# ISIS

## Interactive Spectral Interpretation System

http://space.mit.edu/CXC/ISIS

Design & Development: John Houck (MIT/CXC)

Design, Testing, Feedback:

David Davis, John Davis, Dan Dewey, David Huenemoerder (MIT/CXC)

Purpose:

Support line−based analysis of high−resolution spectra

Features:

− provides an interface to the APED spectroscopy database

− supports fitting models to data

− scriptable

− small, fast and portable

# Assembling Spectral Models

## Load emissivity database

```
plasma (aped);
```

## Define model parameters

```
load_model ("model.dat");
```

```
# id       Temp       Density     Abund     Norm    Vturb      redshift      Nh
#          K          cm^-2                         km/s                   cm^-2
   1       2.e6       1.e-3        1.0      1.0     500.0        0.0       1.e20
   2       4.e6       1.e-3        1.0      2.0     500.0        0.0       1.e20
        Fe=1.3 Si=2.2 Mg=1.4
   3       6.e6       1.e-3        1.0      3.0     500.0        0.0       1.e20
   4       8.e6       1.e-3        1.0      2.0     500.0        0.0       1.e20
        Fe=1.3
```

## Define wavelength grid

```
(lo, hi) = linear_grid (1,25,3000);
```

## Compute total flux on the chosen grid

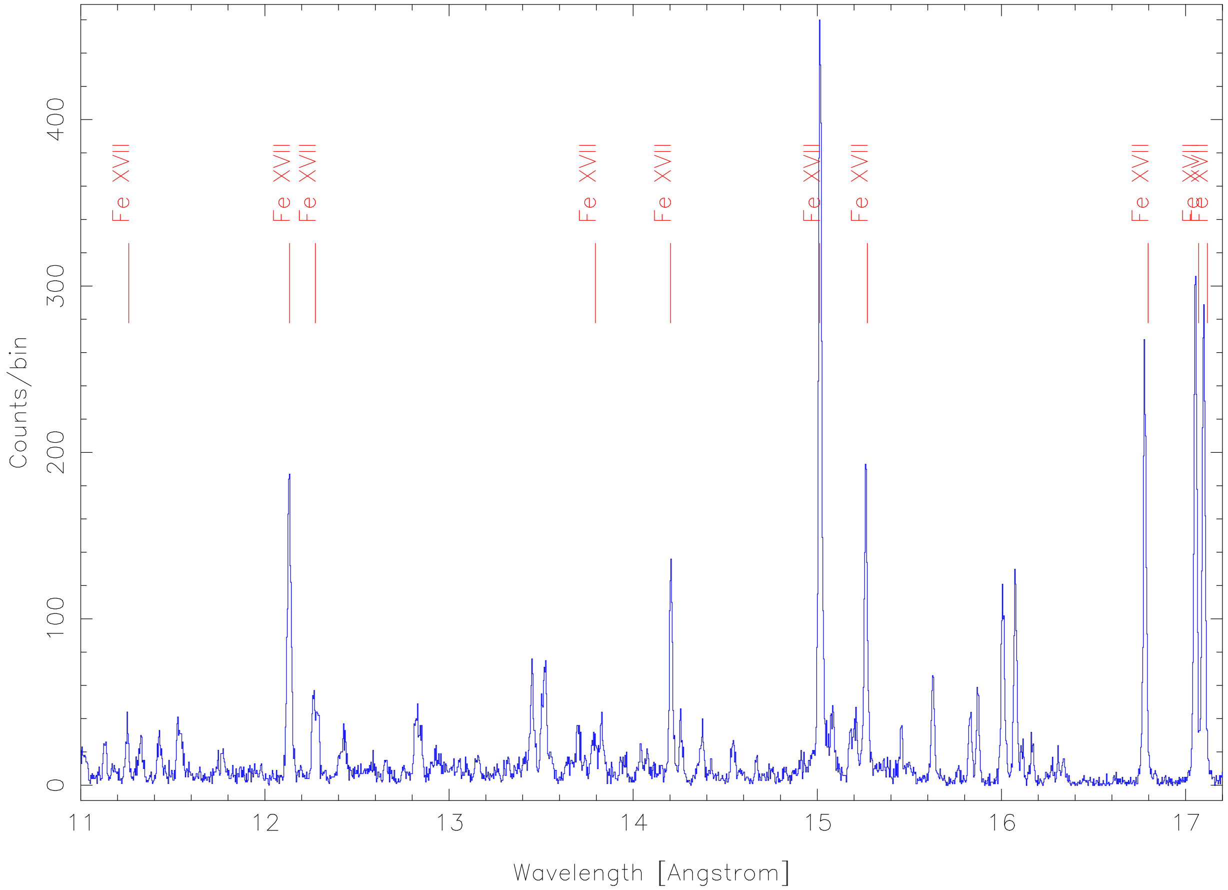
```
flux = model_spectrum (lo, hi);
```

# Access to Emission Line Data

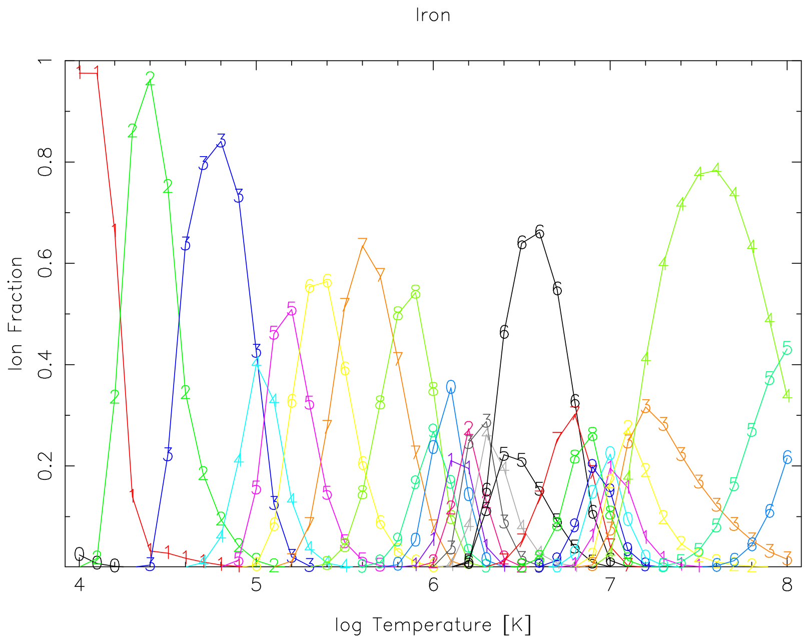
## Use ’where’ command to select groups of lines:

*10 brightest Fe XVII lines between 1−25 Angstrom in the current spectral model:*

```
b = brightest (10, where(el_ion(Fe, 17) and wl(1,25)));
plot_group (b);
```



# Ionization & Emissivity vs. T, n



## Define T, n grid:

*log−spaced temperature grid (K):* `t = 10.0^[6:7.5:0.02];`

## Interpolate spectroscopy database values:

*Ion fraction vs. T:* `frac = ion_frac (O, 8, t);`

*Line emissivity vs. T:* `e = line_em (idx, t);`

*Ionization balance at a given T:* `(X, frac) = ion_bal (Fe, 1.e6);`

# Data Analysis Features

## Reads standard PHA files and responses

```
load_data ("pha2.fits");
load_arf ("arf.fits);
load_rmf ("rmf.fits");
```

## Interactive data handling

```
rebin_data (1, 25);
flux_corr (1);
s = region_counts (1, wlmin, wlmax);
d = get_data_counts (1);
```

## Versatile model fitting

*Fit multiple data−sets simultaneously*

*User−defined* {

- *fit−functions (C or S−Lang)*
- *RMF function (C subroutine)*
- *response−kernel (e.g. pileup)*

```
import ("xspec");
```

# Help & Documentation

*Interactive help:*

```
apropos ("fits");
help ("load_rmf");

isis> plot_data_counts;
Usage:  plot_data_counts (hist_index, [style]);
```

*Reference Manual:* *PDF*, *postscript*

*Web Pages:*

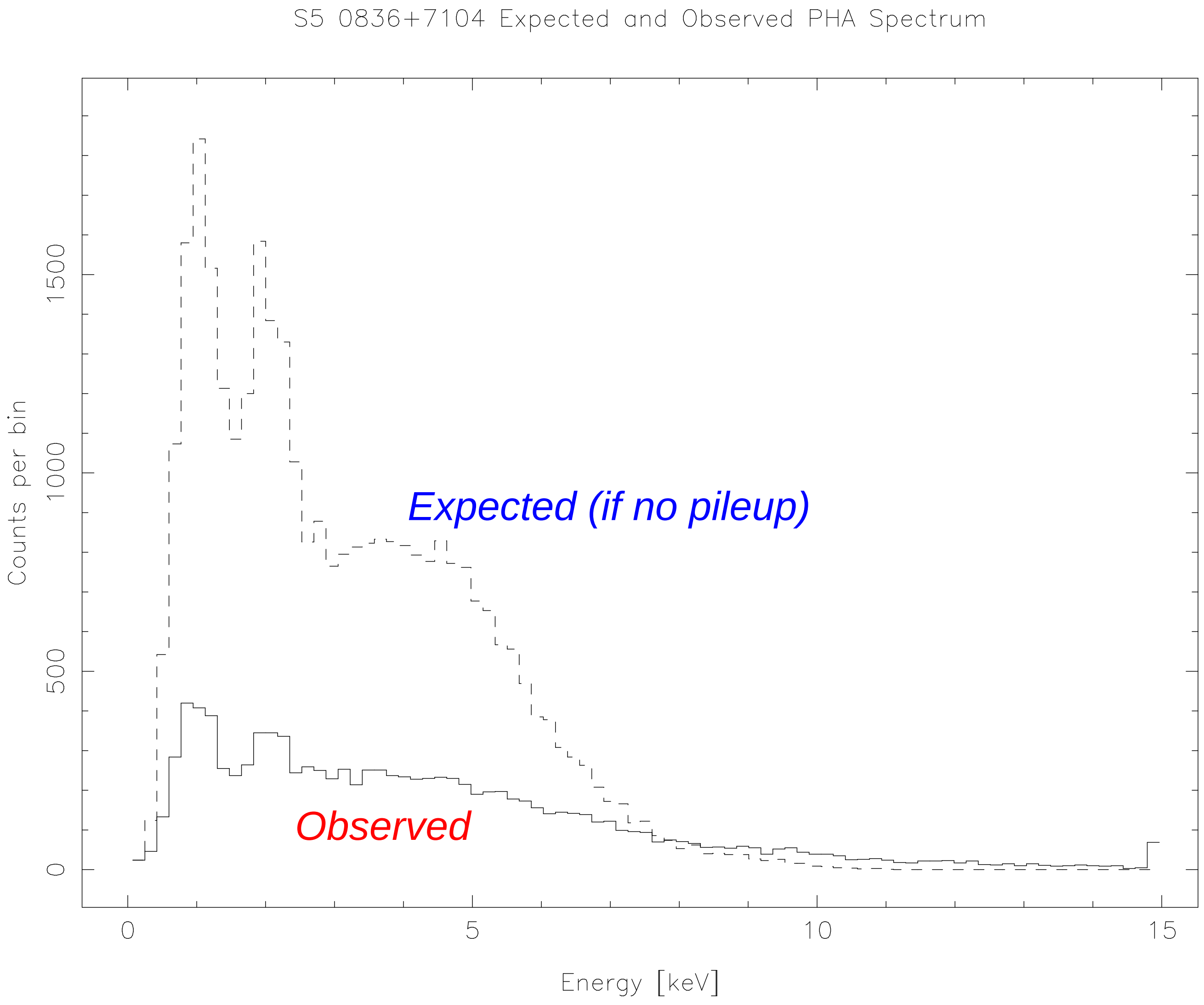
```
http://space.mit.edu/CXC/ISIS
http://www.s-lang.org/
```

*Mailing list:* `isis-users@space.mit.edu`

# Effect of Photon Pileup on CCD spectra

- *grade migration*
- *non−linear instrument response*

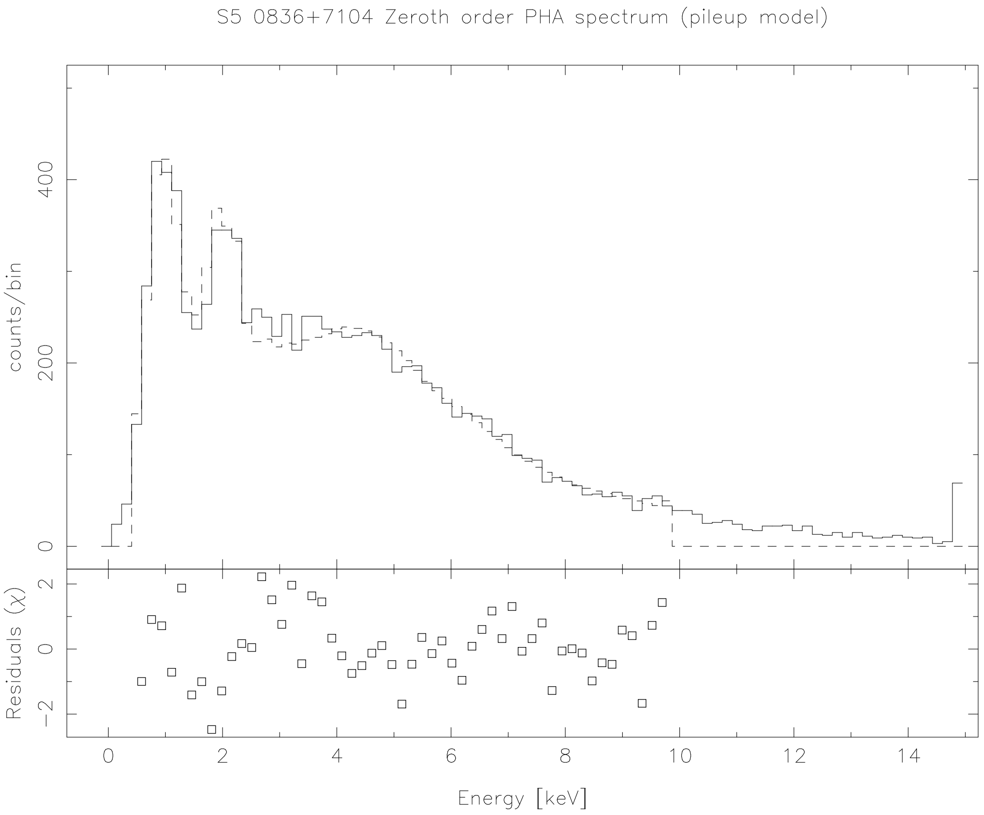
*Davis (2001) ApJ submitted*



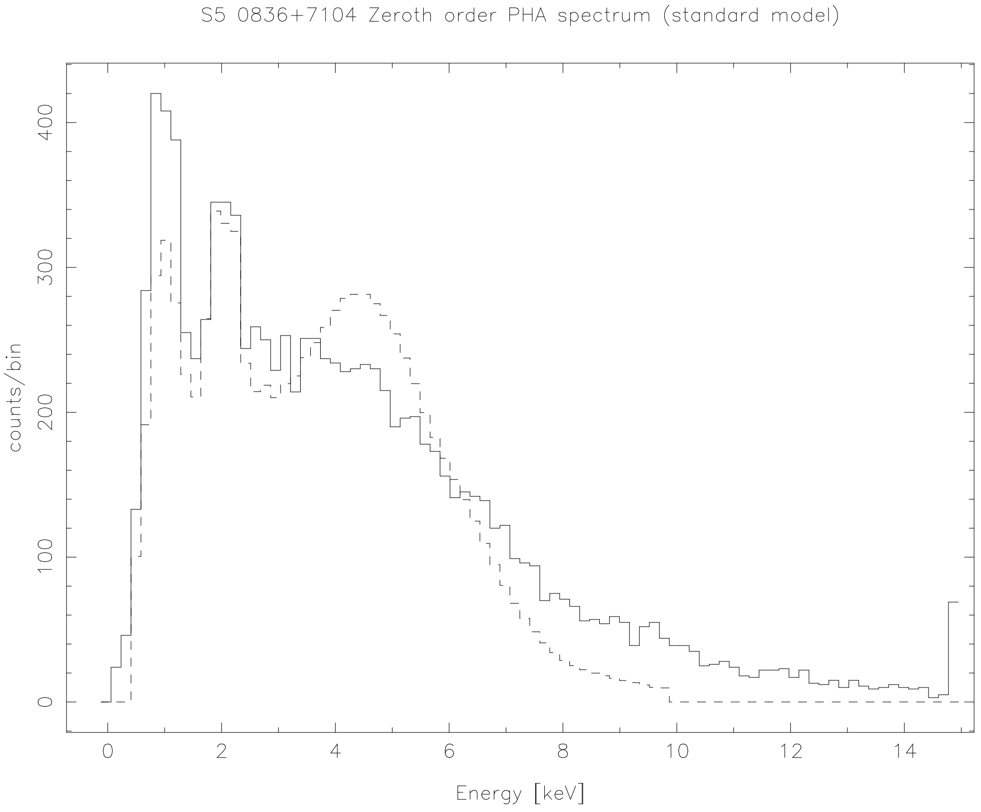
Davis (2001) ApJ submitted
(http://space.mit.edu/~davis/papers/pileup2001.ps)

## *Pileup Model fit:*

```
set_kernel (1, "pileup");
```

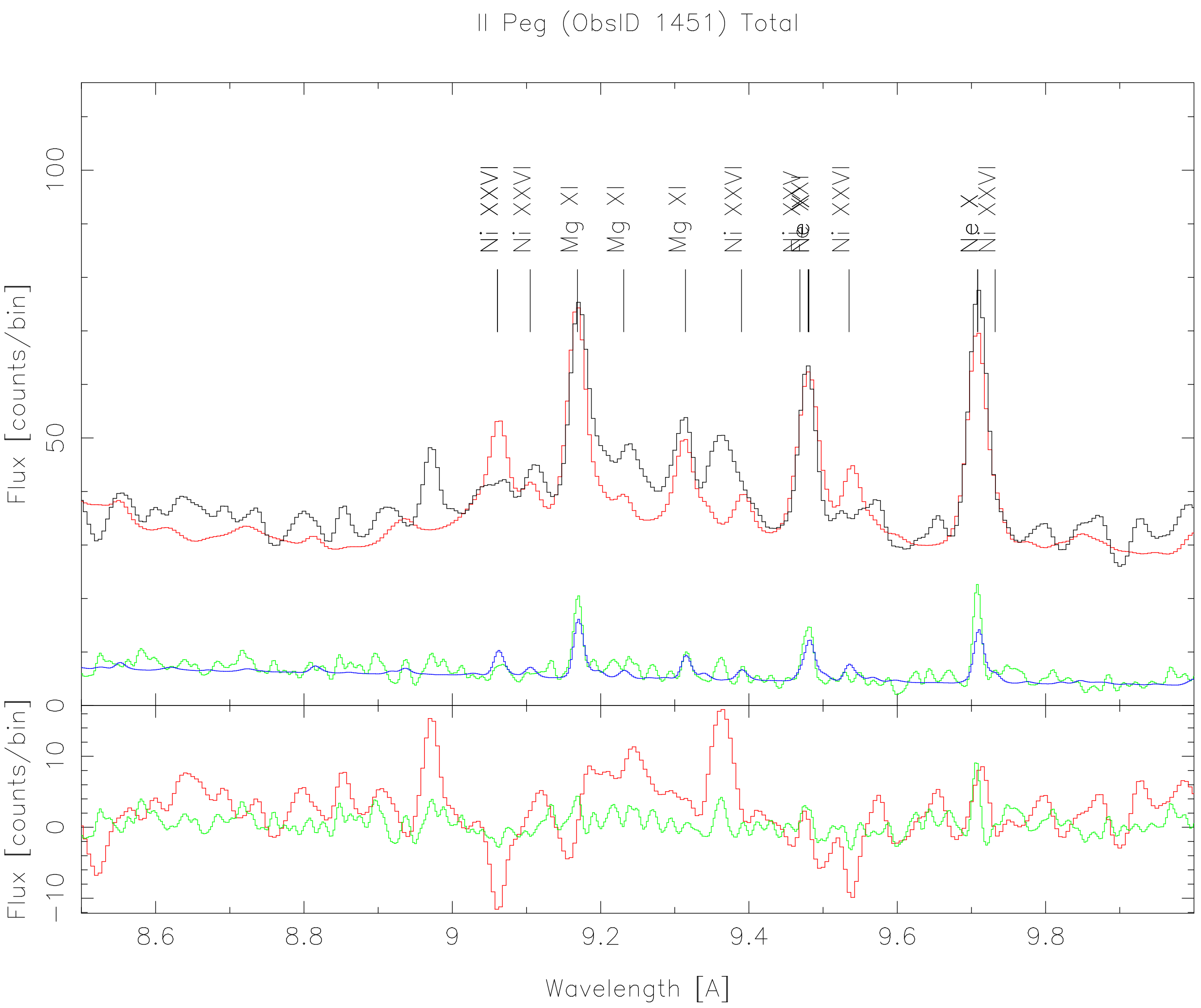


## *Standard Fit:*



# II Pegasi (Coronally active binary star), 45 ksec

Huenemoerder, Canizares & Schulz (2001) ApJ, submitted



http://space.mit.edu/ASC/analysis/IIPeg/IIPeg.html

Black = MEG +/− first orders

Red = Model for MEG, folded through ARF & RMF

Green = HEG +/− first orders

Blue = Model for HEG, folded through ARF & RMF

Line IDs = 15 brightest lines in this region for the chosen emissivity model (APEC, DEM fit)

Data and models smoothed by a Gaussian with dispersion equal to the grating resolution.

# HETG Calibration Analysis

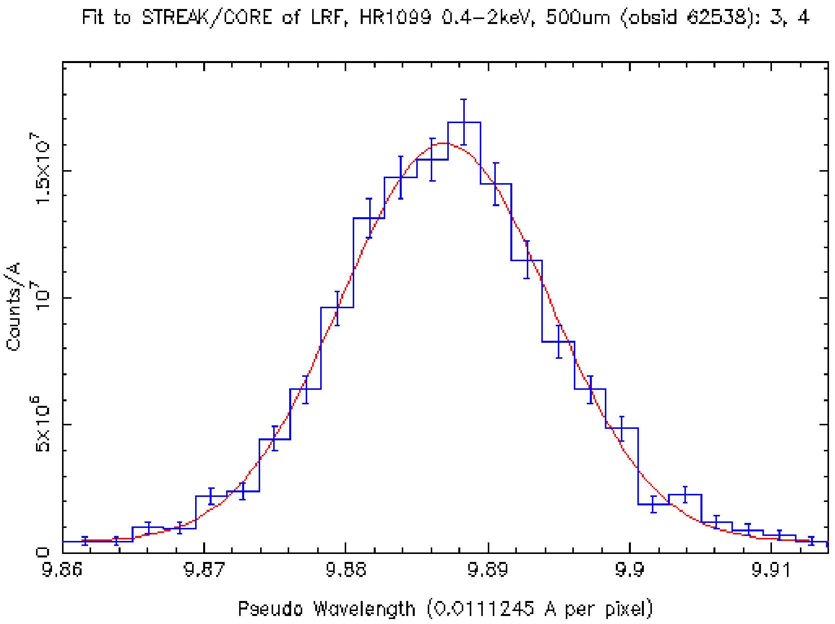
*Dan Dewey's HETG calibration work:*

*−line response function*

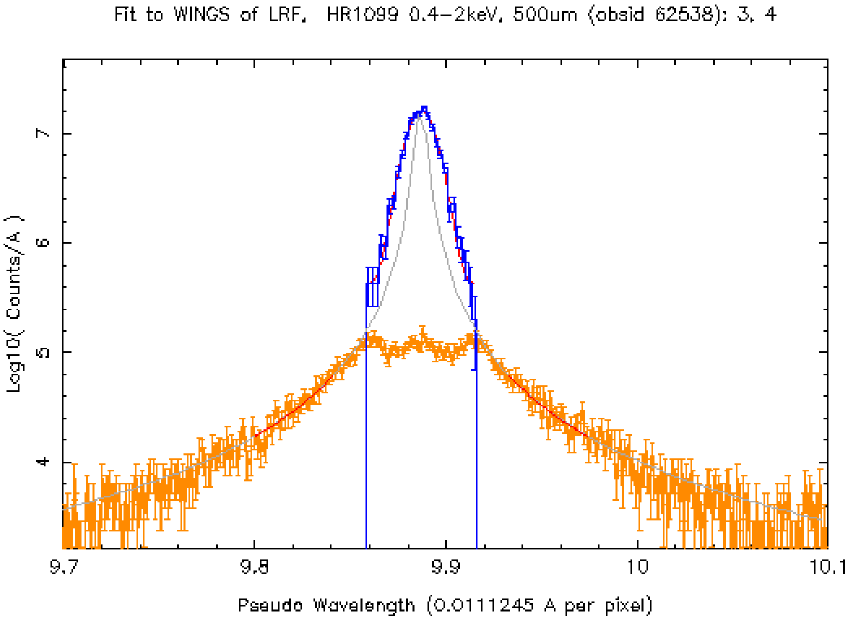
*− wavelength accuracy*

`http://space.mit.edu/HETG/technotes.html`
`http://space.mit.edu/HETG/technotes/wings/wings.html`

LRF core (Gaussian):



LRF wings (Lorentzian):



```
%
%  ISIS demo:
%
%  April 2001
%
%  This demo illustrates a number of ISIS features including
%    - scriptability
%    - spectroscopy database access
%    - user-defined fit functions
%    - access to the PGPLOT library
%    - S-Lang array-based math
%

% Load datafiles, responses, database, spectrum model
define do_init ()
{
   import ("xspec");
   plasma (aped);
   load_model ("aped_2t.dat");

   variable dir = path_concat (_isis_srcdir, "examples/data");
   () = load_data (dir + "/acisf01318N003_pha2.fits.gz", 9);
   () = load_arf (dir + "/acisf01318_000N001MEG_-1_garf.fits.gz");
   () = load_rmf (dir + "/acismeg1D1999-07-22rmfN0002.fits.gz");

   assign_arf (1,1);
   assign_rmf (1,1);

   % notice the 1-20 A region of dataset 1.
   xnotice (1, 1, 20);
}

% A simple user-defined fit-function:
define spec_fit (lo, hi, par)
{
   return par[0] * model_spectrum (lo, hi);
}
add_slang_function ("spec", ["norm"]);

% Use the APED database to compute an APED spectrum
define do_aped ()
{
   fit_fun ("spec(1)");
   set_par (1,1);
   eval_counts;
}

% Use the XSPEC module to compute a MEKAL spectrum
define do_mekal ()
{
   load_par ("mekal_2t.p");
   eval_counts;
}

% Open an X11 plot window with defaults set
% to display black lines on a white background
define new_window ()
{
   () = open_plot ("/xwin");
   _pgscr (0, 1,1,1);
   _pgscr (1, 0,0,0);
}
%
```

```
% Plot counts-data vs. the model folded through the ARF & RMF
% Overlay the _input_ spectrum with bright lines labeled.
define do_plot (xmin, xmax)
{
   xrange (xmin, xmax);
   yrange (-85, );

   % Plot data vs. model
   plot_data_counts (1,1);
   oplot_model_counts (1, red);

   % Overlay the input spectrum, shifted downward for clarity
   variable m, t;
   m = get_model_flux (1);
   t = get_arf_exposure (1);
   ohplot (m.bin_lo, m.bin_hi, 10*t*m.value - 50, blue);

   % Adjust line-label format style
   variable fmt = line_label_default_style ();
   fmt.top_frac = 0.15;
   fmt.bottom_frac = 0.1;
   fmt.offset = -1.05;
   fmt.angle = -90.0;
   fmt.char_height = 0.75;

   % Label the 10 brightest lines in the model
   % with wavelengths in the interval [xmin, xmax]
   variable list, lines;
   list = where (wl(xmin,xmax));
   lines = brightest (10, list);
   plot_group (lines, 1, fmt);
}

% Usage:
%
% do_init;
%
% new_window;
% do_aped;
% do_plot (11,13);
%
% new_window;
% do_mekal;
% do_plot (11,13);
```