OGIP Calibration Memo CAL/SW/93-011

CALCRPSF USERS GUIDE

Rehana Yusaf Ian M George

& Gail Rohrbach

HEASARC
Codes 664 & 668,
NASA/GSFC,
Greenbelt, MD20771

Version: 1995 Apr 20

SUMMARY

`calcrpsf` is a task within the `caltools` sub-package of `ftools` which provides a number of functions associated with the generation and manipulation of Point Spread Function datasets. Here we describe this functionality.
This document is up-to-date for the `ftools` v3.3 Public Release (`calcrpsf` version 1.3.1).

LOG OF SIGNIFICANT CHANGES

Release Date	**Sections Changed**	**Brief Notes**
1994 Mar 10		Original Version
1995 Jan 29	All	Made compatible with `LaTeX2HTML` s/w
1995 Feb 06	All	Improved user-friendliness
1995 Apr 21	3.3	Section added

Contents

1 INTRODUCTION

`calcrpsf` is essentially a multi-task wrapper, or FORTRAN 'script', to spawn a number of `ftools` associated with the generation and manipulation of Point Spread Function (*psf*) datasets. All the `ftools` spawned by `calcrpsf` are of course able to be run independently (*i.e.* outside `calcrpsf`), however the task was written in order to minimize the number of tasks users must run (and hence reduce the associated book-keeping) in order to perform certain common types of analysis on *psf* datasets.

At the current time its functionality is limited to the generation and manipulation of 1-dimensional, Radial *psf* (RPSF) datasets. However a number of future enhancements are planned to deal with 2-dimensional *psf* (image), and Encircled Energy Function datasets.

1.1 Overview

`calcrpsf` provides the following functionality:

- The conversion of an observational RPSF dataset exported from IRAF/PROS to the OGIP format for such a dataset
  (via the `st2rpsf` task in the `ftools/caltools` package)
- Rebining of an observational RPSF dataset
  (via the `rbnrpsf` task in the `ftools/caltools` package)
- Generation of a theoretical RPSF dataset
  (see Section 4.3)
- Conversion of an RPSF dataset to an ASCII file with embedded QDP commands
  (via the `rpsfqdp` task in the `ftools/caltools` package)

Users have the facility to perform some or all the above steps within a single `calcrpsf` run.

1.2 Supported FITS file formats

All tasks read and write FITS files which adhere to the format given in the OGIP Calibration Memo CAL/GEN/92-020, with the exception of the format conversion tasks `st2rpsf` (input in non-OGIP format) and `rpsfqdp` (output in non-OGIP format).

The CAL/GEN/92-020 document is available on-line:

- via the WWW with links from the URL:
  `/docs/heasarc/caldb/caldb_doc.html`
  to both postscript and HTML formats.
- in postscript on the `legacy.gsfc.nasa.gov` via anonymous ftp or gopher as
  `caldb/docs/memos/cal_gen_92_020/cal_gen_92_020.ps`

2 INPUT PARAMETERS & OPERATION

2.1 Input Parameters

In the light of the functionality described in Section 1.1, `calcrpsf` has the following input parameters:

- infil [character string]
  The name of the input file. The file required obviously depends upon which tasks are to be spawned as detailed below. Users who simply wish to generate a theoretical profile should enter `NONE`.

- outfil [character string]
  The name of the final output file. The task which produces the output file obviously depends upon which is the last task to be spawned (as detailed below).

- qst2rpsf [logical]
  Logical flag indicating whether the task `st2rpsf` is to be spawned by `calcrpsf`.

  - If yes, then the value of the infil parameter above should be the name of the FITS file produced by running the `imcnts` task in the `PROS/xspatial` sub-package of `IRAF`, followed by the task `stwfits` in the `stsdas/fitsio` sub-package of `IRAF` (see Section 5 for further details).
  - If no other tasks are to be spawned from within `calcrpsf`, then the value of the outfil parameter should be the name of the desired o/p file. (If other tasks are spawned, then intermediate files will be produced & removed whilst `calcrpsf` is executing; see Section 2.2).

- qrbnrpsf [logical]
  Logical flag indicating whether the task `rbnrpsf` is to be spawned by `calcrpsf`.

  - If yes, but `st2rpsf` has **NOT** been run, then the value of the infil parameter above should be the name of the FITS file (in OGIP-standard format) to be rebinned.
  - If no other tasks are to be spawned after `rbnrpsf` from within `calcrpsf`, then the value of the outfil parameter should be the name of the desired o/p file. (If other tasks are spawned, then intermediate files will be produced & removed whilst `calcrpsf` is executing; see Section 2.2).

- qpred [character string (pseudo-logical)]
  A character string indicating whether one of the available tasks to generate a theoretical RPSF dataset is to be spawned by `calcrpsf` (see Section 4.3).

  - If yes, but **NEITHER** `st2rpsf` nor `rbnrpsf` have been run, then the value of the infil parameter above should either be the name of the FITS file to which the theoretical RPSF dataset is to be appended, or `NONE` indicating a new file is to be created.

  - If rpsfqdp is **NOT** to be spawned afterwards from within `calcrpsf`, then the value of the outfil parameter should be the name of the desired o/p file. (If other tasks are spawned, then intermediate files will be produced & removed whilst `calcrpsf` is executing; see Section 2.2).

  The special character `?` displays a list of instruments for which theoretical generators are available.

- qrpsfqdp [logical]
  Logical flag indicating whether the task `rpsfqdp` is to be spawned by `calcrpsf`.
  - If yes, but **NEITHER** of the tasks `st2rpsf`, `rbnrpsf` has been run, **AND** a theoretical RPSF has **NOT** been generated, then the value of the infil parameter above should be the name of the i/p RPSF dataset to be converted to ASCII.
  - If no then the value of the outfil parameter should be the name of the desired o/p file.

- telescop [character string]
  The name of the mission/satellite (*e.g.* `ROSAT`) carrying the instrument for which the theoretical RPSF dataset is to be modelled. This parameter is only required by `calcrpsf` if infil = `NONE`, and qpred = `yes`.

- instrume [character string]
  The name of the instrument (*e.g.* `PSPCB`, `HRI`, *etc*) for which the theoretical RPSF dataset is to be modelled. This parameter is only required by `calcrpsf` if infil = `NONE`, and qpred = `yes`.

- chatter [integer] (hidden)
  Integer flag to indicate how chatty `calcrpsf` is at execution. A value of 9 is the default, with lower/higher values producing quieter/verbose output respectively. Users who discover bugs within `calcrpsf` are encouraged to send a transcript of their `calcrpsf` run at high chatter flag along with the bug report. This significantly speeds up locating the problem.

- schatter [integer] (hidden)
  Integer flag to indicate how chatty the tasks spawned by `calcrpsf` are be during their execution. A value of 5 is the default, with lower/higher values producing quieter/verbose output respectively. Users who discover bugs within the tasks spawned by `calcrpsf` are encouraged to send a transcript of their `calcrpsf` run with high values of the chatter & schatter flags along with the bug report. This significantly speeds up locating the problem.

2.2 Temporary Files

The following temporary files can be generated in the local directory by `calcrpsf` during its execution. Which files are actually created by a given `calcrpsf` run obviously depends on which combination of tasks are spawned:

- `st2rpsf.tmp` (generated by `st2rpsf`)
- `rbnrpsf.tmp` (generated by `rbnrpsf`)
- `rpsfpred.tmp` (generated by any of the RPSF generators listed in Section 4.3)

All these files should be automatically removed on successful execution. However under certain circumstances (for instance if `calcrpsf` crashes) this might not be the case, and the files can/should be removed manually.

2.3 Warnings on Use

Users should be aware that many of the `ftools` spawned by `calcrpsf` have hidden parameters (see Section 4). These parameters enable users some flexibility in 'customizing' their output, and in some cases provide added functionality. However in order to minimize the number of parameters users need to define prior to execution of `calcrpsf`, `calcrpsf` itself does not have the full functionality that is provided were all the spawned tasks are executed independently (*ie* by not running `calcrpsf`). Furthermore, since `calcrpsf` simply spawns the `ftools`, all hidden parameters will be used. This has the disadvantage that users may have occasionally set hidden parameters within the parameter files of the spawned task (*eg* in a previous execution) and get unexpected results, but the advantage that users are able to perform some degree of customization by manually changing the hidden parameters within the parameter file of the spawned tasks. Users should therefore ensure that they understand which hidden parameters are set within the parameter files of the spawned tasks.

Due to the generation of temporary files (Section 2.2), `calcrpsf` must be run from a local directory in which the user has write privilege.

Since this task spawns other `ftools`, it is crucial that users have their account properly set up. Specifically, users must have the path to their local copy of the `ftools` executables, and the environment variables pointing to the local system & user copies of the parameter files defined within the set-up files executed by such spawned jobs. For example, users running the c-shell on unix/ultrix platforms must have the above defined within their `.cshrc` file (**NOT** their `.login` file). Users who use the `ftools` initialization procedures recommended by a given `ftools` release should experience no difficulties. Those that do not are on their own.

3 EXAMPLES

In this section we provide sample `calcrpsf` sessions which demonstrate the major components of its operation. User input is in **bold**, and comments in *italics*.

3.1 Spawning all the tasks

system prompt > **calcrpsf**
`** CALCRPSF 1.3.0`
... calcrpsf parameters:
`Input filename ():` **myfile.tab**
`Output filename ():` **myfile.out**
`Run st2rspf ? (yes):`
`Run rbnrpsf ? (no): {\bf yes}`
`Generate a theoretical RPSF dataset ? (yes):`
`Run rpsfqdp ? (yes):`
... Note that the default values of the last four parameters were 'learnt' from the last time this user ran the task calcrpsf. Above, three of the four default values were acceptable, and hence the user simply hit return
... end of calcrpsf parameters:
`*** spawning ST2RPSF to convert o/p from STWFITS:`
`st2rpsf chatter=5 infil="myfile.tab" outfil=st2rpsf.tmp`
... the above is the expression spawned to run st2rpsf

```
       PROGRAM ST2RPSF
       ---------------

   STW format PSF -> OGIP format PSF
```

... additional parameters required for st2rpsf
`Please enter Telescope name ():`**ROSAT**
`Please enter Instrument name ():`**PSPCB**
`Please enter Minimum PI channel for image (10):`
`Please enter Maximum PI channel for image (30):`
`Please enter background count rate (ct/pixel) (0.00015):`**0.0002**
... Note that the default values of the last three parameters were 'learnt' from the last time this user ran the task st2rpsf. Above two of the three default values were acceptable, and hence the user simply hit return
`Main ST2RPSF Ver 1.0.5`
`PROGRAM ST2RPSF Ver 1.0.5 COMPLETED`
... control is now returned to calcrpsf in preparation of spawning rbnrpsf

`*** spawning RBNRPSF to convert rebin RPSF dataset:`
`rbnrpsf chatter=5 infil="st2rpsf.tmp" outfil=rbnrpsf.tmp`

... again, above is the spawned expression. Note that both the i/p and o/p files are temporary
... one additional parameter is required for rbnrpsf:
`Please enter minimum No. counts/bin : (100):` **20**
... the user wishes to reduce the minimum number from that entered (and 'leant') last time the user ran rbnrpsf

```
Main RBNRPSF Ver 1.1.1
Program RBNRPSF completed
```

... control is now returned to calcrpsf in preparation for spawning the theoretical RPSF generator. The values entered above concerning the mission/instrument are written to all temporary files created thus far, hence calcrpsf is able to automatically determine which generator to use. Here it is the ROSAT PSPC

```
 *** spawning PCRPSF to generate prediced RPSF dataset:
pcrpsf chatter=5 infile="rbnrpsf.tmp" outfile=rpsfpred.tmp
```

... above is the command string spawned for pcrpsf, and again both the i/p and o/p files are temporary
... four additional parameters are required for RPSF generator for the ROSAT PSPC:
`Please enter PHA filename : ():`**pspc.pha**
`Please enter Off Axis histogram filename : ():` **pspc.pha**
`Please enter RMF filename : (pspcb_jan12.rmf):`
`Please enter background count rate (ct/pixel) : (): %`
... above, the user has specified that the off-axis histogram is in the same file (but a different extension) as the PHA dataset. Also the user specified that the background count rate to be used by the theoretical RPSF generator should be read from the i/p file to pcrpsf (i.e. rbnrpsf.tmp)

```
 Main PCRPSF Ver 1.0.6
 Program PCRPSF completed
```

... control is now returned to calcrpsf in preparation for spawning rpsfqdp

```
*** spawning RPSFQDP to convert RPSF dataset to QDP:
rpsfqdp chatter=5 datafile="rpsfpred.tmp" outfile=myfile.out
       PROGRAM RPSFQDP
       ---------------

   FITS format PSF -> QDP format PSF
MAIN RPSFQDP Ver 1.0.3
RPSFQDP COMPLETED
```

... at this stage, all the temporary files are being removed

```
** CALCRPSF 1.3.0   Finished
```

system prompt >

3.2 Generating a theoretical dataset only

system prompt > **calcrpsf**
`** CALCRPSF 1.3.0`
`Input filename (myfile.tab):`**NONE**
`Output filename (myfile.out):`**!myfile.out**
... Note that the user is preceeding the desired o/p filename with an ! in order to force any pre-existing file named myfile.out to be removed
`Generate a theoretical RPSF dataset ? (yes):` **?**
... User wants to first querry what is available

```
  ... PSF datasets can be generated for the following instruments:
      ROSAT HRI        (using hrirpsf)
      ROSAT PSPC       (using pspcrpsf)
```

`Generate a theoretical RPSF dataset ? (?):`**yes**
... User is now prompted for mission/instrument
`Telescope (ROSAT):`
`Instrument (PSPCB):`**HRI**
`Run rpsfqdp ? (yes):`**n**

```
 *** spawning HRIRPSF to generate prediced RPSF dataset:
hrirpsf chatter=5 infile="NONE" outfile=myfile.out
```

... this is the spawned string
... etc ...
... etc ...

3.3 How to make "pretty" QDP plots for extended sources

The "RPSF s/w suite of FTOOLS" (`st2rpsf`, `rbnrpsf`, `pcrpsf`, `hrirpsf` ... *etc* ...) is currently designed to enable users only to check and demonstrate whether the radial profiles of their sources are (or are not) consistent with that expected from a point source. (They are **NOT** designed to allow a detailed deconvolution of the observed vs true radial profiles from an extended source). Thus, by default the tasks which generate the theoretical profiles (`pcrpsf`, `hrirpsf`... *etc* ...) make an implicit assumption that the observational dataset is indeed from a point source and calculate the appropriate profile based upon the number source counts (having subtracted any background the user or i/p files have specified).

Unfortunately this has the consequence that in cases when the source is indeed extended, the theoretical curve will often over-shoot the observed dataset in the innermost regions. Of course, this is exactly what one would expect from the above arguments. However a number of users have quite rightly asked why they cant use the RPSF s/w suite to produce *"pretty"* QDP plots to demonstrate the *"excess"* extended emission where the observed and theoretical profiles agree in the innermost radial bins, and the data sits above the model at radii appropriate to the extended emission.

This is possible (though a little painful), and this recipe expains how. The basic "trick" is to appropriately rescale the total number of **SOURCE** photons the theoretical PSF generator uses.

STEP-1: Run `st2rpsf`

It is highly recommended that `st2rpsf` version 1.0.7 or higher is run and in "Calculate bkgd mode" by setting the `bkgd` parameter to `C` (or answering the prompt appropriately).

system prompt > **st2rpsf**
`Please enter FITS STW Filename[rho1_cnt.fits]` **my.file**
`Please enter Telescope name[ROSAT]`
`Please enter Instrument name[PSPCB]`
`Please enter Minimum PI channel for image[]` **12**
`Please enter Maximum PI channel for image[]` **200**
`Please enter background count rate (ct/pixel)[]` **C**
`Please enter inner radius for bkgd calc (arcmins)[]` **3**
`Please enter RPSF FITS output filename[]` **st2rpsf.out**
`Main ST2RPSF Ver 1.0.7`
`Calculated bkgd value :         0.007110285`
`WARNING: Actual sum of pixels is < 90% of sum of pixels calculated`
`using area of of circle, where the outer radius is used`
`NOTE: This may be due to regions being excluded`
`PROGRAM ST2RPSF Ver 1.0.7 COMPLETED`
system prompt >

STEP-2: Note a few of the values we will need below

A bunch of values are calculated and written as keywords to the o/p file by `st2rpsf` which we will need in STEP-6 below. These can be noted by running either of the FTOOLS `fdump` or `fkeyprint`. The crucial keywords (and their values in this example) are:

- `SUMRCTS` `1.9624E4`
- `BACKGRND` `7.11028464E-03`
- `PIXSIZE` `1.38932315E-04` (in DEGREES ... see below)

Below we use the `fkeyprint` on `SUMRCTS` as a demo:

system prompt > **fkeyprint**
`Name of FITS file and [ext#][rho1_cnt.fits]` **st2rpsf.out[1]**
`Enter the keyname (8 characters or less)[sumrcts]` **SUMRCTS**
`# FILE: st2rpsf.out`
`# KEYNAME:` **SUMRCTS**

`# EXTENSION:    1`
`SUMRCTS = 1.96240000E04`
system prompt >

STEP-3: Running `rbnrpsf`

The observational dataset (from `st2rpsf`) can be rebinned if desired using `rbnrpsf`. Since this step has no impact on the recipe, its ignored here.

STEP-4: Generating a theoretical PSF dataset

The "first attempt" theoretical dataset can be generated in the normal way using the appropriate task (`pcrpsf` for the ROSAT PSPC, `hrirpsf` for the ROSAT HRI, ... *etc* ...). Later (STEP-6) we'll generate different "customized" theorertical profile based upon the discrepancy between this one and the observational dataset. Using the ROSAT PSPC:
system prompt > **pcrpsf**
`Please enter Radial Profile File[none]` **st2rpsf.out**
`Please enter PHA filename :[rho3_obs.pha]`
`Please enter Off Axis histogram filename :[%]`
`Please enter RMF filename :[../../pspcb_gain2_256.rmf]`
`Please enter Output filename :[]` **pcrpsf.test1**
`Please enter background count rate (ct/pixel) :[%]`
`Main PCRPSF Ver 2.1.0`
` Program PCRPSF completed`
system prompt >
Remember when running in this mode (*ie* when an i/p file has been specified), the theoretical curve and observational dataset are both present as different extensions of the output file (`pcrpsf.test1` in the above example).

STEP-5: Make a QDP file of observed & "first attempt" theoretical curves

In the standard way, combine the observational and datasets into a QDP file:
system prompt > **rpsfqdp**
... etc, etc ..
`Please enter FITS Radial Profile Filename :[]` **pcrpsf.test1**
`Please enter QDP output filename :[]` **rpsfqdp.test1**
`MAIN RPSFQDP Ver 1.0.5`
`RPSFQDP COMPLETED`
system prompt >
How we look at the plot and determine by what factor (if any) we wish to rescale the normalization of the theoretical curve in order to make it "pretty".

STEP-6: Generating a "customized" theoretical PSF dataset

Now we make a new "customized/rescaled" version of the theoretical dataset. This is achieved by "pretending" not to having an observational dataset at all ... thus entering all necessary paramters by hand.
system prompt > **pcrpsf**
`Please enter Radial Profile File[none]`
`Please enter PHA filename :[rho3_obs.pha]`

```
Please enter Off Axis histogram filename :[%]
Please enter RMF filename :[../../pspcb_gain2_256.rmf]
Please enter Output filename :[] pcrpsf.test2
Please enter background count rate (ct/pixel) :[%] 7.11028464E-03
Enter Sum of source counts under curve :[1.9624E3]
Please enter pixel size (arcmins/pixel):[1.38932315E-04] 8.33593e-3
Please enter minimum energy boundary (keV) :[.12]
Please enter maximum energy boundary (keV) :[2.0]
Main PCRPSF Ver 2.1.0
 Program PCRPSF completed
system prompt >
```

The only things to note here are:

1. `NONE` should be enterd as the i/p filename
2. the backgrd count rate is from the `BACKGRND` keyword above
3. the pixel size is from the `PIXSIZE` keyword above, BUT MUST BE ENTERED in arcmin NOT degrees (sorry !)
4. the Sum of source counts is from the `SUMRCTS` keywords above, BUT MULTIPLIED BY THE RESCALING FACTOR determined from STEP-5 (a factor of 0.1 is assumed here)

Given we did not specify an i/p file, the o/p file from this run (`pcrpsf.test2`) ONLY contains the theoretical curve.

STEP-7: Make a QDP file of observed & "customized" theoretical curves

Now we combine and plot our observational dataset with our new customized theoretical curve, using `rpsfqdp`. Remember, at this point:

- `pcrpsf.test1` contains the observed data and "bad" theoretical curve
- `pcrpsf.test2` contains the "customized" theoretical curve (only)

thus we can specify the theoretical curve to be used is that from `pcrpsf.test2` using the hidden parameter `predfile`:
system prompt > **rpsfqdp predfile=pcrpsf.test2**
... etc, etc ..

```
Please enter FITS Radial Profile Filename :[] pcrpsf.test1
Please enter QDP output filename :[] rpsfqdp.test2
MAIN RPSFQDP Ver 1.0.5
RPSFQDP COMPLETED
```

system prompt >

Now the theoretical curve in `rpsfqdp.test2` should have been rescaled compared to that in the previous QDP file.

STEP-8: Iterate

So now one simply has to iterate STEP-6 and STEP-7 until the desired degree of prettiness is achived.

4 ADDITIONAL NOTES ON SPAWNED TASKS

In this section we list the important hidden parameters of the tasks spawned by `calcrpsf`, along with any warnings on their use.

4.1 st2rpsf - To convert the RPSF o/p from stwfits to an OGIP-standard format

`st2rpsf` reads i/p data from a FITS file produced by the `IRAF` task `stwfits` (within the `stsdas/fitsio` sub-package), assumed to contain a 1-dimensional radial profile of an image, and writes an o/p FITS data file in OGIP standard format for radial profiles. The data is written in the form of a BINTABLE in the first extension of the o/p file.

Hidden parameters to st2rpsf

None

Warnings on usage

- It should be noted that the FITS extension created by `stwfits` containing the dataset **MUST** contain an annular source region descriptor **expressed in arcseconds**. More specifically, an extension containing (say) a 30 bin radial profile between 0 and 10 arcminutes from a source at coordinates (123,456) requires the presence of the FITS keyword `SOU_A`, the value of which must adhere to the following format:

  – `SOU_A   = 'ANNULUS 123. 456. 0." 600." n=30`

  This enables physical units to be assigned to the radial bins as required by OGIP standards. Note that the coordinates of the source are not used by `st2rpsf` and hence can be expressed in any coordinate scheme (eg image pixels, RA & dec etc).
  The above mandatory information required within the i/p file to `st2rpsf` can be obtained in a straightforward way by specifying the source region descriptor (parameter) for imcnts with the inner & outer radii specified in arcseconds. Thus in the above example, the region parameter to `imcnts` should be in the form:

  – `region = "a 123 456 0" 600" n=30"`

  where the coordinates (123,456) are in either image pixels or RA & dec.
  Given these inputs, the default values for the `stwfits` parameters can be used (see also Section 5).

4.2 rbnrpsf - To rebin an RPSF dataset

Often, it is desireable to rebin a radial *psf* dataset such that each new radial bin contains a minimum number of 'source' counts. This can be achieved using the task `rbnrpsf`.

As input this task takes the names of the input & output files, along with the minimum number of source counts required in each new radial bin. The task assumes that the input dataset also contains a background component (which is constant with radial bin, but may be zero). By default the task will read the background level from the `BACKGRND` keyword within the i/p file. However users may override this value upon execution by explicitly entering their own background level via a hidden parameter.

Hidden parameters to rbnrpsf

- bkgd [real] (hidden)
  Background count rate (in counts per pixel), should the user wish to override the value written in the `BACKGRND` keyword in the i/p file. The special/default value (`%`) indicates that the value from the i/p data file should be used.

Warnings on usage

- If a radial profile has been background subtracted, with an over-estimate for the background rate, negative counts can be present in the input dataset. It is important to note that when `rbnrpsf` performs the rebinning, these negative values are (negatively) included within the summation to determine whether the requested number of minimum counts are in the new bin. This often results in wide output bins.
- In order to obtain an optimum resolution on output from `rbnrpsf` (especially close to the putative 'peak' at small radii), it is preferable if the input dataset is somewhat 'over-resolved' in the radial direction.
  For example, in the case of *ROSAT* PSPC data extracted using the `imcnts` task within the `xray/xspatial` sub-package of `IRAF` over a range of 0–600 arcsec, at least 80 radial bins are recommended.

4.3 Theoretical Radial PSF generators

A separate `ftool` exists for each mission/instrument combination, with the requirements & options somewhat dependent upon the complexity of the instrument. A quick-reference table of theoretical RPSF generators currently available/planned is given in Table 4.3, and in the following sections we give a brief description of the inputs required for the available `ftools`, listing any points which should be kept in mind during their use. More detailed descriptions of the characteristics of the various instruments can be found elsewhere (and are noted below).

Table 1: Theoretical Radial PSF generators currently available/planned

Mission	**Instrument**	**Task Name Name**	**FTOOLS subpackage**	**Section**
ROSAT	HRI	`hrirpsf`	`rosat`	4.3.1
	PSPC	`pcrpsf`	`rosat`	4.3.2

4.3.1 ROSAT HRI (hrirpsf)

`hrirpsf` generates a theoretical radial (1-dimensional) point spread function dataset for the ROSAT HRI, and writes the results to an output FITS file (in OGIP-standard format for RPSFs).

A number of options are available, controlled via user-defined parameters, such that the task hopefully serves the needs of users interested in generating predicted profiles for direct comparision with observational datasets, and users who simply require a theoretical profile. Besides being able to specify the inner & outer radii, and number of steps used to calculate the theoretical RPSF, the following functionality is also availble:

- Users may define an input FITS file (via the parameter infile) containing an observed radial profile dataset (in OGIP-standard format). If such a dataset is entered, then a number of variables required for the construction of the theoretical RPSF dataset are (or can be) read from this FITS extension, enabling a direct comparison between the observed and theoretical profiles (see below).
- The PSF of the ROSAT HRI is a function of off-axis angle, hence the angle is required in order to calculate the correct theoretical curve. A 'nodding' of the spacecraft was performed for most (if not all) 'pointed' HRI observations, and an off-axis histogram is stored within the 'DETECTOR' extension of PHA files conforming to OGIP standards for the HRI. Such a histogram can be specified using the detfil parameter within `hrirpsf`. If such a histogram is not available/required, an off-axis angle can be specified directly via the parameter off_ang.
- Users may specify the strength of the background (via the parameter bkgd in units of counts per pixel) which will be scaled and added appropriately to the theoretical profile generated. If an observational dataset has been entered via the parameter infile, then the specification of this parameter will override the value stored within the dataset.

The algorithm currently in use was supplied by Larry David (CfA) in 1993 Nov. However, in tests performed by Jane Turner (NASA/GSFC), the model was found to sometimes give a less than perfect representation of an observational dataset. Often, this is the result of variations in the quality of the aspect solution of the observational data. It is therefore advised that the use of this model be limited to performing relatively crude tests as to whether a source is extended and/or performing corrections to count rates derived from extraction cells smaller than the RPSF. It is **NOT** recommended that this model be used for detailed deconvolution of extended sources.

Hidden parameters to hrirpsf

- rad_min [real] (hidden)
  Inner radius for the predicted model RPSF dataset in arcmin. The default value is 0.0 and appropriate for most applications.
- rad_max [real] (hidden)
  Outer radius for the predicted model RPSF dataset in arcmin. The default value is 10.0 and appropriate for most applications.
- n_rad [integer] (hidden)
  The number of radial bins to be used in generating the RPSF dataset. The default value is 300. For most applications it is suggested that this be a large number (at least 100) in order to obtain sufficient resolution to see the structure of the theoretical RPSF of the HRI. The number of bins does NOT have to be the same as the number of bins in any observational dataset entered via infile.

4.3.2 ROSAT PSPC (pcrpsf)

`pcrpsf` generates a theoretical radial (1-dimensional) point spread function dataset for the ROSAT PSPC, and writes the results to an output FITS file (in OGIP-standard format for RPSFs).

A number of options are available, controlled via user-defined parameters, such that the task hopefully serves the needs of users interested in generating predicted profiles for direct comparision with observational datasets, and users who simply require a theoretical profile. Besides being able to specify the inner & outer radii, and number of steps used to calculate the theoretical RPSF, the following functionality is also availble:

- Users may define an input FITS file (via the parameter infile) containing an observed radial profile dataset (in OGIP-standard format). If such a dataset is entered, then a number of variables required for the construction of the theoretical RPSF dataset are (or can be) read from this FITS extension, enabling a direct comparison between the observed and theoretical profiles (see below).
- The PSF of the ROSAT PSPC is a strong function of energy. Users may therefore enter a PHA file (via the parameter phafile) to be used to weight the calculated theoretical profile with the number of counts observed in each PHA channel over the range of:

  - channels defined by the observational dataset entered via the infile parameter (if infile is entered), or
  - energies specified by the parameters en_min & en_max (if no observational dataset is specified, *i.e.* infil = `NONE`)

  If a PHA file is not entered, then a uniform weighting will be assumed across the above range of channels/energies. If a PHA file is entered (using phafile) and/or an observational dataset is entered (using infile), then the EBOUNDS extension (usually found within an RMF) is required to perform the detector channel → incident energy conversion and must be entered via the rmffile parameter.

- The PSF of the ROSAT PSPC is also a strong function of off-axis angle, hence the angle is required in order to calculate the correct theoretical curve. A 'nodding' of the spacecraft was performed for most (if not all) 'pointed' PSPC observations, and an off-axis histogram was stored within the 'DETECTOR' extension of PHA files conforming to OGIP standards for the PSPC. Such a histogram can be specified using the detfil parameter within `pcrpsf`. If such a histogram is not available/required, an off-axis angle can be specified directly via the parameter off_ang.

- Users may specify the strength of the background (via the parameter bkgd in units of counts per pixel) which will be scaled and added appropriately to the theoretical profile generated. If an observational dataset has been entered via the parameter infile, then the specification of this parameter will override the value stored within the dataset.

The algorithm currently in use is valid over the energy range 0.07–3.0 keV and for all off-axis angles, and was supplied by Gunther Hasinger of MPE. A more detailed description can be found in OGIP Calibration Memo CAL/ROS/93-015 (Hasinger *et al* 1994 *Legacy* **4**, 40).

Hidden parameters to pcrpsf

- rad_min [real] (hidden)
  Inner radius for the predicted model RPSF dataset in arcmin. The default value is 0.0 and appropriate for most applications.

- rad_max [real] (hidden)
  Outer radius for the predicted model RPSF dataset in arcmin. The default value is 10.0 and appropriate for most applications.

- n_rad [integer] (hidden)
  The number of radial bins to be used in generating the RPSF dataset. The default value is 300. For most applications it is suggested that this be a large number (at least 100) in order to obtain sufficient resolution to see the structure of the theoretical RPSF of the PSPC. The number of bins does NOT have to be the same as the number of bins in any observational dataset entered via infile.

4.4 qdprpsf - To dump an RPSF dataset to ASCII with embedded QDP commands

In order to facilitate the display of radial profiles, the task `rbnrpsf` is available which reads an OGIP-standard RPSF dataset and writes the data to an ASCII file including QDP commands to enable immediate plotting using XANADU task QDP/PLT.

This task was used to create the numerous figures demonstrating the *psf* of the *ROSAT* PSPC presented in the articles by Hasinger *et al* (1992,1993,1994) published in *Legacy - The Journal of the HEASARC*.

Hidden parameters to rpsfqdp

- predfile [character string]
  The name of the file containing a predicted radial (1-dimensional) PSF dataset. The special (default) value (`%`) indicates that the predicted RPSF dataset is contained within the file specified by the datafile parameter.

Note on Useage

- The task writes QDP commands such that the window is automatically scaled to contain the data, and that a logarithmic y-axis scale is used when QDP/PLT plots the output file. Under certain circumstances this scaling may be incorrect (and will always be incorrect when the radial profile data to be plotted contain zero or negative values). The logarithmic scaling can be turned off using the command `log off`, and axes can be rescaled using `rescale x -1 10`, `rescale y -1e-3 1e2` from within QDP. Further help with QDP is provided by an interactive help.

5 CREATING RADIAL PROFILES IN IRAF

In this section we list outline the steps required in `IRAF` in order to create and write an RPSF dataset.

5.1 imcnts - How to use it to create an RPSF dataset

`imcnts` is a task in the `xray/xspatial` sub-package of `IRAF`.

... section incomplete

5.2 stwfits - How to write an ST table to a FITS file

`stwfits` is a task in the `stsdas/fitsio` sub-package of `IRAF`.

... section incomplete

ACKNOWLEDGMENTS

We thank the numerous people who have made suggestions the `calcrpsf` task and/or this memo.