The AtomDB Charge Exchange Model

Randall Smith, Adam Foster

April 8, 2014

The ACX package includes two primary tools: the XSPEC package ACX, which includes a number of related charge exchange models described below, and a program 'dacx', which displays line strengths from ACX models based on parameters identical to those in the XSPEC models.

Intallation

The acx package makes an honest, but limited, attempt to be installable on multiple platforms. It should work on any modern Linux system and on Macs running OSX 10.6 or later. It may work on other machines; if it fails, however, please consider how much you paid for this before complaining too loudly. Complaining politely, however, may prove useful, especially if you're willing to work with us to see how to fix it.

Since the primary purpose of ACX is to create a model usable in XSPEC, it is important to have XSPEC (and, in fact, the entire HEASOFT package) available on the computer you're going to install it on. HEASOFT is available at `http://heasarc.gsfc.nasa.gov/docs/software/lheasoft/`, in case you do not have it already installed. If you plan to use acx, I strongly recommend you use the 'build-from-source' option, as you're going to want to make sure the C compiler you used to build HEASOFT is the same as the one you built acx with. This may not be absolutely necessary, but I've found that the XSPEC model creation tool is a bit fragile so it's best to humor it. The following procedure should work so long as you have HEASOFT installed and ready when you run the compilation step. That means you should be able to type '`xspec`' at the command line and have xspec start **before** you begin the compilation process. If you don't, acx will **not** compile the xspec module. The installation procedure is as follows, for version 1.0.0:

```
unix> source $HEADAS/headas-init.csh
unix> tar zxf acx-1.0.0.tar.gz
unix> cd acx-1.0.0
unix> ./configure
unix> make
```

This should compile all the necessary libraries, along with the dacx code (in the src/ directory) and the XSPEC models in the xspec/ directory. Note that the installation is done in place; if you really want to use the '`make install`' option, you can try it but there are no guarantees.

If the above does **not** work and all you really want is the xspec module, I recommend restarting from the beginning and just trying to compile the XSPEC module thusly:

```
unix> source $HEADAS/headas-init.csh
unix> tar zxf acx-1.0.0.tar.gz
unix> cd acx-1.0.0/xspec
unix> cp acx.h.in acx.h
...edit acx.h to put in the full path to the acx-1.0.0directory
...
unix> initpackage acx model.dat .
unix> hmake
```

This will compile just the XSPEC model. It's important that when you edit the `acx.h` file you define the DIRECTORY variable to just point to the top level directory, ending in acx-1.0.0, not the data/ subdirectory. The code tacks on the /data/ part itself.

The XSPEC module

Installing into XSPEC

If the installation worked out ok, you should now be able to install the acx model into your running copy of XSPEC. acx requires a fairly large chunk of memory to install, and seems to work best if you load it right after starting XSPEC. Not 'working best' translates into unexpected core dumps, losing all of your fits, so I urge you to install acx immediately upon starting XSPEC if you plan to use it.

Installing acx into XSPEC is simple:

```
XSPEC> lmod acx /path/to/acx-1.0.0/xspec
```

If this completes without an error message, you should be ready to go.

The basic acx / vacx models

For more information about the physics in the acx models, please see the Smith et al. (2014, ApJ) and Smith et al. (2012, AN) papers in this directory. These instructions provide only a how-to guide. The `acx` model takes seven parameters, defined here:

kT The equilibrium ion population distribution "temperature", in keV. Note that this does not a true Maxwellian velocity distribution; instead, it sets the ion population as though it were in collisional ionization equilibrium created by electrons at this temperature. In practice, ion population distributions involved in charge exchange may not be represented by a single temperature or even multiple temperatures. However, this is a necessary simplification to make an XSPEC model practicable.

FracHe0 The fraction of neutral Helium (relative to the total neutral population, assumed to be H and He) in the plasma. By default set to a cosmic value of 1/10th He, or 0.090909.

Abundanc The relative (to Anders & Greveese 1989 solar values) abundance of metals in the plasma. **Note:** This should be kept frozen in almost all cases, because a change in the overall metal abundance will act as a change in the normalization value. In a pure charge exchange spectrum there is no way to measure the absolute abundances since Hydrogen, the typical astronomical normalizing abundance, does not create any detectable charge exchange emission. Users strongly urged to ensure they understand why this is so before using acx.

redshift The redshift of the emitting plasma.

swcx For most cases, this should be 0; set it to 1 for modeling solar wind charge exchange (SWCX). See § for more details.

model The assumed model for the distribution of n and l in the charge exchanging ions. See § for more details.

norm A scaling for the emissivity of the plasma. In concept, this should depend upon the densities of the charge exchanging ions and neutrals and their distance from Earth. However, the acx model is an approximation that does not include the actual cross section of the charge exchange process itself, and so the norm has no physical use except

as a relative scaling. Users strongly urged to ensure they understand what this means before using acx.

The corresponding variable-abundance version `vacx` takes similar parameters except it allows for the relative abundances of individual elements to be varied. At least one of these elements should be frozen, however, to avoid problematic interactions with the XSPEC norm.

The swcx flag

The sole difference between the swcx=0 and the swcx=1 models is in how the charge exchange model is implemented. In the standard model, suitable for sources galactic or extragalactic sources, the assumption is made that once an ion interacts via charge exchange with a neutral atom, it will then immediately find *another* neutral and will repeat the charge exchange process until the ion is fully neutralized. Thus a fully-stripped O^{+8} ion will emit not only O VIII lines, it will also emit O VII, O VI, and on down until the oxygen can no longer undergo charge exchange. Physically, the picture is that of a hot ion from some source – a supernova shock, galactic superwind, etc – has run into a dense neutral cloud. In this case, the CX cross section is so large that the process will neutralize the ion in a very short distance, certainly smaller than can be resolved with an X-ray telescope.

The only time this statement is not true is if we are actually within the cloud of ions – in this case, coming from the solar wind and interacting with neutral matter in the heliosphere. In this case, it's entirely possible the ion will not interact again within the field of view of the telescope. One way to imagine this process is to consider following the progress of a large coronal mass ejection (CME) as it travels through the heliosphere. At early times, while the CME is just leaving the Sun, it will have mostly highly-ionized gas, so any CX will only show high ions. After it crosses one AU and heads out into the more distant reaches of the solar system, it will show lower ions with a lower temperature. This would be clearly require swcx=1, as just because an O^{+8} ion undergoes CX and emits a O VIII photon does not imply we will also see an equal number of O VII and O VI photons in the same observation. Instead, a O^{+7} ion will travel further from the Sun and eventually undergo more CX reactions.

As a practical matter, then, these models will have far less emission for a given temperature and normalization value than the standard models.

The acxion model

This model is useful largely as a diagnostic of charge exchange patterns from particular ions. The model takes 5 parameters, listed here:

FracHe0 Same as the acx model.

El Z for the ion, or the number of protons in the ion

rmJ The ionization stage; 0 for neutral, Z for fully-stripped.

redshift Same as the acx model.

model Same as the acx model.

norm Same as the acx model.

Thus, to find out what the spectral shape of charge exchange emission from Ne^{10} is, use El=10, rmJ=10.

The meaning of the 'model' parameter

The model parameter sets how the code assumes electrons from the neutral participant in the charge exchange land on the ion. Essentially, there are two primary questions: which (1) principal quantum number n state and (2) total angular momentum l state will be populated? The spin state could also vary (and does, in some cases, as a function of impact velocity), but we assume this distribution is done equally in all cases. This latter assumption is based on a lack of available data, however, and not out of any physical principle.

There are 16 different models included in the initial distribution. These come about as we have two different versions of how the n state is distributed, and eight different cases for l.

Predicting the actual line emission following charge exchange thus requires first determining the exact atomic level (or distribution of levels) of the charge-exchanged ion. We follow the approximation described by Janev & Winter (1985), who found that the peak of the principal quantum number n distribution is at

$$n' = q\sqrt{\frac{I_H}{I_p}}\left(1 + \frac{q-1}{\sqrt{2q}}\right)^{-1/2} \tag{1}$$

where again q is the charge of the ion, I_H is the ionization energy of the neutral ion (assumed here to be hydrogen), and I_p is the ionization potential

in atomic units. In the case of the ACX model parameter, models 1-4 and 9-12 assume all the ions ended up in this level, while models 5-8 and 13-16 use a weighted distribution between, so that if n' is equal to (say) 4.7, then 30% of the ions would populate $n = 4$ while 70% would be in $n = 5$. In practice, then, models 5-8 and 13-16 should be a more accurate depiction of the real distribution, although a comparison between related models (*e.g.* 3 & 7, or 4 & 8) shows that there is relatively little difference in the final spectrum in practice.

The primary uncertainty in the process, however, is the angular momentum (l) of the exchanged electron. The correct result will be velocity-dependent, which is problematic since in most cases the input ion velocity (or position) will not be known. Following the approximate nature of this model, we address this uncertainty by simply providing a range of options for the model.

By default we use orbital angular momentum distributions based on the nl of the captured electron, creating the 1-8 model cases. This approach best handles the intermediate weight ions where LS coupling is inappropriate and L is not a reliable quantum number. However, it does present a different set of problems with heavier ions where there is significant configuration mixing. For example, defining which levels truly represent a captured $11f$ electron is not exact.

For reasons of convenience, we initally developed methods that replaced the orbital angular momentum l with the total orbital angular momentum L. This was used for the fits presented in Smith et al (2012, 2014) because there is no practical difference between the two for the Li-, He- and H-like ions which dominate X-ray spectra, and L is stored explicitly in AtomDB. These no longer our default models, however, and are now labeled the 9-16 models. They may be removed in future releases.

Given the qualitative nature of these distributions, we currently include both approaches in the distributed ACX model. Regardless of these differences, within all the levels which can be created by each capture with varying L, S, and J quantum numbers, we distribute the population using statistical weighting $(2J + 1)$.

The four different distributions considered based on either orbital (1-8) or total (9-16) angular momemtum are:

1. "Even:" weighted evenly by angular momentum, models 1, 5, 9, and 13.

2. "Statistical:" weighted by the relative statistical weight of each level, models 2, 6, 10, and 14.

3. "Landau-Zener:", models 3, 7, 11, and 15, weighted by the function

$$W(l) = \frac{l(l+1)(2l+1) \times (n-1)! \times (n-2)!}{(n+l)! \times (n-l-1)!} \tag{2}$$

4. "Separable:", models 4, 8, 12, and 16, weighted by the function

$$W(l) = \frac{(2l+1)}{Z} \times \exp\Big[\frac{-l \times (l+1)}{z}\Big] \tag{3}$$

The latter two methods in this list are from Janev & Winter (1985). The separable is the default, preferably with the second n distribution model using the orbital angular approach, making the standard starting method model 8. We hope that providing the alternative models we will allow users to test the sensitivity of their data to the model approximation. Despite the simple nature of these models, we expect they will be useful to check if charge exchange could or could not be responsible for some or all of an observed spectrum.

For convenience, Table 1 lists each case and the assumptions involved. For slow winds ($v < 1000\,\mathrm{km/s}$), either model 7 or 8 is recommended, along with tests for sensitivity using other values. The statistical model would only be applicable for high-velocity CX, while the even distribution would be a good test for sensitivity.

The dacx program

The dacx program is designed to output the line strengths from a charge exchange model using the values obtained from an XSPEC fit. The code outputs the line id's, their wavelengths, and the line strengths in units of photons $\mathrm{cm}^{-2}\mathrm{s}^{-1}$. It uses the ever-popular IRAF interface, which should be familiar to all HEASOFT users via the `pset`, *punlearn* and other commands. To run dacx, first set the environment variable PFILES to ../pfiles and then just type dacx:

```
unix> setenv PFILES ../pfiles
unix> bin/dacx
```

The code will prompt you for all the necessary options, which are for convenience also listed here:

OutputFileName The name of the optional output file; set to STDOUT if you want screen output

Wavelength Set this to yes if you want to work in wavelength units (=Å). Otherwise, the code assumes Energy units (=keV)"

LambdaMin The minimum wavelength (or energy) to output

LambdaMax The maximum wavelength (or energy) to output

MinEmiss The minimum emissivity (in photons $cm^{-2}s^{-1}$) to still output

kT Equilibrium ion balance temperature

FracHe Fraction by number of Neutral He

Abundance Abundance relative to solar (Anders and Grevesse)

Model Model number (1-8)

Norm XSPEC Normalization

redshift Redshift (hidden; must be actively set with pset)

swcx Standard model (if no) or Solar Wind Charge Exchange (if yes). (hidden; must be actively set with pset)

C Relative abundance of Carbon (hidden; must be actively set with pset)

N Relative abundance of Nitrogen (hidden; must be actively set with pset)

O, Relative abundance of Oxygen (hidden; must be actively set with pset)

Ne Relative abundance of Neon (hidden; must be actively set with pset)

Mg Relative abundance of Magnesium (hidden; must be actively set with pset)

Al Relative abundance of Aluminum (hidden; must be actively set with pset)

Si Relative abundance of Silicon (hidden; must be actively set with pset)

S Relative abundance of Sulfur (hidden; must be actively set with pset)

Ar Relative abundance of Argon(hidden; must be actively set with pset)

Ca Relative abundance of Calcium (hidden; must be actively set with pset)

Fe Relative abundance of Iron (hidden; must be actively set with pset)

Ni Relative abundance of Nickel (hidden; must be actively set with pset)

clobber If yes, can overwrite existing output file. (hidden; must be actively set with pset)

Table 1: Meanings of the (v)acx model value

Value	Assumptions
1	All CX into one n shell, even distribution by orbital angular momentum l.
2	All CX into one n shell, statistical distribution by orbital angular momentum l.
3	All CX into one n shell, Landau-Zener distribution by orbital angular momentum l.
4	All CX into one n shell, Separable distribution by orbital angular momentum l.
5	CX distributed by weight into neighboring n shells, even distribution by orbital angular momentum l.
6	CX distributed by weight into neighboring n shells, statistical distribution by orbital angular momentum l.
7	CX distributed by weight into neighboring n shells, Landau-Zener distribution by orbital angular momentum l.
8	CX distributed by weight into neighboring n shells, Separable distribution by orbital angular momentum l.
9	All CX into one n shell, even distribution by total angular momentum L.
10	All CX into one n shell, statistical distribution by total angular momentum L.
11	All CX into one n shell, Landau-Zener distribution by total angular momentum L.
12	All CX into one n shell, Separable distribution by total angular momentum L.
13	CX distributed by weight into neighboring n shells, even distribution by total angular momentum L.
14	CX distributed by weight into neighboring n shells, statistical distribution by total angular momentum L.
15	CX distributed by weight into neighboring n shells, Landau-Zener distribution by total angular momentum L.
16	CX distributed by weight into neighboring n shells, Separable distribution by total angular momentum L.