Contents

Chapter 1

Introduction

1.1 XSPEC

XSPEC is a command-driven, interactive, X-ray spectral-fitting program. It is designed to be completely detector-independent so that it can be used for any spectrometer. To our knowledge, XSPEC has been used to analyze data from HEAO-1 A2, *Einstein Observatory*, EXOSAT, Ginga, ROSAT, BBXRT, ASCA, CGRO, and IUE. It also has been used for simulations for XTE and AXAF. This manual describes XSPEC v9.00, which runs under VAX/VMS, Alpha/OpenVMS, SunOS, Solaris 2.x, HP/Unix, DEC/Ultrix, Alpha/OSF, SGI, IBM/AIX, and NeXT (with the Oasys fortran compiler) systems.

The new user is advised to read Chapter 2, which introduces spectral fitting and the XSPEC approach, Chapter 3, which gives an overview of the program commands, and Chapter 4, which contains walk-throughs of XSPEC sessions. She then should experiment with XSPEC and, if necessary, look up individual commands in Chapter 5, or descriptions of the spectral models in use, in Chapter 6. Some of the FTOOLS that can operate on XSPEC files are listed in Chapter 7. XSPEC uses the PLT plotting package, which is described briefly in Appendix A and in more detail in the "QDP/PLT User's Guide" (Tennant, 1989). Users possessing X-ray spectra with small numbers of counts per bin are referred to Appendix B, which describes the C-statistic option. Users interested in adding their own models can read how to do so in Appendix C. The XSPEC/PLT parser is described in Appendix D and, finally, some programs that operate on the old-XSPEC format files are described in Appendix E.

1.2 New in Version 9

The following list summarizes the major changes to XSPEC since the previous edition of the user's manual.

- An `abund` command to switch between different Solar abundance tables.
- A `goodness` command to perform a Monte-Carlo goodness-of-fit calculation for non-background-subtracted data.
- A `systematic` command to apply a uniform systematic uncertainty to the model.
- An `untie` command to unlink two parameters that have been linked together.

- A `weight` command to change the weighting scheme used in the calculation of chi-squared. The new option is `gehrels` which uses $1 + \sqrt{N + 0.75}$, an approximation for small number statistics.

- The `step` command has a new option `step best` which resets non-stepped parameters to the best-fit after each grid point. The opposite is `step current`.

- `par1 = par2 + number` sets par1 to be equal to par2 plus an additive constant.

- `ignore` and `notice` work with energies if given real numbers as their arguments.

- Multiple spectra can be stored in the same FITS file and read individually. The command `data filenm{N}` reads spectrum N in filenm.pha.

- The behaviour of the `data` command is changed so that `data 1 filename` does not remove files 2–** but `data filename` does.

- `model none` removes all models.

- The following new models have been added and old ones improved:

  - An ionized absorber (`absori` from Magdziarz and Zdziarski) has been added.
  - A model (`ascac`) has been added for simultaneous analysis of ASCA spectra with corrections for energy-dependent telescope scattering.
  - Power-law differential emission models (`cemkl and cevmkl`) using the MEKAL code have been added.
  - A model (`comptt`) for Titarchuk's comptonization of a Wien law spectrum has been added.
  - Sixth-order Chebyshev polynomial differential emission models (`cp6mkl and cp6vmkl`) using the MEKAL code have been added.
  - MEKAL models (`mekal and vmekal`) have been added. These are new versions of the MEKA model using Fe L calculations supplied by Liedahl.
  - A pair plasma model (`nteea` from Magdziarz and Zdziarski) has been added.
  - Models of exponentially cut-off power-laws reflected from neutral matter and ionized matter (`pexrav, pexriv` from Magdziarz and Zdziarski).
  - A photo-electric absorption model (`phabs`) using the latest relative abundances and cross-sections has been added.
  - A model (`plabs`) has been added for absorption as a power-law in energy, useful for things like dust.
  - A model (`redge`) for recombination edge emission has been added.
  - A model of comptonized reflection from neutral matter (`tita_a` from Magdziarz and Zdziarski) has been added.
  - A model for redshifted absorption with variable iron abundance (`zvfeabs`) has been added.

- The plasma emission models now use relative Solar abundances as set by the `abund` command. The default is from Anders & Grevesse and the new option is from Feldman. The changes in this new abundance table are as follows : C : $3.63 \times 10^{-4} \rightarrow 3.98 \times 10^{-4}$, N : $1.12 \times 10^{-4} \rightarrow 1.00 \times 10^{-4}$, Ne : $1.23 \times 10^{-4} \rightarrow 1.29 \times 10^{-4}$, Ar : $3.63 \times 10^{-6} \rightarrow 4.47 \times 10^{-6}$, Fe: $4.68 \times 10^{-5} \rightarrow 3.24 \times 10^{-5}$.
- All absorption models except `wabs` and `zwabs` use the same relative abundances as the plasma emission models as set by the `abund` command.
- The `bremss`, `vbremss`, and `zbremss` now have a normalization which is not temperature-dependent.

- For local models the lmodel.dat file needs an extra line starting with the word "null" than before. See the example file in the release.
- A number of changes have been made to table models.
  - All table models now use a FITS format. The program `ctable` converts old format table models into FITS.
  - Multiple table models can be used simultaneously.
  - The Raymond-Smith model now runs off a table in the standard table model FITS format.
  - An `etable` model reads an additive table model type file and then takes the exponential of the negative of the input.
  - The `mtable` model now defaults to 1 rather than 0 when there is no tabulated model for part of the response energy range.
- The C-statistic has a new normalization suggested by John Castor which provides a goodness-of-fit.
- All vectors for a given dataset are plotted in the same colour.
- XSPEC can be called as a subroutine (through xspec/src/subr/xscmnd.f)
- Model functions can give a result depending on which file the model is being calculated for. **This has meant a change in the subroutine interface, the variable WORKSP has been removed and a new integer IFL has been included - see files in xspec/src/functions for examples.** For most models IFL will not be used but must still be in the subroutine call.
- All VMS Fortran structures have been removed from the XSPEC source code.
- A few bugs have been fixed.
  - The *gain fit* command is a little more robust (but should still be used with care). It now handles features in the detector efficiency curve correctly provided that they are included in the arf or no arf is used and only a response matrix is given.
  - There was a minor error when calculating errors on grouped data when both the data and background contained bins with zero counts.
  - The wavelength plots were in /keV rather than /A as claimed.
  - `ignore bad` now works for arbitrarily many ignore ranges.

1.3 How to contact us

XSPEC is now distributed and maintained under the aegis of the GSFC High Energy Astrophysics Science Archival Research Center (HEASARC). If you have any problems with XSPEC please e-mail us at **xanprob@athena.gsfc.nasa.gov**. If you want to be added to our electronic mailing list for announcements of new releases then send mail to **listserv@athena.gsfc.nasa.gov** with the main body of the message comprising the line **subscribe XANADU Your Name**.

XSPEC and the other parts of the XANADU package are available by anonymous *ftp* from **legacy.gsfc.nasa.gov** in **software/xanadu** or by DECnet copy in **lheavx::casper$dka500:[xandist.savesets]** (this site may change so contact us if you try to DECnet copy and fail).

1.4 Relationship to FTOOLS

We are attempting to move to using FITS format files for all XSPEC input and output files. In particular, XSPEC now fully supports FITS format spectral and response files. The FTOOLS package (also available by anonymous *ftp* from **legacy.gsfc.nasa.gov**) contains a large number of tools for manipulating FITS format files, including a number of specialised tools for spectral and response files. Some of these tools are listed in Chapter 7.

1.5 History

The first version of XSPEC was written in 1983 at the Institute of Astronomy, Cambridge, under VAX/VMS by Rick Shafer. It was written to perform spectral analysis of data from the ESA EXOSAT X-ray observatory, which was launched that year. XSPEC is a descendant of a series of spectral-fitting programs written at NASA/Goddard Space Flight Center for the OSO-8, HEAO-1 and *Einstein Observatory* missions. The initial design was the fruit of many discussions between Rick Shafer and Andy Szymkowiak.

Rick Shafer subsequently joined the EXOSAT group, where he enhanced the VAX/VMS version in collaboration with Frank Haberl. Meanwhile, Keith Arnaud ported XSPEC to a Sun/UNIX operating system. The two implementations of XSPEC diverged for several years until a determined and coordinated effort was made to recover a single version. The results of that effort was XSPECv6, described in the first edition of this manual. Subsequent editions have covered later versions of the program.

In recent years XSPEC has become the *de facto* standard for X-ray spectroscopic analysis for the ROSAT mission and the *de jure* standard for the BBXRT, ASCA, and XTE missions. The HEASARC, which has supported the developement of XSPEC since 1990, intends to use XSPEC for future missions as well as for those whose data are being resurrected.

1.6 Acknowledgements

This project would not have been possible without ignoring the advice of far more people than can be mentioned here. We would like to thank all our colleagues for their suggestions, bug reports, and (especially) source code. The GSFC X-ray astronomy group are particularly thanked for their patience exhibited while functioning as the beta test site for new releases. The initial developement of XSPEC was funded by a Royal Society grant to Prof. Andy Fabian and subsequent developement was partially funded by the European Space Agency's EXOSAT project and now by the NASA/GSFC HEASARC.

Chapter 2

Spectral Fitting and XSPEC

2.1 Introduction

This chapter is comprised of a brief description of the basics of spectral fitting, their application in XSPEC, and some helpful hints on how to approach particular problems.

2.2 The Basics of Spectral Fitting

Although we use a spectrometer to try to find out the spectrum of a source, what the spectrometer obtains is not the actual spectrum, but rather photon counts (C) within specific instrument channels, (I). This observed spectrum is related to the actual spectrum of the source (f(E)), such that:

$$C(I) = \int_0^\infty f(E)R(I,E)dE$$

where R(I,E) is the instrumental response and is proportional to the probability that an incoming photon of energy E will be detected in channel I. Ideally, then, we would like to determine the actual spectrum of a source, f(E), by inverting this equation, thus deriving f(E) for a given set of C(I). Regrettably, this is not possible in general, as such inversions tend to be non-unique and unstable to small changes in C(I). (For examples of attempts to circumvent these problems see Blissett & Cruise 1979; Kahn & Blissett 1980; Loredo & Epstein 1989).

The usual alternative is to try to choose a model spectrum, f(E), that can be described in terms of a few parameters (i.e., f(E,p1,p2,...)), and match, or "fit" it to the data obtained by the spectrometer. For each f(E), a predicted count spectrum ($C_p(I)$) is calculated and compared to the observed data (C(I)). Then a "fit statistic" is computed from the comparison, which enables one to judge whether the model spectrum "fits" the data obtained by the spectrometer.

The model parameters then are varied to find the parameter values that give the most desirable fit statistic. These values are referred to as the *best-fit parameters*. The model spectrum, $f_b(E)$, made up of the best-fit parameters is considered to be the *best-fit model*.

The most common fit statistic in use for determining the "best-fit" model is χ^2, defined as follows:

$$\chi^2 = \sum (C(I) - C_p(I))^2/(\sigma(I))^2$$

where $\sigma(I)$ is the error for channel I (e.g., if C(I) are counts then $\sigma(I)$ is usually estimated by $\sqrt{C(I)}$; see e.g. Wheaton et.al. 1995 for other possibilities).

Once a "best-fit" model is obtained, one must ask two questions:

1. First, one must ask how confident one can be that the observed C(I) can have been produced by the best-fit model $f_b(E)$. The answer to this question is known as the "goodness-of-fit" of the model.

   The χ^2 statistic provides a well-known goodness-of-fit criterion for a given number of degrees of freedom (ν, which is calculated as the number of channels minus the number of model parameters) and for a given confidence level. If χ^2 exceeds a critical value (tabulated in many statistics texts) one can conclude that $f_b(E)$ is *not* an adequate model for C(I). As a general rule, one wants the "reduced χ^2" (χ^2/ν) to be approximately equal to one ($\chi^2 \sim \nu$). A reduced χ^2 that is much greater than one indicates a poor fit, while a reduced χ^2 that is much less than one indicates that the errors on the data have been over-estimated.

   Even if the best-fit model ($f_b(E)$) *does* pass the "goodness-of-fit" test, one still cannot say that $f_b(E)$ is the *only* acceptable model. For example, if the data used in the fit are not particularly good, one may be able to find many different models for which adequate fits can be found. In such a case, the choice of the correct model to fit is a matter of scientific judgement.

2. Second, for a given best-fit parameter (p1), one must determine the range of values within which one can be confident the true value of the parameter lies. The answer to this question gives one the "confidence interval" for the parameter.

   The confidence interval for a given parameter is computed by varying the parameter value until the χ^2 increases by a particular amount above the minimum, or "best-fit" value.

   The amount that the χ^2 is allowed to increase (also referred to as the critical $\Delta\chi^2$) depends on the confidence level one requires, and on the number of parameters whose confidence space is being calculated. The critical $\Delta\chi^2$ for common cases are given in the following table (from Avni, 1976):

Confidence	Parameters		
	1	2	3
0.68	1.00	2.30	3.50
0.90	2.71	4.61	6.25
0.99	6.63	9.21	11.30

   There is a good discussion of confidence ranges in Press *et al.*, (1992) for readers who want more details[1].

[1] The precise reader should note that it is not correct to refer to a confidence interval as giving the probability of the true parameter value being in a given range. Such a statement is not possible in classical statistics. What the confidence interval actually shows is the probability that a new observation would give a certain parameter value, assuming that the best-fit value obtained for the current observation is the true value of the parameter

2.3 The XSPEC implementation

To summarize the preceding section, the main components of spectral fitting are as follows:

1. The observed spectrum (C(I))
2. The instrumental response (R(I,E))
3. A model spectrum (f(E))

These components are used in the following manner:

1. A parameterized model is created that one feels represents the actual spectrum of the source.
2. One then gives values to the model parameters.
3. Based on the parameter values given, one predicts the count spectrum that would be detected by the spectrometer in a given channel for such a model.
4. Then, one compares the predicted spectrum to the spectrum actually obtained by the instrument.
5. The values of the parameters of the model are manipulated until one finds the best fit between the theoretical model and the observed data.
6. One then calculates the "goodness" of the fit to determine how well the model explains the observed data, and calculates the confidence intervals for the model's parameters.

This section describes how XSPEC performs these tasks.

2.3.1 C(I): The Observed Spectrum

To obtain the observed spectrum (C(I)), for a given observation, XSPEC uses two files: the data file, and the background file (for FITS file format see Arnaud, George & Tennant 1992[2]). The data file tells XSPEC how many total photon counts were detected by the instrument in a given channel. XSPEC then uses the background file to derive a background-subtracted C(I) in units of counts per second. The background is scaled to the data by the ratio of the BACKSCAL keywords in the data and background files. It also is scaled for exposure times (EXPOSURE keyword). The background-subtracted count rate is given by :

$$C(I) = D(I)/t_D - (b_D/b_B)B(I)/t_B$$

where $D(I)$ and $B(I)$ are the counts in the data and background files; t_D and t_B are the exposure times in the data and background files; and b_D and b_B are the BACKSCAL values from the data and background files.

When this is done XSPEC has an observed spectrum to which the model spectrum can be fit.

[2]The FTOOLS distribution includes in the directory **callib/src/gen** the subroutines `rdpha1.f` and `wtpha1.f`, which can be used in programs reading and writing FITS-format spectral files

2.3.2 R(I,E): The Instrumental Response

Before XSPEC can take a given set of parameter values and predict the spectrum that would be detected by a given instrument, XSPEC must know the specific characteristics of the instrument. This information is known as the **detector response**. The response (R(I,E)), if you recall, is proportional to the probability that an incoming photon of energy E will be detected in channel I. As such, the response is a continuous function of E. This continuous function is converted to a discrete function by the creator of a **response matrix** who defines the energy ranges (E_J) such that:

$$R_D(I,J) = \int_{E_{J-1}}^{E_J} R(I,E)dE$$

XSPEC reads both the energy ranges, E_J, and the response matrix ($R_D(I,J)$) from a **response file** (for FITS file format see George et al 1992[3]) in a compressed format that only stores non-zero elements. XSPEC also includes an option to use an **auxiliary response file** (George et al 1992[4]), which contains an array $A_D(J)$ that XSPEC multiplies into $R_D(I,J)$ as follows:

$$R_D(I,J) \rightarrow R_D(I,J) * A_D(J)$$

Conventionally, the response is in units of cm^2.

2.3.3 f(E): The Model Spectrum

The model spectrum, f(E), is calculated within XSPEC using the energy ranges defined by the response file :

$$f_D(J) = \int_{E_{J-1}}^{E_J} f(E)dE$$

and is in units of photons/cm^2/s. XSPEC allows the construction of composite models consisting of additive components (e.g., power-laws, blackbodys, and so forth), and multiplicative components, which modify additive components by an energy-dependent factor (e.g., photoelectric absorption, edges, ...). Models can be defined in algebraic notation. For example, the following expression:

$$\mathrm{phabs}(\mathrm{power} + \mathrm{phabs}(\mathrm{bbody}))$$

defines an absorbed blackbody (phabs (bbody)) added to a power-law (power). The result then is modified by another absorption component (phabs).

For a more detailed explanation of models, see Chapter 6.

---

[3]The FTOOLS distribution includes in the directory **callib/src/gen** the subroutines `rdebd1.f`, `rdrmf1.f`, `wtebd1.f`, `wtrmf1.f`, which can be used in programs reading and writing FITS format response files

[4]The FTOOLS distribution includes in the directory **callib/src/gen** the subroutines `rdarf1.f` and `wtarf1.f`, which can be used in programs reading and writing FITS format auxiliary response files

2.3.4 Fits and Confidence Intervals

Once data have been read in and a model defined, XSPEC uses a modified Levenberg-Marquardt algorithm (based on CURFIT from Bevington, 1969) to find the best-fit values of the model parameters. The algorithm used is a local one, so the user should be aware that it is possible for the fitting process to get stuck in a local minimum and not find the global best-fit. The process also goes much faster (and is more likely to find the true minimum) if the initial model parameters are set to sensible values.

At the end of a fit, XSPEC will write out the best-fit parameter values, along with estimated confidence intervals. These confidence intervals are one sigma and are calculated from the derivatives of the fit statistic with respect to the model parameters. However, the confidence intervals are not reliable and should be used for indicative purposes only.

XSPEC has separate commands (`error` or `uncertainty`) to derive confidence intervals for one interesting parameter, which it does by fixing the parameter of interest at a particular value and fitting for all the other parameters. New values of the parameter of interest are chosen until the appropriate delta-statistic value is obtained. XSPEC uses a bracketing algorithm followed by an iterative cubic interpolation to find the parameter value at each end of the confidence interval.

To compute confidence regions for several parameters at a time XSPEC runs a grid on these parameters. XSPEC also will display a contour plot of the confidence regions of any two parameters.

2.4 Helpful Hints

2.4.1 Scripts and the Save command

One can write XSPEC commands into a file and have them executed by entering `filename` at the XSPEC prompt. The `save` command writes out a file of XSPEC commands, which, when run, will restore XSPEC to the state it was in when the `save` command was issued. This command is very useful when reading a large number of data sets and/or fitting complicated models. One also should do a `save` after doing something new or complicated, so if a disaster occurs it is possible to recover.

2.4.2 How to return to the XSPEC¿ prompt

The string /* acts as an emergency escape back to the XSPEC prompt. If one gives this string in answer to any question then one should be bounced out of whatever one is doing and returned to the XSPEC prompt. Counterexamples to this should be sent to xanprob@athena.gsfc.nasa.gov.

2.4.3 Simulations

If one has a response file for an instrument and wants to make some fake spectra, then the command `fakeit none` should be used. XSPEC will prompt the user for the response filename from which to build the simulated data. It is important to note that one must define a model prior to issuing the `fakeit` command.

2.4.4 Plotting Devices and Hardcopy

XSPEC uses the PGPLOT package, which comes with a standard set of device drivers. Any X-windows terminal should support /xw while xterm windows should support /xt.

The easiest way of making a hardcopy of an XSPEC plot is to use the `iplot` command and then at the PLT prompt to enter `@hard`. This will make a file called `pgplot.ps` and send it to the default printer[5].

2.4.5 Data groups

The most common use of XSPEC is to fit one or more data sets with responses to a particular model. However, it is often useful to be able to fit simultaneously several data sets with a model whose parameters can be different for each data set. A simple example would be a number of data sets that we expect to have the same model spectrum shape but different normalizations. XSPEC caters to this need through the use of data groups. When files are read in they can be labelled as belonging to a particular data group. When a model is defined a set of model parameters is allocated for each data group. These parameters can all vary freely or they can be linked together across data groups as required.

To set up data groups, the `data` command should be given as in the following example :

```
XSPEC> data 1:1 file1 1:2 file2 2:3 file3
```

which sets up two data groups. The first data group comprises data sets from file1 and file2, and the second data group takes the data set from file3. Now when a model is defined, XSPEC will give two sets of model parameters, one for the first datagroup and one for the second.

2.4.6 What to do when you have small numbers of counts

The χ^2 statistic assumes that all the spectral channels are Gaussian distributed. If there are small numbers of counts in a channel this will not be true. An alternative statistic, the C-statistic, can be used in this case provided that the data has not been background subtracted. The C-statistic now in use does provide a goodness-of-fit and this can be checked using the `goodness` command which uses Monte Carlo methods. This C-statistic can also provide confidence intervals in exactly the same way as χ^2. So give the command :

```
XSPEC> stat cstat
```

and then use the `fit` and `error` commands as usual. An alternative approach is to continue using the χ^2 statistic but change the weighting to provide a better approximation to a Gaussian distribution in the small number limit. This can be done using the `weight gehrels` command.

2.4.7 Binning and Grouping data

Often one does not want to use the full resolution of a spectrum, either because the channels oversample the spectral resolution or because the S/N is low. XSPEC and the associated programs provide a number of ways of handling this. Firstly, the XSPEC

[5]If this fails ask your local XSPEC manager to fix the file `$XANADU/lib/xcoms/hard.pco`

`setplot rebin` command can be used to add channels together in the plot. It is important to realise that this effects only the plot and not the data being fitted. Two FTOOLS are available to bin and group data for fitting purposes. RBNPHA bins up the data in a non-reversible manner and should only be used to ensure that the number of bins in the spectrum is the same as that in the response. GRPPHA is the tool of choice for grouping the data to get adequate S/N or number of counts in each channel. GRPPHA does not actually add together channels, but instead sets a flag which is read by XSPEC and causes XSPEC to sum the appropriate channels. If a data file is read with some grouping then XSPEC will apply the same operation to any background or response files used.

2.4.8 How to shut XSPEC up (somewhat)

The `fit` command in XSPEC will run a certain number of iterations and then query the user whether he or she wants to continue. While useful under most circumstances, the constant questioning can be a pain if one leaves XSPEC running and expects to find it finished when one gets back, or if one habitually runs scripts. One way around this problem is to reset the number of iterations before the question is asked by giving a single argument on the `fit` command. For example, `fit 100` will run 100 iterations before asking a question. A more drastic solution is to use the `query` command. `query yes` will suppress all questions and assume that their answer is yes while `query no` will suppress all questions but assume that their answer is no.

2.4.9 References

Arnaud, K.A., George, I.M., Tennant, A.F., 1992. Legacy, 2, 65.
Avni, Y., 1976. ApJ, 210, 642.
Bevington, P.R., 1969. Data Reduction and Error Analysis for the Physical Sciences, McGraw-Hill.
Blissett, R.J., Cruise, A.M., 1979. MNRAS, 186, 45.
George, I.M., Arnaud, K.A., Pence, W., Ruamsuwan, L., 1992. Legacy, 2, 51.
Kahn, S.M., Blissett, R.J., 1980. ApJ, 238, 417.
Loredo, T.J., Epstein, R.I., 1989. ApJ, 336, 896.
Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes (2nd edition), p687ff, CUP.
Wheaton, W.A. et.al., 1995. ApJ, 438, 322.

Chapter 3

An overview of XSPEC

3.1 Command syntax

XSPEC is a command-driven, interactive program. The user will see a prompt whenever input is required. If the user is uncertain about how to respond to a particular prompt, typing `?` usually will produce a list of valid responses.

XSPEC is designed to allow complicated, multi-instrument analysis, so most commands can take arguments specifying more than one data set. Arguments in XSPEC may be separated by either blanks or commas.

A single argument can define a range. The ranges are delimited by two dashes (--). A colon (:) is used to separate ranges (e.g., the phrase 1--2:11--24 refers to channels 11 – 24 in files 1 and 2).

Operating-system commands can be given from within XSPEC by prefixing the command with dollar sign ($). A set of XSPEC commands in a file can be input using `@filename`. This command file can take arguments. Note that all commands, with the exception of `fakeit`, can be abbreviated to unique strings. Command recall and inline editing are available and configurable by the user. The `alias` command enables a user to define his or her own command in terms of other XSPEC commands.

3.2 Reading data

XSPEC reads in spectra from spectral files using the `data` command. Several datasets may be specified in one command. Several datasets may be stored in a single file and accessed separately. A particular data set in use may be replaced by another or dropped entirely. The input data file contains pointers to background, redistribution and auxiliary response files, but these pointers may be overridden by the `back`, `resp`, and `arf` commands. All these commands have the same syntax as `data`. PHA channels may be left out of fitting using the `ignore` command and included again using the `notice` command. These commands have a syntax allowing the same channels to be specified for more than one input file. The ignore and noticed ranges can be specified either as channels or as energies. An auxiliary background file, called the correction file, also can be included using the `corfile` command. Its use is described in the section on fitting.

3.3 Defining models

XSPEC allows users to fit data with models constructed from individual components. These components may be either additive, multiplicative, or mixing. Multiplicative components multiply the model by an energy-dependent factor. Mixing components link models for different datagroups. Multiplicative components can act on individual components, on groups of components, or on the entire model. The `model` command defines the model to be used and prompts for the starting values of its parameters. The user also can set the allowed range of the parameter. One parameter can be fixed to be a multiple of another or to be another plus a constant. Parameters can be unlinked using the `untie` command.

The value of an individual parameter can be changed with the command `newpar`. Note that `newpar 0` will print out the current parameter values. Parameters can be fixed at their current value with the `freeze` command and allowed to vary freely with the `thaw` command. Individual components can be added or subtracted from the model using `addcomp` and `delcomp`.

3.4 Fitting

The basic fit command is called `fit`. This command performs a minimization using the Levenberg-Marquardt algorithm. `fit` can take two arguments, those being 1) the number of iterations to be performed before the user is asked whether to continue; and 2) the change in fit statistic that defines convergence. The fit statistics available within XSPEC are the χ^2 and C statistics: the `statistic` command specifies which one is to be used. Other fit-minimization algorithms are available, and can be selected using the `method` command. For non-background-subtracted data the `goodness` command does a Monte Carlo calculation of the goodness-of-fit. The weighting algorithm used to calculate χ^2 can be altered by the `weight` command. A systematic model uncertainty can be included using the the `systematic` command.

The `error` or `uncertain` command calculates error bounds for one interesting parameter for the specified parameters and confidence levels. To produce multi-dimensional errors the `steppar` command is used to generate a fit-statistic grid. Two-dimensional grids may be expressed as contour plots. The model normalization can be set using the `renorm` command. The normalization of the correction file background can be set with `cornorm`, and can be set to minimize the fit statistic with `recornorm`. The gain of the response matrix can be adjusted using the `gain` command, which includes an option to fit for the gain.

3.5 Plotting

XSPEC plotting is performed using the PLT interface. There are two basic commands: `plot` and `iplot`. The `plot` command makes a plot and returns the user to the XSPEC prompt, while the `iplot` command leaves the user in the interactive plotting interface, thus allowing the user to edit the plot. A variety of different quantities may be plotted, including the data and the current model; the fit residuals; the ratio of data to model; the contributions to the fit statistic; the theoretical model; the unfolded (incident) spectrum; the detector efficiency; and the fit- statistic contours. All data plots can have an x-axis of channels, energy, or wavelength, which are specified with `setplot channel`, `setplot energy`, `setplot wavelength` respectively. The plotting device to be used is set using `setplot device` or `cpd`. Separate spectra may be added together and channels binned up (for plotting purposes only) using `setplot group` and `setplot rebin`. There is an option to plot individual additive model components on data plots, this option is enabled by `setplot add` and disabled by `setplot noadd`. A stack of PLT commands can be created and manipulated with `setplot command`, `setplot delete`, and `setplot list`. This command stack then is applied to every plot. A plot can be spooled to the printer by issuing the command `@hard` at the PLT prompt.

3.6 Getting help

XSPEC has an interactive help facility that is accessed using the `help` command. Help on individual commands is available using the `help model` command and on models using `help model`. If you need further help or want to send us a suggestion then mail to xanprob@athena.gsfc.nasa.gov.

3.7 Simulated data

The `fakeit` command is used to generate simulated data. The current response matrix and model (a model must be defined prior to using the fakeit command) are used to create fake data. The user is prompted for various options. To make fake data when only a response matrix is available, give the command `fakeit none`.

3.8 Fluxes, luminosities, and equivalent widths

The `flux` command calculates the flux from the current model in the given energy range. This energy range should be within that defined by the current response matrix. If a larger energy range is required, then the `dummyrsp` command should be given. If this command is used, the user should reset the response matrix using `resp` before returning to fitting, or incorrect results will be generated.

The `lumin` command calculates the luminosity for the source redshift given. The cosmological parameters used to calculate luminosity are changed with the `cosmo` command. The `eqwidth` command determines the equivalent width of a model component, usually a line. The command `thleqw` calculates the expected equivalent width for a fluorescence line of specified yield. The user of either of these last two commands should read the help descriptions carefully.

3.9 Miscellaneous

XSPEC retains a memory of the last 20 commands. These commands can be reviewed or rerun using the `reexecute` command. A log of the session can be written to an ASCII file using the `log` command.

A more complete description of the session can be written to a binary file using the `history` command. The `dump` command writes the current data and model into the history file. The amount of output that XSPEC writes is set by the `chatter` command, which takes two arguments applying to the terminal and to the log file.

Information on the current XSPEC status can be printed out using the `show` command. The `save` command writes the current XSPEC status to a command file, which later can be run to reset XSPEC to the same configuration. During the fitting procedure, XSPEC sometimes will ask the user whether he or she wishes to continue. This question can be suppressed with the `query` command. The `time` command writes out system-timing information. Finally, XSPEC can be terminated with the `exit` or `quit` commands.

Chapter 4

Walks through XSPEC

4.1 Introduction

This chapter demonstrates the use of XSPEC. The brief discussion of data and response files is followed by fully-worked examples using real data that include all the screen input and output with a variety of plots. The topics covered are as follows: defining models, fitting data, determining errors, fitting more than one set of data simultaneously, simulating data, and producing plots.

4.2 Brief Discussion of XSPEC Files

At least two files are necessary for use with XSPEC: a data file and a response file. In some cases, a file containing background may also be used, and, in rare cases, a *correction* file is needed to adjust the background during fitting. However, most of the time the user need only specify the data file, as the names and locations of the correct response and background files should be written in the header of the data file by whatever program created the files.

4.3 Fitting Models to Data: An Example from EXOSAT

The 6-s X-ray pulsar 1E1048.1–5937 was observed by *EXOSAT* in June 1985 for 20 ks. In this example, we'll conduct a general investigation of the spectrum from the Medium Energy (ME) instrument, i.e. follow the same sort of steps as the original investigators (Seward, Charles & Smale,1986). The ME spectrum and corresponding response matrix were obtained from the HEASARC On-line service.

Once installed, XSPEC is invoked by typing `xspec`, as in this example on **rosserv**, a

Sun/4 workstation. The first command we issue is `data` followed by the name of the data file, `e1048_me.pha`:

```
rosserv% xspec

                XSPEC 9.00 17:34:45 12-Jul-95

  Plot device not set, use "cpd" to set it

  Type "help" or "?" for further information
 XSPEC> data e1048_me.pha
 Net count rate (cts/cm^2/s) for file   1  2.2665E-03+/-  8.1922E-05
```

The `data` command tells the program to read the data as well as the response file that is named in the header of the data file. In general, XSPEC commands can be truncated, provided they remain unambiguous. Since the default extension of a data file is `.pha`, the above command can be abbreviated to `da e1048_me`, if desired. To obtain help on the `data` command, or on any other command, type `help command` followed by the name of the command.

One of the first things most users will want to do at this stage—even before fitting models—is to look at their data. The plotting device should be set first, with the command `cpd` (*change plotting device*). Here, we use the Tektronix 4010 terminal emulator:

```
 XSPEC> cpd /tek4010
```

To see a list of alternative plot devices, type `cpd ?` There are more than 20 different things that can be plotted, all related in some way to the data, the model, the fit and the instrument. To see them, type:

```
 XSPEC> plot ?
  Choose from the following 'plot' sub-commands:
 data      counts    ldata     residuals chisq     delchi    ratio
 summary   contour   efficien  emodel    insensitv model     sensitvty
 ufspec    eufspec
 Insert selection: (data)
```

The most fundamental is the data plotted against instrument channel (`data`); next most fundamental, and more informative, is the data plotted against channel energy. To do this plot, use the XSPEC command `setplot energy`. Figure A shows the result of the commands:

```
 XSPEC> setplot energy
 XSPEC> plot data
```

People familiar with *EXOSAT* ME data will recognize the spectrum to be soft, absorbed and without an obvious bright iron emission line.

We are now ready to fit the data with a model. Models in XSPEC are specified using the `model` command, followed by a list of one or more model components chosen from the suite of more than seventy. There are two basic kinds of model: *additive models* which, after being convolved with the instrument response, prescribe the number of counts per energy bin (e.g., a bremsstrahlung continuum); and *multiplicative models*, which apply an energy-dependent multiplicative factor to additive models before convolution (e.g., an absorption edge). At least one additive component must be included in the model, but there is no restriction on the number of multiplicative models. To see what components are available, type `model ?` (for information about a specific component, type `help model` followed by the name of the component):

```
XSPEC> model ?
 Possible additive models are :
 bbody    bbodyrad bknpower bremss   cemekl   cevmkl   cflow    compLS
 compST   compTT   cp6mkl   cp6vmkl  cutoffpl disk     diskbb   diskline
 diskm    disko    gaussian meka     mekal    mkcflow  nteea    pegpwrlw
 pexrav   pexriv   pliref   plrefl   powerlaw raymond  redge    step
 tita_a   useradd1 vbremss  vmeka    vmekal   vmcflow  vraymond zbbody
 zbremss  zgauss   zplrefl  zpowerlw atable
 Possible multiplicative models are :
 absori   constant cabs     cyclabs  dust     edge     expabs   expfac
 highecut hrefl    notch    pcfabs   phabs    plabs    smedge   spline
 SSS_ice  usermul1 uvred    varabs   wabs     wndabs   zedge    zhighect
 zpcfabs  zphabs   zvarabs  zvfeabs  zwabs    zwndabs  mtable   etable
 Possible mixing models are :
 ascac
```

Given the quality of our data, as shown by the plot, we'll choose an absorbed power law, specified as follows :

```
XSPEC> model wabs powerlaw
```

Or, abbreviating unambiguously:

```
XSPEC> mo wa po
```

Either way, the user is prompted straight away for the initial values of the parameters. Hitting return in response to a prompt enters the default values. As well as the parameter values themselves, users also may specify step sizes and ranges (`value`, `delta`, `min`, `bot`, `top`, and `max values`), but here, we'll enter the defaults:

```
 mo = wabs[1] (powerlaw[2])
Input parameter value, delta, min, bot, top, and max values for ...
```

```
Mod parameter  1 of component  1  1 wabs     nH 10^22
   1.000      1.0000E-03     0.          0.      1.0000E+05  1.0000E+06

Mod parameter  2 of component  2  1 powerlaw PhoIndex
   1.000      1.0000E-02  -10.00      -9.000       9.000       10.00

Mod parameter  3 of component  2  0 powerlaw norm
   1.000      1.0000E-03     0.          0.      1.0000E+05  1.0000E+06

  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (powerlaw[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22   1.00000     +/-      0.
    2    2    2       powerlaw    PhoIndex   1.00000     +/-      0.
    3    3    2       powerlaw    norm       1.00000     +/-      0.
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
    3 variable fit parameters
 Chi-Squared =     4.9092E+08 using   125 PHA bins.
 Reduced chi-squared =     4.0239E+06
```

The `Current statistic` is χ^2 and is huge for the initial, default values—mostly because the power law normalization is two orders of magnitude too large. This is easily fixed using the `renorm` command :

```
XSPEC> ren
 Chi-Squared =      840.5     using   125 PHA bins.
 Reduced chi-squared =      6.890
```

We are not quite ready to fit the data (and obtain a better χ^2), because not all of the 125 PHA bins should be included in the fitting: some are below the lower discriminator of the instrument and therefore do not contain valid data; some have negative count rates because of imperfect background subtraction at the margins of the pass band; and some may not contain enough counts for χ^2 to be strictly meaningful. To find out which channels to discard (*ignore* in `XSPEC` terminology), users must consult mission-specific documentation, which will inform them of discriminator settings, background subtraction problems and other issues. For the mature missions in the HEASARC archives, this information already has been encoded in the headers of the spectral files as a list of "bad" channels. Simply issue the command:

```
XSPEC> ignore bad
 Chi-Squared =      788.1     using    85 PHA bins.
 Reduced chi-squared =      9.611
```

We can see that 40 channels are bad—but do we need to ignore any more? These channels are bad because of certain instrument properties: other channels still may need to be ignored because of the shape and brightness of the spectrum itself. Figure B shows the result of plotting the data in channels (using the commands `setplot channel` and `plot data`). We see that the channels above channel 33 start to contain negative counts. We also see, comparing Figure B with Figure A, which bad channels were ignored. Although visual inspection is not the most rigorous method for deciding which channels to ignore (more on this subject later), it's good enough for now, and will at least prevent us from getting grossly misleading results from the fitting. To ignore channels above 33:

```
XSPEC> ignore 34-**
 Chi-Squared =      666.0     using    31 PHA bins.
 Reduced chi-squared =      23.79
```

The same result can be achieved with the command `ignore 34-125`. The inverse of `ignore` is `notice`, which has the same syntax.

We are now ready to fit the data. Fitting is initiated by the command `fit`. As the fit proceeds, the screen displays the status of the fit for each iteration until either the fit converges to the minimum χ^2, or the user is asked whether the fit is to go through another set of iterations to find the minimum. The default number of iterations is ten, but in our example, convergence is attained after eight:

```
XSPEC> fit
 Chi-Squared  Lvl  Fit param # 1     2          3
   288.68     -4         0.       2.188      8.1403E-03
   48.264     -5         0.       1.914      8.5292E-03
   33.220     -6         0.       1.995      8.8701E-03
   33.214     -7         0.       1.996      8.8726E-03
   32.930      0     3.7669E-02   1.996      8.8858E-03
   32.591     -1     6.9828E-02   2.001      9.0296E-03
   31.325     -2     0.2030       2.062      9.8992E-03
   30.753     -3     0.4391       2.172      1.1650E-02
   30.022     -4     0.5011       2.198      1.2305E-02
Number of trials exceeded - last iteration delta =   0.7311
 Continue fitting? (Y)
   30.014     -5     0.5037       2.199      1.2338E-02
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (powerlaw[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1     1    1       wabs        nH 10^22  0.503747     +/- 0.29087
    2     2    2       powerlaw    PhoIndex   2.19871     +/- 0.12856
    3     3    2       powerlaw    norm      1.233789E-02 +/- 0.24629E-02
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
```

```
 Chi-Squared =      30.01     using    31 PHA bins.
 Reduced chi-squared =      1.072
```

The fit is good: reduced χ^2 is 1.072 for $31 - 3 = 29$ degrees of freedom. To see the fit and the residuals, we can use the command `plot data resid`. The result is shown in Figure C.

The screen output shows the best-fitting parameter values, as well as approximations to their errors. These errors should be regarded as indications of the uncertainties in the parameters and should *not* be quoted in publications. The true errors, i.e. the confidence ranges, are obtained using the `error` command:

```
XSPEC> error 1 2 3
 Fit Index   Confidence Range (     2.706)
     1    3.934827E-02     1.01198
     2     1.99501         2.41692
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
 *WARNING*:RENORM: No variable model to allow renormalization
     3    8.963043E-03    1.734988E-02
```

Here, the numbers 1, 2, 3 refer to the parameter numbers in the `Fit par` column of the screen output. For the first parameter, the column of absorbing hydrogen atoms, the 90% confidence range is $3.9\times10^{20} < N_H < 1.0\times10^{22}$ cm^{-2}. This corresponds to an excursion in χ^2 of 2.706. The reason these "better" errors are not given automatically as part of the `fit` output is that they entail further fitting. When the model is simple, this does not require much CPU, but for complicated models the extra time can be considerable. The warning message is generated because there are no free normalizations in the model while the error is being calculated on the normalization itself. In this case, the warning may safely be ignored.

What else can we do with the fit? One thing is to derive the flux of the model. The data by themselves only give the instrument-dependent count rate. The model, on the other hand, is an estimate of the true spectrum emitted. In XSPEC, the model is defined in physical units independent of the instrument. The command `flux` integrates the current model over the range specified by the user:

```
XSPEC> flux 2 10
 Model flux 3.5488E-03 photons (2.2489E-11 ergs)cm**-2 s**-1 (  2.000- 10.000)
```

Here, we have chosen the standard X-ray range of 2–10 keV and find that the energy flux is 2.2×10^{-11} erg cm^{-2} s^{-1}. Note that `flux` will integrate only within the energy range of the current response matrix. If the model flux outside this range is desired—in effect, an extrapolation beyond the data—then the command `dummyrsp` should be used. This command reads in a dummy response that covers the range required. For example, if we want to know the flux of our model in the *ROSAT* PSPC band of 0.2–2 keV, we enter:

```
XSPEC> dummyrsp 0.2 2
 Chi-Squared =      3586.     using    31 PHA bins.
 Reduced chi-squared =      128.1
XSPEC> flux 0.2 2
 Model flux 4.4267E-03 photons (8.9990E-12 ergs)cm**-2 s**-1 (  0.200-  2.000)
```

The energy flux, at 9.0×10^{-12} erg cm^{-2} s^{-1}, is lower in this band but the photon flux is higher. To get our original response matrix back we enter :

```
XSPEC> response
 Chi-Squared =      30.01     using    31 PHA bins.
 Reduced chi-squared =      1.072
```

The fit, as we've remarked, is good, and the parameters are constrained. But unless the purpose of our investigation is merely to measure a photon index, it's a good idea to check whether alternative models can fit the data just as well. We also should derive upper limits to components such as iron emission lines and additional continua, which, although not evident in the data nor required for a good fit, are nevertheless important to constrain. First, let's try an absorbed black body:

```
XSPEC> mo wa bb
 mo = wabs[1] (bbody[2])
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  1 of component  1 wabs     nH 10^22
   1.000      1.0000E-03      0.          0.       1.0000E+05  1.0000E+06

Mod parameter  2 of component  2 bbody    kT(keV)
   3.000      1.0000E-02  1.0000E-04  1.0000E-02   100.0       200.0

Mod parameter  3 of component  2 bbody    norm
   1.000      1.0000E-03      0.          0.       1.0000E+05  1.0000E+06

  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (bbody[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22   1.00000     +/-      0.
```

```
   2    2    2       bbody       kT(keV)    3.00000     +/-      0.
   3    3    2       bbody       norm       1.00000     +/-      0.
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
   3 variable fit parameters
 Chi-Squared =     3.3333E+09 using    31 PHA bins.
 Reduced chi-squared =     1.1905E+08
XSPEC> fit
 Chi-Squared  Lvl  Fit param # 1     2           3
   1103.2     -1         0.       1.925      3.0014E-04
   789.99     -1         0.      0.7774      1.5796E-04
   374.84     -2         0.      0.9160      3.5594E-04
   174.80     -3         0.      0.9008      2.4199E-04
   136.09     -4         0.      0.8922      3.0098E-04
   120.19     -5         0.      0.8914      2.6655E-04
   115.88     -6         0.      0.8912      2.8554E-04
   114.38     -7         0.      0.8911      2.7471E-04
   113.92     -8         0.      0.8911      2.8078E-04
Number of trials exceeded - last iteration delta =   0.4537
 Continue fitting? (Y)
   113.78     -9         0.      0.8911      2.7734E-04
   113.73    -10         0.      0.8911      2.7928E-04
   113.71    -11         0.      0.8911      2.7818E-04
   113.71    -12         0.      0.8911      2.7880E-04
   113.71      0         0.      0.8913      2.7859E-04
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (bbody[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
   1    1    1       wabs        nH 10^22        0.     +/- -1.0000
   2    2    2       bbody       kT(keV)   0.891259     +/- 0.16956E-01
   3    3    2       bbody       norm      2.785949E-04 +/- 0.74475E-05
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
 Chi-Squared =      113.7     using    31 PHA bins.
 Reduced chi-squared =      4.061
```

Note that when more than 10 iterations are required for convergence the user is asked whether or not to continue at the end of each set of 10. Saying *no* at these prompts is a good idea if the fit is not converging quickly. Conversely, to avoid having to keep answering the question, i.e., to increase the number of iterations before the prompting question appears, begin the fit with, say:

```
XSPEC> fit 100
```

This command will put the fit through 100 iterations before pausing.

The black body fit is obviously not a good one. Not only is χ^2 large, but the best-fitting N_H is rather low. Inspection of the residuals confirms this: the pronounced wave-like shape is indicative of a bad choice of overall continuum (see Figure D). Let's try thermal bremsstrahlung next:

```
XSPEC> mo wa br
 mo = wabs[1] (bremss[2])
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  1 of component  1 wabs     nH 10^22
   1.000      1.0000E-03     0.          0.      1.0000E+05  1.0000E+06

Mod parameter  2 of component  2 bremss   kT(keV)
   7.000      5.0000E-02  0.1000       1.000       100.0       200.0

Mod parameter  3 of component  2 bremss   norm
   1.000      1.0000E-03     0.          0.      1.0000E+05  1.0000E+06

  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (bremss[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22   1.00000     +/-      0.
    2    2    2       bremss      kT(keV)    7.00000     +/-      0.
    3    3    2       bremss      norm       1.00000     +/-      0.
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
    3 variable fit parameters
 Chi-Squared =     3.2534E+08 using    31 PHA bins.
 Reduced chi-squared =     1.1619E+07
XSPEC> fit
 Chi-Squared  Lvl  Fit param # 1     2           3
   42.435     -4         0.       5.087      3.5213E-03
   27.906     -5         0.       5.373      3.5658E-03
   27.906     -6         0.       5.370      3.5687E-03
   27.906      1         0.       5.370      3.5687E-03
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (bremss[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22        0.     +/- -1.0000
    2    2    2       bremss      kT(keV)    5.36973     +/- 0.41151
    3    3    2       bremss      norm     3.568679E-03 +/- 0.26339E-03
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
 Chi-Squared =      27.91     using    31 PHA bins.
```

```
 Reduced chi-squared =     0.9966
```

Bremsstrahlung is a better fit than the black body—and is as good as the power law—although it shares the low N_H. With two good fits, the power law and the bremsstrahlung, it's time to scrutinize their parameters in more detail.

From the *EXOSAT* database on HEASARC, we know that the target in question, 1E1048.1–5937, has a Galactic latitude of 24 arcmin, i.e., almost on the plane of the Galaxy. In fact, the data base also provides the value of the Galactic N_H based on 21-cm radio observations. At 4×10^{22} cm^{-2}, it is higher than the 90 percent-confidence upper limit from the power-law fit. Perhaps, then, the power-law fit is not so good after all. What we can do is fix (*freeze* in XSPEC terminology) the value of N_H at the Galactic value and refit the power law. Although we won't get a good fit, the shape of the residuals might give us a clue to what is missing. To freeze a parameter in XSPEC, use the command `freeze` followed by the parameter number, like this:

```
XSPEC> freeze 1
Number of variable fit parameters =    2
```

The inverse of `freeze` is `thaw`:

```
XSPEC> thaw 1
Number of variable fit parameters =    3
```

Alternatively, parameters can be frozen using the `newpar` command, which allows all the quantities associated with a parameter to be changed. The second quantity, `delta`, is the step size used in the fitting, and, if set to a negative number, will cause the parameter to be frozen. In our case, we want N_H frozen at 4×10^{22} cm^{-2}, so we go back to the power law best fit and do the following :

```
XSPEC> newpar 1
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  1 of component  1 wabs     nH 10^22
  0.5038      1.0000E-03     0.          0.      1.0000E+05  1.0000E+06
4,0
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (powerlaw[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22   4.00000     frozen
    2    2    2       powerlaw    PhoIndex   2.19872     +/- 0.12856
    3    3    2       powerlaw    norm      1.233807E-02 +/- 0.24628E-02
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
    2 variable fit parameters
```

```
 Chi-Squared =      776.6     using    31 PHA bins.
 Reduced chi-squared =      26.78
```

Note the useful trick of giving a value of zero for `delta` in the `newpar` command. This has the effect of changing zero to the negative of its current value. If the parameter is free, it will be frozen, and if frozen, thawed. The same result can be obtained by putting everything onto the command line, i.e., `newpar 1 4, 0`, or by issuing the two commands, `newpar 1 4` followed by `freeze 1`. Now, if we fit again, we get the following set of parameters:

```
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (powerlaw[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22   4.00000     frozen
    2    2    2       powerlaw    PhoIndex   3.51070     +/- 0.66531E-01
    3    3    2       powerlaw    norm      0.102057     +/- 0.81193E-02
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
 Chi-Squared =      113.6     using    31 PHA bins.
 Reduced chi-squared =      3.916
```

The fit is not good. In Figure E we can see why: there appears to be a surplus of softer photons, perhaps indicating a second continuum component. To investigate this possibility we can add a component to our model. The `addcomp` command lets us do this without having to respecify the model from scratch. Here, we'll add a black body component. The numerical argument of `addcomp` is the component number, which indicates where in the model the component should go. Thus, `addcomp 3 bb` will add a black body after the power law:

```
XSPEC> addcomp 3 bb
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  4 of component  3 bbody    kT(keV)
   3.000      1.0000E-02  1.0000E-04  1.0000E-02   100.0       200.0
2,0
Mod parameter  5 of component  3 bbody    norm
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06
1.e-5
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (powerlaw[2] + bbody[3])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22   4.00000     frozen
    2    2    2       powerlaw    PhoIndex   3.51070     +/- 0.66531E-01
```

```
  3    3    2       powerlaw    norm      0.102057     +/- 0.81193E-02
  4    4    3       bbody       kT(keV)    2.00000     frozen
  5    5    3       bbody       norm      1.000000E-05 +/-      0.
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
   3 variable fit parameters
 Chi-Squared =      110.6     using    31 PHA bins.
 Reduced chi-squared =      3.948
```

Notice that in specifying the initial values of the black body, we have frozen kT at 2 keV (the canonical temperature for nuclear burning on the surface of a neutron star in a low-mass X-ray binary) and started the normalization at zero. Without these measures, the fit might "lose its way". Notice, too, the line in the screen output that indicates where the command (`addcomp 3`) placed the blackbody, `mo = wabs[1] (powerlaw[2] + bbody[3])`. Now, if we fit, we get (not showing all the iterations this time):

```
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 mo = wabs[1] (powerlaw[2] + bbody[3])
 Model Fit Model   Component     Parameter   Value
 par   par comp
   1    1    1       wabs        nH 10^22   4.00000     frozen
   2    2    2       powerlaw    PhoIndex   4.76352     +/- 0.15957
   3    3    2       powerlaw    norm      0.303830     +/- 0.43608E-01
   4    4    3       bbody       kT(keV)    2.00000     frozen
   5    5    3       bbody       norm      2.275169E-04 +/- 0.34069E-04
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 Chi-Squared =      51.84     using    31 PHA bins.
 Reduced chi-squared =      1.851
```

The fit is better than the one with just a power law *and* the fixed Galactic column, but it is still not good. Thawing the black body temperature and fitting gives us:

```
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 mo = wabs[1] (powerlaw[2] + bbody[3])
 Model Fit Model   Component     Parameter   Value
 par   par comp
   1    1    1       wabs        nH 10^22   4.00000     frozen
   2    2    2       powerlaw    PhoIndex   6.12218     +/- 0.38323
   3    3    2       powerlaw    norm      0.801310     +/- 0.22076
   4    4    3       bbody       kT(keV)    1.21930     +/- 0.87138E-01
   5    5    3       bbody       norm      2.592895E-04 +/- 0.28472E-04
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
```

```
 Chi-Squared =      35.61     using    31 PHA bins.
 Reduced chi-squared =      1.319
```

This, of course, is a better fit, but the photon index of the power law has ended up extremely and implausibly steep. Looking at this odd model (with the command `plot model`), we see, in Figure F, that the black body and the power law have changed places, in that the power law provides the soft photons required by the high absorption, while the black body provides the harder photons.

We could continue to search for a plausible, well-fitting model, but the data, with their limited signal-to-noise and energy resolution, probably don't warrant it (the original investigators published only the power law fit). There is, however, one final, useful thing to do with the data: derive an upper limit to the presence of a fluorescent iron emission line. First we delete the black body component using the inverse of `addcomp`, namely, `delcomp`:

```
XSPEC> delcomp 3
  mo = wabs[1] (powerlaw[2])
 Chi-Squared =      1236.     using    31 PHA bins.
 Reduced chi-squared =      42.62
```

Then we thaw N_H and refit to recover our original, best fit (output not shown). Then, we add a gaussian emission line of fixed energy and width:

```
XSPEC> addcomp 3 gaussian
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  4 of component  3 gaussian LineE
   6.500      5.0000E-02      0.          0.       100.0       100.0
6.4 0
Mod parameter  5 of component  3 gaussian Sigma
  0.1000      5.0000E-02      0.          0.       10.00       20.00
,0
Mod parameter  6 of component  3 gaussian norm
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06
1.e-4
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (powerlaw[2] + gaussian[3])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22  0.504116     +/- 0.29075
    2    2    2       powerlaw    PhoIndex   2.19884     +/- 0.12845
    3    3    2       powerlaw    norm      1.234063E-02 +/- 0.24683E-02
    4    4    3       gaussian    LineE      6.40000     frozen
    5    5    3       gaussian    Sigma     0.100000     frozen
    6    6    3       gaussian    norm      1.000000E-04 +/-      0.
```

```
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
   4 variable fit parameters
 Chi-Squared =      32.64     using    31 PHA bins.
 Reduced chi-squared =      1.209
```

The energy and width have to be frozen because,in the absence of an obvious line in the data, the fit would be completely unable to converge on meaningful values. Besides, our aim is to see how bright a line at 6.4 keV can be and still not ruin the fit. To do this, we fit first and then use the `error` command to derive the maximum allowable iron line normalization. We then set the normalization at this maximum value with `newpar` and, finally, derive the equivalent width using the `eqwidth` command. That is:

```
XSPEC> err 6
 Fit Index   Confidence Range (     2.706)
 Parameter pegged at hard limit         0.
 with delta ftstat=     1.800
     6          0.        1.421908E-04
XSPEC> new 6 1.421908E-04
    4 variable fit parameters
 Chi-Squared =      33.25     using    31 PHA bins.
 Reduced chi-squared =      1.231
XSPEC> eqwidth 3
 Additive group equiv width for model 3 (gaussian): 780. eV
```

Things to note:

- The true minimum value of the gaussian normalization is less than zero, but the `error` command stopped searching for a $\Delta\chi^2$ of 2.706 when the minimum value hit zero, the "hard" lower limit of the parameter. Hard limits can be adjusted with the `newpar` command, and they correspond to the quantities `min` and `max` associated with the parameter values. In fact, according to the screen output, the value of $\Delta\chi^2$ corresponding to zero normalization is 1.805.
- Like `addcomp` and `delcomp`, the command `eqwidth` takes the component number as its argument.
- The upper limit on the equivalent width of a 6.4 keV emission line is high (780 eV)!

4.4 Fitting Data Files Simultaneously: Examples from Einstein and Ginga

XSPEC has the very useful facility of allowing models to be fitted simultaneously to more than one data file. It is even possible to group files together and to fit different models simultaneously. Reasons for fitting in this manner include:

- *The same target is observed at several epochs but, although the source stays constant, the response matrix has changed.* When this happens, the data files cannot be added together; they have to be fitted separately. Fitting the data files simultaneously yields tighter constraints.
- *The same target is observed with different instruments.* The GIS and SIS on *ASCA*, for example, observe in the same direction simultaneously. As far as XSPEC is concerned, this is just like the previous case: two data files with two responses fitted simultaneously with the same model.
- *Different targets are observed, but the user wants to fit the same model to each data file with some parameters shared and some allowed to vary separately.* For example, if you have a series of spectra from a variable AGN, you might want to fit them simultaneously with a model that has the same, common photon index but separately vary the normalization and absorption.

Other scenarios are possible—the important thing is to recognize the flexibility of XSPEC in this regard.

As an example of the first case, we'll fit two spectra derived from two separate *Einstein* Solid State Spectrometer (SSS) observations of the cooling-flow cluster Abell 496. Although the two observations were carried out on consecutive days (in August 1979), the response is different, due to the variable build-up of ice on the detector. This problem bedeviled analysis during the mission; however, it has now been calibrated successfully and is incorporated into the response matrices associated with the spectral files in the HEASARC archive. The SSS also provides an example of how background correction files are used in XSPEC.

To fit the same model with the same set of parameters to more than one data file, simply enter the names of the data files after the `data` command:

```
XSPEC> data a496_sss_b a496_sss_c
Net count rate (cts/cm^2/s) for file   1  4.4313E-03+/-  7.6922E-05( 86.1% total)
Net count rate (cts/cm^2/s) for file   2  4.5637E-03+/-  7.0531E-05( 86.0% total)
Net correction flux for file  1=    9.1230E-04
Net correction flux for file  2=    9.4587E-04
```

Although it's transparent to the user, XSPEC also has read in the associated:

- response files (`a496_sss_b.rsp` & `a496_sss_c.rsp`),
- background files (`a496_sss_b.bck` & `a496_sss_c.bck`) and
- correction files (`a496_sss_b.cor` & `a496_sss_c.cor`).

These files are all listed in the header of the corresponding data files (`a496_sss_b.pha` & `a496_sss_c.pha`).

To ignore channels, the file number (1 & 2 in this example) precedes the range of channels to be ignored. Here, we wish to ignore, for both files, the first channel and channels 86–94. This can be done by specifying the files one after the other with the range of ignored channels:

```
XSPEC> ignore 1:1 1:86-94 2:1 2:86-94
```

or by specifying the range of file number with the channel range:

```
XSPEC> ignore 1-2:1 86-94
```

In this example, we'll fit a cooling-flow model under the reasonable assumption that the small SSS field of view sees mostly just the cool gas in the middle of the cluster. We'll freeze the values of the maximum temperature (the temperature from which the gas cools) and of the abundance to the values found by instruments such as the *Ginga* LAC and the *EXOSAT* ME, which observed the entire cluster. The minimum gas temperature is frozen at 0.1 keV; the "slope" is frozen at zero (isobaric cooling) and the normalization is given an initial value of 100 solar masses per year:

```
XSPEC> mo wa cflow
 mo = wabs[1] (cflow[2])
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  1 of component  1 wabs     nH 10^22
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06
0.045
Mod parameter  2 of component  2 cflow    slope
      0.      1.0000E-02  -5.000      -5.000       5.000       5.000
,0
Mod parameter  3 of component  2 cflow    lowT
  0.1000      1.0000E-03  8.0800E-02  8.0800E-02   79.90       79.90
,0
Mod parameter  4 of component  2 cflow    highT
   4.000      1.0000E-03  8.0800E-02  8.0800E-02   79.90       79.90
,0
Mod parameter  5 of component  2 cflow    Abundanc
   1.000      1.0000E-02      0.          0.       5.000       5.000
0.5,0
Mod parameter  6 of component  2 cflow    Redshift
      0.     -0.1000          0.          0.       100.0       100.0
0.032
Mod parameter  7 of component  2 cflow    norm
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06
100
```

```
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 mo = wabs[1] (cflow[2])
 Model Fit Model   Component     Parameter   Value
 par   par comp
   1    1    1       wabs        nH 10^22  4.500000E-02 +/-      0.
   2    2    2       cflow       slope           0.     frozen
   3    3    2       cflow       lowT      0.100000     frozen
   4    4    2       cflow       highT      4.00000     frozen
   5    5    2       cflow       Abundanc  0.500000     frozen
   6    6    2       cflow       Redshift  3.200000E-02 frozen
   7    7    2       cflow       norm       100.000     +/-      0.
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
   2 variable fit parameters
 Chi-Squared =      2728.     using   168 PHA bins.
 Reduced chi-squared =      16.43
XSPEC> fit
 Chi-Squared  Lvl  Fit param # 1     2           3           4
                5           6           7
   427.62     -3     0.2120          0.      0.1000       4.000
              0.5000      3.2000E-02   285.3
   378.97     -4     0.2663          0.      0.1000       4.000
              0.5000      3.2000E-02   321.1
   378.12     -5     0.2744          0.      0.1000       4.000
              0.5000      3.2000E-02   325.8
   378.11     -6     0.2755          0.      0.1000       4.000
              0.5000      3.2000E-02   326.3
   378.11     -7     0.2757          0.      0.1000       4.000
              0.5000      3.2000E-02   326.4
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 mo = wabs[1] (cflow[2])
 Model Fit Model   Component     Parameter   Value
 par   par comp
   1    1    1       wabs        nH 10^22  0.275718     +/- 0.99672E-02
   2    2    2       cflow       slope           0.     frozen
   3    3    2       cflow       lowT      0.100000     frozen
   4    4    2       cflow       highT      4.00000     frozen
   5    5    2       cflow       Abundanc  0.500000     frozen
   6    6    2       cflow       Redshift  3.200000E-02 frozen
   7    7    2       cflow       norm       326.450     +/-  6.0192
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 Chi-Squared =      378.1     using   168 PHA bins.
 Reduced chi-squared =      2.278
```

As we can see, χ^2 is not good, but the high statistic could be because we have yet to adjust the correction file. Correction files in XSPEC take into account detector features that cannot be completely prescribed *ab initio* and which must be fitted at the same time as the model. *Einstein* SSS spectra, for example, have a background feature the level of which varies unpredictably. Its spectral form is contained in the correction file, but its normalization is determined by fitting. This fitting is set in motion using the command `recornrm` (*reset the correction-file normalization*):

```
XSPEC> reco 1
  File #     Correction
      1      0.4153 +/-  0.0672
 After correction norm adjustment  0.415 +/- 0.067
 Chi-Squared =      340.0     using   168 PHA bins.
 Reduced chi-squared =      2.048
XSPEC> reco 2
  File #     Correction
      2      0.4890 +/-  0.0597
 After correction norm adjustment  0.489 +/- 0.060
 Chi-Squared =      272.9     using   168 PHA bins.
 Reduced chi-squared =      1.644
```

This process is iterative, and, in order to work, must be used in tandem with fitting the model. Successive fits and recorrections are applied until the fit is stable, i.e., until further improvement in χ^2 no longer results. Of course, this procedure is only worthwhile when the model gives a reasonably good account of the data.

```
XSPEC> fit
 .
 .
 Chi-Squared =      228.3     using   168 PHA bins.
 Reduced chi-squared =      1.375
XSPEC> reco 1
  File #     Correction
      1      0.6139 +/-  0.0672
 After correction norm adjustment  1.478 +/- 0.162
 Chi-Squared =      219.6     using   168 PHA bins.
 Reduced chi-squared =      1.323
XSPEC> reco 2
  File #     Correction
      2      0.6805 +/-  0.0597
 After correction norm adjustment  1.392 +/- 0.122
 Chi-Squared =      209.3     using   168 PHA bins.
 Reduced chi-squared =      1.261

XSPEC> fit
 .
 .
```

```
 Chi-Squared =      201.3     using   168 PHA bins.
 Reduced chi-squared =      1.213
XSPEC> reco 1
  File #     Correction
      1      0.6976 +/-  0.0672
 After correction norm adjustment  1.136 +/- 0.110
 Chi-Squared =      199.7     using   168 PHA bins.
 Reduced chi-squared =      1.203
XSPEC> reco 2
  File #     Correction
      2      0.7613 +/-  0.0597
 After correction norm adjustment  1.119 +/- 0.088
 Chi-Squared =      197.9     using   168 PHA bins.
 Reduced chi-squared =      1.192
XSPEC> fit
.
.
 Chi-Squared =      196.5     using   168 PHA bins.
 Reduced chi-squared =      1.184
XSPEC> reco 1
  File #     Correction
      1      0.7327 +/-  0.0672
 After correction norm adjustment  1.050 +/- 0.096
 Chi-Squared =      196.2     using   168 PHA bins.
 Reduced chi-squared =      1.182
XSPEC> reco 2
  File #     Correction
      2      0.7953 +/-  0.0597
 After correction norm adjustment  1.045 +/- 0.078
 Chi-Squared =      195.9     using   168 PHA bins.
 Reduced chi-squared =      1.180

XSPEC> fit
 Chi-Squared  Lvl  Fit param # 1     2           3           4
               5           6           7
   195.64     -3     0.3268          0.      0.1000       4.000
             0.5000      3.2000E-02   306.9
   195.64     -4     0.3271          0.      0.1000       4.000
             0.5000      3.2000E-02   307.0
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (cflow[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22  0.327075     +/- 0.12541E-01
    2    2    2       cflow       slope          0.     frozen
    3    3    2       cflow       lowT      0.100000     frozen
    4    4    2       cflow       highT      4.00000     frozen
```

```
   5    5    2       cflow       Abundanc  0.500000     frozen
   6    6    2       cflow       Redshift  3.200000E-02 frozen
   7    7    2       cflow       norm       307.017     +/-  6.4184
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 Chi-Squared =      195.6     using   168 PHA bins.
 Reduced chi-squared =      1.179
```

The final value of χ^2 is much better than the original, but is not quite acceptable. However, the current model has only two free parameters: further explorations of parameter space would undoubtedly improve the fit.

We'll leave this example and move on to look at another kind of simultaneous fitting: one where the same model is fitted to two different data files. This time, not all the parameters will be identical. The massive X-ray binary Centaurus X-3 was observed with the LAC on *Ginga* in 1989. Its flux level before eclipse was much lower than the level after eclipse. Here, we'll use XSPEC to see whether spectra from these two phases can be fitted with the same model, which differs only in the amount of absorption. This kind of fitting relies on introducing an extra dimension, the *group*, to the indexing of the data files. The files in each group share the same model but not necessarily the same parameter values, which may be shared as common to all the groups or varied separately from group to group. Although each group may contain more than one file, there is only one file in each of the two groups in this example. Groups are specified with the `data` command, with the group number preceding the file number, like this:

```
XSPEC> da 1:1 losum 2:2 hisum
Net count rate (cts/cm^2/s) for file   1  3.5022E-02+/-  8.8726E-05
Net count rate (cts/cm^2/s) for file   2  0.3428    +/-  7.8069E-04
```

Here, the first group makes up the file `losum.pha`, which contains the spectrum of all the low, pre-eclipse emission. The second group makes up the second file, `hisum.pha`, which contains all the high, post-eclipse emission. Note that file number is "absolute" in the sense that it is independent of group number. Thus, if there were three files in each of the two groups (`lo1.pha`, `lo2.pha`, `lo3.pha`, `hi1.pha`, `hi2.pha` & `hi3.pha`, say), rather than one, the six files would be specified as follows:

```
XSPEC> da 1:1 lo1 1:2 lo2 1:3 lo3 2:4 hi1 2:5 hi2 2:6 hi3
```

The `ignore` command works, as usual, on file number, and does not take group number into account. So, to ignore channels 1–3 and 37–48 of both files:

```
XSPEC> ignore 1-2:1-3 37-48
```

The model we'll use at first to fit the two files is an absorbed power law with a high-energy cut-off:

```
XSPEC> mo wa highecut po
```

After defining the model, the user is prompted for two sets of parameter values, one for the first group of data files (`losum.pha`), the other for the second group (`hisum.pha`). Here, we'll enter the absorption column of the first group as 10^{24} cm^{-2} and enter the default values for all the other parameters in the first group. Now, when it comes to the second group of parameters, we enter a column of 10^{22} cm^{-2} and then hit returns to enter defaults for the other parameters. The rule being applied here is as follows: to tie parameters in the second group to their equivalents in the first group, hit return when entering the second-group parameters; to allow parameters in the second group to vary independently of their equivalents in the first group, enter different values explicitly. Like this:

```
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  1 of component  1 wabs     nH 1  for data group 1
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06
100
Mod parameter  2 of component  2 highecut cuto  for data group 1
   10.00      1.0000E-02  1.0000E-04  1.0000E-02   100.0       200.0

Mod parameter  3 of component  2 highecut fold  for data group 1
   15.00      1.0000E-02  1.0000E-04  1.0000E-02   100.0       200.0

Mod parameter  4 of component  3 powerlaw PhoI  for data group 1
   1.000      1.0000E-02  -3.000      -2.000       9.000       10.00

Mod parameter  5 of component  3 powerlaw norm  for data group 1
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06

Mod parameter  6 of component  4 wabs     nH 1  for data group 2
   100.0      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06
1
Mod parameter  7 of component  5 highecut cuto  for data group 2
   10.00      1.0000E-02  1.0000E-04  1.0000E-02   100.0       200.0

Mod parameter  8 of component  5 highecut fold  for data group 2
   15.00      1.0000E-02  1.0000E-04  1.0000E-02   100.0       200.0

Mod parameter  9 of component  6 powerlaw PhoI  for data group 2
   1.000      1.0000E-02  -3.000      -2.000       9.000       10.00

Mod parameter 10 of component  6 powerlaw norm  for data group 2
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06

  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] highecut[2] (powerlaw[3])
```

```
 Model Fit Model   Component     Parameter   Value                        Data
 par   par comp                                                           group
   1    1    1       wabs        nH 10^22   100.000     +/-       0.          1
   2    2    2       highecut    cutoffE    10.0000     +/-       0.          1
   3    3    2       highecut    foldE      15.0000     +/-       0.          1
   4    4    3       powerlaw    PhoIndex   1.00000     +/-       0.          1
   5    5    3       powerlaw    norm       1.00000     +/-       0.          1
   6    6    4       wabs        nH 10^22   1.00000     +/-       0.          2
   7    2    5       highecut    cutoffE    10.0000     = par    2            2
   8    3    5       highecut    foldE      15.0000     = par    3            2
   9    4    6       powerlaw    PhoIndex   1.00000     = par    4            2
  10    5    6       powerlaw    norm       1.00000     = par    5            2
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
   6 variable fit parameters
Chi-Squared =     2.5086E+07 using    66 PHA bins.
Reduced chi-squared =     4.1809E+05
```

Notice how the summary of the model, displayed immediately above, is different now that we have two groups, as opposed to one (as in all the previous examples). We can see that of the 10 model parameters, 6 are free (i.e., 4 of the second group parameters are tied to their equivalents in the first group). Fitting this model results in a huge χ^2 (not shown here), because our assumption that only a change in absorption can account for the spectral variation before and after eclipse is clearly wrong. Perhaps scattering also plays a role in reducing the flux before eclipse. This could be modeled (simply at first) by allowing the normalization of the power law to be smaller before eclipse than after eclipse. To decouple tied parameters, we change the parameter value in the second group to a value—any value—different from that in the first group (changing the value in the first group has the effect of changing both without decoupling). As usual, the `newpar` command is used:

```
XSPEC> newpar 10 1
    7 variable fit parameters
 Chi-Squared =     2.5086E+07 using    66 PHA bins.
 Reduced chi-squared =     4.2518E+05
```

After fitting (not shown), this decoupling reduces χ^2 by a factor of six to 15,599, but this is still too high. Indeed, this simple attempt to account for the spectral variability in terms of "blanket" cold absorption and scattering does not work. More sophisticated models, involving additional components and partial absorption, should be investigated.

4.5 Using XSPEC to Simulate Data: an Example from ASCA

In several cases, analyzing simulated data is a powerful tool to demonstrate feasibility. For example:

- *To support an observing proposal.* That is, to demonstrate what constraints a proposed observation would yield.
- *To support a hardware proposal.* If a response matrix is generated, it can be used to demonstrate what kind of science could be done with a new instrument.
- *To support a theoretical paper.* A theorist could write a paper describing a model, and then show how these model spectra would appear when observed. This, of course, is very like the first case.

Here, we'll use XSPEC to see how an *ASCA* observation of the elliptical galaxy NGC 4472 can constrain the condition of the hot gas. The first step is to define a model on which to base the simulation. The way XSPEC creates simulated data is to take the current model, convolve it with the current response matrix, while adding noise appropriate to the integration time specified. Once created, the simulated data can be analyzed in the same way as real data to derive confidence limits.

We begin by looking in the literature for the best estimate of the NGC 4472 spectrum. *BBXRT* observed the galaxy in 1990 and the results were published in Serlemitsos et al., (1993). They found a flux in the 0.5–4.5 keV range of 6.7×10^{-12} erg cm^{-2} s^{-1}, a temperature range of $0.74 < kT < 0.98$, an abundance range (as a fraction of solar) of $0.09 < A < 0.46$ and a column range of $5.0 \times 10^{20} < N_H < 3.7 \times 10^{21}$ cm^{-2}. A Raymond-Smith spectral model was found to give a good fit. We specify this model at first with the median parameter values, except for the normalization of the Raymond-Smith, which we leave at its default value of unity at first (but adjust later):

```
 mo = wabs[1] (raymond[2])
XSPEC> mo wa ray
Input parameter value, delta, min, bot, top, and max values for ...
Mod parameter  1 of component  1 wabs     nH 10^22
   1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06
0.21
Mod parameter  2 of component  2 raymond  kT(keV)
   1.000      1.0000E-02  8.0000E-03  8.0000E-03   64.00       64.00
0.86
Mod parameter  3 of component  2 raymond  Abundanc
   1.000     -1.0000E-03      0.          0.       5.000       5.000
0.27
Mod parameter  4 of component  2 raymond  Redshift
```

```
     0.     -1.0000E-03      0.          0.       2.000       2.000

 Mod parameter  5 of component  2 raymond  norm
    1.000      1.0000E-03      0.          0.      1.0000E+05  1.0000E+06

   ---------------------------------------------------------------------------
   ---------------------------------------------------------------------------
   mo = wabs[1] (raymond[2])
   Model Fit Model   Component     Parameter   Value
   par   par comp
     1    1    1       wabs        nH 10^22  0.210000     +/-      0.
     2    2    2       raymond     kT(keV)   0.860000     +/-      0.
     3    3    2       raymond     Abundanc  0.270000     frozen
     4    4    2       raymond     Redshift        0.     frozen
     5    5    2       raymond     norm       1.00000     +/-      0.
   ---------------------------------------------------------------------------
   ---------------------------------------------------------------------------
     3 variable fit parameters
```

We now can derive the correct normalization by using the commands `dummyrsp`, `flux` and `newpar`. That is, we'll determine the flux of the model with the normalization of unity (this requires a response matrix to cover the *BBXRT* band—we use a dummy response here). We then work out the new normalization and reset it:

```
 XSPEC> dummy 0.5 4.5
 XSPEC> flux 0.5 4.5
  Model flux 0.2912     photons (5.1221E-10 ergs)cm**-2 s**-1 (   0.500-  4.500)
 XSPEC> newpar 5 0.013
     3 variable fit parameters
 XSPEC> flux
  Model flux 3.7860E-03 photons (6.6587E-12 ergs)cm**-2 s**-1 (   0.500-  4.500)
```

Here, we have changed the value of the normalization (the fifth parameter) from 1 to $6.7 \times 10^{-12}/5.1221 \times 10^{-10} = 0.013$ to give the flux observed by *BBXRT* (6.7×10^{-12} erg cm^{-2} s^{-1} in the energy range 0.5–4.5).

The next stage, creating the simulated data, requires a response matrix. In this example, we'll use the pre-flight *ASCA* SIS matrix, `sis_small.rmf`. The simulation is initiated with the command `fakeit`. If the argument `none` is given, the user will be prompted for the name of the response matrix. If no argument is given, the current response will be used:

```
 XSPEC> fakeit none
 For fake data, file #  1 needs response file: sis_small.rmf
```

There then follows a series of prompts asking the user to specify whether he or she wants counting statistics (yes!), the name of the fake data file (`ngc4472_sis.fak` in our example), and the integration time `T` (40,000 seconds—the other quantities `A, Bkg, cornorm` can be left at their default values).

```
Use counting statistics in creating fake data? (y)
 Input optional fake file prefix (max 12 chars):
Override default values for file #    1
 Fake data filename (sis_small.fak): ngc4472_sis.fak
 T, A, Bkg, cornorm ( 1.0000 , 1.0000 , 1.0000 , 0. ): 40000
 Net count rate (cts/cm^2/s) for file   1  0.4010    +/-  3.1747E-03
 Chi-Squared =      198.2     using   256 PHA bins.
 Reduced chi-squared =     0.7835
```

We now have created a file containing a simulated spectrum of NGC 4472. As is usual before fitting, we need to check which channels to ignore. This time, we'll examine the actual numbers of counts in each channel and reject those that have fewer than 20 per channel. We use `iplot counts` and see that our criterion requires us to ignore channels 1–20 and 101–256:

```
XSPEC> ign 1-20 101-256
 Chi-Squared =      74.79     using    80 PHA bins.
 Reduced chi-squared =     0.9713
```

As expected, χ^2 is excellent even before fitting because the model and the data have the same shape. But the point of this simulation is to determine confidence ranges. First, we thaw the value of the abundance (fixed by default), fit and then use the `error` command:

```
XSPEC> thaw 3
 Number of variable fit parameters =    4
XSPEC> fit
 Chi-Squared  Lvl  Fit param # 1     2           3           4
                5
   69.395     -3     0.2236      0.8557      0.2607          0.
             1.3502E-02
   69.355     -4     0.2238      0.8554      0.2604          0.
             1.3536E-02
   69.355      3     0.2238      0.8554      0.2604          0.
             1.3536E-02
  ---------------------------------------------------------------------------
  ---------------------------------------------------------------------------
  mo = wabs[1] (raymond[2])
  Model Fit Model   Component     Parameter   Value
  par   par comp
    1    1    1       wabs        nH 10^22  0.223845     +/- 0.85826E-02
    2    2    2       raymond     kT(keV)   0.855436     +/- 0.42361E-02
```

```
   3    3    2       raymond     Abundanc  0.260423     +/- 0.12511E-01
   4    4    2       raymond     Redshift       0.      frozen
   5    5    2       raymond     norm      1.353650E-02 +/- 0.45268E-03
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 Chi-Squared =      69.35     using    80 PHA bins.
 Reduced chi-squared =     0.9126
XSPEC> err 1 2 3
 Fit Index   Confidence Range (     2.706)
     1    0.210177        0.238260
     2    0.848311        0.862247
     3    0.240804        0.282047
```

These confidence ranges show that an *ASCA* observation would definitely constrain the parameters, especially the column and abundance, more tightly than the original *BBXRT* observation. Of course, whether these constraints are sufficient depends on the theories being tested. When producing and analyzing simulated data, it is crucial to keep in mind the purpose of the proposed observation, for the potential parameter space that can be covered with simulations is almost limitless.

4.6 Producing Plots: Modifying the Defaults

The final results of using XSPEC are usually one or more tables containing confidence ranges and fit statistics, and one or more plots showing the fits themselves. So far, all the plots shown have the default settings, but it is possible to edit plots to get closer to the appearance what you want.

The plotting package used by XSPEC is **PGPLOT**, which is comprised of a library of low-level tasks. At a higher level is **QDP**, the interactive program that forms the interface between the XSPEC user and **PGPLOT**. **QDP** has its own manual; it also comes with on-line help. Here, we show how to make some of the most common modifications to plots.

To initiate interactive plotting in XSPEC, use the command `iplot` instead of the usual `plot`. In this example, we'll take the simulated *ASCA* SIS spectrum of the previous section and make the following modifications to the `data` plot:

- Change the aspect ratio
- Change the labels
- Rescale the x-axis and y-axis
- Change the y-axis to be a logarithmic scale
- Thicken the lines and make the characters smaller to make the hardcopy look better

- Produce a postscript file

After the `iplot` command, the plot itself appears, followed by the `QDP` prompt:

```
XSPEC> iplot data
PLT>
```

The first thing we'll do is change the aspect ratio of the box that contains the plot (*viewport* in QDP terminology). The viewport is defined by the coordinates of the lower left and upper right corners of the page, normalized so that the width and height of the page are unity. The labels fall outside the viewport, so if the full viewport were specified, only the plot would appear. The default box has a viewport with corners at (0.1, 0.1) and (0.9, 0.9). For our purposes, we want a viewport with corners at (0.2, 0.2) and (0.8, 0.7): with this size and shape, the hardcopy will fit nicely on the page and not have to be reduced for photocopying. To change the viewport, use the command `viewport` followed by the coordinates:

```
PLT> viewport 0.2 0.2 0.8 0.7
```

Next we want to change two of the labels: the label at the top, which currently says only *data*, and the label that specifies the filename. This change is a straightforward one using the `label` command, which takes as arguments a location description and the text string:

```
PLT> label top Simulated Spectrum of NGC 4472
PLT> label file ASCA SIS
```

Other location descriptors are available, including `x` and `y` for the x-axis and y-axis, respectively. To get help on a QDP command, type `help` followed by the name of the command at the `PLT>` prompt. Note that QDP commands can be abbreviated, just like XSPEC commands. To see the results of changing the viewport and the labels, just enter the command `plot`:

```
PLT> plot
```

The two changes we want to make next are to rescale the axes and to change the y-axis to a logarithmic scale. The commands for these changes also are straightforward: the `rescale` command takes the minimum and maximum values as its arguments, while the `log` command takes `x` or `y` as arguments:

```
PLT> rescale x 0.4 2.5
PLT> rescale y 0.01 1
PLT> log y
```

To revert to a linear scale, use the command `log off y`. All that is left to change are the thickness of the lines (the default, least for postscript files that are turned into hardcopies, is too fine) and the size of the characters (we want slightly smaller characters). The `lwidth` command does the former: it takes a width as its argument: the default is 1: we'll reset it to 3. The `csize` command does the latter, taking a normalization as its argument. One (1) will not change the size, a number less than one will reduce it and a number bigger than one will increase it.

```
 PLT> lwidth 3
 PLT> csize 0.8
```

Finally, to produce a postscript file that we can print, we use the `hardcopy` command:

```
PLT> hardcopy ngc4472_sis.ps/ps
```

Here, we have given the file the name **ngc4472_sis.ps**. It will be written into the current directory. The suffix **/ps** tells the program to produce a postscript file.

The result of all this manipulation is shown proudly in Figure G.

4.7 Figure Captions

Figure A: The result of the command `plot data` when the data file in question is the *EXOSAT* ME spectrum of the 6-s X-ray pulsar 1E1048.1–5937 available from the HEASARC on-line service. The axes, labels etc. have not been modified.

Figure B: The result of the command `plot data` after the command `ignore bad` on the *EXOSAT* ME spectrum 1E1048.1–5937.

Figure C: The result of the command `plot data` with: the ME data file from 1E1048.1–5937; "bad" and negative channels ignored; the best-fitting absorbed power-law model; the residuals of the fit

Figure D: As for Figure C, but the model is the best-fitting absorbed black body. Note the wave-like shape of the residuals which indicates how poor the fit is, i.e. that the continuum is obviously *not* a black body.

Figure E: As for Figure C & D, but the model is the best-fitting power law with the absorption fixed at the Galactic value. Under the assumptions that the absorption really is the same as the 21-cm value and that the continuum really is a power law, this plot provides some indication of what other components might be added to the model to improve the fit.

Figure F: The result of the command `plot model` in the case of the ME data file from 1E1048.1–5937. Here, the model is the best-fitting combination of power law, black body and fixed Galactic absorption. The three lines show the two continuum components (absorbed to the same degree) and their sum.

Figure G: A simulated *ASCA* SIS spectrum of NGC 4472 produced to show how a plot can be modified by the user.

Chapter 5

XSPEC commands

5.1 Summary of Commands

`abund`	- Set the Solar abundance table.
`addcomp`	- Add a component to the model.
`alias`	- Create an alias for a command string.
`arf`	- Read an auxiliary response file.
`backgrnd`	- Reset the files to be used for background subtraction.
`buffer`	- Enable/disable/list the command buffer.
`chatter`	- Control the verbosity of XSPEC.
`corfile`	- Reset the files to be used for background correction.
`cornorm`	- Reset the normalization to be used in correcting the background.
`cosmo`	- Set H0 and q0.
`cpd`	- Alias for `setplot device`
`data`	- Input one or more PHA data files
`delcomp`	- Delete a component from the model
`dummyrsp`	- Create a 'dummy' response, covering a given energy range
`dump`	- Write out history packages of the observed spectrum and model.
`eqwidth`	- Calculate a model component's equivalent width.
`error`	- Determine a single parameter confidence region.
`exit`	- Wind up any hardcopy plots and exit from XSPEC.
`fakeit`	- Produce simulated data files for sensitivity studies.
`fit`	- Find the best fit model parameters.

`flux`	-	Calculate the current model's flux over an energy range.
`freeze`	-	Do not allow a model parameter to vary during the fit.
`gain`	-	Perform a simple modification of the response gain.
`goodness`	-	Monte Carlo calculation of goodness-of-fit for C-statistic.
`hardcopy`	-	Make a hardcopy plot and spool to the printer.
`help`	-	Obtain help on XSPEC commands.
`history`	-	Open a history file, in order to save fit results.
`ignore`	-	Ignore a range of PHA channels in future fit operations.
`improve`	-	Try to find a better minimum (requires MINUIT).
`iplot`	-	As plot command but interactive using PLT.
`log`	-	Open the log file to save output.
`lumin`	-	Calculate the current model's luminosity over a given rest frame energy range and redshift.
`method`	-	Set the minimization method.
`model`	-	Define the model to be used when fitting the data.
`newpar`	-	Modify the model parameters.
`notice`	-	Restore a range of PHA channels for future operations.
`pause`	-	Exit via the FORTRAN PAUSE statement.
`plot`	-	Plot various information on the current plot device.
`prompt`	-	Change the command prompt.
`query`	-	Switch on/off prompt to continue fitting.
`quit`	-	An alias for `exit`.
`recall`	-	Recall an old command.
`recornrm`	-	Adjust correction norms to minimize the fit statistic, holding the model fixed.
`reexecute`	-	Reexecute previous commands.
`renorm`	-	Adjust the model norms, and/or allow automatic renorming.
`response`	-	Reset the files used to determine the detector responses.
`save`	-	Save aspects of the current state to a command file.
`setplot`	-	Modify the plot device and other values used by the plot routines.
`script`	-	Open the script file to save all commands input.
`show`	-	Display current file and model information.
`statistic`	-	Change the fit statistic in use.
`steppar`	-	Step through a range of parameter values; perform a fit at each step.
`suggest`	-	Make a suggestion or report a bug to the XSPEC gnomes.
`systematic`	-	Set the model systematic error.

`thaw`	-	Allow a model parameter to vary during the fit.
`thleqw`	-	Calculates expected fluorescent line equivalent width.
`time`	-	Display elapsed time and other statistical information.
`uncertain`	-	Alias for `error`.
`untie`	-	Untie linked parameters.
`weight`	-	Change the weighting function used for chi-squared fits.

Exceptional responses to the command prompt:

an empty line	– Nothing performed, prompt repeated.
/	– Any remaining arguments will have the values given on the last invocation of the command.
<`EOF`>	– same as `quit`. Under VAX/VMS this is <CTRL-Z>, otherwise use `/*`.
`?` (or anything else)	– Write a list of the commands.

5.2 Description of Syntax

The individual commands are treated in alphabetical order in the following section. The novice would be well-served by reading the treatments of the `data`, `model`, `newpar`, and `fit` commands, in that order, then the other commands as needed. The write-up for each command includes a brief description of the purpose, an outline of the correct syntax, a more detailed discussion of the command assumptions and purpose, and a series of examples. Some commands have one or more subcommands that will be similarly described following the command.

In the command description, the syntax uses the following conventions:

- <an argument> – an argument to the command
- <compound argument> =:: <argument a> <argument b> – the definition of a compound argument
- <an argument>... – a repeated string of arguments of the same type
- [<an argument>] – an optional argument
- {<arg a> or <arg b>} – a choice between an argument of type < arg a > or < arg b > must be made.

5.3 abund

Set the Solar abundance table used in the plasma emission and photoelectric absorption models.

```
XSPEC> abund <option>
```

where <option> is either 'angr', from Anders E. & Grevesse N. (1989, Geochimica et Cosmochimica Acta 53, 197), or 'feld', from Feldman U. (1992, Physica Scripta 46, 202). The tables are :

Element	angr	feld
H	1.00e+0	1.00e+0
He	9.77e-2	9.77e-2
C	3.63e-4	3.98e-4
N	1.12e-4	1.00e-4
O	8.51e-4	8.51e-4
Ne	1.23e-4	1.29e-4
Na	2.14e-6	2.14e-6
Mg	3.80e-5	3.80e-5
Al	2.95e-6	2.95e-6
Si	3.55e-5	3.55e-5
S	1.62e-5	1.62e-5
Cl	1.88e-7	1.88e-7
Ar	3.63e-6	4.47e-6
Ca	2.29e-6	2.29e-6
Cr	4.84e-7	4.84e-7
Fe	4.68e-5	3.24e-5
Ni	1.78e-6	1.78e-6
Co	8.60e-8	8.60e-8

5.4 addcomp

Add a component to the current model.

```
XSPEC> addcomp <component# > <component >
```

where <**component#**> is the position in the model specification for the component and <**component**> is its name. The user is prompted for parameter values for the component.

Examples: Suppose that the current model specification is `wa po`.

XSPEC> addcomp 3 ga	! Change the model to wa po ga
XSPEC> addcomp 1 bb	! Change the model to bb wa po ga

5.5 alias

Alias a command string.

```
XSPEC> alias[/<qual>] [<name>] [<command string>]
```

where `<name>` is the alias to be used, `<command string>` is the command to be aliased, and `<qual>` is one of the options below.

5.5.1 /delete

Delete the specified alias.

```
XSPEC> alias/delete <name>
```

5.5.2 /previous

Set the alias to the last command given.

```
XSPEC> alias/previous <name>
```

5.5.3 /show

Show the current aliases.

```
XSPEC> alias/show <name>
```

<name> can contain wild cards (i.e., $*$).

5.5.4 /system

Specify that the alias is system-wide. When used with the `/show` qualifier, it prints out system aliases.

```
XSPEC> alias/system <name> <command string>
```

```
XSPEC> alias/show/system <name>
```

5.5.5 /all

Used with the `/show` qualifier to give both user and system aliases.

```
XSPEC> alias/show/all <name>
```

<name> can contain wild cards (i.e., $*$).

5.6 arf

Read in one or more auxiliary response files (ARF). An ARF gives area versus energy and is used to modify the response matrix for a data set. The file must be in the OGIP standard format.

```
XSPEC> arf [<filespec>...]
```

where <filespec> =:: [<data set num>] <file name>... and where `<file name>` is the name of the auxiliary response file to be used for the response of the associated data set. `<data set num>` is the data set number for the first `<filename>` in the specification, `<data set num>`+1 is the data set number for the next file, and so on. If no `<data set num>` is given in the first `<filespec>` it is assumed to be 1. If no file specifications are entered, then none of the data set responses are modified. An error message is printed if the data set number is greater than the current number of data sets (as determined from the last use of the `data` command). A file name `none` indicates that no auxiliary response is to be used for that data set. No auxiliary response means that any incident spectrum will produce no counts for those particular channels. If a file is not found or cannot be opened for input, then the user is prompted for a replacement auxiliary response file. An <EOF> at this point is equivalent to `none`. See the `data` command for ways to totally remove the data set from consideration.

Examples: It is assumed that there are currently three data sets:

XSPEC> arf a,b,c	! New files for the auxiliary response are given for all three files.
XSPEC> arf 2 none	! No auxiliary response will be used for the second file.
XSPEC> arf,,d.fits	! d.fits becomes the auxiliary response for the second file.

5.7 backgrnd

Modify one or more of the files used in background subtraction.

`XSPEC> backgrnd [<filespec>...]`

where <filespec> =:: [<data set num>] <file name>... and where `<file name>` is the name of the PHA file to be used for background subtraction for the associated data set. `<data set num>` is the data set number for the first `<file name>` in the specification, `<data set num>+1` is the number for the next file, and so on. If no `<data set num>` is given in the first `<filespec>` it is assumed to be one (1). If no file specifications are entered, then none of the data set backgrounds are modified. An error message is printed if `<data set num>` is greater than the current number of data sets (as determined from the last use of the `data` command. A file name `none` indicates that no background subtraction is to be performed for that data set. If a file is not found or cannot be opened for input, then the user is prompted for a replacement background file. An <EOF> at this point is equivalent to using `none` as the background. See the `data` command for ways of removing the data set from consideration. The user also is prompted for a replacement if the background file has a different number of PHA channels than the associated data set. A warning will be printed out if the background detector ID is different than that of the associated data set.

The current ignore status for channels is not affected by the `bkgrnd` command. (See the `ignore` and `notify` command).

Note that background files have the same format as the PHA files used for the `data` command.

Examples: It is assumed that there are currently three data sets:

XSPEC> backgrnd a,b,c	! New files for background subtraction are given for all three data sets.
XSPEC> backgrnd 2 none	! No background subtraction will be done for the second data set.
XSPEC> backgrnd,,d	! /cd.pha becomes the background for the second data set.

5.8 buffer

Enable/disable/list the recall buffer.

`XSPEC> buffer [<on> or <off> or <list>]`

This command runs the `%ed% on`, `%ed% off`, and `%recall%all` commands respectively (see Appendix D). `buffer on` turns on command line editing and recall using the arrow/control keys, `buffer off` turns it off, and `buffer list` gives the command in recall stack.

5.9 chatter

Control the verbosity of XSPEC.

`XSPEC> chatter <chatter level> <log chatter>`

where `<chatter level>` and `<log chatter>` are integer values. The initial value for each argument is 10. Higher values will encourage XSPEC to tell the user more, lower values tell the user less, or make XSPEC "quieter." `<chatter level>` applies to the terminal output, while `<log chatter>` controls the verbosity in the log file. Currently, the maximum chattiness is 25. Values below five should be avoided, as they tend to make XSPEC far too obscure. Some commands may temporarily modify the chattiness, such as the `error` command. A chattiness of 25 will generate a lot of debug output.

Examples:

XSPEC> chatter 10	! Set the terminal chattiness to 10, same as the initial value.
XSPEC> chatter,,0	! Set the chattiness for the log file to very low. This setting essentially disables the log file output.
XSPEC> chatter 5	! Make XSPEC very quiet.
XSPEC> chatter 10 25	! Restore the terminal chattiness to the initial level, while in the log file XSPEC will tell all (particularly when new data files are read in).

5.10 corfile

Reset the files used for background correction.

```
XSPEC> corfile [<filespec>...]
```

where `<filespec>` is the same as for the `backgrnd` command. The correction file can be associated with a data set to further adjust the count rates. It is a PHA file whose count rate is multiplied by the current associated correction norm (see the `cornorm` and `recornrm` command) and then subtracted from the input uncorrected data. The correction norm is not changed by running the corfile command. Default values for the correction file and norm are included in the data PHA file. Unlike the background file, the correction data does NOT contribute to the measurement error.

A file name of `none` is equivalent to no correction file used. If an input file can not be opened or found, an error message is printed and the user prompted for a replacement. As with the `backgrnd` command, the correction file is checked against the associated data set for number of channels, grouping status, and detector ID.

The current ignore status for channels is not affected by the `corfile` command. Note that correction files have the same format (SF type `XSPEC data`) as the PHA files used by the `data` command. (See the Implementation note for the `data` command for default directory and extension information).

Examples: It is assumed that there are currently three data sets:

XSPEC> corfile a,b,c	! New correction files are used for all three data sets.
XSPEC> corfile 2 none	! No correction will be done for the second data set.
XSPEC> corfile ,,d	! The 2nd file now uses `d.pha` as its correction.

5.11 cornorm

Reset the normalization used in correcting the background.

```
XSPEC> cornorm [[<data set range>...]  [<cor norm>]]...
```

where <data set range> =:: <first data set no.>-<last data set no.> is a range of data sets to which the correction is to be applied and `<cor norm>` is the value to be used for the normalization. A decimal point (.) is used to distinguish a correction norm from a single data set data set-range. If no correction norm is given, then the last value input is used (the initial value is one (1)). If no range is given, then the last single range input is modified. (See the `corfile` command.)

Examples: Assume that there are four data sets, all with associated correction files already defined, either by default in their PHA file, or explicitly by using the `corfile` command.

XSPEC> cornorm 1-4 1.	! The correction norm for all four is set to 1.0.
XSPEC> cornorm 0. 1-2 0.3	! The correction norm for the last input range (which was 1-4) is set to 0., then files 1 and 2 are reset to 0.3.
XSPEC> cornorm 4	! File 4 has the correction also set to 0.3.
XSPEC> cornorm 1 4 -.3	! Files 1 and 4 are set to -.3.
XSPEC> cornorm .7	! File 4 (as the last input single range) is set to 0.7.

5.12 cosmo

Set the cosmology used (i.e., H0 and q0).

```
XSPEC> cosmo [<H0> [<q0>]]
```

where `<H0>` is the Hubble constant in km/s/Mpc and `<q0>` is the deceleration parameter.

Examples:

XSPEC> cosmo 100	! Set H0=100 km/s/Mpc

XSPEC> cosmo,,0. ! Set q0=0.

5.13 cpd

This command is an alias for `setplot device`.

```
XSPEC> cpd <PGPLOT device> or none
```

5.14 data

Input one or more spectral data files.

```
XSPEC> data <file spec>...
```

where <filespec> =:: [[<datagroup number>:]<data set number>] <filename>...

5.14.1 filename

If a particular file is not found or cannot be opened for input for some reason, then the user is prompted for a replacement file name. An <EOF> at this point is equivalent to typing `none`.

The default extension for all files is `.pha`, so all other extensions, (e.g. `.fak`) must be entered explicitly. The default directory is the current user directory when XSPEC is invoked.

When a new file is input, by default all its PHA channels are considered good channels for fitting and plotting purposes (see the `ignore` and `notice` commands), unless the file is replacing a previously-input file that had exactly the same number of total PHA channels, in which case the particular channels noticed are not modified.

If the file contains multiple spectra then the appropriate one can be specified by appending {N} to the end of the filename, where N is the row number of the spectrum in the file. Alternatively, the spectrum can be specified by {column_name=value} where column_name is the name of an ascii column in the table.

The individual spectral data files are created outside of XSPEC by detector-specific software. They are organized as `XSPEC data` files, but often referred to as PHA files. The PHA file contains such information as integration time, detector effective area, and a scaling factor that estimates the expected size of the internal background. The data file also contains the names of the default files to be used for background subtraction and for the detector sensitivity versus incident photon energy (the response file). A data file has the total observed counts for a number of channels and a factor for the size of any systematic error. Each channel is converted to a count-rate-per-unit area (assumed cm^2). The default background file is used for background subtraction, and an error term is calculated using Poisson statistics and any systematic error indicated in the file.

5.14.2 data set

Since XSPEC can fit simultaneously several sets of data (e.g., from multiple detectors), the `data` command will allow multiple files to be input. Elsewhere in XSPEC, each set of data is referred to by its associated data set number. `<data set number>`, distinguishable from the list of file names because it is an integer, is the data set number for the first `<filename>` in the specification, `<data set number>`+1 is the data set number for the next file, and so on. If no `<data set number>` is given in the first `<filespec>` it is assumed to be one (1), so that the first data file input is `data set 1`, the next file is `data set 2`, and so on. A skipped-over argument indicates that the data set for that position (as input in an earlier invocation of `data`) will continue to be used. If the filename input is `none`, that data set is completely removed and any higher-number data sets are renumbered.

The `data` command determines the current total number of data sets. If the command line is NOT terminated by a slash (/), the total number of datsets is the largest data set number given for any files explicitly input, or the largest value of a `<data set num>` argument. If the line is terminated by a slash (/), then the current number of data sets is the previous total number of datsets or the number as determined from the command line, whichever is greater. The exception to this rule is that if there are NO arguments to the `data` command, then the number of data sets is unchanged. (To remove all the data files from consideration would require a command like `data 0`.)

5.14.3 datagroup

XSPEC places each data set into a data group. Each data group has its own set of parameters for the defined model. These parameters can be either independent from data group to data group, or they can be linked across data groups using the standard XSPEC syntax. At present, each data group must have the same model, but some components can have normalizations set to zero. If no data group is specified, then the default is to place all data sets in the same data group. The model must be redefined when the number of data groups in use is changed.

Examples:

XSPEC> data a	! The file `a.pha` is read in as the first (and only) data set.
XSPEC> data,,b	! `b.pha` becomes the second data set, the first data set is unmodified (e.g. it is still `a.pha`).
XSPEC> data c 3 d,e,f	! `c.pha` replaces `a.pha` as the first data set; `d.pha`, `e.pha`, and `f.pha` provide the, third, fourth, and fifth data sets.
XSPEC> data g/	! `g.pha` replaces `c.pha` as the first data set; the slash (/) indicates that the 2nd through the 5th data sets remain as before.
XSPEC> data 2 none/	! The string none indicates that the 2nd data set (`b.pha`) is to be totally removed. The current total number of data sets thus becomes one less (4). The current data sets are `g.pha`, `d.pha`, `e.pha`, and `f.pha`.
XSPEC> data h,,	! The current total number of data sets becomes 2, the current data sets are from `h.pha` and `d.pha`.
XSPEC> data	! There is no change in the data status.
XSPEC> data 1	! The number of data sets is set explicitly to one, that being from `h.pha`.
XSPEC> data 1:1 a 2:2 b 3:3 c	! Read a.PHA into data group 1, `b.pha` into data group 2, and `c.pha` into data group 3.
XSPEC> data 1:1 a 1:2 b 2:3 c	! Read `a.pha` and `b.pha` into data group 1, and `c.pha` into data group 2.

XSPEC> data a{3}Read the third spectrum in the file `a.pha`.

5.15 delcomp

Delete components from the current model.

`XSPEC> delcomp <comp num range>`

where `<comp num range>` is range of positions in the model specification of the components to be deleted.

Examples: Suppose that the current model specification is 'wa po ga ga'.

XSPEC> delcomp 3-4	! Changes the model to wa po.
XSPEC> delcomp 1	! Changes the model to po

5.16 dummyrsp

Create a "dummy" response, covering a given energy range.

```
XSPEC> dummyrsp [<Low Energy> [<High Energy> [<No.  of ranges>
[<logarithmic or linear>]]]]
```

When examining several aspects of the current model, such as plotting it or determining flux, XSPEC uses the current evaluation array. This, in turn, is defined by the current response files being used, which depend on the various detectors. For example, low energy data sets (such as those from the EXOSAT LEs) may have responses covering 0.05 to 2 keV, while non-imaging proportional counters can span the range from 1 to 30 keV. If the user wishes to examine the behavior of the model outside of the current range, then he or she temporarily must create a dummy response file that will cause the model to be evaluated from `<Low Energy>` to `<High Energy>`, using `<no. of ranges>` as the number of steps into which the range is logarithmically or linearly divided. The initial default values for the arguments are .01 keV, 100 keV and 200 logarithmic energy steps. Note that the dummyrsp has no physically correct way of mapping the model into the data PHA channels, so the user should not try to fit–or plot–the data while the dummyrsp is active. The previous response matrices can be reimplemented with the `response` command, with no arguments. Any use of the `data` and `notice` commands also will replace the dummy response with a correct set of matrices.

Examples:

XSPEC> dummyrsp	! Create the dummy response with the default limits, initially .01, 100, and 200 bins.
XSPEC> dummyrsp .001 1	! Create a dummy response with 200 bins that cover the range from .001 to 1 keV.
XSPEC> dummyrsp ,,,500	! The same range, but now with 500 bins.
XSPEC> dummyrsp ,,,,lin	! The same range, but now with linearly spaced bins.
XSPEC> response	! Restore any previous correct responses.

5.17 dump

Write out a history package of observed and model spectra.

```
XSPEC> dump [<option>]
```

Two options currently are available: `ecdata` and `model`. `ecdata` writes out the data and folded model while `model` writes out the input model.

Examples:

XSPEC> dump	! The first dump command uses ecdata as default. It writes a history package containing the observed PHA spectrum and the folded model versus channel energy.
XSPEC> dump model	! Write history package for the model spectrum in photons/cm^2 s keV.

5.18 eqwidth

Determine the equivalent width of a model component.

```
XSPEC> eqwidth [[RANGE <frac range>] <model component number>]...
```

The command calculates the integrated photon flux produced by an additive model component (FLUX), the location of the peak of the photon spectrum (E), and the flux (photons per keV) at that energy of the continuum (CONTIN). The equivalent width is then defined as EW = FLUX / CONTIN in units of keV. The `continuum` currently is defined to be those other components of the selected model's additive group (see the `model` command). Thus, neither components in other additive groups nor the effect of any multiplicative components (e.g., absorption) are used in the calculation of the continuum or the equivalent width.

There are certain models with a lot of structure where, were they the continuum, it might be inappropriate to estimate the continuum flux at a single energy. The `RANGE` keyword allows the setting of a fractional size to be used over which the continuum model is integrated (from E∗(1-<fracr>) to E∗(1+<fracr>). The initial value for <fracr> is 0.05.

Examples: The current model is assumed to be M1 (A1 + A2 + A3 + A4 + M2 (A5)), where the Mx models are multiplicative and the Ax models are additive.

XSPEC> eqwidth 3	! Calculate the total flux of component A2 (the third component of the model) and find its peak energy (E). The continuum flux is found by the integral flux of A1+A3+A4, using the range of 0.95∗E to 1.05∗E to estimate the flux.
XSPEC> eqwidth range .1	! As before, but now the continuum is estimated from its behavior over the range 0.9∗E to 1.1∗E.
XSPEC> eqwidth range 0	! Now the continuum at the single energy range (E) will be used.
XSPEC> eqwidth range .05 2	! Now the component A1 is used as the feature, and A2+A3+A4 are used for the continuum. The range has been reset to the original value.
XSPEC> eqwidth 1	! Illegal, as M1 is not an additive component.
XSPEC> eqwidth 7	! Illegal, as there is no other additive component with A5 in its group to be used as the continuum.

5.19 error

Determine the confidence region for a model parameter.

```
XSPEC> error [[STOPAT <ntrial> <toler>] [MAXIMUM <redchi>] [<delta
fit statistic>] [<model param range>...]]...
```

where <model param range> =:: <first param>-<last param> determines the ranges of parameters to be examined, and `<delta fit statistic>` (distinguished from the model parameter indices by the inclusion of a period (.)), is the change-in-fit statistic used. Each indicated parameter is varied, within its allowed hard limits, until the value of chi-squared, minimized by allowing all the other non-frozen parameters to vary, is equal to the last value of chi-squared determined by the `fit` command is increased by the indicated `<delta fit statistic>`. Note that before the `error` command is executed, the data must be fitted. The initial default values are the range 1-1 and the delta chi-squared of 2.706, equivalent to the 90% confidence region for a single interesting parameter.

The number of trials and the tolerance for determining when the critical fit statistic is reached can be modified by preceeding them with the `STOPAT` keyword. Initially, the values are 10 trials with a tolerance of 0.01 in fit statistic. If a new minimum is found in the course of finding the error, then the calculation is aborted and control returned to the user.

The `MAXIMUM` keyword ensures that `error` will not be run if the reduced chi-squared of the best fit exceeds `<redchi>`. The default value for `<redchi>` is 2.0.

Examples: Assume that the current model has four model parameters.

XSPEC> error 1-4	! Estimate the 90% confidence ranges for each parameter.
XSPEC> error 9.0	! Estimate the confidence range for parameters 1-4 with delta fit statistic = 9.0, equivalent to the 3 sigma range.
XSPEC> error 2.706 1 3 1. 2	! Estimate the 90% ranges for parameters 1 and 3, and the 1. sigma range for parameter 2.
XSPEC> error 4	! Estimate the 1. sigma range for parameter 4.
XSPEC> error stop 20,,3	! Estimate the 1-sigma range for parameter 3 after resetting the number of trials to 20. Note that the tolerance field had to be included (or at least skipped over).

5.20 exit

The command to end the current XSPEC run.

```
XSPEC> exit
```

After an `exit`, the current plot files are closed. An <EOF> will have an identical result.

5.21 fakeit

Produce PHA files with simulated data.

```
XSPEC> fakeit [<file spec>...]
```

where <file spec> =:: [<file number>] <file name>... is similar to the syntax used in the `backgrnd`, `corfile`, and `response` command.

The `fakeit` command is used to create a number of artificial PHA files, where the current model is folded through response curves and then added to a background file. Photon statistics can be included optionally. The integration time, area, background scaling, and correction norm are requested for each file. By default, the background, response, correction file, and numerical information are taken from the currently-defined data. If the argument line is empty, then it is assumed that the number of `fakeit` files produced is equal to the current number of data sets. The file names input as arguments are PHA files to be used as background. In these cases, their default response files and numerical data are used as the defaults when creating the associated fake data file. If `none` is given as an argument, then the user is prompted for the response file to be used and the default numerical data is set to 1., except the correction norm, which is set to zero. For each file, the user will be prompted for a PHA file name. Poisson counting statistics are applied optionally when the data are created.

After the PHA files are created, they are read in automatically as the current datafiles. The ignore status is completely reset.

Examples: There are currently four data files, the second of which has a current background file `CURRB.PHA` and the last of which has no background being subtracted, equivalent to a background file `none`.

XSPEC> fakeit backa,,none 4	!	Four fake data PHA files are created. The first and second use BACKA.PHA and CURB.PHA for background, and their associated response files for the response. The third and fourth files will have no background files; the user will be prompted for the response file to be used. The 4 indicates that four data files will be created.

5.22 fit

Find the best fit model parameters for the current data.

```
XSPEC> fit [<no. iter> [<delta fit statistic>]]
```

where `<no. iter>` is the maximum no. of iterations, and `<delta fit statistic>` is the critical change in the fit statistic. The default values are 10 and 0.01, respectively. After each iteration, the value of chi-squared and the parameters are printed.

Examples:

XSPEC> fit	! Fit with the default number of iterations and critical delta chi-squared.
XSPEC> fit 60	! Fit with 60 as the number of iterations.
XSPEC> fit ,,1.e-5	! Fit with 1.e-5 as the critical delta.

5.23 flux

Calculate the flux of the current model between certain limits.

```
XSPEC> flux [<lowEnergy> [<hiEnergy>]]
```

where `<low Energy>` and `<hiEnergy>` are the values over which the flux is calculated. Initial default values are 2 to 10 keV. The flux is given in units of photons/cm^2/s and ergs/cm^2/s. The energy range must be contained by the range covered by the current data sets (which determine the range over which the model is evaluated). Values outside this range will be reset automatically to the extremes. Note that the energy values are two separate arguments, and are NOT connected by a dash (see parameter ranges in the `freeze` command).

Examples: The current data have significant responses to data within 1.5 to 18 keV.

XSPEC> flux	! Calculate the current model flux over the default range.
XSPEC> flux 6.4 7.0	! Calculate the current flux over 6.4 to 7 keV.
XSPEC> flux 1 10	! The flux is calculated from 1.5 keV (the lower limit of the current response's sensitivity) to 10 keV.

5.24 freeze

Do not allow indicated model parameters to vary. (See also `thaw`.)

```
XSPEC> freeze [<model parameter range>...]
```

where <model parameter range> =:: <model parameter no.> OR <model param no.>-<model param no.>. The indicated model parameter or range of model parameters will be marked so they cannot be varied by the `fit` command by setting their associated `<delta>` to less than zero (see `newpar` command). By default, the range will be the last range input by either a `freeze` or `thaw` command.

Examples: Currently there are six parameters, initially all unfrozen.

XSPEC> freeze 2	! Parameter 2 is frozen.
XSPEC> freeze 4-6	! Parameters 4, 5, and 6 are frozen.
XSPEC> thaw 2 3-5	! Parameters 2, 4, and 5 are thawed, parameter 3 is unaffected.
XSPEC> freeze	! Parameters 3,4,5 are frozen (the last range input by a freeze or thaw command).

5.25 gain

Modify a response file gain, in a particularly simple way. ∗CAUTION∗ This command is to be used with extreme care for investigation of the response properties. To properly fit data, the response matrix should be recalculated explicitly (outside of XSPEC) using any modified gain information derived.

```
XSPEC> gain < <file no.> <slope> <intercept>...
```

The initial default file number is 1; later, the default is the number of the file last modified. Initially, all responses are assumed to have nominal gains, determined implicitly by the data in the response files. This is equivalent to a `<slope>` of 1 and an `<intercept>` of zero.

The gain is automatically reset to nominal whenever a new data file or response file is defined for that file number, but not by any use of the `ignore`, `notice`, or `dummyrsp` commands.

5.25.1 Algorithm

The `gain` command shifts the energies on which the response matrix is defined and shifts the effective area curve to match. The effective area curve stored by XSPEC is either the ARF, if one was in use, or is calculated from the RSP file as the total area in each energy range. This means that if there are sharp features in the response then these will only be handled correctly by the gain command if they are in the ARF or if no ARF is input. The new energy, E', is related to the old energy, E, by E' = E/slope - intercept.

Examples: We assume that the current response for data set one is `rspfile`.

XSPEC> gain 1 0.98	! The first data set's response is adjusted with a slope of 0.98.
XSPEC> gain 1,,.03	! The offset also is moved now by 0.03 channels.
XSPEC> gain ,, 1 0	! Now the response is explicitly restored to the nominal value.
XSPEC> gain ,, 1.1 -.5	! Now it is adjusted, with slope 1.1 and offset -0.05.
XSPEC> response 1 none	! The response is removed.
XSPEC> response rspfile	! `rspfile.rsp` is used as the response, the gain is reset to nominal.

5.26 goodness

Perform a Monte Carlo calculation of the goodness-of-fit for the C-statistic

```
XSPEC> goodness [<no.  of realizations>]
```

This command simulates <no. of realizations> spectra based on the model and writes out the percentage of these simulations with C less than that for the data. If the observed spectrum was produced by the model then this number should be around 50%.

5.27 hardcopy

Make a hardcopy of the current plot and spool to the printer.

```
XSPEC> hardcopy
```

This command runs the command file `hardcopy.xcm`, which lives in the `xcoms` subdirectory of the XANADU library directory and may need to be altered for your system. Note that the plot device must be reset manually after a `hardcopy` command.

5.28 help

Obtain help on the XSPEC commands, their syntax, and examples of their use.

```
XSPEC> help [<topic list>]
```

The `help` command uses the XHELP interactive help facility. The optional topic list can be used to go directly to a particular topic and sub-topic. The `XHELP` facility has a number of special input characters used to access different sub-topics. The string `??` will produce a summary of these sub-topics. To exit back to XSPEC, use an <EOF> or type <CR> until you leave the help facility.

Examples: In the following example, only the results of the initial invocation (in response to the XSPEC> prompt) are given. Subsequent 'XHELP' characters and subtopic names can be given, until control is returned to XSPEC via the <EOF>.

XSPEC> help	! Go to the top of the help file.
XSPEC> help command fit	! Go to the help text for the `fit` command.
XSPEC> he com he exam	! Go to the example text for the `help` command.
XSPEC> help model pow	! Go to the help text for the `powerlaw` model.

5.29 history

Open a file for outputting history records.

```
XSPEC> history <filename> or NONE
```

where <`filename`> is the name of the file to be opened for output. The string `none` will close the current history file. The history file produces SF format packages that can be read by other programs. For example, the results of the XSPEC `steppar` command are placed in the `steppar` package, which in return is used by the XPLOT program to plot confidence contours. The history file is not directly human-readable. A more printable recounting of an XSPEC session is produced with the log file (see the `log` command). Currently implemented history packages:

data files	produced by	`data`, `fakeit`
assoc. files		`data`, `backgrnd`, `response`, `corfile`, `fakeit`
channels		`data,ignore,notice`
model		`addcomp`, `model`, `delcomp`
param def		`addcomp`, `model`, `define`, `delcomp`
param range		`newpar`, `addcomp`, `model`, `freeze`, `delcomp`
fit		`fit`
steppar		`steppar`
error		`error`
equiv width		`eqwidth`
flux		`flux`
flum		`lumin`
spectrum		`dump ecdata`
photon		`dump model`
time		`time`

5.30 ignore

Ignore data channels. (See also `notice`.)

```
XSPEC> ignore <range1> <range2> ... <rangeN>
```

where the Ith range is <rangeI> =:: <data set range>:<channel/energy range> or <channel/energy range>. If no <`data set range`> is given, then the previous range is used (the initial default range is file one (1) only). <data set range> =:: <init data set>–<last data set> or <single data set> where `init data set`>, <`last data set`>, and <`single data set`> are data set numbers, in the order that they were input with the `data` command. <channel/energy range> =:: <initial channel/energy>–<last channel/energy> or <single channel/energy>. If integers are given for the channel/energy ranges then channels will be used while if reals are given then energies. If only the last channel is indicated, then a default value of one (1) is used for the initial channel. Channels remain ignored until they are explicitly noticed with the `notice` command, or if a data set is replaced with another data set with a different number of PHA channels. When a channel is ignored, the counts and other arrays are compressed, freeing up array space. Thus, the energy range covered by the current response matrix may be reduced.

Examples: Assume that 4 data sets have been read in, the first 2 with 100 channels and the last 2 with 50 channels.

XSPEC> ignore **:1-10	! The first 10 channels of all 4 data sets are ignored.
XSPEC> ignore 80-**	! An attempt will be made to ignore channels ≥80 in all four data sets (as that was the last data set range specified). As a result,only channels 80-100 will be ignored for data sets 1 and 2. No change will occur for data sets 3 and 4, as they have no channels greater than 50.
XSPEC> ign 4:1-20 3:30-40 45-**	! Channels 11-20 for data set 4 are ignored (1-10 were ignored already) while channels 30-40 and 45-50 of data set 3 are ignored.
XSPEC> ignore 1:1-5	! No channels are ignored, as these were ignored at the beginning.
XSPEC> ign 2:1.-5.	! Ignore all channels between 1 and 5 keV in the second dataset.

5.31 iplot

Interactive plotting on the current plot device.

```
XSPEC> iplot <plot type>
```

This command works like the plot command (see the plot command description), but allows the user to change the plot and to add text to the plot interactively using the PLT package.

5.32 log

Open a log file, and control whether command lines are echoed.

```
XSPEC> log <log file> <log command chat.> <indirect adjustment>
```

where <log file> is the name of the file to be opened (default extension is .log). If no arguments are on the line, then the default file name is xspec.log. If <log file> matches the string none, then the current log file is closed. <log command chat.> is an integer that determines the importance of echoing the input command lines on the log file. The line is not echoed if <log command chat.> is greater than the current log chattiness level (see the chatter command). <indirect adjustment> is an integer added to the value of <log command chat.> for every nested level of indirection in the command processor (i.e., the use of command files). The default values of `<log command chat.>` and `<indirect adjustment>` are 10 and 1.

Examples: In these examples we assume that the current log chatter value is 10 and the commands are being processed in a command file that was invoked from the terminal level using the @ convention, so that the nesting level is one.

XSPEC> log	! Turn on the log file (default `xspec.log`). Commands will not be logged from the indirect file, but will be logged on return to the terminal level.
XSPEC> log none	! Close the log file.
XSPEC> log mylog 9	! Open the log file (`mylog.log`) and now even the indirect commands will be logged (at least for the first level of indirection).
XSPEC> log ,, 12 0	! Now even at the terminal level the commands are not logged.
XSPEC> chatter ,, 12	! Modifying the log file chattiness will restore command logging even at arbitrarily deep levels of indirection.

5.33 lumin

Calculate the luminosity of the current model for a given redshift and rest frame energy range.

```
XSPEC> lumin [<lowEnergy> [<hiEnergy>] [<redshift>]
```

where `<low Energy>` and `<hi Energy>` are the rest frame energies over which the luminosity is calculated and `<redshift>` is the source redshift. Initial default values are 2 to 10 keV for 0 redshift. The luminosity is given in units of ergs/s. The energy range redshifted to the observed range must be contained by the range covered by the current data sets (which determine the range over which the model is evaluated). Values outside this range will be automatically reset to the extremes. Note: The energy values are two separate arguments, and are NOT connected by a dash (see parameter ranges in the `freeze` command description).

Examples: The current data have significant response to data within 1 to 18 keV.

XSPEC> lumin,,,0.5	! Calculate the current model luminosity over the default range for z=0.5

XSPEC> lumin 6.4 7.0 ! Calculate the current luminosity over 6.4 to 7 keV.

5.34 model

Define the form of the theoretical model to be fit to the data.

```
XSPEC> model <component1> <component2> ... <componentN>
```

where `<componentJ>` is one of the models known to XSPEC. Descriptions of these models may be accessed using the `help model` command. The models are divided into two types: additive and multiplicative models. Additive models are those directly associated with emissivity, such as power laws, thermal models, emission lines, etc. The net effect of two independent additive models is just the sum of their individual emissivities. Multiplicative models, on the other hand, do not directly produce photons, but instead modify (by an energy-dependent multiplicative parameter) the spectrum produced by one or more additive components. Examples of multiplicative models are photoelectric absorption models, edges, absorption lines, etc.

A list of all the currently-installed models is given in response to the component `?`.

5.34.1 Algebraic Format

The model can be specified using an algebraic format where the plus sign (+) is used to add together components, and parentheses specify which subgroups a multiplicative component modifies. The algebraic format is translated by XSPEC into the correct format based on component order (see below). Whenever the model parameter list is written out, an algebraic description of the model is included. Note, however, that the correct use of `addcomp` and `delcomp` does require knowledge of the component order.

5.34.2 Rules for Combining Models

In order that a particular multiplicative model may be applied to some, but not all, additive models, the following syntax is used: the N components are grouped into subgroups of additive and multiplicative models:

```
XSPEC> [<mult. subgroup 0>] <group1> <group2> ... <groupI>
```

where each group is <groupJ> =:: <add. subgroupJ> [<mul. subgroupJ> or NULL], a string of one or more additive model components followed by an optional list of one or more multiplicative model components. The model named `null` is a special string used for punctuation to separate two adjacent additive subgroups without an intervening multiplicative model. Each model in a particular multiplicative subgroup is applied to the sum of all the additive models in the immediately preceeding subgroup. Any models in the optional multiplicative subgroup 0 are applied to the net spectrum from the sum of all the groups. Therefore, if Aij is the ith additive model in the jth group, and similarly, Mij is a multiplicative model, the total spectrum is given by

$$(\mathrm{M10}\ast\mathrm{M20}ast...)\ \ast\ \ [(\mathrm{A11}+\mathrm{A21}+...)\ast(\mathrm{M11}\ast\mathrm{M21}\ast...)] + [(\mathrm{A12}+\mathrm{A22}+...)\ast...] +...\ .$$

Each component may have one or more parameters that can be varied during the fit (see the `newpar` command write-up).

5.34.3 Background Models

A component can be specified as a background model by appending the string `/b` to the component name. A background model is convolved with the instrumental redistribution matrix but not with the effective area. This provides a means for including a description of detector-instrumental and particle backgrounds. If a response (RMF) matrix but not an auxiliary (ARF) file has been input, then the effective-area curve is calculated by assuming that the rows of the RMF sum to one. This will lead to errors at the top and bottom of the energy range of the detector.

Examples: Assume that M1,...,Mn are multiplicative models and A1,...,An are additive models.

XSPEC> model A1	!	The single component A1 is the model
XSPEC> model A1 A2	!	model = A1 + A2
XSPEC> model ,,,M1	!	model = (A1+A2)*M1
XSPEC> model M2/	!	model = M2*[A2*M1]
XSPEC> model ,,,,a1	!	model = M2*[A2*M1]+A1
XSPEC> model a1,a1,m1	!	model = (A1+A1)*M1 Note that though first and second components are the same form, their parameters are varied separately.
XSPEC> mo A1,null,A2,M1	!	model = A1 + [A2*M1]
XSPEC> mo M1(A1+M2(A2))	!	model = M1*[A1+M2*A2]

XSPEC> mo ,,,,,B1/b	! model is as above but with B1 included as a background model.

5.35 newpar

Adjust one or more of the model parameters.

```
XSPEC> newpar [<index range> [<parameter spec list> or <coupling
list>]]
```

where <parameter spec list> =:: <param value> <delta> <parameter range spec>, <parameter range spec> =:: <hard minimum> <soft minimum> <soft maximum> <hard maximum>, and <coupling list> =:: <" = " sign> <index coupled to first index> <multiplication factor for coupling> or <coupling list> =:: <" = " sign> <index coupled to first index> <" + " sign> <additive constant for coupling>. The model parameters are accessed through their model parameter indices. For example, the first parameter of the first model component generally is model parameter 1, etc. The first command line argument, `<index range>`, gives the indices' parameters to be modified by the `newpar` command. The default value is the range from the previous invocation of `newpar`. The remaining arguments can be used to update the parameter specification. If the parameter specification is omitted from the command line, then the user is explicitly prompted for it.

The first two arguments of the parameter specification are:

<param value>	! the trial value of the parameter used initially in the fit;
<delta>	! the "step size" used in the numerical determination of the derivatives used in the fitting process, when delta is set to zero, the parameter is not adjustable during the fit.

The four arguments of the range specification determine the range of acceptable values for the parameter. The soft limits should include the range of expected parameter behavior. Between the hard and soft limits, the parameter is made stiffer to adjustment by the minimization routine invoked by the `fit` command. The parameter never is allowed to have a value at or outside the hard limits. (One exception to this rule: the parameter can equal one of the hard limits if the soft limits have been placed outside the hard limit).

A carriage return or a slash (/) will terminate the input of the parameter specification. The series of prompted specification inputs can be explicitly terminated by an <EOF>.

Coupling of parameters allows two or more parameters in a model to always have the same value (or to differ by a constant multiplicative factor) and to have these coupled parameters free in the fit. An equal sign (=) after the `<first index>` indicates that coupling of parameters is desired. Then the `<index coupled to first index>` and a `<multiplication factor for coupling>` (default=1.0) follow.

Examples: The total number of model parameters for the example is four.

XSPEC> newpar 2 0.1	! The value of the second parameter is set to 0.1.
XSPEC> newpar 3	! The program will prompt for a specification for the 3rd parameter,
0.001,0	! which has its value set to 0.001 and its delta set to zero, fixing it in later fits. The program now prompts for a specification for the 4th parameter,
21	! which is set to 21. As there is no 5th parameter, the program displays a summary and returns to command level.
XSPEC> newpar ,,.001	! The value of the delta of the 3rd parameter (which is the default index as it was the first parameter modified in the previous newpar invocation) is set to 0.001, allowing it to be adjusted during any fits.
XSPEC> newpar 0	! No parameters are modified, but a summary of the current parameter values will be printed.

The total number of parameters for this example is six.

XSPEC> newpar 4 = 1	! The value of parameter 4 is set to the value of parameter 1. This has the consequence of model parameter 4 being frozen at the value of parameter 1 during subsequent fitting procedures. If model parameter 1 is a free parameter, then both parameters 1 and 4 change their values simultaneously in the fit procedure. Note the spaces around the "=" sign!
XSPEC> newpar 4 = 1 0.9	! The value of parameter 4 is set to 0.9 times the value of parameter 1.
XSPEC> newpar 6 = 3 .1	! The value of parameter 6 is set to 0.1 times the value of parameter 3.
XSPEC> thaw 6	! Makes parameter 6 independent of parameter 3 and a free parameter.

5.36 notice

Notice data channels. (See also `ignore`.)

```
XSPEC> notice <range1> <range2> ... <rangeN>
```

where the Ith range is <rangeI> =:: <data set range>:<channel/energy range> or <channel/energy range>. If no `<data set range>` is given, then the previous range is used (the initial default range is file one (1) only). <data set range> =:: <init data set>–<last data set> or <single data set>, where `<init data set>`, `<last data set>`, and `<single data set>` are data set numbers, in the order they were input with the `data` command. <channel/energy range> =:: <initial channel/energy>–<last channel/energy> or <single channel/energy>. If integers are given for the channel/energy ranges then channels will be used while if reals are given then energies. If only the last channel is indicated, then a default value of 1 is used for the initial channel. Channels remain noticed until they are explicitly ignored with the `ignore` command. When a data set is replaced by another data set with a different number of PHA channels, all input channels automatically are noticed. When a channel is noticed, the response matrix must be recreated to include the new channel's response. Thus all the response files are re-read (this might take awhile). The energy range covered by the current response matrix may thereby be increased.

Examples: Assume that 4 data sets have been read in, the first 2 having 100 channels and the last 2 having 50 channels. Assume also that channels 1-10 of all four data sets are ignored and that channels 80-100 of data sets 1 and 2 are ignored.

XSPEC> notice **:1-10	! The first 10 channels of all 4 data sets are noticed.
XSPEC> notice 80-**	! An attempt will be made to notice channels $\geq$80 in all 4 data sets (as that was the last data set range specified) but the result is that only channels 80-100 will be noticed for data sets 1 and 2, with no change for data sets 3 and 4 as they have no channels greater than 50.
XSPEC> notice 1:1-5	! No channels are noticed, as these channels were noticed in the beginning.

5.37 pause

Exit the program using the FORTRAN pause command.

```
XSPEC> pause
```

VAX VMS specific notes: The program is exited with a FORTRAN `pause` statement. This allows the user to execute a limited selection of VMS commands (those that do not effectively run a new process) and then return to XSPEC using the VMS `continue` command. In particular, the default directory can be modified.

Examples:

XSPEC> pause	! Leave XSPEC with the Fortran pause.
$ set default [common.newdir]	! Reset the current default directory to the `[common.newdir]` directory.
$ continue	! Return to XSPEC
XSPEC> data filenew	! Read in a new data set `filenew.pha` from the new directory.

5.38 plot

Make one or more plots to the current plot device (see `setplot device`).

```
XSPEC> plot <plot type> <plot type>
```

`<plot type>` is a keyword describing the various plots allowed. Any of the options `counts`, `data`, or `ldata` can be followed by a second option, which should be one of `chisq`, `delchi`, `ratio`, `residuals`, or `none`. If one of these second options is given, then a two-pane plot is drawn with the first option in the upper pane and the second option in the lower. (see also `iplot`)

5.38.1 chisq

Plot contributions to `chisq`. The contribution is plotted +ve or -ve depending on whether the residual is +ve or -ve.

5.38.2 contour

Plot a two-dimensional fit-statistic contour plot of the last `steppar` run.

```
XSPEC> plot contour [<min. fit stat.> [<num. levels> [<levels>]]]
```

where `<min. fit stat.>` is the minimum fit statistic relative to which the delta fit statistic is calculated, `<num. levels>` is the number of contour levels to use and <levels> := <level1> ... <levelN> are the contour levels in the delta fit statistic.

`contour` will plot the fit statistic grid calculated by the last `steppar` command (which should have gridded on two parameters). A small plus sign (+) will be drawn on the plot at the parameter values corresponding to the minimum found by the most recent fit.

problems

The fit statistic confidence contours are often drawn based on a relatively small grid (i.e., 5x5). To fully understand what these plots are telling you, it is useful to know a couple of points concerning how the software chooses the location of the contour lines.

The contour plot is drawn based only on the information contained in the sample grid. For example, if the minimum fit statistic occurs when parameter 1 equal 2.25 and you use `steppar 1 1.0 5.0 4`, then the grid values closest to the minimum are 2.0 and 3.0. This could mean that there are no grid points where delta-fit statistic is less than your lowest level (which defaults to 1.0). As a result, the lowest contour will not be drawn. This effect can be minimized by always selecting a `steppar` range that causes XSPEC to step very close to the true minima. For the above example, using `steppar 1 1.25 5.25 4`, would have been a better selection.

The location of a contour line between grid points is designated using a linear interpolation. Since the fit statistic surface is often quadratic, a linear interpolation will result in the lines being drawn inside the true location of the contour. The combination of this and the previous effect sometimes will result in the minimum found by the `fit` command lying outside the region enclosed by the lowest contour level.

Examples:

steppar 2 0.5 1. 4 3 1. 2. 4	! Create a 5*5 grid for parameters 2 and 3
plot contour	! Plot out a grid with three contours with delta fit statistic of 2.3, 4.61 and 9.21
plot cont,,4,1.,2.3,4.61,9.21	! same as above, but with a delta fit statistic = 1 contour.

5.38.3 counts

Plot the data (with the folded model, if defined) with the y-axis being numbers of counts in each bin.

5.38.4 data

Plot the data (with the folded model, if defined).

5.38.5 delchi

Plot the residuals in terms of sigmas with error bars of size one.

5.38.6 efficien

Plot the total response efficiency versus incident photon energy.

5.38.7 eufspec

Plot the unfolded spectrum and the model in Ef(E). The contributions to the model of the various additive components also are plotted.

5.38.8 insensitv

Plot the insensitivity of the current data set to changes in the incident spectra (experimental).

5.38.9 ldata

Plot the data (with the folded model, if defined) with a logarithmic y-axis.

5.38.10 model

Plot the current incident model spectrum (Note: This is NOT the same as an unfolded spectrum.) The contributions of the various additive components also are plotted.

5.38.11 ratio

Plot the data divided by the folded model.

5.38.12 residuals

Plot the data minus the folded model.

5.38.13 sensitvty

Plot the sensitivity of the current data set to changes in the incident spectra (experimental).

5.38.14 sum

A pretty plot of the data and residuals against both channels and energy.

5.38.15 ufspec

Plot the unfolded spectrum and the model. The contributions to the model of the various additive components also are plotted.

5.39 prompt

Change the command prompt.

```
XSPEC> prompt <command prompt>
```

5.40 query

Switch on/off the continue fitting questions.

```
XSPEC> query <option>
```

5.40.1 no

Assume that the answer to the continue fitting question is no.

```
XSPEC> query no
```

The continue fitting question in `fit`, `steppar`, and `error` will not be asked when the number of trials is exceeded. Also, when the number of trials to find the `error` is exceeded, a question will not be asked. In both cases the program will behave as if a `no` answer was given to the question, that is, it will stop and return to the XSPEC prompt.

5.40.2 on

Switch on the continue fitting question.

```
XSPEC> query on
```

The continue fitting question in `fit`, `steppar`, and `error` will be asked when the number of trials is exceeded. Also when the number of trials to find the `error` is exceeded a question will be asked.

5.40.3 yes

Assume that the answer to the continue fitting question is yes.

```
XSPEC> query yes
```

The continue fitting question in `fit`, `steppar`, and `error` will not be asked when the number of trials is exceeded. Also, when the number of trials to find the `error` is exceeded, a question will not be asked. In both cases the program will behave as if a `yes` answer was given to the question, that is, it will carry on.

5.41 quit

The same as `exit`

```
XSPEC> quit
```

5.42 recall

Recall a previous command.

```
XSPEC> recall [<number>]
```

Without an argument, `recall` lists the commands stored. Note that `recall` runs-off a file, so commands from a previous session can be recovered.

5.43 recornrm

Adjusts the indicated data sets' correction norm by a single multiplicative factor that minimizes the fit statistic. This approach to fitting the correction norm is considered an INTERIM solution.

```
XSPEC> recornrm [<file range>...]
```

where <data set range> := <low data set>[–<high data set>] (see the notes on `ranges` under input formats). All the data sets specified by the one or more data set ranges on a single invocation of the command are multiplied by a single number, which is chosen to best reduce the fit statistic. If no ranges are given, then the last SINGLE range input is used. If you wish to fit a data set's correction norm individually, then refer only to that data set.

Note: The use of the `recornom` command requires that a response and model be defined. It is perhaps best if a preliminary fit is performed. The user then should alternate between `fit`'s and `recornrm`'s until a stable solution is achieved. If the model is not a good fit, this process may not converge.

Examples:

XSPEC> recornrm **	! All the files' correction norms are adjusted by a single number.
XSPEC> recornrm 1-3 5	! Files 1,2,3, and 5 are adjusted.
XSPEC> recornrm	! File 5 (the last range input) is adjusted by itself.
XSPEC> recornrm 2	! File 2 is adjusted

5.44 reexecute

`reexecute`, with no argument, provides a list of up to 20 of the last commands. A desired selection of these can then be re-executed by number. This command DOES NOT recall previous commands to the command line for changing. It is useful for repeating a sequence of commands after data files have been changed.

```
XSPEC> reexecute [<command number list>]
```

where `<command number list>` is a list of numbers and/or number ranges that denote previously-called commands. If no `<command number list>` is given, then a list of up to 20 of the last commands is provided.

Examples:

XSPEC> reex	! Provides list of the last commands (maximum 20).
XSPEC> reex 4 15-18	! Re-executes commands 4,15,16,17,18.

5.45 renorm

Renormalize model, or change `renorm` conditions.

```
XSPEC> renorm [AUTO or NONE or PREFIT]
```

The `renorm` command will adjust the normalizations of the model by a single multiplication factor, which is chosen to minimize the fit statistic. Such a renorm will be performed explicitly whenever the command is used without a key-word, or during certain XSPEC commands, as determined by the following key-words:

- AUTO – Renormalize after a `model`, `newpar`, and before a `fit`.

- PREFIT – Renormalize only before a `fit`.

- NONE – Perform no automatic renormalizations, i.e., only perform them when a `renorm` command is given explicitly.

5.46 response

Modify one or more of the matrices used to describe the response of the associated data set to incident X-rays.

```
XSPEC> response [<filespec>...]
```

where <filespec> =:: [<data set num>] <file name>..., where `<file name>` is the name of the response file to be used for the response of the associated data set. `<data set num>` is the data set number for the first `<file name>` in the specification, `<data set num>`+1 is the number for the next file, and so on. If no `<data set num>` is given in the first `<filespec>`, it is assumed to be one (1). If no file specifications are entered, then none of the data set responses are modified. An error message is printed if the data set number is greater than the current number of data sets (as determined from the last use of the data command. A file name `none` indicates that no response is to be used for that data set. This situation means that any incident spectrum will produce no counts for those particular channels. If a file is not found or can not be opened for input, then the user is prompted for a replacement response file. An <EOF> at this point is equivalent to using `none` as the response. See the `data` command for ways to totally remove the data set from consideration. The user also is prompted for a replacement if the response file has a different number of PHA channels than the associated data set. A warning will be printed out if the response detector ID is different from the associated data set's.

The current ignore status for channels is not affected by the command. (See the `ignore` and `notice` commands).

Examples: It is assumed that there are currently three data sets:

XSPEC> response a,b,c	! New files for the response are given for all three files.
XSPEC> response 2 none	! No response will be used for the second file.
XSPEC> response,,d	! `d.pha` becomes the response for the second file.

5.47 save

Save aspects of the current state to a command file.

```
XSPEC> save <option> <filename>
```

If no `<file name>` is given, then the file `savexspec.xcm` is created. If you don't give the extension to the file name the default is `.xcm`. The `<option>`s currently allowed are `model`, `files` and `all`. The `model` option writes out commands to recreate the current model and parameter values, the `files` option writes out commands to read-in the current data sets, and the `all` option does both of the above. The default option is `model`. To recover the saved context use the command `@filename`.

Examples:

XSPEC> save,,fname	! Write out model commands to the file `fname.xcm`.
XSPEC> save	! Same as above into file `savexspec.xcm`.
XSPEC> save files fname	! Write out data file commands.

5.48 script

Open a script file.

```
XSPEC> script <script file>
```

where `<script file>` is the name of the file to be opened (default extension is `.xcm`). If no arguments are on the line, then the default file name is `xspec.xcm`. If `<script file>` matches the string `none`, then the current script file is closed. The script file saves all commands that are input. This command is useful for users who use the same set of commands repeatedly. Once a script file is written and saved, the user then can re-run the same set of commands on other data simply by opening the script file.

Examples:

XSPEC> script	! Turn on the script file (default `xspec.xcm`).
XSPEC> script none	! Close the script file.
XSPEC> script myscript	! Open the script file (`myscript.xcm`).

5.49 setplot

Set one of the various plot options.

```
XSPEC> setplot <subcommand string>
```

where <`subcommand string`> is a keyword followed in some cases by arguments.

5.49.1 add

Show individual additive model components on the data plots.

```
XSPEC> setplot add
```

The opposite is `setplot noadd`.

5.49.2 channel

Change the x-axis on data and residual plots to channels.

```
XSPEC> setplot channel
```

5.49.3 command

Add a PLT command to the command list.

```
XSPEC> setplot command <PLT_command>
```

where PLT_command is any valid PLT command. Every time you use `setplot command`, that command is added to the list that is passed to PLT when you use `plot` or `iplot`. The most common use of `setplot command` is to add a common label to all plots produced. You should be careful when using this command, because XSPEC does not check to see if you have entered a valid PLT command. These commands are appended to the list that XSPEC creates to generate the plot and so `setplot command` will override these values (this can either be a bug or a feature, depending on what you have done!)

See also `setplot delete` and `setplot list`.

Example:

XSPEC> setp c LA OT Crab	! Add the label "Crab" to future plots.

5.49.4 delete

Delete a PLT command from the command list.

```
XSPEC> setplot delete command_number
```

where `command_number` is the number of a PLT command that had been entered previously using `setplot command`. This command is used to delete commands from the list passed to PLT when you use the XSPEC `plot` or `iplot` commands.

Examples:

XSPEC> setp c LA OT Testing	! PLT label command
XSPEC> setp c LWidth 5	! PLT line-width command
XSPEC> setplot lis	! List the PLT command stack.
1: LAbel OT Testing	!
2: LWidth 5	!
XSPEC> setplot del 1	! Delete the first command in the stack.
XSPEC> setplot lis	!
1: LWidth 5	!

5.49.5 device

Set the device used for plots.

```
XSPEC> setplot device <PGPLOT device> or NONE
```

where `<PGPLOT device>` is a string that gives an (optional) file name and device plot type (see below and the PGPLOT documentation for options available). Entering the device `none` will close the current plot device and disable plotting until a subsequent `setplot device` is issued. Note that if an environment (UNIX) or logical (VMS) variable called `PGPLOT_TYPE` exists, then XSPEC automatically sets it as the plot device on invocation.

PGPLOT devices A number of plot device types are supported in XSPEC. Useful options for hardcopy are provided by the `hardcopy` command at the XSPEC level and the `@hard` command at the PLT command level. Both of these options run from command files that may be found in the **$XANADU/lib/coms** or **XANADU:[LIB.COMS]** directory. These command files may need to be altered for your system.

sun The PGPLOT devices for the Sun are :

```
/gf             ! Graphon GO-230
/null           ! Null device
/ps             ! Postscript file
/retro          ! Retrographics VT640
/tek4010        ! Tektronix-4010 VT640
/tk4100         ! Tektronix-4100
/vt125          ! VT125, VT240, or VT241 terminals
/vps            ! Portrait mode Postscript file
/xdisp          ! pgdisp or figdisp server
/xterm          ! XTERM Tek terminal emulator
/xwindow        ! Window on Xwindow server
/ppmwindow      ! A PPM file (produced via an Xwindow)
/xgwindow       ! Window on Xwindow server
```

dec The PGPLOT devices for DEC/Ultrix and Alpha/OSF are :

```
/gf             ! Graphon GO-230
/null           ! Null device
/ps             ! Postscript file
/retro          ! Retrographics VT640
/tek4010        ! Tektronix-4010 VT640
/tk4100         ! Tektronix-4100
/xterm          ! XTERM Tek terminal emulator
/xwindow        ! Window on Xwindow server
/vt125          ! VT125, VT240, or VT241 terminals
/xgwindow       ! Window on Xwindow server
/ppmwindow      ! A PPM file (produced via an Xwindow)
```

vms The PGPLOT devices for VAX/VMS are :

```
/gf             ! Graphon GO-230
/hidmp          ! Houston Instruments HIDMP pen plotter
/file           ! Graphics MetaFile device
/null           ! Null device
/pk             ! Peritek Corp. VCH-Q frame-buffer video
/ps             ! Postscript file
/printronix     ! Printronix P300 or P600 dot-matrix printer
/peritek        ! Peritek Corp. VCH-Q frame-buffer video
/retro          ! Retrographics VT640
/tek4010        ! Tektronix-4010 VT640
/tfile          ! Tektronix-4014 (12 bit addressing)
/tk4100         ! Tektronix-4100
/versatec       ! Versatec V80 dot-matrix printer
/vps            ! Portrait mode Postscript file
/vt125          ! VT125, VT240, or VT241 terminals
/ws0            ! VAX/VMS workstation
/x11            ! DECwindows
/sx11           ! DECwindows with sockets
```

Examples:

XSPEC> setplot device /te	! sets the device to the tektronix.
XSPEC> setplot device none	! closes the plot file.

5.49.6 energy

Change the x-axis on data and residual plots to energy in units of keV.

```
XSPEC> setplot energy
```

5.49.7 group

Define a range of data sets to be in the same group for plotting purposes only.

```
XSPEC> setplot group <data set range>...
```

where <data set range> is a range of contiguous data sets to be treated as a single data set for plotting purposes. The data sets still are fit individually. If multiple ranges are given, each range becomes a single group. Initially, all data sets read in are treated as single data sets.

Examples: Assume that there are five data sets currently read in, all of them ungrouped initially.

XSPEC> setplot group 1-4	! The first four data sets are treated as one group, with the fifth data sets on its own. Thus all plots will appear to have two data sets.
XSPEC> setplot group 1 2 3 4	! The data sets are reset to each be in their own group.
XSPEC> setplot group 2-3 4-5	! Now there are three plot groups, being data set 1, by itself, and data sets 2-3 and 4-5 as groups.
XSPEC> setplot group 1-**	! All the data sets are placed in a single plot group.

5.49.8 list

List all the PLT commands in the command list.

```
XSPEC> setplot list
```

See `setplot delete` for an example of use.

5.49.9 noadd

Do not show individual additive model components on the data plots.

```
XSPEC> setplot noadd
```

The opposite is `setplot add`.

5.49.10 rebin

Define characteristics used in rebinning the data (for plotting purposes ONLY).

```
XSPEC> setplot rebin <min. significance> <max no. bins>
```

In plotting the data from a data set (or group of data sets, see `setplot group`), adjacent bins are combined until they have a significant detection at least as large as `<min. significance>` (in sigma). However, no more than `<max no. bins>` may be so combined. Initial values are 0. and 1, respectively. This arguement effects only the presentation of the data in plots. It does not change the fitting, in particular the number of degrees of freedom. The values given are applied to all the plotted data.

Examples:

XSPEC> setplot rebin 3. 5	! Bins are plotted that have at least 3 sigma, or are grouped in sets of 5 bins.
XSPEC> setplot rebin 5	! The significance is increased to 5 sigma.
XSPEC> setplot rebin,,10	! The plotted bins can be grouped into up to 10 bins in reaching the 5 sigma significance criterion.

5.49.11 wave

Change the x-axis on data and residual plots to wavelength in units of angstroms.

```
XSPEC> setplot wave
```

5.50 show

List selected information to the user's terminal (and the log file, if open).

```
XSPEC> show [<selection>]
```

where <selection> is a key word to select the information to be printed. If omitted, it is the information last asked for. Initially, the default selection is `all`. Selections are:

all	! All the information.
allfile	! All file information = files+noticed+rates.
files	! File names, associated coefficients.
noticed	! Channel ranges noticed for each file.
parameters	! Current model parameter values.

pha	! Current data, error and model values for each channel.
rates	! Folded model, correction rates for each file.

Note that `show pha` requires the chatter level to be set to ≥ 11.

5.51 statistic

Change the fit statistic in use.

```
XSPEC> statistic (chi or ml)
```

The options are chi-squared (chi) or C-statistic (cstat). Note that the C-statistic has a couple of limitations : a) it cannot be used on background-subtracted data; b) if the calculated model goes to zero in any channel, then the format statistic goes to infinity - in practice this is trapped and a large value of the statistic should result.

5.52 steppar

Perform a fit while stepping the value of a parameter through a given range. Useful for determining confidence ranges in situations where greater control is needed than given with the `error` command.

```
XSPEC> steppar <step spec.> [<step spec.>...]
```

where <step spec.> ::= [LOG or NOLOG] [current or best] <param index> <low value> <high value> <no. steps>. The indicated parameter is stepped from `<low value>` to `<high value>` in `<no. steps>+1` trials. The stepping is either linear or log. Initially, the stepping is linear but it can be changed by the optional string `log` before the parameter index. `nolog` will force the stepping to be returned to the linear form. The number of steps is set initially to 10. At each value, the parameter is frozen, a fit performed, and the resulting value of chi-squared given. If `best` is given as an argument then the non-stepped parameters are reset to the best-fit values at each grid point. Alternatively, if `current` is given as an argument then the non-stepped parameters are started at their values after the last grid point (the default).

If multiple `<step spec.>` are given for different parameters, then a raster scan of the parameter ranges is performed. At the end of the set, the parameters and chi-squared are restored to the values they had initially.

Examples: Assume that the current model has four parameters:

XSPEC> steppar 3 1.5 2.5	!	Step parameter 3 from 1.5 to 2.5 in steps of .1.
XSPEC> steppar log	!	Repeat the above, only use multiplicative steps of 1.0524.
XSPEC> step nolog 2 -.2 .2 20	!	Step parameter 2 linearly from -.2 to .2 in steps of 0.02.

5.53 suggest

Get advice on what to do with bug reports and enhancement requests.

```
XSPEC> suggest
```

If the suggestion is the result of a catastrophic error, please give the traceback list, particularly the line number and name of the last XSPEC subroutine. Clarity and completeness will improve the response to your suggestion.

5.54 systematic

```
XSPEC> systematic [<model systematic error>]
```

Set a systematic error term on the model to be added in quadrature to that on the data when evaluating chi-squared. The default value is zero.

5.55 thaw

Allow indicated parameters to vary. (See also `freeze`.)

```
XSPEC> thaw [<parameter range>...]
```

where <parameter range> =:: <parameter no.> or <param no.>–<param no.>. The indicated parameter, or range of parameters, will be marked as variable by the fitting commands, by setting their associated <delta> to greater than zero (see `newpar` command). (Note that parameters with <delta> values equal to zero are not affected. The `newpar` command must be used to change these.) By default, the range will be the last range input by either a `freeze` or `thaw` command. See the `freeze examples` for an example of the use of the `thaw` command.

5.56 thleqw

Determine the expected line equivalent width of a fluorescent line.

```
XSPEC> thleqw [<line energy> <line fluorescent yield> <absorption
edge energy> <upper energy limit> <bin number> <log or lin> <max
edge depth>]
```

The `thleqw` command integrates the photon flux absorbed in an edge with the maximum depth from the edge energy to the upper energy limit. To get good resolution in the edge, the command uses the `dummyrsp` command with the edge energy as low energy, the upper energy limit as high energy and the bin number for the number of ranges (see `dummyrsp` command). The calculation for the edge absorption is done in the same way as in the `edge` command. The fluorescent yield is used to calculate the photon flux going into a fluorescent line emission. The continuum flux is calculated in the same way as it is in the `eqwidth` command. No lines are added to the continuum.

This model assumes a spherically symmetric distribution of absorbing material around the X-ray source with a column density obtained by the `wabs` model. The defaults are 1.0 keV, 0., .01 keV, 100. keV, 200, and 0.0, which gives 0 for the equivalent width. They are chosen this way since the `dummyrsp` command defaults are 0.01 keV, 100. keV and 200.

Since the `thleqw` command uses `dummyrsp`, the old response must be read in again when working further on the same spectrum.

Examples:

XSPEC> thl 6.4,.34,7.1,20.,,,.015	! Calculate the expected equivalent width for a line at 6.4 keV with a fluorescent yield of 34%. The absorption edge is integrated from 7.1 keV to 20. keV in 200 steps. The maximum depth is .015.
XSPEC> thleqw ,,,,,,100	! As before, but with 100 steps
XSPEC> thleqw	! As before.

Don't forget to read the response file in again if you want to do further work on the same data.

5.57 time

Get some information about the program run time.

```
XSPEC> time
```

Provides the following:

elapsed real time
elapsed CPU time
buffered I/O count
direct I/O count
page fault count

5.58 uncertain

A synonym for `error`.

5.59 untie

Untie the specified parameter from any links to other parameters.

```
XSPEC> untie <parameter>
```

5.60 weight

Change the weighting function used in the calculation of chi-squared.

```
XSPEC> weight [standard | special]
```

Standard weighting uses $\sqrt{N}$ or the statistical error given in the input spectrum while special weighting uses $1 + \sqrt{N + 0.75}$, a better approximation when N is small.

Chapter 6

XSPEC V9 Models

6.1 Summary of Models

`absori`	-	Ionized absorber.
`ascac`	-	ASCA PSF mixing model.
`atable`	-	Additive table model.
`bbody`	-	Blackbody spectrum.
`bbodyrad`	-	Blackbody spectrum with norm proportional to surface area.
`bknpower`	-	Broken power law.
`bremss`	-	Thermal bremsstrahlung.
`cabs`	-	Compton scattering (non-relativistic).
`cemekl`	-	Multi-temperature mekal.
`cevmkl`	-	Multi-temperature vmekal.
`cflow`	-	Cooling flow model.
`compls`	-	Comptonization spectrum after Lamb and Sanford 1979.
`compst`	-	Comptonization spectrum after Sunyaev and Titarchuk 1980.
`comptt`	-	Comptonization spectrum after Titarchuk 1994.
`constant`	-	Energy-independent multiplicative factor.
`cp6mkl`	-	6th-order Chebyshev polynomial DEM using MKL.
`cp6vmkl`	-	6th-order Chebyshev polynomial DEM using variable abund. MKL.
`cutoffpl`	-	Power law with high energy exponential cutoff.
`cyclabs`	-	Cyclotron absorption line.
`disk`	-	Disk model.

`diskbb`	-	Multiple blackbody disk model.
`diskline`	-	Line emission from relativistic accretion disk.
`diskm`	-	Disk model with gas pressure viscosity.
`disko`	-	Modified blackbody disk model.
`dust`	-	Dust scattering out of the beam.
`edge`	-	Absorption edge.
`etable`	-	Table model for exponential of -1 times the input.
`expabs`	-	Low-energy exponential cutoff.
`expfac`	-	Exponential factor.
`gaussian`	-	Simple gaussian line profile.
`highecut`	-	High energy cutoff.
`hrefl`	-	Simple reflection model good up to 15 keV.
`meka`	-	Mewe-Gronenschild-Kaastra thermal plasma (1992).
`mekal`	-	Mewe-Kaastra-Liedahl thermal plasma (1995).
`mkcflw`	-	Cooling flow model based on meka.
`mtable`	-	Multiplicative table model.
`notch`	-	Notch line absorption.
`nteea`	-	Pair plasma model.
`pcfabs`	-	Partial covering fraction absorption.
`pegpwrlw`	-	Power law with pegged normalization.
`pexrav`	-	Exponentially cut-off power-law reflected from neutral matter.
`pexriv`	-	Exponentially cut-off power-law reflected from ionized matter.
`phabs`	-	Photo-electric absorption
`plabs`	-	Absorption model with power-law dependence on energy.
`pliref`	-	Power law illumination of ionized material.
`plrefl`	-	Power law illumination of cold material.
`powerlaw`	-	Simple photon power law.
`raymond`	-	Raymond-Smith thermal plasma.
`smedge`	-	Smoothed absorption edge.
`spline`	-	Spline multiplicative factor.
`SSS_ice`	-	Einstein Observatory SSS ice absorption.
`step`	-	Step function convolved with gaussian.
`tita_a`	-	Compton reflection from neutral matter.
`tsabs`	-	Thomson scattering.
`useradd1`	-	The user defined additive model.
`usermul1`	-	User defined multiplicative model.

`uvred`	-	UV reddening.
`varabs`	-	Photo-electric absorption with variable abundances.
`vbremss`	-	Thermal bremsstrahlung spectrum with variable H/He.
`vmeka`	-	M-G-K thermal plasma with variable abundances.
`vmekal`	-	M-K-L thermal plasma with variable abundances.
`vmcflow`	-	Cooling flow model based on vmeka.
`vraymond`	-	Raymond-Smith thermal plasma with variable abundances.
`wabs`	-	Photo-electric absorption.
`wndabs`	-	Photo-electric absorption with low energy window.
`zbbody`	-	Redshifted blackbody.
`zbremss`	-	Redshifted thermal bremsstrahlung.
`zedge`	-	Redshifted absorption edge.
`zgauss`	-	Redshifted gaussian.
`zhighect`	-	Redshifted high energy cut-off.
`zpcfabs`	-	Redshifted partial covering absorption.
`zphabs`	-	Redshifted photoelectric absorption.
`zplrefl`	-	Redshifted plrefl model.
`zpowerlw`	-	Redshifted power law.
`zvarabs`	-	Redshifted photo-electric absorption with variable abundances.
`zvfeabs`	-	Redshifted absorption with variable iron abundance.
`zwabs`	-	Redshifted “Wisconsin absorption.”
`zwndabs`	-	Redshifted photo-electric absorption with low energy window.

6.2 Model descriptions

This chapter contains information on specific, installed XSPEC models, as well as the use of the user-defined model facility. The parameters are given as `par1`, `par2`, ... and `K`, which is the normalization. Additive models have formulae giving $A(E)$, while multiplicative models have formulae giving $M(E)$.

6.3 absori

An ionized absorber calculated by Magdziarz and Zdziarski (MNRAS 1995).

`par1`	=	power-law photon index.
`par2`	=	Hydrogen column in units of 10^{22} cm^{-2}.

`par3`	=	Absorber temperature in K.
`par4`	=	Absorber ionization state (L/nR^2).
`par5`	=	Redshift.
`par6`	=	Iron abundance relative to Solar.

6.4 ascac

Mixing model for ASCA data. Written for cluster data so uses beta or two power-law surface brightness models. Includes a calculation of the telescope effective area so no arf should be applied to input files. Note that this model is very slow if any of the parameters are free.

The model is used by reading spectra in as separate datagroups. Each input file requires the XFLT0001 keyword set to a different number (eg if concentric annuli are in use then number outwards). The normalizations for each datagroup should be linked since the ascac model takes care of the relative normalizations based on the surface brightness model used.

`par1`	=	Alpha.
`par2`	=	Beta.
`par3`	=	Core (arcmin).
`par4`	=	Switch (0 $\Rightarrow$ beta model, 1 $\Rightarrow$ 2-power-law).

6.5 atable

An additive table model. The filename to be used should be given immediately after `atable` in the `model` command. For example,

```
XSPEC> model atable mymod.mod
```

uses `mymod.mod` as the input for the model.

6.6 bbody

A blackbody spectrum.

$$A(E) = \texttt{K}\ 8.0525 E^2 dE / (\texttt{par1}^4 (\exp(E/kt) - 1))$$

where :

par1	=	temperature kT in keV
K	=	$L_{39}/\ D_{10}^2$, where L_{39} is the source luminosity in units of 10^{39} ergs/sec and D_{10} is the distance to the source in units of 10 kpc

6.7 bbodyrad

A blackbody spectrum with normalization proportional to the surface area.

$$A(E) = \texttt{K}\ 1.0344 \times 10^{-3} E^2 dE / (\exp(E/\texttt{par1}) - 1)$$

where :

par1	=	temperature kT in keV
K	=	R_{km}^2/D_{10}^2, where R_{km} is the source radius in km and D_{10} is the distance to the source in units of 10 kpc

6.8 bknpower

A broken power law.

$$\begin{aligned} A(E) &= \texttt{K}\ (E/1\text{keV})^{-\texttt{par1}} && \text{for } E \le \texttt{par2} \\ &= \texttt{K}\ \texttt{par2}^{(\texttt{par3}-\texttt{par1})} \times (E/1\text{keV})^{-\texttt{par3}} && \text{for } E \ge \texttt{par2} \end{aligned}$$

where :

par1	=	power law photon index for E < break energy
par2	=	break point for the energy in keV
par3	=	power law photon index for E > break energy
K	=	photons/keV/cm^2/s at 1 keV

6.9 bremss

A thermal bremsstrahlung spectrum based on the Kellogg, Baldwin & Koch (ApJ 199, 299) polynomial fits to the Karzas & Latter numerical values. A routine from Kurucz is used in cases of high temperature. The He abundance is assumed to be 8.5

`par1`	=	plasma temperature in keV
`K`	=	$(3.02 \times 10^{-15}/(4\pi D^2)) \int n_e n_I dV$, where D is the distance to the source (cm) and n_e, n_I are the electron and ion densities (cm^{-3})

6.10 cabs

Non-relativistic Compton scattering.

$$M(E) = \exp(-\mathtt{par1}\sigma(E))$$

where $\sigma(E)$ is the Thomson cross-section.

`par1` = hydrogen column (in units of 10^{22} atoms/cm^2)

6.11 cemekl

A multi-temperature plasma emission model built from the mekal code. Emission measures follow a power-law in temperature (ie emission measure from temperature T is proportional to $(T/\mathtt{par2})^{\mathtt{par1}}$. The abundance ratios are set by the abund command.

`par1`	=	index for power-law emissivity function
`par2`	=	maximum temperature
`par3`	=	nH (cm^{-3})
`par4`	=	Abundance relative Solar
`par5`	=	Redshift
`K`	=	Normalization

6.12 cevmkl

A multi-temperature plasma emission model built from the mekal code. Emission measures follow a power-law in temperature (ie emission measure from temperature T is proportional to $(T/\mathtt{par2})^{\mathtt{par1}}$. The abundances are relative to the Solar abundances set by the abund command.

`par1`	=	index for power-law emissivity function
`par2`	=	maximum temperature
`par3`	=	nH ($cm^{-}3$)
`par4-par17`	=	Abundances for He, C, N, O, Ne, Na, Mg, Al, Si, S, Ar, Ca, Fe, Ni wrt Solar (defined by the abund command)
`par18`	=	Redshift
`K`	=	Normalization

6.13 cflow

A cooling flow model after Mushotzky & Szymkowiak ("Cooling Flows in Clusters and Galaxies," ed. Fabian, 1988). An index of zero for the power-law emissivity function corresponds to emission measure weighted by the inverse of the cooling time at that temperature. The model assumes $H_0 = 50$ and $q_0 = 0$. The abundance ratios are set by the abund command.

`par1`	=	index for power-law emissivity function
`par2`	=	low temperature (keV)
`par3`	=	high temperature (keV)
`par4`	=	abundance relative Solar
`par5`	=	redshift
`K`	=	Mass accretion rate (solar mass/yr)

6.14 compls

A Comptonization spectrum after Lamb and Sanford, 1979, *M.N.R.A.S*, 288, 555. This model calculates the self-Comptonization of a bremsstrahlung emission from an optically-thick spherical plasma cloud with a given optical depth and temperature. It was popular for Sco X-1.

`par1`	=	temperature in keV
`par2`	=	optical depth

`K`	=	?

6.15 compst

A Comptonization spectrum after Sunyaev and Titarchuk 1980, *A&A*, 86, 121. This model is the Comptonization of cool photons on hot electrons.

`par1`	=	temperature in keV
`par2`	=	optical depth
`K`	=	?

6.16 comptt

This is an analytic model describing Comptonization of soft photons in a hot plasma, developed by L. Titarchuk (see ApJ, 434, 313). This replaces the Sunyaev-Titarchuk Comptonization model in the sense that the theory is extended to include relativistic effects. Also, the approximations used in the model work well for both the optically thin and thick regimes. The Comptonized spectrum is determined completely by the plasma temperature and the so-called β parameter which is independent of geometry. The optical depth is then determined as a function of β for a given geometry. Thus `par5` switches between spherical and disk geometries so that β is not a direct input here. This parameter MUST be frozen. If `par5` > 0, β is obtained from the optical depth using analytic approximation (e.g. Titarchuk 1994). If `par5` < 0 and $0.1 < \tau < 10$, β is obtained by interpolation from a set of accurately calculated pairs of β and τ from Sunyaev & Titarchuk 1985 (A&A 143, 374). In this incarnation of the model, the soft photon input spectrum is a Wien law [x**2 exp(-x) photons] because this lens itself to a particularly simple analytical form of the model. For present X-ray detectors this should be adequate. Note that in energy flux space the peak of the Wien law occurs at 3kT as opposed to 2.8kT for a blackbody. The spectral fitter is strongly urged to read the following references before using this model in order to fully understand and appreciate the phyiscal assumptions made: 1. Titarchuk, L., 1994, ApJ, 434, 313 2. Hua, X-M., Titarchuk, L., 1994, ApJ, xxx, xxx 3. Titarchuk, L., Lyubarskij, Y., 1995, ApJ, submitted.

`par1`	=	Redshift.
`par2`	=	Input soft photon (Wien) temperature (keV).
`par3`	=	Plasma temperature (keV).
`par4`	=	Plasma optical depth.
`par5`	=	Geometry switch. ABS(`par5`) $<= 1$: disk, > 1: sphere.
	=	`par5` $>= 0$: use analytic approx for β vs. τ.

	=	`par5` < 0 : Get beta vs. tau from interpolation.
`K`	=	?

6.17 constant

An energy-independent multiplicative factor.

`par1`	=	factor

6.18 cp6mkl

A multi-temperature MEKAL model using sixth-order Chebyshev polynomial for the differential emission measure (eg Lemen etal. ApJ 341, 474, 1989). The abundance is relative to the Solar abundances set by the abund command.

`par1`	=	Chebyshev polynomial coefficient 1
`par2`	=	Chebyshev polynomial coefficient 2
`par3`	=	Chebyshev polynomial coefficient 3
`par4`	=	Chebyshev polynomial coefficient 4
`par5`	=	Chebyshev polynomial coefficient 5
`par6`	=	Chebyshev polynomial coefficient 6
`par7`	=	H density (cm$^-$3)
`par8`	=	abundance wrt to Solar
`par9`	=	Redshift
`K`	=	$(10^{-14}/(4\pi D^2)\int n_e n_H dV$, where D is the luminosity
	=	distance to the source (cm), n_e is the electron density (cm$^-$3),
	=	and n_H is the hydrogen density (cm$^-$3)

6.19 cp6vmkl

A multi-temperature MEKAL model using sixth-order Chebyshev polynomial for the differential emission measure (eg Lemen etal. ApJ 341, 474, 1989). The abundance is relative to the Solar abundances set by the abund command.

`par1`	=	Chebyshev polynomial coefficient 1
`par2`	=	Chebyshev polynomial coefficient 2
`par3`	=	Chebyshev polynomial coefficient 3

par4	=	Chebyshev polynomial coefficient 4
par5	=	Chebyshev polynomial coefficient 5
par6	=	Chebyshev polynomial coefficient 6
par7	=	H density (cm$^-$3)
par8-par21	=	Abundances of He, C, N, O, Ne, Na, Mg, Al, Si, S,
	=	Ar, Ca, Fe, Ni wrt Solar (defined by the abund command)
par22	=	Redshift
K	=	$(10^{-14}/(4\pi D^2)\int n_e n_H dV$, where D is the luminosity
	=	distance to the source (cm), n_e is the electron density (cm$^-$3),
	=	and n_H is the hydrogen density (cm$^-$3)

6.20 cutoffpl

A power law with high energy exponential cutoff.

$$A(E) = \mathtt{K}(E/1\mathrm{keV})^{-\mathtt{par1}}\exp(-E/\mathtt{par2})$$

where :

par1	=	power law photon index
par2	=	cutoff energy of exponential cutoff (in keV)
K	=	photons/keV/cm^2/s at 1 keV

6.21 cyclabs

A cyclotron absorption line as used in pulsar spectra. See Mihara et al., *Nature*, 1990 or Makishima et al. *PASJ*, 1990.

$$M(E) = \exp(-\mathtt{par1}(\mathtt{par3}E/\mathtt{par2})^2/((E-\mathtt{par2})^2+\mathtt{par3}^2)+\mathtt{par4}(\mathtt{par5}E/(2\mathtt{par2}))^2/((E-2\mathtt{par2})^2+\mathtt{par5}^2)))$$

par1	=	depth of the fundamental
par2	=	cyclotron energy
par3	=	width of the fundamental
par4	=	depth 2nd harmonic
par5	=	width of the 2nd harmonic

6.22 disk

The spectrum from an accretion disk, where the opacities are dominated by free-free absorption, i.e., the so-called blackbody disk model. Not correct for a disk around a neutron star.

`par1`	=	accretion rate in Eddington Luminosities
`par2`	=	central mass in solar mass units
`par3`	=	inner disk radius in gravitational (= 3 Schwarzschild) radii
`par4`	=	distance in kpc
`K`	=	$\cos(i)/d^2$, where i is the inclination of the disk and d is the distance in units of 10 kpc

6.23 diskbb

The spectrum from an accretion disk consisting of multiple blackbody components. For example, see Mitsuda et al., *PASJ*, 36, 741, (1984), Makishima et al., *ApJ* 308, 635, (1986).

`par1`	=	temperature at inner disk radius (keV)
`K`	=	$((\mathrm{Rin/km})/(\mathrm{D/10kpc}))^2 \cos\theta$, where Rin is the inner disk radius, D the distance to the source, and θ the angle of the disk

6.24 diskline

A line emission from a relativistic accretion disk. See Fabian et al., *MNRAS* 238, 729.

`par1`	=	line energy
`par2`	=	power law depend. of emissivity (10 for disk)
`par3`	=	inner radius (units of GM/c^2)
`par4`	=	outer radius (units of GM/c^2)
`par5`	=	inclination (degrees)
`K`	=	$\mathrm{photons/cm^2/s}$ in the spectrum

6.25 diskm

A disk model with gas pressure viscosity. The spectrum from an accretion disk where the viscosity scales as the gas pressure. From Stella and Rosner 1984, *ApJ*, 277, 312.

`par1`	=	accretion rate in Eddington luminosities
`par2`	=	central mass in units of solar mass
`par3`	=	inner disk radius in gravitational (= 3 Schwarzschild) radii
`par4`	=	viscosity
`K`	=	$\cos(i)/d^2$, where i is the inclination of the disk and d is the distance in units of 10 kpc

6.26 disko

A modified blackbody disk model. The spectrum from the inner region of an accretion disk where the viscosity is dominated by radiation pressure.

`par1`	=	accretion rate in Eddington luminosities
`par2`	=	central mass in units of solar mass
`par3`	=	inner disk radius in gravitational (= 3 Schwarzschild) radii
`par4`	=	viscosity
`K`	=	$\cos(i)/d^2$, where i is the inclination of the disk and d is the distance in units of 10 kpc

6.27 dust

A modification of a spectrum due to scattering-off dust on the line-of-sight. The model assumes that the scattered flux goes into a uniform disk whose size has a 1/E dependence and whose total flux has a $1/E^2$ dependence.

`par1`	=	scattering fraction at 1 keV
`par2`	=	size of halo at 1 keV in units of the detector beamsize

6.28 edge

An absorption edge.

$$\begin{aligned} M(E) &= 1 && \text{for } E \leq \texttt{par1} \\ &= \exp(-\texttt{par2}((E/\texttt{par1})^3)) && \text{for } E \geq \texttt{par1} \end{aligned}$$

where :

par1	=	threshold energy
par2	=	absorption depth at the threshold

6.29 etable

An exponential table model. The filename to be used should be given immediately after etable in the model command. For example,

```
XSPEC> model etable mymod.mod
```

uses mymod.mod as the input for the model. XSPEC will multiply the contents of the model by -1 then take the exponential ie this model is for calculating absorption functions. For specifications of the table model file, see the OGIP memo 92-009 on the FITS file format for table model files (available on the WWW or by anonymous ftp from legacy.gsfc.nasa.gov under caldb/docs/memos).

6.30 expabs

A low-energy exponential cutoff.

$$M(E) = \exp(-\texttt{par1}/E)$$

where :

par1	=	e-folding energy for the absorption

6.31 expfac

An exponential modification of a spectrum.

$$\begin{aligned} M(E) &= 1. - \texttt{par1}\exp(-\texttt{par2}(E/1\mathrm{keV})) && \text{for } E > \texttt{par3} \\ M(E) &= 1. && \text{for } E < \texttt{par3} \end{aligned}$$

par1	=	amplitude of effect
par2	=	exponential factor
par3	=	start energy of modification

6.32 gaussian

A simple gaussian line profile. If the width is ≤ 0, then it is treated as a delta function.

$$A(E) = \texttt{K}\ (1./\texttt{par2}\sqrt{(2\pi)})\exp(-0.5((E-\texttt{par1})/\texttt{par2})^2)$$

where :

par1	=	line energy in keV
par2	=	line width (σ) in keV
K	=	total photons/cm^2/s in the line

6.33 highecut

A high energy cutoff.

$$\begin{aligned} A(E) &= \exp((\texttt{par1} - E)/\texttt{par2}) && \text{for } E \geq \texttt{par1} \\ A(E) &= 1 && \text{for } E \leq \texttt{par1} \end{aligned}$$

where :

par1	=	cutoff energy in keV
par2	=	e-folding energy in keV

6.34 hrefl

A simple multiplicative reflection model due to Tahir Yaqoob. Parameters are as follows:

`par1`	=	minimum angle (degrees) between source photons incident on the slab and the slab normal (=arctan(Ri/H).
`par2`	=	maximum angle (degrees) between source photons incident on the slab and the slab normal (=arctan(Ro/H).
`par3`	=	Angle (degrees) between the observer's line of sight and the slab normal.
`par4`	=	Iron abundance relative to Solar.
`par5`	=	Iron K-edge energy.
`par6`	=	Fraction of the direct flux seen by the observer.
`par7`	=	Normalization of the reflected continuum.
`par8`	=	redshift.

This model gives the reflected X-ray spectrum from a cold, optically thick, circular slab with inner and outer radii (Ri & Ro, respectively) illuminated by a point source a height H above the centre of the slab. The main difference between this model and other reflection models is that analytic approximations are used for the Chandrasekar H functions (and their integrals) and ELASTIC SCATTERING is assumed (see Basko 1978, *ApJ*, 223, 268). The elastic-scattering approximation means that the model is ONLY VALID UP TO 15 keV in the source frame. Future enhancements will include fudge factors that will allow extension up to 100 keV. The fact that no integration is involved at any point makes the routine very fast and particularly suitable for generating error contours, especially when fitting a large number of data channels. The model is multiplicative, and so can be used with ANY incident continuum.

Suppose the incident photon spectrum is N(E) photons/cm/cm/s/keV and that the incident continuum is steady in time, and suppose further that the reflected continuum from the slab is R(E). When you multiply the incident spectrum with HREFL, what you actually get is the following:

$$model(E) = \mathtt{par6}N(E) + \mathtt{par7}R(E)$$

Thus, the actual physical situation described above corresponds to `par6`=1.0 and `par7`=1.0. You may decide to float `par6` and/or `par7`. In that case, you must decide what the best-fitting values of these parameters mean physically for your case. It may imply time-lags between the direct and reflected components, different source and/or disk geometries to those assumed, or something else.

6.35 meka

An emission spectrum from hot diffuse gas based on the model calculations of Mewe and Gronenschild (as amended by Kaastra). The model includes line emissions from several elements.

`par1`	=	plasma temperature in keV
`par2`	=	hydrogen density in cm^{-3}
`par3`	=	Metal abundances (He fixed at cosmic). The elements included are C, N, O, Ne, Na, Mg, Al, Si, S, Ar, Ca, Fe, Ni. Abundances are set by the abund command.
`par4`	=	redshift, z
`K`	=	$(10^{-14}/(4\pi D^2))\int n_e n_H dV$, where D is the distance to the source (cm), n_e and n_H are the electron and H densities (cm^{-3})

The references for the MEKA model are as follows :

- Mewe, R., Gronenschild, E.H.B.M., and van den Oord, G.H.J. 1985, *A&AS*, 62, 197
- Mewe, R., Lemen, J.R., and van den Oord, G.H.J. 1986, *A&AS*, 65, 511
- Kaastra, J.S. 1992, An X-Ray Spectral Code for Optically Thin Plasmas (Internal SRON-Leiden Report, updated version 2.0)

Similar credit may also be given for the adopted ionization balance e.g.:

- Arnaud, M., and Rothenflug, M. 1985, *A&AS*, 60, 425

6.36 mekal

An emission spectrum from hot diffuse gas based on the model calculations of Mewe and Kaastra with Fe L calculations by Liedahl. The model includes line emissions from several elements.

`par1`	=	plasma temperature in keV
`par2`	=	hydrogen density in cm^{-3}
`par3`	=	Metal abundances (He fixed at cosmic). The elements included are C, N, O, Ne, Na, Mg, Al, Si, S, Ar, Ca, Fe, Ni. Abundances are set by the abund command.
`par4`	=	redshift, z

`K`	=	$(10^{-14}/(4\pi D^2)) \int n_e n_H dV$, where D is the distance to the source (cm), n_e and n_H are the electron and H densities (cm^{-3})

The references for the MEKAL model are as follows :

- Mewe, R., Gronenschild, E.H.B.M., and van den Oord, G.H.J. 1985, *A&AS*, 62, 197
- Mewe, R., Lemen, J.R., and van den Oord, G.H.J. 1986, *A&AS*, 65, 511
- Kaastra, J.S. 1992, An X-Ray Spectral Code for Optically Thin Plasmas (Internal SRON-Leiden Report, updated version 2.0)

Similar credit may also be given for the adopted ionization balance e.g.:

- Arnaud, M., and Rothenflug, M. 1985, *A&AS*, 60, 425

6.37 mkcflow

A cooling flow model after Mushotzky & Szymkowiak ("Cooling Flows in Clusters and Galaxies" ed. Fabian, 1988). This one uses the MEKA model for the individual temperature components and differs from `cflow` in setting the emissivity function to be the inverse of the cooling time. The model assumes H0=50 and q0=0. Abundance ratios are set by the `abund` command.

`par1`	=	low temperature (keV)
`par2`	=	high temperature (keV)
`par3`	=	abundance relative to Solar
`par4`	=	redshift
`K`	=	Mass accretion rate (solar mass/yr)

6.38 mtable

A multiplicative table model. The filename to be used should be given immediately after `mtable` in the `model` command. For example,

```
XSPEC> model mtable mymod.mod ...
```

uses `mymod.mod` as the input for the model.

6.39 notch

A notch line absorption. This is model is equivalent to a very saturated absorption line.

$$\begin{aligned} M(E) &= (1 - \mathtt{par3}) \quad \text{for } \mathtt{par1} - \mathtt{par2}/2 < E < \mathtt{par1} + \mathtt{par2}/2 \\ &= 1 \quad\quad\quad\quad\quad\; \text{for all other } E \end{aligned}$$

where :

`par1`	=	line energy (keV)
`par2`	=	line width (keV)
`par3`	=	covering fraction

6.40 nteea

A pair plasma model from Magdziarz and Zdziarski. Consult Andrzej Zdziarski for details.

`par1`	=	nonthermal electron compactness
`par2`	=	blackbody compactness
`par3`	=	scaling factor for reflection (1 for isotropic source above disk)
`par4`	=	blackbody temperature in eV
`par5`	=	the maximum Lorentz factor
`par6`	=	thermal compactness (0 for pure nonthermal plasma)
`par7`	=	Thomson optical depth of ionization electrons (e.g., 0)
`par8`	=	electron injection index (0 for monoenergetic injection)
`par9`	=	minimum Lorentz factor of the power law injection (not used for mono)
`par10`	=	minimum Lorentz factor for nonthermal reprocessing (> 1; <= `par9`)
`par11`	=	radius in cm (for Coulomb/bremsstrahlung only)
`par12`	=	pair escape rate (0-1)
`par13`	=	cosine of inclination angle
`par14`	=	iron abundance relative to the solar iron abundance (7.52)
`K`	=	?

6.41 pcfabs

A partial covering fraction absorption. The relative abundances are set by the `abund` command.

$$M(E) = \mathtt{par2}\exp(-\mathtt{par1}\sigma(E)) + (1 - \mathtt{par2})$$

where σ(E) is the photo-electric cross-section (NOT including Thomson scattering) (see `phabs`) and:

`par1`	=	equivalent hydrogen column (in units of 10^{22} atoms/cm^2)
`par2`	=	covering fraction ($0 <$ par2 $<= 1.$) (dimensionless)

6.42 pegpwrlw

A power law with pegged normalization.

$$A(E) = \mathtt{K}(E/1\mathrm{keV})^{-\mathtt{par1}}$$

where :

`par1`	=	photon index of power law (dimensionless)
`par2`	=	lower peg energy range
`par3`	=	upper peg energy range
`K`	=	flux (in units of 10^{-12} erg/cm^2/s) over the energy range `par2`–`par3` unless `par2` = `par3`, in which case it is the flux (in micro-Jy) at `par2`

6.43 pexrav

Exponentially cut-off power-law spectrum reflected from neutral material. See Magdziarz & Zdziarski (MNRAS 1995) for details. The output spectrum is the sum of the cutoff power law and the reflection component. The reflection component alone can be obtained for scale (see below) = rel_refl < 0. Then the actual reflection normalization is —scale—. Note that you need to change then the limits of rel_refl. The range of rel_refl in that case should exclude zero (as then the direct component appears). If $E_c = 0$, there is no cutoff in the power law. Version with variable iron abundance and new opacities of Balucinska & McCammon (1992, and 1994, private communication). As expected in AGNs, H and He are assumed to be fully ionized.

par1 = γ, power law photon index, N_E prop. to $E^{-\gamma}$
par2 = E_c, the cutoff energy in keV (if $E_c = 0$ there is no cutoff; one needs to change the lower limit for that)
par3 = scale, scaling factor for reflection; if < 0, no direct component (scale=1 for isotropic source above disk)
par4 = redshift, z
par5 = iron abundance relative to the solar iron abundance (7.52)
par6 = cosine of inclination angle
K = photon flux at 1 keV of the power-law only in the observed frame.

6.44 pexriv

Exponentially cut-off power-law spectrum reflected from neutral material. See Magdziarz & Zdziarski (MNRAS 1995) for details. The output spectrum is the sum of the cutoff power law and the reflection component. The reflection component alone can be obtained for scale (see below) = rel_refl < 0. Then the actual reflection normalization is —scale—. Note that you need to change then the limits of rel_refl. The range of rel_refl in that case should exclude zero (as then the direct component appears). If $E_c = 0$, there is no cutoff in the power law. Version with variable iron abundance and new opacities of Balucinska & McCammon (1992, and 1994, private communication).

par1 = γ, power law photon index, N_E prop. to $E^{-\gamma}$
par2 = E_c, the cutoff energy in keV (if $E_c = 0$ there is no cutoff; one needs to change the lower limit for that)
par3 = scale, scaling factor for reflection; if < 0, no direct component (scale=1 for isotropic source above disk)
par4 = redshift, z
par5 = iron abundance relative to the solar iron abundance (7.52)
par6 = cosine of inclination angle
par7 = disk temperature in K
par8 = disk ionization parameter (L/nR^2)
K = photon flux at 1 keV of the power-law only in the observed frame.

6.45 phabs

A photoelectric absorption using Balucinska-Church and McCammon cross-sections. The relative abundances are set by the abund command.

$$A(E) = \exp(-\mathtt{par1}\sigma(E))$$

where $\sigma(E)$ is the photo-electric cross-section (NOT including Thomson scattering) and

par1	=	equivalent hydrogen column (in units of 10^{22} atoms/cm^2)

6.46 plabs

Absorption as a power-law in energy. Useful for things like dust.

$$M(E) = \mathtt{par2}(E/1\mathrm{keV})^{-\mathtt{par1}}$$

par1	=	index
par2	=	coefficient

6.47 pliref

The total spectrum resulting from a power-law illumination of material of an arbitrary ionisation state, i.e., the power law together with its reflection spectrum, calculated self-consistently for ionised material. The reflected spectrum is calculated following Lightman and White (1988), *Ap.J.*, 335, 57, but with opacities calculated by solving the Saha equation, balancing photo-ionisation with recombination. The amount of photo-ionisation is determined from the ionisation parameter, ξ=L/(nR^2), where L is the total photo-ionising luminosity of the source, assumed to be the power law integrated between 5eV and 300 keV; n is the density of the illuminated material; and R is its distance from the X-ray source. Note that this definition of ξ differs in bandpass from the 13.6eV - 13.6 keV used by Kallman and McCray (1982), *Ap. J. Supp.*, 50, 263, so that their ξ is around an order of magnitude less than that used here. The recombination rate is dependent on the temperature of the material: the approximations used here are generally valid between 1e3 and 1e6 K. Photo-ionisation cross-sections are taken from Reilman and Manson (1979), Ap. J. Supp., 40, 815, recombination coefficients from Shull and Van Steenburg (1982), *Ap. J. Supp.*, 48, 95, and abundances from Lang (1974), *Astrophysical Formulae.*

The reflected spectrum is assumed to scale as the cosine of the inclination angle. This is not a bad approximation at small angles, but is increasingly inaccurate at large inclinations, especially as doppler effects are not included in this code. See George and Fabian (1991), *M.N.R.A.S.*, 249, 352; and Matt, Perola and Piro (1991), *Astron. Astrophys.*, 247, 25 for more accurate calculations of inclination effects. As Compton down-scattering is important, the X-ray spectrum beyond the 2-20 keV band can affect the calculated shape of the reflection spectrum. The recommended maximum energy cut-off is 0.6 (=

300 keV). See Done et al., (1992) *Ap. J.*, 395, 275 for more details and an example of use.

par1	=	photon index of the intrinsic, illuminating power law
par2	=	fraction of power law escaping towards observer. Setting this to unity is equivalent to assuming an isotropic X-ray source.
par3	=	inclination of the disk to the observer (degrees)
par4	=	solid angle subtended by the disk/2π. Setting this to unity gives the spectrum as calculated in Lightman and White (1988).
par5	=	temperature of illuminated material (kelvin)
par6	=	ionisation state of illuminated material, ξ=L/(nR^2)
par7	=	maximum energy extent of the power law in units of m_ec^2
K	=	photons/keV/cm^2/s at 1 keV

6.48 plrefl

The total spectrum resulting from a power-law illumination of cold material, i.e., the power law together with its reflection spectrum. The reflected spectrum is calculated following Lightman and White (1988), *Ap.J.*, 335, 57, but with opacities calculated for cold material from Morrison and McCammon (1983), *Ap. J.*, 270, 119. The reflected spectrum is assumed to scale as the cosine of the inclination angle. This is not a bad approximation at small angles but is increasingly inaccurate at large inclinations, especially as doppler effects are not included in this code. See George and Fabian (1991), *M.N.R.A.S.*, 249, 352; and Matt, Perola and Piro (1991), *Astron. Astrophys.*, 247, 25 for more accurate calculations of inclination effects. As Compton down-scattering is important, the X-ray spectrum beyond the 2-20 keV band can affect the calculated shape of the reflection spectrum. The recommended maximum energy cut-off is 0.6 (= 300 keV).

par1	=	photon index of the intrinsic, illuminating power law
par2	=	fraction of power law escaping towards observer. Setting this to unity is equivalent to assuming an isotropic X-ray source.
par3	=	inclination of the disk to the observer (degrees)
par4	=	solid angle subtended by the disk/2π. Setting this to unity gives the spectrum as calculated in Lightman and White (1988).
par5	=	maximum energy extent of the power law in units of m_ec^2
K	=	photons/keV/cm^2/s at 1 keV

6.49 powerlaw

Simple photon power law.

$$A(E) = \mathtt{K}(E/1\mathrm{keV})^{-\mathtt{par1}}$$

where :

`par1`	=	photon index of power law (dimensionless)
`K`	=	photons/keV/cm^2/s at 1 keV.

6.50 raymond

An emission spectrum from hot, diffuse gas based on the model calculations of Raymond and Smith including line emissions from several elements. This model interpolates on a grid of spectra for different temperatures. The grid is logarithmically spaced with 80 temperatures ranging from 0.008 to 80 keV.

`par1`	=	plasma temperature in keV
`par2`	=	Metal abundances (He fixed at cosmic) The elements included are C, N, O, Ne, Mg, Si, S, Ar, Ca, Fe, Ni and their relative abundances are set by the `abund` command.
`par3`	=	redshift, z
`K`	=	$(10^{-14}/(4\pi D^2))\int n^2 dV$, where D is the distance to the source (cm) and n is the electron density (cm^{-3})

6.51 redge

Recombination edge emission.

$$\begin{aligned} A(E) &= 0. && \text{for } E < \mathtt{par1} \\ A(E) &= K(1/\mathtt{par2})\exp(-(E-\mathtt{par1})/\mathtt{par2}) && \text{for } E > \mathtt{par1} \end{aligned}$$

`par1`	=	threshold energy
`par2`	=	plasma temperature (keV)
`K`	=	total photons/cm^2/s in the line

6.52 smedge

A smeared edge (Ebisawa PhD thesis, implemented by Frank Marshall).

$$
\begin{aligned}
M(E) &= 1. && \text{for } E \leq \mathtt{par1} \\
M(E) &= \exp(-\mathtt{par2}((E/\mathtt{par1})^{\mathtt{par3}})(1-\exp((\mathtt{par1}-E))/\mathtt{par4})) && \text{for } E \geq \mathtt{par1}
\end{aligned}
$$

par1	=	the threshold energy (keV)
par2	=	the maximum absorption factor at threshold
par3	=	index for photo-electric cross-section (normally -2.67)
par4	=	smearing width (keV)

6.53 spline

A cubic spline modification.

par1	=	start x-value
par2	=	start y-value
par3	=	end y-value
par4	=	start dy/dx
par5	=	end dy/dx
par6	=	end x-value

6.54 SSS ice

The Einstein Observatory SSS ice absorption.

par1	=	ice thickness parameter.

6.55 step

A step function convolved with a gaussian.

$$N(E) = \mathtt{K}(1 - \mathrm{erf}((E - \mathtt{par1})/\sqrt{(}2)/\mathtt{par2}))/2$$

par1	=	start energy (keV)
par2	=	gaussian sigma (keV)
K	=	step amplitude

6.56 tita_a

Reflection of a comptonized spectrum from Magdziarz & Zdziarski (MNRAS 1995). The output spectrum is the sum of the comptonization spectrum and the reflection component. The reflection component alone can be obtained for scale (see below) = rel_refl < 0. Then the actual reflection normalization is —scale—. Note that you need to change then the limits of rel_refl. The range of rel_refl in that case should exclude zero (as then the direct component appears). Version with variable iron abundance and new opacities of Balucinska & McCammon (1992, and 1994, private communication). As expected in AGNs, H and He are assumed to be fully ionized.

`par1`	=	Temperature in keV
`par2`	=	photon spectral index
`par3`	=	scale, scaling factor for reflection; if < 0, no direct component
	=	(scale=1 for isotropic source above disk)
`par4`	=	redshift, z
`par5`	=	iron abundance relative to the solar iron abundance (7.52)
`par6`	=	cosine of inclination angle
`K`	=	The photon flux at 1 keV (photons/keV/cm^2/s) of the
	=	incident spectrum only (without reflection) and in the earth frame.

6.57 tsabs

A Thomson scattering.

$$A(E) = \exp(-\sigma(E)\texttt{par1})$$

where σ(E) is the cross-section for Thomson scattering

`par1`	=	electron column density (10^{22} cm^{-2})

6.58 useradd1

The user-defined additive model (VMS-specific). The user can have his or her own model with up to 20 parameters + norm, but it must be linked with the `MKUMD` command, and must be defined with the `SETUMD` command. A set of user models have been pre-linked. These “official” user models can be accessed using the `SETUMD` command before running XSPEC.

6.59 usermul1

A user-defined multiplicative model (VMS-specific). The user can have his or her own model with up to 20 parameters. (The model must be linked with the `MKUMD` command and defined with the `SETUMD` command.)

6.60 uvred

A UV reddening using Seaton's law (*M.N.R.A.S.*, 187, 75p). Valid from 1000-3704 angstroms.

`par1`	=	E(B-V)

6.61 varabs

A photoelectric absorption with variable abundances using Balucinska-Church and McCammon cross-sections. The column for each element is in units of the column in a solar abundance column of an equivalent hydrogen column density of 10^{22} cm^{-2}. The Solar abundance table used is set by the abund command.

`par1-par18`	=	equivalent columns for H, He, C, N, O, Ne, Na, Mg, Al, Si, S, Cl, Ar, Ca, Cr, Fe, Ni, Co

6.62 vbremss

A thermal bremsstrahlung spectrum with variable He/H. Based on the Kellogg, Baldwin & Koch polynomial fits to the Karzas & Latter numerical values. A routine from Kurucz is used in the case of high temperatures.

`par1`	=	plasma temperature (keV)
`par2`	=	n(He)/n(H) in units of Solar ratio
`K`	=	$(3.02 \times 10^{-15}/(4\pi D^2)) \int n_e n_I dV$, where D is the distance to the source (cm) and n_e, n_I are the electron and ion densities (cm^{-3})

6.63 vmeka

An emission spectrum from hot diffuse gas based on the model calculations of Mewe and Gronenschild (as amended by Kaastra). The model includes line emissions from several elements. Abundances are the number of nuclei per Hydrogen nucleus relative to the Solar abundances set by the `abund` command.

`par1`	=	plasma temperature in keV
`par2`	=	hydrogen density in cm^{-3}
`par3-par16`	=	Abundances for He, C, N, O, Ne, Na, Mg, Al, Si, S,
	=	Ar, Ca, Fe, Ni wrt Solar (defined by the abund command)
`par17`	=	redshift, z
`K`	=	$(10^{-14}/(4\pi D^2)) \int n_e n_h dV$, where D is the distance to the source (cm), n_e and n_H are the electron and H densities (cm^{-3})

6.64 vmekal

An emission spectrum from hot diffuse gas based on the model calculations of Mewe and Kaastra with Fe L calculations by Liedahl. The model includes line emissions from several elements. Abundances are the number of nuclei per Hydrogen nucleus relative to the Solar abundances set by the abund command.

`par1`	=	plasma temperature in keV
`par2`	=	hydrogen density in cm**-3
`par3-par16`	=	Abundances for He, C, N, O, Ne, Na, Mg, Al, Si, S,
	=	Ar, Ca, Fe, Ni wrt Solar (defined by the abund command)
`par17`	=	redshift, z
`K`	=	$(10^{-14}/(4\pi D^2)) \int n_e n_h dV$, where D is the distance to the source (cm), n_e and n_H are the electron and H densities (cm^{-3})

6.65 vmcflow

A cooling flow model after Mushotzky & Szymkowiak ("Cooling Flows in Clusters and Galaxies" ed. Fabian, 1988). This one uses the `vmeka` model for the individual temperature components, but is otherwise identical to `mkcflow`. Abundances are relative to Solar as set by the `abund` command.

`par1`	=	low temperature (keV)
`par2`	=	high temperature (keV)

`par3-par16`	=	Abundances for He, C, N, O, Ne, Na, Mg, Al, Si, S,
	=	Ar, Ca, Fe, Ni wrt Solar (defined by the abund command)
`par17`	=	redshift
`K`	=	Mass accretion rate (solar mass/yr)

6.66 vraymond

A version of the XSPEC model `raymond`, with independent variables for all abundances and for the continuum. Abundances are the number of nuclei per Hydrogen nucleus relative to the Solar abundances as set by the `abund` command.

`par1`	=	plasma temperature (keV)
`par2-par13`	=	Abundances for He, C, N, O, Ne, Mg, Si, S,
	=	Ar, Ca, Fe, Ni wrt Solar (defined by the abund command)
`par14`	=	redshift z
`K`	=	$(10^{-14}/(4\pi D^2))\int n_e n_h dV$, where D is the distance to the source (cm), n_e and n_H are the electron and H densities (cm^{-3})

6.67 wabs

A photo-electric absorption using Wisconsin (Morrison and McCammon; ApJ 270, 119) cross-sections.

$$M(E) = \exp(-\mathtt{par1}\sigma(E))$$

where σ(E) is the photo-electric cross-section (NOT including Thomson scattering) and

`par1`	=	equivalent hydrogen column (in units of 10^{22} atoms/cm^2)

6.68 wndabs

A photo-electric absorption from approximation to a warm absorber using Baluncinska-Church and McCammon cross-sections. Relative abundances are set by the abund command.

$$M(E) = \exp(-\mathtt{par1}\sigma(E)) \quad \text{for } E \geq \mathtt{par2}$$
$$= 1. \quad \text{for } E \leq \mathtt{par2}$$

where σ(E) is the photo-electric cross-section (NOT including Thomson scattering) and

`par1`	=	equivalent hydrogen column (in units of 10^{22} atoms/cm^2)
`par2`	=	window energy (keV)

6.69 zbbody

A redshifted blackbody spectrum.

$$A(E) = \mathtt{K}\ 8.0525(E(1+\mathtt{par2}))^2 dE/((kT)^4(\exp(E(1+\mathtt{par2})/\mathtt{par1})-1))/(1+\mathtt{par2})$$

where :

`par1`	=	temperature kT in keV
`par2`	=	redshift
`K`	=	L39/(D10)^2, where L39 is the source luminosity in units of 10^{39} ergs/sec and D10 is the distance to the source in units of 10 kpc

6.70 zbremss

A thermal bremsstrahlung spectrum based on the Kellogg, Baldwin & Koch polynomial fits to the Karzas & Latter numerical values. A routine from Kurucz is used in the case of high temperatures.

`par1`	=	plasma temperature in keV
`par2`	=	redshift
`K`	=	$(3.02 \times 10^{-15}/(4\pi D^2)) \int n_e n_I dV$, where D is the distance to the source (cm) and n_e, n_I are the electron and ion densities (cm^{-3})

6.71 zedge

An absorption edge.

$$
\begin{aligned}
M(E) &= 1 && \text{for } E < \texttt{par1} \\
&= \exp(-\texttt{par2}((E(1+\texttt{par3})/\texttt{par1})^3)) && \text{for } E > \texttt{par1}
\end{aligned}
$$

where :

par1	=	threshold energy
par2	=	absorption depth at threshold
par3	=	redshift

6.72 zgauss

A simple gaussian line profile. If the width is ≤ 0, then it is treated as a delta function.

$$A(E) = (1./(\sqrt{(}2\pi\texttt{par2}^2)) \exp(-.5((E(1+\texttt{par3}) - \texttt{par1})/\texttt{par2})^2))/(1+\texttt{par3})$$

where :

par1	=	line energy in keV
par2	=	line width (sigma) in keV
par3	=	redshift
K	=	total photons/cm^2/s in the line

6.73 zhighect

A high energy cutoff.

$$
\begin{aligned}
A(E) &= \exp((\texttt{par1} - E(1+\texttt{par3}))/\texttt{par2}) && \text{for } E > \texttt{par1} \\
A(E) &= 1 && \text{for } E < \texttt{par1}
\end{aligned}
$$

where :

par1	=	cutoff energy in keV
par2	=	e-folding energy in keV
par3	=	redshift

6.74 zpcfabs

A partial covering fraction absorption. Relative abundances are set by the `abund` command.

$$M(E) = \mathtt{par2}\exp(-\mathtt{par1}\sigma(E(1+\mathtt{par3}))) + (1-\mathtt{par2})$$

where σ(E) is the photo-electric cross-section (NOT including Thomson scattering) (see `phabs`) and

`par1`	=	equivalent hydrogen column (in units of 10^{22} atoms/cm^2)
`par2`	=	covering fraction ($0 < \mathtt{par2} \leq 1.$) (dimensionless)
`par3`	=	redshift

6.75 zplrefl

The total spectrum resulting from a power-law illumination of cold material, i.e., the power law together with its reflection spectrum. The reflected spectrum is calculated following Lightman and White (1988), *Ap.J.*, 335, 57, but with opacities calculated for cold material from Morrison and McCammon (1983), *Ap. J.*, 270, 119. The reflected spectrum is assumed to scale as the cosine of the inclination angle. This is not a bad approximation at small angles, but is increasingly inaccurate at large inclinations, especially as doppler effects are not included in this code. See George and Fabian (1991), *M.N.R.A.S.*, 249, 352; and Matt, Perola and Piro (1991), *Astron. Astrophys.*, 247, 25 for more accurate calculations of inclination effects. As Compton downscattering is important, the X–ray spectrum beyond the 2-20 keV band can affect the calculated shape of the reflection spectrum. The recommended maximum energy cutoff is 0.6 (= 300 keV).

`par1`	=	photon index of the intrinsic, illuminating power law
`par2`	=	fraction of power law escaping towards observer. Setting this to unity is equivalent to assuming an isotropic X-ray source.
`par3`	=	inclination of the disk to the observer (degrees)
`par4`	=	solid angle subtended by the disk/2π. Setting this to unity gives the spectrum as calculated in Lightman and White (1988).
`par5`	=	maximum energy extent of the power law in units of $m_e c^2$
`par6`	=	redshift
`K`	=	photons/keV/cm^2/s at 1 keV

6.76 zpowerlw

A simple photon power law.

$$A(E) = \mathtt{K}(E(1+\mathtt{par2})/(1\mathrm{keV}))^{(-\mathtt{par1})}/(1+\mathtt{par2})$$

where :

par1	=	photon index of power law (dimensionless)
par2	=	redshift
K	=	photons/keV/cm^2/s at 1 keV

6.77 zvarabs

A photoelectric absorption with variable abundances using Balucinska-Church and McCammon cross-sections. The column for each element is in units of the column in a solar abundance column of an equivalent hydrogen column density of 10^{22} cm^2. The Solar abundance table used is set by the abund command.

par1–par18	=	equivalent columns for H, He, C, N, O, Ne, Na, Mg, Al, Si, S, Cl, Ar, Ca, Cr, Fe, Ni, Co
par19	=	redshift

6.78 zvfeabs

Redshifted photoelectric absorption with all abundances tied to Solar except for iron. The Fe K edge energy is a free parameter.

par1	=	equivalent hydrogen column (in units of 10^{22} cm^{-2})
par2	=	abundance relative to Solar
par3	=	iron abundance relative to Solar
par4	=	Fe K edge energy
par5	=	Redshift

6.79 zwabs

A photo-electric absorption using Wisconsin (Morrison and McCammon; ApJ 270, 119) cross-sections.

$$M(E) = \exp(-\mathtt{par1}\sigma(E(1+\mathtt{par2})))$$

where σ (E) is the photo-electric cross-section (NOT including Thomson scattering) and

`par1`	=	equivalent hydrogen column (in units of 10^{22} atoms/cm^2)
`par2`	=	redshift

6.80 zwndabs

Photo-electric absorption from approximation to a warm absorber using Balucinska-Church and McCammon cross-sections. Relative abundances are set by the `abund` command.

$$\begin{aligned} M(E) &= \exp(-\mathtt{par1}\sigma(E(1+\mathtt{par3}))) && \text{for } E \geq \mathtt{par2} \\ &= 1. && \text{for } E \leq \mathtt{par2} \end{aligned}$$

where σ (E) is the photo-electric cross-section (NOT including Thomson scattering) and

`par1`	=	equivalent hydrogen column (in units of 10^{22} atoms/cm^2)
`par2`	=	Window energy (keV)
`par3`	=	redshift

Chapter 7

Associated programs

7.1 Introduction

A number of programs and subroutine libraries are available for manipulating the files used by XSPEC. Most of these are part of the FTOOLS package of tasks to read, write, and modify FITS-format files. There are a few tasks in the XANADU distribution for manipulating the older pre-FITS file formats (see Appendix E). However, we recommend that you use the FTOOLS conversion tasks to turn older files into the FITS format and then use the FTOOLS tasks to manipulate them.

The FTOOLS package is available by anonymous ftp from **legacy.gsfc.nasa.gov** in `software/xanadu/ftools` and through the WWW at the URL :

http://heasarc.gsfc.nasa.gov/0/docs/software/ftools/ftools_menu.html

If the FTOOLS package is installed, then help is available on any task by typing `fhelp taskname`. To get a complete list of FTOOLS, type `fhelp ftools`.

7.2 FTOOLS reading tasks

FDUMP

Prints the contents of a FITS file to the screen or to a file.

DMPRMF

Prints the contents of a FITS RMF file to the screen or to a file. This tool prints the RMF file in a more legible fashion than FDUMP.

7.3 FTOOLS manipulation tasks

GRPPHA

Defines (or redefines) and/or displays the grouping and quality flags, the important keywords, and the fractional systematic errors.

RBNPHA

Compresses a FITS PHA file to a user-defined number of channels. The output is a new file containing the revised PHA extension plus a direct copy of any other extensions in the original file.

RBNRMF

Compresses a FITS RMF file (containing the detector response matrix) in channel-space to give a user-defined number of resulting channels.

FPARKEY

Changes the value of a keyword in a FITS extension header.

MATHPHA

Performs arithmetical operations on PHA files.

ADDRMF

Adds together RMFs.

MARFRMF

Multiplies an RMF file by an ARF file.

7.4 FTOOLS conversion tasks

SF2PHA

Converts an old pre-FITS-format spectral file into a FITS PHA file.

RSP2RMF

Converts an old pre-FITS-format response file into a FITS RMF file.

STW2PHA

Converts the output of STWFITS (IRAF.stsdas.fitsio) into a FITS PHA file.

7.5 FTOOLS subroutines

The directory **callib/src/gen** contains a number of subroutines for reading and writing the extensions in FITS format spectral and response files. These are summarized below.

RDPHA1	Read a spectrum extension
WTPHA1	Write a spectrum extension
RDRMF1	Read the matrix extension
WTRMF1	Write the matrix extension
RDEBD1	Read the channel boundaries extension
WTEBD1	Write the channel boundaries extension
RDARF1	Read the effective area extension
WTARF1	Write the effective area extension

Appendix A

Overview of PLT

A.1 Command summary

`CLear`	–	Immediately clear the graphics device.
`COlor`	–	Change the default colour index.
`CONtour`	–	Produce a contour plot.
`CPD`	–	Change the plotting device.
`CQuit`	–	Clear the graphics device and return control to XSPEC.
`CSize`	–	Change the default character size.
`Error`	–	Control whether errors are displayed and used in fitting.
`EXit`	–	Exit PLT and return control to XSPEC.
`Fit`	–	Fit the PLT model to the data.
`FNy`	–	Evaluate the model at the specified location.
`FOnt`	–	Change the default text font.
`Freeze`	–	Freeze a parameter value.
`GAp`	–	Change the default gap size between the data and the edge.
`Grid`	–	Control the location of the major and minor tic marks.
`Hardcopy`	–	Make a file that can later be printed.
`HElp`	–	Obtain help on any PLT command.
`Imodel`	–	Numerically integrate the model over specified range.
`LAbel`	–	Add or remove labels from the plot.
`LIne`	–	Control whether a line is used to connect data points.
`LOg`	–	Control whether data is plotted using a $\log_{10}$ scale.
`LStyle`	–	Change the default style of the line connecting the data points.
`LWidth`	–	Change the default line width.
`MArker`	–	Control whether the data points are plotted with markers.
`MOdel`	–	Define a PLT model.
`Newpar`	–	Change a parameter value associated with the model.
`Plot`	–	Immediately re-plot the data.
`PRompt`	–	Change the "`PLT>`" prompt.
`Rescale`	–	Reset the minimum and maximum plot range.

`SCr`	–	Change the color representation of the specified color index.
`SHow`	–	Display the values of PLT internal variables.
`SKip`	–	Control how PLT divides data into vectors.
`STatistics`	–	Compute various statistical properties of the data.
`THaw`	–	Allow a parameter value to vary during a fit.
`Time`	–	Control whether the time stamp is plotted.
`Uncertainty`	–	Compute the uncertainty in a parameter value.
`VErsion`	–	Display date of the most recent modification to PLT.
`Viewport`	–	Control the size of the viewport plotting area.
`WData`	–	Write a QDP data file to disk.
`WEnviron`	–	Write both QDP data and header files to disk.
`WHead`	–	Write a QDP header file to disk.
`WModel`	–	Write a model file to disk.
`Xaxis`	–	Define the method used to calculate the x-variable.
`Yaxis`	–	Define the y-axis scale for a contour plot.
`$`	–	Execute operating system commands.
`@`	–	Read commands from a PLT command (.PCO) file.

A.2 XSPEC graphics

Extensive documentation for the PLT graphics routine is available in the "The QDP/PLT Users's Guide" and also from the interactive help. This appendix is intended to provide information to assist in using PLT from within the XSPEC program.

Within XSPEC, it is possible to set your graphics device using the `cpd` command. Any PGPLOT[1] device supported by your local version of PGPLOT is accepted. The `cpd` command can be used to display a list of all PGPLOT devices. If you fail to enter a device name, you will be prompted for a PGPLOT device every time you generate a new plot.

From XSPEC, there are two ways to call the PLT routine. The `plot` command will produce a graph and control will return immediately to XSPEC. To go into interactive plot mode, use `iplot`. This will produce exactly the same graph but the "`PLT>`" prompt will appear. At this point, you can enter PLT commands to inspect interesting parts of the graph, add labels, or make a hardcopy file for later printing.

A.3 Getting started with PLT

Typing `HElp` at the "`PLT>`" prompt will provide you with help concerning the PLT commands. The PLT interactive help is used in exactly the same way as the XSPEC

[1]PGPLOT is the name of a Graphics Subroutine Library written by T. J. Pearson at the California Institute of Technology.

help. If you wish to read the interactive help on PLT without running the PLT routine, you should use the (system level) XHELP PLT command.

The most common command used in PLT is `Rescale`. Using `Rescale X` followed by two numbers, will set the minimum and maximum of the plotted x-range to the numbers specified. If you type `Rescale X` (with no arguments), the minimum and maximum values will be reset to their default values. Likewise, `Rescale Y` can be used to set the y-range.

Most PLT commands do not cause the screen to be updated the way the `Rescale` command does, thus allowing you to make several changes to the the graph without having to wait for the screen to be updated after every change. If, at any time, you wish to see what the current graph looks like, enter the `Plot` command, which will cause the display to be redrawn.

The `LAbel` command can be used to add labels to various locations on the graph. For example, typing `LAbel Top EXOSAT was great` will cause the message 'EXOSAT was great' to appear at the top of the graph the next time the display is redrawn. The command `LAbel Top` followed by a return will remove that message.

The `Hardcopy` command is used create a file that can later be printed. `Hardcopy` does not reproduce what currently is visible on the graphics display, but rather what you would see if you re-issued the `Plot` command. The command `Hardcopy ?` can be used to see what the default hardcopy device is. If you do not like the default, you can override it. For example, assume that your version of PGPLOT supports the QMS device. To generate a file suitable for printing on a QMS printer you would enter `Hardcopy /QMS` and a `PGPLOT.QMPLOT` file will be generated in your current directory. Many sites have created a system-wide indirect file that will both create the hardcopy file, and then immediately print it. If you are at one of these sites, then `@HARD` will directly produce a plot on the printer.

In PLT it is possible to fit models to the displayed data. However, it is important to remember that PLT does not fold the model through the detector response, and therefore, PLT models should not be used to fit X-ray spectra.

To exit PLT and return to the XSPEC level, enter the PLT `EXit` command. When you return to XSPEC, any changes that you made to the plot will be forgotten.

Appendix B

Fitting with few counts/bin

B.1 Theory

Cash (ApJ **228**, 939) showed that the χ^2 minimization criterion is a very bad one if any of the observed data bins had few counts. A better criterion is to use a likelihood function :

$$C = 2\sum_{i=1}^{N}(y(x_i) - y_i \ln y(x_i) + \ln y_i!)$$

where y_i are the observed data and $y(x_i)$ the values of the function. Minimizing C for some model gives the best-fit parameters. Furthermore, this statistic can be used in the same, familiar way as the χ^2 statistic to find confidence intervals. One finds the parameter values that give $C = C_{min} + N$, where N is the same number that gives the required confidence for the number of interesting parameters as for the χ^2 case.

Castor (priv. comm.) has pointed out that a better function to use is :

$$C = 2\sum_{i=1}^{N}(y(x_i) - y_i + y_i(\ln y_i - \ln y(x_i)))$$

This differs from the first function by a quantity that depends only upon the data. This second function does provide a goodness-of-fit criterion similar to that of χ^2 and it is now used in XSPEC. It is important to note that the C-statistic assumes that the error on the counts is pure Poisson, and thus it cannot deal with data that already has been background subtracted, or has systematic errors.

B.2 Practice

XSPEC uses a variant of Marquardt's algorithm described in §11.5 of "Data Reduction and Error Analysis for the Physical Sciences" by Bevington. (The reader is advised that this description is designed to be read in conjunction with Bevington.) The algorithm turns on finding a matrix α_{jk} and a vector β_k such that the equation :

$$\beta_k = \sum_j \delta a_j \alpha_{jk}$$

gives sensible values of the change in parameters, δa_j, for the fitting function. Bevington §11.4 gives the derivation of α and β and shows that β is parallel to the gradient of χ^2.

Now the C statistic has a gradient with respect to the parameters of the fitting function of :

$$-\frac{1}{2}\frac{\partial C}{\partial a_k} = \sum_i (\frac{y_i}{y} - 1)\frac{\partial y}{\partial a_k} = \beta_k$$

So, following Bevington, expand $y(x_i)$ about y_0 :

$$y(x_i) = y_0(x_i) + \sum_j \frac{\partial y_0(x_i)}{\partial a_j}\delta a_j$$

substitute into C and minimize with respect to the changes in the parameters :

$$\frac{\partial C}{\partial \delta a_k} = 2\sum_i (\frac{\partial y_0(x_i)}{\partial a_k} - y_i(y_0(x_i) + \sum_j \frac{\partial y_0(x_i)}{\partial a_j}\delta a_j)^{-1}\frac{\partial y_0(x_i)}{\partial a_k}) = 0$$

so to first order in the parameter changes :

$$\sum_i (\frac{y_i}{y_0(x_i)} - 1)\frac{\partial y_0(x_i)}{\partial a_k} = \sum_j (\sum_i \frac{y_i}{y_0^2(x_i)}\frac{\partial y_0(x_i)}{\partial a_j}\frac{\partial y_0(x_i)}{\partial a_k}\delta a_j)$$

or :

$$\beta_k = \sum_j \delta a_j \alpha_{jk}$$

where :

$$\alpha_{jk} = \sum_i \frac{y_i}{y_0^2(x_i)} \frac{\partial y_0(x_i)}{\partial a_j} \frac{\partial y_0(x_i)}{\partial a_k}$$

These α and β then are substituted for those used in the χ^2 case and the algorithm works as required.

There is one further difference in XSPEC between the χ^2 and likelihood methods, which is caused by the fact that XSPEC uses an analytic formula for setting the model normalisation. In the χ^2 case, this means multiplying the current model by :

$$(\sum_i \frac{y_i y(x_i)}{\sigma_i^2})/(\sum_i \frac{y(x_i)^2}{\sigma_i^2})$$

where σ_i is the error on y_i. In the likelihood case the corresponding factor is :

$$\sum_i y_i / \sum_i y(x_i)$$

Appendix C

Adding models to XSPEC

C.1 Standard models

If you have a model that will be used frequently, you may want to include it among the standard models so that it will be listed after the `model ?` command. To do so, you first must create a subroutine that calculates the model spectrum given an input array of energy bins and an array of parameter values. The spectrum for an **additive** model should be in terms of photons/cm^2/s (not photons/cm^2/s/keV), while for a **multiplicative** model it is the multiplicative factor for that bin.

The five arguments in the calling sequence are EAR(0:NE), NE, PARAM(*), IFL, and PHOTAR(NE). EAR(0:NE) are input and are the boundaries for the NE energy bins. These are set by the response matrices of the detectors in use, so you should not make any assumptions about the values of EAR. PARAM(*) are input and are the values of the model parameters. IFL is an integer which specifies which dataset these energies are for, it exists to allow multi-dimensional models where the function might also depend on eg pulse-phase in a variable source. PHOTAR(NE) is the output array and should not be assumed to have any particular values on input. When you have created your model subroutine, move it to the XSPEC local**functions** subdirectory (**local/spectral/xspec/src/functions**).

To set the default parameters and tell XSPEC that your model exists, you must edit the file `lmodel.dat` in the local**model** subdirectory (**local/spectral/xspec/src/model**). The file `lmodel.dat.EXAMPLE` can be used as a template. The first line for each model gives the model name, the number of parameters, the low and high energies for which the model is valid, and the name of the subroutine in the **functions** subdirectory. There then should be one succeeding line for each parameter. Each line contains the parameter name, a default starting value, the hard minimum, the soft minimum, the soft maximum, the hard maximum, and the initial delta.

C.1.1 Unix

Edit `$XANADU/spectral/xspec/src/makefile.xan` to set the LFUNLIB variable as indicated in the line that is commented out. Now

```
cd $XANADU/local/spectral/xspec/src/functions
xmake LIB
cd $XANADU/spectral/xspec/src/model
msetup
alib addmod $XANBIN/lib/libxspec2.a
alib mulmod $XANBIN/lib/libxspec2.a
alib mixmod $XANBIN/lib/libxspec2.a
mv model_* $XANADU/spectral/xspec/manager
cd ..
xfor xspec
mv xspec.o xspec_$EXT.o xmake link
xmake install
```

C.1.2 VMS

In `[SPECTRAL.XSPEC.SRC]XSPEC.OPT` add `XANBIN:[LIB]LIBXSPEC_LFN/LI` to the link list. Now:

```
SET DEF XANADU:[LOCAL.SPECTRAL.XSPEC.SRC.FUNCTIONS]
FORALL
LIBALL XANBIN:[LIB]LIBXSPEC_LFN
SET DEF XANADU:[SPECTRAL.XSPEC.SRC.MODEL]
MSETUP
ALIB ADDMOD XANBIN:[LIB]LIBXSPEC2
ALIB MULMOD XANBIN:[LIB]LIBXSPEC2
ALIB MIXMOD XANBIN:[LIB]LIBXSPEC2
REN MODEL_*.* XANADU:[SPECTRAL.XSPEC.MANAGER]
SET DEF [-]
XFOR XSPEC
LNK XSPEC
XSTALL XSPEC
```

C.2 User models under VAX/VMS

In the VAX/VMS version of XSPEC, it is possible to have a user-written model in a shareable executable that is separate from the main program. This executable has the advantage that only the user model needs to be linked and not XSPEC as a whole. It is particularly useful for sophisticated models not used very often and keeps the program smaller. The current version of XSPEC can have one additive and one multiplicative user model but it is easy with the method described above to add more user models. In **xanadu:[spectral.xspec.functions]** two template routines can be found; `XSUSA1.FOR` for the additive model and `XSUSM1.FOR` for the multiplicative model. The user can start from those routines and write his or her own model. Only the name of the subroutine in the file and the variables returned to XSPEC must be the same. The variables are described in `XSUSA1.FOR` and `XSUSM1.FOR`. The files also contain the subroutines `MOUSA1` and `MOUSM1`, which initialize parameter names, initial values for the model parameters and the number of parameters of the model–which can be over 20, such as the normalization in the case of the additive model.

The new user model then can be compiled in the normal way and linked with the `MKUMD` command, which is supplied with the software. For a debug, version `DBUMD` can be used. The shareable executable then is ready, and the `SETUMD` command can be used to set the logical `UMDSHR` to point to the right user model before starting the XSPEC session. In this way, the user can have several user models as shareable executables and only needs to point to the one he intends to use in XSPEC.

Example Suppose the user has his additive model routine in the file `USER1.FOR` in **USRDISK:[RONTGEN.XSPEC]**.

$ set def usrdisk:[rontgen.xspec]	!	Change directory.
$ fortran user1	!	Compile.
$ mkumd user1	!	Link the executable.
$ setumd user1	!	Set logical umdshr.
$ xspec	!	Start xspec.
XSPEC> model wabs useradd1	!	Define model.

C.3 Table models

A very simple way of fitting with user-defined models is available for a particular class of models. These are models that can be defined by a grid of spectra, with the elements of the grid covering the range of values of the parameters of the model. For instance, for a one-parameter model, a set of model spectra can be tabulated for different values of the parameter (P1, P2, P3, etc.) The correct model spectra for a value P then is calculated by interpolation on the grid. The generalisation to more parameters works in the obvious way. As with standard models, the spectra should be in terms of flux-per-bin and not flux-per-keV. Any set of energy bins can be used, and XSPEC will interpolate the model spectra onto the appropriate energy bins for the detectors in use. It is therefore a good idea to choose energy bins such that the spectrum is well-sampled over the range of interest. The file structure for these models is a FITS format and is described in the

OGIP memo OGIP/92-009 (available from the legacy.gsfc.nasa.gov anonymous ftp site or the HEASARC WWW site).

Appendix D

The XSPEC/PLT parser

D.1 Introduction

All communication with the user in XSPEC is performed through the XSPEC parser, which itself sits on top of the PLT parser. The PLT parser provides a set of low level routines with a number of features such as inline editing and command recall, while the XSPEC parser handles the more complicated XSPEC-specific operations such as extraction of ranges and reprompting for failed inputs. This appendix summarizes the features of these parsers from the point of view of the user.

D.2 Basics

The parser routines provide a general, easy interface between a program and the user, particularly for command-driven or other highly interactive programs. The basic cycle begins with the user inputting a line (optionally after some prompt) on the terminal. This line then can be analyzed by the program as a string of arguments. The program then can try to interpret each argument as a number or as a character string. Character strings can be checked if they are one of an allowed list of options. The actual details of how a line is interpreted depend on the particular program and how it uses the various parser facilities. An example of an input line might be:

```
Foo , 4.7 "Some information" - ! A comment
```

which is divided into three arguments: 'Foo', '4.7' (which can be interpreted as a number), and "Some information."

When parsing an input line, there are several special characters:

blank	( )	(a horizontal tab is also a blank)
comma	(,)	
special delim.	(|)	(the vertical bar)
comment char.	(!)	
begin quote	(")	(Note, the begin and end quote are the
end quote	(")	same character by default, the ")
internal com.	(%)	

There are special characters whose interpretation depends on their position in the line:

infinite skip	(/)	
continuation	(-)	
end of file	(/*)	or CTRL-Z under VMS
indirect file	(@)	

Note that if command editing and recall are enabled, then a variety of control characters have special meanings for the parser (see below).

D.3 Delimiting arguments

When parsing a line for an argument, initial and trailing blanks are ignored. In addition, an argument field is delimited by a comma, or some of the special characters (particularly the comment, end of input, special delimiter characters). If an argument has no non-blank characters before a delimiter is reached, the argument is said to be an empty, null, or skipped argument. Conventionally, an empty argument is to be treated as indicating some default value. For example:

```
foo            !  is a line consisting of a single argument 'foo'.
foo bar
foo,bar
foo   bar
foo,  bar
foo   , bar    !  are all lines with two arguments 'foo' followed
foo|bar        !  by 'bar'.
,, foo         !  have two initial empty arguments followed by
, , foo        !  'foo'.
bar,,foo       !  has 'bar', an empty argument and 'foo'.
```

D.4 Comments

When looking for arguments, the comment character is treated logically as the end of the line. Thus, any following characters are ignored (and can be used as a comment). Thus, the line

```
 foo !  bar     !  has a single argument 'foo' followed by a comment.
                !  Comment strings cannot be read by the program.
 , !a comment   !  has two empty arguments before the comment.
```

D.5 Quotes

If a user wishes to input an argument that contains a blank, comma, comment character, special delimiter, etc., then the argument should begin with the begin quote (“) character, and have an end quote (”) character at the end. Reaching the end of the input line is equivalent to having the end quote character. If you want to have the string include the end quote character, then you should double the character (similar to the way that FORTRAN handles character strings with embedded single quotes). By default, the begin quote and end quote characters are the same (the ASCII double-quote: "), although a particular program that uses XPARSE is allowed to change these (and all special characters) to different characters. For example:

```
"This argument contains blanks"
"This argument ends with a single """," followed by a 2nd argument"
"The initial "" allow the words following the ! to be read"
" A final "" is optional for the last argument
```

D.6 The infinite skip

The function of some characters depends on their location in the input line. For example, if the infinite skip character (/) is the last non-blank character on the line, or the last before the comment character, it is treated as a character delimiter, similar to a comma. In addition, it is equivalent to an infinite series of skipped or empty arguments existing before the end-of-line condition is raised. This treatment is mostly for compatibility with the standard FORTRAN List Directed I/O standards where a slash (/) indicates that all remaining arguments of a read are to retain their current values. If the character is placed anywhere else in the line, it is just an ordinary character and will be treated as part of an argument. Examples of lines that use the infinite skip character:

```
Foo, bar, bletch /
Foo bar bletch,/
```

```
Foo bar bletch/ ! a comment after the infinite skip char
```

all have three non-empty arguments, followed by infinite skips. In the following cases the slash is NOT treated as an infinite skip character:

```
 file/device    !  This is a single argument 'file/device'
 foo/,          !  This is a single argument 'foo/'
 "foo/"         !  As is this.
```

D.7 The continuation character

If the continuation character (-) is the last non-blank character on a line, or the last before the comment character, then the next line of input will be treated as logically concatenated to the current line. For terminal input, the user will receive a prompt "->" indicating that the line is treated as a continuation. Note that the continuation character is NOT a delimiter, and thus must be separated from any earlier argument by a blank or some other delimiter. Examples:

```
These two lines -
are really one line !with 7 arguments
This line is not concatenated- !the 5th argument is "concatenated-"
with the next.
Even if you concatenate -!you are allowed to have comments
on each input line ! this was equivalent to an 8 argument line.
```

Because of the way continuations are processed, the user may not see the prompt until the program processes the last argument on the line. Thus, the type-ahead for the continuation may be 'invisible'.

D.8 The terminator

When a user inputs a line, the first few characters are checked for two special strings: the EOF string and the indirect character. Both terminators must start at the first character of the string. The EOF string (by default the /*) is a way of producing an end-of-file signal for the read. For terminal input, an equivalent way of generating an end-of-file is a CTRL-Z (this also can be used to terminate the previous input line). For example:

```
/*  ! This is a generated EOF
 /* ! This is a single argument input string "/*".
```

D.9 The indirect character

The indirect character tells the program to start reading its input from the file specified on the remainder of the line. The line is NOT parsed out and all characters are considered significant for the file name. In particular, a comment field is NOT allowed. Indirect files can be nested generally to a depth of 10, although circumstances may reduce the acceptable depth. The default extension for an indirect file is `.XCO`, although this default is under the control of the program (in XSPEC it is `.XCM` and in PLT `.PCO`). An example:

```
@file
```

will cause subsequent input to be read from FILE.XCO. This command file can contain strings of the form **%N%**, where N is an integer, which will be replaced by the Nth argument in the invocation, e.g., **@file 5.0 -2** will replace **%1%** by 5.0 and **%2%** by -2. If arguments are not given on the command line, then **%N%** will be replaced with nothing (i.e., removed).

D.10 Matching keyword arguments

A common type of argument that a program might expect would be a character string that is matched against a list of possible keywords, such as a list of commands to tell the program what to do next. In general, case is not significant for such matches, (i.e., 'KEYWORD' would match 'keyword'). Also, partial matches are usually allowed, (i.e., 'key' is a partial match to 'keyword'. If the list of allowed matches includes two such partial matches, then usually the first in order is taken. For example if the match-list is ('keyword','keyhole',...) then 'key' matches 'keyword'. Of course, any exact match takes precedence, in ('keyword',...,'key') then 'key' matches 'key'.

The particular program in use may modify any of these conditions individually. Case may be made significant, partial matches may be disallowed, or, if allowed, then only unique partial matches (so that 'keyh' would be necessary for a legal argument matched against the list ('keyhole','keyword',...)).

Some programs will use a special-choice subroutine, which will prompt a user to input a single argument that will be compared with a list of choices. An example of such an interface might be as follows:

```
Please choose a keyword: (default)
```

where the string in parentheses is returned as the default if the user just returns an empty line. If the user returns a ?, then a list of all the legal options will be displayed. An EOF should be handled by the program as an exceptional reply (e.g., to revert to a previous stage of the program).

An even more special case is when the user is prompted to provide an answer to a question:

```
Do you know how to respond to this? (y)
```

In these cases, the possible answers are 'yes' and 'no'. Again, the default value is in the parentheses.

D.11 Parser internal commands

All parser internal commands are delimited by the internal command character, which defaults to %. Internal commands must be both prefixed and suffixed by this character. Internal commands are trapped by the parser, acted upon, and are not passed through to the calling program. Therefore, internal commands are common to all applications. The individual commands are described below.

D.11.1 %Echo%

When reading a script, the default is to echo the prompt and the response read. This default is very useful, as it allows you to watch the effect the script is having on your current program. However, in certain cases it is useful to suppress this information. This can be done with the **%Echo%off** command. For example:

```
one              !  This line will be echoed
%e%off           !  This line will not be echoed
two              !  No echo
%e%on            !  Still no echo
four             !  This line will be echoed.
```

Currently, **%Echo%** is a global command. Therefore, if you write any scripts that turn off the echo, you should add an **%e%on** line at the bottom to turn the echo back on for the next script.

D.11.2 %EDit%

By default, the parser does not support line editing because line editing involves single character IO, which may be slow when used across a network, and because line editing may not work on all terminals. Therefore, you should try out line editing interactively, and, if you like it, activate it by defining the logical name/environment variable `GTBUF_EDIT` as specified below.

Line editing can be activated by typing **%ed% on** at any prompt. Once activated, you can use the cursor keys and some control keys allow you to 'edit' your current command line. Since the parser saves the 20 most recently entered commands, a previously entered command can be recalled, edited, and then re-issued.

The **%ed%** command also allows you to set the control keys associated with the various editing functions. This is done by entering a string of two-character commands. The first character should be selected from (U, D, L, R, B, E, W, X, Z) to denote (Up, Down, Left, Right, Begin, End, reWrite, erase, eof), and the second character tells the parser what control character to associate with that command. The the string 'UP' would cause ˆP to be associated with the Up command. The following table lists the default control keys, function name, and action:

Function	Default	Action
Up	ˆB	Cursor up, recalls the previous command
Down	ˆN	Cursor down, go to next command
Left	ˆD	Moves cursor left in the current line
Right	ˆF	Moves cursor right in the current line
Begin	ˆA	Moves cursor to the beginning of the current line
End	ˆE	Moves cursor to the end of the current line
reWrite	ˆR	Rewrites the current line
X(erase)	ˆU	Erases the current line
Z(eof)	ˆZ	The end-of-file character

The default settings are:

%ED%OFF UB DN LD RF BH EE WR XU ZZ

which use the same control keys as VMS when editing is turned on.

If you like line editing and wish to activate it automatically, then you should define the logical name/environment variable to contain the string that you normally would enter following the **%ed%** command. For example:

VMS : `DEFINE GTBUF_EDIT "ON"`
UNIX: `setenv GTBUF_EDIT "ON UP DN LB RF BA EE"`
DOS : `SET GTBUF_EDIT=ON`

For both the VMS and DOS examples, line editing is activated with the default control key assignments. For the UNIX example, editing is activated but with emacs-style control keys. Note, for the UNIX example, ˆR would still rewrite the line.

The command **%ed%?** can be used to display your current edit settings and the command **%ed%ver** will display the EDICOM version number.

VMS users should note that editing is implemented using the SYS$QIO function using terminal read and write operations. This means that it is impossible to use line editing to input data from a file. This situation can cause problems in batch jobs or when reading a VMS command file. Of course, all standard operations, such as reading a script, will work. To avoid problems in batch, you should make sure that `GTBUF_EDIT` is only defined for interactive jobs.

UNIX users should note that if a program crashes when line editing is active, your terminal may be left in a no-echo state. You should use the UNIX `reset` command to reset the terminal characteristics.

D.11.3 %Log%

This command will cause a log file to be created. The log file should contain a copy of everything that appears on your terminal screen after the log command has been issued. The default name of the script file is `LOG.LOG`, although various programs can override this default. To create a log file with a different name, place the name after the log command, thus `%Log% silly` will create a log file called `silly.log`.

D.11.4 %N%

This command is only valid in scripts and will be replaced by the parser with the Nth argument of the script and then passed on to the calling program. Arguments of scripts are not limited to numbers but also may be strings.

D.11.5 %Recall%

The command to display the recall command stack is called **%Recall%**. Currently only one argument is supported– 'All'–which will display the entire command stack. Note that currently this is not the same command stack as is used by the XSPEC `reexecute` command.

D.11.6 %Script%

This command will cause a script file to be created. The script file records all prompts and corresponding user responses. The format of the file is response, comment character, prompt. Thus, the script file can be used as a history file showing your response to the various prompts, or it can be used as an indirect command file. The default name of the script file is `SCRIPT.XCO`, although various programs can override this default. To create a script file with a different name, place the name after the script command, thus `%Script% data.xco` will create a script file called `data.xco`.