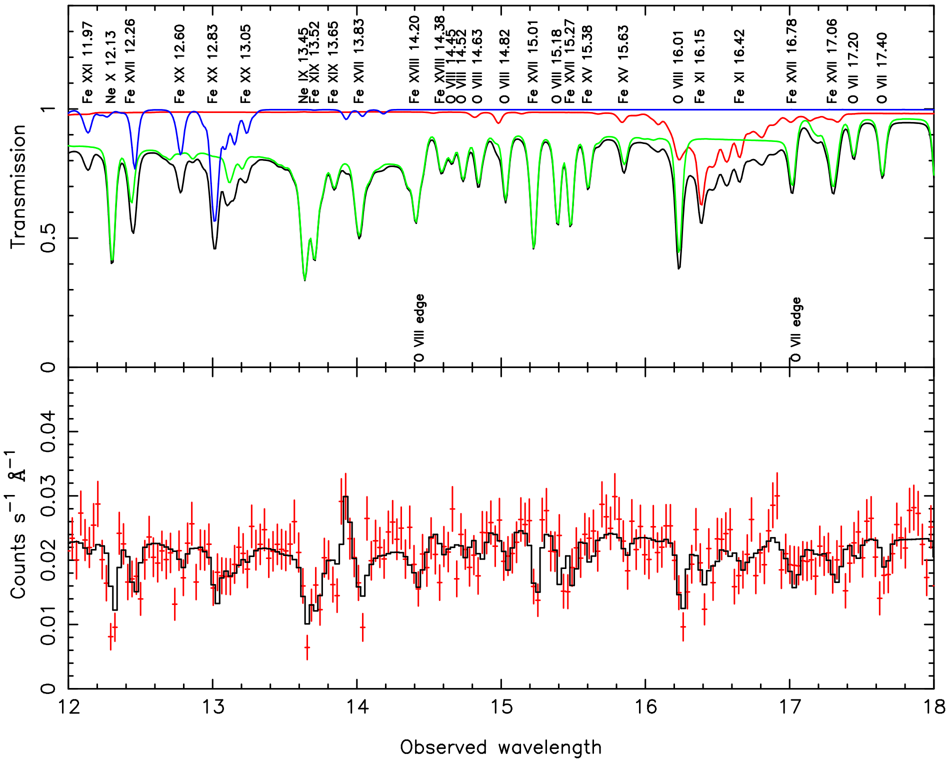
SPEX User's Manual

Jelle Kaastra

Rolf Mewe & Ton Raassen



Version 2.0

February 9, 2005

0.1 Preface

At the time of releasing version 2.0 of the code, the full documentation is not yet available, although major parts have been completed (you are reading that part now). This manual will be extended during the coming months. Check the version you have (see the front page for the date it was created). In the mean time, there are several useful hints and documentation available at the web site of Alex Blustin (MSSL), who I thank for his help in this. That web site can be found at:
http://www.mssl.ucl.ac.uk/ ajb/spex/spex.html
The official SPEX website is:
http://www.sron.nl/divisions/hea/spex/index.html
We apologize for the temporary inconvenience.

Contents

Chapter 1

Introduction

1.1 Preface

At the time of releasing version 2.0 of the code, the full documentation is not yet available, although major parts have been completed (you are reading that part now). This manual will be extended during the coming months. Check the version you have (see the front page for the date it was created). In the mean time, there are several useful hints and documentation available at the web site of Alex Blustin (MSSL), who I thank for his help in this. That web site can be found at:
http://www.mssl.ucl.ac.uk/ ajb/spex/spex.html
The official SPEX website is:
http://www.sron.nl/divisions/hea/spex/index.html
We apologize for the temporary inconvenience.

1.2 Sectors and regions

1.2.1 Introduction

In many cases an observer analysis the observed spectrum of a single X-ray source. There are however situations with more complex geometries.
Example 1: An extended source, where the spectrum may be extracted from different regions of the detector, but where these spectra need to be analysed simultaneously due to the overlap in point-spread function from one region to the other. This situation is e.g. encountered in the analysis of cluster data with ASCA or BeppoSAX.
Example 2: For the RGS detector of XMM-Newton, the actual data-space in the dispersion direction is actually two-dimensional: the position z where a photon lands on the detector and its energy or pulse height E as measured with the CCD detector. X-ray sources that are extended in the direction of the dispersion axis ϕ are characterised by spectra that are a function of both the energy E and off-axis angle ϕ. The sky photon distribution as a function of (ϕ,E) is then mapped onto the (z,E)-plane. By defining appropriate regions in both planes and evaluating the correct (overlapping) responses, one may analyse extended sources.
Example 3: One may also fit simultaneously several time-dependent spectra using the same response, e.g. data obtained during a stellar flare.
It is relatively easy to model all these situations (provided that the instrument is understood sufficiently, of course), as we show below.

1.2.2 Sky sectors

First, the relevant part of the sky is subdivided into sectors, each sector corresponding to a particular region of the source, for example a circular annulus centered around the core of a cluster, or an arbitrarily shaped piece of a supernova remnant, etc.
A sector may also be a point-like region on the sky. For example if there is a bright point source superimposed upon the diffuse emission of the cluster, we can define two sectors: an extended sector for the cluster emission, and a point-like sector for the point source. Both sectors might even overlap, as this example shows!
Another example: the two nearby components of the close binary α Centauri observed with the XMM -Newton instruments, with overlapping point-spread-functions of both components. In that case we would have two point-like sky sectors, each sector corresponding to one of the double star's components.
The model spectrum for each sky sector may and will be different in general. For example, in the case of an AGN superimposed upon a cluster of galaxies, one might model the spectrum of the point-like AGN sector using a power law, and the spectrum from the surrounding cluster emission using a thermal plasma model.

1.2.3 Detector regions

The observed count rate spectra are extracted in practice in different regions of the detector. It is necessary here to distinguish clearly the (sky) sectors and (detector) regions.

A detector region for the XMM EPIC camera would be for example a rectangular box, spanning a certain number of pixels in the x- and y-directions. It may also be a circular or annular extraction region centered around a particular pixel of the detector, or whatever spatial filter is desired. For the XMM RGS it could be a specific "banana" part of the detector-coordinate CCD pulse-height plane (z,E).
Note that the detector regions need not to coincide with the sky sectors, neither should their number to be equal! A good example of this is again the example of an AGN superimposed upon a cluster of galaxies. The sky sector corresponding to the AGN is simply a point, while, for a finite instrumental psf, its extraction region at the detector is for example a circular region centered around the pixel corresponding to the sky position of the source.
Also, one could observe the same source with a number of different instruments and analyse the data simultaneously. In this case one would have only one sky sector but more detector regions, namely one for each participating instrument.

1.3 Different types of spectral components

In a spectral model SPEX uses two different types of components, called *additive* and *multiplicative* components respectively. Additive components have a normalisation that determines the flux level. Multiplicative components operate on additive components. A delta line or a power law are typical examples of additive components. Interstellar absorption is a typical example of multiplicative components.

The redshift component is treated as a multiplicative component, since it operates on additive components.

Additive components can be divided into two classes: simple components (like power law, blackbody etc.) and plasma components, that use our atomic code. For the plasma components it is possible to plot or list specific properties, while for the simple models this is not applicable.

Multiplicative components can be divided into 3 classes. First, there are the absorption-type components, like interstellar absorption. These components simply are an energy-dependent multiplication of the original source spectrum. SPEX has both simple absorption components as well as absorption components based upon our plasma code. The second class consists of shifts: redshift (either due to Doppler shifts or due to cosmology) is the prototype of this. The third class consists of convolution-type operators. An example is Gaussian velocity broadening.

For more information about the currently defined spectral components in SPEX, see chapter 4.

Chapter 2

Syntax overview

2.1 Quit: finish the program

The quit option exits the execution of SPEX, closes the open plot-devices and scratch files (if any) and, if requested outputs the cpu-time statistics.

2.1.1 Syntax

The following syntax rule applies:
quit - quit the program as described above.

2.2 System

2.2.1 Overview

Sometimes it can be handy if SPEX interacts with the computer system, for example if you run it in command mode. You might want to check the existence of certain file, or run other programs to produce output for you, and depending on that output you want to continue SPEX.
Therefore there is an option to execute any shell type commands on your machine, using the fortran "call system" subroutine.
Another useful goody is the possibility to stop SPEX automatically if you find some condition to occur; this might be useful for example if you have a program running that calls SPEX, and depending on the outcome of SPEX you might want to terminate the execution. This is achieved in SPEX by testing for the existence of a file with a given filename; if the file exists, SPEX stops immediately execution and terminates; if the file does not exist, SPEX continues normally.

2.2.2 Syntax

The following syntax rules apply:
system exe #a - execute the command #a on your UNIX/linux shell
system stop #a - stop SPEX if the file #a exists

2.2.3 Examples

`system exe "ls -l"` - give a listing of all file names with length in the current directory
`system exe "myfortranprogram"` - execute the fortran program with name "myfortranprogram"
`system stop testfile` - stop SPEX if the file with name testfile exists; if this file does not exist, continue with SPEX

2.3 Menu: Menu settings

2.3.1 Overview

When command lines are typed, it frequently happens that often the first keywords are identical for several subsequent lines. This may happen for example when a plot is edited. SPEX offers a shortcut to this by using the menu command.

2.3.2 Syntax

The following syntax rules apply:
menu none - Quit the current menu text settings (i.e., return to the default spex prompt.
menu text #a - For all following commands, the text string #a will be appended automatically before the following commands

2.3.3 Examples

`menu text plot` - All following commants will get the "plot" keyword put in front of them. So if the next commant would be "plot dev xs" it is sufficient to type "dev xs" instead.
`menu none` - Return to the normal SPEX prompt.
`menu text "par 1 2"` - All following commands wil get the "par 1 2" keywords put in front of them. The next command could be "t val 4.", which will be expanded to the full "par 1 2 t val 4." to set the temperature of sector 1, component 2 to 4 keV. Note that here the text has three keywords (par, 1, 2) and hence it has to be put between "", to indicate that it is a single text string. If there is only one keyword, these "" are not necessary.

2.4 Watch

2.4.1 Overview

If you have any problems with the execution of SPEX, you may set the watch-options. For example, if the computation time needed for your model is very large, it could be wise to check which parts of the program consume most of the cpu-time. You might then hope to improve the performance by assuming other parameters or by re-structuring a part of the code. In this case, you set the "time" flag to true (see below), and at the end of the program you will see how much time was spent in the most important subroutines.
Another case occurs in the unlikely case that SPEX crashes. In that case it is recommended to re-execute the program, saving all commands onto a log-file and use the "sub" flag to report the entering and exiting of all major subroutines. This makes it more easy to find the source of the error.
Timing is done by the use of the stopwatch package by William F. Mitchell of the NIST, which is free available at the web. If the time flag is set to true, on exit SPEX will report for each subroutine the following execution times (in s):

- The user time, i.e. the cpu-time spent in this subroutine
- The system time, i.e. the overhead system time for this subroutine
- The wall time, i.e. the total time spent while executing this subroutine.

Also the totals for SPEX as a whole are given (this may be more then the sum of the subroutine components, since not all subroutines are listed separately; the wall tiem for SPEX as a whole also includes the time that the program was idle, waiting for input from the user).

2.4.2 Syntax

The following syntax rules apply:
watch time #l - set the "time" flag to true or false.
watch sub #l - set the flag that SPEX causes to report each major subroutine it enters or exits

2.4.3 Examples

`watch time t` - set the "time" flag to true
`watch sub f` - set the subroutine report flag to false

2.5 Log: Making and using command files

2.5.1 Overview

In many circumstances a user of SPEX wants to repeat his analysis for a different set of parameters. For example, after having analysed the spectrum of source A, a similar spectral model and analysis could be tried on source B. In order to facilitate such analysis, SPEX offers the opportunity to save the commands that were used in a session to an ascii-file. This ascii-file in turn can then be read by SPEX to execute the same list of commands again, or the file may be edited by hand.

The command files can be nested. Thus, at any line of the command file the user can invoke the execution of another command file, which is executed completely before execution with the current command file is resumed. Using nested command files may help to keep complex analyses manageable, and allow the user easy modification of the commands.

For example, the user may have a command file named *run* which does his entire analysis. This command file might start with the line "log exe mydata" that will run the command file *mydata* that contains all information regarding to the data sets read, further data selection or binning etc. This could be followed by a second line in *run* like "log exe mymodel" that runs the command file *mymodel* which could contain the set-up for the spectral model and/or parameters. Also, often used plot settings (e.g. stacking of different plot types) could easily placed in separate command files.

In order to facilitate the readability of the command files, the user can put comment lines in the command files. Comment lines are recognized by the first character, that must be #. Also blank lines are allowed in the command file, in order to enhance (human) readability.

In general, all commands entered by the user are stored on the command file. Exceptions are the commands read from a nested command file (it is not necessary to save these commands, since they are already in the nested command file). Also, help calls and the command to open the file ("log save #a") are not stored.

Before saving, all commands are expanded to the full keywords, in those cases where abbrevated commands were used. However, for execution this is not important, since the interpreter can read both abbrevated and full keywords.

When a command file is read and the end of file is reached, the text "Normal end of command file encountered" is printed on the screen, and execution of the calling command file is resumed, or if the command file was opened from the terminal, control is passed over to the terminal again.

Finally, it is also possible to store all the output that is printed on the screen to a file. This is usefull for long sessions or for computer systems with a limited screen buffer. The output saved this way could be inspected later by (other programs of) the user. It is also usefull if SPEX is run in a kind of batch-mode.

2.5.2 Syntax

The following syntax rules apply for command files:

log exe #a - Execute the commands from the file #a. The suffix ".com" will be automatically appended to this filename.
log save #a [overwrite] [append] - Store all subsequent commands on the file #a. The suffix ".com" will be automatically appended to this filename. The optional argument "overwrite" will allow to overwrite an already existing file with the same name. The argument "append" indicates that if the file already exists, the new commands will be appended at the end of this file.
log close save - Close the current command file where commands are stored. No further commands will be written to this file.
log out #a [overwrite] [append] - Store all subsequent screen output on the file #a. The suffix ".out" will be automatically appended to this filename. The optional argument "overwrite" will allow to overwrite an already existing file with the same name. The argument "append" indicates that if the file already exists, the new output will be appended at the end of this file.
log close output - Close the current ascii file where screen output is stored. No further output will be written to this file.

2.5.3 Examples

`log save myrun` - writes all subsequent commands to a new file named "myrun.com". However, in case the file already exists, nothing is written but the user gets a warning instead.
`log save myrun append` - as above, but appends it to an existing file
`log save myrun overwrite` - as above, but now overwrites without warning any already existing file with the same name.
`log close save` - close the file where commands are stored.
`log exe myrun` - executes the commands in file myrun.com.
`log output myrun` - writes all subsequent output to file myrun.out.
`log close output` - closes the file above.

2.6 Distance: set the source distance

2.6.1 Overview

One of the main principles of SPEX is that spectral models are in principle calculated at the location of the X-ray source. Once the spectrum has been evaluated, the flux received at Earth can be calculated. In order to do that, the distance of the source must be set.
SPEX allows for the simultaneous analysis of multiple sky sectors. In each sector, a different spectral model might be set up, including a different distance. For example, a foreground object that coincides partially with the primary X-ray source has a different distance value.
The user can specify the distance in a number of different units. Allowed distance units are shown in the table below.

Table 2.1: SPEX distance units

Abbrevation	Unit
spex	internal SPEX units of 10^{22} m (this is the default)
m	meter
au	Astronomical Unit, $1.49597892\,10^{11}$ m
ly	lightyear, $9.46073047\,10^{15}$ m
pc	parsec, $3.085678\,10^{16}$ m
kpc	kpc, kiloparsec, $3.085678\,10^{19}$ m
mpc	Mpc, Megaparsec, $3.085678\,10^{22}$ m
z	redshift units for the given cosmological parameters
cz	recession velocity in km/s for the given cosmological parameters

The default unit of 10^{22} m is internally used in all calculations in SPEX. The reason is that with this scaling all calculations ranging from solar flares to clusters of galaxies can be done with single precision arithmetic, without causing underflow or overflow. For the last two units (z and cz), it is necessary to specify a cosmological model. Currently this model is simply described by H_0, Ω_m (matter density), Ω_Λ (cosmological constant related density), and Ω_r (radiation density). At startup, the values are:

H_0: 70 km/s/Mpc ,

Ω_m: 0.3 ,

Ω_Λ: 0.7 ,

Ω_r: 0.0

i.e. a flat model with cosmological constant. However, the user can specify other values of the cosmological parameters. Note that the distance is in this case the luminosity distance.
Note that the previous defaults for SPEX ($\mathrm{H}_0 = 50$, $q_0 = 0.5$) can be obtained by putting $\mathrm{H}_0 = 50$, $\Omega_m = 1$, $\Omega_\Lambda = 0$ and $\Omega_r = 0$.

WARNING: *when H_0 or any of the Ω is changed, the luminosity distance will not change, but the equivalent redshift of the source is adjusted. For example, setting the distance first to z=1 with the default H_0=70 km/s/Mpc results into a distance of* $2.039\,10^{26}$ *m. When H_0 is then changed to 100 km/s/Mpc, the distance is still* $2.168\,10^{26}$ *m, but the redshift is adjusted to 1.3342.*

WARNING: *In the output also the light travel time is given. This should not be confused with the (luminosity) distance in light years, which is simply calculated from the luminosity distance in m!*

2.6.2 Syntax

The following syntax rules apply to setting the distance:
distance [sector #i:] #r [#a] - set the distance to the value #r in the unit #a. This optional distance unit may be omittted. In that case it is assumed that the distance unit is the default SPEX unit of 10^{22} m. The distance is set for the sky sector range #i:. When the optional sector range is omitted, the distance is set for all sectors.
distance show - displays the distance in various units for all sectors.
distance h0 #r - sets the Hubble constant H_0 to the value #r.
distance om #r - sets the Ω_m parameter to the value #r.
distance ol #r - sets the Ω_Λ parameter to the value #r.
distance or #r - sets the Ω_r parameter to the value #r.

2.6.3 Examples

`distance 2` - sets the distance to 2 default units, i.e. to 2E22 m.
`distance 12.0 pc` - sets the distance for all sectors to 12 pc.
`distance sector 3 0.03 z` - sets the distance for sector 3 to a redshift of 0.03.
`distance sector 2 :  4 50 ly` - sets the distance for sectors 2-4 to 50 lightyear.
`distance h0 50.` - sets the Hubble constant to 50 km/s/Mpc.
`distance om 0.27` - sets the matter density parameter Ω_m to 0.27
`distance show` - displays the distances for all sectors, see the example below for the output format.

```
SPEX> di 100 mpc
 Distances assuming H0 =  70.0 km/s/Mpc, Omega_m = 0.300 Omega_Lambda = 0.700 Omega_r = 0.000
Sector        m       A.U.         ly         pc        kpc        Mpc  redshift         cz   age(yr)
----------------------------------------------------------------------------------------------------
   1 3.086E+24 2.063E+13 3.262E+08 1.000E+08 1.000E+05  100.0000    0.0229     6878.7 3.152E+08
----------------------------------------------------------------------------------------------------
```

2.7 Elim: set flux energy limits

2.7.1 Overview

SPEX offers the opportunity to calculate the model flux in a given energy interval for the current spectral model. This information can be viewed when the model parameters are shown (see section 2.25).
For each additive component of the model, the following quantities are listed:

the observed photon number flux (photons $\mathrm{m}^{-2}\,\mathrm{s}^{-1}$) at earth, including any effects of galactic absorption etc.

the observed energy flux ($\mathrm{W\,m}^{-2}$) at earth, including any effects of galactic absorption etc.

the intrinsic number of photons emitted at the source, not diluted by any absorption etc., in photons s^{-1}.

the intrinsic luminosity emitted at the source, not diluted by any absorption etc., in W.

The following units can be used to designate the energy or wavelength range: keV (the default), eV, Ryd, J, Hz, Å, nm.

2.7.2 Syntax

The following syntax rules apply:
elim #r1 #r2 [#a] - Determine the flux between #r1 #r2, in units given by #a, and as listed above. The default at startup is $2-10$ keV. If no unit is given, it is assumed that the limits are in keV.

WARNING: *When new units or limits are chosen, the spectrum must be re-evaluated (e.g. by giving the "calc" comand) in order to determine the new fluxes.*

WARNING: *The lower limit must be positive, and the upper limit must always be larger than the lower limit.*

2.7.3 Examples

`elim 0.5 4.5` - give fluxes between 0.5 and 4.5 keV
`elim 500 4500 ev` - give fluxes between 500 and 4500 eV (same result as above)
`elim 5 38 a` - give fluxes in the 5 to 38 Å wavelength band

2.8 Egrid: define model energy grids

2.8.1 Overview

SPEX operates essentially in two modes: with an observational data set (read using the data commands), or without data, i.e. theoretical model spectra. In the first case, the energy grid neede to evaluate the spectra is taken directly from the data set. In the second case, the user can choose his own energy grid.
The energy grid can be a linear grid, a logarithmic grid or an arbitrary grid read from an ascii-file. It is also possible to save the current energy grid, whatever that may be. In case of a linear or logarithmic grid, the lower and upper limit, as well as the number of bins or the step size must be given.
The following units can be used for the energy or wavelength: keV (the default), eV, Ryd, J, Hz, Å, nm.
When the energy grid is read or written from an ascii-file, the file must have the extension ".egr", and contains the bin boundaries in keV, starting from the lower limit of the first bin and ending with the upper limit for the last bin. Thus, the file has 1 entry more than the number of bins! In general, the energy grid must be increasing in energy and it is not allowed that two neighbouring boundaries have the same value.
Finally, the default energy grid at startup of SPEX is a logarithmic grid between 0.001 and 100 keV, with 8192 energy bins.

2.8.2 Syntax

The following syntax rules apply:
egrid lin #r1 #r2 #i [#a] - Create a linear energy grid between #r1 and #r2, in units given by #a (as listed above). If no unit is given, it is assumed that the limits are in keV. The number of energy bins is given by #i
egrid lin #r1 #r2 step #r3 [#a] - as above, but do not prescribe the number of bins, but the bin width #r3. In case the difference between upper and lower energy is not a multiple of the bin width, the upper boundary of the last bin will be taken as close as possible to the upper boundary (but cannot be equal to it).
egrid log #r1 #r2 #i [#a] - Create a logarithmic energy grid between #r1 and #r2, in units given by #a (as listed above). If no unit is given, it is assumed that the limits are in keV. The number of energy bins is given by #i
egrid log #r1 #r2 step #r3 [#a] - as above, but do not prescribe the number of bins, but the bin width (in log E) #r3.
egrid read #a - Read the energy grid from file #a.egr
egrid save #a - Save the current energy grid to file #a.egr

WARNING: *The lower limit of the energy grid must be positive, and the upper limit must always be larger than the lower limit.*

2.8.3 Examples

`egrid lin 5 38 step 0.02 a` - create a linear energy grid between $5-38$ Å, with a step size of 0.02 Å.
`egrid log 2 10 1000` - create a logarithmic energy grid with 1000 bins between $2-10$ keV.

`egrid log 2 10 1000 ev` - create a logarithmic energy grid with 1000 bins between 0.002 – 0.010 keV.
`egrid read mygrid` - read the energy grid from the file mygrid.egr.
`egrid save mygrid` - save the current energy grid to file mygrid.egr.

2.9 Abundance: standard abundances

2.9.1 Overview

For the plasm models, a default set of abundances is used. All abundances are calculated relative to those standard values. The current default abundance set is Anders & Grevesse ([1989]). The following abundances (Table 2.2) can be used in SPEX:

Table 2.2: Standard abundance sets

Abbrevation	Reference
reset	default (=Anders & Grevesse)
ag	Anders & Grevesse ([1989])
allen	Allen ([1973])
ra	Ross & Aller ([1976])
grevesse	Grevesse et al. ([1992])
gs	Grevesse & Sauval ([1998])

For the case of Grevesse & Sauval (1998) we adopted their meteoritic values (in general more accurate than the photospheric values, but in most cases consistent), except for He (slightly enhanced in the solar photosphere due to element migration at the bottom of the convection zone), C, N and O (largely escaped from meteorites) Ne and Ar (noble gases).
In Table 2.3 we show the values of the standard abundances. They are expressed in logarithmic units, with hydrogen by definition 12.0. For Allen ([1973]) the value for boron is just an upper limit.

WARNING: *For Allen ([1973]) the value for boron is just an upper limit.*

2.9.2 Syntax

The following syntax rules apply:
abundance #a - Set the standard abundances to the values of reference #a in the table above.

2.9.3 Examples

`abundance gs` - change the standard abundances to the set of Grevesse & Sauval ([1998])
`abundance reset` - reset the abundances to the standard set

Table 2.3: Abundances for the standard sets

Z	elem	AG	Allen	RA	Grevesse	GS
1	H	12.00	12.00	12.00	12.00	12.00
2	He	10.99	10.93	10.80	10.97	10.99
3	Li	1.16	0.70	1.00	1.16	3.31
4	Be	1.15	1.10	1.15	1.15	1.42
5	B	2.6	¡3.0	2.11	2.6	2.79
6	C	8.56	8.52	8.62	8.55	8.52
7	N	8.05	7.96	7.94	7.97	7.92
8	O	8.93	8.82	8.84	8.87	8.83
9	F	4.56	4.60	4.56	4.56	4.48
10	Ne	8.09	7.92	7.47	8.08	8.08
11	Na	6.33	6.25	6.28	6.33	6.32
12	Mg	7.58	7.42	7.59	7.58	7.58
13	Al	6.47	6.39	6.52	6.47	6.49
14	Si	7.55	7.52	7.65	7.55	7.56
15	P	5.45	5.52	5.50	5.45	5.56
16	S	7.21	7.20	7.20	7.21	7.20
17	Cl	5.5	5.60	5.50	5.5	5.28
18	Ar	6.56	6.90	6.01	6.52	6.40
19	K	5.12	4.95	5.16	5.12	5.13
20	Ca	6.36	6.30	6.35	6.36	6.35
21	Sc	1.10	1.22	1.04	3.20	3.10
22	Ti	4.99	5.13	5.05	5.02	4.94
23	V	4.00	4.40	4.02	4.00	4.02
24	Cr	5.67	5.85	5.71	5.67	5.69
25	Mn	5.39	5.40	5.42	5.39	5.53
26	Fe	7.67	7.60	7.50	7.51	7.50
27	Co	4.92	5.10	4.90	4.92	4.91
28	Ni	6.25	6.30	6.28	6.25	6.25
29	Cu	4.21	4.50	4.06	4.21	4.29
30	Zn	4.60	4.20	4.45	4.60	4.67

2.10 Ibal: set type of ionisation balance

2.10.1 Overview

For the plasma models, different ionisation balance calculations are possible.
Currently, the default set is Arnaud & Rothenflug ([1985]) for H, He, C, N, O, Ne, Na, Mg, Al, Si, S, Ar, Ca and Ni, with Arnaud & Raymond ([1992]) for Fe. Table 2.4 lists the possible options.

Table 2.4: Ionisation balance modes

Abbrevation	Reference
reset	default (=ar92)
ar85	Arnaud & Rothenflug ([1985])
ar92	Arnaud & Raymond ([1992]) for Fe,
	Arnaud & Rothenflug ([1985]) for the other elements

2.10.2 Syntax

The following syntax rules apply:
ibal #a - Set the ionisation balance to set #a with #a in the table above.

2.10.3 Examples

`ibal reset` - Take the standard ionisation balance
`ibal ar85` - Take the Arnaud & Rothenflug ionisation balance

2.11 Var: various settings for the plasma models

2.11.1 Overview

For the plasma models, there are several quantities that have useful default values but can be adjusted manually for specific purposes. We list them below.

Free-bound emission

Usually the freebound emission takes most of the computing time for the plasma models. This is because it needs to be evaluated for all energies, ions and atomic sublevels that are relevant. In rder to reduce the computational burden, there is a parameter `gfbacc` in SPEX that is approximately the accuracy level for the free-bound calculation. The default value is 10^{-3}. If this quantity is higher, less shells are taken into account and the computation proceeds faster (but less accurate). The users are advised not to change this parameter unless they are knowing what they do!

Line emission contributions

By default, all possible line emission processes are taken into account in the plasma models. For testing purposes, it is possible to include or exclude specific contributions. These are listed below in Table 2.5.

Table 2.5: Possible line emission processes

Abbrevation	Process
ex	electron excitation
rr	radiative recombination
dr	dielectronic recombination
ds	dielectronic recombination satellites
ii	inner shell ionisation

Doppler broadening

By default, thermal Doppler broadening is taken into account. However, the user can put this off for testing purposes.

Atomic data

The user can choose between the "old" Mekal code (the current default) and updated calculations for the ions for which this is possible.

Mekal code

We have made several minor improvements to the original Mekal plasma model. These improvements are included by default. However, it is possible to discard some of these improvements by going back to the old code. The improvements are classified as follows (with the appropriate syntax word added here).

Wav: wavelength corrections according to the work of Phillips et al. 1999, based upon Solar flare spectra; in addition, the 1s-np wavelengths of Ni XXVII and Ni XXVIII have been improved.
Fe17: The strongest Fe XVII lines are corrected in order to agree with the rates as calculated by Doron & Behar (2002).
Update: several minor corrections: the Si IX C7 line has been deleted; the Si VIII N6 line now only has the 319.83 Å line strength instead of the total triplet strength; the Ni XVIII and Fe XVI Na1A and NA1B lines plus their satellites have been deleted.

2.11.2 Syntax

The following syntax rules apply:
var gacc #r - Set the accuracy `gfbacc` for free-bound emission. Default is 10^{-3}, maximum value 1 and minimum value 0. Do not change if you do not know what it does.
var gacc reset - Reset `gfbacc` to its default value.
var line #a #l - For process #a (where #a is one of the abbreviations in Table 2.5) the process is allowed (if #l is true) or disabled (if #l is false). By default, all processes are allowed.
var line reset - Enable all line emission processes
var line show - Show the status of the line emission processses
var doppler #l - Include Doppler broadeing (the default) if #l is true, or exclude it if #l is false.
var calc old - Use the old Mekal code
var calc new - Use the new updated atomic data
var newmekal wav #l - if true (the default), use the updated wavelengths for the Mekal code
var newmekal fe17 #l - if true (the default), use the updated Fe XVII calculations for the Mekal code
var newmekal update #l - if true (the default), use the updates for some lines for the Mekal code
synvar newmekal all #lif true (default), use all the above three corrections for the Mekal code

2.11.3 Examples

`var gacc 0.01` - Set the accuracy `gfbacc` for free-bound emission.to 0.01
`var gacc reset` - Reset the accuracy `gfbacc` for free-bound emission.to its default value of 0.001
`var line ex f` - Exclude electron excitation
`var line ds t` - Include dielectronic satellites
`var line reset` - Include all line emission processes
`var line show` - Show status of all line emission proceses
`var doppler f` - Do not use thermal Doppler bvroadening
`var calc new` - Use the new atomic data
`var newmekal wav f` - Use the original Mekal wavelengths instead
`var newmekal fe17 t` - Use the updated Fe XVII calculations
`var newmekal all f` - Go back to the full old Mekal code
`var newmekal all t` - Take the full updated Mekal code

2.12 Ion: select ions for the plasma models

2.12.1 Overview

For the plasma models, it is possible to include or exclude specific groups of ions from the line calulations. This is helpful if a better physical understanding of the (atomic) physics behind the spectrum is requested.
Currently these settings *only* affect the line emission; in the calculation of the ionisation balance as well as the continuum always all ions are taken into account (unless of course the abundance is put to zero).

2.12.2 Syntax

The following syntax rules apply:
ions show - Display the list of ions currently taken into account
ions use all - Use all possible ions in the line spectrum
ions use iso #i: - Use ions of the isoelectronic sequences indicated by #i: in the line spectrum
ions use z #i: - Use ions with the atomic numbers indicated by #i: in the line spectrum
ions use ion #i1 #i2: - Use ions with the atomic number indicated by #i1 and ionisation stage indicated by #i2: in the line spectrum
ions ignore all - Ignore all possible ions in the line spectrum
ions ignore iso #i: - Ignore ions of the isoelectronic sequences indicated by #i: in the line spectrum
ions ignore z #i: - Ignore ions with the atomic numbers indicated by #i: in the line spectrum
ions ignore ion #i1 #i2: - Ignore ions with the atomic number indicated by #i1 and ionisation stage indicated by #i2: in the line spectrum

2.12.3 Examples

`ions ignore all` - Do not take any line calculation into account
`ions use iso 3` - Use ions from the $Z = 3$ (Li) isoelectronic sequence
`ions use iso 1:2` - Use ions from the H-like and He-like isoelectronic sequences
`ions ignore z 26` - Ignore all iron ($Z = 26$) ions
`ions use 6 5:6` - Use C V to C VI
`ions show` - Display the list of ions that are used

2.13 Data: read response file and spectrum

2.13.1 Overview

In order to fit an observed spectrum, SPEX needs a spectral data file and a response matrix. These data are stored in FITS format, tailored for SPEX (see section 5.7). The data files need not necessarily be located in the same directory, one can also give a pathname plus filename in this command.

2.13.2 Syntax

The following syntax rules apply: data #a1 #a2 - Read response matrix #a1 and spectrum #a2
data delete instrument #i - Remove instrument #i from the data set
data merge sum #i: - Merge instruments in range #i: to a single spectrum and response matrix, by adding the data and matrices
data merge aver #i: - Merge instruments in range #i: to a single spectrum and response matrix, by averaging the data and matrices
data save #i #a [overwrite] - Save data #a from instrument #i with the option to overwrite the existent file. SPEX automatically tags the .spo extension to the given file name. No response file is saved.
data show - Shows the data input given, as well as the count (rates) for source and background, the energy range for which there is data the response groups and data channels. Also the integration time and the standard plotting options are given.

2.13.3 Examples

`data mosresp mosspec` - read a spectral response file named mosresp.res and the corresponding spectral file mosspec.spo. Hint, although 2 different names are used here for ease of understanding, it is eased if the spectrum and response file have the same name, with the appropriate extension.
`data delete instrument 1` - delete the first instrument
`data merge aver 3:5` - merge instruments 3–5 into a single new instrument 3 (replacing the old instrument 3), by averaging the spectra. Spectra 3–5 could be spectra taken with the same instrument at different epochs.
`data merge sum 1:2` - add spectra of instruments 1–2 into a single new instrument 1 (replacing the old instrument 1), by adding the spectra. Useful for example in combining XMM-Newton MOS1 and MOS2 spectra.
`data save 1 mydata` - Saves the data from instrument 1 in the working directory under the filename of mydata.spo
`data /mydir/data/mosresp /mydir/data/mosspec` - read the spectrum and response from the directory /mydir/data/

2.14 Ignore: ignoring part of the spectrum

2.14.1 Overview

If one wants to ignore part of a data set in fitting the input model, as well as in plotting this command should be used. The spectral range one wants to ignore can be specified as a range in data channels or a range in wavelength or energy. Note that the first number in the range must always be smaller or equal to the second number given. If multiple instruments are used, one must specify the instrument as well. If the data set contains multiple regions, one must specify the region as well. So per instrument/region one needs to specify which range to ignore. The standard unit chosen for the range of data to ignore is data channels. To undo ignore, see the use command (see section 2.15).

The range to be ignored can be specified either as a channel range (no units required) or in eiter any of the following units: keV, eV, Rydberg, Joule, Hertz, Å, nanometer, with the following abbrevations: kev, ev, ryd, j, hz, ang, nm.

2.14.2 Syntax

The following syntax rules apply:

ignore [instrument #i1] [region #i2] #r - Ignore a certain range #r given in data channels of instrument #i1 and region #i2. The instrument and region need to be specified if there are more than 1 data sets in one instrument data set or if there are more than 1 data set from different instruments.

ignore [instrument #i1] [region #i2] #r unit #a - Same as the above, but now one also specifies the unites #a in which the range #r of data points to be ignored are given. The units can be either Å(ang) or (k)eV.

2.14.3 Examples

`ignore 1000:1500` - Ignores data channels 1000 till 1500.

`ignore region 1 1000:1500` - Ignore data channels 1000 till 1500 for region 1.

`ignore instrument 1 region 1 1000:1500` - Ignores the data channels 1000 till 1500 of region 1 from the first instrument.

`ignore instrument 1 region 1 1:8 unit ang` - Same as the above example, but now the range is specified in units of Å instead of in data channels.

`ignore 1:8 unit ang` - Ignores the data from 1 to 8 Å, only works if there is only one instrument and one region included in the data sets.

2.15 Use: reuse part of the spectrum

2.15.1 Overview

This command can only be used after the ignore command was used to block out part of the data set in the fitting. The command re-includes thus (part of) the data set for fitting. As it undoes what the ignore command did (see section 2.14) the syntax and use of both are very similar. Again one can/must specify the instrument and region for one wants the data to be re-included. Further can one specify the units chosen to give the range of data points to be included. The standard unit is that of data channels and does not need to be specified. The data points re-included with use will automatically also be plotted again.
The range to be reused can be specified either as a channel range (no units required) or in eiter any of the following units: keV, eV, Rydberg, Joule, Hertz, Å, nanometer, with the following abbrevations: kev, ev, ryd, j, hz, ang, nm.

2.15.2 Syntax

The following syntax rules apply:
use [instrument #i1] [region #i2] #r - Use the range #r in fitting the data. The instrument #i1 and the region #i2 must be specified if either there are more than 1 instrument or more than 1 region used in the data sets.
use [instrument #i1] [region #i2] #r unit #a - Same as the above, but now the units #a in which the range #r of data points to be used is specified in any of the units mentioned above.

2.15.3 Examples

`use 1000:1500` - Re-include the data channels 1000:1500 for fitting the data.
`use instrument 1 region 1 1000:1500` - Re-include the data channels 1000:1500 of the first instrument and the first region in the data set for fitting the data and plotting.
`use instrument 1 region 1 1:1.5 unit kev` - Same as the above, but now the unites are specified as keV instead of in data channels.
`use 1000:1500 unit ev` - Re-iclude the data points between 1000 and 1500 eV for fitting and plotting. Note that in this case the input data should come from only one instrument and should contain no regions.

2.16 Bin: rebin the spectrum

2.16.1 Overview

This command rebins the data (thus both the spectrum file and the response file) in a manner as described in Sect. 2.13. The range to be rebinned can be specified either as a channel range (no units required) or in eiter any of the following units: keV, eV, Rydberg, Joule, Hertz, Å, nanometer, with the following abbrevations: kev, ev, ryd, j, hz, ang, nm.

2.16.2 Syntax

The following syntax rules apply:
bin #r #i - This is the simplest syntax allowed. One needs to give the range, #r, over at least the input data channels one wants to rebin. If one wants to rebin the whole input file the range must be at least the whole range over data channels (but a greater number is also allowed). #i is then the factor by which the data will be rebinned.
bin [instrument #i1] [region #i2] #r #i - Here one can also specify the instrument and region to be used in the binning. This syntax is necessary if multiple instruments or regions are used in the data input.
bin [instrument #i1] [region #i2] #r #i [unit #a] - In addition to the above here one can also specify the units in which the binning range is given. The units can be eV, Å, or any of the other units specified above.

2.16.3 Examples

`bin 1:10000 10` - Rebins the input data channel 1:10000 by a factor of 10.
`bin instrument 1 1:10000 10` - Rebins the data from the first instrument as above.
`bin 1:40 10 unit a` - Rebins the input data between 1 and 40 Å by a factor of 10.

2.17 Obin: optimal rebinning of the data

2.17.1 Overview

This command rebins (a part of) the data (thus both the spectrum and the response) to the optimal bin size given the statistics of the source as well as the instrumental resolution. This is recommended to do in **all** cases, in order to avoid oversampling of the data. The theory and algorithms used for this rebinning ared escribed in detail in Sect. **??**. A simple cartoon of this is: binning to 1/3 of the FWHM, but the factor of 1/3 depends weakly upon the local count rate at the given energy and the number of resolution elements. The better the statistics, the smaller the bin size.

2.17.2 Syntax

The following syntax rules apply:
obin #i1: - Simplest command allowed. #i1: is the range in data channels over which the binning needs to take place.
obin #r1: #i: unit #a - The same command as above, except that now the ranges over which the data is to be binned (#r1:) are specified in units (#a) different from data channels. These units can be eV, keV, Å, as well as in units of Rydberg (ryd), Joules (j), Hertz (hz) and nanometers (nm).
obin [instrument #i1:] [region #i2:] #i3: - Here #i3: is the same as #i1: in the first command. However, here one can specify the instrument range #i1: and the region range #i2: as well, so that the binning is done only for one given data set.
obin [instrument #i1:] [region #i2:] #r1: [unit #a] - This command is the same as the above, except that here one can specify the range over which the binning should occur in the units specified by #a. These units can be eV, Å, keV, as well as in units of Rydberg (ryd), Joules (j), Hertz (hz) and nanometers (nm).

2.17.3 Examples

`obin 1:10000` - Optimally bins the data channels 1:10000.
`obin 1:4000 unit ev` - Does the same as the above, but now the data range to be binned is given in eV, from 1−4000 eV, instead of in data channels.
`obin instrument 1 region 1 1:19 unit a` - Bins the data from instrument 1 and region 1 between 1 and 19 Å.

2.18 Vbin: variable rebinning of the data

2.18.1 Overview

This command rebins (a part of) the data (thus both the spectrum and the response) such that the signal to noise ratio in that part of the spectrum is at least a given constant. Thus instead of taking a fixed number of data channels for binning, the number of data channels binned is here variable as is the bin width. On top of specifying the S/N ratio, you also specify the minimal bin width. This is useful for rebinning low count data, which have no strong or sharp features or lines in it.

WARNING: *For high resolution spectra with sharp and strong lines, this binning can lead to very wrong results. In this case either the emission lines or the continuum (in case of absorption lines) have a much higher signal to noise ratio than the other component. As a result this function will try to bin the line and continuum together, resulting in a smoothing out of the lines.*

WARNING: *It is not always guaranteed that the minimum signal-to-noise ratio is obtained in all channels. This is an effect of the applied algorithm. Channels with the highest S/N ratio and neighboring bins are merged until sufficient S/N ratio is obtained. This process is continued for the remaining number of bins. At the end of the process a few bins with a low S/N ratio will remain. These are merged with their neighbors, resulting in a possibly lower S/N ratio for that bin.*

2.18.2 Syntax

The following syntax rules apply:
vbin #i1: #i2 #r - Simplest command allowed. #i1: is the range in data channels over which the binning needs to take place. #i2 in the minimum amount of data channels that need to be binned, and #r is the minimal S/N ratio one wants the complete data set to have.
vbin #r1: #i #r2 unit #a - The same command as above, except that now the ranges over which the data is to be binned (#r1:) are specified in units (#a) different from data channels. These units can be eV, keV, Å, as well as in units of Rydberg (ryd), Joules (j), Hertz (hz) and nanometers (nm).
vbin [instrument #i1:] [region #i2:] #i3: #i4 #r - Here #i3: and #i4 are the same as #i1: and #i2 in the first command, also #r is the minimal S/N ratio required. However, here one can specify the instrument range #i1: and the region range #i2: as well, so that the binning only is performed for a certain data set.
vbin [instrument #i1:] [region #i2:] #r1: #i2 #r2 [unit #a] - This command is the same as the above, except that here one can specify the range over which the binning should occur in the units specified by #a. These units can be eV, Å, keV, as well as in units of Rydberg (ryd), Joules (j), Hertz (hz) and nanometers (nm).

2.18.3 Examples

`vbin 1:10000 3 10` - Bins the data channels 1:10000 with a minimal bin width of 3 data channels, and a minimum S/N ratio of 10.
`vbin 1:4000 3 10 unit ev` - Does the same as the above, but now the data range to be binned is given in eV, from 1−4000 eV, instead of in data channels.

`vbin instrument 1 region 1 1:19 3 10 unit a` - Bins the data from instrument 1 and region 1 between 1 and 19 Å with the same minimum bin width and S/N as above.

2.19 Syserr: systematic errors

2.19.1 Overview

This command calculates a new error, adding the systematic error of the source and the background to the Poissonian error in quadrature. One must specify both the systematic error as a fraction of the source spectrum as well as of the subtracted background spectrum. The total of these fractions can either be less or greater than 1.

WARNING: *One should first rebin the data, before running syserr. Run syserr however before fitting the data or finding errors on the fit.* WARNING: *Running syserr multiple times will increase the error every time. If the input to syserr is wrong one should restart SPEX and rerun syserr with the correct values to calculate the total error correctly.*

2.19.2 Syntax

The following syntax rules apply:
syserr #i: #r1 #r2 - The shortest version of this command. #i: is the range in data channels for which the systematic error is to be calculated and added (in quadrature) to the Poissonian error. #r1 is then the the relative systematic error due to the source and #r2 the relative systematic error due to the background.
syserr [instrument #i1:] [region #i2:] #i3: #r1 #r2 - In this syntax one can also specify the instrument and the region one wants to calculate the combined error for. Both can be ranges as well. #i3: has the same role as #i: in the above command, and #r1 and #r2 are the same as above.
syserr [instrument #i1:] [region #i2:] #i3: #r1 #r2 [unit #a] - Exact same command as above, except that now the data range (#i3:) for which the errors are to be calculated are given in units different than data channels. These units can be Å (ang), eV (ev), keV (kev), Rydbergs (ryd), Joules (j), Hertz (hz) and nanometers (nm). This is the most general command.

2.19.3 Examples

`syserr 1:100000 0.3 0.5` - Calculates the combined Poissonian and systematic error for data channels 1:100000, where the fraction of the systematic error of the source is 0.3 and the background is 0.5.
`syserr 0:2000 0.3 0.5 unit ev` - The same as the above command, expect that now the error calculation is performed between 0 and 2000 eV instead of data channels.
`syserr instrument 2 region 1 0:2000 0.3 0.5 unit ev` - The same as the above command, but now the error calculation is only performed for the data set from the second instrument and the first region thereof.

2.20 Multiply: scaling of the response matrix

2.20.1 Overview

This command multiplies (a component of) the response matrix by a constant.

WARNING: *If this command is repeated for the same component then the original response matrix is changed by the multiplication of the constants. For example, after multiplying the response matrix by a factor of 2, the original matrix is recovered by multiplying the intermediate result by 0.5.*

WARNING: *The instrument number must be given in this command even if you use only a single instrument.*

2.20.2 Syntax

The following syntax rules apply:
multiply #i1 [component #i2] #r - Multiplies the response matrix of component #i2 of instrument #i1 by a constant #r.

2.20.3 Examples

`multiply 1 3.5` - Multiplies the response matrix from instrument 1 by a factor of 3.5.
`multiply 1 component 2 3.5` - Multiplies the second component of instrument 1 by the constant 3.5.

2.21 Shiftplot: shift the plotted spectrum for display purposes

2.21.1 Overview

This command shifts the observed spectrum as well as the model spectrum by adding a constant or by multiplying a constant, as far as plots are concerned. The true data files do not change, so the shift is only for representational purposes.
There are basically two options, indicated by the mode variable *plotshift*: For *plotshift*=1, a constant *shift* is added to the plotted spectrum of a given part of the data, for *plotshift*=2 the plotted spectrum is multiplied by the constant *shift*.
The multiplicative constant shift (*plotshift*=2) is generally preferred for log-log plots, while for linear plots a constant additive shift (*plotshift*=1) is preferred.

WARNING: *In the case of addition (plotshift=1), the addition constant is given in counts/s. This thus leads to a different (energy-dependent) constant being added to the plotted spectrum if the units are not in counts/s. For the multiplicative case this is of course not the case.*

2.21.2 Syntax

The following syntax rules apply:
shiftplot #i #r - #i indicates whether a constant should be added (#i=1) or multiplied (#i=2) to the spectrum and model. #r is then the constant to be added or multiplied by.

shiftplot [instrument #i1:] [region #i2:] #i3 #r - Shift the plot using a factor #r. Here #i3 (= *plotshift*) determines whether the constant is added (*plotshift=1*) or multiplied (*plotshift=2*). The optional instrument range #i1: can be used to select a single instrument or range of instruments for which the shift needs to be applied, and the optional region range #i2: the region for these instruments.

2.21.3 Examples

`shiftplot 1 2.0` - Adds 2 counts/s to the plotted spectrum. Since no instrument or region is specified, this applies for all spectra.
`shiftplot 2 10.0` - Multiplies the plotted spectrum by a factor of 10.0
`shiftplot instrument 1 region 2 1 2` - Adds 2 counts/s to the plotted spectrum for the data from instrument 1 and region 2.
`shiftplot region 2:3 1 2` - Adds 2 counts/s to the plotted spectrum for the data for all instruments and regions 2−3.

2.22 Sector: creating, copying and deleting of a sector

2.22.1 Overview

This allows one to create, delete, copy and show the number of sectors, used for the analysis of the data. A sector is a region on the sky with its own spectral model, or a part of the lightcurve of a variable source with its time dependent spectrum, etc. See for more details about sectors and regions section 1.2. For doing spectral fitting of data sets, the sectors need to be specified in the response matrix of the data: the response file should tell which sector number corresponds to a given part of the matrix.

2.22.2 Syntax

The following syntax rules apply:
sector new - Creates a new sector, which can have its own model.
sector show - Gives the number of sectors that are currently used.
sector copy #i - Copies the model for sector #i to a new sector.
sector delete #i - Deletes sector #i.

2.22.3 Examples

`sector new` - Creates a new sector.
`sector copy 2` - Creates a new sector, with the same spectral model as used in sector 2. This can be useful if the spectra of the different sectors are very similar in composition.
`sector delete 3` - Deletes sector number 3.

2.23 Comp: create, delete and relate spectral components

2.23.1 Overview

In fitting or evaluating a spectrum, one needs to build up a model made out of at least 1 component. This set of commands can create a new component in the model, as well as delete any component. Usually we distinguish two types of spectral components in SPEX.
The *additive* components correspond to emission components, such as a power law, a Gaussian emission line, a collisional ionization equilibrium (CIE) component, etc.
The second class (dubbed here *multiplicative* components for ease) consists of operations to be applied to the additive components. Examples are truly multiplicative operations, such as the Galactic absorption, where the model spectrum of any additive component should be multiplied by the transmission of the absorbing interstellar medium, warm absorbers etc. Other operations contained in this class are redshifts, convolutions with certain velocity profiles, etc.
The user needs to define in SPEX which multiplicative component should be applied to which additive components, and in which order. The order is important as operations are not always communative. This setting is also done with this component command.
If multiple sectors are present in the spectral model or response matrix (see Sect. 1.2) then one has to specify the spectral components and their relation for each sector. The possible components to the model are listed and described in Sect. 4.
Note that the order that you define the components is not important. However, for each sector, the components are numbered starting from 1, and these numbers should be used when relating the multiplicative components to the additive components.
If you want to see the model components and the way they are related, type "model show".

WARNING: *If in any of the commands as listed above you omit the sector number or sector number range, the operation will be done for all sectors that are present. For example, having 3 sectors, the "comp pow" command will define a power law component for each of the three sectors. If you only want to define/delete/relate the component for one sector, you should specify the sector number(s). In the very common case that you have only one sector you can always omit the sector numbers.*

WARNING: *After deleting a component, all components are re-numbered! So if you have components 1,2,3 for example as pow, cie, gaus, and you type "comp del 2", you are left with 1=pow, 2=gaus.*

2.23.2 Syntax

The following syntax rules apply:
comp [#i:] #a - Creates a component #a as part of the model for the (optional) sector range #i:
comp delete [#i1:] #i2: - Deletes the components with number from range #i2: for sector range (optional) #i1. See also the warning above
comp relation [#i1:] #i2: #i3,...,#in - Apply multiplicative components #i3, . . . , #in (numbers) in this order, to the additive components given in the range #i2: of sectors in the range #i1 (optional). Note that the multiplicative components must be separated by a ","

2.23.3 Examples

`comp pow` - Creates a power-law component for modeling the spectrum for all sectors that are present.
`comp 2 pow` - Same as above, but now the component is created for sector 2.
`comp 4:6 pow` - Create the power law for sectors 4, 5 and 6
`com abs` - Creates a Morrison & McCammon absorption component.
`comp delete 2` - Deletes the second created component. For example, if you have 1 = pow, 2 = cie and 3 = gaus, this command delets the cie component and renumbers 1 = pow, 2 = gaus
`comp del 1:2` - In the example above, this will delete the pow and cie components and renumbers now 1 = gaus
`comp del 4:6 2` - If the above three component model (pow, cie, gaus) would be defined for 8 sectors (numbered 1–8), then this command deletes the cie component (nr. 2) for sectors 4–6 only.
`comp rel 1 2` - Apply component 2 to component 1. For example, if you have defined before with "comp pow" and "comp "abs" a power law and galactic absorption, the this command tells you to apply component 2 (abs) to component 1 (pow).
`comp rel 1 5,3,4` - Taking component 1 a power law (pow), component 3 a redshift operation (reds), component 4 galactic absorption (abs) and component 5 a warm absorber (warm), this command has the effect that the power law spectrum is multiplied first by the transmission of the warm absorber (5=warm), then redshifted (3=reds) and finally multiplied by the transmission of our galaxy (4=abs). Note that the order is always from the source to the observer!
`comp rel 1:2 5,3,4` - Taking component 1 a power law (pow), component 2 a gaussian line (gaus), and 3–5 as above, this model applies multiplicative components 5, 3 and 4 (in that orer) to the emission spectra of both component 1 (pow) and 2 (cie).
`comp rel 7:8 1:2 5,3,4` - As above, but only for sectors 7 and 8 (if those are defined)

2.24 Model: show the current spectral model

2.24.1 Overview

This commands prints the current spectral model, for each sector, to the screen. The model is the set of spectral components that is used, including all additive and multiplicative components. For all additive components, it shows in which order the multiplicative components are applied to the additive (emitted) components. See Sect. 2.23 for more details.

2.24.2 Syntax

The following syntax rules apply:
model show - Prints the model for all sectors to the screen.
model show #i - Prints the model for sector #i to the screen.

2.24.3 Examples

`model show 2` - Prints the model for the second sector

2.25 Par: Input and output of model parameters

2.25.1 Overview

This command is used as an interface to set or display the parameters of the spectral model. Each model parameter (like temperature T, abundance, normalization etc.) has a set of attributes, namely its value, status, range and coupling. This is illustrated by the following example. Assume we have a spectral model consisting of a thermal plasma. Consider the silicon abundance (acronym in the model: 14). It can have a value (for example 2.0, meaning twice the solar abundance). Its status is a logical variable, true (thawn) if the parameter can be adjusted during the spectral fitting proces or false (frozen) if it should be kept fixed during the spectral fit. The range is the allowed range that the parameter can have during the spectral fit, for example 0–1000 (minimum – maximum values). This is useful to set to constrain a spectral search to a priori "reasonable" values (e.g. abundances should be non-negative) or "safe" values (SPEX could crash for negative temperatures for example). Finally the coupling allows you to couple parameters, for example if the Si abundance is coupled to the Fe abundance, a change in the Fe abundance (either defined by the user or in a spectral fitting process) will result automatically in an adjustment of the Si abundance. This is a useful property in practical cases where for example a spectrum has strong Fe lines and weaker lines from other species, but definitely non-solar abundances. In this case, one may fit the Fe abundance, but the statistics for the other elements may not be sufficient to determine their abundance accurately; however a priori insight might lead you to think that it is the global metallicity that is enhanced, and therefore you would expect the Si abundance to be enhanced in the same way as Fe.
With the par command, you can set for each parameter individually or for a range of parameters in the same command, their attributes (value, status, range and coupling). Also, you can display the parameters on the screen or write them on a SPEX command file, which allows you to start another run with the current set of parameters.
When setting parameters, you can also specify the sector (range) and component (range). For your first call, it is assumed that you refer to sector 1, component 1, first parameter. In all subsequent calls of the parameter command, it is assumed that you refer to the same sector(s) and component(s) as your last call, unless specified differently. This is illustrated by the following example. Suppose you have the following model: power law (component 1), blackbody (component 2) and RGS line broadening (lpro, component 3). If you type in the commands in that order, this is what happens:

- par val 2 – sets the norm (= first parameter) of the power law to value 2
- par gam val 3 – sets the photon index of the power law to value 3
- par 2 t val 2.5 – sets the temperature of the blackbody to 2.5 keV
- par norm val 10 – sets the norm of the blackbody to 10
- par 1:2 norm v 5 – sets the norm of both the PL and BB to 5
- par val 4.2 – sets the norm of both the PL and BB to 4.2
- par 3 file myprofile.dat – sets for the LPRO component the file name with the broadening kernal to myprofile.dat

2.25.2 Syntax

The following syntax rules apply:
par [#i1:] [#i2:] [#a1:] avalue #a2 - Assign the value #a2 to the parameters specified by the (range of) name(s) #a1 of component range #i2: of sectors #i1:. The sector (range), component (range) and parameter name (range) are optional. If #a2 should be an empty text string, specify the value "none" (without quotes, and all four characters in lowercase). If thei are not specified, the sector(s), component(s) and parameter(s) of the last call are used. This command holds for input of text strings. For real numbers, use the following command (with "avalue" replaced by "value" and #a2
par [#i1:] [#i2:] [#a:] value #r - Assign the value #r to the parameters specified by the (range of) name(s) #a of component range #i2: of sectors #i1:. The sector (range), component (range) and parameter name (range) are optional. If not specified, the sector(s), component(s) and parameter(s) of the last call are used.
par [#i1:] [#i2:] [#a:] status #l - As above but for the status of a (range of) parameters; #l = T (true, thawn) means a parameter that can be adjusted during a spectral fit, #l = F (false, froze, fixed) means a parameter that will remain fixed during the spectral fitting.
par [#i1:] [#i2:] [#a:] range #r1 #r2 - As above but fore the allowed range of the parameter. #r1 denotes the lower limit and #r2 denotes the upper limit. Both should be specified.
par [#i1:] [#i2:] [#a1:] couple [#i3:] [#i4:] #a2: - Here #i1:, #i2:, #a1: are the sector(s), component(s) and parameter(s) that are to be coupled. The parameter (s) to which they are coupled are specified by #a2: and the optional #i3:, #i4:. If ranges are used, take care that the number of sectors, components and parameters on the right and left match.
par [#i1:] [#i2:] [#a1:] couple [#i3:] [#i4:] #a2: factor #r - As above, but couple using a scaling factor #r.
par [#i1:] [#i2:] #a: decouple - Decouples the parameter(s) #a: of (optional) components #i2: of of (optional) sector(s) #i1:. Inverse operation of the couple command.
par show [free] - Shows on the display screen all parameters of all components. If the optional keyword free is given, shows only parameters with fit status "T" (true, thawn), i.e., the free parameters in a spectral fit.
par write #a [overwrite] - Writes all parameters to a SPEX command file #a. #a should be specified without extension, SPEX automatically adds the .com extension. If the optional overwrite command is given, then the file #a will be overwritten with the new values.

2.25.3 Examples

`par val 10` - Sets the value of the current parameter to 10.
`par t val 4` - Sets the parameter named "t" to 4.
`par 06:28 value 0.3` - Sets the parameters named 06:28 (for example the abundances of C ($Z = 6$), N ($Z = 7$), ... Ni ($Z = 28$)) to the value 0.3.
`par 2 nh val 0.01` - Sets the parameter named "nh" of component 2 to the value 0.01
`par 1 1 norm value 1E8` - Sets parameter named "norm" of component 1 of sector 1 to the value 10^8.
`par file avalue myfile.with.data` - sets the ascii-type parameter named "file" to the value "myfile.with.data" (without quotes).
`par file avalue none` - sets the ascii-type parameter named "file" to the value "" (i.e. an empty string)

`par status frozen` - Sets the fit status of the current parameter to "frozen" (fixed).
`par 1 3 t stat thawn` - Specifies that parameter "t" of the third component of the model for sector 1 should be left free (thawn) during spectral fitting.
`par 2 3 gamm range 1.6 1.9` - Limit parameter "gamm" of component 3 of sector 2 to the range 1.6 – 1.9.
`par norm range -1E8 1E8` - Set the range for the parameter "norm" between -10^8 and $+10^8$. This command is necessary if for example the model is a delta line used to mimick an absorption line, which normally has a default minimum value of 0. (for an emission line).
`par 1 1 norm couple 2 1 norm` - Couples the norm of the first component for the first sector to the norm of the first component of the model for the second sector.
`par 1 1 norm couple 2 1 norm factor 3` - The same command as the above, but now the norm of the first component in the model for sector 1 is 3 times the norm of the first component of the model for sector 2. For example, if the norm of the 1st component of sector 2 gets the value 40, then the norm of the 1st component of sector 2 will automatically be updated to a value of $3 \times 40 = 120$.
`par 3:5 02:30 couple 1 02:30` - Couples the abundances of He–Zn of components 3, 4 and 5 to the He–Zn abundances of component 1.
`par norm decouple` - Decouples the norm of the current component from whatever it was coupled to.
`par show` - Shows all the parameters of the current model for all sectors and how they are coupled (if applicable). For each parameter it shows the value, status, range and coupling information, as well as info on its units etc. It also shows the fluxes and restframe luminosities of the additive components, photon flux (phot/m**2/s) and energy flux (W/m**2) are the values observed at Earth (including any transmission effects like Galactic absorption or intrinsic absoprtion that have been taken into account), and the nr. of photons (photons/s) and luminosity (W) are all as emitted by the source, without any attenuation.
`par show free` - As above, but only displays the parameters that have the status thawn.
`par write mypar overwrite` - SPEX writes all parameters for the model to a file named mypar.com. Any previously existing file mypar.com is overwritten.
`par write mypar` - Same command as above, but now a new file is created. If there already exists a file with the same filename SPEX will give an error message.

2.26 Simulate: Simulation of data

2.26.1 Overview

This command is used for spectral simulations. The user should start with a spectral model and a spectral data set (both matrix and spectrum). After giving the "simulate" command, the observed spectrum will be replaced by the simulated spectrum in SPEX. Note that the original spectrum file (with the .spo extension) is not overwritten by this command, so no fear to destroy your input data! Different options exist and several parameters can be set:

- Instrument, region: the instrument(s) and region(s) for which the simulation should be done (i.e., if you have more instruments you can do the simulation only for one or a few instruments)
- time: set the exposure time t (s) of the source spectrum as well as the background error scale factor f_b. This last option allows you to see what happens if for example the background would have a ten times higher accuracy (for $f_b = 0.1$).
- syserr: add a systematic error to both the source and background spectrum. An alternative way to introduce systematic errors is of course to use the syserr command (Sect. 2.19). Take care not to set the systematic errors twice, and remember that rebinning your spectrum later will reduce the systematic errors, as these will be added in quadrature to the statistical errors. So first rebin and then add systematics!
- noise: either randomize your data or just calculate the expected values
- back: do the background subtraction or keep it in the data

WARNING: *A response matrix and spectrum of the region and the instrument you want to simulate are necessary, as SPEX needs the response matrix as well as the background to be subtracted for the simulations.*

WARNING: *If your background is taken from the same observation as your source, and you multiply the original exposure time with a factor of S, you should put f_b to $S^{-0.5}$, reflecting the fact that with increasing source statistics the background statistics also improves. This is the case for an imaging observation where a part of the image is used to determine the background. If instead you use a deep field to subtract the background, then the exposure time of your background will probably not change and you can safely put $f_b = 1$ for any exposure time t.*

WARNING: *If your subtracted background (one of the columns in the .spo file) is derived from a low statistics Poissonian variable (for example a measured count rate with few counts per channel), then scaling the background is slightly inaccurate as it takes the observed instead of expected number of background counts as the starting point for the simulation.*

2.26.2 Syntax

The following syntax rules apply:
simulate [instrument #i1:] [region #i2:] [time #r1 #r2] [syserr #r3 #r4] [noise #l1] [back #l2]
- #i1: and #i2: specify the instrument and region (range) to be used in the simulation. #r1 is the exposure time t, #r2 the background error scale factor f_b. If omitted, t will be taken as the exposure time of the current data set, and $f_b = 1$. #r2 must always be specified if t is

specified. In case of doubt, put $f_b = 1$. #r3 and #r4 are the systematic errors as a fraction of the source and background spectrum, respectively; again both should be specified or omitted together. Default values are 0. If #l1 is true, Poissonian noise will be added (this is the default). If #l2 is true, the background will be subtracted (this is the default).

2.26.3 Examples

`simulate` - This simulates a new spectrum/dataset with Poissonian noise and background subtraction with the same exposure time and background as the original data set.
`simulate noise f` - As above, but without Poissonian noise. The nominal error bars are still plotted.
`simulate back f` - As first example, but now no background is subtracted.
`simulate time 10000 1.` - Simulate for 10^4 s exposure time and the same background as the template spectrum.
`simulate time 10000 0.2` - Simulate for 10^4 s exposure time but with a 5 times more accurate subtracted background (the background errors are 5 times smaller).
`simulate syserr 0.1 0.2` - As before but with a systematic error of 10 % of the source spectrum and 20 % of the subtracted background spectrum added in quadrature.
`simulate instrument 2:4 region 3 time 5000 1 syserr 0.05 0.01 noise f back f` - Simulate a spectrum for region 3 of instruments 2–4, with 5000 s exposure time t, the nominal background error scaleng $f_b = 1$, a systematic error of 5 % of the source and 1 % of the background, no Poissonian noise and no background subtraction.

2.27 Calculate: evaluate the spectrum

2.27.1 Overview

This command evaluates the current model spectrum. When one or more instruments are present, it also calculates the model folded through the instrument. Whenever the user has modified the model or its parameters manually, and wants to plot the spectrum or display model parameters like the flux in a given energy band, this command should be executed first (otherwise the spectrum is not updated). On the other hand, if a spectral fit is done (by typing the "fit" command) the spectrum will be updated automatically and the calculate command needs not to be given.

2.27.2 Syntax

The following syntax rules apply:
calc - Evaluates the spectral model.

2.27.3 Examples

`calc` - Evaluates the spectral model.

2.28 Fit: spectral fitting

2.28.1 Overview

With this command one fits the spectral model to the data. Only parameters that are thawn are changed during the fitting process. Options allow you to do a weighted fit (according to the model or the data), have the fit parameters and plot printed and updated during the fit, and limit the number of iterations done.
Here we make a few remarks about proper data weighting. χ^2 is usually calculated as the sum over all data bins i of $(N_i - s_i)^2/\sigma_i^2$, i.e.

$$\chi^2 = \sum_{i=1}^{n} \frac{(N_i - s_i)^2}{\sigma_i^2}, \tag{2.1}$$

where N_i is the observed number of source plus background counts, s_i the expected number of source plus background counts of the fitted model, and for Poissonian statistics usually one takes $\sigma_i^2 = N_i$. Take care that the spectral bins contain sufficient counts (either source or background), recommended is e.g. to use at least $\sim$10 counts per bin. If this is not the case, first rebin the data set whenever you have a "continuum" spectrum. For line spectra you cannot do this of course without loosing important information! Note however that this method has inaccuracies if N_i is less than $\sim$100.
Wheaton et al. ([1995]) have shown that the classical χ^2 method becomes inaccurate for spectra with less than $\sim$ 100 counts per bin. This is *not* due to the approximation of the Poisson statistic by a normal distribution, but due to using the *observed* number of counts N_i as weights in the calculation of χ^2. Wheaton et al. showed that the problem can be resolved by using instead $\sigma_i^2 = s_i$, i.e. the *expected* number of counts from the best fit model.
The option "fit weight model" allows to use these modified weights. By selecting it, the expected number of counts (both source plus background) of the current spectral model is used onwards in calculating the fit statistic. Wheaton et al. suggest to do the following 3-step process, which we also recommend to the user of SPEX:

1. first fit the spectrum using the data errors as weights (the default of SPEX).
2. After completing this fit, select the "fit weight model" option and do again a fit
3. then repeat this step once more by again selecting "fit weight model" in order to replace s_i of the first step by s_i of the second step in the weights. The result should now have been converged (under the assumption that the fitted model gives a reasonable description of the data, if your spectral model is way off you are in trouble anyway!).

There is yet another option to try for spectral fitting with low count rate statistics and that is maximum likelyhood fitting. It can be shown that a good alternative to χ^2 in that limit is

$$C = 2 \sum_{i=1}^{n} s_i - N_i + N_i \ln(N_i/s_i). \tag{2.2}$$

This is strictly valid in the limit of Poissonian statistics. If you have a background subtracted, take care that the subtracted number of background counts is properly stored in the spectral data file, so that raw number of counts can be reconstructed.

2.28.2 Syntax

The following syntax rules apply:
fit - Execute a spectral fit to the data.
fit print #i - Printing the intermediate results during the fitting to the screen for every n-th step, with n=#i (most useful for $n = 1$). Default value: 0 which implies no printing of intermediate steps.
fit iter #i - Stop the fitting process after #i iterations, regardless convergence or not. This is useful to get a first impression for very cpu-intensive models. To return to the default stop criterion, type fit iter 0.
fit weight model - Use the current spectral model as a basis for the statistical weight in all subsequent spectral fitting.
fit weight data - Use the errors in the spectral data file as a basis for the statistical weight in all subsequent spectral fitting. This is the default at the start of SPEX.
fit method classical - Use the classical Levenberg-Marquardt minimisation of χ^2 as the fitting method. This is the default at start-up.
fit method cstat - Use the C-statistics instead of χ^2 for the minimisation.

2.28.3 Examples

`fit` - Performs a spectral fit. At the end the list of best fit parameters is printed, and if there is a plot this will be updated.
`fit print 1` - If followed by the above fit command, the intermediate fit results are printed to the screen, and the plot of spectrum, model or residuals is updated (provided a plot is selected).
`fit iter 10` - Stop the after 10 iterations or earlier if convergence is reached before ten iterations are completed.
`fit iter 0` - Stop fitting only after full convergence (default).
`fit weight model` - Instead of using the data for the statistical weights in the fit, use the current model.
`fit weight data` - Use the data instead for the statistical weights in the fit.
`fit method cstat` - Switch from χ^2 to C-statistics.
`fit method clas` - Switch back to χ^2 statistics.

2.29 Error: Calculate the errors of the fitted parameters

2.29.1 Overview

This command calculates the error on a certain parameter or parameter range, if that parameter is thawn. Standard the 1σ rms error is calculated. (this is *not* equivalent to 68 %, as the relevant distributions are χ^2 and not normal!). So $\Delta\chi^2$ is 2, for a single parameter error. The $\Delta\chi^2$ value can be set, such that for instance 90 % errors are determined.
SPEX determines the error bounds by iteratively modifying the parameter of interest and calculating χ^2 as a function of the parameter. During this process the other free parameters of the model may vary. The iteration stops when $\chi^2 = \chi^2_{\rm min} + \Delta\chi^2$, where $\Delta\chi^2$ is a parameter that can be set separately. The standard errors of SPEX (rms errors are for $\Delta\chi^2 = 2$. The iteration steps are displayed. It is advised to check them, because sometimes the fit at a trial parameter converges to a different solution branch, therefore creating a discontinuous jump in χ^2. In those situations it is better to find the error bounds by varying the search parameter by hand.
Finally note that SPEX remembers the parameter range for which you did your lat error search. This saves you often typing in sector numbers or component numbers if you keep the same spectral component for your next error search.

WARNING: *A parameter must have the status "thawn" in order to be able to determine its errors.*

2.29.2 Syntax

The following syntax rules apply:
error [[[#i1:] #i2:] #a:] - Determine the error bars for the parameters specified by the sector range #i1: (optional), component range #i2 (optional) and parameter range #a: (optional). If not specified, the range for the last call will be used. On startup, this is the first parameter of the first component of the first sector.
error dchi #r - This command changes the $\Delta\chi^2$, to the value #r. Default at startup and recommende value to use is 2, for other confidence levels see Table 2.6.
error start #r - This command gives an initial guess of the error bar, from where to start searching the relevant error. This can be helpful for determining the errors on normalization parameters, as otherwise SPEX may from a rather small value. To return to the initial situation, put #r=0 (automatic error search).

2.29.3 Examples

`error norm` - Find the error for the normalization of the current component
`error 2:3 norm:gamm` - determines the error on all free parameters between "norm" and "gamm" for components 2:3
`error start 0.01` - Start calculating the error beginning with an initial guess of 0.01
`error dchi 2.71` - Calculate from now onwards the 90% error, for 1 degree of freedom. (Not recommended, use r.m.s. errors instead!)

Table 2.6: $\Delta\chi^2$ as a function of confidence level, P, and degrees of freedom, ν.

	ν					
P	1	2	3	4	5	6
r.m.s.	2.00	4.00	6.00	8.00	10.00	12.00
68.3%	1.00	2.30	3.53	4.72	5.89	7.04
90%	2.71	4.61	6.25	7.78	9.24	10.6
95.4%	4.00	6.17	8.02	9.70	11.3	12.8
99%	6.63	9.21	11.3	13.3	15.1	16.8
99.99%	15.1	18.4	21.1	13.5	25.7	27.8

2.30 Step: Grid search for spectral fits

2.30.1 Overview

A grid search can be performed of χ^2 versus 1, 2, 3 or 4 parameters. The minimum, maximum and number of steps for each parameter may be adjusted. Steps may be linear or logarithmic. For each set of parameters, a spectral fit is made, using the last spectral fit executed before this command as the starting point. For each step, the parameters and χ^2 are displayed. This option is useful in case of doubt about the position of the best fit in the parameter space, or in cases where the usual error search is complicated. Further it can be useful incases of complicated correlations between parameters.

WARNING: *Take care to do a spectral fit first!*

WARNING: *Beware of the cpu-time if you have a fine grid or many dimensions!*

2.30.2 Syntax

The following syntax rules apply:
step dimension #i - Define the number of axes for the search (between 1–4).
step axis #i1 parameter [[#i2] #i3] #a range #r1 #r2 n #i4 - Set for axis #i1, optional sector #i2, optional component #i3 and parameter with name #a the range to the number specified by #r1 and #r2, with a number of steps n given by #i4. If $n > 0$, a linear mesh between #r1 and #r2 will be taken, for $n < 0$, a logarithmic grid will be taken.
step - Do the step search. Take care to do first the "step dimension" command, followed by as many "step axis" commands as you entered the number of dimensions.

2.30.3 Examples

Below we give a worked out example for a CIE model where we took as free parameters the normalization "norm", temperature "t", Si abundance "14" and Fe abundance "26". To do a 3-dimensional grid search on the temperature (logarithmic between 0.1 and 10 keV with 21 steps), Fe abundace (linear between 0.0 and 1.0 with 11 steps) and Si abundance (linear between 0.0 and 2.0 with 5 steps, i.e. values 0.0, 0.4, 0.8, 1.2, 1.6 and 2.0) the following commands should be issued:
`fit` - Do not forget to do first a fit
`step dimension 3` - We will do a three-dimensional grid search
`step axis 1 parameter 1 1 t range 0.1 10.0 n -21` - The logarithmic grid for the first axis, temperature; note the − before the 21!
`step axis 2 par 26 range 0 1 n 11` - The grid for axis 2, Fe abundance
`step axis 3 par 14 range 0 2 n 5` - Idem for the 3rd axis, Silicon
`step` - Now do the grid search. Output is in the form of ascii data on your screen (and/or output file if you opened one).

2.31 DEM: differential emission measure analysis

2.31.1 Overview

SPEX offers the opportunity to do a differential emission measure analysis. This is an effective way to model multi-temperature plasmas in the case of continuous temperature distributions or a large number of discrete temperature components.

The spectral model can only have one additive component: the DEM component that corresponds to a multi-temperature structure. There are no restrictions to the number of multiplicative components. For a description of the DEM analysis method see document SRON/SPEX/TRPB05 (in the documentation for version 1.0 of SPEX) Mewe et al. (1994) and Kaastra et al. (1996). SPEX has 5 different dem analysis methods implemented, as listed shortly below. We refer to the above papers for more details.

1. reg – Regularization method (minimizes second order derivative of the DEM; advantage: produces error bars; disadvantage: needs fine-tuning with a regularization parameter and can produce negative emission measures.

2. clean – Clean method: uses the clean method that is widely used in radio astronomy. Useful for "spiky emission measure distributions.

3. poly – Polynomial method: approximates the DEM by a polynomial in the $\log T - - \log Y$ plane, where Y is the emission measure. Works well if the DEM is smooth.

4. mult – Multi-temperature method: tries to fit the DEM to the sum of Gaussian components as a function of $\log T$. Good for discrete and slightly broadened components, but may not always converge.

5. gene – Genetic algorithm: using a genetic algorithm try to find the best solution. Advantage: rather robust for avoiding local subminima.Disadvantage: may require a lot of cpu time, and each run produces slightly different results (due to randomization).

In practice to use the DEM methods the user should do the following steps:

1. Read and prepare the spectral data that should be analysed

2. Define the dem model with the "comp dem" command

3. Define any multiplicative models (absorption, redshifts, etc.) that should be applied to the additive model

4. Define the parameters of the dem model: number of temperature bins, temperature range, abundances etc.

5. give the "dem lib" command to create a library of isothermal spectra.

6. do the dem method of choice (each one of the five methods outlined above)

7. For different abundances or parameters of any of the spectral components, first the "dem lib" command must be re-issued!

2.31.2 Syntax

The following syntax rules apply:
dem lib - Create the basic DEM library of isothermal spectra
dem reg auto - Do DEM analysis using the regularization method, using an automatic search of the optimum regularization parameter. It determines the regularisation parameter R in such a way that $\chi^2(R) = \chi^2(0)[1 + s\sqrt{2/(n-n_{\rm T}}]$ where the scaling factor $s = 1$, n is the number of spectral bins in the data set and $n_{\rm T}$ is the number of temperature components in the DEM library.
dem reg auto #r - As above, but for the scaling factor s set to #r.
dem reg #r - Do DEM analysis using the regularization method, using a fixed regularization parameter R =#r.
dem chireg #r1 #r2 #i - Do a grid search over the regularization parameter R, with #i steps and R distributed logarithmically between #r1 and #r2. Useful to scan the $\chi^2(R)$ curve whenever it is complicated and to see how much "penalty' (negative DEM values) there are for each value of R.
dem clean - Do DEM analysis using the clean method
dem poly #i - Do DEM analysis using the polynomial method, where #i is the degree of the polynomial
dem mult #i - Do DEM analysis using the multi-temperature method, where #i is the number of broad components
dem gene #i1 #i2 - Do DEM analysis using the genetic algorithm, using a population size given by #i1 (maximum value 1024) and #i2 is the number of generations (no limit, in practice after ~100 generations not much change in the solution. Experiment with these numbers for your practical case.
dem read #a - Read a DEM distribution from a file named #a which automatically gets the extension ".dem". It is an ascii file with at least two columns, the first column is the temperature in keV and the second column the differential emission measure, in units of 10^{64} m^{-3} keV^{-1}. The maximum number of data points in this file is 8192. Temperature should be in increasing order. The data will be interpolated to match the temperature grid defined in the dem model (which is set by the user).
dem save #a - Save the DEM to a file #a with extension ".dem". The same format as above is used for the file. A third column has the corresponding error bars on the DEM as determined by the DEM method used (not always relevant or well defined, exept for the regularization method).
dem smooth #r - Smoothes a DEM previously determined by any DEM method using a block filter/ Here #r is the full width of the filter expressed in $^{10}\log T$. Note that this smoothing will in principle worsen the χ^2 of the solution, but it is sometimes useful to "wash out" some residual noise in the DEM distribution, preserving total emission measure.

2.31.3 Examples

`dem lib` - create the DEM library
`dem reg auto` - use the automatic regularization method
`dem reg 10.` - use a regularization parameter of $R = 10$ in the regularization method
`dem chireg 1.e-5 1.e5 11` - do a grid search using 11 regularisation parameters R given by 10^{-5}, 10^{-4}, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10^4, 10^5.

`dem clean` - use the clean method
`dem poly 7` - use a 7th degree polynomial method
`dem gene 512 128` - use the genetic algorithm with a population of 512 and 128 generations
`dem save mydem` - save the current dem on a file named mydem.dem
`dem read modeldem` - read the dem from a file named modeldem.dem
`dem smooth 0.3` - smooth the DEM to a temperature width of 0.3 in ${}^{10}\log T$ (approximately a factor of 2 in temperature range).

2.32 Ascdump: ascii output of plasma properties

2.32.1 Overview

One of the drivers in developing SPEX is the desire to be able to get insight into the astrophysics of X-ray sources, beyond merely deriving a set of best-fit parameters like temperature or abundances. The general user might be interested to know ionic concentrations, recombination rates etc. In order to facilitate this SPEX contains options for ascii-output.
Ascii-output of plasma properties can be obtained for any spectral component that uses the basic plasma code of SPEX; for all other components (like power law spectra, gaussian lines, etc.) this sophistication is not needed and therefore not included. There is a choice of properties that can be displayed, and the user can either display these properties on the screen or write it to file.
The possible output types are listed below. Depending on the specific spectral model, not all types are allowed for each spectral component. The keyword in front of each item is the one that should be used for the appropriate syntax.

plas: basic plasma properties like temperature, electron density etc.

abun: elemental abundances and average charge per element.

icon: ion concentrations, both with respect to Hydrogen and the relevant elemental abundance.

rate: total ionization, recombination and charge-transfer rates specified per ion.

rion: ionization rates per atomic subshell, specified according to the different contributing processes.

pop: the occupation numbers as well as upwards/downwards loss and gain rates to all quantum levels included.

elex: the collisional excitation and de-excitation rates for each level, due to collisions with electrons.

prex: the collisional excitation and de-excitation rates for each level, due to collisions with protons.

rad: the radiative transition rates from each level.

two: the two-photon emission transition rates from each level.

grid: the energy and wavelength grid used in the last evaluation of the spectrum.

clin: the continuum, line and total spectrum for each energy bin for the last plasma layer of the model.

line: the line energy and wavelength, as well as the total line emission (photons/s) for each line contributing to the spectrum, for the last plasma layer of the model.

con: list of the ions that contribute to the free-free, free-bound and two-photon continuum emission, followed by the free-free, free-bound, two-photon and total continuum spectrum, for the last plasma layer of the model.

tcl: the continuum, line and total spectrum for each energy bin added for all plasma layers of the model.

tlin: the line energy and wavelength, as well as the total line emission (photons/s) for each line contributing to the spectrum, added for all plasma layers of the model.

tcon: list of the ions that contribute to the free-free, free-bound and two-photon continuum emission, followed by the free-free, free-bound, two-photon and total continuum spectrum, added for all plasma layers of the model.

cnts: the number of counts produced by each line. Needs an instrument to be defined before.

snr: hydrodynamical and other properties of the supernova remnant (only for supernova remnant models such as Sedov, Chevalier etc.).

heat: plasma heating rates (only for photoionized plasmas).

ebal: the energy balance contributions of each layer (only for photoionized plasmas).

dem: the emission measure distribution (for the pdem model)

col: the ionic column densities for the hot, pion, slab, xabs and warm models

tran: the transmission and equivalent width of absorption lines and absorption edges for the hot, pion, slab, xabs and warm models

warm: the column densities, effective ionization parameters and temperatures of the *warm* model

2.32.2 Syntax

The following syntax rules apply for ascii output:
ascdump terminal #i1 #i2 #a - Dump the output for sky sector #i1 and component #i2 to the terminal screen; the type of output is described by the parameter #a which is listed in the table above.
ascdump file #a1 #i1 #i2 #a2 - As above, but output written to a file with its name given by the parameter #a1. The suffix ".asc" will be appended automatically to this filename.

WARNING: *Any existing files with the same name will be overwritten.*

2.32.3 Examples

`ascdump terminal 3 2 icon` - dumps the ion concentrations of component 2 of sky sector 3 to the terminal screen.
`ascdump file mydump 3 2 icon` - dumps the ion concentrations of component 2 of sky sector 3 to a file named mydump.asc.

2.33 Plot: Plotting data and models

2.33.1 Overview

The plot command cause the plot to be (re)drawn on the graphics device. Multiple graphics devices can be defined in one SPEX session. For example, a plot can be sent to both a postscript and a xs device.
A user can also set the number of plot frames in the currently active device(s), e.g. the data and model can be displayed in one frame while the residuals can be displayed in a frame below. The user has to specify what is to be plotted in in each frame.
In each plot frame, there can be different sets that can be specified by the "set" keyword in the plot command. For example, for the plot type data, each set corresponds to the spectrum of an instrument or region; with this option it is therefore possible to give different instruments different colours or plot symbols, etc., or just to omit one of the instruments from the plot.
The plot command is also used to select the axis plot units and scales as well as character commands like font style, line weights, plot colours, etc. Finally, the plot command can also be used to dump the data to an ascii file.

WARNING: *To make a plot, always start with a "plot device" command to open any plot device you wish. Next select a plot type using "plot type". After this you can give any plot commands to modify or display the plot.*

2.33.2 Syntax

The following syntax rules apply for plot:
plot - (Re)draws the plot on the graphics device. Take care to open at least one device first (with a "plot dev" command)
plot frame new - Creates a new additional plot frame, where an other plot type can be displayed (or the same plot type but with different units). Take care to position the viewport of the new and old frames to the desired locations.
plot frame #i - Sets frame number #i to be the currently active frame. After this command has been issued, all plot commands will only affect the currently active frame.
plot frame delete #i - Deletes frame number #i.
plot type #a - Specifies what is to be plotted in the selected frame. #a can be any of the plot types specified in Sect. 3.2.
plot device #a1 [#a2] - Selects a graphics device #a1 and optional file name #a2 (in the case of a postscript or gif device). File extensions (for example .ps) should be specified. For valid devices, see Sect. 3.1.
plot close #i - Close graphics device number #i. Note, always close a postscript device before quiting /spex, otherwise the contents may be corrupted.
plot hlan - Make a hardcopy in landscape orientation and send it to the standard printer. Use is made of the unix command "lp -c filename".
plot hpor - As above, but for portrait orientation
plot x lin - Plot the x-axis on a linear scale
plot x log - Plot the x-axis on a log scale
plot y lin - Plot the y-axis on a linear scale
plot y log - Plot the y-axis on a log scale
plot y mix #r1 #r2 - Plot the y-axis on a mixed linear/logarithmic scale; for y-values below #r1 the data will be plotted linear, and for y-values above #r1 the data will be plotted loga-

rithmically; the linear part occupies the fraction of the total frame height given by #r2. See also Sect. 3.8.2.
plot z lin - Plot the z-axis on a linear scale. The z-axis is defined for two-dimensional plots like contour plots.
plot z log - As above, but using a log scale.
plot ux #a - Set the plot units on the x-axis, #a can be Å, keV, eV or whatever unit that is allowed. The allowed values depend upon the plot type. See Sect. 3.8.1 for allowed units for each plot type.
plot uy #a - Same as above, but for the y-axis
plot uz #a - Same as above, but for the z-axis
plot rx #r1 #r2 - set the plot range on the x-axis from #r1 to #r2
plot ry #r1 #r2 - Same as above, but for the y-axis
plot rz #r1 #r2 - Same as above, but for the z-axis
plot view default #l - For #l is true, the default viewport is used. For #l is false, the default viewport is overruled and you can specify your own viewport limits with the "set view x" and "set view y" commands. Useful for stacking different plot frames.
plot view x #r1 #r2 - Set the x viewport range from #r1 to #r2, where #r1 and #r2 must be in the range 0–1.
plot view y #r1 #r2 - Same as above, but for the y-range.
plot view x1 #r - Set the lower x viewport limit to #r
plot view x2 #r - Set the upper x viewport limit to #r
plot view y1 #r - Set the lower y viewport limit to #r
plot view y2 #r - Set the upper y viewport limit to #r
plot view back #i - Set the viewport background colour to value #i. See Sect. 3.3 for the allowed plot colours.
plot view transp #l - If true, set the viewport background to to transparent, the plot frame will be shown with the background colour selected by the "plot view back" command. Default is false.
plot box disp #l - Display a box around the viewport, true/false
plot box edge #a #l - If true (default), display the edges of the plot box, #a should be one of the following keywords: top, bottom, left, right. Note that whether you see the edges or not also depends on the settings defined by the "plot box disp" command.
plot box axis x #l - Plots an x-axis with tickmarks at the line $y = 0$. You only see this of course if your y-range includes the value 0.
plot box axis y #l - As above, but for the y-axis
plot box grid x #l - Plot a grid for the x-axis
plot box grid y #l - Plot a grid for the y-axis
plot box ymix #l - Not implemented yet
plot box numlab bottom #l - Show the tick number labels at the bottom of the plot, true/false
plot box numlab top #l - Same as above, but for the top
plot box numlab left #l - Same as above, but for the left
plot box numlab right #l - Same as above, but for the right
plot box numlab vert #l - Plot the tick number labels on the y-axis vertically or horizontally, set #l to true for vertical tnumbers and false for horizontal numbers.
plot box numlab xscal #i - Way to denote the numerical numbers along the x-axis. Three values are allowed: 0=automatic, 1=forced decimal labeling, 2=forced exponential labeling. Default is 0.
plot box numlab yscal #i - As above, but for the y-axis

plot box tick invert x #l - Draw the tick marks on the x-axis on the inside or outside the box, set #l to true for outside and false for inside (default).
plot box tick invert y #l - Same as above, but for the y-axis.
plot box tick minor x #l - Draw minor tick marks on the x-axis, true/false
plot box tick minor y #l - Same as above, but for the y-axis
plot box tick major x #l - Draw major tick marks on the x-axis, true/false
plot box tick major y #l - Same as above, but for the y-axis
plot box tick distance x #r - Set the distance between the major/labelled tick marks on the x-axis to #r
plot box tick distance y #r - Same as above, but for the y-axis
plot box tick subdiv x #i - Draw #i minor tick marks between each major tick mark on the x-axis
plot box tick subdiv y #i - Same as above, but for the y-axis
plot box col #i - Set the box colour to colour number #i. See Sect. 3.3 for the allowed plot colours.
plot box lt #i - Set the box line type to #i. See Sect. 3.4 for allowed line types.
plot box lw #i - Set the box line weight to number #i. See Sect. 3.4 for more about line weights.
plot box fh #r - Set the box font height to number #i.
plot box font #i - Set the box font to number #i. See Sect. 3.5 for more details about text fonts.
plot cap #a disp #l - If #l is true, display the caption (default). For more about captions see Sect. 3.6. Here and below, #a can be x, y, z, id, lt or ut.
plot cap #a col #i - Plot caption #a in colour number #i. See Sect. 3.3 for valid colour indices.
plot cap #a back #i - Set the background colour for caption #a to #i.
plot cap #a lw #i - Set the font line weight for caption #a to #i.
plot cap #a fh #r - Set the font height for caption #a to value #r.
plot cap #a font #i - Set the font type for caption #a to #i.
plot cap #a1 text #a - Set the text for caption #a1 to #a2. Note that the caption text should be put between quotion marks, like "best fit results" if you want to see the text: best fit results.
plot cap #a1 side #a2 - Plot caption #a1 at the side of the frame specified by #a2, which may stand for t (top), b (bottom), lh (left, horizontal), rh (right, horizontal), lv (left, vertical) and rv (right, vertical).
plot cap #a off #r - Offset caption #a by #r from the edge of the viewport, where #r is in units of the character height. Enter negative values to write inside the viewport.
plot cap #a coord #r - Plot caption #a along the specified edge of the viewport, in units of the viewport length, where $0.0 \leq \#r \leq 1.0$.
plot cap #a fjust #r - Controls justification of the caption #a parallel to the specified edge of the viewport. If #r = 0.0, the left hand of #a will be placed at the position specified by "coord" above; if #r = 0.5, the center of the string will be placed at "coord", if #r = 1.0 the right hand of #a will be placed at "coord". Other values can be used but are less useful.
plot string new #r1 #r2 #a - Plot a new string with text as specified in #a at x=#r1 and y = #r2. See Sect. 3.5 for more deatils about text strings. Also do not forget to put #a between "" if it consists of more than one word (i.e., if it contains spaces).
plot string del #i: - Delete string numbers specified by the range #i from the plot.
plot string #i: disp #l - If true (default), display the strings specified by the range #i:.
plot string #i: text #a - Change the text of strings #i: to #a
plot string #i1: col #i2 - Set the colours of strings #i1: to #i2
plot string #i1: back #i2 - Set the background colour for the strings #i1: to the value #i2.
plot string #i1: lw #i2 - Set the line weight of strings #i1: to #i2.

plot string #i: fh #r - Set the font height of strings #i to #r.
plot string #i1: font #i2 - Set the font style of strings #i1 to #i2.
plot string #i: x #r - Set the x position of strings #i: to #r.
plot string #i: y #r - Set the y position of string #i: to #r.
plot string #i: angle #r - Set the angle of strings #i: to #r.
plot string #i: fjust #r - Controls justification of the strings #i: parallel to the specified edge of the viewport. If #r = 0.0, the left hand of the strings will be placed at the position specified by "x y" above; if #r = 0.5, the center of the strings will be placed at "x y", if #r = 1.0 the right hand of #i: will be placed at "x y". Other values can be used but are less useful.
plot string #i: box #l - If #l is true, plot a box around the text strings #i:. The default value is false (no box).
plot string #i1: box lt #i2 - Set the line style of the box around the strings #i1: to the value #i2.
plot string #i1: box lw #i2 - As above, but for the line weight specified by #i2.
plot string #i1: box col #i2 - As above, but for the colour index for the box specified by #i2.
plot set #i: - Selects data set numbers as specified by #i:. Afterwards most plot commands will only affect data sets #i:
plot set all - Selects all data sets that are present. All subsequent plot commands will be executed for all data sets.
plot line disp #l - If #l is true, plots a connecting line through the data points, (default is false).
plot line col #i - Set the colour of the connecting line to #i.
plot line lt #i - Set the line style of the connecting line to #i.
plot line lw #i - Set the line weight of the connecting line to #i.
plot line his #l - If #l is true, plot the connecting line in histogram format (default is false).
plot elin disp #l - If #l is true, plots a connecting line through the end points of the error bars, (default depends upon the plot type).
plot elin col #i - Set the colour of the connecting line through the end points of the error bars to #i.
plot elin lt #i - Set the line style of the connecting line through the end points of the error bars to #i.
plot elin lw #i - Set the line weight of the connecting line through the end points of the error bars to #i.
plot elin his #l - If #l is true, plot the connecting line through the end points of the error bars in histogram format (default is false).
plot model disp #l - If #l is true, plot the current model corresponding to the relevant data set (default is true).
plot model col #i - Set the colour of the model to number #i.
plot model lt #i - Set the line style of the model to number #i.
plot model lw #i - Set the line weight of the model to number #i.
plot model his #l - If #l is true, plot the model in histogram format (default is true).
plot back disp #l - If #l is true, plot the subtracted background (default is true).
plot back col #i - Set the colour of the subtracted background to number #i.
plot back lt #i - Set the line style of the subtracted background to number #i.
plot back lw #i - Set the line weight of the subtracted background to number #i.
plot back his #l - If true, plot the subtracted background in histogram format (default is true).
plot fill disp #l - If #l is true, fill the curve below the model with the colour specified by the next command or the default colour.
plot fill col #i - Change the filling colour to #i.

plot fill lt #i - Change the line type of the filling lines to #i.
plot fill lw #i - Change the line weight of the filling lines to #i.
plot fill style #i - Change the style of the filling lines to the value #i. Here #i has values between 1–4, with the following meaning: 1 = solid filling (default), 2 = outline, 3 = hatched, 4 = cross-hatched.
plot fill angle #r - Set the angle for the filling lines for hatched filling. Default is 45 degrees.
plot fill sep #r - Set the distance between the filling lines for hatched filling. The unit spacing is 1 % of the smaller of the height or width of the viewing surface. This should not be zero.
plot fill phase #r - The phase between the hatch lines that fill the area.
plot data disp #l - If #l is true, display the data.
plot data errx #l - If #l is true, display the error bars in the x-direction.
plot data erry #l - If #l is true, display the error bars in the y-direction.
plot data col #i - Give the data colour index #i.
plot data lt #i - Give the data line style #i.
plot data lw #i - Give the data line weight #i.
plot data fh #r - Give the symbols for the data font height #r.
plot data symbol #i - Plot the data with symbol number #i. For symbol numbers, see Sect. 3.7.
plot adum #a [overwrite] [append] - Dump the data and model in the plot in an ascii file with filename #a. The extension ".qdp" will automatically be appended. Note that the data will be written as they are, i.e. if you have a logarithmic x-axis or y-axis, the logs of the plotted quantities will be written. If you want to replot your data later with for example the qdp package, take care that you plot the data in SPEX on a lin-lin frame before you execute the "plot adum" command. Also note that the data will be written in the units that were specified in the plot (energy, wavelength or whatever is applicable. If the optional "append" keyword is present, the data will be appended to any existing file with the name #a; if the optional "overwrite" keyword is present, any pre-existing file with the name #a will be overwritten by the new data.

2.33.3 Examples

`plot device xs` - Open the graphic device xs (xserver)
`plot device ps myplot.ps` - Select a postscript device connected to the file name myplot.ps
`plot type data` - Plot the data on the selected graphics device(s)
`plot ux angstrom` - Set the x-axis plot units to Å
`plot uy angstrom` - Set the y-axis plot units to Counts/s/Å
`plot frame new` - Open a new frame in the selected graphics device(s)
`plot frame 2` - Go to the 2nd frame, all plot commands will now only affect frame 2
`plot type chi` - Plot the residuals in frame 2
`plot uy rel` - Set the y-axis plot units in frame 2 to (Obsereve - Model)/Model
`plot view default f` - Set the default viewport keyword to false so that new user viewport values can be specified for frame 2
`plot view y2 0.2` - Set the upper y viewport limit of frame 2 to 0.2 of the full device window
`plot cap id disp f` - Do not display the id caption of frame 2
`plot cap ut disp f` - Do not display the upper top caption of frame 2
`plot cap lt disp f` - Do not display the lower top caption of frame 2
`plot ux a` - Set the x-axis plot units of frame 2 to Å
`plot ry -1 1` - Set the y-axis plot range of frame 2 to between a lower limit of -1 and an upper limit of 1

`plot frame 1` - Go to frame 1
`plot view default f` - Set the default viewport keyword to false so that new user viewport values can be specified for frame 1
`plot view y1 0.25` - Set the lower y viewport limit of frame 1 to 0.25 of the full device window
`plot de cps filename.ps` - Open a colour postscript graphics device and write the output file to filename.ps
`plot` - Redraw the plot on all frames and devices
`plot close 2` - Close device number 2, which is the postscript device in this case

Chapter 3

More about plotting

3.1 Plot devices

The following devices are available (may depend somewhat on your local operating system, i.e. the allowed devices in your local pgplot installation):

1. null : Null device (no output)
2. xwin : Workstations running X Window System
3. xser : Persistent window on X Window System
4. ps : PostScript printers, monochrome, landscape
5. vps : Postscript printers, monochrome, portrait
6. cps : PostScript printers, color, landscape
7. vcps : PostScript printers, color, portrait
8. gif : GIF-format file, landscape
9. vgif : GIF-format file, portrait

WARNING: *Note that when you enter the device name in SPEX you do not need to enter the leading shlash; this is only required in other applications using the pgplot library.*

3.2 Plot types

There are several plot types available in SPEX. below we list them together with the acronym that should be used when defining this plot type (in a "plot type" command).

- data – a spectrum observed with an instrument plus optionally the folded (predicted) model spectrum and/or the subtracted background spectrum.
- model – the model spectrum, not convolved by any instrumental profile
- area – the effective area of the instrument. If the response matrix of the instrument contains more components, also the contribution of each component is plotted separately.
- fwhm – the peak energy of the response matrix (as the model) as well as the FWHM limits (as the data) are plotted. The FWHM limits are the energies at which the response reaches half of its maximum value. If for the y-axis the option "de/e" is chosen, all calculated energies are divided by the energy of the incoming photon.
- chi – the fit residuals, either expressed in units such as counts/s or the same units as the spectrum, or expressed in terms of numer of standard deviations, or as a relative error.
- dem – a differential emission measure distribution).

Table 3.1: Available plot colours.

Value	Colour
00	Black (background)
01	White (default)
02	Red
03	Green
04	Blue
05	Cyan (Green + Blue)
06	Magenta (Red + Blue)
07	Yellow (Red + Green)
08	Orange (Red + Yellow)
09	Green + Yellow
10	Green + Cyan
11	Blue + Cyan
12	Blue + Magenta
13	Red + Magenta
14	Dark Gray
15	Light Gray

3.3 Plot colours

Table 3.1 lists the plot colours that are available. See also Fig. 3.3.

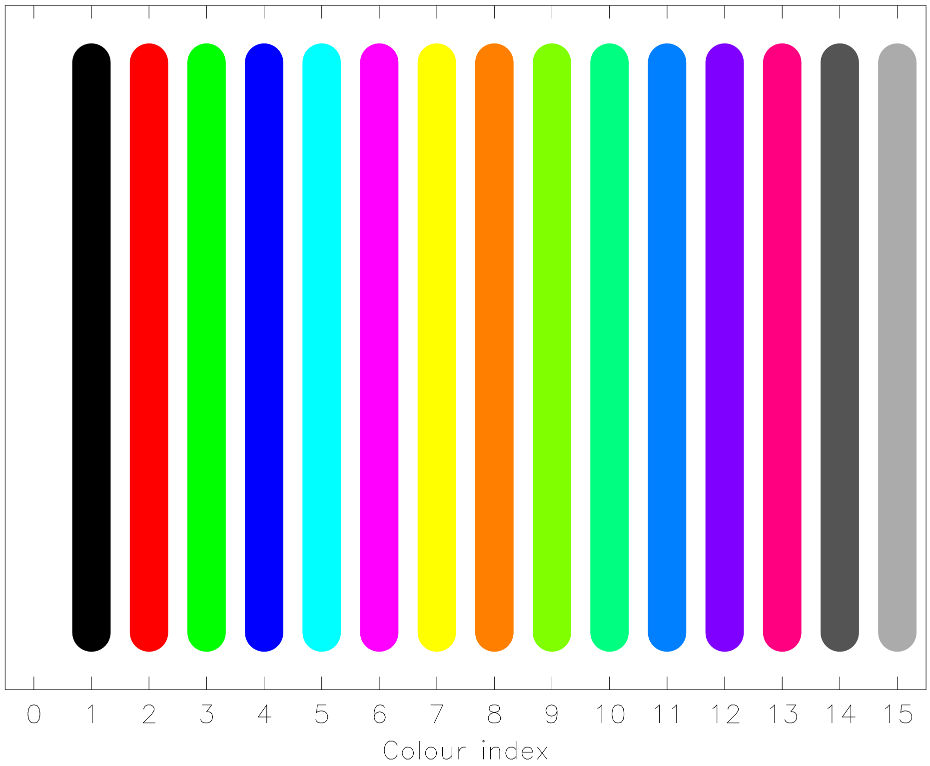


Figure 3.1: Plot colours available with pgplot. Note that on a hardcopy colour 1 is black and the background (colour 0) is white, while on most terminal screens colour 0 is balck and colour 1 is white.

3.4 Plot line types

When drawing lines, the following line types are possible:

1. `------------`
2. `- - - - - - -`
3. `.-.-.-.-.-.-.`
4. `.............`
5. `-..-..-..-..-`

The first is the default value.
Further, the user can specify the thickness of the lines. Default value is 1, larger values give thicker lines. Maximum value is 200. Recommended value for papers: use line weight 3. For an example of different line weights, see also Fig. 3.4.

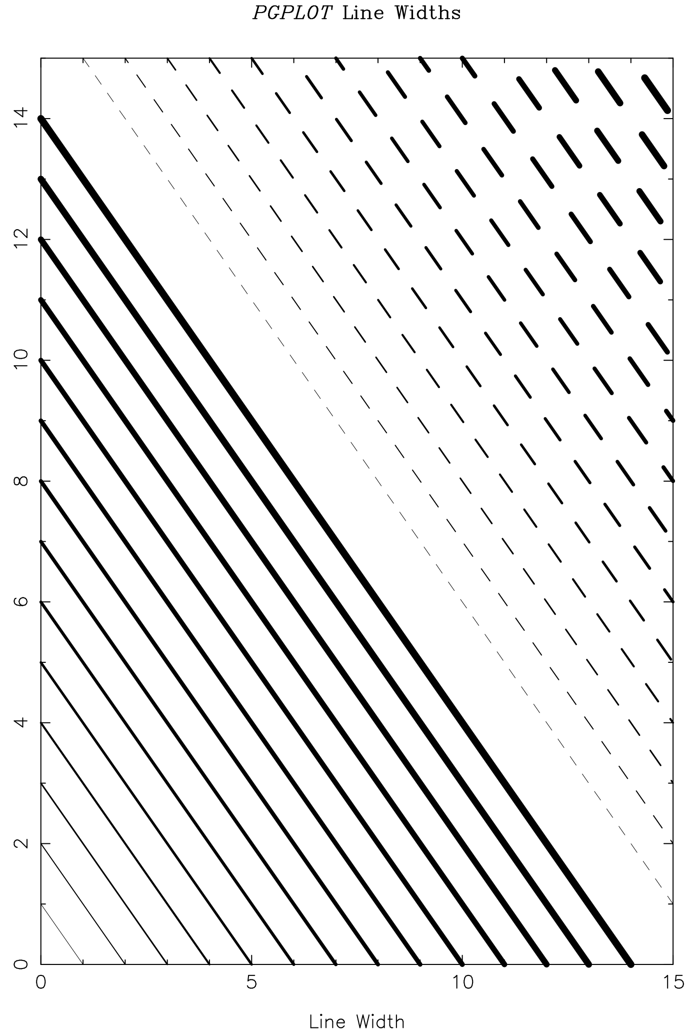


Figure 3.2: Example of different line weights.

3.5 Plot text

In SPEX it is possible to draw text strings on the plot surface or near the frame of the plot. Several properties of the text can be adjusted, like the fontheight, font type, colour, orientation and location of the text. See Fig. 3.5.1 for an example of possible fonts.

3.5.1 Font types (font)

The following fonts types are allowed (values between 1–4):

1. normal font (default)
2. roman font
3. italic font
4. script font

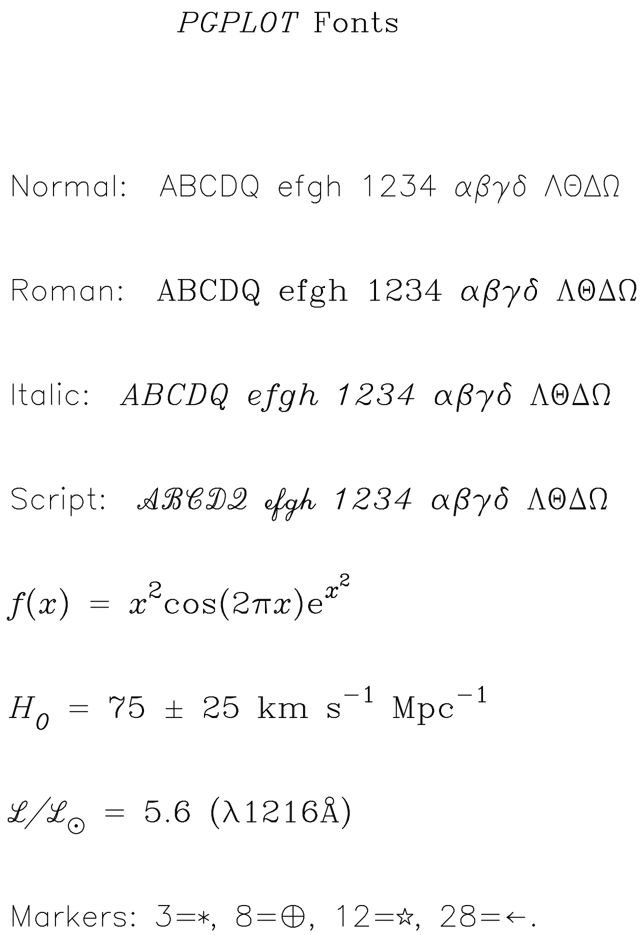


Figure 3.3: Example of different fonts.

Table 3.2: A list of available escape sequences.

Seq.	description
`\u`	start a superscript or end a subscript. A `\u` must be ended by a `\d`!
`\d`	start a subscript or end a superscript. A `\d` must be ended by a `\u`!
`\\`	backspace (i.e. do not advance textpointer after plotting the previous character)
`\A`	Ångstrom symbol (Å)
`\g`x	greek letter corresponding to roman letter x
`\fn`	switch to Normal font
`\fr`	switch to Roman font
`\fi`	switch to Italic font
`\fs`	switch to Script font
`\(`n`)`	character number n, see *pgplot manual appendix B, tab. 1*

Table 3.3: List of upper- and lower case Greek letters (G) and their corresponding Roman letters (R).

R :	A	B	G	D	E	Z	Y	H	I	K	L	M	N	C	O	P	R	S	T	U	F	X	Q	W
G :	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π	P	Σ	T	Υ	Φ	X	Ψ	Ω
R :	a	b	g	d	e	z	y	h	i	k	l	m	n	c	o	p	r	s	t	u	f	x	q	w
G :	α	β	γ	δ	ϵ	ζ	η	θ	ι	κ	λ	μ	ν	ξ	o	π	ρ	σ	τ	υ	ϕ	χ	ψ	ω

3.5.2 Font heights (fh)

The font height can be entered as a real number, the default value is 1. A useful value for getting results that are still readable for publications is 1.3. Note that if you make the font height for the captions of the figure too large, you may loose some text off the paper.

3.5.3 Special characters

The texts strings that are plotted may be modified using the pgplot escape sequences. These are character–sequences that are not plotted, but are interpreted as instructions to change the font, draw superscripts or subscripts, draw non-ASCII characters, Greek letters, etc. All escape sequences start with a backslash character (\). A list of the defined escape sequences is given in tab. 3.2. A lookup table for Greek letters is presented in tab. 3.3. Some useful non-ASCII characters are listed in tab. 3.4. Fig. 3.4 shows some examples of the use of pgplot escape sequences in character strings.

Table 3.4: Some useful non-ASCII character sequences.

$\sim$	2248	$\pm$	2233	∇	0583
$\approx$	0248	$\mp$	2234	$\surd$	2267
$\cong$	0250	$\times$	2235	$\int$	2268
$\propto$	2245	$\div$	2237	$\oint$	2269
$\neq$	2239	$\equiv$	2240	∞	2270
$\leqq$	2243	$\dagger$	2277	∂	2286
$\geqq$	2244	$\ddagger$	2278	$\odot$	2281

Displayed	pgplot escape sequence
$f(x) = x^2 \cos(2\pi x)$	`\fif(x) = x\u2\d \frcos\fi(\fr2\fi\gpx)`
$H_0 = 75 \pm 25$ km s^{-1} Mpc^{-1}	`\fiH\d0\u\fr = 75 \(2233) 25 km s\u-1\d Mpc\u-1\d`
$\mathcal{L}/\mathcal{L}_\odot = 5.6$ (λ1216Å)	`\fsL/L\fr\d\(2281)\u = 5.6 (\gl1216\A)`

Figure 3.4: Some examples of the use of pgplot escape sequences.

3.6 Plot captions

Spex has a set of captions defined for each plot. The properties of these captions as well as the text can be modified by the user. For the properties (font types etc.) see Sect. 3.5.
The following captions exist:

- x : the x-axis, plotted below the frame
- y : the y-axis, plotted left from the frame
- z : the z-axis (for contour plots and images), plotted at the upper left edge of the frame
- ut : the upper title text, plotted in the middle above the frame
- lt : the lower title text, plotted between the upper title text and the frame
- id : the plot identification, plotted to the upper right corner of the frame (usually contains the SPEX version and time and date).

With the "plot cap" series of commands, the text, appearance and position of all these captions can be modified.

3.7 Plot symbols

When plotting data, the data can be labeled with different symbols as indicated in Fig. 3.7.

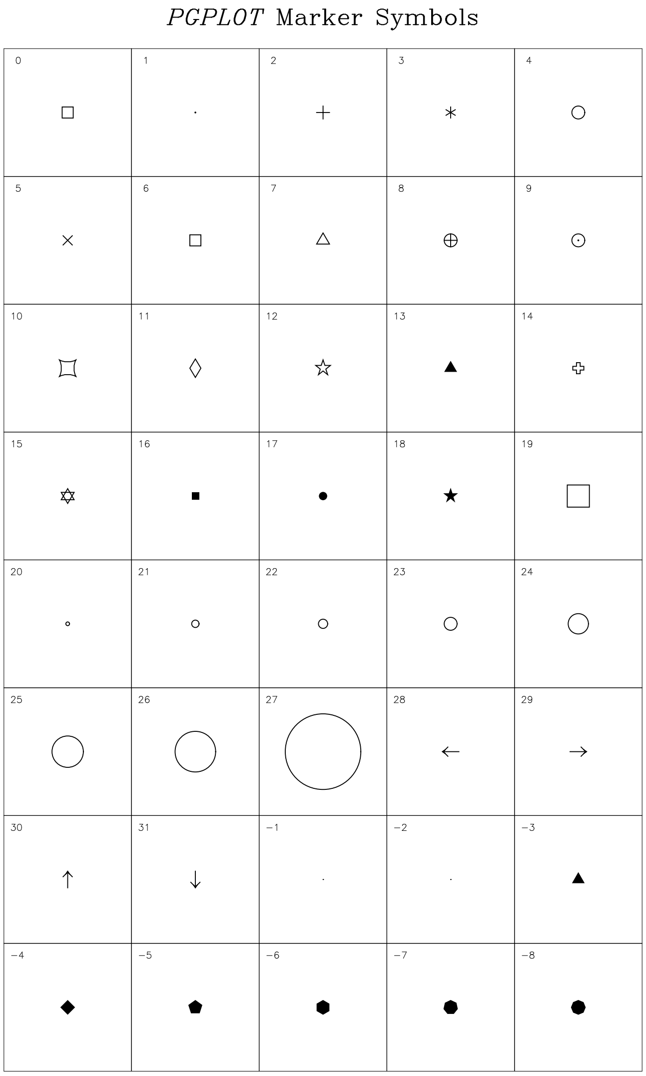


Figure 3.5: Plot symbols that are available.

3.8 Plot axis units and scales

3.8.1 Plot axis units

Different units can be chosen for the quantities that are plotted. Which units are available depends upon the plot type that is used, for example an observed spectrum (data), a Differential Emission Measure distribution (DEM) etc. For the availabe plot types see Sect. 3.2.

Note also that depending upon the plot type, there are two or three axes available, the x-axis, y-axis and z-axis. The last axis is only used for contour maps or images, and corresponds to the quantity that is being plotted in the contours or image.

The units that are allowed for the x-axis are listed in Tables 3.5–3.6, and for the y-axis in Tables 3.7–3.12.

WARNING: *For the plot type data option, there are options (starting with "f") that are essentially counts* m^{-2}. *In these cases, the observed count rates (counts/s/keV for instance) are divided by the nominal effective area at the relevant energy. This nominal effective area is determined for a flat input spectrum, and is defined as the ratio of predicted count rate and model spectrum flux. Response parts below 0.75 times the original photon energy are ignored in the estimation of the predicted count rate, in an attempt to remove higher order spectral lines. Note however that if your spectrum really contains significant higher order lines, you cannot remove these from your plot. Also beware of end-effects. If you want to see the underlying model, it is often better to select the plot type model!*

3.8.2 Plot axis scales

The axes in a plot can be plot in different ways, either linear or logarithmic.

For the y-axis we also have made an option for a mixed linear and logarithmic scale. The lower part of the plot will be on a linear scale, the upper part on a logarithmic scale. This option is useful for example for plotting a background-subtracted spectrum. Often there are bins with a low count rate which scatter around zero after background suntraction; for those channels a linear plot is most suited as a logarithmic scaling cannot deal with negative numbers. On the other hand, other parts of the spectrum may have a high count rate, on various intemsity scales, and in that case a logarithmic representation might be optimal. The mixed scaling allows the user to choose below which y-value the plot will be linear, and also which fraction of the plotting surface the linear part will occupy. For an example see Fig. 3.8.2.

Table 3.5: x-axis units for plot type model, data, chi, area, fwhm and spec:

bin	Bin number
kev	keV (kilo-electronvolt), default value
ev	eV (electronvolt)
ryd	Rydberg units
hz	Hertz
ang	Å (Ångstrom)
nm	nm (nanometer)

Table 3.6: x-axis units for plot type dem:

bin	Bin number
kev	keV (kilo-electronvolt), default value
ev	eV (electronvolt)
ryd	Rydberg units
k	K (Kelvin)
mk	MK (Mega-Kelvin)

Table 3.7: y-axis units for plot type model, spec:

	Photon numbers:
cou	Photons $\mathrm{m^{-2}\,s^{-1}\,bin^{-1}}$
kev	Photons $\mathrm{m^{-2}\,s^{-1}\,keV^{-1}}$ (default value)
ev	Photons $\mathrm{m^{-2}\,s^{-1}\,eV^{-1}}$
ryd	Photons $\mathrm{m^{-2}\,s^{-1}\,Ryd^{-1}}$
hz	Photons $\mathrm{m^{-2}\,s^{-1}\,Hz^{-1}}$
ang	Photons $\mathrm{m^{-2}\,s^{-1}}$ Å^{-1}
nm	Photons $\mathrm{m^{-2}\,s^{-1}\,nm^{-1}}$
	Energy units:
wkev	$\mathrm{W\,m^{-2}\,keV^{-1}}$
wev	$\mathrm{W\,m^{-2}\,eV^{-1}}$
wryd	$\mathrm{W\,m^{-2}\,Ryd^{-1}}$
whz	$\mathrm{W\,m^{-2}\,Hz^{-1}}$
wang	$\mathrm{W\,m^{-2}}$ Å^{-1}
wnm	$\mathrm{W\,m^{-2}\,nm^{-1}}$
jans	Jansky (10^{-26} $\mathrm{W\,m^{-2}\,Hz^{-1}}$
	νF_ν units:
iw	$\mathrm{W\,m^{-2}}$
ij	Jansky Hz (10^{-26} $\mathrm{W\,m^{-2}}$)
	Various:
bin	Bin number

Table 3.8: y-axis units for plot type data:

bin	Bin number
cou	Counts
cs	Counts s^{-1}
kev	Counts s^{-1} keV^{-1} (default value)
ev	Counts s^{-1} eV^{-1}
ryd	Counts s^{-1} Ryd^{-1}
hz	Counts s^{-1} Hz^{-1}
ang	Counts s^{-1} $Å^{-1}$
nm	Counts s^{-1} nm^{-1}
fkev	Counts m^{-2} s^{-1} keV^{-1}
fev	Counts m^{-2} s^{-1} eV^{-1}
fryd	Counts m^{-2} s^{-1} Ryd^{-1}
fhz	Counts m^{-2} s^{-1} Hz^{-1}
fang	Counts m^{-2} s^{-1} $Å^{-1}$
fnm	Counts m^{-2} s^{-1} nm^{-1}

Table 3.9: y-axis units for plot type chi:

bin	Bin number
cou	Counts
cs	Counts s^{-1}
kev	Counts s^{-1} keV^{-1} (default value)
ev	Counts s^{-1} eV^{-1}
ryd	Counts s^{-1} Ryd^{-1}
hz	Counts s^{-1} Hz^{-1}
ang	Counts s^{-1} $Å^{-1}$
nm	Counts s^{-1} nm^{-1}
fkev	Counts m^{-2} s^{-1} keV^{-1}
fev	Counts m^{-2} s^{-1} eV^{-1}
fryd	Counts m^{-2} s^{-1} Ryd^{-1}
fhz	Counts m^{-2} s^{-1} Hz^{-1}
fang	Counts m^{-2} s^{-1} $Å^{-1}$
fnm	Counts m^{-2} s^{-1} nm^{-1}
dchi	(Observed - Model) / Error (default)
rel	(Observed - Model) / Model

Table 3.10: y-axis units for plot type area:

bin	Bin number
m2	m^2 (default)
cm2	cm^2

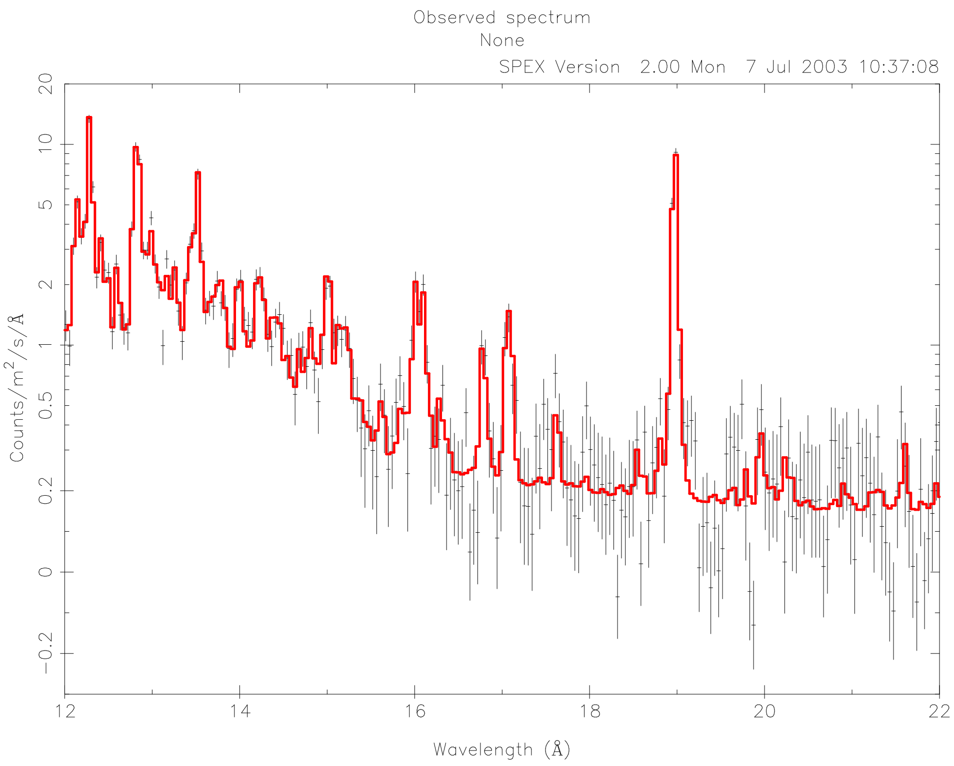


Figure 3.6: Example of the mixed y-axis scale. Simulation of a Chandra LETGS spectrum for a thermal plasma with a temperature of 1 keV.

Table 3.11: y-axis units for plot type fwhm:

bin	Bin number
kev	keV (kilo-electronvolt), default value
ev	eV (electronvolt)
ryd	Rydberg units
hz	Hertz
ang	Å (Ångstrom)
nm	nm (nanometer)
de/e	$\Delta E/E$

Table 3.12: y-axis units for plot type dem: the emission measure Y is defined as $Y \equiv n_{\mathrm{e}} n_{\mathrm{H}} V$ and is expressed in units of $10^{64}\,\mathrm{m}^{-3}$. Here n_{e} is the electron density, n_{e} is the Hydrogen density and V the emitting volume.

bin	Bin number
em	Y (default)
demk	dY/dT, per keV
demd	dY/dT, per K

Chapter 4

Spectral models

4.1 Overview of spectral components

For more information on the definition of spectral components and the different types, see Sect. 1.3. Currently the following models are implemented in SPEX, see Table 4.1.

Table 4.1: Available spectral components

acronym	description
	additive components:
pow	Power law
delt	Delta line
gaus	Gaussian line
bb	Blackbody
mbb	Modified blackbody
dbb	Accretion disk blackbody
cie	Collisional ionisation equilibrium spectrum
neij	Non-equilibrium ionisation spectrum
sed	Sedov adiabatic SNR model
chev	Chevalier adiabatic SNR model with reverse shock
soli	Solinger isothermal SNR model
band	Band isothermal SNR model with reverse shock
pdem	Differential emission measure model, polynomials
cf	Isobaric cooling flow model
wdem	Power law differential emission measure with high T cut-off
dem	Differential emission measure model, for DEM analysis
refl	Reflection model of Zycki
file	Table model from file
	multiplicative components, shifts:
reds	Redshift model
	multiplicative components, convolutions:
vgau	Gaussian velocity profile
vblo	Square velocity profile
vpro	Arbitrary velocity profile (needs input file)
lpro	Spatial profile modeling for RGS (needs input file)
laor	Laor relativistic line profile
	multiplicative components, absorption/transmission:
absm	Morrison & McCammon ISM absorption
euve	EUVE absorption model Rumph et al. (H+He)
hot	SPEX absorption by plasma in CIE
slab	absorption by a slab with adjustable ionic columns
xabs	absorption by a slab in photoionization equilibrium
warm	absorption by a slab with continuous distribution in ionization
knak	transmission piecewise power law

4.2 Pow: Power law model

The power law spectrum as given here is a generalization of a simple power law with the possibility of a break, such that the resultant spectrum in the $\log F - \log E$ plane is a hyperbola. The spectrum is given by:

$$F(E) = AE^{-\Gamma}e^{\eta(E)}, \tag{4.1}$$

where E is the photon energy in keV, F the photon flux in units of 10^{44} ph s^{-1} keV^{-1}, and the function $\eta(E)$ is given by

$$\eta(E) = \frac{r\eta + \sqrt{r^2\xi^2 + b^2(1-r^2)}}{1-r^2}, \tag{4.2}$$

with $\xi \equiv \ln(E/E_0)$, and E_0, r and b adjustable parameters. For high energies, ξ becomes large and then η approaches $2r\xi/(1-r^2)$, while for low energies ξ approaches $-\infty$ and as a consequence η goes to zero. Therefore the break $\Delta\Gamma$ in the spectrum is $\Delta\Gamma = 2r\xi/(1-r^2)$. Inverting this we have

$$r = \frac{\sqrt{1+(\Delta\Gamma)^2} - 1}{|\Delta\Gamma|}. \tag{4.3}$$

The parameter b gives the distance (in logarithmic units) from the interception point of the asymptotes of the hyperbola to the hyperbola. A value of $b = 0$ therefore means a sharp break, while for larger values of b the break gets smoother.
The simple power law model is obtained by having $\Delta\Gamma = 0$, or the break energy E_0 put to a very large value.

WARNING: *By default, the allowed range for the photon index Γ is (-10,10). If you manually increase the limits, you may run the risk that SPEX crashes due to overflow for very large photon indices.*

WARNING: *Note the sign of Γ: positive values correspond to spectra decreasing with energy. A spectrum with $\Delta\Gamma > 0$ therefore steepens/softens at high energies, for $\Delta\Gamma < 0$ it hardens.*
As an extension, we allow for a different normalisation, namely the integrated luminosity L in a given energy band $E_1 - -E_2$. If you choose this option, the parameter "type" should be set to 1. The reason for introducing this option is that in several cases you may have a spectrum that does not include energies around 1 keV. In that case the energy at which the normalisation A is determined is outside your fit range, and the nominal error bars on A can be much larger than the actual flux uncertainty over the fitted range. Note that the parameters E_1 and E_2 act independently from whatever range you specify using the "elim" command. Also, the luminosity is purely the luminosity of the power law, not corrected for any transmission effects that you may have specified in other spectral components.

WARNING: *When you do spectral fitting, you* **must** *keep either A or L a fixed parameter! The other parameter will then be calculated automatically whenever you give the calculate or fit command. SPEX does not check this for you! If you do not do this, you may get unexpected results / crashes.*

WARNING: *The conversion factor between L and A is calculated numerically and not analytically (because of the possible break). In the power law model, photon fluxes above the nominal limit (currently $e^{34} = 5.8 \times 10^{14}$ in unscaled units) are put to the maximum value in order to prevent numerical overflow. This implies that you get inaccurate results for low energies, for example for a simple power law with $\Gamma = 2$ the results (including conversion factors) for $E < 10^{-7}$ keV become inaccurate.*
The parameters of the model are:

norm - Normalisation A of the power law, in units of 10^{44} $\mathrm{ph\,s^{-1}\,keV^{-1}}$ at 1 keV. Default value: 1.
gamm - The photon index Γ of the spectrum. Default value: 2.
dgam - The photon index break $\Delta\Gamma$ of the spectrum. Default value: 0. and frozen. If no break is desired, keep this parameter 0 (and frozen!).
e0 - The break energy E_0 (keV) of the spectrum. Default value: 10^{10} and frozen.
b - Smoothness of the break b. Default: 0.
type - Type of normalisation. Type= 0 (default): use A; type= 1: use L.
elow - E_1 in keV, the lower limit for the luminosity calculation. Default value: 2 keV.
eupp - E_2 in keV, the upper limit for the luminosity calculation. Default value: 10 keV. Tkae care that $E_2 > E_1$.
lum - Luminosity L between E_1 and E_2, in units of 10^{30} W.

4.3 Delt: delta line model

The delta line model is a simple model for an infinitely narrow emission line.
The spectrum is given by:

$$F(E) = A\delta(E - E_0), \tag{4.4}$$

where E is the photon energy in keV, F the photon flux in units of 10^{44} ph s^{-1} keV^{-1}, E_0 is the line energy of the spectral line (in keV) and A is the line normalisation (in units of 10^{44} ph s^{-1}). The total line flux is simply given by A.
To ease the use for dispersive spectrometers (gratings) there is an option to define the wavelength instead of the energy as the basic parameter. The parameter *type* determines which case applies: type=0 (default) corresponding to energy, type=1 corresponding to wavelength units.

WARNING: *When changing from energy to wavelength units, take care about the frozen/thawn status of the line centroid and FWHM*

WARNING: *You need to do a "calc" or "fit" command to get an update of the wavelength (for type=0) or energy (type=1).*

The parameters of the model are:
norm - Normalisation A, in units of 10^{44} ph s^{-1}. Default value: 1.
e - The line energy E_0 in keV. Default value: 6.4 keV.
type - The type: 0 for energy units, 1 for wavelength units.
w - The line wavelength λ in Å. Default value: 20 Å.

4.4 Gaus: Gaussian line model

The Gaussian line model is the simplest model for a broadened emission line.
The spectrum is given by:

$$F(E) = Ae^{(E - E_0)^2/2\sigma^2}, \tag{4.5}$$

where E is the photon energy in keV, F the photon flux in units of 10^{44} ph s^{-1} keV^{-1}, E_0 is the average line energy of the spectral line (in keV) and A is the line normalisation (in units of 10^{44} ph s^{-1}). The total line flux is simply given by A. Further σ is the Gaussian width, which is related to the full width at half maximum $FWHM$ by $FWHM = \sigma\sqrt{\ln 256}$ or approximately $FWHM = 2.3548\sigma$.
To ease the use for dispersive spectrometers (gratings) there is an option to define the wavelength instead of the energy as the basic parameter. The parameter *type* determines which case applies: type=0 (default) corresponding to energy, type=1 corresponding to wavelength units.

Warning: *Do not confuse σ with $FWHM$ when interpreting your fit results with other data.*

Warning: *When changing from energy to wavelength units, take care about the frozen/thawn status of the line centroid and FWHM*

Warning: *You need to do a "calc" or "fit" command to get an update of the wavelength (for type=0) or energy (type=1).*

The parameters of the model are:
norm - Normalisation A, in units of 10^{44} ph s^{-1}. Default value: 1.
e - The line energy E_0 in keV. Default value: 6.4 keV.
fwhm - The line $FWHM$, in keV.
type - The type: 0 for energy units, 1 for wavelength units.
w - The line wavelength λ in Å. Default value: 20 Å.
awid - The line $FWHM$, in Å.

4.5 Bb: Blackbody model

The surface *energy* flux of a blackbody emitter is given by

$$F_\nu = \pi B_\nu = \frac{2\pi h\nu^3/c^2}{e^{h\nu/kT}-1} \tag{4.6}$$

(cf. Rybicky & Lightman [1979], chapter 1). We transform this into a spectrum with energy units (conversion from Hz to keV) and obtain for the total *photon* flux:

$$S(E)\mathrm{d}E = 2\pi c[10^3 e/hc]^3 \frac{E^2}{e^{E/T}-1}\mathrm{d}E \tag{4.7}$$

where now E is the photon energy in keV, T the temperature in keV and e is the elementary charge in Coulomb. Inserting numerical values and multiplying by the emitting area A, we get

$$N(E) = 9.883280 \times 10^7\, E^2 A/(e^{E/T}-1) \tag{4.8}$$

where N(E) is the photon spectrum in units of 10^{44} photons/s/keV and A the emitting area in 10^{16} m^2.
The parameters of the model are:
norm - Normalisation A (the emitting area, in units of 10^{16} m^2. Default value: 1.
t - The temperature T in keV. Default value: 1 keV.

4.6 Mbb: Modified blackbody model

This model describes the spectrum of a blackbody mdified by coherent Compton scattering. This is in several instances a much better description than a simple blackbody (for example accretion disk spectra of AGN). The physical background is described for example by Rybicki & Lightman ([1979]), pages 218–219. The formulae that we use here with a derivation are given by Kaastra & Barr ([1989]). From that work we derive the spectrum (10^{44} photons/s/keV):

$$N(E) = 1358. \frac{AE^{0.25}}{e^{E/T}(e^{E/T}-1)} \tag{4.9}$$

where E is the photon energy in keV, T the temperature in keV and A the normalisation in units of 10^{26} $\mathrm{m}^{0.5}$, defined by

$$A = n_{\mathrm{e}}^{0.5} O \tag{4.10}$$

with n_{e} the electron density (units: 10^{20} m^{-3}) and O the emitting source area (units: 10^{16} m). The parameters of the model are:
norm - Normalisation A (in units of 10^{26} $\mathrm{m}^{0.5}$). Default value: 1.
t - The temperature T in keV. Default value: 1 keV.

4.7 Dbb: Disk blackbody model

We take the model for a standard Shakura-Sunyaev accretion disk. The radiative losses from such a disk are given by

$$Q = \frac{3GM\dot{M}(1-\sqrt{r_i/r})}{8\pi r^3}, \tag{4.11}$$

where Q is the loss term in W m^{-2} at radius r, M the mass of the central object, $\dot{M}$ the accretion rate through the disk and r_i the inner radius. If this energy loss is radiated as a blackbody, we have

$$Q = \sigma T^4 \tag{4.12}$$

with σ the constant of Stefan-Boltzmann and $T(r)$ the local temperature of the blackbody. The total spectrum of such a disk is then obtained by integration over all radii. We do this integration numerically. Note that for large r, $T \sim r^{-3/4}$.

WARNING: *A popular disk model (diskbb in XSPEC) assumes this temperature dependence over the full disk. However, we correct it using the factor* $(1-\sqrt{r_i/r})$ *in* Q *which corresponds to the torque-free condition at the inner boundary of the disk.*

The photon spectrum of the disk is now given by

$$N(E) = \frac{8\pi^2 E^2 r_i^2 \cos i}{h^3 c^2} f_d(E/kT_i, r_o/r_i), \tag{4.13}$$

where i is the inclination of the disk (0 degrees for face-on disk, 90 degrees for an edge-on disk), E the photon energy, r_o the outer edge of the disk, and T_i is defined by

$$T_i^4 = 3GM\dot{M}/8\pi r_i^3 \sigma \tag{4.14}$$

and further the function $f_d(y, r)$ is defined by

$$f_d(y, r) = \int_1^r \frac{x\,\mathrm{d}x}{e^{y/\tau} - 1} \tag{4.15}$$

where $\tau(x)$ is defined by $\tau^4(x) = (1 - 1/\sqrt{x})/x^3$.

In addition to calculating the spectrum, the model also allows to calculate the average radius of emission R_e at a specified energy E_r. This is sometimes useful for time variability studies (softer photons emerging from the outer parts of the disk).

Given the fit parameters T_i and r_i, using (4.14) it is straightforward to calculate the product $M\dot{M}$. Further note that if $r_i = 6GM/c^2$, it is possible to find both M and $\dot{M}$, provided the inclination angle i is known.

The parameters of the model are:

norm - Normalisation A $(= r_i^2 \cos i)$, in units of 10^{16} m^2. Default value: 1.

t - The nominal temperature T_i in keV. Default value: 1 keV.

ro - The ratio of outer to inner disk radius, r_o/r_i

ener - Energy E_r at which the average radius of emission will be calculated

rav - Average radius R_e of all emission at energy E_r specified by the parameter above. Note that this is not a free parameter, it is calculated each time the model is evaluated.

4.8 Cie: collisional ionisation equilibrium model

This model calculates the spectrum of a plasma in collisional ionisation equilibrium (CIE). It consists essentially of two steps, first a calculation of the ionisation balance and then the calculation of the X-ray spectrum. The basis for this model is formed by the mekal model, but several updates have been included.

4.8.1 Temperatures

Some remarks should be made about the temperatures. SPEX knows three temperatures that are used for this model.

First there is the electron temperature T_{e}. This is usually referrred to as "the" temperature of the plasma. It determines the continuum shape and line emissivities and the resulting spectrum is most sensitive to this.

Secondly, there is the ion temperature T_{i}. This is only important for determining the line broadening, since this depends upon the thermal velocity of the ions (which is determined both by T_{i} and the atomic mass of the ion). Only in high resolution spectra the effects of the ion temperature can be seen.

Finally, we have introduced here the ionization balance temperature T_{b} that is used in the determination of the ionization equilibrium. It is the temperature that goes into the calculation of ionization and recombination coefficients. In equilibrium plasmas, the ratio $R_{\mathrm{b}} \equiv T_{\mathrm{b}}/T_{\mathrm{e}}$ is 1 by definition. It is unphysical to have R_{b} not equal to 1. Nevertheless we allow for the possibility of different values of R_{b}, in order to mimick out of equilibrium plasmas. For $R_{\mathrm{b}} < 1$, we have an ionizing plasma, for $R_{\mathrm{b}} > 1$ a recombining plasma. Note that for ionizing plasmas SPEX has also the nei model, which takes into account explicitly the effects of transient (time dependent) ionization.

It is also possible to mimick the effects of non-isothermality in a simple way. SPEX allows for a Gaussian emission measure distribution

$$Y(x) = \frac{Y_0}{\sqrt{2\pi\sigma_T}} e^{-(x-x0)^2/2\sigma_T^2} \tag{4.16}$$

where Y_0 is the total, integrated emission measure, $x \equiv \log T$ and $x_0 \equiv \log T_0$ with T_0 the average temperature of the plasma (this is entered at the "T" value in SPEX). Usually (default) $\sigma_T = 0$ and in that case the normal isothermal spectrum is chosen. Note that for larger values of σ_T the cpu time may become larger due to the fact that the code has to evaluate many isothermal spectra.

4.8.2 Line broadening

Apart from line broadening due to the thermal velocity of the ions (caused by $T_{\mathrm{i}} > 0$) it is also possible to get line broadening due to (micro)turbulence. In this case the line broadening is determined by v_{micro}, which is given by $v_{\mathrm{micro}} \equiv \sqrt{2}\sigma_v$ with σ_v the Gaussian velocity dispersion in the line of sight. Thus, without thermal broadening the FWHM of the line would be 1.6651 times v_{micro}.

4.8.3 Density effects

It is also possible to vary the electron density n_{e} of the plasma. This does not affect the overall shape of the spectrum (i.e., by changing n_{e} only SPEX still uses the previous value of the emission measure $Y \equiv n_{\mathrm{H}} n_{\mathrm{e}} V$), but spectral lines that are sensitive to the electron density will get different intensities. Usually this occurs for higher densities, for example under typical ISM conditions ($n_{\mathrm{e}} = 1$ cm^{-3}) this is not an important effect.

4.8.4 Non-thermal electron distributions

The effects of non-thermal electron distribution on the spectrum can be included. See Sect. 4.30 for more details.

4.8.5 Abundances

The abundances are given in Solar units. Which set of solar units is being used can be set using the "var abun" command (see Sect. 2.11). For spectral fitting purposes it is important to distinguish two situations.

In the first case there is a strong thermal continuum. Since in most circumstances the continuum is determined mainly by hydrogen and helium, and the X-ray lines are due to the other elements, the line to continuum ratio is a direct measurement of the metal abundances compared to H/He. In this situation, it is most natural to have the hydrogen abundance fixed and allow only for fitting of the other abundances (as well as the emission measure).

In the other case the thermal continuum is weak, but there are strong spectral lines. Measuring for example the Fe abundance will give large formal error bars, not because the iron lines are weak but because the continuum is weak. Therefore, all abundances will get rather large error bars, and despite the fact of strong O and Fe lines, the formal error bars on the O/Fe ratio becomes large. This can be avoided by choosing another reference element, preferentially the one with the strongest lines (for example Fe). Then the O/Fe ratio will be well constrained, and it is now only the H abundance relative to Fe that is poorly constrained. In this case it is important to keep the nominal abundance of the reference element to unity. Also keep in mind that in general we write for the normalisation $n_{\mathrm{e}} n_{\mathrm{X}} V$ in this case; when the reference element is H, you mays substitute X=H; however when it is another element, like O, the normalisation is still the product of $n_{\mathrm{e}} n_{\mathrm{X}} V$ where X stands for the fictitious hydrogen density derived from the solar O/H ratio.

Example: suppose the Solar O abundance is 1E-3 (i.e. there is 0.001 oxygen atom per hydrogen atom in the Sun). Take the reference atom oxygen ($Z = 8$). Fix the oxygen abundance to 1. Fit your spectrum with a free hydrogen abundance. Suppose the outcome of this fit is a hydrogen abundance of 0.2 and an emission measure 3 (in SPEX units). This means $n_{\mathrm{e}} n_{\mathrm{X}} V = 3 \times 10^{64}$ m^{-3}. Then the "true" emission measure $n_{\mathrm{e}} n_{\mathrm{H}} V = 0.2 \times 3 \times 10^{64}$ m^{-3}. The nominal oxygen emission measure is $n_{\mathrm{e}} n_{\mathrm{O}} V = 0.001 \times 3 \times 10^{64}$ m^{-3}, and nominally oxygen would have 5 times overabundance as compared to hydrogen, in terms of solar ratios.

4.8.6 Parameter description

WARNING: *When you make the temperature too low such that the plasma becomes completely neutral, the model will crash. This is because in that case the electron density becomes zero, and the emission measure is undefined. The nominal temperature limits that are implemented in SPEX usually avoid that condition, but the results may depend somewhat upon the metallicity because of the charge exchange processes in the ionisation balance.*

WARNING: *In high resolution spectra, do not forget to couple the ion temperature to the electron temperature, as otherwise the ion temperature might keep its default value of 1 keV during spectral fitting and the line widths may be wrong.*

WARNING: *Some people use instead of the emission measure $Y \equiv n_{\mathrm{H}} n_{\mathrm{e}} V$, the quantity $Y' = n_{\mathrm{e}}^2 V$ as normalisation. This use should be avoided as the emission is proportional to the product of electron and ion densities, and therefore use of Y' makes the spectrum to depend nonlinear on the elemental abundances (since an increase in abundances also affects the $n_{\mathrm{e}}/n_{\mathrm{H}}$ ratio).*

The parameters of the model are:

norm - the normalisation, which is the emission measure $Y \equiv n_{\mathrm{H}} n_{\mathrm{e}} V$ in units of 10^{64} m^{-3}, where n_{e} and n_{H} are the electron and Hydrogen densities and V the volume of the source. Default value: 1.

t - the electron temperature T_{e} in keV. Default value: 1.

sig - the width σ_T in logarithmic units. Default value: 0. (no temperature distribution i.e. isothermal)

ed - the electron density n_{e} in units of 10^{20} m^{-3} (or 10^{14} cm^{-3}). Default value: 10^{-14}, i.e. typical ISM conditions, or the low density limit.

it - the ion temperature T_{i} in keV. Default value: 1

rt - the ratio of ionization balance to electron temperature, $R_{\mathrm{b}} = T_{\mathrm{b}}/T_{\mathrm{e}}$ in keV. Default value: 1.

vmic - the (micro)turbulent velocity v_{micro}, in km/s. Default value 0.

ref - reference element. Default value 1 (hydrogen). See above for more details. The value corresponds to the atomic number of the reference element.

01 - Abundance of hydrogen (H, Z=1)
in solar units. Default 1. 02 - Abundance of helium (He, Z=2)
in solar units. Default 1.

...

30 - Abundance of zinc (Zn, Z=30)
in solar units. Default 1. file - Filename for the nonthermal electron distribution. If not present, nonthermal effects are not taken into account (default).

4.9 Neij: Non-Equilibrium Ionisation Jump model

This model calculates the spectrum of a plasma in non-equilibrium ionisation (NEI). For more details about NEI calculations, see Sect. 4.31.

The present model calculates the spectrum of a collisional ionisation equilibrium (CIE) plasma with uniform electron density n_e and temperature T_1, that is instantaneously heated or cooled to a temperature T_2. It then calculates the ionisation state and spectrum after a time t. Obviously, if t becomes large the plasma tends to an equilibrium plasma again at temperature T_2.

The parameters of the model are:

t1 - Temperature T_1 before the sudden change in temperature, in keV. Default: 0.002 keV.

t2 - Temperature T_2 after the sudden change in temperature, in keV. Default: 1 keV.

u - Ionization parameter $U = n_e t$, in 10^{20} m^{-3} s. Default: 10^{-4}.

The following parameters are the same as for the cie-model:

ed - Electron density in 10^{20} m^{-3}

it - Ion temperature in keV

vmic - Micro turbulent velocity in km/s

ref - Reference element

01. . . 30 - Abundances of H to Zn

file - Filename for the nonthermal electron distribution

4.10 Dem: Differential Emission Measure model

This model calculates a set of isothermal CIE spectra, that can be used later in one of the DEM analysis tools (see the "dem" commands of the syntax). The CIE spectra are evaluated for a logarithmic grid of temperatures between T_1 and T_2, with n bins.

WARNING: *For the DEM methods to work, the dem model is the only allowed additive component that can be present. No other additive components are allowed. But of course the spectrum of the dem model may be modified by applying any combination of multiplicative models (redshifts, absorptions, line broadening, etc.).*

WARNING: *Because of the above, do not use the "fit" commands when you have a dem model. If you really need to fit, use the pdem model instead.*

WARNING: *In general, the spacing between the different temperature components should not be smaller than 0.10 in* $\log T$. *Smaller step sizes will produce unstable solutions.*

The parameters of the model are:
t1 - Lower temperature T_1 in keV. Default value: 0.001 keV.
t2 - Upper temperature T_2 in keV. Default value: 100 keV.
nr - Number of temperature bins. Default value: 64.

The following parameters are the same as for the cie-model:
ed - Electron density in 10^{20} m^{-3}
it - Ion temperature in keV
vmic - Micro turbulent velocity in km/s
ref - Reference element
01...30 - Abundances of H to Zn
file - Filename for the nonthermal electron distribution

4.11 Pdem: DEM models

4.11.1 Short description

The pdem model is intended to be used for differential emission measure analysis, simultaneous with fitting of abundances etc. of an optically thin plasma.
It works as follows. The user gives a a number of temperature grid points n, a minimum temperature T_1, a maximum temperature T_n, a total emission measure Y and relative contributions $y_1 \ldots y_n$. SPEX assumes that the grid points between T_1 and T_n are distributed logarithmically. The relative contributions $y_1 \ldots y_n$ represents the values of $\mathrm{d}Y/\mathrm{d}\ln T$ (note the logarithm!) at the grid points. SPEX then interpolates in the $\log T_i - \log y_i$ space on a finer grid using splines. That temperature grid on which the data are being evaluated has a fine mesh: step size is about 0.10 in $\log T$ (finer is not usefull because uniqueness is no longer guaranteed), with the additional constraint that the number of mesh points is at least n and not larger than 64 (practical limit in order to avoid excessive cpu-time). The emission measure distribution is then simply scaled in such a way that its sum over the fine mesh equals the total emission measure Y that went into the model.

WARNING: *At least one of the y_i values should be kept frozen during fitting, when Y is a free parameter, otherwise no well defined solution can be obtained! If Y is fixed, then all y_i can be free.*
The parameters of the model are:
norm - Normalisation, i.e. total integrated emission measure Y in units of 10^{64} m^{-3}
t1 - Lower temperature T_1 in keV
tn - Upper temperature T_n in keV
npol - Number of temperature grid points n; minimum value 2, maximum value 8
y1 - Relative contribution y_1 at T_1
y2 - Relative contribution y_2 at T_2

...
y8 - Relative contribution y_8 at T_8; note that the higher contributions y_i are neglected if $i > n$
The following parameters are the same as for the cie-model:
ed - Electron density in 10^{20} m^{-3}
it - Ion temperature in keV
vmic - Micro turbulent velocity in km/s
ref - Reference element
01...30 - Abundances of H to Zn
file - Filename for the nonthermal electron distribution

Note that the true emission measure on the finer mesh can be displayed by using the "ascdump term # # dem" command; you will get a list of temperature (keV) versus emission measure.

4.12 Cf: isobaric cooling flow differential emission measure model

This model calculates the spectrum of a standard isobaric cooling flow. The differential emission measure distribution $\mathrm{d}Y(T)/\mathrm{d}T$ for the isobaric cooling flow model can be written as

$$D(T) \equiv \mathrm{d}Y(T)/\mathrm{d}T = \frac{5\dot{M}k}{2\mu m_{\rm H}\Lambda(T),} \tag{4.17}$$

where $\dot{M}$ is the mass deposition rate, k is Boltzmann's constant, μ the mean molecular weight (0.618 for a plasma with 0.5 times solar abundances), $m_{\rm H}$ is the mass of a hydrogen atom, and $\Lambda(T)$ is the cooling function. We have calculated the cooling function Λ using our own SPEX code for a grid of temperatures and for 0.5 times solar abundances. The spectrum is evaluated by integrating the above differential emission measure distribution between a lower temperature boundary T_1 and a high temperature boundary T_n. We do this by creating a temperature grid with n bins and calculating the spectrum for each temperature.

WARNING: *Take care that n is not too small in case the relevant temperature is large; on the other hand if n is large, the computational time increases. Usually a spacing with temperature differences between adjacent bins of 0.1 (in* log*) is sufficient and optimal.*

WARNING: *The physical relevance of this model is a matter of debate*

The parameters of the model are:
norm - The mass deposition rate $\dot{M}$ in $\mathrm{M}_\odot\,\mathrm{yr}^{-1}$
t1 - Lower temperature cut-off temperature T_1. Default: 0.1 keV.
tn - Upper temperature cut-off temperature T_n. Default: 1 keV.
nr - Number of temperature bins n used in the integration. Default value: 16
p - Slope $p = 1/\alpha$. Default: 0.25 ($\alpha = 4$).
cut - Lower temperature cut-off, in units of $T_{\rm max}$. Default value: 0.1.

The following parameters are the same as for the cie-model:
ed - Electron density in 10^{20} m^{-3}
it - Ion temperature in keV
vmic - Micro turbulent velocity in km/s
ref - Reference element
01...30 - Abundances of H to Zn
file - Filename for the nonthermal electron distribution

4.13 Wdem: power law differential emission measure model

This model calculates the spectrum of a power law distribution of the differential emission measure distribution. It appears to be a good empirical approximation for the spectrum in cooling cores of clusters of galaxies. It was first introduced by Kaastra et al. ([2004]), and is defined as follows:

$$\frac{\mathrm{d}Y}{\mathrm{d}T} = \begin{cases} 0 & \text{if } T \le T_{\mathrm{min}}; \\ cT^{\alpha} & \text{if } T_{\mathrm{min}} < T < T_{\mathrm{max}}; \\ 0 & \text{if } T \ge T_{\mathrm{max}}. \end{cases} \tag{4.18}$$

Here Y is the emission measure $Y \equiv n_{\mathrm{H}} n_{\mathrm{e}} V$ in units of 10^{64} m^{-3}, where n_{e} and n_{H} are the electron and Hydrogen densities and V the volume of the source.
For $\alpha \to \infty$, we obtain the isothermal model, for large α a steep temperature decline is recovered while for $\alpha = 0$ the emission measure distribution is flat. Note that Peterson et al. ([2003]) use a similar parameterisation but then for the differential luminosity distribution). In practice, we have implemented the model (4.18) by using the integrated emission measure Y_{tot} instead of c for the normalisation, and instead of α its inverse $p = 1/\alpha$, so that we can test isothermality by taking $p = 0$. The emission measure distribution for the model is binned to bins with logarithmic steps of 0.10 in $\log T$, and for each bin the spectrum is evaluated at the emission measure averaged temperature and with the integrated emission measure for the relevant bin (this is needed since for large α the emission measure weighted temperature is very close to the upper temperature limit of the bin, and not to the bin centroid). Instead of using T_{min} directly as the lower temperature cut-off, we use a scaled cut-off c such that $T_{\mathrm{min}} = cT_{\mathrm{max}}$.

WARNING: *Take care that $c < 1$. For $c = 1$, the model becomes isothermal, regardless tha value of α!*
The parameters of the model are:
norm - Integrated emission measure between T_{min} and T_{max}
t0 - Maximum temperature T_{max}, in keV. Default: 1 keV.
p - Slope $p = 1/\alpha$. Default: 0.25 ($\alpha = 4$).
cut - Lower temperature cut-off c , in units of T_{max}. Default value: 0.1.

The following parameters are the same as for the cie-model:
ed - Electron density in 10^{20} m^{-3}
it - Ion temperature in keV
vmic - Micro turbulent velocity in km/s
ref - Reference element
01...30 - Abundances of H to Zn
file - Filename for the nonthermal electron distribution

4.14 Refl: Reflection model

This model was kindly provided by Piotr Zycki. The related programs in XSPEC are FELIPL ans FERFSCHW. The first one gives the reflected continuum (i.e. PEXRAV/PEXRIV) plus the line with correct energy and intensity, the second one is the first one with the relativistic smearing added. Both are combined into the single refl model in SPEX.

It is a model of reflection from a constant density X-ray illuminated atmosphere. It computes the Compton-reflected continuum (cf. Magdziarz & Zdziarski ([1995]) and the iron K alpha line (cf. Zycki & Czerny [1994]), as described in Zycki, Done & Smith ([1999]). In addition it can be convolved with a relativistic diskline model (for Schwarzschild geometry).

Chemical elements taken into account in this model are H, He, C, N, O, Ne, Mg, Si, S and Fe. The standard abundances are taken from Morrison & McCammon ([1983]).

The incoming spectrum is characterized by:

$$N_i(E) = AE^{-\Gamma} \exp\left[-E/E_c\right], \tag{4.19}$$

where E is the photon energy in keV, $N_i(E)$ the number of photons per per second per keV, Γ is the photon index and E_c a cut-off energy. The normalisation A is in units of 10^{44} photons $\mathrm{s}^{-1}\,\mathrm{keV}^{-1}$ at 1 keV, just the same as in the standard power law model. The total spectrum $N(E)$ is then given by

$$N(E) = N_i(E) + sR(E), \tag{4.20}$$

where $R(E)$ is the reflected spectrum and s is a scaling factor.

The parameters of the model are:

norm - Normalisation A of the power law

gamm - The photon index Γ of the ionising spectrum

ecut - The cut-off energy (keV) of the ionising spectrum. If no cut-off is desired, take this parameter zero (and keep it frozen!).

pow - If pow=1, the incoming power law is added to the spectrum (default); if pow=0, only the reflected spectrum is given

disk - If disk=1, the spectrum will be convolved with an accretion disk profile (default); if disk=0, this is not done

fgr - Full general relativity used (default, for fgr=1)

t - Temperature of the reflector (disk) in keV

xi - Ionisation parameter $\xi = L/nr^2$ in the usual (c.g.s. based) units of 10^{-9} W m

abun - The abundance of all metals excluding H and He, in solar units

feab - The iron abundance with respect to the other metals

cosi - The cosine of the inclination angle of the disk. $\cos i = 0$ ($i = \pi/2$) corresponds to edge-on

scal - Scale s for reflection. For an isotropic source above the disk $s = 1$. This value corresponds to seeing equal contributions from the reflected and direct spectra.

q - Emissivity index for the accretion disk; default value 3.

r1 - Inner radius of the disk in units of GM/c^2. Default: 10.

r2 - Outer radius of the disk in units of GM/c^2. Default: 10^4.

4.15 File: model read from a file

This model reads a spectrum from a file. The user inputs a spectrum at n energy bins ($n > 1$). The file containing the spectrum should have the follwing format:

- first line: n
- second line: E_1, S_1
- third line: E_2, S_2
- ...
- last line: E_n, S_n

Here E_i is the energy of a photon in keV, and S_i is the spectrum in units of 10^{44} photons $\mathrm{s}^{-1}\,\mathrm{keV}^{-1}$. All energies E_i must be positive, and the grid must be monotonically increasing (two subsequent energies may not have the same value). Also, the spectrum S_i must be strictly positive (i.e. $S_i = 0$ is not allowed).
SPEX then calculates the spectrum by linear interpolation in the $\log E - \log S$ plane (this is why fluxes must be positive). For $E < E_1$ and $E > E_n$ however, the spectrum is put to zero. Finally, the spectrum is multiplied by the scale factor N prescribed by the user.
The parameters of the model are:
norm - Normalisation factor N.Default value: 1.
file - The file name of the file containing the spectrum

4.16 Spln: spline continuum model

Sometimes the continuum of an X-ray source may be too complex to model with known physical components. A situation like that may be found in AGN continua, which are a complex superposition of hard power law, soft continuum excess, relativistically broadened and "normal" broad lines with a priori unknown line shape, etc., while in addition a superimposed warm absorber may have well defined narrow absorption lines. In that case it might be useful to fit the continuum with an arbitrary profile in order to get first an accurate description of the absorber, and then after having "removed" the absorber try to understand the underlying continuum spectrum.

For these situations the *spln* model introduced here is useful. It allows the user to model the continuum within two boundaries b_1 and b_2 with a cubic spline.

The algorithm works as follows. The user selects the limits b_1 and b_2 as well as the number of grid points n. SPEX then creates a grid $x_1, x_2, \ldots, x_n$ with uniform spacing in b (see below for details). The spectrum at these grid points is contained in the corresponding array $y_1, y_2, \ldots, y_n$. These have the usual units of 10^{44} photons s^{-1} keV^{1} used throughout SPEX, and is the spectrum emitted at the source. The parameters y_i can be adjusted during the spectral fitting, but b_1, b_2 and n and thereby x_i remain fixed. At intermediate points between the x_i, the photon spectrum is determined by cubic spline interpolation on the (x_i, y_i) data pairs. We take a natural spline, i.e. at x_1 and x_n the second derivative of the spline is zero.

Outside of the range b_1–b_2 however, the photon spectrum is put to zero, i.e. no extrapolation is made!

Finally note that we have chosen to define the spline in logarithmic space of y, i.e. the log of the photon spectrum is fit by a spline. This is done in order to guarantee that the spectrum remains non-negative everywhere. However, the y-values listed is the (linear) photon spectrum itself.

There are four different options for the energy grid x_i, indicated by the parameter *type*:

- *type*=1: b_1 is the lower energy in keV, b_2 is the upper energy in keV, and the grid is linear in energy in between.
- *type*=2: b_1 is the lower energy in keV, b_2 is the upper energy in keV, and the grid is logarithmic in energy in between.
- *type*=3: b_1 is the lower wavelength in Å, b_2 is the upper wavelength in Å, and the grid is linear in wavelength in between.
- *type*=4: b_1 is the lower wavelength in Å, b_2 is the upper wavelength in Å, and the grid is logarithmic in wavelength in between.

Note that the logarithmic grids can also be used if one wants to keep a fixed velocity resolution (for broadened line features for example). Further, each time that the model is being evaluated, the relevant values of the x_i grid points are evaluated.

WARNING: *Be aware that if you just set b_1, b_2 and n but do not issue the "calc" command or the "fit" command, the x_i values have not yet been calculated and any listed values that you get with the "par show" command will be wrong. After the first calculation, they are right.*

WARNING: *If at any time you change one of the parameters type, b_1, b_2 or n, the y_i values will not be appropriate anymore as they correspond to the previous set of x_i values.*

The maximum number n of grid points that is allowed is 999, for practical reasons. Should you wish to have a larger number, then you must define multiple *spln* components, each spanning its own (disjunct) b_1–b_2 range.
It should be noted, however, that if you take n very large, spectral fitting may become slow, in particular if you take your initial guesses of the y_i parameters not too close to the true values. The reason for the slowness is two-fold; first, the computational time for each fitting step is proportional to the number of free parameters (if the number of free parameters is large). The second reason is unavoidable due to our spectral fitting algorithm: our splines are defined in log photon spectrum space; if you start for example with the same value for each y_i, the fitting algorithm will start to vary each parameter in turn; if it changes for example parameter x_j by 1, this means a factor of 10; since the neighbouring points (like x_{j-1} and x_{j+1} however are not adjusted in thid step, the photon spectrum has to be drawn as a cubic spline through this very sharp function, and it will show the well-known over-and undershooting at the intermediate x-values between x_{j-1} and x_j and between x_j and x_{j+1}; as the data do not show this strong oscillation, χ^2 will be poor and the fitting algorithm will decide to adjust the parameter y_j only with a tiny amount; the big improvement in χ^2 would only come if *all* values of y_i were adjusted simultaneously.
The parameters of the model are:
type - The parameter *type* defined above; allowed values are 1–4. Default value: 1.
n - The number of grid points n. Should be at least 2.
low - Lower x-value b_1.
upp - Upper x-value b_2. Take care to take $b_2 > b_1$.
x001 - First x-value, by definition equal to b_1. x-values are not allowed to vary (i.e. you may not fit them).
x002 - Second x-value
x003 - Third x-value
... - Other x-values
x999 - last x-value, by definition equal to b_n. If $n < 999$, replace the 999 by the relevant value (for example, if $n = 237$, then the last x-value is x237).
y001 - First y-value. This is a fittable parameter.
x002 - Second y-value
x003 - Third y-value
... - Other y-values
x999 - last y-value. If $n < 999$, replace the 999 by the relevant value (for example, if $n = 237$, then the last y-value is y237).

4.17 Rrc: Radiative Recombination Continuum model

This is a simplified model, aimed to calculate radiative recombination continua for photoionized plasmas. It is a simple, limited shortcut to a more fully complete model for the emission from a recombining plasma.

The user essentially prescribes the emission measure for each ion as well as the radiation temperature, and then this model calculates the *continuum* emission corresponding to this temperature and set of ionic emission measures. Line radiation is *not* taken into account. However, for physical self-consistency, we take account of all three continuum emission components: Bremsstrahlung, two photon emission and free-bound radiation (= the RRC).

The reason for having no physical model to couple the ionic emission measures (contrary to for example the CIE model), is that this allows the user to fit these emission measures without making a priori assumptions about the ionization equilibrium. The user might then combine later the set of derived emission measures with any of his relevant models.

WARNING: *Take care that for too high temperatures, two photon emission might be stronger than the free-bound (RRC) emission!*

WARNING: *Take care that the fit parameters are emission measures of a given ion, while the radiation occurs in the next ion. For example radiative recombination of* O IX *to* O VIII *is proportional to the emission measure of* O IX *($n_{\mathrm{e}}n_{\mathrm{O\,IX}}V$), but produces an emission edge in* O VIII *at 14.22 Å.*

WARNING: *No recombination is possible to neutrals, so therefore there is no* H I, O I *or* Fe I *in this model.*

The parameters of the model are:

t - The temperature T in keV. Default value: 1 keV.
h2 - The H II emission measure $n_{\mathrm{e}}n_{\mathrm{H\,II}}V$ in units of 10^{64} m^{-3}. Default value: 0.
he2 - The He II emission measure $n_{\mathrm{e}}n_{\mathrm{He\,II}}V$ in units of 10^{64} m^{-3}. Default value: 0.
he3 - The He III emission measure $n_{\mathrm{e}}n_{\mathrm{He\,III}}V$ in units of 10^{64} m^{-3}. Default value: 0.

...
ni29 - The Ni XXIX emission measure $n_{\mathrm{e}}n_{\mathrm{Ni\,XXIX}}V$ in units of 10^{64} m^{-3}. Default value: 0.

4.18 Reds: redshift model

This multiplicative model applies a redshift z to an arbitrary additive component. If a photon is emitted at energy E, the redshift model will put it at energy $(1+z)E$. In addition, a time dilatation correction is applied such that the spectrum $S(E)$ (expressed as photons/s per bin) is divided by $1+z$.

However, it is necessary to distinguish two different cases:

case 1 (flag=0): the redshift component can be used to calculate the effects of the cosmological redshift. The cosmological redshift has the effect of the energy shift and the time dilatation as outlined above. Therefore, the above procedure is essentially correct and this is what was used in older versions of SPEX (before version 2.00.10). However, what was overlooked there is that in the determination of the finally observed spectrum/flux as seen on Earth, a division by $4\pi d^2$ is necessary. For this distance d, we took the luminosity distance. However, the factors $(1+z)^2$ are absorbed into the definition of the luminosity distance. Therefore, in the older versions all fluxes were too small by a factor of $(1+z)^2$. Since we want to retain the luminosity distance as the natural distance measure for spectra, it appears necessary to multiply the spectrum as calculated by the original subroutine by a factor of $(1+z)^2$. But in the other case (redshifts caused by the motion of a source at any distance), this re-correction should not be done, reason why we introduce the other option. *in summary, for redshift components corresponding to cosmological redshifts, the option flag=0 (default) must be used.*

case 2 (flag=1): this is the old case for a source with a redshift caused by motion away from us. It should be used for any velocity fields other than the Hubble flow.

The parameters of the model are:

z - Redshift z. Default value: 0.

flag - Flag: 0 (cosmological redshift) or 1 (velocity redshift). Default value: 0.

4.19 Vgau: Gaussian velocity broadening model

This multiplicative model broadens an arbitrary additive component with a Gaussian Doppler profile, characterized by the Gaussian σ. The broadening kernel is therefore proportional to $e^{-(c^2/2\sigma^2)(E-E_0)^2/E_0^2}$.

The parameters of the model are:
sig - Gaussian velocity broadening σ, in km/s. Default value: 1 km/s

4.20 Vblo: Rectangular velocity broadening model

This multiplicative model broadens an arbitrary additive component with a rectangular Doppler profile, characterized by the half-width v. Therefore if a delta-line at energy E is convolved with this component, its full energy width will be $2Ev/c$, and line photons get a rectangular distribution between $E - Ev/c$ and $E + Ev/c$. Of course, any line or continuum emission component can be convolved with the this broadening model.
The parameters of the model are:
vblo - Velocity broadening half-width v, in km/s. Default value: 3000 km/s.

4.21 Vpro: Velocity profile broadening model

This multiplicative model broadens an arbitrary additive component with an arbitrarily shaped Doppler profile, characterized by the half-width v and a profile shape $f(x)$. The resulting spectrum $S_c(E)$ is calculated from the original spectrum $S(E)$ as

$$S_c(E) = \int f(\frac{E - E_0}{E_0}\frac{v}{c})S(E_0)\mathrm{d}E_0. \tag{4.21}$$

The function $f(x)$ must correspond to a probability function, i.e. for all values of x we have

$$f(x) \geq 0 \tag{4.22}$$

and furthermore

$$\int\limits_{-\infty}^{\infty} f(x)\mathrm{d}x = 1. \tag{4.23}$$

In our implementation, we do not use $f(x)$ but instead the cumulative probability density function $F(x)$, which is related to $f(x)$ by

$$F(x) \equiv \int\limits_{-\infty}^{x} f(y)\mathrm{d}y, \tag{4.24}$$

where obviously $F(-\infty) = 0$ and $F(\infty) = 1$. The reason for using the cumulative distribution is that this allows easier interpolation and conservation of photons in the numerical integrations.

If this component is used, you must have a file available which we call here "vprof.dat" (but any name is allowed). This is a simple ascii file, with n lines, and at each line two numbers: a value for x and the corresponding $F(x)$. The lines must be sorted in ascending order in x, and for $F(x)$ to be a proper probability distribution, it must be a non-decreasing function i.e. if $F(x_i) \leq F(x_{i+1})$ for all values of i between 1 and $n-1$. Furthermore, we demand that $F(x_1) \equiv 0$ and $F(x_n) \equiv 1$.

Note that both x and $F(x)$ are dimensionless. The parameter v serves as a scaling parameter for the total amount of broadening. Of course for a given profile there is freedom for the choice of both the x-scale as well as the value of v, as long as e.g. $x_n v$ remains constant. In practice it is better to make a logical choice. For example, for a rectangular velocity broadening (equivalent to the *vblo* broadening model) one would choose $n = 2$ with $x_1 = -1$, $x_2 = 1$, $F(x_1) = 0$ and $F(x_2) = 1$, and then let v do the scaling (this also allows you to have v as a free parameter in spectral fits). If one would instead want to describe a Gaussian profile (for which of course also the *vgau* model exists), one could for example approximate the profile by taking the x-scale in units of the standard deviation; an example with a resolution of 0.1 standard deviation and a cut-off approximation at 5 standard deviations would be x =-5, -4.9, -4.8, ..., 4.8, 4.9, 5.0 with corresponding values for F given by $F = 0$, 0.00000048, 0.00000079, ..., 0.99999921, 0.99999952, 1.

The parameters of the model are:

v - Velocity broadening parameter v, in km/s. Default value: 1 km/s.

file - Ascii character string, containing the actual name of the vprof.dat file

4.22 Lpro: Spatial broadening model

This multiplicative model broadens an arbitrary additive component with an arbitrarily shaped spatial profile, in the case of dispersive spectrometers such as the RGS of XMM-Newton. In many instances, the effects of a spatially extended source can be approximated by making use of the fact that for small off-axis angles θ the expression $\mathrm{d}\lambda/\mathrm{d}\theta$ is almost independent of wavelength λ. This holds for example for the RGS of XMM-Newton (for which $\mathrm{d}\lambda/\mathrm{d}\theta = 0.124/m$ Å/arcmin, with m the spectral order).

We can utilize this for a grating spectrum as follows. Make an image $I(\Delta\theta)$ of your source projected onto the dispersion axis, as a function of the off-axis angle $\Delta\theta$. From the properties of your instrument, this can be transformed into an intensity $I(\Delta\lambda)$ as function of wavelength using $\Delta\lambda = \Delta\theta \mathrm{d}\lambda/\mathrm{d}\theta$. Assume that the spatial profile $I(\theta)$ is only non-zero within a given angular range (i.e. the source has a finite extent). Then we can transform $I(\Delta\lambda)$ into a probability distribution $f(x)$ with $f = 0$ for very small or large values of x (here and further we put $x = \Delta\lambda$).

The resulting spatially convolved spectrum $S_c(\lambda)$ is calculated from the original spectrum $S(\lambda)$ as

$$S_c(\lambda) = \int f(\lambda - \lambda_0) S(\lambda_0) \mathrm{d}\lambda_0. \tag{4.25}$$

The function $f(x)$ must correspond to a probability function, i.e. for all values of x we have

$$f(x) \geq 0 \tag{4.26}$$

and furthermore

$$\int_{-\infty}^{\infty} f(x) \mathrm{d}x = 1. \tag{4.27}$$

In our implementation, we do not use $f(x)$ but instead the cumulative probability density function $F(x)$, which is related to $f(x)$ by

$$F(x) \equiv \int_{-\infty}^{x} f(y) \mathrm{d}y, \tag{4.28}$$

where obviously $F(-\infty) = 0$ and $F(\infty) = 1$. The reason for using the cumulative distribution is that this allows easier interpolation and conservation of photons in the numerical integrations.

If this component is used, you must have a file available which we call here ”vprof.dat” (but any name is allowed). This is a simple ascii file, with n lines, and at each line two numbers: a value for x and the corresponding $F(x)$. The lines must be sorted in ascending order in x, and for $F(x)$ to be a proper probability distribution, it must be a non-decreasing function i.e. if $F(x_i) \leq F(x_{i+1})$ for all values of i between 1 and $n-1$. Furthermore, we demand that $F(x_1) \equiv 0$ and $F(x_n) \equiv 1$.

Note that $F(x)$ is dimensionless. In addition, we allow for two other parameters: a scale factor s and an offset λ_o. Usually, $s = 1$, but if s is varied the resulting broadening scales proportional to s. This is useful if for example one has an idea of the shape of the spatial profile, but wants to measure its width directly from the observed grating spectrum. In addition, the parameter λ_o can be varied if the absolute position of the source is unknown and a small linear shift in wavelength is necessary.

WARNING: *This model can be applied to grating spectra (like RGS), but if you include in your fit also other data (for example EPIC), the same broadening will also be applied to that other* *data SET. Usually this effect is small for not too extended sources, but it is worth taking this into consideration. In future versions we intend to circumvent this problem.*

WARNING: *The above approximation of spatially extended sources assumes that there are no intrinsic spectral variations over the surface area of the X-ray source. Only total intensity variations over the surface area are taken into account. Whenever there are spatial variations in spectral shape (not in intensity) our method is strictly speaking not valid, but still gives more accurate results than a point-source approximation. In principle in those cases a more complicated analysis is needed.*

The parameters of the model are:

s - Scale parameter s, dimensionless. Default value: 1 km/s.

dlam - Offset parameter λ_o, in Å. Default value: 0 Å.

file - Ascii character string, containing the actual name of the vprof.dat file

4.23 Laor: Relativistic line broadening model

This multiplicative model broadens an arbitrary additive component with a relativistic line profile. The relativistic line profile is interpolated from tables produced by Laor ([1991]). In his model, the line transfer function is determined for emission from the equatorial plane around a Kerr black hole, assuming an emissivity law $I(\cos\theta) \sim 1. + 2.06\cos\theta$. The transfer function is calculated at a grid of 35 radii ($r_n = 1600/(n+1)^2$ for $n = 1, \ldots, 35$, in units of GM/c^2), 31 inclinations uniformly spaced in $\cos i$, and 300 frequencies, logarithmically spaced between 0.1 and 5 times the emission frequency, respectively.
Using these numbers, a radial integration is done using an emissivity law

$$I(r) \sim 1/(r^2 + h^2)^{q/2}, \tag{4.29}$$

where h is a characteristic scale height and q an asymptotic power law index (for large r, $I(r) \sim r^{-q}$). The integration is done between an inner radius r_1 and an outer radius r_2. Given the radial grid that is provided in Laor's data, the outer radius extends out to at most $400GM/c^2$.

WARNING: *Of course, any line or continuum emission component can be convolved with the this broadening model; for continuum components the computational time may be very large, however, due to the convolution involved.*

WARNING: *The outer radius cannot be larger than* $400GM/c^2$

The parameters of the model are:
r1 - Inner radius of the disk, in units of GM/c^2. The minimum allowed value is 1.234 (for a maximally rotating Kerr black hole). For a Schwarzschild black hole, one should take $r_i = 6$. Default value: 1.234.
r2 - Outer radius of the disk, in units of GM/c^2. Keep this radius less than 400 (default value)
q - Emissivity slope q as described above. Default value: 2.
h - Emissivity scale height. Default value: 0.
i - Inclination angle (in degrees) of the disk (angle between line of sight and the rotation axis of the disk). Default value: 45 degrees.

4.24 Knak: segmented power law transmission model

The knak model is used to model the transmission of any spectrum using a set of contiguous segments with a power-law transmission at each segment.
This component can be useful for new instruments, in order to test large scale calibration errors (effective area errors for example), but other applications can also be made, of course. For example, if the spectrum of the source has an unknown continuum shape with a superimposed absorption model, it is a good trick to model the continuum by a power law, modify that by a knak model with adjustable or fixed transmission at some relevant energies, and then apply the absorption. An example of this last application can be found in Porquet et al. ([2004]).
The Transmission is given by:

$$T(E) = c_i E^{a_i} \qquad \text{if } E_i < E < E_{i+1} \tag{4.30}$$

for each value of i between 0 and n, the number of grid points. The transmission is 1 for $E < E_1$ and $E > E_2$. Further, instead of using the constants c_i and a_i, we use instead the values of the transmission at E_i, i.e. $T_i \equiv T(E_i) = c_i E_i^{a_i}$. This allows for a continuous connection between neighbouring segments.
Finally, for historical reasons we use here a wavelength grid instead of an energy grid; the corresponding nodes λ_i should therefore be in strictly increasing order.

WARNING: *When applying this model, take care that at least one of the n transmission values is kept fixed (otherwise you may run the risk that your model is unconstrained, for example if the normalisation of the continuum is also a free parameter).*
The parameters of the model are:
n - The number of grid points. Maximum value is 9.
w1 - Wavelength λ_1 (Å) of the first grid point
f1 - Transmission $T(\lambda_1)$ at λ_1.
w2 - Wavelength λ_2 (Å) of the second grid point
f1 - Transmission $T(\lambda_2)$ at λ_2.
 - ...
w9 - Wavelength λ_9 (Å) of the last grid point
f9 - Transmission $T(\lambda_9)$ at λ_9.

Note that if $n < 9$, the values of T_i and λ_i will be ignored for $i > n$.

4.25 Etau: simple transmission model

This model calculates the transmission $T(E)$ between energies $E1$ and E_2 for a simple (often unphysical!) case:

$$T(E) = e^{-\tau(E)}, \tag{4.31}$$

with the optical depth $\tau(E)$ given by:

$$\tau(E) = \tau_0 E^a. \tag{4.32}$$

In addition, we put here $T \equiv 1$ for $E < E_1$ and $E > E_2$, where E_1 and E_2 are adjustable parameters. This allows the user for example to mimick an edge. Note however, that in most circumstances there are more physical models present in SPEX that contain realistic transmissions of edges! If you do not want or need edges, simply keep E_1 and E_2 at their default values.

Note that τ_0 should be non-negative. For $a > 0$ the spectrum has a high-energy cut-off, for $a < 0$ it has a low-energy cut-off, and for $a = 0$ the transmission is flat. The larger the value of a, the sharper the cut-off is.

The model can be used a s a basis for more complicated continuum absorption models. For example, if the optical depth is given as a polynomial in the photon energy E, say for example $\tau = 2 + 3E + 4E^2$, one may define three *etau* components, with τ_0 values of 2, 3, and 4, and indices a of 0, 1 and 2. This is because of the mathematical identity $e^{-(\tau_1+\tau_2)} = e^{-\tau 1} \times e^{-\tau 2}$.

The parameters of the model are:

tau0 - Optical depth τ_0 at $E = 1$ keV. Default value: 1.
a - The index a defined above. Default value: 1.
e1 - Lower energy E_1 (keV). Default value: 10^{-20}.
e2 - Upper energy E_2 (keV). Default value: 10^{20}.
f - The covering factor of the absorber. Default value: 1 (full covering)

4.26 Absm: Morrison & McCammon absorption model

This model calculates the transmission of neutral gas with cosmic abundances as published first by Morrison & McCammon ([1983]). It is a widely used model. The following is useful to know when this model is applied:

1. The location of the absorption edges is not always fully correct; this can be seen with high resdolution grating spectra
2. The model fails in the optical/UV band (i.e., it does not become transparent in the optical)
3. No absorption lines are taken into account
4. The abundances cannot be adjusted

If all the above is of no concern (as is the case in many situations), then the Morrison & McCammon model is very useful. In case higher precision or more detail is needed, the user is advised to use the "hot" model with low temperature in SPEX, which gives the transmission of a slab in collisional ionisation equilibrium.
The parameters of the model are:
nh - Hydrogen column density in 10^{28} m^{-2}. Default value: 10^{-4} (corresponding to 10^{24} m^{-2}, a typical value at low Galactic latitudes).
f - The covering factor of the absorber. Default value: 1 (full covering)

4.27 Euve: EUVE absorption model

This model calculates the transmission of gas with cosmic abundances as published first by Rumph et al. ([1994]). It is a widely used model for the transimission at wavelenghts covered by the EUVE satellite ($\lambda > 70$ Å). As it these wavelengths metals are relatively unimportant, it only takes into account hydrogen and helium, but for these elements it takes into account resonances. However it is not recommended to use the model for harder X-rays, due to the neglect of metals such as oxygen etc.
Otherwise the user is advised to use the "absm" model or the "hot" model with low temperature in SPEX, which gives the transmission of a slab in collisional ionisation equilibrium.
The parameters of the model are:
nh - Hydrogen column density in 10^{28} m^{-2}. Default value: 10^{-4} (corresponding to 10^{24} m^{-2}, a typical value at low Galactic latitudes).
he1 - The He I / H I ratio. Default value: 0.1.
he2 - The He II / H I ratio. Default value: 0.01.
f - The covering factor of the absorber. Default value: 1 (full covering)

4.28 Hot: collisional ionisation equilibrium absorption model

This model calculates the transmission of a plasma in collisional ionisation equilibrium with cosmic abundances.

For a given temperature and set of abundances, the model calculates the ionisation balance and then determines all ionic column densities by scaling to the prescribed total hydrogen column density. Using this set of column densities, the transmission of the plasma is calculated by multiplying the transmission of the individual ions.

The transmission includes both continuum and line opacity. For a description of what is currently in the absorption line database, we refer to Sect. sect:absmodels. You can mimick the transmission of a neutral plasma very easy by putting the temperature to 0.5 eV (0.0005 keV).

WARNING: *For solar abundances, do not take the temperature much lower than 0.0005 keV, because if the plasma is completely neutral, the code will crash; a tiny fraction of ions such as Fe II or Na II will help to keep a few free electrons in the gas without affecting the transmission too much. You can check the ion concentrations by giving an "asc ter ... icon" command.*

The parameters of the model are:

nh - Hydrogen column density in 10^{28} m^{-2}. Default value: 10^{-4} (corresponding to 10^{24} m^{-2}, a typical value at low Galactic latitudes).

t - the electron temperature T_e in keV. Default value: 1.

rt - the ratio of ionization balance to electron temperature, $R_b = T_b/T_e$ in keV. Default value: 1.

The following parameters are common to all our absorption models:

f - The covering factor of the absorber. Default value: 1 (full covering)

v - Root mean square velocity σ_v

rms - Rms velocity σ_b of line blend components

dv - Velocity distance Δv between different blend components

zv - Average systematic velocity v of the absorber

The following parameters are the same as for the cie-model (seer there for a description):

ref - Reference element

01...30 - Abundances of H to Zn

file - Filename for the nonthermal electron distribution

4.29 Absorption models

4.29.1 Introduction

In most astrophysical sitiuations we have to take into account absorption of photons between the emitting region and the observer. Apart from a few standard models like the ones by Morrison & McCammon ([1983], our "absm" model) and Rumph et. al. ([1994], our "euve" model) we have constructed our own absorption models based upon the atomic database used by SPEX. Essentially, we adopt a few steps, which will be described below. First, we produce a set of column densities, in different ways for the diferent absorption models (see Sect. 4.29.3). Next, using a dynamical model for the absorber we calculate its transmission (Sect. 4.29.4). For these calculations, we use our atomic database as described in Sect. 4.29.5.
A basic assumption in all the absorption models is that there is no re-emission, i.e. we look through an absorbing nmedium that has a very small solid angle as seen from the X-ray source. This allows essentially to calculate the transmission simply as $e^{-\tau(E)}$ with $\tau(E)$ the optical depth.

4.29.2 Thomson scattering

The above approach also allows us to include Thomson-scattering in the transmission. Any source photon aimed at the observer but that suffers from Thomson scattering is scattered out of the line of sight and hanec in this approximation is not seen by the observer. We have included not simply the Thomson cross-section σ_T but have taken the Klein-Nishina correction into account (see Rybicki & Lightman ([1979], eqn. 7.5 (exact expression) and 7.6 (approximations)). The evaluation of the exact formula for the cross section is non-trivial, as terms up to the third power in $x = E/m_{\rm e}c^2$ cancel; we have extended the low-energy polynomial approximation (7.6.a) of Rybicki & Lightman by comparing to quadruple precision calculations using the exact formula, and made the following approximation that has a relative accuracy of better than 3×10^{-4} for all energies, when evaluated using single precision arithmetics:

$$\sigma = \begin{cases} \sigma_T(1 - 2x + 5.2x^2 - 13.3x^3 + 32.685x^4) & x < 0.05; \\ 0.75\sigma_T \left[\frac{1+x}{x^3}\left\{\frac{2x(1+x)}{1+2x} - \ln(1+2x)\right\} + \frac{\ln(1+2x)}{2x} - \frac{1+3x}{(1+2x)^2}\right] & 0.05 < x < 5000; \\ 0.375\sigma_T(\ln(2x)+0.5)/x & x > 5000. \end{cases} \tag{4.33}$$

4.29.3 Different types of absorption models

We have a set of spectral models available with different levels of sophistication and applicability, that we list below.

Slab model

The *slab* model calculates the transmission of a slab of material, where all ionic column densities can be chosen independently. This has the advantage that a spectrum can be fit without any knowledge of the ionisation balance of the slab. After a spectral fit has been made, one may try to explain the observed column densities by comparing the with predictions from any model (as calculated by SPEX, Cloudy, XSTAR, ION or any other existing (photo)ionization code).

Xabs model

In the *xabs* model, the ionic column densities are not independent quantities, but are linked through a set of runs using a photo ionization code. See the description of the *xabs* model for more details about this. The relevant parameter is the ionization parameter $\xi = L/nr^2$, with L the source lumninosity, n the hydrogen density and r the distance from the ionizaing source. The advantage of the *xabs* model over the *slab* model is that all relevant ions are taken into account, also those which would be detected only with marginal significance using the *slab* model. In some circumstances, the combined effect of many weak absorption features still can be significant. A disadvantage of the *xabs* model happens of course when the ionization balance of the source is different from the ionization balance that was used to produce the set of runs with the photo ionization code. In that case the *xabs* model may fail to give an acceptable fit, while the *slab* model may perform better.

Warm model

In the *warm* model, we construct a model for a continuous distribution of column density N_{H} as a function of ξ. It is in some sense comparable to the differential emission measure models used to model the emission from multi-temperature gas. Here we have absorption from multi-ionization gas. Depending upon the physics of the source, this may be a better approximation than just the sum of a few *xabs* components. A disadvantage of the model may be (but this also depends upon the physics of the source), that all dynamical parameters for each value of ξ are the same, like the outflow velocity and turbulent broadening. If this appears to be the case in a given source, one may of course avoid this problem by taking multiple, non-overlapping *warm* components.

Hot model

In the *hot* model, we link the different ionic column densities simply by using a collisional ionsation (CIE) plasma. It may be useful in situations where photoionisation is relatively unimportant but the source has a non-negligible optical depth. A special application is of course the case for a low temperature, where it can be used to mimick the absorption of (almost) neutral gas.

Pion model

Finally we have in SPEX the pion model, which does a self-consistent photo ionization calculation of the slab of material.

4.29.4 Dynamical model for the absorbers

For each of the absorption models described in the previous section, we have the freedom to prescribe the dynamics of the source. The way we have implemented this in SPEX is described below.
The transmission $T(\lambda)$ of the slab is simply calculated as

$$T(\lambda) = \exp[-\tau_c(\lambda) - \tau_l(\lambda)] \tag{4.34}$$

with τ_c and τ_l the total continuum and line optical depth, respectively. As long as the thickness of the slab is not too large, this most simple approximation allows a fast computation of the spectrum, which is desirable for spectral fitting.

In particular UV observations of AGN show that the absorption lines can often be decomposed into multiple velocity components. In the X-ray band these are not always fully resolvable, which led us to the following approach. Each absorption line is split into different velocity components, using

$$\tau_l(v) = \sum_i \tau_i \exp\left[-(v - v_i)^2/2\sigma_{\rm v}^2\right] \tag{4.35}$$

(or the equivalent generalisation to the Voigt profile). Further, we take

$$v_i = v_0 + i\,\Delta v, \tag{4.36}$$

$$\tau_i = K \exp\left[-v_i^2/2\sigma_{\rm b}^2\right], \tag{4.37}$$

where v_0 is the average velocity of the blend (a negative value corresponds to a blue-shift or outflow), Δv is the separation between the velocity components, and the r.m.s. width of the blend $\sigma_{\rm b}$ is in general larger than the intrinsic width $\sigma_{\rm v}$ of the components (do never confuse both σ's!). The normalization K is defined in such a way that $\sum \tau_i = \tau_0$. Finally, the total optical depth τ_0 is given by

$$\tau_0 = 0.106 f N_{20} \lambda/\sigma_{\rm v,100}. \tag{4.38}$$

Here f is the oscillator strength, λ the wavelength in Å, $\sigma_{\rm v,100}$ the velocity dispersion in units of 100 km/s and N_{20} the total column density of the ion in units of 10^{20} m^{-2}.
This dynamical structure offers the user a broad range of applicability. However, we advise the user to use the extension with $\sigma_{\rm b}$ with caution! Always start with the most simple case. The default values for SPEX are defined in such a way that $\sigma_{\rm b} = 0$. This will produce the "normal" case of single absorption lines. In that case, the velocity separation Δv is an irrelevant parameter.
Finally, we make a remark on the r.m.s. line width of individual lines, $\sigma_{\rm v}$. In our code, this *only* includes the turbulent broadening of the lines. The thermal broadening due to motion of the ions is included by adding it in quadrature to the tutbulent broadening. The only exception is the *slab* model, where of course due to the lack of underlying physics the thermal broadening is unknown, and therefore in using the slab model one should be aware that $\sigma_{\rm v}$ also includes a thermal contribution.

4.29.5 Atomic database for the absorbers

The continuum opacities are taken from Verner et al. ([1995]). Line opacities and wavelengths for most ions are from Verner et al. ([1996]), with the following additions and exceptions. For the L-shell ions of iron (Fe XVII − Fe XXIV) we used HULLAC calculations. The wavelengths of these L-shell transitions were adjusted according to Phillips et al. ([1999]). Also the L-shell ions of Si VIII–Si XII, S X–S XIV and Ar XV were calculated using the HULLAC code. In addition, we added the 2p–3d inner shell resonance lines of Fe I−Fe XVI from Behar et al. ([2002]). These inner shell resonance lines occur mainly in the 16−17 Å band and appear to be extremely important diagnostic tools, since the normal resonance lines of these lowly ionized iron ions have their wavelengths mainly in the inaccessible EUV band. Also we included the inner shell K-shell transitions of oxygen ions (O I – O VI) from the work of Behar. We adjusted the wavelength of the strongest O V line to 22.374 ± 0.003 Å, a value measured (M. Schmidt, P. Beiersdorfer, private communication) at the University of California EBIT-I electron beam ion trap, which has been optimized for laboratory X-ray astrophysics measurements (Beiersdorfer [2003]). L-shell transitions of Ni I–Ni XVIII were calculated by Raassen using Cowan's code, with the neglect of line broadening due to auto-ionizations. Lines from O VII up to $n = 100$ were calculated by extrapolation of the data for $n \leq 10$. Finally, we added two C IV lines from the work of Nahar et al. ([2001]). For some hydrogenic ions (C, N, O, Ne, Mg, Si, S and Ar) we added the transitions from principal quantum numbers 11–20 to the ground using our own SPEX dtatabase. K-shell transitions in Fe II–Fe XXIV were included from the series of papers by Palmeri et al. ([2003]), Mendoza et al. ([**?**]) and Bautista et al. ([2004]).

4.30 Non-thermal electron distributions

In the current version of SPEX it is possible to include the effects of non-thermal (NT) electron distributions. Such NT distributions affect the spectrum in two ways: by changing the ionization balance and the emitted spectrum. Currently we take both effects into account for all our plasma models. The only exception is the NEI model, for non-equilibrium ionisation. The way we calculate the ionisation balance does not allow us to include the NT effects in the ionisation balance, but we do take it into account in the spectrum.

The way we implemented it is as follows. Our atomic data (like collision strengths) are all parameterized in a way that allow for analytical integration over simple electron distributions like delta-functions, Maxwellians or power laws. But there are other processes like radiative recombination where such an analytical approach fails. However, one way or the other, we use expressions for Maxwellian-averaged rates in all our calculations. In principle, it is possible to decompose any electron distribution as a linear combination of Maxwellians. The total rate, for example a recombination rate, is then the sum of the rates caused by the individual Maxwellian components.

Therefore, in all our rate calcualtions we build the total rate based upon such a linear combination of Maxwellians. This is done for all relevant processes (ionization, recombination, excitation, etc.)

In all our thermal models, there is an ascii-parameter called "file"; if this value is defined (i.e. when a file name of an existing file is entered), it will read the parameters of the Maxwellians from an ascii-file with that filename. If there is not such a file, or if the filename is reset by entering the "par file aval none" command, no file will be taken into account (i.e., we have a simple, single Maxwellian again).

The (ascii) file should have the following format. On the first line, the number of Maxwellians is given. Then for each Maxwellian there is a separate line containing two numbers: the first number is the temperature of the Maxwellian, in units of the main temperature of the plasma; the second number is the total number of electrons in this Maxwellian, relative to the main component. It is wise to have the parameters of the main component as the first line of this list.

Let us give an example. Suppose we have a plasma with three components: 2, 20 and 200 keV Maxwellians, with electron densities of 3000, 300 and 30 m^{-3}, respectively. In this case the parameters of the *cie* model should be: temperature 2 keV, electron density 3000 m^{-3}, and the corresponding file should be as follows:

```
3
1 1
10 0.1
100 0.01
```

The first line tells us that there are three Maxwellians. The second line contains the parameters of the first Maxwellian, that we scale here to 1 1 (it is the one with temperature 2 keV and electron density 3000 m^{-3}). The third lines contain the second Maxwellian, which has a 10 times higher temperature but also a 10 times lower electron density as the first component. Finally, the fourth line contains the parameters of the third Maxwellian, which has a 100 times higher temperature and a 100 times lower electron density as the first component.

WARNING: *Note that you can give your ascii-file any name, except of course the name "none", which is reserved for resetting the file name.*

WARNING: *Note that both numbers for each Maxwellian must be positive, i.e. positive temperatures and electron densities.*

4.31 Non-equilibrium ionisation (NEI) calculations

Our current implementation of time-dependent, non-equilibrium ionisation spectra is based upon the work of Kaastra & Jansen ([1993]) (KJ hereafter). The method consists essentially upon calculating the transition matrix (containing ionisation and recombination rates) for a grid of temperatures, calculating the eigenvalues and eigenvectors of these matrices and also the inverse eigenvector matrix, and storing those on disk. The ionisation balance is calculated then by dividing the temperature history in several steps, with the steps defined in such a way that the temperature over each time interval is close to one temperature grid point. Then for each step the time evolution of the plasma is calculated using eigenvalue decomposition and using the pre-calculated coefficients for the relevant temperature. For more details see KJ.

Here we make a few changes with respect to KJ. First, KJ adopted a prescribed history of the Hydrogen density, and calculate the corresponding electron density at each time from the ion concentrations at that time. This was motivated by the argument that for a fixed plasma element the number of Hydrogen atoms remains constant over time, while the number of electrons depends upon the degree of ionisation and may be time-dependent, thus not known before the ionisation balance has beeen solved completely. But the situation is slightly more complicated. If the plasma is not (nearly) completely ionised, then for normal abundances there must be a considerable population of neutral Hydrogen or Helium. But in that case charge exchange reactions become important, and should be taken into account in the transition matrix. This however is impossible to incorporate in the KJ method, since in this case there are 3 free parameters ($n_{\rm e}/n_{\rm H}$, $n_{\rm e}/n_{\rm He}$ and T) instead of 1 (T), and storage of matrices etc. becomes physically impossible. Knowing that the method of KJ becomes inaccurate for low ionisation rates we decided to leave the prescribed $n_{\rm H}$ concept and return to the prescribed $n_{\rm e}$ concept. This has the advantage that the equations for the ionisation balance of each chemical element can be calculated independently from the other elements; KJ need to calculate the concentrations of all elements for each time step in order to determine the conversion from Hydrogen to electron density.

Another rationale behind the above-mentioned modification is the following. For self-similar supernova remnants, a class of problems where non-equilibrium ionisation applies, the hydrodynamics is inconsistent with a time- or place varying electron/Hydrogen ratio: that would introduce another scale length and destroy the assumptions behind self-similarity. Thus, for SNR models it is necessary to assume that the plasma is (almost) completely ionised in order to assure self-consistency. We will neglect small changes in the electron/Hydrogen ratio of SNRs therefore.

The second modification concerns the temperature grid. Contrary to KJ, we use a keV grid instead of a K grid. Further, since we now include 30 chemical elements instead of 15, we had to reduce the step size h of the grid from 0.05 in $\log T$ to 0.10, in order not to get disk storage problems. This change however is compensated by a better interpolation technique used here. KJ just rounded all temperatures to the nearest grid point. Here we interpolate the calculated concentrations between two temperature grid points linearly, which is one order of magnitude (h) better. A rough estimate learns that the error in the method of KJ per decade is nh^2 which gives for $n = 1/h$ an error of order 0.05, while the current implementation has typical errors per decade of nh^3 which gives a typical error of 0.01 per decade.

4.32 Supernova remnant (SNR) models

For the calculation of SNR models we follow Kaastra & Jansen ([1993]). But also here we make some modifications in the implementation.

The X-ray spectrum of a SNR is characterised in lowest order by its average temperature T, ionisation parameter U and emission measure Y. If one calculates a specific model however, these average values are not known a priori. A relevant set of input parameters could be the interstellar density, the shock radius and the explosion energy. These are the parameters used by KJ. A disadvantage is that this choice is not suitable for spectral fitting: changing one of the parameters gives correlated changes in T, U and Y, which is an undesirable feature in spectral fitting. KJ solved this by inputting a guess of T, U, and Y and using these guess parameters as the fitting parameters. After fitting, the model had to be calculated another time in order to find the relation between guessed and true parameters. here we take another choice. We will select input parameters which scale with T, U and Y respectively but are connected to the SNR parameters independent of the hydrodynamical model. The first parameter is the shock temperature

$$T_s, \tag{4.39}$$

the second the shock ionisation parameter defined by

$$U_s \equiv n_{\rm e} t \tag{4.40}$$

where $n_{\rm e}$ is the electron density before the shock (for a completely ionised gas), and t the age of the remnant. Further we define the shock reduced emission measure by

$$Y_s \equiv n_{\rm e} n_{\rm H} r_s^3 / d^2 \tag{4.41}$$

where r_s is the main shock radius, and d the distance to the source. We have introduced here also the electron density n_e instead of the more natural Hydrogen density, because this allows an efficient use of the model for spectral fitting: if one fits a spectrum with variable chemical abundances, then using the current scaling we need to calculate (for a given T_s, U_s and Y_s) the relative ion concentrations only once; for other abundances, the relative concentrations may be simply scaled without a need to redo completely the ionisation balance calculations. Finally we introduced the distance in the definition of Y_s in order to allow an easy comparison with spectra observed at earth. These 3 input parameters T_s, U_s and Y_s will serve as input parameters for all hydrodynamical models. They are linked to other parameters as follows:

$$f_{\rm p} k T_s = f_{\rm T} \epsilon_\rho m_{\rm p} r_s^2 / t^2 \tag{4.42}$$

where ϵ_ρ is a dimensionless factor representing the average mass in units of a proton mass per Hydrogen atom ($\rho = \epsilon_\rho m_{\rm p} n_{\rm H}$ with $m_{\rm p}$ the proton mass), and $f_{\rm p}$ is another dimensionless factor, representing the ratio of electron plus ion density over the Hydrogen density. Further, $f_{\rm T}$ is a model-dependent dimensionless scaling factor which is given by

$$f_{\rm T} = \frac{f_v^2}{\eta - 1} \tag{4.43}$$

with f_v the velocity scaling discussed below and η the density jump at the main shock front, which depends upon the Hydrodynamical model. Another scaling law links the gas velocity to the temperature:

$$v_2 = f_v r_s / t \tag{4.44}$$

with v_2 the post-shock gas velocity (not to be confused with the shock velocity!). The velocity scaling factor is given by

$$f_v \equiv \frac{(n-3)(\eta-1)}{(n-s)\eta} \tag{4.45}$$

with the understatement that for the Sedov and Solinger model one should take formally the limit $n = 5$.
The explosion energy E_0 is linked to the other parameters by

$$E_0 = \frac{\alpha r_s^5 \rho}{t^2}, \tag{4.46}$$

where α is the dimensionless energy integral introduced by Sedov. The value of α depends both upon the hydrodynamical model and the value for s and n, the density gradient scale heights of the interstellar medium and the stellar ejecta.
If we express T_s in keV, U_s in 10^{20} m^{-3}s, Y_s in 10^{20} m^{-5}, r_s in units of 10^8 m, $n_{\rm H}$ in 10^{20} m^{-3}, d in units of 10^{22} m and t in s, we obtain from (4.42) the value for the proton mass of $m_{\rm p} = 1.043969 \times 10^5$ keVs^2(10^{16}m^2)$^{-1}$. Using this value and defining $f_m = f_T \epsilon_\rho m_{\rm p}/f_{\rm p}$, the scaling laws for SNRs may be written as

$$n_{\rm e} = \frac{T_s^{1.5} U_s^3}{f_m^{1.5} Y_s d^2} \tag{4.47}$$

$$t = \frac{f_m^{1.5} Y_s d^2 n_{\rm e}}{T_s^{1.5} U_s^2 n_{\rm H}} \tag{4.48}$$

$$r = \frac{f_m Y_s d^2 n_{\rm e}}{U_s^2 T_s n_{\rm H}} \tag{4.49}$$

$$v = \frac{f_v T_s^{0.5}}{f_m^{0.5}} \tag{4.50}$$

$$E_0 = [1.6726231 \times 10^{33}\ \mathrm{J}] \frac{\alpha \epsilon_\rho f_m^{0.5} Y_s^2 d^4 n_{\rm e}}{U_s^3 T_s^{0.5} n_{\rm H}} \tag{4.51}$$

$$M = [8.409 \times 10^{-14}\ \mathrm{M}_\odot] \frac{\epsilon_\rho \beta \eta f_m^{1.5} Y_s^2 d^4 n_{\rm e}}{U_s^3 T_s^{1.5} n_{\rm H}} \tag{4.52}$$

where M is the mass integral, and β is the dimensionless mass integral (defined by $\beta = 4\pi \int \frac{\rho}{\rho_s} \frac{r}{r_s}^2 \mathrm{d}\frac{r}{r_s}$).

Chapter 5

Response matrices

5.1 Respons matrices

A theoretical framework is developed in order to estimate the optimum binning of X-ray spectra. Expressions are derived for the optimum bin size for the model spectra as well as for the observed data using different levels of sophistication. It is shown that by not only taking into account the number of photons in a given spectral model bin but also its average energy (not necessarily equal to the bin center) the number of model energy bins and the size of the response matrix can be reduced by a factor of 10–100. The response matrix should then not only contain the response at the bin center, but also its derivative with respect to the incoming photon energy. Practical guidelines are given how to construct the optimum energy grids as well as how to structure the response matrix. Finally a few examples are given to illustrate the present methods.

5.2 Introduction

Until recently, X-ray spectra of cosmic X-ray sources were obtained using instruments like proportional counters and GSPC's with moderate spectral resolution (typically 5–20 %). With the introduction of CCD detectors a major leap in resolution has been achieved (up to 2 %).
These X-ray spectra are usually analysed by a forward folding technique. First a spectral model appropriate for the observed source is chosen. This model spectrum is then convolved with the instrument response, represented usually by a response matrix. The convolved spectrum is compared to the observed spectrum, and the parameters of the model are varied in a fitting procedure in order to get the best solution.
This "classical" way of analysing X-ray spectra has been widely adopted and is implemented e.g. in spectral fittting packages such as XSPEC (Arnaud, 1996) and SPEX (Kaastra et al., 1996).
When high resolution data are available (like in optical spectra), the instrumental broadening is often small compared to the intrinsic line widths. Instead of forward folding and fitting, at each energy the observed spectrum is divided by the nominal effective area (straightforward deconvolution).
Although straightforward deconvolution, due to its simplicity, seems to be attractive for high resolution X-ray spectroscopy, it might fail in several situations. For example, the grating spectrometers of EUVE have a high spectral resolution, but it is not possible to separate the spectral orders at the longer wavelengths. Therefore only careful instrumental calibration *and* proper modelling of the short wavelength spectrum can help. A similar situation holds for the low energy gratings of AXAF. For the RGS detector on board of XMM, 30 % of all line flux is contained in broad scattering wings due to the mirror and grating.
However, the application of "standard" concepts, like a response matrix, is not at all trivial for these high resolution instruments. For example, with the RGS of XMM, the properly binned response matrix has a size of 120 Megabytes, counting only non-zero elements. This cannot be handled by most present day computer systems, if analysis packages like XSPEC or SPEX are applied to these data sets. Also, the larger spectral resolution enhances considerably the computation time needed to evaluate the spectral models. Since the models applied to AXAF and XMM data will be much more complex than those applied to data from previous missions, computational efficieny is important to take into acount.
For these reasons we critically re-evaluate the concept of response matrices and the way spectra are analysed. In fact, we conclude that it is necessary to drop the classical concept of a matrix, and to use instead a modified approach. First we evaluate the required binning for both the model and the data grid. We also show how to properly handle different features like redshift components, velocity broadening, absorption edges, etc. The proper treatment of these effects is critically intermixed with the overall accuracy of the spectral model.

5.3 Data binning

5.3.1 Introduction

An important problem in data analysis is how the data should be sampled. For example, most X-ray detectors are able to count the individual photons (events) that hit the detector. A number of properties of the event, like its arrival time, position or energy may be recorded. The electronics of most detectors operate in such a way that not the energy value (or a related pulse height) is recorded, but a digitized version of it. Then a histogram is produced containing the number of events as a function of the energy or pulse height channel. The bin size of these data channels ideally should not exceed the resolution of the instrument, otherwise important information may be lost. On the other hand, if the bin size is too small, one may have to deal with low statistics per data channel or with a large computational overhead caused by the large number of data channels. In this section we derive the optimum bin size for the data channels. We will start with the Shannon theorem and derive expressions for the errors made by using a particular bin size. From these errors and the statistical noise on the data it is then possible to arrive at the optimum bin size.

5.3.2 The Shannon theorem

There is an important theorem which helps to derive the optimum binsize for any signal. This is the Shannon (1949) sampling theorem, also sometimes attributed to Nyquist. This theorem states the following:
Let $f(t)$ be a continuous signal. Let $g(\omega)$ be its Fourier transform, given by

$$g(\omega) = \int\limits_{-\infty}^{\infty} e^{i\omega t} \mathrm{d}t. \tag{5.1}$$

If $g(\omega) = 0$ for all $|\omega| > W$ for a given frequency W, then $f(t)$ is band-limited, and in that case Shannon has shown that

$$f(t) = f_s(t) \equiv \sum_{n=-\infty}^{\infty} f(n\Delta) \frac{\sin \pi(t/\Delta - n)}{\pi(t/\Delta - n)}. \tag{5.2}$$

In (5.2), the bin size $\Delta = 1/2W$. Thus, a band-limited signal is completely determined by its values at an equally spaced grid with spacing Δ.
We can translate the above Shannon theorem easily in terms of a response function for an X-ray detector. In this case the variable t denotes the energy. In practice, many response functions are not band-limited (example: a Gaussian line profile). It is common practice in those cases to neglect the Fourier power for frequencies above W, and to reconstruct $f(t)$ from this cut-off Fourier transform. The arguing is that for sufficiently large W, the neglected power in the tail of the Fourier spectrum is small and hence (5.2) gives a fair approximation to the true signal.
The problem with such a treatment is that it is not obvious what the best choice for W and hence the bin size Δ is. There are theorems which express the maximum error in $f(t)$ in terms of the neglected power. Weiss (1963) has derived an expression for the maximum error introduced (see also Jerri, 1977):

$$E_a = |f(t) - f_s(t)| \leq \frac{2}{\pi} \int\limits_{W}^{\infty} |g(\omega)\mathrm{d}\omega. \tag{5.3}$$

One could now proceed by e.g. assuming that this maximum error holds over the entire range where a significant signal is detected, and then, depending upon the number of events used to sample $f(t)$, determine W. However, this yields a large over-estimation of the true error, and hence a much smaller bin size than really necessary.
The point to make here is that (5.3) is only useful for continuous signals that are sampled at a discrete set of time intervals, with no measurements in between. However X-ray spectra are sampled differently. First, the measured X-ray spectrum is determined by counting individual photons. The energy of each photon is digitized to a discrete data channel number. Therefore the measured X-ray spectrum is essentially a histogram of the number of events as a function of channel number. The essential difference with the previous discussion is that we do not measure the signal at the data channel boundaries, but we measure the sum (integral) of the signal between the data channel boundaries. Hence for X-ray spectra it is more appropriate to study the integral of $f(t)$ instead of $f(t)$ itself. This will be elaborated in the next paragraph.

5.3.3 Integration of Shannon's theorem

We have shown that for X-ray spectra it is more appropriate to study the integrated signal instead of the signal itself. Let us assume that the X-ray spectrum $f(t)$ represents a true probability distribution. In fact, it gives the probability that a photon will be detected at the data channel t. The cumulative probability density distribution function $F(t)$ is given by

$$F(t) = \int\limits_{-\infty}^{t} f(x)\mathrm{d}x. \tag{5.4}$$

If we insert (5.4) into the Shannon reconstruction (5.2) we obtain after interchanging the integration and summation, and keeping into mind that we cannot evaluate $F(t)$ at all arbitrary points but *only* at those grid points $m\Delta$ for integer m where also f_s is sampled:

$$F_s(m\Delta) = \frac{\Delta}{\pi} \sum_{n=-\infty}^{\infty} f(n\Delta)\Big\{\frac{\pi}{2} + \mathrm{Si}[\pi(m-n)]\Big\}. \tag{5.5}$$

The function $\mathrm{Si}(x)$ is the sine-integral as defined e.g. in Abramowitz & Stegun (1965). It is an antisymmetric function: $\mathrm{Si}(-x) = -\mathrm{Si}(x)$, and for large x it oscillates with an amplitude decreasing proportional to $1/x$ towards the constant $\pi/2$. From (5.5) it can be easily seen that in the limit of $\Delta \rightarrow 0$ with $t = m\Delta$ fixed this expression reduces to (5.4). For illustration, we list in table 5.1 some values of the weights used in the summation.
The expression (5.5) for F_s equals F if $f(t)$ is band-limited. We see in that case that at the grid points F is completely determined by the value of f at the grid points. By inverting this relation, one could express f at the grid points as a unique linear combination of the F-values at the grid. Since Shannon's theorem states that $f(t)$ for arbitrary t is determined completely by the f-values at the grid, we infer that $f(t)$ can be completely reconstructed from the discrete set of F-values. And then, by integrating this reconstructed $f(t)$, also $F(t)$ is determined.
We conclude that also $F(t)$ is completely determined by the set of discrete values $F(m\Delta)$ at $t = m\Delta$ for integer values of m, provided that $f(t)$ is band-limited.
In cases of non-bandlimited responses, we will use (5.5) to approximate the true cumulative distribution function at the energy grid. In doing this, a small error is made. These errors are known as aliasing errors. The errors can be calculated easily by comparing $F_s(m\Delta)$ by the true $F(m\Delta)$ values. Then, for an actually observed spectrum containing counting noise, these

Table 5.1: Weights $w_{mn} = 1/2 + \mathrm{Si}[(\pi(m-n)]/\pi$ used in the cumulative Shannon approximation.

$m-n$	w_{mn}	$m-n$	w_{mn}
-9	-0.0112	9	1.0112
-8	0.0126	8	0.9874
-7	-0.0144	7	1.0144
-6	0.0168	6	0.9832
-5	-0.0201	5	1.0201
-4	0.0250	4	0.9750
-3	-0.0331	3	1.0331
-2	0.0486	2	0.9514
-1	-0.0895	1	1.0895

sampling errors can be compared to the counting noise errors. We will conclude that the binning Δ is sufficient if the sampling errors are sufficiently small as compared to the noise errors. What is sufficient will be elaborated further.

For the comparison of the aliased model to the true model we consider two statistical tests, the classical χ^2 test and the Kolmogorov-Smirnov test. This is elaborated in the next sections.

5.3.4 The χ^2 test

Suppose that $f_0(x)$ is the unknown, true probability density function (pdf) describing the observed spectrum. Further let $f_1(x)$ be an approximation to f_0 using e.g. the Shannon reconstruction as discussed in the previous sections, with a bin size Δ. The probability p_{0n} of observing an event in the data bin number n under the hypothesis H_0 that the pdf is f_0 is given by

$$p_{0n} = \int_{(n-1)\Delta}^{n\Delta} f_0(x)\mathrm{d}x \tag{5.6}$$

and similar the probability p_{1n} of observing an event in the data bin number n under the hypothesis H_1 that the pdf is f_1 is

$$p_{1n} = \int_{(n-1)\Delta}^{n\Delta} f_1(x)\mathrm{d}x. \tag{5.7}$$

We define the relative difference δ_n between both probabilities as

$$p_{1n} \equiv p_{0n}(1+\delta_n). \tag{5.8}$$

Let the total number of events in the spectrum be given by N. Then the random variable X_n, defined as the number of observed events in bin n, has a Poisson distribution with an expected value $\mu_n = Np_{0n}$.

The classical χ^2 test for testing the hypothesis H_0 against H_1 is now based on the random variable:

$$X^2 \equiv \sum_{n=1}^{k} \frac{(X_n - Np_{0n})^2}{Np_{0n}}. \tag{5.9}$$

The hypothesis H_0 will be rejected if X^2 becomes too large. In (5.9) k is the total number of data bins. X^2 has a (central) χ^2 distribution with k degrees of freedom, if the hypothesis H_0 is true, and in the limit of large N.

However if H_1 holds, X^2 no longer has a central χ^2 distribution. It is straightforward to show that if f_1 approaches f_0, then under the hypothesis H_1 X^2 has a non-central χ^2 distribution with non-central parameter

$$\lambda_c = N \sum_{n=1}^{k} p_{0n}\delta_n^2. \tag{5.10}$$

This result was derived by Eisenhart (1938). It is evident that λ_c becomes small when f_1 comes close to f_0. For $\lambda_c = 0$ the non-central χ^2 distribution reduces to the "classical" central χ^2 distribution. The expected value and variance of the non-central χ^2 distribution are simply $k + \lambda_c$ and $2k + 4\lambda_c$, respectively.

We now need to find out a measure how much the probability distribution of X^2 under the assumptions H_0 and H_1 differ. The bin size Δ will be acceptable if the corresponding probability distributions for X^2 under H_0 (a central χ^2 distribution) and H_1 (a non-central χ^2 distribution) are close enough.

Using the χ^2 test, H_0 will in general be rejected if X^2 becomes too large, say larger than a given value c_α. The probability α that H_0 will be rejected if it is true is called the size of the test. Let us take as a typical value $\alpha = 0.05$. The probability that H_0 will be rejected if H_1 is true is called the power of the test and is denoted by $1 - \beta$. An ideal statistical test of H_0 against H_1 would also have β small (large power). However, in our case we are not interested in getting the most discriminative power of both alternatives, but we want to know how small λ_c can be before the alternative distributions of X^2 become indistinguishable. In other words,

$$\begin{aligned} \mathrm{Prob}(X^2 > c_\alpha | H_1) &= f\alpha & (5.11)\\ \mathrm{Prob}(X^2 > c_\alpha | H_0) &= \alpha, & (5.12) \end{aligned}$$

where ideally f should be 1 or close to 1. For $f = 1$, both distributions would be equal and $\lambda_c = 0$, resulting in $\Delta = 0$. We adopt here as a working value $f = 2$, $\alpha = 0.05$. That is, using a classical χ^2 test H_0 will be rejected in only 5 % of the cases, and H_1 in 10 % of the cases. In general we are interested in spectra with a large number of bins k. In the limit of large k, both the central and non-central χ^2 distribution can be approximated by a normal distribution with the appropriate mean and variance. Using this approximation, the criterion (5.11) translates into

$$k + \sqrt{2k}q_\alpha = k + \lambda_c + \sqrt{2k + 4\lambda_c}q_{f\alpha} \tag{5.13}$$

where now q_α is given by $G(q_\alpha) = 1 - \alpha$ with G the cumulative probability distribution of the standard normal distribution. For the standard values we have $q_{0.05} = 1.64485$ and $q_{0.10}$=1.28155. Solving (5.13) in the asymptotic limit of large k yields

$$\lambda_c = \sqrt{2k}(q_\alpha - q_{f\alpha}) \equiv p(\alpha, f)\sqrt{2k}. \tag{5.14}$$

For our particular choice of $\alpha = 0.05$ and $f = 2$ we have $p(\alpha, f) = 0.3633$. Other values are listed in table 5.2.
We see that the central displacement λ_c must be typically a small fraction of the natural "width" $\sqrt{2k}$ of the χ^2 distribution. The constant of proportionality $p(\alpha, f)$ does not depend strongly upon α as can be seen from table 5.2. The dependence on f is somewhat stronger, in the sense that if f approaches 1, $p(\alpha, f)$ approaches 0 as it should be.

Table 5.2: Scaling factors for the displacement parameter

α	f	$p(\alpha, f)$
0.01	2.0	0.27260
0.025	2.0	0.31511
0.05	2.0	0.36330
0.10	2.0	0.47513
0.05	1.2	0.09008
0.05	1.5	0.20532
0.05	2.0	0.36330

As a typical value we use the value of 0.3633 for $\alpha = 0.05$ and $f = 2$. Combining (5.10) and (5.14) yields

$$\epsilon = (2k)^{0.25} [p(\alpha, f)]^{0.5} N^{-0.5}, \tag{5.15}$$

where the probability-weighted relative error ϵ is defined by

$$\epsilon^2 \equiv \sum_{n=1}^{k} p_{0n} \delta_n^2. \tag{5.16}$$

5.3.5 Extension to complex spectra

The χ^2 test and the estimates that we derived hold for any spectrum. In (5.15) N should represent the total number of counts in the spectrum and k the total number of bins. Given a spectrum with pdf $f_0(x)$, and adopting a certain bin width Δ, the Shannon approximation $f_1(x)$ can be determined, and accordingly the value of ϵ^2 can be obtained from (5.16). By using (5.15) the value of N corresponding to Δ is found, and by inverting this relation, the bin width as a function of N is obtained.
This method is not always practical, however. First, for many X-ray detectors the spectral resolution is a function of the energy, hence binning with a uniform bin width over the entire spectrum is not always desired.
Another disadvantage is that regions of the spectrum with good statistics are treated the same way as regions with poor statistics. Regions with poor statistics contain less information than regions with good statistics, hence can be sampled on a much coarser grid.
Finally, a spectrum often extends over a much wider energy band than the spectral resolution of the detector. In order to estimate ϵ^2, this would require that first the model spectrum, convolved with the instrument response, should be used in order to determine $f_0(x)$. However, the true model spectrum is known in general only *after* spectral fitting. The spectral fitting can be done only after a bin size Δ has been set.

It is much easier to characterize the errors in the line-spread function of the instrument for a given bin size Δ, since that can be done independently of the incoming photon spectrum. If we restrict our attention now to a region r of the spectrum with a width of the order of a resolution element, then in this region the observed spectrum is mainly determined by the convolution of the lsf in this resolution element r with the true model spectrum in a region with a width of the order of the size of the resolution element. We ignore here any substantial tails in the lsf, since they are of little importance in the determination of the optimum bin size/resolution.
If all photons within the resolution element would be emitted at the same energy, the observed spectrum would be simply the lsf multiplied by the number of photons N_r. For this situation, we can easily evaluate ϵ_r^2 for a given binning. If the spectrum is not mono-energetic, then the observed spectrum is the convolution of the lsf with the photon distribution within the resolution element, and hence any sampling errors in the lsf will be more smooth (smaller) than in the mono-energetic case. It follows that if we determine ϵ_r^2 from the lsf, this is an upper limit to the true value for ϵ^2 within the resolution element r.
In (5.10) the contribution to λ_c for all bins is added. The contribution of a resolution element r to the total is then given by $N_r\epsilon_r^2$, where N_r must be the number of photons in the resolution element. We keep here as a working definition of a resolution element the size of a region with a width equal to the FWHM of the instrument. An upper limit to N_r can be obtained by counting the number of events within one FWHM and multiplying it by the ratio of the total area under the lsf (should be equal to 1) to the area under the lsf within one FWHM (should be smaller than 1).
We now demand that the contribution to λ_c from all resolution elements is the same. Accordingly, resolution elements with many counts need to have f_1 closer to f_0 than regions with less counts.
If there are R resolution elements in the spectrum, we demand therefore for each resolution element r the following condition:

$$\lambda_{c,r} \equiv N_r\epsilon_r^2 = \lambda_c/R \tag{5.17}$$

with

$$\epsilon_r^2 \equiv \sum_{n=1}^{k_r} p_{0n}\delta_n^2, \tag{5.18}$$

where now p_{0n} and δ_n are related to the lsf, *not* to the observed spectrum.
Further k_r is the number of bins within the resolution element. The value of λ_c in (5.17) is given again by (5.14) where now k represents the total number of bins in the spectrum, i.e.

$$k = \sum_{r=1}^{R} k_r. \tag{5.19}$$

Combining everything, we have for each resolution element r:

$$\epsilon_r = (2k)^{0.25}[p(\alpha, f)]^{0.5}(RN_r)^{-0.5} \tag{5.20}$$

5.3.6 Difficulties with the χ^2 test

The first step in applying the theory derived above is evaluating (5.18) for a given lsf. Let us take a Gaussian lsf, sampled with a step size of 1σ as an example. Table 5.3 summarizes the intermediate steps.

Table 5.3: Contribution to $\lambda_{c,r}$ for a gaussian lsf

n	p_{0n}	p_{1n}	$p_{0n} - p_{1n}$	$p_{0n}\delta_n^2$
-8	$6.11\ 10^{-16}$	$3.04\ 10^{-4}$	$-3.04\ 10^{-4}$	$1.51\ 10^{8}$
-7	$1.28\ 10^{-12}$	$-3.33\ 10^{-4}$	$3.33\ 10^{-4}$	$8.65\ 10^{4}$
-6	$9.85\ 10^{-10}$	$3.65\ 10^{-4}$	$-3.65\ 10^{-4}$	$1.35\ 10^{2}$
-5	$2.86\ 10^{-7}$	$-3.99\ 10^{-4}$	$3.99\ 10^{-4}$	$5.57\ 10^{-1}$
-4	$3.14\ 10^{-5}$	$4.60\ 10^{-4}$	$-4.29\ 10^{-4}$	$5.86\ 10^{-3}$
-3	$1.32\ 10^{-3}$	$8.77\ 10^{-4}$	$4.41\ 10^{-4}$	$1.48\ 10^{-4}$
-2	$2.14\ 10^{-2}$	$2.18\ 10^{-2}$	$-4.10\ 10^{-4}$	$7.85\ 10^{-6}$
-1	$1.36\ 10^{-1}$	$1.36\ 10^{-1}$	$3.04\ 10^{-4}$	$6.78\ 10^{-7}$
0	$3.41\ 10^{-1}$	$3.42\ 10^{-1}$	$-1.14\ 10^{-4}$	$3.80\ 10^{-8}$
1	$3.41\ 10^{-1}$	$3.42\ 10^{-1}$	$-1.14\ 10^{-4}$	$3.80\ 10^{-8}$
2	$1.36\ 10^{-1}$	$1.36\ 10^{-1}$	$3.04\ 10^{-4}$	$6.78\ 10^{-7}$
3	$2.14\ 10^{-2}$	$2.18\ 10^{-2}$	$-4.10\ 10^{-4}$	$7.85\ 10^{-6}$
4	$1.32\ 10^{-3}$	$8.77\ 10^{-4}$	$4.41\ 10^{-4}$	$1.48\ 10^{-4}$
5	$3.14\ 10^{-5}$	$4.60\ 10^{-4}$	$-4.29\ 10^{-4}$	$5.86\ 10^{-3}$
6	$2.86\ 10^{-7}$	$-3.99\ 10^{-4}$	$3.99\ 10^{-4}$	$5.57\ 10^{-1}$
7	$9.85\ 10^{-10}$	$3.65\ 10^{-4}$	$-3.65\ 10^{-4}$	$1.35\ 10^{2}$
8	$1.28\ 10^{-12}$	$-3.33\ 10^{-4}$	$3.33\ 10^{-4}$	$8.65\ 10^{4}$

It is seen immediately that for large values of $|n|$, the probability p_{0n} of a photon falling in bin n decreases rapidly. Also, the approximation is always better than 0.0005. However, the *relative* error of the approximation increases as n increases. Therefore, the largest contribution to χ^2 comes from the tails of the Gaussian distribution! In practice one could circumvent this by cutting the summation at some finite n. It is well known that for the χ^2 test to be valid, the expected number of events in each bin should not be too small. The summation could be stopped where $N_r p_{0n}$ becomes a few. However, since the largest contribution to ϵ_r^2 comes from the largest value of $|n|$, the value of ϵ_r^2 will be a very steep function of the precise cut-off criterion, which is an undesirable effect.
An alternative solution can be obtained by using the Kolmogorov-Smirnov test, which is elaborated in the following section.

5.3.7 The Kolmogorov-Smirnov test

A good alternative to the χ^2 test for the comparison of probability distributions is the Kolmogorov-Smirnov test. This powerful, non-parametric test is based upon the test statistic D given by

$$D = \max|S(x) - F(x)|, \tag{5.21}$$

where $S(x)$ is the *observed* cumulative distribution for the sample of size N. Clearly, if D is large $F(x)$ is a bad representation of the observed data set. The statistic $D' \equiv \sqrt{N}D$ for large N (typically 10 or larger) has the limiting Kolmogorov-Smirnov distribution. The hypothesis that the true cumulative distribution is F will be rejected if $D' > c_\alpha$, where the critical value c_α corresponds to a certain size α of the test. A few values for the confidence level are given in table 5.5. The expected value of the statistic D' is $\sqrt{\pi/2}\ln 2$=0.86873, and the standard

Table 5.4: Maximum difference d_n for a gaussian lsf

n	p_{0n}	d_n
-8	$6.106\ 10^{-16}$	$1.586\ 10^{-4}$
-7	$1.279\ 10^{-12}$	$-1.740\ 10^{-4}$
-6	$9.853\ 10^{-10}$	$1.909\ 10^{-4}$
-5	$2.857\ 10^{-7}$	$-2.079\ 10^{-4}$
-4	$3.138\ 10^{-5}$	$2.209\ 10^{-4}$
-3	$1.318\ 10^{-3}$	$-2.202\ 10^{-4}$
-2	$2.140\ 10^{-2}$	$1.896\ 10^{-4}$
-1	$1.359\ 10^{-1}$	$-1.139\ 10^{-4}$
0	$3.413\ 10^{-1}$	$2.675\ 10^{-9}$
1	$3.413\ 10^{-1}$	$1.139\ 10^{-4}$
2	$1.359\ 10^{-1}$	$-1.896\ 10^{-4}$
3	$2.140\ 10^{-2}$	$2.202\ 10^{-4}$
4	$1.318\ 10^{-3}$	$-2.209\ 10^{-4}$
5	$3.138\ 10^{-5}$	$2.079\ 10^{-4}$
6	$2.857\ 10^{-7}$	$-1.909\ 10^{-4}$
7	$9.853\ 10^{-10}$	$1.740\ 10^{-4}$
8	$1.279\ 10^{-12}$	$-1.586\ 10^{-4}$

deviation is $\pi\sqrt{1/12-(\ln 2)^2/\pi}$=0.26033.

Table 5.5: Critical valuse c_α for the Kolmogorov-Smirnov test as a function of the size of the test α

α	c_α	α	c_α
0.01	1.63	0.27	1.00
0.05	1.36	0.50	0.83
0.10	1.22	0.95	0.52
0.15	1.14	0.98	0.47
0.20	1.07	0.99	0.44

Similar to the case of the χ^2 test, we can argue that a good criterion for determining the optimum bin size Δ is that the maximum difference

$$\lambda_k \equiv \sqrt{N}\ \max_m |F_s(m\Delta) - F(m\Delta)| \tag{5.22}$$

should be sufficiently small as compared to the statistical fluctuations described by D'. Similar to (5.11) we can derive equations determining λ_k for a given size of the test α and f:

$$F_{\rm KS}(c_\alpha) = 1-\alpha, \tag{5.23}$$
$$F_{\rm KS}(c_\alpha - \lambda_k) = 1 - f\alpha. \tag{5.24}$$

Equation (5.23) gives the critical value for the Kolmogorov-statistic as a function of the size α, under the assumption H_0. Equation (5.24) gives the critical value for the Kolmogorov-statistic

as a function of the size α, under the assumption H_1. Taking $\alpha = 0.05$ and $f = 2$, we find $\lambda_k = 0.134$. Again, as in the case of the χ^2 test, this quantity depends only weakly upon α: for $\alpha = 0.01, f = 2$ we would have $\lambda = 0.110$.
Similar for the case of the χ^2 test, we now change our focus to the case of multiple resolution elements. Suppose that in each of the R resolution elements we perform a Kolmogorov-Smirnov test. How can we combine the tests in the different regions? For the χ^2 test we could simply add the contributions from the different regions. For the Kolmogorov-Smirnov test it is more natural to test for the maximum δ:

$$\delta \equiv \max_r \sqrt{N_r} D_r, \tag{5.25}$$

where as before N_r is the number of photons in the resolution element r, and D_r is given by (5.21) over the interval r. Note that the integral of $S_r(x)$ and $F_r(x)$ over the interval r should be 1. Our test statistic therefore combines "normalized" KS-tests over each resolution element. All the $\sqrt{N_r} D_r$ are independently distributed according to a KS distribution, hence the cumulative distribution function for their maximum is simply the Rth power of a single stochastic variable with the KS distribution. Therefore we obtain instead of (5.23, 5.24):

$$F_{\mathrm{KS}}(c_\alpha)^R = 1 - \alpha, \tag{5.26}$$
$$F_{\mathrm{KS}}(c_\alpha - \lambda_k)^R = 1 - f\alpha. \tag{5.27}$$

The hypothesis H_0 is rejected if $\delta > c_\alpha$. For our choice of $\alpha = 0.05$ and $f = 2$ we can calculate λ_k as a function of R by solving numerically the above equations using the exact Kolmogorov-Smirnov cumulative probability distribution function. In fig. 5.1 we show the solution. It is seen that it depends only weakly upon R; it has a maximum value 0.134 at $R = 1$ and is less than 3 times smaller at $R = 10^{10}$. We can approximate λ_k as a function of R by the following polynomial:

$$\lambda_k(R) = \sum_{i=0}^{5} a_i \left(\frac{{}^{10}\log(R)}{10}\right)^i, \tag{5.28}$$

where the numbers a_i are given in table 5.6. Note the factor of 10 in the denominator! The rms deviation of the approximation is less then 10^{-4}. The corresponding value for c_α is plotted in fig. 5.2, and is approximated with an rms deviation of 0.0004 by the polynomial:

$$c_\alpha(R) = \sum_{i=0}^{5} b_i \left(\frac{{}^{10}\log(R)}{10}\right)^i, \tag{5.29}$$

where the numbers b_i are given in table 5.6. This approximation is only valid for the range $0.5 < R < 10^{10}$.

5.3.8 Examples

Let us apply the theory derived before to some specific line spread functions. First we consider a Cauchy (or Lorentz) distribution, given by

$$f(x) = \frac{a}{\pi(a^2 + x^2)}, \tag{5.30}$$

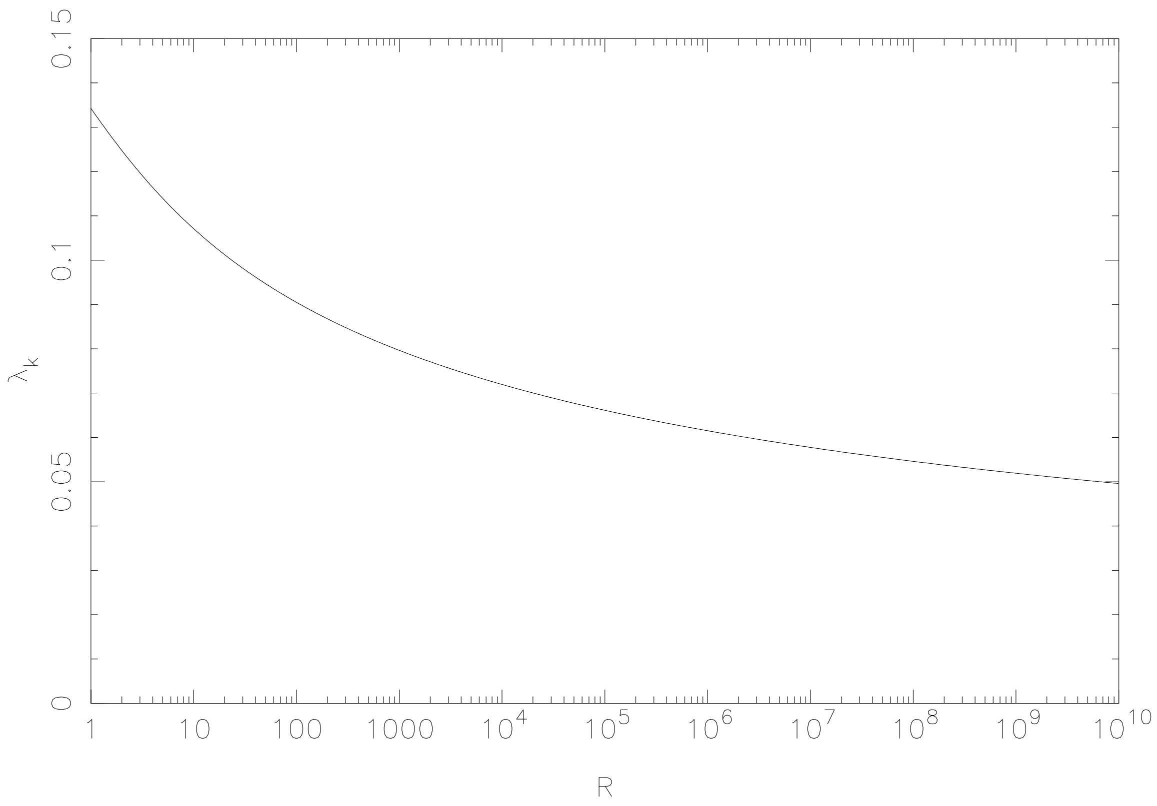


Figure 5.1: Displacement λ_k as a function of the number of resolution elements R. A polynomial approximation is given in the text.

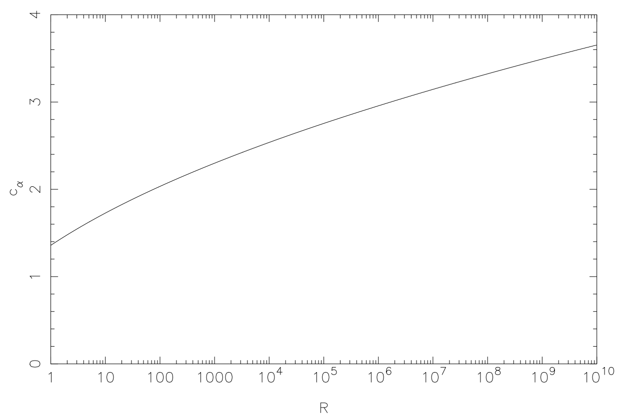


Figure 5.2: Critical value c_α as a function of the number of resolution elements R. A polynomial approximation is given in the text.

where a is the width parameter. The FWHM equals $2a$. Next we consider the Gaussian distribution, given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}, \tag{5.31}$$

Table 5.6: Coefficients for the approximations for λ_k and c_α

i	a_i	b_i
0	0.1342	1.3596
1	-0.3388	4.0609
2	0.7994	-4.3522
3	-1.1697	5.2225
4	0.9053	-3.7881
5	-0.2811	1.1491

with σ the standard deviation. The FWHM is given by $\sqrt{\ln 256}\sigma$, which is approximately 2.3548σ.
We have computed the value for $\delta \equiv \lambda_k/\sqrt{N}$ numerically for both distributions. We have taken into account also the effect of possible phase shifts, i.e. that the center of the distribution does not coincide with a grid point but with a certain phase of the grid. The maximum bin width Δ/FWHM as a function of the maximum absolute difference δ between the true and approximated cumulative density distribution is plotted for both distributions in fig. 5.3.

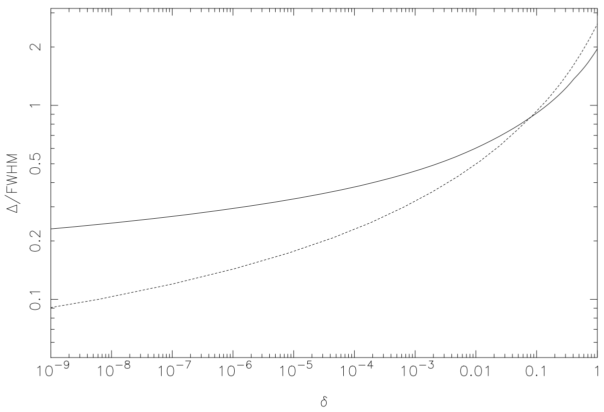


Figure 5.3: Required bin width Δ as a function of the accuracy parameter δ for a Gaussian distribution (solid line) and a Cauchy distribution (dashed line).

We can approximate both curves with sufficient accuracy by a polynomial in $x \equiv^{10} \log \delta$ as

$$^{10}\log(\Delta/\mathrm{FWHM}) = \sum_{i=0}^{4} c_i(\frac{^{10}\log(\delta)}{10})^i, \tag{5.32}$$

$$^{10}\log(\Delta/\mathrm{FWHM}) = \sum_{i=0}^{4} g_i(\frac{^{10}\log(\delta)}{10})^i, \tag{5.33}$$

for the Cauchy distribution (5.32) and Gaussian distribution (5.33) respectively, with the coefficents given in table 5.7.
The rms deviation of these fits over the range $10^{-9} < \delta < 1$ is 0.0086 for the Cauchy distribution and 0.0070 for the Gauss distribution. Outside of this range the approximation gets worse, however those values for δ are of no practical use.

Table 5.7: Coefficients for the approximations for the Cauchy and Gauss distribution.

i	c_i	g_i
0	0.400	0.277
1	5.065	3.863
2	9.321	8.470
3	9.333	9.496
4	3.584	3.998

The bin size as a function of the number of resolution elements R and number of photons per resolution element N_r is now obtained by combining (5.32) or (5.33) with (5.28) and using

$$\lambda_k = \sqrt{N_r}\delta. \tag{5.34}$$

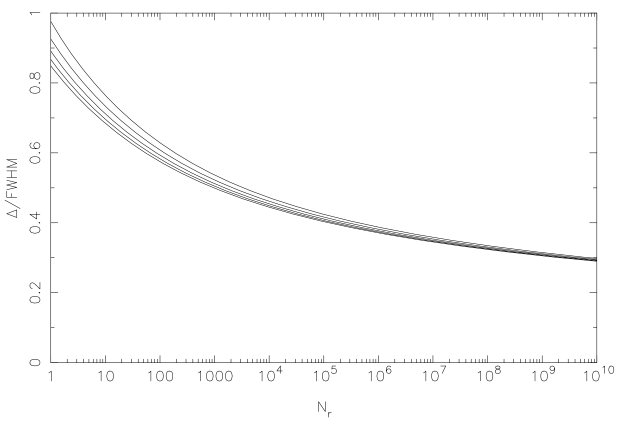


Figure 5.4: Required bin width Δ relative to the FWHM for a Gaussian distribution as a function of N_r for R=1, 10, 100, 1000 and 10000 from top to bottom, respectively.

In fig. 5.4 we show the required binning, expressed in FWHM units, for a Gaussian lsf as a function of R and N_r, and in fig. 5.5 for a Lorentzian.
For the Gaussian, the resolution depends only weakly upon the number of counts N_r. However in the case of a pure Lorentzian profile, the required bin width is somewhat smaller than for a Gaussian. This is due to the fact that the Fourier transform of the Lorentzian has relatively more power at high frequencies than a Gaussian ($\exp\left[-\omega\sigma\right]$ versus $\exp\left[-(\omega\sigma)^2/2\right]$ respectively). For low count rate parts of the spectrum, the binning rule 1/3 FWHM usually is too conservative!

5.3.9 Final remarks

We have estimated conservative upper bounds for the required data bin size. In the case of multiple resolution elements, we have determined the bounds for the worst case phase of the grid with respect to the data. In practice, it is not likely that all resolution elements would have the worst possible alignment. In fact, for the Gaussian lsf, the phase-averaged value for δ at a given bin size Δ is always smaller than 0.83, and is between 0.82–0.83 for all values of

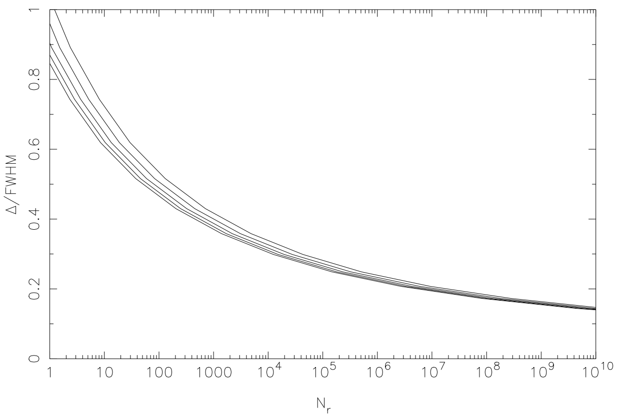


Figure 5.5: Required bin width Δ relative to the FWHM for a Lorentz distribution as a function of N_r for R=1, 10, 100, 1000 and 10000 from top to bottom, respectively.

Δ/FWHM < 1.3. However we recommend to use the conservative upper limits as given by (5.33).
Another issue is the determination of N_r, the number of events within the resolution element. We have argued that an upper limit to N_r can be obtained by counting the number of events within one FWHM and multiplying it by the ratio of the total area under the lsf (should be equal to 1) to the area under the lsf within one FWHM (should be smaller than 1). For the Gaussian lsf this ratio equals 1.314, for other lsfs it is better to determine it numerically, and not to use the Gaussian value 1.314.
Finally, in some cases the lsf may contain more than one component. For example, the grating spectra obtained with the EUVE SW, MW and LW detector have first, second, third etc. order contributions. Other instruments have escape peaks or fluorescent contributions. In general it is advisable to determine the bin size for each of these components individually, and simply take the smallest bin size as the final one.

5.4 Model binning

5.4.1 Model energy grid

The spectrum of any X-ray source is given by its photon spectrum $f(E)$. $f(E)$ is a function of the continuous variable E, the photon energy, and has e.g. units of photons $\mathrm{m}^{-2}\,\mathrm{s}^{-1}\,\mathrm{keV}^{-1}$. When we reconstruct how this source spectrum $f(E)$ is observed by an instrument, $f(E)$ is convolved with the instrument response $R(c, E)$ in order to produce the observed count spectrum $s(c)$ as follows:

$$s(c) = \int R(c, E) f(E) \mathrm{d}E \tag{5.35}$$

where the variable c denotes the "observed" (see below) photon energy and s has units of counts $s^{-1}\,\mathrm{keV}^{-1}$. The response function has the dimensions of an (effective) area, and can be given in e.g. m^2.
The variable c was denoted as the "observed" photon energy, but in practice it is some electronic signal in the detector used, the strength of which usually cannot take arbitrary values, but can have only a limited set of discrete values. A good example of this is the pulse-height channel for a CCD detector.
In almost all circumstances it is not possible to do the integration in eqn. (5.35) analytically, due to the complexity of both the model spectrum and the instrument response. For that reason, the model spectrum is evaluated at a limited set of energies, corresponding to the same pre-defined set of energies that is used for the instrument response $R(c, E)$. Then the integration in eqn. (5.35) is replaced by a summation. We call this limited set of energies the *model energy grid* or shortly the model grid. For each bin j of this model grid we define a lower and upper bin boundary E_{1j} and E_{2j} as well as a bin centroid E_j and a bin width ΔE_j, where of course E_j is the average of E_{1j} and E_{2j}, and $\Delta E_j = E_{2j}$ - E_{1j}.
Provided that the bin size of the model energy bins is sufficiently smaller than the spectral resolution of the instrument, the summation approximation to eqn. (5.35) is in general sufficiently accurate. The response function $R(c, E)$ has therefore been replaced by a response matrix R_{ij}, where the first index i denotes the data channel, and the second index j the model energy bin number.
We have explicitly:

$$S_i = \sum_j R_{ij} F_j, \tag{5.36}$$

where now S_i is the observed count rate in counts/s for data channel i and F_j is the model spectrum (photons $\mathrm{m}^{-2}\,\mathrm{s}^{-1}$) for model energy bin j.
This basic scheme is widely used and has been implemented e.g. in the XSPEC and SPEX (version 1.10 and below) packages.

5.4.2 Evaluation of the model spectrum

The model spectrum F_j can be evaluated in two ways. First, the model can be evaluated at the bin centroid E_j, taking essentially

$$F_j = f(E_j) \Delta E_j. \tag{5.37}$$

This is appropriate for smooth continuum models like blackbody radiation, power laws etc. For line-like emission it is more appropriate to integrate the line flux within the bin analytically, and taking

$$F_j = \int\limits_{E_{1j}}^{E_{2j}} f(E)\mathrm{d}E. \tag{5.38}$$

Now a sometimes serious flaw occurs in most spectral analysis codes. The observed count spectrum S_i is evaluated straightforward using eqn. (5.36). Hereby it is assumed that all photons in the model bin j have exactly the energy E_j. This has to be done since all information on the energy distribution within the bin is lost and F_j is essentially only the total number of photons in the bin. Why is this a problem? It should not be a problem if the model bin width ΔE_j is sufficiently small. However, this is (most often) not the case.

Let us take an example. The standard response matrix for the ASCA SIS detector uses a uniform model grid with a bin size of 10 eV. At a photon energy of 1 keV, the spectral resolution (FWHM) of the instrument is about 50 eV, hence the line centroid of an isolated narrow line feature containing N counts can be determined with a statistical uncertainty of $50/2.35\sqrt{N}$ eV (we assumed here for simplicity a Gaussian instrument response, for which the FWHM is approximately 2.35σ). Thus, for a moderately strong line of say 400 counts, the line centroid can be determined in principle with an accuracy of 1 eV, ten times better than the step size of the model energy grid. If the true line centroid is close to the boundary of the energy bin, there is a mis-match (shift) of 5 eV between the observed count spectrum and the predicted count spectrum, at about the 5σ significance level! If there are more of these lines in the spectrum, it is not excluded that never a satisfactory fit (in a χ^2 sense) is obtained, even in those cases where we know the true source spectrum and have a perfectly calibrated instrument! The problem becomes even more worrysome if e.g. detailed line-centroiding is done in order to derive velocity fields, like has been done e.g. for the Cas A supernova remnant.

A simple way to resolve these problems is just to increase the number of model bins. This robust method always works, but at the expense of a lot of computing time. For CCD-resolution spectra this is perhaps not a problem, but with the increased spectral resolution and sensitivity of the grating spectrometers of AXAF and XMM this becomes cumbersome.

For example, with a spectral resolution of 0.04 Å over the 5–38 Å band, the RGS detector of XMM will have a bandwidth of 825 resolution elements, which for a Gaussian response function translates into 1940 times σ. Several of the strongest sources to be observed by this instrument will have more than 10000 counts in the strongest spectral lines (example: Capella), and this makes it necessary to have a model energy grid bin size of about $\sigma/\sqrt{(}10000)$. The total number of model bins is then 194 000.

Most of the computing time in thermal plasma models stems from the evaluation of the continuum. The reason is that e.g. the recombination continuum has to be evaluated for all energies, for all relevant ions and for all relevant electronic subshells of each ion. On the other hand, the line power needs to be evaluated only once for each line, regardless of the number of energy bins. Therefore the computing time is approximately proportional to the number of bins. Going from ASCA-SIS resolution (1180 bins) to RGS resolution (194 000 bins) therefore implies more than a factor of 160 increase in computer time. And then it can be anticipated that due to the higher spectral resolution of the RGS more complex spectral models are needed in order to explain the observed spectra than are needed for ASCA, with more free parameters, which also leads to additional computing time. And finally, the response matrices of the RGS become extremely

large. The RGS spectral response is approximately Gaussian in its core, but contains extended scattering wings due to the gratings, up to a distance of 0.5 Å from the line center. Therefore, for each energy the response matrix is significantly non-zero for at least some 100 data channels, giving rise to a response matrix of 20 million non-zero elements. In this way the convolution of the model spectrum with the instrument response becomes also a heavy burden.
For the AXAF-LETG grating the situation is even worse, given the wider energy range of that instrument as compared to the XMM-RGS.
Fortunately there are more sophisticated ways to evaluate the spectrum and convolve it with the instrument response, as is shown in the next subsection.

5.4.3 Binning the model spectrum

We have seen in the previous subsection that the "classical" way to evaluate model spectra is to calculate the total number of photons in each model energy bin, and to act as if all flux is at the center of the bin. In practice, this implies that the "true" spectrum $f(E)$ within the bin is replaced by its zeroth order approximation $f_0(E)$, given by

$$f_0(E) = N\delta(E - E_j), \tag{5.39}$$

where N is the total number of photons in the bin, E_j the bin centroid as defined in the previous section and δ is the delta-function.
In this section we assume for simplicity that the instrument response is Gaussian, centered at the true photon energy and with a standard deviation σ. Many instruments have line spread functions with Gaussian cores. For instrument responses with extended wings (e.g. a Lorentz profile) the model binning is a less important problem, since in the wings all spectral details are washed out, and only the lsf core is important. For a Gaussian profile, the FWHM of the lsf is given by

$$\text{FWHM } = \sqrt{\ln(256)}\sigma \simeq 2.35\sigma. \tag{5.40}$$

How can we measure the error introduced by using the approximation (5.39)? Here we will compare the cumulative probability distribution function (cdf) of the true bin spectrum, convolved with the instrumental lsf, with the cdf of the approximation (5.39) convolved with the instrumental lsf. The comparison is done using a Kolmogorov-Smirnov test, which is amongst the most powerful tests in detecting the difference between probability distributions, and which we used successfully in the section on data binning. We denote the lsf-convolved cdf of the true spectrum by $\phi(c)$ and that of f_0 by $\phi_0(c)$. By definition, $\phi(-\infty) = 0$ and $\phi(\infty) = 1$, and similar for ϕ_0.
Similar to section 5.3, we denote the maximum absolute difference between ϕ and ϕ_0 by δ. Again, $\lambda_k \equiv \sqrt{N}\delta$ should be kept sufficently small as compared to the expected statistical fluctuations $\sqrt{N}D$, where D is given again by eqn. (5.21). Following section 5.3 further, we divide the entire energy range into R resolution elements, in each of which a KS test is performed. For each of the resolution elements, we determine the maximum value for λ_k from (5.26)–(5.28), and using (5.34) we find for a given number of photons N_r in the resolution element the maximum allowed value for δ in the same resolution element.
The approximation f_0 to the true distribution f as given by eqn. (5.39) fails most seriously in the case that the true spectrum within the bin is also a δ-function, but located at the bin boundary, at a distance $\Delta E/2$ from the assumed line center at the bin center.

Using the Gaussian lsf, it is easy to show that the maximum error δ_0 for f_0 to be used in the Kolmogorov-Smirnov test is given by

$$\delta_0 = \frac{1}{2\sqrt{2\pi}}(\frac{\Delta E}{\sigma}) \tag{5.41}$$

This approximation holds in the limit of $\Delta E \ll \sigma$. Inserting eqn. (5.40) we find that the bin size should be smaller than

$$\frac{\Delta E}{\mathrm{FWHM}} < 2.1289\ \lambda(R) N^{-0.5}. \tag{5.42}$$

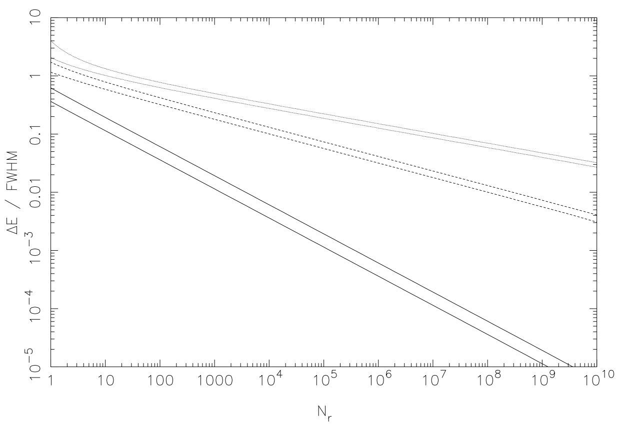


Figure 5.6: Maximum model energy grid bin width ΔE in terms of the FWHM as a function of the number of photons N_r in the bin, using different approximations f_0 (solid line), f_1 (dashed line) and f_2 (dotted line) as discussed in the text. The calculation is done for $R = 1$ (upper curves) or $R = 1000$ (lower curves) resolution elements.

Exact numerical results are shown in fig. 5.6. The approximation (5.41) is a lower limit to the exact results, so can be safely used. A somewhat better approximation (but still a lower limit) appears to be the following:

$$\frac{\Delta E}{\mathrm{FWHM}} = 2.129\delta_0 - 0.004\delta_0^2 + 0.555\delta_0^3. \tag{5.43}$$

As expected, one sees that for $N = 10000$ one needs to bin the model spectrum by about 0.01 times the FWHM.
A further refinement can be reached as follows. Instead of putting all photons at the bin center, we can put them at their average energy. This first-order approximation can be written as

$$f_1(E) = N\delta(E - E_a), \tag{5.44}$$

where E_a is given by

$$E_a = \int\limits_{E_{1j}}^{E_{2j}} f(E) E \mathrm{d}E \tag{5.45}$$

In the worst case for approximation f_0 (namely a narrow line at the bin boundary) the approximation f_1 yields exact results. In fact, it is easy to see that the worst case for f_1 is a situation with *two* δ-lines of equal strength, one at each bin boundary. In that case the width of the resulting count spectrum ϕ_1 will be broader than σ. Again in the limit of small ΔE, it is easy to show that the maximum error δ_1 for f_1 to be used in the Kolmogorov-Smirnov test is given by

$$\delta_1 = \frac{1}{8\sqrt{2\pi e}}(\frac{\Delta E}{\sigma})^2 \tag{5.46}$$

where e is the base of the natural logarithms. Accordingly, the limiting bin size for f_1 is given by

$$\frac{\Delta E}{\mathrm{FWHM}} < 2.4418\ \lambda(R)^{0.5} N^{-0.25}. \tag{5.47}$$

Exact results are plotted again in fig. 5.6. The better numerical approximation is

$$\frac{\Delta E}{\mathrm{FWHM}} = 2.44\delta_1^{0.5} - 0.05\delta_1 + 1.95\delta_1^{1.5}. \tag{5.48}$$

It is seen immediately that for large N the approximation f_1 requires a significantly smaller number of bins than f_0.
We can improve further by not only calculating the average energy of the photons in the bin (the first moment of the photon distribution), but also its variance (the second moment).
In this case we approximate

$$f_2(E) = N\exp[(E - E_a)/2\tau^2)], \tag{5.49}$$

where τ is given by

$$\tau = \int_{E_{1j}}^{E_{2j}} f(E)(E - E_a)^2 \mathrm{d}E \tag{5.50}$$

The resulting count spectrum is then simply Gaussian, with the average value centered at E_a and the width slightly larger than the instrumental width σ, namely $\sqrt{\sigma^2 + \tau^2}$.
The worst case for f_0 again occurs for two δ-lines at the opposite bin boundaries, but now with unequal strength. It can be shown again in the small bin width limit that

$$\delta_2 = \frac{1}{36\sqrt{6\pi}}(\frac{\Delta E}{\sigma})^3 \tag{5.51}$$

and that this maximum occurs for a line ratio of $6\sqrt{3} - 1$. The limiting bin size for f_2 is given by

$$\frac{\Delta E}{\mathrm{FWHM}} < 2.2875\ \lambda(R)^{1/3} N^{-1/6}. \tag{5.52}$$

Exact results are plotted again in fig. 5.6. The better numerical approximation is

$$\frac{\Delta E}{\mathrm{FWHM}} = 2.29\delta_2^{1/3} - 0.11\delta_2^{2/3} + 2.72\delta_2. \tag{5.53}$$

5.4.4 Which approximation to choose?

It is now time to compare the different approximations f_0, f_1 and f_2 as derived in the previous subsection. It can be seen from fig. 5.6 that the approximation f_1 implies an order of magnitude or more improvement over the classical approximation f_0. However, the approximation f_2 is only slightly better than f_1. Moreover, the computational burden of approximation f_2 is large. The evaluation of (5.50) is rather straightforward to do, although care should be taken with single machine precision: first the average energy E_a should be determined, and then this value should be used in the estimation of τ. A more serious problem is that the width of the lsf should be adjusted from σ to $\sqrt{\sigma^2+\tau^2}$. If the lsf is a pure Gaussian, this can be done analytically; however for a slightly non-Gaussian lsf the true lsf should be convolved in general numerically with a Gaussian of width τ in order to obtain the effective lsf for the particular bin, and the computational burden is quite heavy. On the other hand, for f_1 only a shift in the lsf is sufficient. Therefore we recommend here to use the linear approximation f_1. The optimum bin size is thus given by (5.28), (5.34) and (5.48).

5.4.5 The effective area

In the previous subsection we have seen how the optimum model energy grid can be determined, taking into account the possible presence of narrow spectral features, the number of resolution elements and the flux of the source. There remains one other factor to be taken into account, and that is the energy dependence of the effective area. In the previous section we have considered merely the effect of the spectral redistribution ("rmf" in XSPEC terms), here we consider the "arf".

If the effective area $A_j(E)$ would be a constant A_j for a given model bin j, and if the photon flux of the bin in photons $\mathrm{m}^{-2}\,\mathrm{s}^{-1}$ would be given by F_j, then the total count rate produced by this bin would be simply A_jF_j. This approach is actually used in the "classical" way of analysing spectra, and it is justified in the approximation that all photons of the model bin would have the energy of the bin centroid. This is not always the case, however, and in the previous subsection we have given arguments that it is better not only to take into account the flux F_j but also the average energy E_a of the photons within the model bin. This average energy E_a is in general not equal to the bin centroid E_j, and hence we need to evaluate the effective area not at E_j but at E_a.

In general the effective area is energy-dependent and it is not trivial that the constant approximation over the width of a model bin is justified. In several cases this will not be the case. We consider here the most natural first-order extension, namely the assumption that the effective area within a model bin is a linear function of the energy. For each model bin j, we write the effective area $A(E)$ in general as follows:

$$A_j(E) = A_{0j} + A_{1j}(E - E_j) + \epsilon_j(E). \tag{5.54}$$

The above approximation is good when the residuals $\epsilon_j(E)$ are sufficiently small. We use a Taylor expansion in order to estimate the residuals:

$$\epsilon_j(E) \approx \frac{1}{2} A''(E_j)(E - E_j)^2, \tag{5.55}$$

where as usual A'' denotes the second derivative of A with respect to the energy. It is immediately seen from the above expressions that the maximum error occurs when E is at one of the bin boundaries. As a consequence, we can estimate the maximum error $\epsilon_{\max}$ as:

$$\epsilon_{\max} = \frac{1}{8}(\Delta E)^2 A''(E_j). \tag{5.56}$$

By using the approximation (5.54) with ϵ neglected we therefore make at most a maximum error given by (5.56) in the effective area. This can be translated directly into an error in the predicted count rate by multiplying $\epsilon_{\max}$ by the photon flux F_j. Alternatively, dividing $\epsilon_{\max}$ by the effective area A_{0j} at the bin center, we get the maximum error in the predicted probability distribution for the resulting counts. This maximum can be taken also as the maximum deviation δ in the cumulative distribution function that we used in the previous sections in the framework of the Kolmogorov-Smirnov tests.
Hence we write

$$\delta = \frac{(\Delta E)^2 A''(E_j)}{8A_{0j}}. \tag{5.57}$$

Using the same theory as in the previous section, we find an expression for the optimum bin size, as far as the effective area is treated in the linear approximation (compare with eqn. (5.47)):

$$\frac{\Delta E}{\mathrm{FWHM}} < \Big(\frac{8A}{E_j^2 A''}\Big)^{0.5} \Big(\frac{E_j}{\mathrm{FWHM}}\Big) \lambda(R)^{0.5} N^{-0.25}. \tag{5.58}$$

The bin width constraint derived here depends upon the dimensionless curvature of the effective area $A/E_j^2 A''$. In most parts of the energy range this will be a number of order unity or less. Since the second prefactor E_j/FWHM is by definition the resolution of the instrument, we see by comparing (5.58) with (5.47) that in general (5.47) gives the most severe constraint upon the bin width, unless either the resolution gets small (what happens e.g. for the Rosat PSPC detector at low energies), or the effective area curvature gets large (what may happen e.g. near the exponential cut-offs caused by filters).
Large effective area curvature due to the presence of exponential cut-offs is usually not a very serious problem, since these cut-offs also cause the count rate to be very low and hence weaken the binning requirements. Of course, discrete edges in the effective area should always be avoided in the sence that edges should always coincide with bin boundaries.
In practice, it is a little complicated to estimate from e.g. a look-up table of the effective area its curvature, although this is not impossible. As a simplification for order of magnitude estimates we can use the case where $A(E) = A_0 e^{bE}$ locally, which after differentiation yields

$$\sqrt{\frac{8A}{E_j^2 A''}} = \sqrt{8}\,\frac{\mathrm{d}\ln E}{\mathrm{d}\ln A}. \tag{5.59}$$

Inserting this into (5.58), we obtain

$$\frac{\Delta E}{\mathrm{FWHM}} < \sqrt{8}\Big(\frac{\mathrm{d}\ln E}{\mathrm{d}\ln A}\Big)\Big(\frac{E_j}{\mathrm{FWHM}}\Big) \lambda(R)^{0.5} N^{-0.25}. \tag{5.60}$$

5.4.6 Final remarks

In the previous two subsections we have given the constraints for determining the optimal model energy grid. Combining both requirements (5.48) and (5.60) we obtain the following optimum bin size:

$$\frac{\Delta E}{\mathrm{FWHM}} = \frac{1}{\frac{1}{w_1} + \frac{1}{w_a}} \tag{5.61}$$

where w_1 and w_a are the values of $\Delta E/\mathrm{FWHM}$ as calculated using (5.48) and (5.60), respectively. This choice of model binning ensures that no significant errors are made either due to inaccuracies in the model or effective area for flux distributions within the model bins that have $E_a \neq E_j$.

5.5 The proposed response matrix

5.5.1 Dividing the response into components

When a response matrix is stored on disk, usually only the non-zero matrix elements are stored and used. This is done in order to save both disk space and computational time. The procedure as used in XSPEC and the older versions of SPEX is then that for each model energy bin j the relevant column of the response matrix is subdivided into groups. The groups are the continuous pieces of the column where the response is non-zero. In general these groups are stored in a specific order: starting from the lowest energy, all groups of a single energy are given before turning to the next higher photon energy.
This is not the optimal situation neither with respect to disk storage nor with respect to computational efficiency, as is illustrated by the following example. For the XMM/RGS, the response consists of a narrow gaussian-like core with in addition a broad scattering component due to the gratings. The FWHM of the scattering component is typically 10 times larger than that of the core of the response. As a result, if the response would be saved as a "classical" matrix, we would end up with one response group per energy, namely the combined core and wings response, since these contributions have overlap, namely in the region of the gaussian-like core. As a result, the response becomes large, being a significant fraction of the product of the total number of model energy bins times the total number of data channels. This is not necessary, since the scattering contribution with its ten times larger width needs to be specified only on a model energy grid with ten times fewer bins, as compared to the gaussian-like core. Thus, by separating out the core and the scattering contribution, the total size of the response matrix can be reduced by about a factor of 10! Of course, as a consequence each contribution needs to carry its own model energy grid with it.
Therefore I propose here to subdivide the response matrix into its physical components. Then for each response component, the optimum model energy grid can be determined according to the methods described in section 3, and this model energy grid for the component can be stored together with the response matrix part of that component. Furthermore, at any energy each component may have at most 1 response group. If there would be more response groups, the component should be sub-divided further.
In X-ray detectors other than the RGS detector the subdivision could be quite different. For example, with the ASCA SIS detector one could split up the response e.g. into 4 components: the main diagonal, the Si fluorescence line, the escape peak and a broad component due to split events.

5.5.2 More complex situations

Outline of the problem

In most cases an observer will analyze the data of a single source, with a single spectrum and response for a single instrument. However, more complicated situations may arise. Examples are:

1. An extended source, where the spectrum may be extracted from different regions of the detector, but where these spectra need to be analysed simultaneously due to the overlap in point-spread function from one region to the other. This situation is e.g. encountered in the analysis of cluster data with ASCA or BeppoSAX.

2. For the RGS of XMM, the actual data-space in the dispersion direction is actually two-dimensional: the position z where a photon lands on the detector and its energy or pulse height c as measured with the CCD detector. X-ray sources that are extended in the direction of the dispersion axis ϕ are characterised by spectra that are a function of both the energy E and off-axis angle ϕ. The sky photon distribution as a function of (ϕ, E) is then mapped onto the (z, c)-plane. By defining appropriate regions in both planes and evaluating the correct (overlapping) responses, one may analyse extended sources.

3. One may also fit simultaneously several time-dependent spectra using the same response, e.g. data obtained during a stellar flare.

It is relatively easy to model all these situations (provided that the instrument is understood sufficiently, of course), as we show below.

Sky sectors

First, the relevant part of the sky is subdivided into *sectors*, each sector corresponding to a particular region of the source, for example a circular annulus centered around the core of a cluster, or an arbitrarily shaped piece of a supernova remnant, etc.
A sector may also be a point-like region on the sky. For example if there is a bright point source superimposed upon the diffuse emission of the cluster, we can define two sectors: an extended sector for the cluster emission, and a point-like sector for the point source. Both sectors might even overlap, as this example shows!
Another example: the two nearby components of the close binary α Cen observed with the XMM instruments, with overlapping psfs of both components. In that case we would have two point-like sky sectors, each sector corresponding to one of the double star's components.
The model spectrum for each sky sector may and will be different in general. For example, in the case of an AGN superimposed upon a cluster of galaxies, one might model the spectrum of the point-like AGN sector using a power law, and the spectrum from the surrounding cluster emission using a thermal plasma model.

Detector regions

The observed count rate spectra are extracted in practice in different *regions* of the detector. It is necessary here to distinguish clearly the (sky) sectors and (detector) regions. A detector region for the XMM EPIC camera would be for example a rectangular box, spanning a certain number of pixels in the x- and y-directions. It may also be a circular or annular extraction region centered around a particular pixel of the detector, or whatever spatial filter is desired. For the XMM RGS it could be a specific "banana" part of the detector-coordinate CCD pulse-height plane (z, c).
Note that the detector regions need not to coincide with the sky sectors, neither should their number to be equal. A good example of this is again the example of an AGN superimposed upon a cluster of galaxies. The sky sector corresponding to the AGN is simply a point, while, for a finite instrumental psf, its extraction region at the detector is for example a circular region centered around the pixel corresponding to the sky position of the source.
Also, one could observe the same source with a number of different instruments and analyse the data simultaneously. In this case one would have only one sky sector but more detector regions, namely one for each participating instrument.

Consequences for the response

In all the cases mentioned above, where there is either more than one sky sector or more than one detector region involved, it is necessary to generate the response contribution for each combination of sky sector - detector region. In the spectral analysis, for each sky sector the model photon spectrum is calculated, and all these model spectra are convolved with the relevant response contributions in order to predict the count spectra for all detector regions. Each response contribution for a sky sector - detector region combination itself may consist again of different response components, as outlined in the previous subsection.
Combining all this, the total response matrix then will consist of a list of components, each component corresponding to a particular sky sector and detector region. For example, let us assume that the RGS has two response contributions, one corresponding to the core of the lsf, and the other to the scattering wings. Let us assume that this instrument observes a triple star where the instrument cannot resolve two of the three components. The total response for this configuration then consists of 12 components, namely 3 sky sectors (assuming each star has its own characteristic spectrum) times two detector regions (a spectrum extracted around the two unresolved stars and one around the other star) times two instrumental contributions (the lsf core and scattering wings).

5.5.3 A response component

We have shown in the previous section how for the combination of a particular source configuration and instrumental properties response matrix components can be defined. In this subsection we investigate how to build such a matrix component.
Let us first return to our discussion on the optimum model energy grid binning. For high-resolution instruments the factor E_j/FWHM in (5.60) is by definition large, and hence for most energies the requirements for the binning are driven by the model accuracy, not by the effective area accuracy. One might ask whether the approximation of a constant effective area within a model bin instead of the linear approximation (5.54) would not be sufficient in that case. The reason is the following.
In order to acquire the high accuracy, we need to convolve the model spectrum for the bin, approximated as a δ-function centered around E_a, with the instrument response. In most cases we cannot do this convolution analytically, so we have to make approximations. From our expressions for the observed count spectrum $s(c)$, eqns. 5.35) and (5.36), it can be easily derived that the number of counts or count rate for data channel i is given by

$$S_i = \int\limits_{c_{i1}}^{c_{i2}} \mathrm{d}c \int \mathrm{d}E R(c, E) f(E) \tag{5.62}$$

where as before c_{i1} and c_{i2} are the formal channel limits for data channel i and S_i is the observed count rate in counts/s for data channel i. Interchanging the order of the integrations and defining the mono-energetic response for data channel i by $\tilde{R}_i(E)$ as follows:

$$\tilde{R}_i(E) \equiv \int\limits_{c_{i1}}^{c_{i2}} R(c, E)\mathrm{d}c, \tag{5.63}$$

we have

$$S_i = \int \mathrm{d}E f(E) \tilde{R}_i(E). \tag{5.64}$$

From the above equation we see that as long as we are interested in the observed count rate S_i of a given data channel i, we get that number by integrating the model spectrum multiplied by the "effective area" $\tilde{R}_i(E)$ for that particular data channel. We have approximated $f(E)$ for each model bin j by (5.44), so that (5.64) becomes:

$$S_i = \sum_j F_j \tilde{R}_i(E_{a,j}), \tag{5.65}$$

where as before $E_{a,j}$ is the average energy of the photons in bin j, given by (5.45) and F_j is the total photon flux for bin j, in e.g. photons $\mathrm{m}^{-2}\,\mathrm{s}^{-1}$. It is seen from (5.65) that we need to evaluate $\tilde{R}_i$ not at the bin center E_j but at $E_{a,j}$, as expected.
Formally we may split-up $\tilde{R}_i(E)$ in an effective area part $A(E)$ and a redistribution part $\tilde{r}_i(E)$ in such a way that

$$\tilde{R}_i(E) = A(E) \tilde{r}_i(E). \tag{5.66}$$

We have chosen our binning already in such a way that we have sufficient accuracy when the total effective area $A(E)$ within each model energy grid bin j is approximated by a linear function of the photon energy E. Hence the "arf"-part of $\tilde{R}_i$ is of no concern. We only need to check how the redistribution ("rmf") part $\tilde{r}_i$ can be calculated with sufficiently accuracy.
For $\tilde{r}_i$ the arguments go exactly the same as for $A(E)$, in the sense that if we approximate it locally for each bin j by a linear function of energy, the maximum error that we make is proportional to the second derivative with respect to E of $\tilde{r}_i$, cf. (5.56).
In fact, for a Gaussian redistribution function the following is straightforward to proove, and is left here as an excercise for the reader:

Theorem 1 *Assume that for a given model energy bin j all photons are located at the upper bin boundary $E_j + \Delta E/2$. Suppose that for all data channels we approximate $\tilde{r}_i$ by a linear function of E, the coefficients being the first two terms in the Taylor expansion around the bin center E_j. Then the maximum error δ made in the cumulative count distribution (as a function of the data channel) is given by (5.47) in the limit of small ΔE.*

The importance of the above theorem is that it shows that the binning for the model energy grid that we have chosen in section 2 is also sufficiently accurate so that $\tilde{r}_i(E)$ can be approximated by a linear function of energy within a model energy bin j, for each data channel i. Since we showed already that our binning is also sufficient for a similar linear approximation to $A(E)$, it follows that also the total response $\tilde{R}_i(E)$ can be approximated by a linear function. Hence we use within the bin j:

$$\tilde{R}_i(E_{a,j}) = \tilde{R}_i(E_j) + \frac{\mathrm{d}\tilde{R}_i}{\mathrm{d}E_j}(E_j)\ (E_{a,j} - E_j). \tag{5.67}$$

Inserting the above in (5.65) and comparing with (5.36) for the classical response matrix, we obtain finally:

$$S_i = \sum_j R_{ij,0} F_j + R_{ij,1}(E_a - E_j) F_j, \tag{5.68}$$

where $R_{ij,0}$ is the "classical" response matrix, evaluated for photons at the bin center (Note: *not* the average over the bin!), and $R_{ij,1}$ is its derivative with respect to the photon energy (Note: *not* to be confused with the data channel!), evaluated also at the bin center. In addition to the "classical" convolution, we thus get a second term containing the relative offset of the photons with respect to the bin center. This is exactly what we intended to have when we argued that the number of bins could be reduced considerably by just taking that offset into account! It is just at the expense of an additional derivative matrix, which means only a factor of two more disk space and computation time. But for this extra expenditure we gained much more disk space and computational efficiency because the number of model bins is reduced by a factor between 10–100!

Finally a practical note. The derivative $R_{ij,1}$ can be calculated in practice either analytically or by numerical differentiation. In the last case it is more accurate to evaluate the derivative by taking the difference at $E + \Delta E/2$ and $E - \Delta E/2$, and wherever possible not to evaluate it at one of these boundaries and the bin center. This last situation is perhaps only un-avoidable at the first and last energy value.

Also, negative response values should be avoided. Thus it should be ensured that $R_{ij,0} + R_{ij,1}h$ is everywhere non-negative for $-\Delta E/2 \le h \le \Delta E/2$. This can be translated into the constraint that $R_{ij,1}$ should be limited always to the following interval:

$$-2R_{ij,0}/\Delta E \le R_{ij,1} \le 2R_{ij,0}/\Delta E. \tag{5.69}$$

Whenever the calculated value of $R_{ij,1}$ should exceed the above limits, the limiting value should be inserted instead. This situation may happen for example for a Gaussian redistribution for responses a few σ away from the center, where the response falls of exponentially. However, the response $R_{ij,0}$ is small anyway for those energies so this limitation is not serious. The only reason is that we want to avoid negative predicted count rates, whatever small they may be.

5.6 Determining the grids in practice

Based upon the theory developed in the previous sections, I present here a practical set of algorithms for the determination of both the optimal data binning and the model energy grid determination. This may be helpful for the practical purpose of developing software for a particular instrument in order to construct the relevant response matrix.

5.6.1 Creating the data bins

Given an observed spectrum obtained by some instrument, the following steps should be performed in order to generate an optimally binned spectrum.

1. Determine for each original data channel i the nominal energy E_{j0}, defined as the energy for which the response at channel i reaches its maximum value. In most cases this will be the nominal channel energy.

2. Determine for each data channel i the limiting points $(i1, i2)$ for the FWHM, in such a way that $R_{k,j0} \geq 0.5\, R_{i,j0}$ for all $i1 \leq k \leq i2$ while the range of $(i1, i2)$ is as broad as possible.

3. By (linear) interpolation, determine for each data channel the points (fractional channel numbers) $c1$ and $c2$ near $i1$ and $i2$ where the response is actually half its maximum value. By virtue of the previous step, the absolute difference $|c1 - i1|$ and $|c2 - i2|$ never can exceed 1.

4. Determine for each data channel i the FWHM in number of channels c_i, by calculating $c2 - c1$. Assure that c_i is at least 1.

5. Determine for each original data channel i the FWHM in energy units (e.g. in keV). Call this W_i. This and the previous steps may of course also be performed directly using instrument calibration data.

6. Determine the number of resolution elements R by the following approximation:

$$R = \sum_i \frac{1}{c_i}. \tag{5.70}$$

7. Determine the parameter $\lambda_k(R)$ using (5.28).

8. Determine for each bin the effective number of events N_r from the following expressions:

$$C_r = \sum_{k=i1-1}^{i2+1} C_k, \tag{5.71}$$

$$h_r = \sum_{k=1}^{N_c} R_{k,j0} / \sum_{k=i1-1}^{i2+1} R_{k,j0}, \tag{5.72}$$

$$N_r = C_r h_r. \tag{5.73}$$

   In the above, C_k is the number of observed counts in channel k, and N_c is the total number of channels. Take care that in the summations $i1 - 1$ and $i2 + 1$ are not out of their valid range $(1, N_c)$. If for some reason there is not a first-order approximation available for the response matrix $R_{k,j}$ then one might simply approximate h_r from e.g. the Gaussian approximation, namely $h_r = 1.314$, cf. section 2. This is justified since the optimum bin size is not a strong function of N_r, cf. fig. 5.4. Even a factor of two error in N_r in most cases does not affect the optimal binning too much.

9. Calculate $\delta = \lambda_k(R)/\sqrt{N_r}$ for each data channel.

10. Using (5.33), determine for each data channel the optimum data bin size in terms of the FWHM. The true bin size b_i in terms of number of data channels is obtained by multiplying this by c_i calculated above during step 4. Make b_i an integer number by ignoring all decimals (rounding it to below), but take care that b_i is at least 1.

11. It is now time to merge the data channels into bins. In a loop over all data channels, start with the first data channel. Name the current channel i. Take in principle all channels k from channel i to $i + b_i - 1$ together. However, check that the bin size does not decrease significantly over the rebinning range. In order to do that check, determine for all k between i and $i + b_i - 1$ the minimum a_i of $k + b_k$. Extend the summation only from channel i to $a_i - 1$. In the next step of the merging, a_i becomes the new starting value i. The process is finished when a_i exceeds N_c.

5.6.2 creating the model bins

After having created the data bins, it is possible to generate the model energy bins. Some of the information obtained from the previous steps that created the data bins is needed.
The following steps need to be taken:

1. Sort the FWHM of the data bins in energy units (W_i) as a function of the corresponding energies E_{j0}. Use this array to interpolate later any true FWHM. Also use the corresponding values of N_r derived during that same stage. Alternatively, one may use directly the FWHM as obtained from calibration files.

2. Choose an appropriate start and end energy, e.g. the nomimal lower and upper energy of the first and last data bin, with an offset of a few FWHMs (for a Gaussian, about 3 FWHM is sufficient). In the case of a lsf with broad wings (like the scattering due to the RGS gratings) it may be necessary to take an even broader energy range.

3. In a loop over all energies as determined in the previous steps, calculate $\delta = \lambda_k(R)/\sqrt{N_r}$. The value of $\lambda_k(R)$ is the same as used in the determination of the data channel grid.

4. Determine also the effective area factor $\frac{\mathrm{d}\ln E}{\mathrm{d}\ln A}$ for each energy; one may do that using a linear approximation.

5. For the same energies, determine the necessary bin width in units of the FWHM using eqn. (5.61). Combining this with the FWHMs determined above gives for these energies the optimum model bin size ΔE in keV.

6. Now the final energy grid can be created. Start at the lowest energy $E_{1,1}$, and interpolate in the ΔE table the appropriate $\Delta E(E_{1,1})$ value for the current energy. The upper bin boundary $E_{2,1}$ of the first bin is then simply $E_{1,1} + \Delta E(E_{1,1})$.

7. Using the recursive scheme $E_{1,j} = E_{2,j-1}$, $E_{2,j} = E_{1,j} + \Delta E(E_{1,j})$ determine all bin boundaries untill the maximum energy has been reached. The bin centroids are simply defined as $E_j = 0.5(E_{1,j} + E_{2,j})$.

8. Finally, if there are any sharp edges in the effective area of the instrument, it is necessary to add these edges to the list of bin boundaries. All edges should coincide with bin boundaries.

5.7 Proposed file formats

In the previous sections it was shown how the optimum data and model binning can be determined, and what the corresponding optimum way to create the instrumental response is. Now I focus upon the possible data formats for these files.
For large data sets, the fastest and most transparant way to save a response matrix to a file would be to use direct fortran or C write statements. Unfortunately not all computer systems use the same binary data representation. Therefore the FITS-format has become the *de facto* standard in many fields of astronomical data analysis. For that reason I propose here to save the response matrix in FITS-format.
A widely used response matrix format is NASA's OGIP format. This is used e.g. as the data format for XSPEC. There are a few reasons that we propose *not* to adhere to the OGIP format here, as listed below:

1. The OGIP response file format as it is currently defined does not account for the possibility of response derivatives. As was shown in the previous sections, these derivatives are needed for the optimum binning. Thus, the OGIP format would need to be updated.

2. In the later versions of OGIP new possibilities of defining the energy grid have been introduced that are prone to errors. In particular the possibility of letting grids start at another index than 1 may (and has led) often to errors. The sofware also becomes unncessarily complicated, having to account for different possible starting indices. Moreover, the splitting into arf and rmf files makes it necessary to check if the indexing in both files is consistent, and also if the indexing in the corresponding spectra is consistent.

3. There are some redundant quantities in the OGIP format, like the "areascal" keyword. When the effective area should be scaled by a given factor, this can be done explicitly in the matrix.

4. As was shown in this work, it is more efficient to do the grouping within the response matrix differently, splitting the matrix into components where each component may have its own energy grid. This is not possible within the present OGIP format.

5.7.1 Proposed response format

I propose to give all response files the extension ".res", in order not to confuse with the ".rmf" or ".arf" files used within the OGIP format.
The file consists of the mandatory primary array that will remain empty, plus a number of extensions. In principle, we could define an extension for each response component. However, this may result into practical problems. For example, the fitsio-package allows a maximum of 1000 extensions. The documentation of fitsio says that this number can be increased, but that the access time to later extensions in the file may become very long.
In principle we want to allow for data sets with an unlimited number of response components. For example, when a cluster spectrum is analysed in 4 quadrants and 10 radial annuli, one might want to extract the spectrum in 40 detector regions and model the spectrum in 40 sky sectors, resulting in principle in at least 1600 response components (this may be more if the response for each sky sector and detector region has more components).
Therefore I propose to use only three additional and mandatory extensions.

The first extension is a binary table with 4 columns and contains for each component the number of data channels, model energy bins, sky sector number and detector region number (see table 5.8).
The second extension is a binary table with 5 columns and contains for each model energy bin for each component the lower model energy bin boundary (keV), the upper model energy bin boundary (keV), the starting data channel, end data channel and total number of data channels for the response group (see table 5.9).
The third extension is a binary table with 2 columns and contains for each data channel, for each model energy bin for each component the value of the response at the bin center and its derivative with respect to energy (see table 5.10). SI units are mandatory (i.e. m^2 for the response, $m^2 keV^{-1}$ for the response derivative).

Table 5.8: First extension to the response file

keyword	description
EXTNAME (=RESP_INDEX)	Contains the basic indices for the components in the form of a binary table
NAXIS1 = 16	There are 16 bytes in one row
NAXIS2 =	This number corresponds to the total number of components (the number of rows in the table)
NSECTOR =	This 4-byte integer is the number of sky sectors used.
NREGION =	This 4-byte integer is the number of detector regions used.
NCOMP =	This 4-byte integer is the totalnumber of response components used (should be equal to NAXIS2).
TFIELDS = 4	The table has 4 columns; all columns are written as 4-byte integers (TFORM='1J')
TTYPE1 = 'NCHAN'	First column contains the number of data channels for each component. Not necessarily the same for all components, but it must agree with the number of data channels as present in the corresponding spectrum file.
TTYPE2 = 'NEG'	Second column contains the number of model energy grid bins for each component. Not necessarily the same for all components.
TTYPE3 = 'SECTOR'	Third column contains the sky sector number as defined by the user for this component. In case of simple spectra, this number should be 1.
TTYPE4 = 'REGION'	Fourth column contains the detector region number as defined by the user for this component. In case of simple spectra, this number should be 1.

Any other information that is not needed for the spectral analysis may be put into additional file extensions, but will be ignored by the spectral analysis program.

Table 5.9: Second extension to the response file

keyword	description
EXTNAME (=RESP_COMP)	Binary table with for each row relevant index information for a single energy of a component; stored sequentially, starting at the lowest component and within each component at the lowest energy.
NAXIS1 = 20	There are 20 bytes in one row
NAXIS2 =	This number must be the sum of the number of model energy bins added for all components (the number of rows in the table).
TFIELDS = 5	The table has 5 columns
TTYPE1 = 'EG1'	The lower energy (keV) as a 4-byte real of the relevant model energy bin
TTYPE2 = 'EG2'	The upper energy (keV) as a 4-byte real of the relevant model energy bin
TTYPE3 = 'IC1'	The lowest data channel number (as a 4-byte integer) for which the response at this model energy bin will be given. The response for all data channels below IC1 is zero. Note that IC1 should be at least 1 (i.e. start counting at channel 1!).
TTYPE4 = 'IC2'	The highest data channel number (as a 4-byte integer) for which the response at this model energy bin will be given. The response for all data channels above IC2 is zero.
TTYPE5 = 'NC'	The total number of non-zero response elements for this model energy bin (as a 4-byte integer). NC is redundant, and should equal IC2-IC1+1, but it is convenient to have directly available in order to allocate memory for the response group.

Finally I note that the proposed response format is used as the standard format by version 2.0 of the SPEX spectral analysis package.

5.7.2 Proposed spectral file format

There exists also a standard OGIP FITS-type format for spectra. As for the OGIP response file format, this format has a few redundant parameters and not absolutely necessary options.
There is some information that is absolutely necessary to have. In the first place the *net source count rate* S_i (counts/s) and *its statistical uncertainty* ΔS_i (counts/s) are needed. These quantities are used e.g. in a classical χ^2-minimization procedure during spectral fitting.
In some cases the count rate may be rather low; for example, in a thermal plasma model at high energies the source flux is low, resulting in only a few counts per data channel. In such

Table 5.10: Third extension to the response file

keyword	description
EXTNAME (=RESP_RESP)	Contains the response matrix elements and their derivatives with respect to model energy. The data are stored sequentially, starting at the lowest component, within each component at the lowest energy bin, and within each energy bin at the lowest channel number.
NAXIS1 = 8	There are 8 bytes in one row
NAXIS2 =	This number must be the sum of the NC values, added for all model energy bins and all components (the number of rows in the table).
TFIELDS = 2	The table has 2 columns
TTYPE1 = 'Response'	The response values $R_{ij,0}$, stored as a 4-byte real **in SI units** i.e. in units of m^2.
TTYPE1 = 'Response_Der'	The response derivative values $R_{ij,1}$, stored as a 4-byte real **in SI units** i.e. in units of $\mathrm{m}^2\mathrm{keV}^{-1}$.

cases it is often desirable to use different statistics for the spectral fitting, for example maximum likelihood fitting. Such methods are often based upon the number of counts; in particular the difference between Poissonian and Gaussian statistics might be taken into account. In order to allow for these situations, also the *exposure time* t_i per data channel is needed. This exposure time needs not to be the same for all data channels; for example, a part of the detector might be switched off during a part of the observation, or the observer might want to use only a part of the data for some reason.
Further, in several situations the spectrum will contain a contribution from background events. The observed spectrum can be corrected for this by subtracting the *background spectrum* B_i scaled at the source position. In cases where one needs to use Poissonian statistics, including the background level, this subtracted background B_i must be available to the spectral fitting program. Also for spectral simulations it is necessary to include the background in the statistics. The background can be determined in different ways, depending upon the instrument. For the spectrum of a point-source obtained by an imaging detector one could extract the spectrum for example from a circular region, and the background from a surrounding annulus and scale the background to be subtracted using the area fractions. Alternatively, one could take an observation of an "empty" field, and using a similar extraction region as for ther source spectrum, one could use the empty field observation scaled by the exposure times to estimate the background. The background level B_i may also contain noise, e.g. due to Poissonion noise in the background observation. In some cases (e.g. when the raw background count rate is low) it is sometimes desirable to smooth the background to be subtracted first; in this case the *nominal background uncertainty* ΔB_i no longer has a Poissonian distribution, but its value can nevertheless be determined. In spectral simulations one may account for the background uncertainty by e.g. simply assuming that the square of the background signal-to-noise ratio $(B_i/\Delta B_i)^2$ has a Poissonian distribution.
Furthermore, there may be systematic calibration uncertainties, for example in the instrumental

effective area or in the background subtraction. These systematic errors may be expressed as a fraction of the source count rate (such that the total *systematic source uncertainty* is $\epsilon_{si}S_i$) and/or as a fraction of the subtracted background (such that the total *systematic background uncertainty* is $\epsilon_{bi}B_i$). Again, these systematic errors may vary from data channel to data channel. They should also be treated different than the statistical errors in spectral simulations: they must be applied to the simulated spectra *after* that the statistical errors have been taken into account by drawing random realisations. Also, whenever spectral rebinning is done, the systematic errors should be averaged and applied to the rebinned spectrum: a 10 % systematic uncertainty over a given spectral range may not become 1 % by just rebinning by a factor of 100, but remains 10 %, while a statistical error of 10 % becomes 1 % after rebinning by a factor of 100.

In the previous sections we have shown how to choose the optimal data binning. The observer may want to rebin further in order to increase the significance of some low-statistics regions in the spectrum, or may want to inspect the unbinned spectrum. Also, during spectral analysis or beforehand the observer may wish to discard certain parts of the spectrum, e.g. data with bad quality or parts of the spectrum that he is not interested in. For these reasons it is also usefull to have the proposed rebinning scheme available in the spectral file.

I propose to add to each data channel three logical flags (either true or false): first, if the data channel is the *first channel* of a group (f_i); next, if the data channel is the *last channel* of a group (l_i); and finally, if the data channel is *to be used* (u_i). A channel may be both first and last of a rebinning group (f_i and l_i both true) in the case of no rebinning. The first data channel $i = 1$ always *must* have f_i true, and the last data channel l_i true. Whenever there are data channels that are not used (u_i false), the programmer should check that the first data channel after this bin that is used gets f_i true and the last data channel before this bin taht is used gets l_i true. The spectral analysis package needs also to check for these conditions upon reading a data set, and to return an error condition whenever this is violated.

Finally, I propose to add the nominal energies of the data bin boundaries c_{i1} and c_{i2} to the data file. This is very usefull if for example the observed spectrum is plotted outside the spectral fitting program. In the OGIP format, this information is contained in the response matrix. I know that sometimes objections are made against the use of these data bin boundaries, expressed as energies. Of course, formally speaking the observed data bin boundaries often do not have energy units; it may be for example a detector voltage, or for grating spectra a detector position. However, given a proper response. However given a corresponding response matrix there is a one-to-one mapping of photon energy to data channel with maximum response, and it is this mapping that needs to be given here. In the case of only a single data channel (e.g. the DS detector of EUVE) one might simply put here the energies corresponding to the FWHM of the response. Another reason to put the data bin boundaries in the spectral file and not in the response file is that the response file might contain several components, all of which relate to the same data bins. And finally, it is impossible to analyse a spectrum without knowing simultaneously the response. Therefore, the spectral analysis program should read the spectrum and response together.

As a last step we must deal with multiple spectra, i.e. spectra of different detector regions that are related through the response matrix. In this case the maximum number of FITS-file extensions of 1000 is a much smaller problem then for the response matrix. It is hard to imagine that anybody might wish to fit more than 1000 spectra simultaneously; but maybe future will prove me to be wrong. An example could be the following. The supernova remnant Cas A has a radius of about $3'$. With a spatial resolution of $10''$, this would offer the possibility of analysing XMM-EPIC spectra of 1018 detector regions. At the scale of $10''$ the responses of neighbouring regions overlap so fitting all spectra simultaneously could be an option$\cdots$.

Therefore it is wise to stay here on the conservative side, and to write the all spectra of different detector regions into one extension. As a result we propose the following spectral file format. After the null primary array the first extension contains the number of regions for the spectra, as well as a binary table with the number of data channels per region (see table 5.11). This helps to allocate memory for the spectra, that are stored as one continuous block in the second extension (see table 5.12).

Table 5.11: First extension to the spectrum file

keyword	description
EXTNAME (=SPEC_REGIONS)	Contains the spectral regions in the form of a binary table
NAXIS1 = 4	There are 4 bytes in one row
NAXIS2 =	This number corresponds to the total number of regions (spectra) contained in the file (the number of rows in the table)
TFIELDS = 1	The table has 1 column, written as 4-byte integer (TFORM='1J').
TTYPE1 = 'NCHAN'	Number of data channels for this spectrum.

Table 5.12: Second extension to the spectrum file

keyword	description
EXTNAME (=SPEC_SPECTRUM)	Contains the basic spectral data in the form of a binary table
NAXIS1 = 28	There are 28 bytes in one row
NAXIS2 =	This number corresponds to the total number of data channels as added over all regions (the number of rows in the table)
TFIELDS = 12	The table has 12 columns.
TTYPE1 = 'Lower_Energy'	Nominal lower energy of the data channel c_{i1} in keV; 4-byte real.
TTYPE2 = 'Upper_Energy'	Nominal upper energy of the data channel c_{i2} in keV; 4-byte real.
TTYPE3 = 'Exposure_Time'	Net exposure time t_i in s; 4-byte real.
TTYPE4 = 'Source_Rate'	Background-subtracted source count rate S_i of the data channel in counts/s; 4-byte real.
TTYPE5 = 'Err_Source_Rate'	Background-subtracted source count rate error ΔS_i of the data channel in counts/s (i.e. the statistical error on 'Source_Rate'); 4-byte real.
TTYPE6 = 'Back_Rate'	The background count rate B_i that was subtracted from the raw source count rate in order to get the net source count rate S_i; in counts/s; 4-byte real.
TTYPE7 = 'Err_Back_Rate'	The statistical uncertainty ΔB_i of 'Back_Rate' in counts/s; 4-byte real.
TTYPE8 = 'Sys_Source'	Systematic uncertainty ϵ_{si} as a fraction of the net source count rate S_i; dimensionless; 4-byte real.
TTYPE9 = 'Sys_Back'	Systematic uncertainty ϵ_{bi} as a fraction of the subtracted background count rate B_i; dimensionless; 4-byte real.
TTYPE10 = 'First'	True if it is the first channel of a group; otherwise false. 4-byte logical.
TTYPE11 = 'Last'	True if it is the last channel of a group; otherwise false. 4-byte logical.
TTYPE12 = 'Used'	True if the channel is to be used; otherwise false. 4-byte logical.

5.8 examples

5.8.1 Introduction

In this section I present some estimates of the optical binning for som e instruments. The ideal situation would be to use the original software that generates the response matrices for these instruments, in order to derive first the optimal binning and then to produce the response matrix in the formats as proposed here. Instead I take here the faster approach to use spectra and responses in OGIP format and to derive the rebinned spectra and responses by interpolation. This is less accurate than the direct method mentioned above, but is sufficiently accurate for the present purposes. A drawback is however that given a "solid" square response matrix, it is difficult if not impossible to extract from that data the different instrumental components that contribute to the matrix, like for example the main diagonal and the split-events for a CCD detector. If we were able to do that (as the instrument-specific software should be able to do) we could reach an even larger compression of the response matrices.
I consider here data from three instruments, with low, medium and high resolution, and work out the optimal binning. The instruments are the Rosat PSPC, ASCA SIS and XMM RGS detectors.

5.8.2 Rosat PSPC

For the Rosat PSPC data I used the spectrum of the cluster of galaxies A 2199 as extracted in the central $1'$. The spectrum was provided by J. Mittaz and had a total count rate of 1.2 counts/s, with an exposure time of 8600 s. The PSPC has a low spectral resolution, as can be seen from fig. 5.7. The number of resolution elements is only 6.7. After rebinning, from the original 256 data channels only 15 remain! This shows that for Rosat PSPC data analysis rebinning is absolutely necessary.
The optimum resolution of the model energy grid below 0.6 keV and above 2 keV is dominated by the effective area curvature (upper right panel of fig. 5.7). It is also seen that the original PSPC model energy grid has a rather peculiar, variable bin size, with sometimes a few very narrow bins, apparently created in the neighbourhood of instrumental edges. This causes the "spikes" in the number of model bins to be merged. Note that this number is a quantity valid for only a single bin; if e.g. a thousand bins need to be merged, this means in practice that only the next few bins are merged, since these next bins will have a more decent value of the number of bins to be merged.

Table 5.13: Rebinning of Rosat PSPC data

Parameter	original OGIP	final value	reduction factor (%)
Data channels	256	15	5.9
Model bins	729	153	21.0
Matrix elements	186624	2295	1.2

Table 5.13 summarizes the gain in bin size as obtained after optimum binning of the data and response matrix. The reduction is quite impressive.

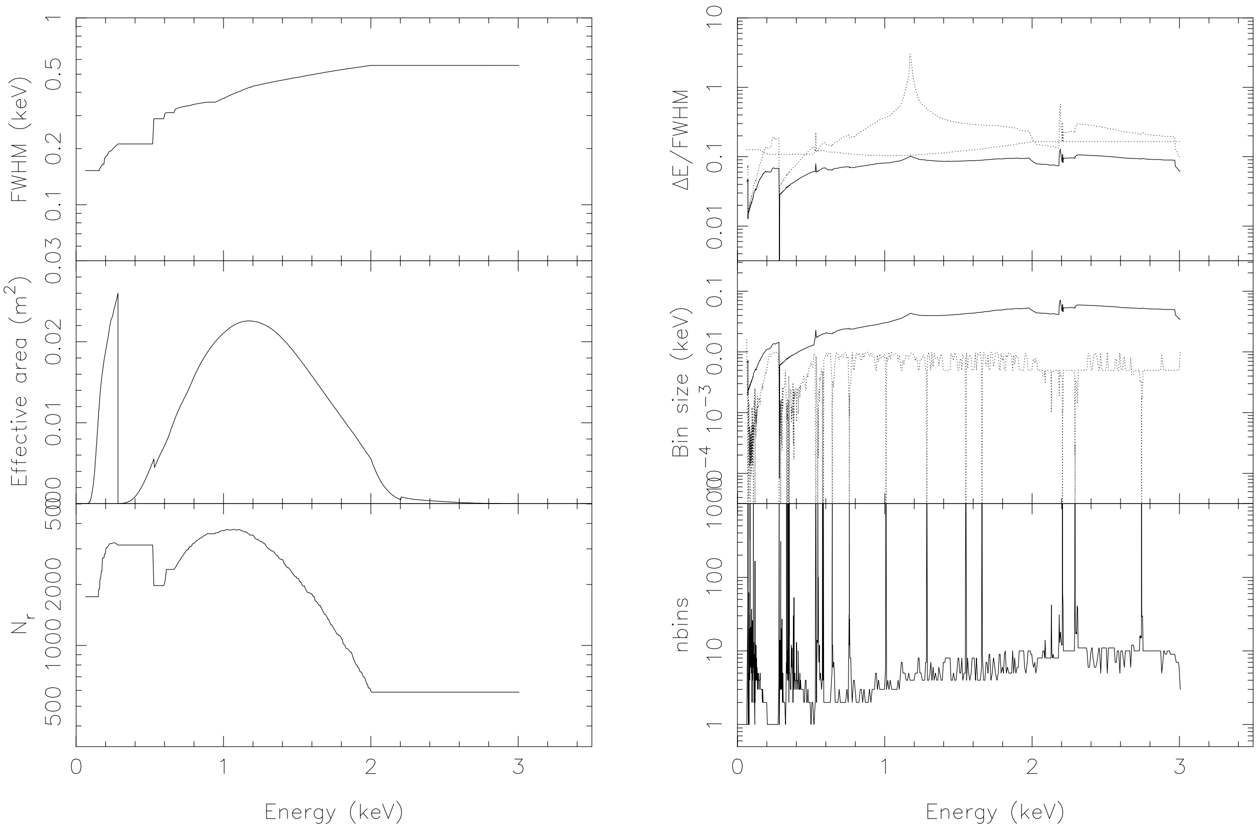


Figure 5.7: Binning of the Rosat PSPC response. Left panel, from top to bottom: instrumental FWHM, effective area $A(E)$, and effective number of events N_r within one resolution element. Right panel, top frame: optimum model energy bin size in units of the FWHM, eqn. (5.61) as the solid line, with the contribution from the effective area eqn. (5.60) as the upper dotted line, the contribution from the model binning eqn. (5.48) as the lower dotted line; middle frame: optimum model energy bin size in keV (solid line) and original bin size in the OGIP-format matrix (dotted line); bottom frame: number of original model energy bins to be taken together.

5.8.3 ASCA SIS0

For the ASCA SIS0 data I used the spectrum of the supernova remnant Cas A. The spectrum was provided by J. Vink and had a total count rate of 52.7 counts/s, with an exposure time of 6000 s. There are 86.2 resolution elements in the spectrum, which extended from 0.35 to 12.56 keV. The results are summarized in fig. 5.8. In the effective number of events per data channel (lower left panel) clearly the strong spectral lines of Si, S and Fe can be seen. The present modelling shows that a slightly smaller bin size is needed near those lines (right panels). Contrary to the PSPC case, for the SIS detector the optimum resolution of the model energy grid is dominated verywhere by the requirements for the model binning, not the effective area curvature (upper right panel of fig. 5.8). This is due to the higher spectral resolution as compared to the PSPC. Only near the instrumental edges below 1 keV a small contribution from the

effective area exists.
The original OGIP-matrix uses a constant model energy grid bin size of 10 eV, as can be seen from the middle-right panel of fig. 5.8. It can be seen from that panel that below 3 keV that binning is much too coarse; one should take into account that the OGIP-matrices are used in the approximation of all flux coinciding with the bin centroid; for that approximation however the optimum bin size would be even a factor of ~10 smaller! On the other hand, the ASCA model grid is over-sampled at high energies.

Table 5.14: Rebinning of ASCA SIS0 data

Parameter	original OGIP	final value	reduction factor (%)
Data channels	1024	192	18.8
Model bins	1168	669	57.2
Matrix elements	503668	69453	13.8

Table 5.14 summarizes the gain in bin size as obtained after optimum binning of the data and response matrix. The reduction is again quite large. It should be noted however that since at low energies the model energy grid binning of the original OGIP data is undersampled, so that the final number of model bins in the optimum case of direct response generation in the new format might be somewhat larger. On the other hand, decomposing the matrix into its physical components reduces again the number of matrix elements considerably, since most of the non-zero matrix elements are caused by the low-resolution split-events that need not a very fine energy mesh.

5.8.4 XMM RGS data

I applied the procedures of this paper also to a simulated RGS spectrum of Capella. The original onboard 3×3pixel binning produces 3068 data channels. The OGIP response matrix used is the one provided by C. Erd, which was calculated for a smooth grid of 9300 model energy bins. The calculation is done for a single RGS. The spectrum has a count rate of 4.0 counts/s, with an exposure time of 80000 s.
After rebinning, there remain only 1420 new data channels. Most of these new data channels contain 2 old data channels, only at the boundaries of the detector 3 old bins are taken together. The number of resolution elements is 611. After rebinning, from the original 3068 data channels 1420 remain.
The optimum model binning is completely dominated by the requirements for the model binning. Effective area effects can be neglected (fig. 5.9). The 8 discrete features in several frames of this figure are artefacts due to the gaps between the 9 CCD detectors of each RGS. This is due to the fact that we started with a pre-fab OGIP type response matrix.
Table 5.15 summarizes the gain in bin size as obtained after optimum binning of the data and response matrix. At first sight the reduction appears not to be too impressive. This is however illusionary. The "classical" OGIP response matrix that was used as the basis for my calculations is heavily under-sampled. Would it have been sampled properly, then the model energy grid should have a size of a few hundred thousand bins! Since this causes the response matrix to be massive and disk consuming, the undersampled matrix was created; this matrix can be safely

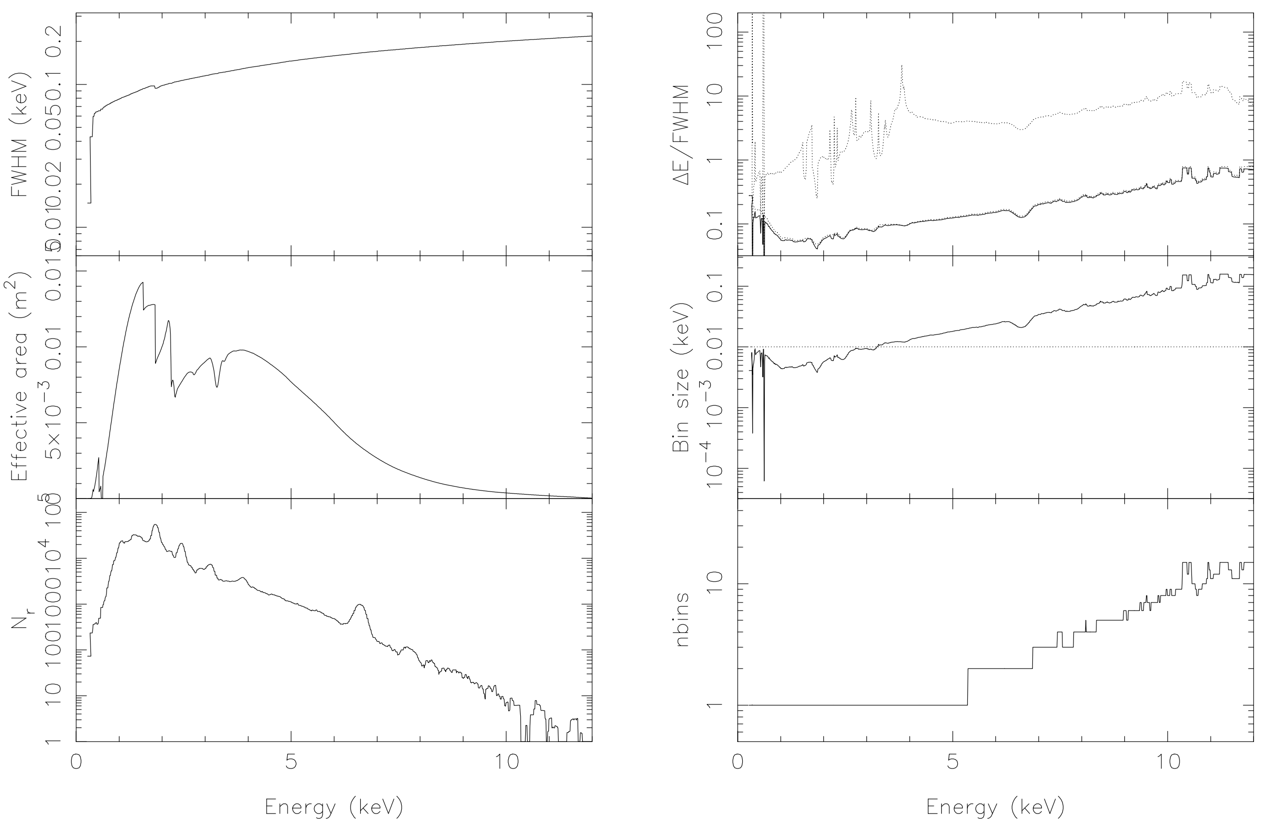


Figure 5.8: Binning of the ASCA SIS response. Panels as for the Rosat PSPC data.

Table 5.15: Rebinning of XMM RGS data

Parameter	original OGIP	final value	reduction factor (%)
Data channels	3068	1420	46.3
Model bins	9300	5964	64.1
Matrix elements	2525647	783631	31.0

used for spectral simulations, but is insufficient for flight data analysis. It is exactly for that reason that the curren theoretical framework has been set up!

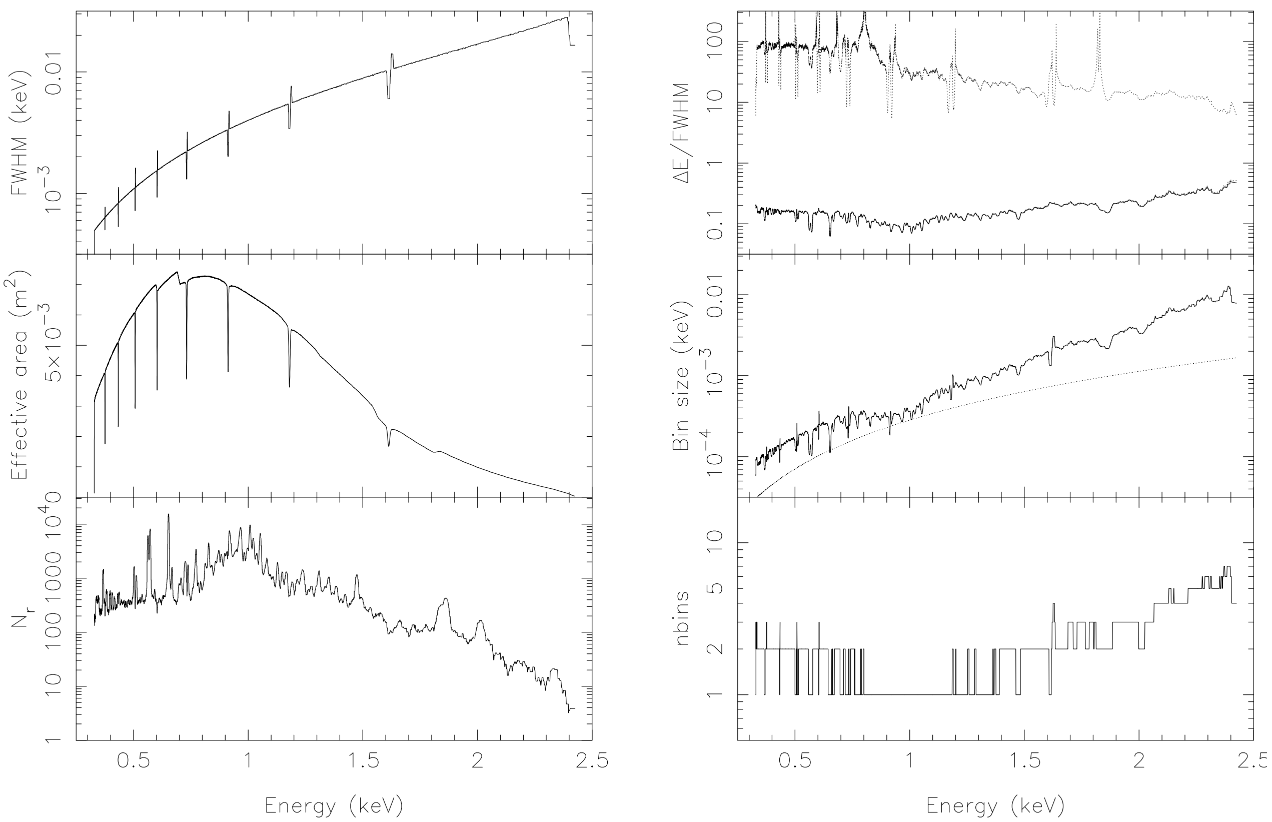


Figure 5.9: Binning of the XMM RGS response. Panels as for the Rosat PSPC data.

5.9 References

Arnaud, K.A., 1996, Astronomical Data Analysis Software and Systems V, eds. Jacoby G. and Barnes J., p17, ASP Conf. Series volume 101.

Eisenhart, C., 1938, Bull. Am. Math. Soc. 44, 32.

Jerri, A.J., 1977, Proc. of the IEEE 65, 1565.

Kaastra, J.S., Mewe, R., Nieuwenhuijzen, H., 1996, UV and X-ray spectroscopy of astrophysical and laboratory plasmas, eds. K. Yamashita and T. Watanabe, p. 411, Univ. Acad. Press.

Shannon, C.E., 1949, Proc. IRE 37, 10.

Weiss, P., 1963, Am. Math. Soc. Notices 10, 351

Chapter 6

Installation and testing

6.1 Testing the software

After installing the software, SPEX can be tested quickly as follows.
Goto the directory $(SPEX)/test. Type the following in the terminal window:

```
spex
log exe test
quit
```

The above should produce a plot mytest.cps, which should be identical to the file test.cps in the same directory (except for the date string).
Also, at the end of your output on the terminal you should have the following (except for machine-dependent rounding errors):

```
Chi-squared value :      1009.59
 Degrees of freedom:   266
 parameter          Chi**2      Delta          Delta
   value            value       parameter     Chi**2
----------------------------------------------------
   5.66249         1010.50     -6.613350E-02    0.91
   5.59636         1013.28     -0.132267        3.69
   5.65690         1010.66     -7.172966E-02    1.07
   5.63537         1011.41     -9.326029E-02    1.81
   5.63051         1011.60     -9.811878E-02    2.01
   5.63117         1011.58     -9.745693E-02    1.98
   5.79476         1010.45      6.613350E-02    0.85
   5.86089         1012.87      0.132267        3.27
   5.80945         1010.87      8.082247E-02    1.27
   5.82815         1011.43      9.951925E-02    1.83
   5.83308         1011.62      0.104452        2.03
   5.83240         1011.59      0.103778        2.00
 Parameter:    5.7286     Errors:  -9.74569E-02,  0.10378
 Normal end of SPEX
 You made 1 errors.
 Thank you for using SPEX
 STOP
```

The above test run is a simple single temperature fit to the MOS1 spectrum of an annulus in the cluster of galaxies A 1795.

Chapter 7

Acknowledgements

7.1 Acknowledgements

We would like to express our special thanks to numerous people who have substantially contributed to the development of various aspects of this work, in alphabetical order:
Frans Alkemade, Gert-Jan Bartelds, Ehud Behar, Alex Blustin, Irma Eggenkamp, Andrzej Fludra, Ed Gronenschild, Theo Gunsing, Jörgen Hansen, Wouter Hartmann, Kurt van der Heyden. Fred Jansen, Jelle Kaastra, Jim Lemen, Duane Liedahl, Pasquale Mazzotta, Rolf Mewe, Hans Nieuwenhuijzen, Bert van den Oord, Ton Raassen, Hans Schrijver, Karel Schrijver, Janusz Sylwester, Rob Smeets, Katrien Steenbrugge, Jeroen Stil, Dima Verner, Jacco Vink, Frank van der Wolf.
During the years the many people have contributed as follows:
Most software is written by Kaastra. Nieuwenhuijzen contributed to the software design.
The line and continuum emission routines were written by Kaastra, based on previous routines by Mewe, Gronenschild, van den Oord, and Lemen (1981-1986).
The line and continuum emission data are based on the work by Mewe and Kaastra. The photo-ionization cross-sections have been provided by Verner. Liedahl contributed to the Fe-L calculations. Nieuwenhuijzen, Raassen and Hansen contributed to the extension of the atomic database.
Behar and Raassen contributed to the atomic data for the absorption models.
Mazzotta worked on the ionization balance.
The non-equilibrium ionization balance routine was developed by Jansen, based on earlier work by Hans Schrijver (1974) and Gronenschild (1981).
The SNR models are based on the work by Kaastra and Jansen (1992), with further extensions and testing by Stil, Eggenkamp and Vink.
Lemen, Smeets, and Alkemade developed an earlier version (1986-1990) of a spectrum synthesis program.
The philosophy of the fitting procedure is based on the experiences obtained with the fitting of EXOSAT data, especially on the software package ERRFIT designed by Jansen and Kaastra.
The original DEM modelling was based on the work by Sylwester and Fludra (1980), while the current DEM regularisation algorithm has been designed by Karel Schrijver and Alkemade. Bartelds contributed to the testing of these models.
The first version (1.0) of the documentation has been prepared – with starting help by Gunsing – by van der Wolf and Nieuwenhuijzen. The second version was set-up by Kaastra with contributions from Steenbrugge, van der Heyden and Blustin.

Chapter 8

References

Bibliography

[1973] Allen, C.W., 1973, Astrophysical quantities, 3rd Ed., Athlone Press, London

[1989] Anders, E., & Grevesse, N., 1989, Geochimica et Cosmochimica Acta 53, 197

[1985] Arnaud, M., & Rothenflug, R., 1985, A&A Supp 60, 425

[1992] Arnaud, M., & Raymond, J.C., 1992, ApJ 398, 394

[2004] Bautista, M.A., Mendoza, C., Kallman, T.R., & Palmeri, P., 2004, A&A 418, 1171

[2002] Behar, E., Sako, M., & Kahn, S.M., 2002, ApJ 563, 497

[2003] Beiersdorfer, P., 2003, Ann. Rev. Astron. Astrophys., 41, 343

[1992] Grevesse, N., Noels, A., Sauval, A.J., 1992, in: Coronal Streamers, Coronal Loops and Coronal and Solar Wind Composition, Proc. 1st SOHO Workshop, ESA SP-348, ESA Publ. Div., ESTEC, Noordwijk, p. 305

[1998] Grevesse, N., & Sauval, A.J., 1998, ScScRev 85, 161

[1989] Kaastra, J.S., & Barr, P., 1989, A&A 226, 59

[1993] Kaastra, J.S., & Jansen, F.A., 1993, A&A Supp 97, 873

[2004] Kaastra, J.S., Tamura, T., Peterson, J.R., et al., 2004, A&A 413, 415

[1991] Laor, A., 1991, ApJ 376, 90

[1995] Magdziarz, P., Zdziarski, A.A., 1995, MNRAS 273, 837

[2004] Mendoza, C., Kallman, T.R., Bautista, M.A., & Palmeri, P., 2004, A&A 414, 377

[1983] Morrison, R., McCammon, D., 1983, ApJ 270, 119

[2001] Nahar, S.N., Pradhan, A.K., & Zhang, H.L., 2001, Phys. Rev. A. 63, 060701

[2003] Palmeri, P., Mendoza, C., Kallman, T.R., Bautista, M.A., & Mélendez, M., 2003, A&A 410, 359

[2003] Peterson, J.R., Kahn, S.M., Paerels, F.B.S., et al. 2003, ApJ 590, 207

[1999] Phillips, K.J.H., Mewe, R., Harra-Murnion, L.K., et al., 1999, A&A Supp. 138, 381

[2004] Porquet, D., Kaastra, J.S., Page, K.L., et al., 2004, A&A 413, 913

[1976] Ross, J.E., & Aller, L.H., 1976, Science 191, 1223

[1994] Rumph, T., Bowyer, S., & Vennes, S., 1994, AJ 107, 2108

[1979] Rybicki, G.B., & Lightman, A.P., 1979, Radiative processes in astrophysics, Wiley, New York

[1995] Verner, D.A., & Yakovlev, D.G., 1995, A&AS 109, 125

[1996] Verner, D.A., Verner, E.M., & Ferland, G.J., 1996, At. Data Nucl. Data Tables 64, 1

[1995] Wheaton, W.A., Dunklee, A.L., Jacobson, A.S., et al., 1995, ApJ 438, 322

[1994] Zycki, P.T., Czerny, B., 1994, MNRAS 266, 653

[1999] Zycki, P.T., Done, C., Smith, D.A., 1999, MNRAS 305, 231

Index