#117 :: Chandra Calibration Workshop & Chandra 8 Years Symposium :: 2007 Oct 22–25, Huntsville, AL

Incorporating Effective Area Uncertainties into Spectral Fitting

Vinay Kashyap (SAO), Hyunsook Lee (SAO), Jeremy Drake (SAO), Pete Ratzlaff (SAO)
Aneta Siemiginowska (SAO), Andreas Zezas (SAO), Alanna Connors (EurekaSci)
David van Dyk (UCIrvine), Taeyoung Park (UPitt), Rima Izem (Harvard)

We have developed a robust and general method to incorporate effective area calibration uncertainties in model fitting of low-resolution spectra. Because such uncertainties are ignored during spectral fits, the error bars derived for model parameters are generally underestimated. Incorporating them directly into spectral analysis with existing analysis packages such as Sherpa and XSPEC is not possible without extensive case-specific simulations, but it is possible to do so in a generalized manner in a Markov-Chain Monte Carlo (MCMC) framework. We describe our implementation of this method here. We use the estimates of ACIS effective area uncertainties (Drake et al. 2007, SPIE, v6270, p49) in a MCMC setting, applied to simulated ACIS data, to estimate the posterior probability densities of power-law model parameters that include the effects of such uncertainties.

This method is applicable directly to any spectral model in all parts of the corresponding parameters space. Because no Gaussian approximations are made in calculating the error bars, and the full posterior probability densities of the parameters are constructed, the derived parameter bounds are optimally sized. The method is also fast and is easily generalizable to accounting for the systematic uncertainties in any type of multiplicative factors.

Strategy

- Generate a sample of ARFs that represent the uncertainty in the effective area [**Figure 1**]
- Simulate a set of low-resolution ACIS spectra for a specific model (in this case, power-law with index $\alpha = 2$ and H column density $N_H = 10^{22}$ cm^{-2}) for 10^4 and 10^5 counts
- Compute the posterior probability density functions (*pdf*s) for the model parameters for all combinations of spectra and ARFs and test that the statistical uncertainty is well determined [**Figure 2**]
- Determine the effect of ARF uncertainty on the posterior *pdf*s [**Figure 3**]
- Estimate the sensitivity of the magnitude of the systematic errors to the number of distinct ARFs [**Figure 4**]
- A new MCMC-based mechanism to include the ARF uncertainty directly within spectral fitting [**Figure 5**]
- A proposal to extend the HEASARC ARF standard such that ARF uncertainties can be codified for general use based on a principal component decomposition of the ARFs [**Figure 6**]

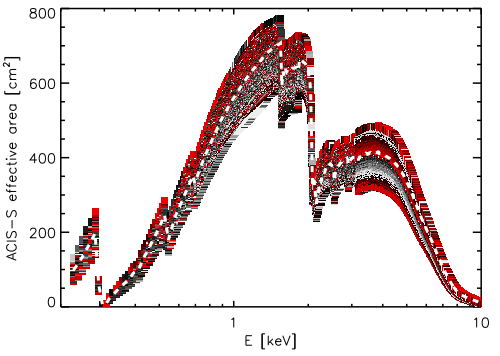

Figure 1: Uncertainty in ACIS-S effective area. The dashed white line shows the default effective area for a nominal observation at the aimpoint, as a function of energy. Numerous effective area curves were synthesized by incorporating the uncertainties in the subsystems (see Drake et al. 2007, CCW Poster #109), and these are shown as the shaded curves that bracket the default. Curves are colored according to how much they differ in toto from the default: black for those which exceed the default and red for the reverse, and the shades represent the extent of the difference. Note that the simulated curves are tangled in a highly complex manner, and the absolute difference between the effective areas does not translate to a segregation of the curves into specific regions.

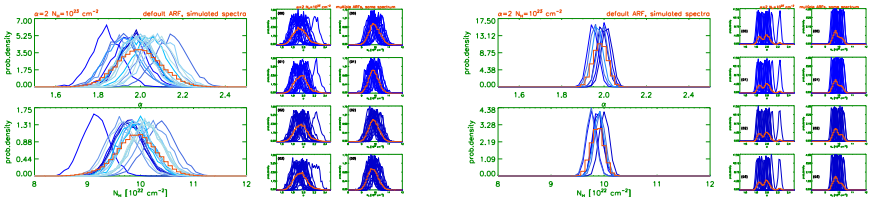

Figure 2: Effect of statistical error on parameter estimates. The posterior *pdf*s of α and N_H calculated for individual pairs of simulated spectrum and ARF are shown for a variety of cases (thin blue histograms). For different spectra adopting the default ARF (**left panels**), and for a small set of representative spectra but adopting different ARFs (**right panels**). The *pdf*s are generated by obtaining parameter draws from a Markov-Chain Monte Carlo (MCMC) algorithm and are binned into a histogram. The thick red stepped histogram represents the *pdf* generated using the draws combined from all the runs shown in each panel. The set of plots on the left are for spectra with 10^4 counts and those on the right, for 10^5 counts. Note that the width of the *pdf* decreases with increasing counts, as is expected: the parameters are better determined when counts statistics are higher. Also note that the *pdf* width for the cumulative case when the ARF is held unchanging is similar to the *pdf* width for the individual runs: the combined *pdf* preserves the statistical error inherent in the data while eliminating the offset biases introduced for individual simulations. The variations in the *pdf*s when the spectrum is held unchanged while the ARFs are changed shows that the effect of the ARF uncertainty on the parameters.

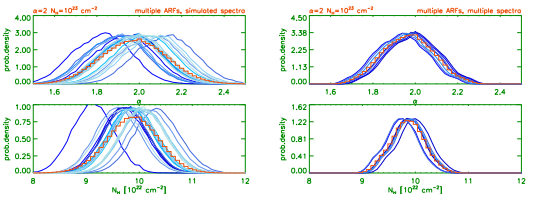

Figure 3: Effect of area uncertainty on parameter estimates. The posterior *pdf*s of α and N_H calculated by first averaging the effect of the ARFs on individual spectra (thin blue curves) are shown. As in **Figure 2**, the *pdf*s are generated using parameter draws from an MCMC algorithm. The *pdf* resulting from combining all the draws is also shown as the red stepped histogram. These curves include the effects of both statistical and systematic errors, and because they represent draws from the true posterior distribution functions, automatically provide the most optimal descriptions of the parameter uncertainties in the presence of effective area uncertainties. Note that the *pdf*s in the high counts case, where the statistical component is relatively suppressed, show that the systematic errors are not Gaussian.

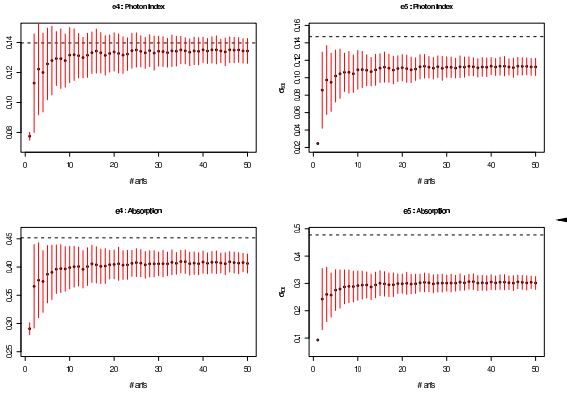

Figure 4: Sensitivity of parameter uncertainty on number of tests. The simplest way to account for the ARF uncertainty is to carry out spectral fits with different realizations of the ARF and combine the resulting *pdf*s to determine the overall error. Clearly, the larger the number of fits done with separate ARFs, the better the estimate will be. This figure shows the magnitude of the total error (small square points) for different numbers of ARFs used. The left panels are for spectra with 10^4 counts and the right panels, for 10^5 counts. The upper panels are for α and the lower ones for N_H. The red vertical bars denote the accuracy with which this error can be determined for any specific set of ARFs, and are determined for each set of N ARFs by running spectral fits $N \times 200$ times choosing different ARFs each time. The "true" estimate of the total error is seen to reach an asymptotic value when only $N \approx 20$ ARFs are used; including more ARFs in the calculation only serves to determine this value more robustly. For comparison, we also show the systematic error estimate computed by Drake et al. (2007, SPIE, v6270, p49; see also Poster #109) as horizontal dashed lines. These points are generated by averaging only the best-fit parameter values calculated for a given spectrum while changing the ARF for each fit. Generally, these values agree with each other, though the values based only on the best-fit tend to increasingly overestimate the magnitude of the total error as the non-Gaussianity of the *pdf*s become more relevant.

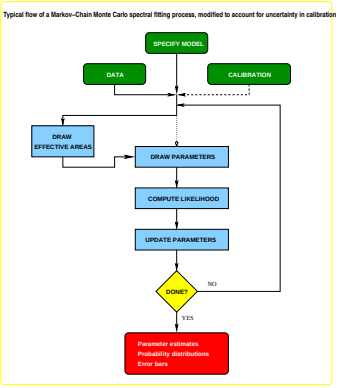

Figure 5: Typical flow of a Markov-Chain Monte Carlo spectral fitting process, modified to account for uncertainty in calibration. We have shown above (**Figures 2, 3 & 4**) that the brute force approach of carrying out spectral fits with different simulated ARFs that represent our uncertainty in the calibration works well and produces reliable estimates of the total error. However, this is extremely inefficient because most of the computational time is wasted in calculating *pdf*s for individual cases. It is possible to speed up the process by two or more orders of magnitude by incorporating the varying ARFs within the calculation. Briefly, if θ are the parameters of interest, we can compute $pdf \sim p(\theta|\mathrm{ARF, Data})\,p(\mathrm{ARF})$, where $p(\mathrm{ARF})$ represents the distribution of ARFs. The manner in which $p(\mathrm{ARF})$ is included is shown here for a typical MCMC data flow diagram. The data and calibration (ARFs, RMFs, etc.) are combined with a spectral model, and the program iterates by drawing new samples of the parameter values (generally as deviations from the current values), computing the new likelihood, and adopting the new parameters as necessary. This process is slightly changed with an additional selection of a new ARF, sampled from $p(\mathrm{ARF})$, prior to drawing new parameter values. We have implented this change in an MCMC based spectral fitting algorithm, and find that we obtain the same *pdf*s as in **Figure 3** with an ≈ 100x improvement in computational speed.

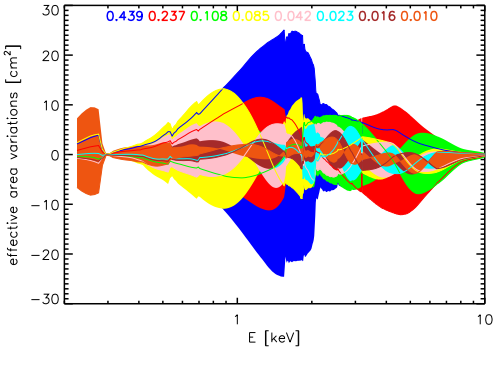

Figure 6: Decomposition of the principal components of variations in the effective area. The procedure described in **Figure 5** relies on the existence of a sample of simulated ARFs, or the ability of the researcher to generate such a sample. This is an onerous burden on most astronomers, but there is a simple workaround. We propose that the ARF uncertainties be decomposed into their most prominent components and stored as files similar to the ARF itself. In order to determine these components, we have used Principal Components Analysis (PCA). This is not a unique solution, but is designed to be eminently practical. The top 8 components of the 1000 ARFs in **Figure 1** are shown here, as deviations from the default ARF, and the shaded regions representing the range of variation accounted for by each component. The fraction of the total variance in the ARFs that is explained by a given component are shown at the top. The 8 components shown here together account for $> 95\%$ of the total variance.

±

+ not necessary to carry out simulations for each model parameter value
+ no simplifying Gaussian assumptions made as to the nature of the error distributions
+ not necessary to know how to generate a sample of ARFs that represent the correct distribution of uncertainties
+ the effect of uncertainties in specific regions in the ARF can be dealt with explicitly by choosing PCA components appropriately
- requires a fitting engine that uses Markov-Chain Monte Carlo techniques
- care must be taken to ensure that a discarded component does not have a large effect on the analysis in the energy range of interest

Proposed Extension to ARF standard

Once a set of ARFs $\{\mathcal{A}_i(E_j), i = 1, N_{ARF}, j = 1, N_{bin}\}$ are generated such that they encompass the range of uncertainty present in our knowledge of the ARF, the differences with respect to the default ARF $\mathcal{A}_0(E_j)$, $\delta\mathcal{A}_{ij} = \mathcal{A}_{ij} - \mathcal{A}_{0j}$ are calculated, and these carry the full uncertainty information. The $\{\delta\mathcal{A}_{ij}\}$ can be decomposed via a principal components analysis to generate eigenvalues $\{e_i\}$ and eigenvectors $\{\nu_i(E_j)\}$. The fraction of the variance in $\{\delta\mathcal{A}_i\}$ accounted for by the k^{th} component is given by $f_k = e_k^2/\sum_{i=1}^{N_{ARF}} e_i^2$. A typical realization of an ARF can be generated as $\{\mathcal{A}'_j = \mathcal{A}_{0j} + \overline{\delta\mathcal{A}_j} + \sum_{k=1}^{N_{comp}} r_i\, e_i\, \nu_{ij}\}$ where $r_i \sim \mathcal{N}(0,1)$ are Gaussian deviates, $\overline{\delta\mathcal{A}_j} = (1/N_{ARF})\sum_i \delta\mathcal{A}_{ij}$ is the average deviation from the default, which is usually expected to be small and close to 0. N_{comp} is formally equal to the number of ARFs in the sample, but can be reduced as needed to discard components that are ignorable. Typically, $N_{comp} \approx 10 - 15$ is sufficient to account for $> 99\%$ of the variance. The $\mathcal{A}'(E_j)$ thus generated is used as the draw from $p(\mathrm{ARF})$ in **Figure 5**.

The results of the PCA decomposition can be stored in files in the same manner as ARFs and distributed widely for incorporating within spectral fitting routines. We have adopted the following format for the file, which is reminiscent of the HEASARC ARF standard (CAL/GEN 92-002)

- `PRIMARY` block: NONE
- `SPECRESP_OFFSET` block: similar to the SPECRESP extension, but containing $\overline{\delta\mathcal{A}_j}$ in place of $\mathcal{A}_{0j}$ in the SPECRESP column
- `PCA_EVALUE` block: an array of N_{comp} values of the eigenvalues e_i, stored in a single column
- `PCA_EVECTOR` block: an array of size $N_{comp} \times N_{bin}$ containing the eigenvectors ν_{ij}, with each row in the file containing the full eigenvector for that component, and with the rows matching one-to-one with those in the extension `PCA_EVALUE`.

This research was supported by NASA/AISRP grant NNG06GF17G and NASA contract NAS8-39073 to the Chandra X-Ray Center