.

COMMENT (optional) serves as spares. It may also contain the information "TIME (or FREQ) AXIS is not regular" when the file has been produced by the program ragtofits(RAGa, 1997). It may also contain the HTML address of the frequency program used to record the data.

CONTENT (optional) specify the title of the spectrogram;

CRVAL1, CRVAL2 (optional) give the first value (at pixel 0) of the x- and y-axis, respectively.

CTYPE1, CTYPE2 (optional) give the titles of the x- and y-axes, respectively.

DATAMIN, DATAMAX (optional) give the lowest and highest array values of the spectrogram, respectively.

DATE-OBS, DATE-END (optional), written in the FITS standard format dd/mm/yy, specify the date of the data acquisition start and end, respectively. Note that the file naming convention, described Section 2.1.2, follows the inverse rule: the year appears first and the day last (yymmdd) in order to sort correctly the files.

END (*required*) signals the end of the keyword header. The primary array is stored beginning at the following record.

EXTEND (*required*) has the value T and tells the reading procedure that the file may contain extensions. Note that a primary array can be read even if the software used—such as the Unix program xv (Bradley, 1996)—cannot read the FITS extensions.

HISTORY (optional) contains information about (known) processes applied to the data before the file has been produced.

INSTRUME (optional) is the name of the spectrometer used for the observation: PHOENIX for data between 1985 and 1994 or IKARUS for older data.

NAXIS (*required*) is always 2 for a 2-dimensional image.

NAXIS1 (*required*) gives the number of pixels in x-direction.

NAXIS2 (*required*) gives the number of pixels in y-direction.

OBJECT (optional) is set to SUN;

ORIGIN (optional) is set to RAG ETH/Zurich;

SIMPLE (*required*) has always the required value T. Tells the file conforms to the basic FITS format.

TELESCOP (optional) is set to Bleien Radio.

TIME-OBS,TIME-END (optional) are written in the form hh:mm:ss.sss and give the start and end time of an observation, respectively. (hh designates the hours in 2-digit format, mm designates the minutes in 2-digit format and ss.sss designates the seconds with a 3-decimal precision.) It corresponds generally to the first value of the x-axis, although this is not compulsory.

1.2.2 Binary extension

If a spectrogram has non-regular axes, the latter must be stored, as previously mentioned, in a lookup table defined in a FITS binary extension (Cotton et al., 1995) that establishes the correspondence between image coordinates and real frequency or time values.

The presence of a non-regular axis can be noticed in two ways:

1. The COMMENT keyword in the primary header contains the text: "TIME AXIS is not regular" or "FREQ AXIS is not regular", or both.

2. The binary extension is not empty. In other words, if the binary extension is empty, the axes are regular.

In practice, generally only the frequency axis has non-equidistant values. To keep matters simple, however, both axes are written even if only one axis is not regular, because the memory loss for redundant information is insignificant with respect to the size of the primary array. A binary extension contains the following keywords:

BITPIX (required) is set to 8 for binary tables.

END (required) terminates the ASCII header, as in the primary header.

EXTNAME (required) Extension name: Axes in the case of RAG FITS.

GCOUNT (required) Group count, is set to 1 in binary tables.

NAXIS (required). This value is always 2 in binary tables.

NAXIS1 (required) Number of bytes per row; it is a function of the length of the axes. For instance, if an image has 1230 floating-point values on the x- and 279 floating-point values in the y-direction, then NAXIS1 = 6036. This number is computed with NAXIS1 $= 1230 \times 4 + 279 \times 4$, where 4 is the number of bytes used to represent a single floating-point value.

NAXIS2 (required) Number of rows: always 2, also required in binary tables.

PCOUNT $= 0$ (required) Random parameter count: it is set to 0 in binary tables.

TFIELDS (required) Number of columns in the table; in the Axes table, TFIELDS = 2, as there is one column per axis.

TDMIN1, TDMAX1 (optional) Minimum and maximum values of the frequency axis, in MHz.

TDMIN2,TDMAX2 (optional) Minimum and maximum values of the time axis. These are stored in seconds after start time, i.e. the value 0 corresponds to the start time stored in TIME-OBS.

TFORM1, TFORM2 (required) Format of the axes values (e.g. integer*4) as stored. The data type notation is defined in the NOST standard (NOST, 1995).

TSCAL1, TSCAL2 (required) Scale parameters to convert between the internal data format and the actual value stored. The formula is:

$$\text{axis read} = \text{axis stored} \times \mathtt{TSCALE}x + \mathtt{TZERO}x, x = 1, 2. \quad (2)$$

TTYPE1, TTYPE2 (optional) Axis labels. The defaults are Frequency and Time, respectively

TUNIT1, TUNIT2 (optional) Axis physical units. The defaults are MHz or s, respectively.

TZERO1, TZERO2 (required) Zero offsets to convert between the internal data format and the actual value stored with the formula (2).

XTENSION= BINTABLE required for binary extensions

1.3 Example of RAG-FITS headers

The following example shows the primary header and the binary extension header of a typical FITS observation.

```
SIMPLE  =                    T /Written by IDL:   3-Sep-1995 08:03:01.00
BITPIX  =                   16 /Integer*2 (short integer)
NAXIS   =                    2 /
NAXIS1  =                 6000 /
NAXIS2  =                  197 /
EXTEND  =                    T /File contains extensions
DATE    = '03/09/95'           /
CONTENT = '92 09 05 flux density' /Title of image
ORIGIN  = 'RAG / ETH Zurich'   /Organization name
TELESCOP= 'Bleien Radio'       /Location of the telescope
INSTRUME= 'Phoenix '           /Name of the spectrometer
OBJECT  = 'Sun     '           /
DATE-OBS= '05/09/92'           / observation start date
TIME-OBS= '11:24:47'           / observation start time
DATE-END= '05/09/92'           / observation end date
TIME-END= '11:34:47'           / observation end time
BZERO   =            90.831380 / scaling offset
BSCALE  =         0.0027719537 /scaling factor
BUNIT   = '45*log10(sfu+10)'   /z-axis title
DATAMIN =            0.0000000 /Minimum element in image
DATAMAX =            181.65999 /Maximum element in image
CRVAL1  =              0.00000 / value on axis 1 at the reference pixel
CRPIX1  =                    0 / reference pixel of axis 1
CTYPE1  = 'time in UT'         / title of axis 1
CDELT1  =             0.100000 / step between first and second elements in
axis"
CRVAL2  =              2974.00 / value on axis 2 at the reference pixel
CRPIX2  =                    0 / reference pixel of axis 2
CTYPE2  = 'frequency in MHz'   / title of axis 2
CDELT2  =             -14.0000 / step between first and second elements in
axis"
COMMENT WARNING: the value of CDELT1 may be rounded !
COMMENT The frequency axis is not regular
COMMENT
HISTORY calibration file:
http://mimas.ethz.ch/IKARUS-PHOENIX/freqency-programs/
                                                c92c9c5c3c11c45c.1.012.lis
END
------------------------------------------------------------------------
XTENSION= 'BINTABLE'           /Written by IDL:   3-Sep-1995 08:03:07.00
BITPIX  =                    8 /
NAXIS   =                    2 /Binary table
```

```
NAXIS1  =                24788 /Number of bytes per row
NAXIS2  =                    1 /Number of rows
PCOUNT  =                    0 /Random parameter count
GCOUNT  =                    1 /Group count
TFIELDS =                    2 /Number of columns
EXTNAME = 'Axes    '           /Extension name
TFORM1  = '6000J   '           /Integer*4 (long integer)
TTYPE1  = 'TIME    '           /Label for column 1
TUNIT1  = 's       '           /Units of column 1
TSCAL1  =       1.3967511e-07 /Scale parameter for column 1
TZERO1  =           299.95001 /Zero offset for column 1
TDMIN1  =             0.00000 /Minimum value in column 1
TDMAX1  =             599.900 /Maximum value in column 1
TFORM2  = '197J    '           /Integer*4 (long integer)
TTYPE2  = 'FREQUENCY'          /Label for column 2
TUNIT2  = 'MHz     '           /Units of column 2
TSCAL2  =       6.7427754e-07 /Scale parameter for column 2
TZERO2  =           1526.0000 /Zero offset for column 2
TDMIN2  =             78.0000 /Minimum value in column 2
TDMAX2  =             2974.00 /Maximum value in column 2
END
---------------------------------------------------------
```

2 Access to the data

Spectrograms of the Radio Astronomy Group are managed on a DEC Station 5000/200 running Ultrix and a DEC Alphastation running Digital Unix. Both workstations also operate a Worldwide Web server, allowing to reach remotely the RAG data (for the http addresses see below). Observations are stored in about 100,000 files and occupy about 10 GBytes of data in form of calibrated data, chronologically sorted. The way how RAG data are retrieved depends on the knowledge about the data and on the information searched. Four different approaches are described below.

- *Using the ASPECT image retrieval system.* This system, allows to formulate queries, to browse through document descriptions and to inspect single documents (see http://www.astro.phys.ethz.ch/rag/aspect/. Queries are specified in a query form (ASPECT, 1996). Query parameters are distributed into the following categories:
  - Direct queries: the file name is directly introduced.
  - Content-based queries: the images are searched on the basis of their information content, such as mean features flux, event bandwidth, etc.

- Metadata queries: the images are searched on the basis of preliminary information, such as date/time of observation and time of the day, or other technical parameters such as modes of the instrument, frequency programs etc.

- *Access to the catalog pages (RAGb, 1997).* This access is broader than the direct access to the retrieval system. It presents a full (hypertext) description of the archive, hypertext links and access to special collections. Moreover, it allows to access the index to the archive, where the directory structure of the archive can be traversed. However, it does not provide the functionality of the ASPECT system, although the latter can be reached from the catalog page.

- *Browse the data chronologically.* This method requires an authorization or an account on the machine `mimas.ethz.ch`. By selecting an appropriate directory, a specialized image visualization software, called `diasession` (RAGc, 1997) allows to browse directly through FITS files. However, this method is relatively slow because of the large size of the the original documents.

- *Access directly the document identification hypercard* (Csillaghy, 1997b) without using the query form of the ASPECT system. This possibility is attractive if the date and the start time of an observation is known in advance and the archived documents do not need to be browsed. The standard form of the hypercard address is

  `http://www.astro.phys.ethz.ch/rag/cgi-bin/fitsview?`*yymmddhhmmss*

  where *yy*, *mm*,*dd*, *hh*, *mm*, and *ss*, are the year, month, day, hour, minute and second of the start of the observation. The document is represented with an image icon displaying the data distribution of the original image. Depending on the authorization status of the user, the corresponding original file can be downloaded to the local machine. Alternatively, similar events can further be searched.

2.1 Structure of the RAG archive

The explanations given below are oriented to people with an account on the machine `mimas.ethz.ch` and assumes that the RAG environment is activated by appending to the login file `.login` the command

```
source /global/helene/local/rag/bin/raglogin.
```

The following environment is described only in terms of logical variables, to keep it independent of the physical location of the data.

Type of data	Observation mode
i	intensity
p	polarization
l	left circular polarization
r	right circular polarization
u	uncalibrated

Table 1: *The association between the characterization of the data type in the file name and the recording modes of the spectrometer*

2.1.1 Directories used in the archive

The root of the directory structure associated with the archive has the logical name, or environment variable, `$RAG_ARCHIVE`. On this file system, information about the archive of observations, data collections, instrument parameters etc. are stored. The subdirectory `$RAG_DATA`, is the root of the directory structure containing the original observations. Files are distributed among three levels of directories: the highest level corresponds to the observation year, the middle level corresponds to the observation month and the lowest level corresponds to the observation day. All have a two-digit format (i.e. 06 for the month of June). Depending on the number of files (limited to 500) in a single directory, a directory level can sometimes be avoided.

2.1.2 Filename convention

Each directory under `$RAG_DATA` contains (at least) three kinds of files: the original observation, the image icon and an enlarged image icon. They are labeled following a filename convention.

- *Original observations:* The filename of an observation has the form:

  $$\textit{filename} = \textit{YearMonthDayHourMinuteSecondX}\texttt{.fit.gz},$$

  where *Year, Month, Day, Hour, Minute, Second* are two-digits numbers designating the observation date and time, as for the directory names. The letter X denotes the type of data recorded and depends on the observing mode. The existing modes are summarized in Table 1. and are further described in another work (Csillaghy, 1997a). The first file name extension `fit` describes that it is a FITS file and the second, (`gz`), if present, that it is compressed. Therefore, the file name is a concatenation of numbers giving the start of the observation with a precision of the order of the second, for example: `830606082023i.fit.gz`.

- *Image icon:* The file name designating the image icons follows the same nomenclature except for the extension which is `icon.gif`:

YearMonthDayHourMinuteSecondX`.icon.gif`

The image icon summarizes in less than 1000 bytes the information content of the observation. It is usually visualized with the browser of ASPECT (Csillaghy, 1997b).

- *Enlarged image icon:* The file name designating the enlarged image icons follows the same nomenclature except for the extension which is `red.gif`:

  YearMonthDayHourMinuteSecondX`red.gif`

  The enlarged icon displays the image axes in addition to the information of the small image icon. It is usually displayed with the event identification hypercard.

3 Software to handle RAG-FITS files

A large number of procedures have been developed to manipulate FITS files, either for interactive languages such as IDL, or for programming languages such as FORTRAN. This section lists software available to read and write RAG-FITS. This list is however not exhaustive; such a list can be found in other works (FITS, 1997).

3.1 IDL Software

The interactive data language IDL (1995) is commonly used by the astronomical community to process images, and a number of routines have been developed to read and write FITS files in this graphics-oriented languages. Some procedures are also available for the similar interactive language PV-WAVE (PV-WAVE, 1992).

3.1.1 The RAG-FITS read/write standalone routines

Two IDL procedures are designed to read and write RAG-FITS files:

`RAGFitsRead,` *fileName, image, xAxis, yAxis* and
`RAGFitsWrite, FILENAME =`*fileName, image, xAxis, yAxis*

where *fileName* is a character string containing the full name of the file (with the name of the extension), *image* is the variable containing the 2-dimensional array data to read or write, and *xAxis, yAxis* are the variables for the axes, in 1-dimensional arrays. A number of options is available to read parts of the information (RAGd, 1997). The axis values are stored in double-precision reals.

The procedures described above are available in the Solar Software Tree environment (see Section 3.1.3 below).

3.1.2 The IDL FITS library

The FITS support office (FITS, 1997) in collaboration with the NASA Goddard Space Flight Center (GSFC) provides an IDL library containing a large collection of procedures to handle FITS files. These procedures are adapted for reading or writing specific parts of the RAG-FITS file. In particular, the following procedures are of interest:

- `FXREAD`, `FXWRITE` reads or writes, respectively, a primary header and data unit (HDU);
- `FXBREAD`, `FXBWRITE` reads or writes, respectively, a binary extension;
- `FXPAR`, obtains the value of a specific parameter in a FITS header.

With the procedures described above, RAG-FITS files can be read without prior knowledge of the internal structure. It is sufficient to know the names of the keywords in the headers, for instance by printing the header wit the command `headlist` (FITS, 1997), and use `FXREAD` and `FXBREAD` to read the data.

3.1.3 The ragview data analysis software

The Ragview software is available in the solar software tree (SSW) distribution (`http://www.lmsal.com/solarsoft/ssw_whatitis.html`). In the tree, it resides under the path `$SSW/radio/ethz`. To use it, you must either install the ssw on your system or upgrade it if you have already installed ssw.

To install the ssw, follow the instructions given at

```
http://www.lmsal.com/solarsoft/ssw_install_howto.html.
```

Make sure to select in the installation form

```
http://www.lmsal.com/solarsoft/ssw_install.html
```

the box "ETHZ", which will include in your site-specific tree the Ragview software branch. Once the installation script is generated, let it run on your system.

To upgrade an existing ssw installation, follow the instructions given at

```
http://www.lmsal.com/solarsoft/ssw_upgrades.html.
```

In the interactive upgrade, you can load the Ragview software by typing

```
ssw_upgrade, /ethz, /spawn.
```

Make sure that you have sufficient privilege to execute the operation or contact your local system administrator.

Once the ssw tree is installed, you will have to set the correct SSW environment variables and IDL setup. This is described at http://www.lmsal.com/solarsoft/ssw_setup.html.

Make sure to include the ETHZ "instrument" in the list of instruments given by the environment variable $SSW_INSTR. You may need to start a new session or source your .login or .cshrc file to make the changes effective. Finally, type sswidl to start idl under ssw environment and in IDL, type ragview to start the user interface.

3.2 Other software

3.2.1 The FITSIO library

The FITSIO library, has been developed to read and write FITS files from FORTRAN and C programs. This library is described by Pence (1992) and can be extended with specific RAG-FITS reading and writing FORTRAN subroutines, called RAGFITSREAD and RAGFITSWRITE, respectively (RAGe, 1997).

References

ASPECT: 1996, *The Ikarus/Phoenix Image Retrieval System ASPECT*, ETH Zurich,
www.astro.phys.ethz.ch/rag/aspect

Benz, A. O., Güdel, M., Isliker, H., Miszkowicz, S., and Stehling, W.: 1991, *Sol. Phys.* **133**, 385,
www.astro.phys.ethz.ch/papers/benz/list.html

Bradley, J.: 1996, *xv*, Sun Microsystems, Inc., Mountain View, Ca.,
http://www.sun.com/sunsoft/catlink/xv/xv.html

Cotton, W. D., Tody, D., and Pence, W. D.: 1995, *A&AS* **113**, 159

Csillaghy, A.: 1997a, *Proc. of the 5th Int. Workshop on Data Analysis in Astronomy*, in press,
www.astro.phys.ethz.ch/papers/csillaghy/erice96.ps.gz

Csillaghy, A.: 1997b, *Ph.D. thesis*, Swiss Federal Institute of Technology, Zurich

FITS: 1997, *FITS Basics and Information*, National Space Science Data Center (NSSDC) / NASA / Goddard Space Flight Center (GSFC),
http://www.nasa.gov/astro/fits/basics_info.html

Grosbøl, P., Arten, R. H., Greisen, E. W., and Wells, D. C.: 1988, *A&AS* **73**, 359

IDL: 1995, *Interactive Data Language User's Guide*, Research Systems, Inc., Boulder, Version 4

NOST: 1995, *Definition of the Flexible Image Transport System*, NASA/Science Office of Standards and Technology

Pence, W. D.: 1992, *FITSIO - A Subroutine Interface to FITS Format Files*, USRA/HEASARC

Perrenoud, M.: 1982, *Sol. Phys.* **81**, 197

PV-WAVE: 1992, *Command Language User's Guide*, Precision Visuals, Inc., Boulder, Version 4.0
RAGa: 1997, *RAGTOFITS usage instructions*, ETH Zurich, http://mimas.ethz.ch/cgi-bin/wave-man?command=ragtofits
RAGb: 1997, *The RAG catalogs*, ETH Zurich, http://mimas.ethz.ch/catalogs.html
RAGc: 1997, *DIASESSION documentation*, ETH Zurich, http://mimas.ethz.ch/cgi-bin/wave-man?command=diasession
RAGd: 1997, *RAGFITSREAD documentation*, ETH Zurich, http://mimas.ethz.ch/cgi-bin/wave-man?command=ragfitsread
RAGe: 1997, *RAGFITSWRITE – Fortran subroutine*, ETH Zurich, ftp://mimas.ethz.ch/pub/software/single-routines/ragfitswrite.for
Stehling, W.: 1995, *SPIE* **2479**, 89
Wells, D. C., Greisen, E. W., and Harten, R. H.: 1981, *A&AS* **44**, 363