

rmfgen

Hassan Siddiqui (XMM-SOC, hsiddiqu@astro.estec.esa.nl)
Richard Saxton (XMM-SOC, rds@star.le.ac.uk)

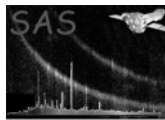
Nov 21 2002

Abstract

Creates an OGIP-compliant redistribution matrix file (RMF) for the source in question.

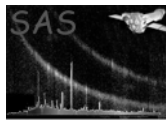
1 Status

version (from **VERSION**): 1.48.2



2 Change history

Revision no.	Date	Author	Comments
1.48	21 Nov 2002	RDS	Support for Timing burst modes added.
1.44.3	25 Feb 2002	RDS	Variable formatting has been implemented which reduces the size of the matrix considerably. The RMF is calculated depending on the patterns represented in the DSS. The pinline parameter is no longer needed.
1.43.4	21 May 2001	RDS	Clarified some aspects of usage. The energy axis is now obtained from the CAL. Elements below 30 eV are zeroed. The execution time has been reduced at the expense of memory usage.
1.41.2	08 Nov 2000	HS	Removed redundant parameters responsedata and photondist and modelpileup
1.41.1	15 Oct 2000	HS	Added new parameter: pinline .
1.41	06 Oct 2000	HS	Changed parameter name responseset to rmfset .
1.40	21 Sep 2000	HS	PN support implemented, and energy grid can now be specified by user
1.37	29 May 2000	HS	Task now generates EBOUNDS table directly from PI channel binning.
1.34	19 May 2000	HS	Implemented comments from J Ballet and J Osborne.
1.33	29 Mar 2000	HS	Updated task description.
1.32	2 Mar 2000	HS	Further updates to interface. Added description of variable-length vector column implementation. Improved description Input/Output Files sections. Mentioned spectral rebinning in Future Developments section.
1.31	29 Oct 1999	HS	Updated parameter interface, and minor modifications to algorithm.
1.28	15 Sep 1999	HS	Added an Invocation section to document, describing how to run the task.
1.5	-	HS	Main proposal in v0.4 of having a combined RMF+ARF generator in rmfgen has been moved to the Future Developments section; the pile-up correction feature is not deemed a high priority, so has not been developed any further. Thus arfgen still contains a correction for flux loss due to pile-up, and rmfgen contains a crude and under-developed correction for spectral distortion, which is not currently available to the user.
1.4	-	HS	Combined RMF and ARF generation into a single task. The pile-up correction algorithm has been replaced by one proposed by J Ballet. Miscellaneous changes made as a consequence of comments from J Ballet.
1.3	-	HS	The code has been ported to C++. The output is now written according to the OGIP ([4]) grouping format, reducing the size of the output file by a factor of approximately 4. A preliminary algorithm to correct for spectral distortion due to pile-up has been added.
1.2	-	HS	new parameter, instrument, has been added. Algorithm modified slightly.



3 Instruments/Modes

Instrument	Mode
EPIC MOS	IMAGING
EPIC PN	IMAGING

4 Use

pipeline processing	no
interactive analysis	yes

5 Description

Further information about the scientific use and accuracy of this task may be found in the document, XMM-SOC-PS-TN-0043, (available from <http://xmm.vilspa.esa.es/calibration/>)

Reformat the detector response and energy bounds information provided by the CAL for the given instrument, corrects for instrumental effects specific to the event selection criteria used and writes the result to a specified dataset (the Redistribution Matrix File or RMF). The dataset conforms to the OGIP standard ([4]).

The RMF matrix generated by **rmfgen** describes the response of the instrument as a function of energy and PI channel; PHA channel responses can not be generated by this task.

This dataset can be used in conjunction with the Ancillary Response File (ARF) dataset (generated by **arfgen**) to perform spectral analysis, the most commonly used analysis package being XSPEC.

Note that details of individual CCF constituents and how the CAL processes such data are beyond the scope of this document, and the reader should refer to the Calibration Access and Data Handbook for such information ([1]).

The following list summarises the current and planned features of **rmfgen**:

Item	Description	Status
1	Grouping of response data above a threshold value	implemented
2	Channel rebinning	implemented
3	Spectrum-response channel range matching	implemented
4	PN instrument support	implemented
5	User-defined energy grid	implemented
6	Modelling for spectral distortion due to pile-up	initial implementation, not available
7	DSS support (including eg pattern selection)	implemented
9	Modelling of spectral distortion due to Charge Transfer Inefficiency (CTI)	to be coded

Items 1-6 are described in more detail below. The remaining items are discussed in the future developments section (13).

5.1 Grouping and threshold

rmfgen generates the o/p rmf in the format specified by OGIP. This format allows response data for a given energy range to be stored in the form of contiguous groups of elements whose values are above a particular threshold. The values below threshold are not written. The fitting package expands this format, replacing those elements in between groups with zero. This is extremely useful in the case of a very sparse response matrix, where values close to zero are of no interest.

The threshold value can be set by the user through the **threshold** parameter and allowed range is between 0 and 1 (the maximum possible response value).

5.2 Variable length column support

rmfgen writes out RMF matrices in variable-length vector column format. This results in smaller matrices than the alternative fixed length format and reduces the demands on machine resources. The default thresholding used is such that elements less than 1.0×10^{-6} are ignored.

5.3 Channel rebinning

Even with the grouping/thresholding mechanism described above, the RMF sizes can still be fairly large. Another possibility is to resample the input spectrum, using an external task such as **evselect**. **rmfgen** will read in the attributes in the spectrum dataset describing that rebinning and offset (**SPECPIX**, **SPECVAL** and **SPECDELT**), and resample the response matrix accordingly. The resampling affects the channel axis of the matrix only; the energy axis is unaffected.

It is expected that a rebinning factor of typically 10 is possible with EPIC MOS spectra, which corresponds to a similar size reduction in the RMF.

5.4 Spectrum-response channel range matching

Fitting packages such as XSPEC do not allow the channel range of the spectrum and that defined in the RMF to differ. XSPEC assumes the spectral range to correspond to the legal range of the **CHANNEL** column, ie the attributes **TLMIN1** and **TLMAX1**.

If the channel range of the spectrum and that of the response data obtained from the CAL are found to differ, **rmfgen** raises a warning and performs one of the following corrective actions:

- If the spectrum contains channels that are outside the response range, **rmfgen** will pad out the response (prior to the grouping described above) to fit the spectral range.
- If the channel range of the spectrum is a subset of the response range, **rmfgen** will truncate the response data to match the channel range.

Note that the only accepted value for **TLMIN** is zero (which is the same as the lower bound for the CAL data, ensuring that the spectral channel range can only be a subset or superset of the response channel range, and not overlapping).

Table 1: Observing mode support within `rmfgen`

Camera	Observing Mode	Support
MOS	FullFrame	Yes
MOS	SmallWindow	Yes
MOS	LargeWindow	Yes
MOS	Timing	No
MOS	Burst	No
PN	FullFrame	Yes
PN	ExtendedFullFrame	Yes
PN	SmallWindow	Yes
PN	LargeWindow	No
PN	Timing	No
PN	Burst	No

5.5 Observing mode support

This version of `rmfgen` supported all the observing modes that were supported by the instrument teams at the time of the SAS v5.3 release. Canned instrument team matrices for some of the missing modes are likely to be made available before the next full SAS release. These may be downloaded from the soc web-site <http://xmm.vilspa.esa.es/ccf/epic/> when ready. A summary of the current state is given in Table 1.

5.6 PN Support

`rmfgen` provides RMFs for PN data, based on the model provided by MPE. The task determines which is the instrument in question from the information in the input spectrum dataset.

The PN response model is spatially dependent and has been defined in terms of the CCD line. The task extracts this from the datasubspace stored in the spectrum file header.

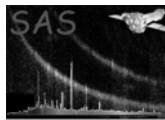
The PN response is also dependent upon the observing mode. This is obtained from the keyword SUBMODE in the header of the input spectrum. If the mode contained in this header is not recognised the PRIME_FULL_WINDOW mode is used by default.

5.7 Pattern support

The RMF of the Epic detectors varies depending on the event grades used to create the spectrum. Different RMFs may be created by the task for single (pattern 0), double (patterns 1–4), triple (patterns 5–8), quadruple (patterns 9–12), single plus double and single plus double plus triple plus quadruple events. The pattern information is read from the datasubspace stored in the spectrum header. The current standard practise is to use singles, doubles or singles plus doubles for Epic-PN and singles or singles plus double plus triple plus quadruple for Epic-MOS but see XMM-SOC-PS-TN-0043 for more details.

5.8 User-defined energy grid

The performance of the task is directly proportional to the number of bins in the energy grid (ie the number of rows of the matrix), while the scientific accuracy of the results of using the matrix improves



(to the limit of the energy resolution of the instrument) the more rows there are in the matrix. This task by default uses the energy grid defined by the CAL to construct the response matrix. This grid currently contains approx 1000 bins for both PN and MOS instruments.

rmfgen now allows the user to specify his/her own grid, thereby allowing the user to decide at which point to compromise performance for accuracy. In addition, a user-specified grid will allow the user to restrict the matrix to a specific energy range of the instrument, and thus allow improved accuracy and a smaller RMF file as output, without having to sacrifice performance.

A user-defined grid can at present be only such that the grid points are evenly distributed across the energy range. It can be specified by setting **withenergybins** = **true** and entering values for the parameters **energymin**, **energymax** and **nenergybins**.

5.9 Examples

5.9.1 A simple example for EPIC-MOS

The simplest way to invoke **rmfgen** is to specify only the spectrum, eg

```
rmfgen spectrumset=spectrum.ds
```

This generates a response dataset, **responsefile.ds**, appropriate for the instrument over which data the spectrum as created, with the pile-up correction switched off.

5.9.2 Creating a named RMF for EPIC-PN

```
rmfgen spectrumset=spectrum.ds rmfset=pnresp.ds
```

This creates an EPIC-PN rmf, called pnresp.ds.

5.9.3 Using the threshold parameter

The example above would generate a dataset, with a threshold of 1.0×10^{-6} . The matrix size may be reduced at the expense of accuracy by lowering the threshold as in the following example:

```
rmfgen spectrumset=spectrum.pha threshold=1e-5 format='var'
```

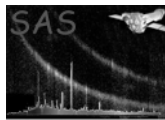
6 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
-----------	------	------	---------	-------------

spectrumset	yes	dataset	spectrum.pha	none
--------------------	-----	---------	--------------	------

Name of the counts spectrum file for which the output RMF dataset is associated.



rmfset	no	dataset	responsefile.ds	none
---------------	----	---------	-----------------	------

Name of output response dataset.

threshold	no	real	1e-5	$0 \leq \text{threshold} \leq 1$
------------------	----	------	------	----------------------------------

Threshold level for grouping. All elements within the matrix below this value are excluded in the formatted output.

withenergybins	no	boolean	false	none
-----------------------	----	---------	-------	------

If **true** use energy grid specified by **energymin**, **energymax**, and **nenergybins**, otherwise, use the grid defined in the CAL.

energymin	no	real	0	none
------------------	----	------	---	------

Used if **withenergybins** = true. Lower energy bound of matrix, in keV.

energymax	no	real	15	none
------------------	----	------	----	------

Used if **withenergybins** = true. Upper energy bound of matrix, in keV.

nenergybins	no	integer	30	none
--------------------	----	---------	----	------

Used if **withenergybins** = true. The number of bins in the energy grid = number of rows in RMF matrix.

format	no	choice	var	fixed, var
---------------	----	--------	-----	------------

Format for **MATRIX** vector column. Set this to 'fixed' to include all matrix elements.

7 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

incompatibleLowerChannelBounds (*warning*)

The channel range specified by the **TLMIN** and **TLMAX** attributes of the column **CHANNEL** differs from the intrinsic channel range of the response data. If the spectral channel range is greater than the response range, **rmfgen** will pad out the response to fit the spectral range. If the spectrum contains channels that are a subset of the response range, **rmfgen** will truncate the response data to match the channel range.

corrective action: Pad out or truncate response data to match channel bounds specified in spectral dataset.

invalidCCFChannels (*warning*)

The program reads the energy boundaries of the spectral channels from the CAL and attempts to rebin these boundaries to the number of bins in the input spectrum. If the ratio of spectral bins to CAL channel boundaries is not an integer the rebinning is not possible and so the boundaries are defined as a regularly increasing set of energies.

corrective action: assume that the channel energies are regularly spaced.

UnknownModeString (*warning*)

The PN rmf generator is dependent on the observing mode. If this is not available from the input spectrum or is not recognised then the task defaults to using full frame mode.

corrective action: Use PRIME_FULL_WINDOW mode

NoDSSPatterns (*warning*)

The RMF generated is dependent on the event patterns used to create the spectrum. If these can not be read from the DSS in the spectrum header then the task assumes that all patterns have been used.

corrective action: Assume all patterns selected

invalidChannelLowBound (*error*)

The TLMIN attribute value for column CHANNEL is not zero

invalidChannelBounds (*error*)

The values of attributes TLMIN and TLMAX for column CHANNEL and attribute DETCHANS are not such that $\text{DETCHANS} \leq \text{TLMAX} - \text{TLMIN} + 1$.

invalidChannelOffset (*error*)

The channel offset specified by attributes SPECPIX, SPECVAL and SPECDELTA does not correspond to an integer value.

incompatibleEnergyRanges (*error*)

If **withenergyranges** is set to true, and **responsedata** is set to 'cal', then this error is raised if the requested range specified by parameters **energymin** and **energymax** is outside the range over which response data is defined in the CAL.

invalidResponseChannels (*error*)

rmfgen is supposed to pad out/truncate the response data to match the spectral channel range. This error is raised if this particular algorithm fails. An SPR should be raised.

noPhaElements (*error*)

Prior to regrouping, no detector channels could be found in the response matrix above the threshold level specified by the **threshold** parameter. This error is raised at a different point in the grouping algorithm to the error **noGroups** below - such an err

groupingInvalid (*error*)

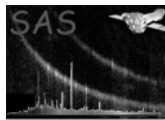
No groups of contiguous response elements above **threshold** could be found in the response matrix, although the maximum number of PI elements found in any one channel is greater than one. This suggests a problem with the grouping algorithm, and an SPR should be raised.

tooFewDataPoints (*error*)

There are less than 2 data points in the dataset specified by the parameter **photondistset** (and **modelpileup** is set to true).

TooManyChannels (*error*)

The spectrum contains too many data channels. The EPIC calibration is defined for a certain PI channel binning. If this is exceeded then the task can not produce a meaningful matrix. Please recreate the spectrum using the standard number of channels or less. The



current standard is 800 MOS bins from PI channels 0 to 11999 and 4096 PN bins from 0 to 20479.

InvalidPatterns (*error*)

The patterns in the data subspace are not the standard selections. See the XMM SAS User Guide for more information about the supported patterns.

8 Input Files

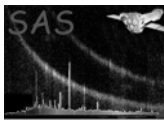
1. Input counts spectrum dataset specified by the parameter **spectrumset**. The format of this dataset is defined in the OGIP document [3], but for the purposes of **rmfgen**, it must contain the following information:
 - The global attributes **TELESCOP**, **INSTRUME**, **FILTER**.
 - A table named **SPECTRUM**, containing the attributes **SPECPIX**, **SPECDELT** and **SPECVAL**. This table in turn contains:
 - A column named **CHANNEL**, which contains the attributes **TLMIN** and **TLMAX**.
2. Input photon spectrum dataset specified by the parameter **photonspectrumset**. This is a DAL dataset containing:
 - A table labelled **DISTRIBUTION**. This in turn contains
 - A 32-bit real column labelled **ENERGY**, containing the energy for each data point (keV).
 - A 32-bit real column labelled **FLUX**, containing the flux level (photons/s/cm²).

This describes the estimated incident photon spectrum. This is used during the pile-up correction process.

This dataset is not used if the parameter **modelpileup** is false.

9 Output Files

1. An OGIP-compliant response dataset. The format of this dataset is defined in the OGIP document [4], but in summary, this contains:
 - A table labelled **MATRIX**, containing the following columns
 - A 32-bit real column labelled **ENERG_LO**
 - A 32-bit real column labelled **ENERG_HI**
 - A 16-bit integer column labelled **N_GRP**
 - A 16-bit integer vector column labelled **F_CHAN**
 - A 16-bit integer vector column labelled **N_CHAN**
 - A 32-bit real vector column labelled **MATRIX**and the following attributes:
 - **TELESCOP**
 - **INSTRUME**
 - **FILTER**
 - **RMFVERSN**
 - **CHANTYPE**



- DETCHANS
- HDUCLASS
- HDUCLAS1
- HDUCLAS2
- HDUCLAS3
- HDUVERS1
- A **EBOUNDS** table, containing the following
 - A 16-bit integer column labelled **CHANNEL**
 - A 32-bit real column labelled **E_MIN**
 - A 32-bit real column labelled **E_MAX**

and the following attributes:

- TELESCOP
- INSTRUME
- FILTER
- RMFVERSN
- CHANTYPE
- DETCHANS
- HDUCLASS
- HDUCLAS1
- HDUCLAS2
- HDUCLAS3
- HDUVERS1

10 Algorithm

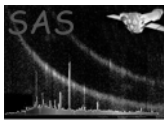
This is divided into four stages:

1. Initialisation
2. Main stage
3. **MATRIX** table creation
4. **EBOUNDS** table creation
5. Cleanup

These are described in detail in the following subsections.

10.1 Initialisation

- Set state of internal RMF data server. Various actions include:
 1. Setting the CAL state to the default CCD chip and mode (Full Frame) for the instrument in question.
 2. Open input counts spectrum file and determine channel range in o/p matrix from the TLMIN and TLMAX attributes for the channel column.
 3. If pile-up is requested, input photon spectrum file for input.
- Create output file



10.2 Main Stage - Generate response matrix

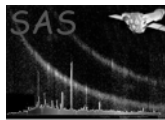
- Get energy ranges for matrix from the CAL
- Write these data out to matrix table in output file
- Foreach energy range
 - Obtain response data from CAL using the corresponding row value for elow and ehig as arguments
 - Perform instrumental corrections. This is currently a placeholder, but will eventually include pile-up and CTI corrections,
 - Normalise response row
 - If necessary, pad out or truncate response row according to the values of TLMIN and TLMAX of the CHANNEL column.
 - If necessary, resample response row according using attributes SPECPIX, SPECVAL and SPECDELTA of the spectral file.
 - Place corrected response row in memory.
- End loop

10.3 MATRIX table creation

- Go through corrected matrix to find out the following statistics:
 1. The largest number of channel groups in a row
 2. The largest number of channel elements in a row above threshold
- Create empty output table header and structure.
- Create a linked list (GROUP_LIST) to store groups of contiguous elements
- Foreach row in corrected matrix:
 - Create a linked list corresponding to a single group (GROUP).
 - Foreach element in row
 - * If element is above threshold, add element to GROUP.
 - * Else push GROUP onto GROUP_LIST, and create a new instance of the GROUP structure for the next group of contiguous elements.
 - End element loop
 - Write out GROUP_LIST to output table.
 - Create a new instance of GROUP_LIST for the next row.
- end row loop

10.4 EBOUNDS table creation

- Write header information for EBOUNDS table
- Get the channel ranges from the CAL and attempt to rebin them for the input spectrum. If this fails: For each spectrum channel, compute the values of the columns E_MIN and E_MAX from the PI channel range corresponding to that channel.
- Write EBOUNDS data to file



10.5 Cleanup

- Close output file, cleanup variables and close the CAL and DAL.

11 Comments

- **rmfgen** generates the RMF based on the properties of only *one* CCD chip of the instrument. In the case of the MOS, this is central chip. In the case of the PN, this is chip 1, ie the chip closest to the optical axis in quadrant 0. A mechanism for providing an emission-weighted RMF will be implemented once DSS support has been coded.
- The RGS is addressed by the task **rgsrmfgen**.
- The size of the o/p RMF is dependent on the spectrum binsize and the number of bins in the energy grid, and can be very large. Heavy consumption of disk space is expected. Memory consumption will also be high as the full RMF is held in memory before being written to disk. Note that the pile-up correction requires knowledge of the entire uncorrected response matrix during each row construction, but accesses this from the CAL, not from memory.

12 Developer's notes

To aid testing, the code uses a 'data-server' concept, which provides the code with a single interface to all data (input parameters, calibration data). This allows the developer to change various data components with 'dummy' values. In particular, instead of using the uncorrected response directly from the CAL the developer can use a dummy version, which can be modified at run-time via the following parameters:

- **responsedata**
- **energymin**
- **energymax**
- **nenergybins**

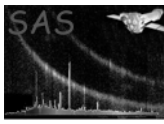
(The parameters **energymin** and **energymax** can also be used when **responsedata** is set to 'cal', to generate a subset of the response defined in the CAL).

The dummy version of the uncorrected RMF is vector of gaussians, although other shapes can be slotted in quite easily.

These parameters are not aimed at the general user (they are commented out) but are useful for debugging and further development. Other modifications, such as modifying the frame period, can be made at compile-time by editing the **RmfSettings** class.

13 Future developments

Future developments include:



- The ability to generate weighted RMFs across all CCDs. The current version generates RMFs for only one position. A future version of this task will generate an RMF over all CCDs, weighted by the source emission within the spatial region of interest. A similar approach has been implemented in **arfgen**.
- Accommodating the spatial distribution of the source flux incident on the detector. This would be useful in order to perform accurate corrections for pile-up and CTI effects. The response model MPE has provided for the PN takes into account the position on the CCD in order to correct for CTI effects. The MOS implementation should operate in a similar way, in that **rmfgen** should provide a position-dependent CTI correction, which involves calculating the noise due to CTI over a given area of the detector, and convolving that with the response.

The spatial distribution could be modelled in terms of an analytical model, which is filtered based on the region of interest, defined probably by the DSS (see also the **arfgen** task description), or in terms of a weighted map.

13.1 Pile up correction

Note that as pile-up is deemed currently to be a low priority issue, no further development in this area is planned in the short term.

If the charge pattern due to two photons coincide on the same frame, and if the pattern is recognisable by the on-board electronics (MOS) or by the pipeline software (PN) a single event will be recorded, with a charge corresponding to the two input charge patterns. For significant pile-up, this leads to a hardening of the spectrum, and a flux loss that varies as a function of energy. **rmfgen** modifies the uncorrected RMF extracted from the CAL to correct for both of these effects.

After discussions with Jean Ballet and Julian Osborne during the SAS V2 Review Meeting, it was proposed to correct for all effects due to pile-up in **rmfgen** only. The **rmfgen** task would then generate a combined RMF+ARF matrix, using the ARF product of **arfgen** as input. The pile-up correction in **arfgen** would then become more-or-less redundant, but would be retained as an option to cater for special circumstances (or, indeed, if this proposed approach does not turn out to be feasible). This section documents the proposal.

13.1.1 Algorithm

The algorithm below is valid for monopixels only.

Compute the combined response matrix $M(E, p) = A(E) \cdot R(E, p)$, where E is the energy, and p is the PI channel, using the following equations (provided by J Ballet):

$$\begin{aligned}
 M(E, p) = & \alpha_1(E) A_0(E) R_0(E, p) \int_x \int_y g(x, y, E) \exp[-W(x, y)] dx dy + \\
 & \frac{T_f}{2} \alpha_1(E) \cdot A_0(E) \int_{E'} \int_x \int_y R_0(E' + E, p) \cdot \alpha_1(E') \cdot f(E') \cdot g(x, y, E') \cdot g(x, y, E) \exp[-W(x, y)] dE' dx dy
 \end{aligned}
 \tag{1}$$

$$W(x, y) = T_f \int \gamma_1(E) \cdot f(E) \cdot g(x, y, E) dx dy
 \tag{2}$$

and

$$f(E) = A_0(E) \cdot dN/dE \quad (3)$$

Where

- $A_0(E)$ is the ARF at energy E obtained from the input arf file,
- $R_0(E, p)$ is the uncorrected response matrix taken from the CAL,
- T_f is the frame time,
- $\alpha_1(E)$ is the expected fraction of incident photons which yield monopixel events at a given energy ([2]),
- $\gamma_1(E)$ is the ‘‘suppression coefficient’’ for single monopixel events at a given energy ([2]),
- $g(x, y)$ is the spatial density (/sq. pixel), ie the PSF for point sources, which is obtained from the CAL
- dN/dE is the incident photon spectrum.

The integration over positions $[x, y]$ is performed in detector coordinates. Only those pixels within a certain region will be considered. This region will be defined by either the DSS or command-line parameters.

13.1.2 Notes

- Note that the combined matrix this algorithm computes is difficult to separate into an RMF and ARF, hence the concept of a combined RMF+ARF in this proposal.
- To perform the pile-up correction, the frame time is required. This is provided, on a CCD-by-CCD basis by the CAL, and is assumed to be constant across the whole exposure. The user should ensure that over-clocked frames are excluded when accumulating the counts spectrum, prior to running **rmfgen**.
- The pile-up algorithm currently only corrects for MOS mono-pixels; accommodating other patterns is in principle straightforward and will be added in due course. How the pattern IDs are ingested is TBD, but it is likely to include them into the source counts spectrum (PI) file, and require this to be inputted via the command line.
- The presence of bad pixels can also distort the patterns detected, which should be catered for in the pile-up correction. However, correcting such an effect is a complex process, and is only significant if the bad pixels are positioned close to the source peak. This may be considered, if the effect is found to be serious, to a later version of the task. It is recommended, however, to remove such events prior to binning the counts spectrum.
- The pile-up correction requires prior knowledge of the incident photon spectrum. Thus an iterative process is suggested in order for a self-consistent solution to be obtained:
 1. Make a first estimate of the incident photon spectrum.
 2. Run **arfgen**, with the pile up correction switched *off*, to generate the initial ARF.
 3. Run **rmfgen**, with the pile-up correction *on*, using the ARF created in the previous step as input to generate the combined response matrix.

4. Using XSPEC, the combined response matrix and the observed counts spectrum for the source in question, generate a new estimate for the incident photon spectrum.
5. Repeat steps 2-4 until convergence - ie there is no significant difference between estimates for the photon spectrum.

Step 1 - making an initial estimate for the incident photon spectrum - can in turn be performed in a number of different ways:

1. In the case of a point source, take an annular region about the core at a distance such that the effect of pile-up is appreciably low. Extract a counts spectrum from this region. Using this region as input run **arfgen**, and **rmfgen** with the pile-up correction switched *off* this time. Run XSPEC using the resultant RMF/ARFs in conjunction with the counts spectrum to make an estimate for the incident photon spectrum.
2. In the case of a point source, there is a monotonic relation between the total incident photon rate and total detected count rate at any energy for monopixels ([2]). Thus run a new TBD task to construct the incident photon spectrum using incident/detected rate curves for each detector channel and the observed monopixel counts spectrum. This task would internally construct the incident photon rate due to monopixels, then divide the spectrum by the fraction of monopixels expected in the non piled-up case ($\alpha_1(E)$ of [2]).
3. Generate a photon spectrum 'by hand' - this is acceptable if the general form of the source spectrum is already known, ie from previous observations.

Alternatively, the user workload may be reduced by performing the response-specific corrections (such as pile-up) *within* the internal XSPEC iteration cycle, or the XSPEC convolution models could be exploited. These possibilities will be examined in due course.

- In the case of the PN, the most important difference that is apparent at the current time is the larger pixel size, which will have an impact on the algorithm details (particularly the suppression terms defined in [2]), but the general framework is expected to be the same.

13.2 CAL usage

The C++ equivalents of the following F90 CAL calls are made:

- CAL_setState
- CAL_getRedistEbins
- CAL_getRedistribution

References

- [1] Christian Erd, Phillipe Gondoin, David Lumb, Rudi Much, Uwe Lammers, and Giuseppe Vacanti. Calibration Access and Data Handbook. Technical Report XMM-PS-GM-20, ESA/SSD, Jan 14 2000. Found at the URL: <http://xmm.vilspa.esa.es/calibration/docs/general/calhb.ps.gz>.
- [2] Ballet J. Pile-up on x-ray ccd instruments. Astron. Astrophys. Suppl., 135:371–381, 1999.
- [3] I.M. George K.A. Arnaud and A.F. Tennant. The OGIP Spectral File Format. Technical Report OGIP/92-007, NASA/GSFC, Sept 1992. Found at the URL: http://legacy.gsfc.nasa.gov/docs/heasarc/ofwg/docs/summary/ogip92-007_summary.html.
- [4] I.M. George K.A. Arnaud and A.F. Tennant. The Calibration Requirements for Spectral Analysis. Technical Report OGIP/92-002, NASA/GSFC, Dec 1998.