ISIS: INTERACTIVE SPECTRAL INTERPRETATION SYSTEM

John C. Houck [a]

[a]MIT Center for Space Research, 77 Massachusetts Ave, Cambridge MA, 02472 USA

ISIS, the Interactive Spectral Interpretation System, is designed to facilitate the interpretation and analysis of high resolution X-ray spectra. It also supports analysis of lower resolution data, such as CCD spectra, via standard instrument response matrices; all XSPEC spectral models are available. In one package, `isis` combines functions to query a database of atomic data and plasma emission models, with functions to manipulate and measure high resolution spectral data. These functions simplify low level operations such as file input, data plotting and database search and retrieval, allowing users to concentrate on higher level analysis issues. `isis` is programmable and extensible, meaning that users can write new functions to simplify repetitive analysis tasks and can extend the `isis` command set by adding those functions (or even user-supplied C programs) as new commands. S-Lang, the interpreted language which provides these features, also provides array-based mathematical functions which greatly simplify common analysis operations.

1. Introduction

High resolution X-ray spectra contain a wealth of information on the physical conditions in an emitting plasma. Understanding these data requires spectroscopists to examine plasma emission models and the underlying atomic data in some detail. Until recently, the voluminous atomic data used to generate plasma emission models was often poorly documented and nearly inaccessible to most users. Efforts by Smith et al., [6] are improving this situation by making available a database of plasma emission models and atomic data in a form which is portable, well documented and easily accessible to users.

The primary design goal of `isis` is to support the analysis of high resolution X-ray spectra by providing scriptable access to a spectroscopy database, together with tools to manipulate and measure high resolution spectral data [2]. By simplifying low level operations such as file I/O, plotting and database search and retrieval, `isis` allows user to concentrate more on higher level analysis issues. Because `isis` is programmable and extensible, scripts can be written to automate repetitive analysis tasks. Users can even extend `isis` by defining new commands written either in its interpreted scripting language (for convenience) or in a compiled language (for faster execution). S-Lang, the interpreted language which makes `isis` extensible, also provides array-based mathematical functions which greatly simplify common analysis operations.

An important secondary design goal was to maximize the usefulness of the software by maximizing its portability. For this reason, `isis` is free software, available under the GNU Public License. It is written in ANSI-C and is intended to be portable across most Unix operating systems. Furthermore, `isis` uses only widely available, free-software components. `cfitsio` and `pgplot` have been widely used by the astronomy community for a number of years and are thoroughly tested and known to run on a wide range of computer systems. Although currently less well known in the astronomy community, S-Lang is very widely used and is a standard component of most Linux distributions. S-Lang is fast, portable and provides an easy-to-use C-like syntax which I believe has a number of clear advantages over other scripting languages such as Perl, Python and Tcl.

In this paper, I briefly describe the principle features of `isis` and provide a few short examples to illustrate how it supports high-resolution spectroscopy. For more information, see the `isis` web page[1].

2. Spectroscopy Database Access

`isis` provides access to the spectroscopy database assembled by Smith et al. [6] which is available on the web[2]. The database contains both atomic data and line and continuum emissivities computed for collisional ionization equilibrium as a function of electron temperature T

[1]See http://space.mit.edu/CXC/ISIS
[2]See http://cxc.harvard.edu/atomdb

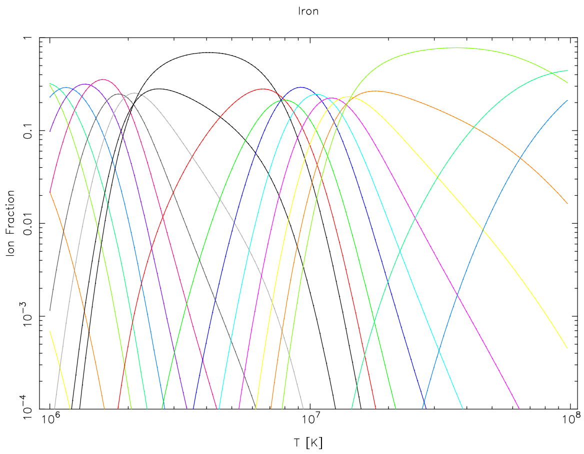

Figure 1. isis provides direct access to ionization balance tables. The curves shown are from [5].

and electron density n. Although isis can access most of the files in the database, a minimum configuration for data analysis requires only the emissivity tables.

isis provides scriptable access to most of the available spectroscopy database files through a variety of S-Lang functions which retrieve numerical values into S-Lang variables. A simple example is shown in Fig. 1 which shows the ionization fraction vs. temperature for a number of Fe XVII ions. A single such ionization balance curve can be generated with a few commands:

```
isis> t = 10.0^[6.0:7.5:0.02];
isis> f = ion_frac (Fe, 17, t);
isis> plot (t, f);
```

In this example, t and f are S-Lang arrays containing a logarithmic (Kelvin) temperature grid and the corresponding ion fractions for Fe XVII.

Using isis, one can generate a wide variety of spectral models using the emissivity tables in the database, including effects such as turbulent line-broadening. Customized multi-component thermal spectra may be defined using the create_aped_fun function by passing it a simple template data structure. For example, to create a two-temperature model with non-solar abundances for oxygen, neon and magnesium:

```
% obtain a template data structure:
  t = default_plasma_state();

% fill in the template with model parameters:
  t.norm = [2.0, 1.0];
  t.temperature = [6.5e6, 8.0e6];
  t.elem = [O, Ne, Mg];
  t.elem_abund = [0.6, 2.0, 1.3];

% define the model:
  create_aped_fun ("xaped", t);
```

In this example, selected fields of a template structure are modified to define a two-temperature model with non-solar abundances. create_aped_fun then created a custom fit-function called xaped, which can then be used for data analysis. It is straightforward to apply arbitrary combinations of such custom models.

Having defined a spectral model, one can examine low-level details such as the total emission contributed by a single ion species or the fractional contribution of various species to a given spectral feature. In Fig. 2, we show an application based on a Chandra/HETGS observation of HR1099. To obtain the database indices of bright lines in a computed spectrum model, one can use the brightest command. To find the 5 brightest neon lines between 10-14 Å in the current spectral model, one could use

```
b = brightest (5, where(el_ion(Ne)
                        and wl(10,14));
```

Here, the where function first retrieves a list of all Ne lines in the specified wavelength range, then the brightest function, selects the 5 brightest. Using that list, one can retrieve all available information on those lines.

3. Versatile Fitting Engine

Because of the variety of instrumental effects that must be accounted for, and the ever-increasing variety of spectral models of interest, multi-wavelength spectral analysis requires versatile software which supports the highest possible degree of user-customization. Progress occurs much more rapidly when spectroscopists can independently develop and apply new spectral models and new approaches to data analysis, without waiting for software developers at some remote site to implement and release suggested modifications to the analysis software.

In adapting isis to solve a variety of data analysis problems, its fitting engine has been repeatedly generalized to support a broader class of problems. The traditional “forward-folding” approach used for most analysis in xspec relates

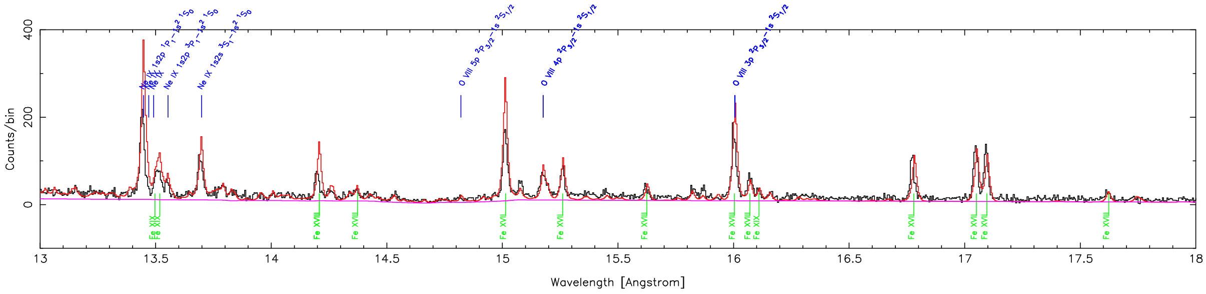

Figure 2. Chandra/HETGS observation of HR1099 [black] with a multi-temperature spectrum model overlaid [red]. The smooth curve shows the location of the model continuum [purple]. Selected bright lines of O, Ne and Fe are labeled.

the predicted counts in each data bin, $C(h)$, to the source model, $S(E)$, using an expression of the form

$$C(h) = t \int \mathrm{d}E\ R(h,E)A(E)S(E) + B(h) \qquad (1)$$

where E is the incident photon energy, h is the detector bin index, $R(h,E)$ is the instrument redistribution function (RMF), $A(E)$ is the effective area function (ARF), $B(h)$ is the background and t is the effective exposure time. Because CCD data affected by photon pile-up is described by a very different expression [3], treating photon pile-up motivated support for a more general form of the above relationship which could encompass a wider range of user-defined applications.

To support such user-defined generalizations, `isis` now "solves" equations of the more general form

$$C_k(h) = t_k \int \mathrm{d}E\ \mathcal{F}_k\Big(R_{\{k\}},\ A_{\{k\}},\ S_{(k)}\Big) + B_k(h) \qquad (2)$$

where k is the dataset index. Each data-set's response may be described by a set of functions $R_{\{k\}}$ and $A_{\{k\}}$, and a fit "kernel" $\mathcal{F}_k$. An example application in which multiple responses are associated with a single dataset is shown in Fig. 3.

Multiple data sets may be fitted simultaneously. Each dataset may have its own background (B), its own set of instrument responses (R, A), and its own "fit-kernel" ($\mathcal{F}$). Even the source spectrum, $S_{(k)}(E)$, may be dataset-specific. This is useful, for example, in cases where one wishes to simultaneously fit multi-wavelength data where

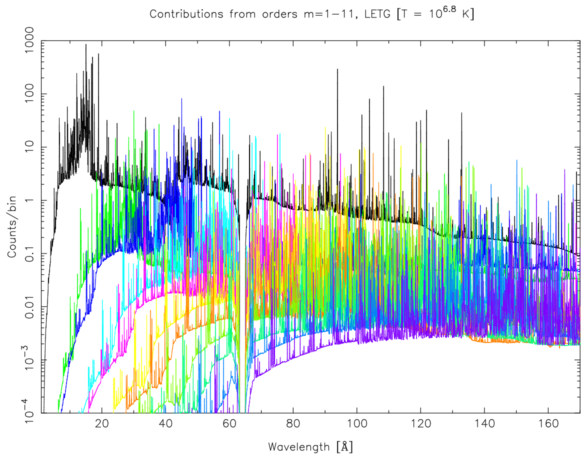

Figure 3. `isis` support for multi-order spectral analysis. To analyze Chandra/LETGS spectra one must account for contributions from higher-orders. `isis` supports folding a spectral model through multiple responses (one for each order) for each dataset. This figure shows the higher-order contributions for orders $m = 1 - 11$ for an `APED` thermal spectrum [6].

different (coupled) emission mechanisms apply to different bands. For example, a synchrotron continuum may contribute flux to the radio, X-ray, and gamma-ray bands, but the X-ray band may also contain significant thermal emission. In such a case, one may wish to simultaneously fit all the datasets, evaluating the synchrotron model in all bands but evaluating the thermal model only in the X-ray band.

I emphasize that all major components of the `isis` fitting engine may be user-defined, either in S-Lang, a compiled language or both. The standard `xspec` kernel and the CCD and grating pile-up kernels [3]; (see also [4], this conference) are built-in and user-defined fit-kernels may be dynamically imported. By default,

$$\mathcal{F}\left(R, A, S\right) = R(h, E)A(E)S(E) \quad (3)$$

so that (2) reduces to the usual forward-folding expression solved by `xspec` [1]. Responses (R, A) are optional; by loading a dataset without an ARF and RMF, it is possible to fit a model directly to the data. Fit-parameters may be arbitrary functions of other fit-parameters.

User-defined source models are fully supported and may be introduced without modifying the `isis` source code and without recompiling. Within `isis`, all `xspec` multiplicative and additive models are available, including table models and local models. Note that, `isis` does not "call" `xspec` via the `xspec` command interpreter. Instead, `isis` is distributed with a module that may be linked directly to the `xspec` subroutine library, allowing much more efficient, direct access. Although `isis` supports a very close linkage with `xspec`, `isis` remains an independent package. If the `xspec` models are not required, `isis` can be used without installing `xspec`.

The `isis` fitting engine provides a variety of fit-statistics including the Cash and χ^2 statistics. The Levenberg-Marquardt and subplex[3] minimization algorithms are built-in. User-defined fit-statistics and minimization algorithms are also supported, and may be imported at run-time.

4. Programmable and Extensible

Although many components of `isis` may be user-defined, it is common for users to use existing components to write S-Lang scripts to automate data analysis tasks.

A convenient way to use `isis` for spectral analysis is to write one or more scripts, usually in the form of several useful functions, and to iteratively edit and run these scripts to achieve the desired result.

For example, one might write a script to load several available data sets, set up the spectroscopy database and compute a fiducial spectrum model. Once that script exists, one can start off later analysis sessions by just running the setup script.

One might then write another script to carry out a few standardized analysis steps, independent of which data set is loaded. Repeating those analysis steps for several data sets might then be a matter of running the script several times. A few simple example scripts may be found on the `isis` web page. [4]

In cases where a large, specialized task is currently unsupported in `isis`, users can extend `isis` by linking in other subroutine libraries, *without* editing *any* of the `isis` source code. This allows large specialized tasks to be handled in compiled code, yet still take advantage of the existing `isis` functions and the ability to script sequences of operations using S-Lang.

REFERENCES

1. Arnaud, K.A., 1996, Astronomical Data Analysis Software and Systems V, eds. Jacoby G. and Barnes J., p17, ASP Conf. Series volume 101.
2. Houck, J.C., and DeNicola, L.A., 2000, ASP Conf. Ser., Vol 216, 591
3. Davis, J.E., 2001, ApJ, 562, 575
4. Davis, J.E., 2002, Proceedings of Workshop on High Resolution X-ray Spectroscopy with XMM-Newton and Chandra, Ed. Branduardi-Raymont, G.,
5. Mazzotta, P., Mazzitelli, G., Colafrancesco, S., Vittorio, N., 1998, A&A.Supp. 133, 403
6. Smith, R. K., Brickhouse, N. S., Liedahl, D. A., & Raymond, J. C. 2001, ApJ, 556, L95.

ACKNOWLEDGEMENTS

Extensive usage and testing by David Huenemoerder, Dan Dewey, John E. Davis and others in the Chandra X-Ray Center at the MIT Center for Space Research has helped find and correct design flaws, making `isis` much easier to use. Good advice and abundant suggestions from John E. Davis have improved `isis` tremendously. The entire development effort was made possible only through the continued support of Claude Canizares, David Huenemoerder and Dan Dewey. This software was developed by the MIT Center for Space Research under contract SV1-61010 from the Smithsonian Institution.

[3] http://netlib.org/opt/subplex.tgz

[4] http://space.mit.edu/CXC/ISIS