Pileup Modeling

John E. Davis
MIT/CXC

- Overview of pileup model
- Data Preparation
- Use of the model in Sherpa and Iris
- Examples
- Summary
Examples

Q0836 + 7104 (Obs 1450)
HETG/ACIS-S
Sept 17, 1999 Claude Canizares

NGC 4579 (Obs 807)
ACIS-S
May 2, 2000 Michael Eracleous
0.8 sec frame time

Nucleus of M81 (Obs 735)
ACIS-S
May 7, 2000 Douglas Swartz
~3' off-axis
"Standard" Model

\[ C(h) = (N \tau) \int dE \, R(h,E) \, A(E) \, S(E) \]

- \( N \) : number of frames
- \( \tau \) : frame time
- \( R(h,E) \) : RMF
- \( A(E) \) : ARF
- \( S(E) \) : spectral model
- \( C(h) \) : PHA spectrum

[Graph: Counts vs. flux, linearity indicated]

\[ C(h) = (N \tau) (1 - f) \int_{\text{E}} E R(h, E) A(E) s(E) \]

\[ + N n e^{-T/g_0} \int_{\text{E}} A(E) f_s(E)/n \]

\[ \sum_{p=1}^{\infty} \frac{e^{-1+p}}{p!} \int_{\text{E}} E R(h, E) \left[ T(A(E)f_s(E)/n) \right]_p \]

n: number of pileup regions
f: PSF fraction of pileup region
\( g_0 \): branching ratio into “good” grades
\( \alpha \): grade migration parameter

Counts vs. flux

“Standard Model”

“Core”

“Wings”
Data Preparation

- Start with level 1 files!

- Level 2 files have had "afterglow" events removed

- 10-20% of the source events are misidentified as afterglow events!

http://cxc.harvard.edu/threads/acisdetectedafterglow/
http://www.astro.psu.edu/users/patb/ciao-recipe/ciao2-recipi

Construct lightcurves using:

- Wings of PSF
- ACIS readout streak
- Do not depend upon counts in the core of the PSF for lightcurves!
Create ARFs, RMFs, and PtIA files

Use nHARF to create an observation specific ARF. Do not use a canned ARF!

Make sure that the value of the EXPOSURE keyword in the ARF is correct.

For an on-axis point source, use a circular extraction region with a 2"
radius—no larger!
Recommended Extraction Region for on-axis point source

2" radius (4 pixels)
Spectral Analysis Programs

\texttt{xspec} (http://heasarc.gsfc.nasa.gov/docs/xanadu/xspec)

* Partial implementation of pileup model
  (full implementation being tested)

\texttt{sherpa} (http://cxc.harvard.edu/ciao)

* ARF must be defined on a uniform grid
* Cannot ignore PHA channels when fitting pileup
* Frame time and fractional exposure must explicitly set by user

\texttt{isis} (http://space.mit.edu/CXC/isis)

* Uses Exposure Time from ARF

See also:
http://space.mit.edu/CXC/analysis/PILECOMP/
**Sherpa Example**

```plaintext
DATA 1 pha.fits
INSTRUMENT 1 = RSP[my1]("rmf.fits", "arf.fits")
xswabs[w]
xspowerlaw[p]
source 1 = w*p
PARAMPROMPT OFF
w.nH = 0.1
w.nH.min = 0
w.nH.max = 1
thaw w.nH
p.norm = 0.006
p.norm.min=0
p.norm.max=0.1
p.PhoIndx=1.7
p.PhoIndx.min=0
p.PhoIndx.max=3
thaw p.norm p.PhoIndx

JPILEUP[jdpl]
PILEUP 1 = jdpl
jdpl.ftime = 3.2
jdpl.n = 1
jdpl.alpha = 0.5
jdpl.f = 0.95
thaw jdpl.f jdpl.alpha
```

---

**Isis Example**

```plaintext
load_dataset ("pha.fits", "rmf.fits", "arf.fits");
fit_fun ("phabs(1)*powerlaw(1)");
set_par ("phabs(1)", 0.1, 0, 0, 1);
set_par ("powerlaw(1)", [0.006, 1.7], [0, 0], [0, 0], [0.1,3.0]);

set_kernel (1, "pileup");
set_par ("pileup<1>", [1,1,0.5,0.95], [1,1,0,0]);

group_data (1, 8);
xnotice_en (1, 0.5, 12);
```

---
Q0836 + 7104

isis> fit_counts;
Parameters[Variable] = 7[5]
Data bins = 67
Chi-square = 63.1
Reduced chi-square = 1.018

isis> list_par;

phabs(1)*powerlaw(1)

tie-to freeze value min max
idx param
1 phabs(1).nh 0 0 0.03381944 0 0.2
2 powerlaw(1).norm 0 0 0.002399927 0 0.01
3 powerlaw(1).PhoIndex 0 0 1.374511 0 3
4 pileup<1>.nregions 0 1 1 1 10
5 pileup<1>.g0 0 1 1 0 1
6 pileup<1>.alpha 0 0 0.8728455 0.35 1
7 pileup<1>.psffrac 0 0 0.9304663 0.9 1

isis> print_kernel(1);

1: 0.367879 0.625742
2: 0.184252 0.273553
3: 0.061522 0.0797253
4: 0.0154067 0.0174266
5: 0.00308658 0.00304732
6: 0.000515305 0.000444061
7: 7.37403e-05 5.54651e-05
8: 9.23323e-06 6.06186e-06
*** pileup fraction: 0.374258

isis> rplot_counts(1);
isis> fit_counts;

Parameters[Variable] = 7[5]
Data bins = 82
Chi-square = 103.6
Reduced chi-square = 1.346

isis> list_par;

phabs(1)*powerlaw(1)

<table>
<thead>
<tr>
<th>idx</th>
<th>param</th>
<th>tie-to</th>
<th>freeze</th>
<th>value</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>phabs(1).nH</td>
<td>0</td>
<td>0</td>
<td>0.02727911</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>powerlaw(1).norm</td>
<td>0</td>
<td>0</td>
<td>0.001469524</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>powerlaw(1).PhoIndex</td>
<td>0</td>
<td>0</td>
<td>1.764569</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>pileup&lt;l&gt;.nregions</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>pileup&lt;l&gt;.g0</td>
<td>0</td>
<td>1</td>
<td>0.8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>pileup&lt;l&gt;.alpha</td>
<td>0</td>
<td>0</td>
<td>0.6937202</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>pileup&lt;l&gt;.psffrac</td>
<td>0</td>
<td>0</td>
<td>0.902147</td>
<td>0.8</td>
<td>1</td>
</tr>
</tbody>
</table>

isis> print_kernel(1);

1: 0.367835  0.688744
2: 0.186795  0.242635
3: 0.0632391 0.0569848
4: 0.0160571 0.0100375
5: 0.00326167 0.00141443
6: 0.000552117 0.000166095
7: 8.01078e-05 1.67181e-05
8: 1.01701e-05 1.47239e-06

*** pileup fraction: 0.311256

isis> rplot_counts(1);
isis> fit_counts;

Parameters[Variable] = 7[5]
  Data bins = 99
  Chi-square = 120.8
  Reduced chi-square = 1.285

isis> list_par;

phabs(1)*powerlaw(1)

<table>
<thead>
<tr>
<th>idx</th>
<th>param</th>
<th>tie-to</th>
<th>freeze</th>
<th>value</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>phabs(1).nH</td>
<td>0</td>
<td>0</td>
<td>0.08063485</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>powerlaw(1).norm</td>
<td>0</td>
<td>0</td>
<td>0.006880267</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>powerlaw(1).PhoIndex</td>
<td>0</td>
<td>0</td>
<td>1.92351</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>pileup&lt;1&gt;.nregions</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>pileup&lt;1&gt;.g0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>pileup&lt;1&gt;.alpha</td>
<td>0</td>
<td>0</td>
<td>0.4810092</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>pileup&lt;1&gt;.psffrac</td>
<td>0</td>
<td>0</td>
<td>0.9578905</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

isis> print_kernel(1);

1: 0.0241037 0.207965
2: 0.0652543 0.270813
3: 0.117772 0.235102
4: 0.159419 0.153076
5: 0.172633 0.0797342
6: 0.155786 0.03461
7: 0.1205 0.0128769
8: 0.0815551 0.0041921
9: 0.0490641 0.0012131
10: 0.0265656 0.000315941
11: 0.0130762 7.48036e-05
12: 0.00590006 1.62349e-05
13: 0.00245736 1.04828e-05

** pileup fraction: 0.792035

isis> rplot_counts(1);
Summary

- Pileup model seems to work well.
- Use level 1 files and keep afterglow events

Future:

Still need to investigate:
- line dominated spectra
- XMM data
- Grating pileup correction is being tested
Using ISIS, you can:

- Efficiently generate multi-component spectral models using the APEC spectroscopy database for collisionally ionized plasmas.
- Examine ionization balance tables plus data on individual emission lines and continua.
- Use new ISIS functionality with familiar XSPEC models. All XSPEC additive and multiplicative models are available, including table models.
- Analyze data formatted as ASCII files or FITS Type I or Type II PHA files.
- Read arbitrary FITS files and columnar ASCII tables.
- Rebin your data interactively, with direct control over how uncertainties are propagated.
- Fit spectral models to your data either directly, or by folding models through the instrument response.
- Generate fake data for Monte-Carlo analysis of model uncertainties and for proposal planning.
- Analyze piled-up CCD spectra using the model developed by John Davis (MIT).
- Do multi-order spectral analysis (e.g., using the Chandra/LETGS) by assigning multiple instrument responses to a single dataset.
- Fit multiple data sets simultaneously, including an optional instrumental background component.
- Define fit-parameters as arbitrary functions of other fit-parameters (including inequality constraints).
- Compute single-parameter confidence limits
- Generate confidence contour maps; save and re-read as FITS images; plot and overplot contours.
- Fit user-defined models coded in S-Lang or C.
- Apply user-defined fit-statistics coded in S-Lang or C.
- Apply user-defined fit-methods coded in C.
- Apply user-defined RMFs defined in software (rather than as a FITS file)

Future Releases:

Please feel free to submit comments and suggestions for new ISIS features to the ISIS mailing list.

Last updated: Apr 11, 2002
This page is maintained by John C. Houck. To comment on it or the material presented here, send email to houck@space.mit.edu.