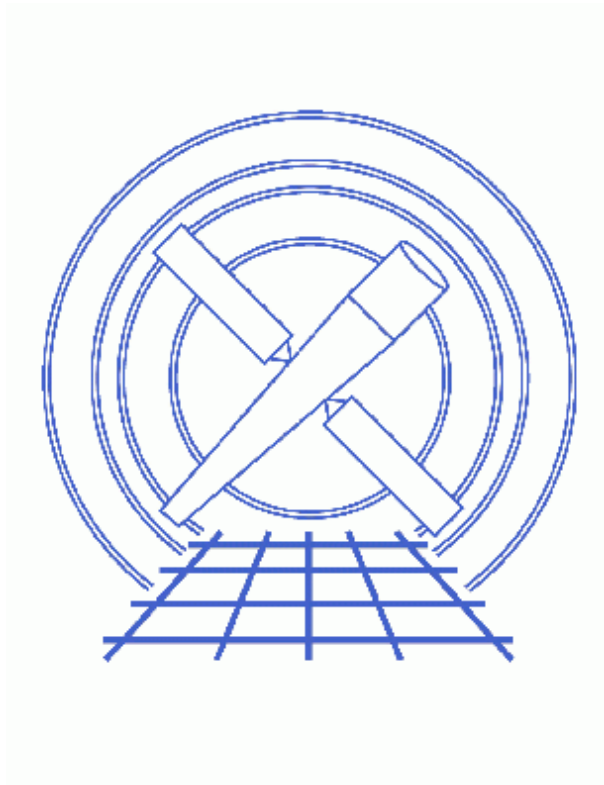


Calculating K-corrections using S-Lang and Sherpa



Sherpa Threads (CIAO 3.4)

Table of Contents

- Getting Started
- Setting up the spectral model
- Evaluate the model
 1. Define the energy grid of the model
 2. Display the model
- Calculate the k-correction
 - ◆ Using calc_kcorr()
 - ◆ How calc_kcorr() works
 - ◆ Displaying the models being evaluated
- Changing the model parameters
- Caveats
 1. Accuracy
- Summary
- History
- Images
 - ◆ Model spectrum
 - ◆ K-correction between $z=0$ and 2
 - ◆ Model spectrum after filtering on energy
 - ◆ Corrections for a family of models

Calculating K-corrections using S-Lang and Sherpa

Sherpa Threads

Overview

Last Update: 1 Dec 2006 – reviewed for CIAO 3.4: no changes

Synopsis:

In this thread we show you how you can use S-Lang together with *Sherpa* to calculate k corrections for any of the spectral models available from *Sherpa*. As an example we calculate the corrections necessary for the flux measured in the 0.5–2.0 keV energy range from a thermal plasma (as described by the Mewe–Kaastra–Liedahl – aka MEKAL – model) for a range of redshifts and gas temperatures. The aim is to emulate the figure presented in Appendix B of Jones et al. 1998, ApJ, 495, 100–114.

We will make use of the `calc_kcorr()` routine provided by the *Sherpa* utility S-Lang script (`sherpa_utils.sl`), part of the CIAO Scripts distribution.

Purpose:

When observing extra-galactic sources, the energies of the detected photons are $(1+z)$ times lower than the energy they had when they were emitted. This means that the flux – and hence luminosity – you measure within a certain pass-band on *Chandra* (or other satellite) does not match the same pass-band in the rest-frame of the object. In order to compare the luminosity of different objects the values should be corrected to a common rest-frame pass band; this correction is known as the "K correction".

Related Links:

- Appendix B of Jones et al. 1998, ApJ, 495, 100–114 compares the K-corrections for different temperatures of an optically thin thermal MEKAL model.
- The K correction by Hogg et al. is a short pedagogical paper which discusses the K correction (it discusses optical magnitudes rather than X-ray fluxes but the principles are the same).
- A large number of other papers discuss the topic of K-corrections; for instance Section IIa of Yoshii & Takahara, ApJ, 1988, 326, 1, Poggianti, A&AS, 1997, 122, 399, Sandage's "The Deep Universe" Lecture Notes and references therein.

Proceed to the HTML or hardcopy (PDF: A4 | letter) version of the thread.

Getting Started

This thread uses the `sherpa_utils.sl` script; for information about the script, consult the help file ("[ahelp sherpa_utils](#)"). The most recent version of `sherpa_utils.sl` is v1.26 (02 Nov 2004):

```
unix% grep Id $ASCDS_CONTRIB/share/slsh/local-packages/sherpa_utils.sl
% $Id: sherpa_utils.sl,v 1.26 2004/11/02 15:53:37 dburke Exp $
```

or, if the script is already loaded into *Sherpa*

```
sherpa> _sherpa_utils_version_string
1.26
```

Note that `$ASCDS_CONTRIB/share/slsh/local-packages/` is the default path in the standard CIAO scripts installation; see the [Scripts page](#) for more information. **Please check that you are using the most recent version before continuing.** If you do not have the script installed or need to update to a newer version, please refer to the [Scripts page](#).

The routines provided in the script can be loaded into *Sherpa* in a number of different ways, as discussed in the [Customizing Sherpa plots](#) thread. In the remainder of this thread it is assumed that the script has been loaded into *Sherpa* using one of these methods.

The routines will print out a "usage" message if called incorrectly. This provides an easy way to find out how to call the routine:

```
sherpa> calc_kcorr
Usage:
  kcorr = calc_kcorr( dnum, z, obslo, obshi [,restlo,resthi] )

where dnum is the number of the source model to use (an integer >=1)
z is a scalar or array (with values >=0), and the energies are scalars.

calc_kcorr;
sherpa>
```

If the corresponding `sherpa_utils.xml` has been placed into `$ASCDS_INSTALL/contrib/doc/xml/` and `ahelp -r` run, then "[ahelp calc_kcorr](#)" will also provide help on the routine.

Setting up the spectral model

In order to calculate the k-correction of a model we first have to define that model using the `SOURCE` command. In the following we use the Mewe–Kaastra–Liedahl thermal plasma model from the XSPEC library (`xsmekal`) and set the plasma temperature and metal abundance to be 7 keV and 0.3 times solar respectively.

```
sherpa> paramprompt_off
Model parameter prompting is off
sherpa> source = xsmekal[plasma]
sherpa> plasma.kt = 7
sherpa> plasma.abund = 0.3
sherpa> show_source 1
Source 1: plasma
xsmekal[plasma] (XSPEC model name: mekal) (integrate: on)
  Param  Type      Value      Min      Max      Units
```

Calculating K-corrections – Sherpa

1	kT thawed	7	1e-03	100	keV
2	nH frozen	1	1e-06	1e+20	per cm**3
3	Abund frozen	0.3	0	1000	
4	Redshift frozen	0	0	10	
5	Switch frozen	1	0	1	
6	norm thawed	1	0	1e+2410**-14 / (4 pi D**2)	Int n_e n_H dV

Since all we are interested in is the ratio of the flux within two pass bands the value of the normalisation is unimportant. However, it is important that the redshift be left at 0, and not changed to the value of your source.

Note: the routines in `sherpa_utils.sl` assume that the model is integrated over each energy bin, ie that `integrate` is set to on. This is true for all the XSPEC models, but not necessarily so for other models: the `show_source` command lists the value of the `integrate` flag for each model component, as shown above.

Evaluate the model

To calculate the k-correction, we need to access the model spectrum – ie flux in units of $\text{erg}/\text{cm}^2/\text{s}$ – for each energy bin.

Since all we need is the model spectrum *before* it enters the telescope, we do *not* need to set up an instrument model (ie supply an ARF or RMF). It is possible to use the `calc_kcorr()` command with a model that has been fitted to a dataset (ie an instrument model has been supplied) but it will only work if it covers the required energy range – the observed range times $(1.0+z_{\text{max}})$ – which may not be the case if using broad energy bands such as 2–10 keV.

1. Define the energy grid of the model

Since we are not fitting the model to a spectrum, *Sherpa* does not know what energy grid to calculate the model. We therefore make use of the DATASPACE command to set one up. It is important to ensure that the energy grid covers both the observed and rest-frame energy bands and that the HISTOGRAM option is supplied, since the XSpec models are integrated over each energy bin.

For the rest-frame energy range (0.5–2.0 keV) and redshift range (0–2) considered in this example, we need to ensure that the model is evaluated over at least 0.5 to 4.0 keV. It does not matter if the energy range is larger than that, so we chose 0.01 to 10 keV, with a grid spacing of 0.01 keV:

```
sherpa> dataspace (0.01:10:0.01) histogram
```

The *Sherpa* utilities package contains a simple function (`set_dataspace()`) that calls the `dataspace` command and ensures that the histogram mode is set. It is of more use from S-Lang scripts than from the *Sherpa* command line, but would be used like:


```
sherpa> set_dataspace(1,0.01,10,0.01)
```

The first parameter (here 1), gives the source/dataset number for which the energy grid is for, and the remaining parameters are the minimum, maximum, and grid step in keV.

2. Display the model

Here we use *Sherpa* commands to display the model and then modify the plot using *ChIPS* commands (since we are in *Sherpa* we need to issue a REDRAW command to display the final plot):

```
sherpa> set_log
sherpa> sherpa.multiplot.xfilter lo = 0.09
sherpa> lp source
sherpa> xlabel "Energy (keV)"
sherpa> ylabel "Spectrum (photon/cm^2/s)"
sherpa> title "XSMEKAL model: kT = 7 keV, Abundance = 0.3"
sherpa> xlabel size 1.4
sherpa> ylabel size 1.4
sherpa> title size 1.4
sherpa> redraw
```

This image  shows the resulting plot.

The `set_log` call ensures that both axes are plotted with a logarithmic scale. The `sherpa.multiplot.xfilter_lo` is used to limit all plots to have a minimum "x" value of 0.08.

Calculate the k-correction

The *Sherpa* utility package contains the `calc_kcorr()` routine which calculates the k-correction for a given spectral model, redshift, and energy bands. Below we show how to use this routine, and then discuss how the routine works.

Using `calc_kcorr()`

Now we have defined the source spectrum, we can use the `calc_kcorr()` S-Lang routine to calculate the k correction. So, if we want the k-correction for an object at redshift 0.4, we just say:

```
sherpa> calc_kcorr( 1, 0.4, 0.5, 2.0 )
0.838862
```

This is the k-correction for a flux measured in the observed 0.5–2.0 keV energy band to get the flux in the rest-frame 0.5–2.0 keV band for an object at a redshift of 0.4. If you want to convert to a different rest-frame energy band you just supply the band as extra parameters, as shown here for the rest-frame 0.1 to 2.4 keV energy range.

```
sherpa> calc_kcorr( 1, 0.4, 0.5, 2.0, 0.1, 2.4 )
1.35431
```

Instead of a single (ie scalar) value for the redshift, a one-dimensional array can be supplied, containing a range of redshifts. Here we calculate the correction values for the redshift range 0 to 2 with a step size of 0.1, and then plot it:

```
sherpa> z = [0.0:2.1:0.1]
sherpa> kc = calc_kcorr( 1, z, 0.5, 2.0 )
sherpa> clear
sherpa> () = curve(z,kc)
sherpa> symbol none
```

Calculating K-corrections – Sherpa

```
sherpa> simpleline
sherpa> xlabel "Redshift"
sherpa> ylabel "k correction"
sherpa> title "XSMEKAL model: kT = 7 keV, Abundance = 0.3"
sherpa> xlabel size 1.4
sherpa> ylabel size 1.4
sherpa> title size 1.4
sherpa> redraw
```

This  shows the resulting plot.

How `calc_kcorr()` works

The k-correction is calculated as the ratio of the flux in the rest-frame energy band to that in the observed band (for instance see [Appendix B](#) of Jones et al. 1998, ApJ, 495, 100–114). If we denote the low and high energies of the observed band as e_{lo} and e_{hi} respectively, then, for a source at redshift z , we need to calculate the flux between these bands:

- A. observed band: $e_{lo} * (1+z)$ to $e_{hi} * (1+z)$
- B. rest-frame band: e_{lo} to e_{hi}

Use the `get_eflux` command:

```
sherpa> erange = [0.5,2.0]
sherpa> z = 0.4
sherpa> rest_flux = get_eflux(1,erange)
sherpa> obs_flux = get_eflux(1,erange*(1+z))
sherpa> print(rest_flux)
dataset          = 1
range            = Double_Type[2]
comp             = NULL
value            = 5.57426e-10
units            = counts
sherpa> print(obs_flux)
dataset          = 1
range            = Double_Type[2]
comp             = NULL
value            = 6.64502e-10
units            = counts
sherpa> kcorr = rest_flux.value / obs_flux.value
sherpa> kcorr
0.838862
```

The `get_eflux()` routine returns a structure whose field `value` contains the source flux within the requested range – here 0.5–2.0 and 0.7–2.8 (0.5*1.4–2.0*1.4) keV for `rest_flux` and `obs_flux` respectively. Dividing the two values gives the k correction for this spectral model, redshift, and energy range.

Note that *Sherpa* is not able to always correctly guess the units of the models; here we have not set up any instrument model so `get_eflux()` is not able to work out that the source flux is in units of $\text{erg}/\text{cm}^2/\text{s}$ and so defaults to counts.

The `calc_kcorr()` routine contains additional code to check for errors, deal with an array of redshifts – rather than the scalar value used above – and to allow different rest-frame and observed-frame energy ranges to be used (e.g. to convert an observed flux in the 0.5–2.0 keV band to the rest-frame 0.1–2.4 keV band).


Displaying the models being evaluated

Here we show the source model over the energy ranges corresponding to the observed and rest-frame 0.5–2.0 keV band used here. It demonstrates how to use the *Sherpa* routines described in [ahelp_get](#) to read models into S-Lang variables and then display them.

```

sherpa> # set up models representing the rest & observed energy ranges
sherpa> source 2 = plasma
sherpa> source 3 = plasma
sherpa> dataspace 2 (0.5:2.0:0.01) histogram
sherpa> dataspace 3 (0.7:2.8:0.01) histogram
sherpa> # get the source data (ie the model amplitudes)
sherpa> rest = get_source(2)
sherpa> obs = get_source(3)
sherpa> rest
Double_Type[150]
sherpa> obs
Double_Type[210]
sherpa> # get the energy axis information
sherpa> rest_axis = get_photon_energy_axes(2)
sherpa> obs_axis = get_photon_energy_axes(3)
sherpa> print(rest_axis)
axistype      = Energy
axisunits     = keV
lo            = Double_Type[150]
hi            = Double_Type[150]
mid           = NULL
sherpa> print(obs_axis)
axistype      = Energy
axisunits     = keV
lo            = Double_Type[210]
hi            = Double_Type[210]
mid           = NULL
sherpa> # set up the plot
sherpa> clear
sherpa> chips.curvestyle = _chips->step
sherpa> chips.symbolstyle = _chips->none
sherpa> limits 0.4 3 0.04 0.8
sherpa> log y
sherpa> # display the two energy ranges (with a slight offset in y)
sherpa> () = curve( rest_axis.lo, rest*0.8 )
sherpa> red
sherpa> () = curve( obs_axis.lo, obs*1.2 )
sherpa> green
sherpa> # Tidy up the plot
sherpa> xlabel "Energy (keV)"
sherpa> ylabel Flux
sherpa> title "XSMEKAL model: kT = 7 keV, Abundance = 0.3"
sherpa> xlabel size 1.4
sherpa> ylabel size 1.4
sherpa> title size 1.4
sherpa> redraw

```

This image  shows the resulting plot: the models were offset by a factor of 20 per cent in flux for display purposes.

Changing the model parameters

In this section we show how the routine can be used to generate correction values for different model parameters (here the plasma temperature). The aim is to emulate Figure 7 of Jones et al. 1998, ApJ, 495, 100–114. We therefore need to calculate the corrections for plasma temperatures of 1, 2, 4, and 10 keV. The following commands are available as both a *Sherpa* script file ([plot_kcorrs.shp](#)) and a S-Lang script ([plot_kcorrs.sl](#)). Here we show the *Sherpa* script version:


```

sherpa> # what redshifts are we interested in?
sherpa> z = [0:2.01:0.01]
sherpa>
sherpa> # energy grid? Need to at least cover 0.5-2.0 keV for the range
sherpa> # z = 0 to 2
sherpa> #
sherpa> elo = 0.5
sherpa> ehi = 2.0
sherpa> dataspace (0.1:7.0:0.01) histogram
sherpa>
sherpa> # set the modeloverride flag so we can re-run this
sherpa> # script in the same session
sherpa> old override = sherpa.modeloverride
sherpa> sherpa.modeloverride = 1
sherpa>
sherpa> # set up the model
sherpa> #
sherpa> paramprompt off
sherpa> source 1 = xsmekal[plasma]
sherpa> plasma.abund = 0.3
sherpa>
sherpa> # kT = 1 keV
sherpa> plasma.kt = 1
sherpa> k1 = calc_kcorr( 1, z, elo, ehi )
sherpa>
sherpa> # kT = 2 keV
sherpa> plasma.kt = 2
sherpa> k2 = calc_kcorr( 1, z, elo, ehi )
sherpa>
sherpa> # kT = 4 keV
sherpa> plasma.kt = 4
sherpa> k4 = calc_kcorr( 1, z, elo, ehi )
sherpa>
sherpa> # kT = 10 keV
sherpa> plasma.kt = 10
sherpa> k10 = calc_kcorr( 1, z, elo, ehi )
sherpa>
sherpa> # we have an analytic solution for a power law
sherpa> kpl = (1+z)^-0.5;
sherpa>
sherpa> # plot the graphs
sherpa> # (could use 'redraw off' but want to reset the
sherpa> # the value at the end of the script so need to use the
sherpa> # S-Lang version)
sherpa> old redraw = chips auto redraw(0)
sherpa> clear
sherpa>
sherpa> ochips = @chips;
sherpa> chips.symbolstyle = _chips->none;
sherpa> chips.curvestyle = _chips->simpleline;
sherpa> () = curve( z, k1 );

```

Calculating K-corrections – Sherpa

```
sherpa> () = curve( z, k2 );
sherpa> () = curve( z, k4 );
sherpa> () = curve( z, k10 );
sherpa>
sherpa> chips.curvecolor = _chips->red;
sherpa> () = curve( z, kpl );
sherpa> dash
sherpa>
sherpa> xlabel 'Redshift'
sherpa> ylabel 'K_{0.5-2.0 keV}'
sherpa> title 'xsmekal model, abundance = 0.3 solar'
sherpa>
sherpa> xlabel size 1.5
sherpa> ylabel size 1.5
sherpa> title size 1.5
sherpa> tickvals size 1.5
sherpa>
sherpa> limits 0 2 0 3
sherpa>
sherpa> label 1.2 2.0 'kT = 1 keV'
sherpa> label 1.2 1.3 'kT = 2 keV'
sherpa> label 1.5 0.9 'kT = 4 keV'
sherpa> label 1.0 0.5 'kT = 10 keV'
sherpa>
sherpa> # display the plot
sherpa> redraw
sherpa>
sherpa> # restore settings
sherpa> () = chips_auto_redraw(old_redraw)
sherpa> sherpa.modeloverride = old_override
sherpa> set state( "chips", ochips )
sherpa>
sherpa> # finished
```

This image  shows the resulting plot, which can be compared to [Figure 7](#) of Jones et al. (1998).

Caveats

1. Accuracy

The accuracy of the k-correction is determined by the width of the energy bins. We shall illustrate this by using the [xspowerlaw](#) model, for which an analytic solution for the k-correction exists: for a photon index of γ , the k-correction is given by $k(z) = (1+z)^{\gamma-2}$. Below we calculate the corrections for two bin widths – 0.1 and 0.01 keV – and compare them to the analytic solution.

```
sherpa> # set up a model (XSPEC models always have integrate set to on)
sherpa> erase all
sherpa> paramprompt off
Model parameter prompting is off
sherpa> source 1 = xspowerlaw[p1]
sherpa> p1.phoindx = 3
sherpa> # set up the redshift range and the analytic solutions
sherpa> z = [0:1.1:0.2]
sherpa> kz = (1.0+z)^(3-2)
sherpa> # calculate the k-correction using two different energy grids
sherpa> dataspace (0.1:10:0.1) histogram
```

Calculating K-corrections – Sherpa

```
sherpa> kz1 = calc_kcorr( 1, z, 0.5, 2.0 )
sherpa> dataspace (0.1:10:0.01) histogram
sherpa> kz2 = calc_kcorr( 1, z, 0.5, 2.0 )
sherpa> # write out the results
sherpa> writeascii( stdout, z, kz, kz1, kz2 )
0      1      1      1
0.2    1.2    1.20158 1.20002
0.4    1.4    1.40296 1.40003
0.6    1.6    1.60421 1.60004
0.8    1.8    1.80538 1.80005
1      2      2.00649 2.00007
```

The output columns list the redshift, analytic solution, then the values calculated by `calc_kcorr()` for the two bin widths (0.1 and then 0.01 keV). As can be seen, the smaller grid spacing produces results closer to the true value. The choice of grid spacing depends on the model(s) you use and the desired accuracy: for instance, in this example the error introduced by using a bin width of 0.1 keV is likely to be much smaller than the error on the measured flux of the source or uncertainties in the model parameters used to calculate the *K* correction.

Summary

This thread has shown how you can use *S-Lang* and *Sherpa* to calculate the k-correction for any model that is known to *Sherpa*. The `sherpa_utils.sl` script provides the `calc_kcorr()` routine to help with this process.

History

14 Dec 2004 updated for CIAO 3.2: script version and path

21 Dec 2005 reviewed for CIAO 3.3: no changes

01 Dec 2006 reviewed for CIAO 3.4: no changes

URL: http://cxc.harvard.edu/sherpa/threads/calc_kcorr/

Last modified: 1 Dec 2006

Image 1: Model spectrum

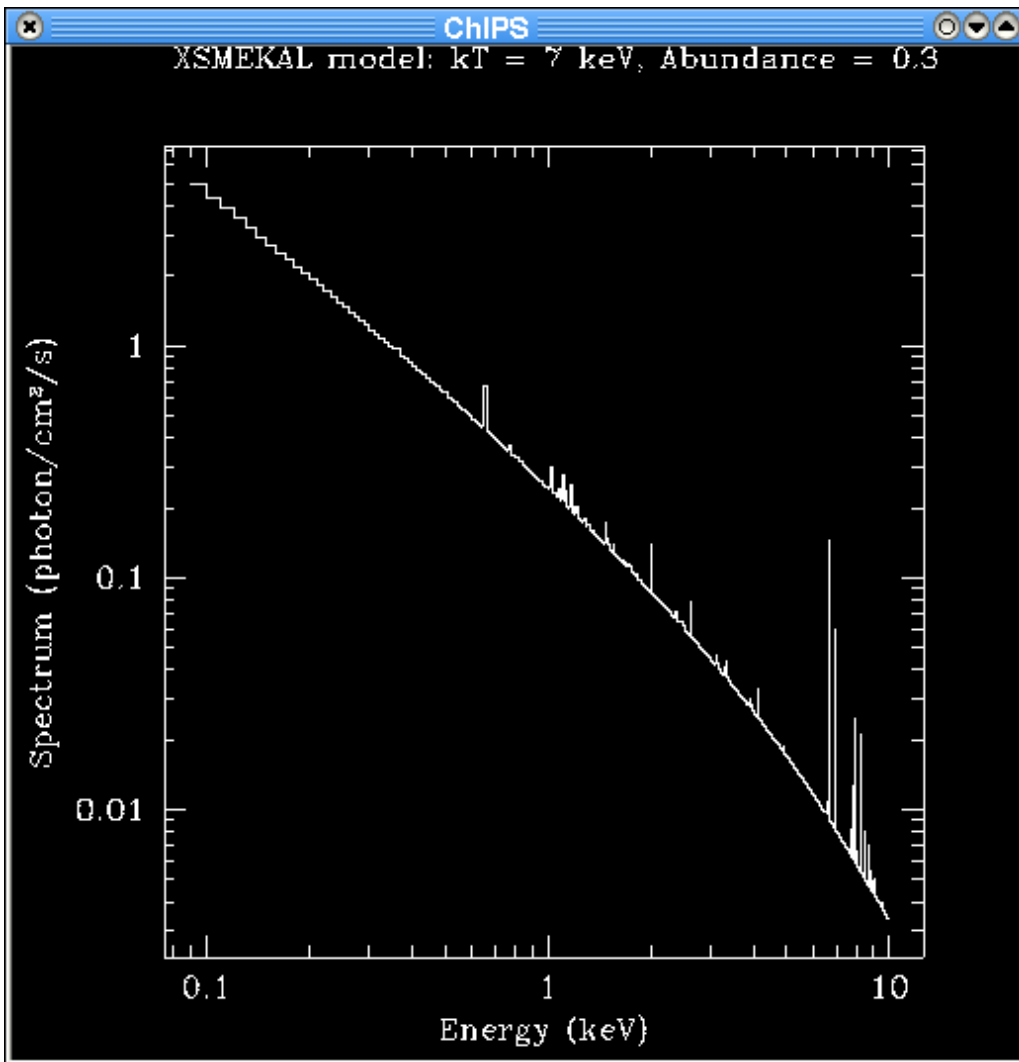


Image 2: K-correction between $z=0$ and 2

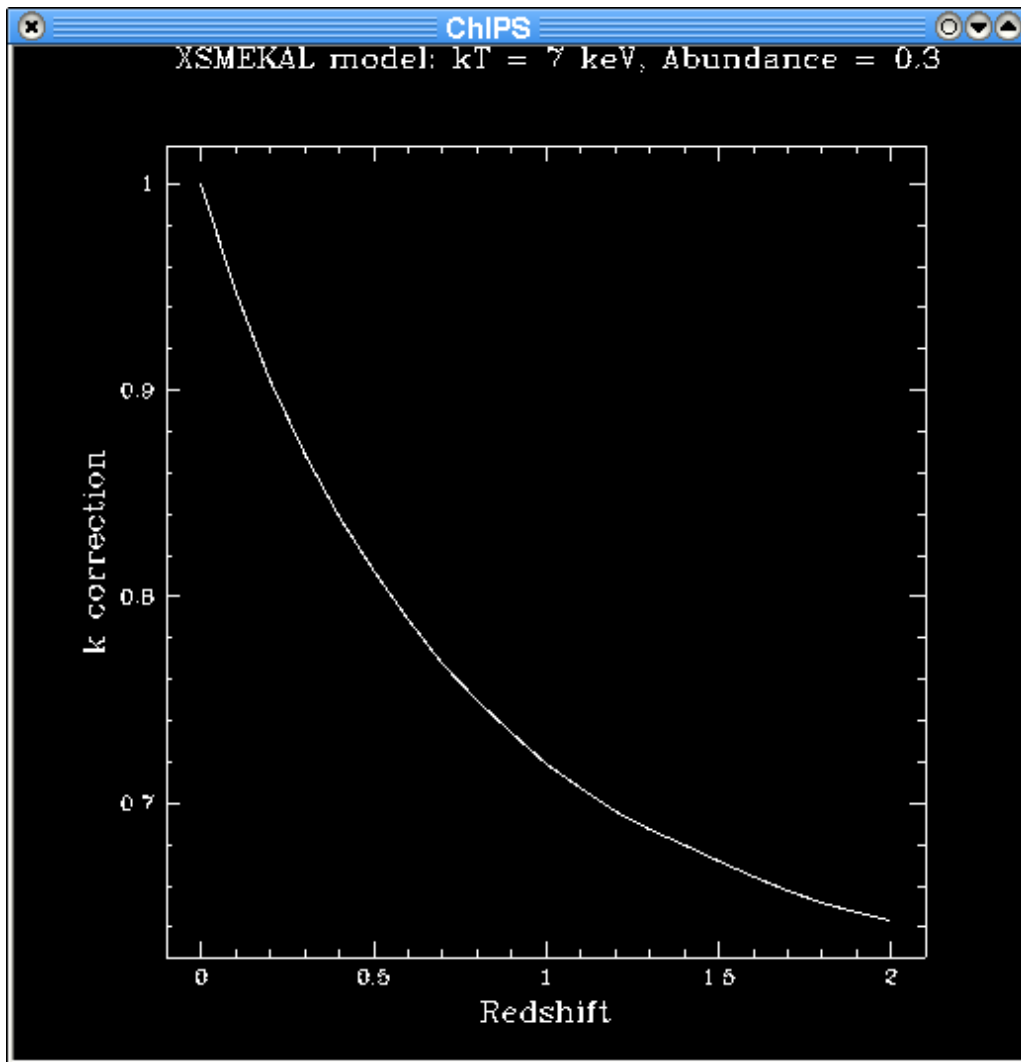
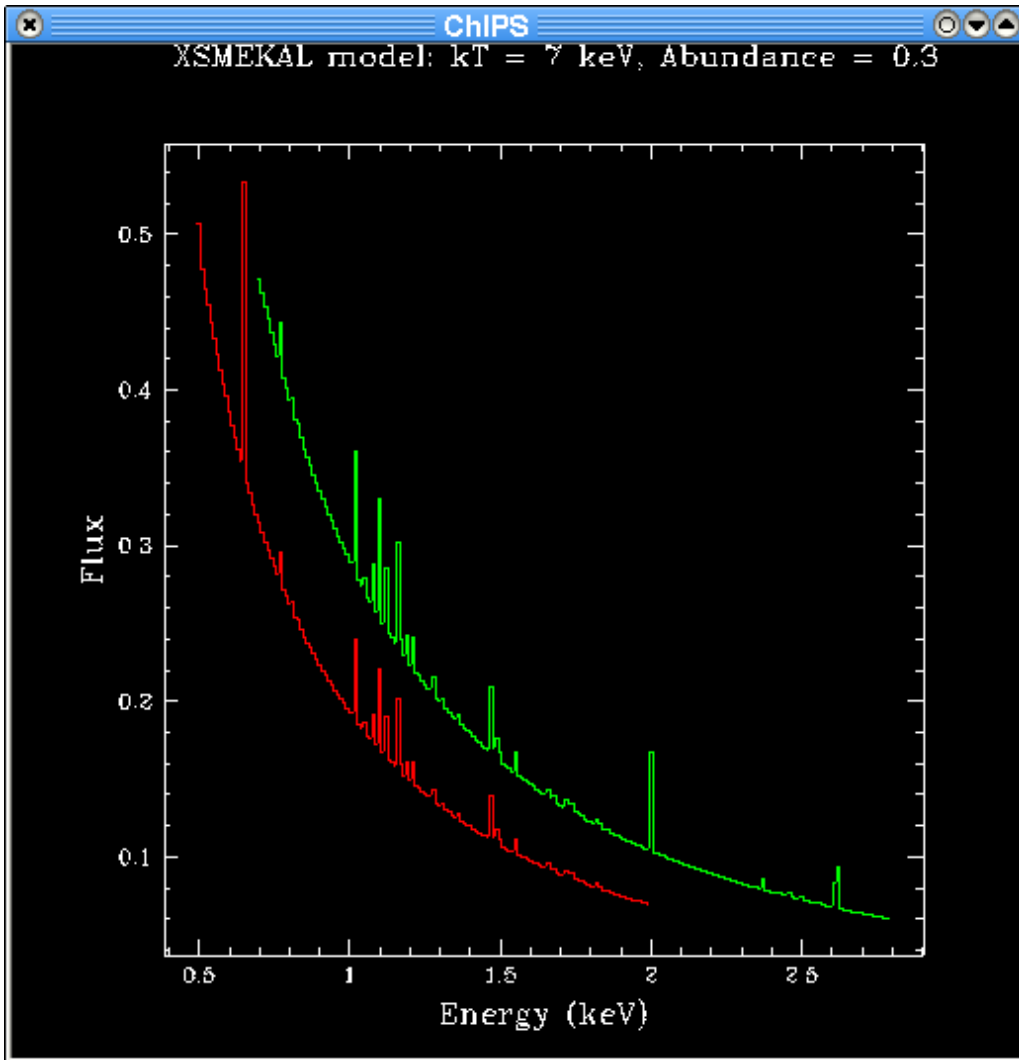


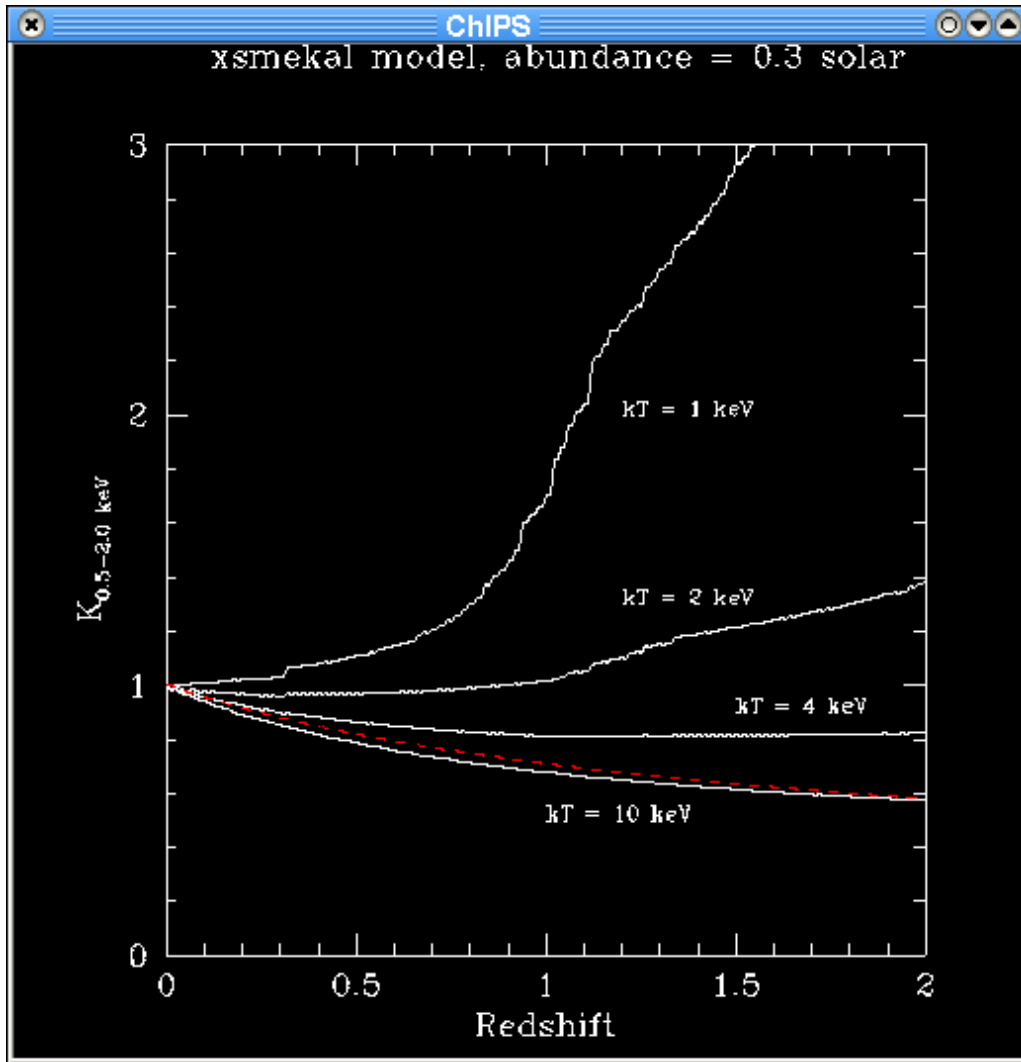
Image 3: Model spectrum after filtering on energy



The red and green lines show the source spectrum after filtering on the rest- and observed-frame energy ranges (they have been offset from each other by 20 per cent for clarity).

The model used here is the XSMEKAL plasma model with a temperature of 7 keV, metallicity abundance of 0.3 times solar, and a redshift of 0. The observed energy range is taken to be 0.5–2.0 keV and the source redshift (for calculating the rest-frame energy range) is 0.4.

Image 4: Corrections for a family of models



The models were chosen to be the same as used in [Appendix B](#) of Jones et al. 1998, ApJ, 495, 100–114.

Calculating K-corrections – Sherpa