Chapter 1

How to run SPEX

SPEX is a spectral fitting program used to fit high-resolution X-ray spectra. The code contains several simple and detailed models that are able to deal with the radiative processes observed in the X-ray band. Because SPEX has a command-line interface, a first-time user should get familiar with the syntax of the commands to be able to work with it. This chapter provides some basic commands and threads to fit X-ray spectra.

1.1 The SPEX data format

The data files containing the spectrum of the source and the response need to be in the correct format. In the SPEX installation, we provide a program called `trafo` to convert OGIP standard fits files into SPEX format (see Chapter 2 for an explanation of trafo). In this chapter, we assume that you already have spectra in SPEX format. For example, the demo spectra from the SPEX web site.

SPEX needs two files per spectrum:

- `<filename>.spo` – This file contains the countrate per energy bin for the source (D_i), as well as the background countrate and the errors (σ_i).
- `<filename>.res` – This file contains the instrumental response: the energy redistribution and effective area ($R_{ij}\ A_j$).

In order to calculate the observed model spectrum, SPEX uses this integral equation to account for the imperfections caused by the instrument:

$$D(c) = \int R(c,E)A(E)S(E)dE \tag{1.1}$$

$$D_i = \sum_{j=1}^{n} R_{ij}A_jS_j \tag{1.2}$$

The `.res` and `.spo` files are so-called FITS files. This is a data format widely used in Astronomy. FITS files can contain images as well as data tables. The software package FTOOLS provided by NASA contains a large number of tools to manipulate FITS files (see references). If you are interested, then you can launch `flaunch` to see which tools are available.

1.2 Loading spectra into SPEX

The SPEX program is started by entering `spex` in a linux terminal window. In the following sections we describe one run of the program. To start SPEX do this:

```
user@linux:~> spex
 Welcome user to SPEX version 2.01.05

SPEX>
```

First, we have to load the data files. This is done using the command `data`. It is a general thing in SPEX that filename extensions are not typed explicitly when issuing a command. If you have a file called `filename.spo` and `filename.res` then you type:

```
SPEX> data filename filename
```

The responsefile (.res) is entered first and then the file containing the spectrum (.spo). You can avoid confusion by giving the same filename to both .res and .spo files. Remember that the order of the words in the commands is very important!

To save you from typing a lot, many commands can be abbreviated by typing just the first few characters. For example, `da` is equivalent to `dat` and `data`.

1.3 Plotting the data

If the `data` command was successful, we can now have a look at the spectra. SPEX offers a lot of different plot commands through the PGPLOT package (known from, for example, Fortran programs). Below is a simple example to create a linear-linear plot in Å:

```
SPEX> plot dev xs
SPEX> plot type data
SPEX> plot x lin
SPEX> plot y lin
SPEX> plot ux a
SPEX> plot uy a
SPEX> plot rx 8. 35.
SPEX> plot ry 0. 0.05
SPEX> plot set 1
SPEX> plot cap ut text "This is my plot"
SPEX> plot cap lt disp false
SPEX> plot cap id disp false
SPEX> plot
```

The sequence above opens a PGPLOT window (`dev xs`) and tells SPEX that the plot will contain data with linear axes (`x lin` and `y lin`) in unit Å (`ux a` and `uy a`). `rx` and `ry` are the ranges on the x and y axes, respectively. Then the color of the data, background spectrum and model are set. The last commands beginning with `plot cap` remove some standard titles and other text around the plot. After you define the plot like in the example above, you can plot it with a single `plot` command.

The y-axis in this plot is in counts s^{-1} Å^{-1}. Ångstrom is not the only unit used in high-energy astrophysics. Usually, the energy of the photons is expressed in keV. In SPEX you can use keV by writing `k` instead of `a` in all commands. For example, `plot ux k` to use keV for the x-axis.

1.4 Ignoring and rebinning

High-resolution X-ray spectra from Chandra and XMM-Newton are usually oversampled (e.g. the energy bins are much smaller than the spectral resolution) and contain a lot more channels then is useful.

Therefore, it is necessary to remove wavelength intervals which contain bad data and rebin your spectrum. In the next example we bin the spectrum over the 8–35 Å range with a factor of 5 and ignore the rest of the spectrum:

```
ign 0:8 unit a
ign 35:100 unit a
bin 8:35 5 unit a
```

The words `unit a` tells SPEX that the ranges (for example 8.0:35.0) are given in Å. If you work with more than one spectrum (from more than one instrument), you can add an extra instrument statement:

```
ign ins 1:2 0:8 unit a
ign ins 1:2 35:100 unit a
bin ins 1:2 8:35 5 unit a
```

Here, instrument 1 to 2 are binned with a factor of 5 over the 8–35 Å range.

1.5 Defining a model

Now we have a clean and rebinned spectrum that is ready to fit. Before we can start fitting, we first need to define a model. It's equivalent to $S(E)$ in Eq. 1.1. The model can contain one or more of these components:

- `absm` Model for interstellar absorption.
- `reds` Redshift.
- `po` Powerlaw.
- `ga` Gaussian.

And there are more (see the SPEX manual)! The following command sequence defines a simple powerlaw model at a certain redshift and absorbed by the interstellar medium. The individual components of the model are loaded one-by-one with the `com` command:

```
SPEX> com reds
SPEX> com absm
SPEX> com po
SPEX> com rel 3 1,2
```

The last command (`com rel 3 1,2`) tells SPEX that component 3, the powerlaw, is first redshifted by component 1 and then absorbed by component 2. The order of the 1 and the 2 is important! Always think what happens in which order on the way from the source to the telescope.

For most sources the distance is more or less known. To get a right luminosity estimate for the source, the expected distance has to be provided to SPEX:

```
SPEX> dist 0.1 z
Distances assuming H0 =  50.0 km/s/Mpc and q0 = 0.500
Sector    m      A.U.        ly        pc       kpc       Mpc  redshift        cz
----------------------------------------------------------------------------------
1 1.894E+25 1.266E+14 2.002E+09 6.139E+08 6.139E+05  613.8689    0.1000   29979.2
----------------------------------------------------------------------------------
```

With this command, the distance to the source is set to a redshift of 0.1. The derived distances for this cosmology are in the output of the `dist` command.

Now we have to estimate the initial parameters. With the command `par show` we can see which parameters there are:

```
SPEX> par show
----------------------------------------------------------------------------------
sect comp mod  acro parameter with unit     value      status    minimum   maximum

   1    1 reds z    Redshift               0.000000     frozen   -1.0      1.00E+10

   1    2 absm nh   Column (1E28/m**2)   9.9999997E-05 thawn     0.0      1.00E+20
   1    2 absm f    Covering fraction     1.000000     frozen    0.0       1.0

   1    3 pow  norm Norm (1E44 ph/s/keV)  1.000000     thawn     0.0      1.00E+20
   1    3 pow  gamm Photon index          2.000000     thawn    -10.       10.
   1    3 pow  dgam Photon index break    0.000000     frozen   -10.       10.
   1    3 pow  e0   Break energy (keV)   1.0000000E+10 frozen    0.0      1.00E+20
   1    3 pow  b    Break strength        0.000000     frozen    0.0       10.
   1    3 pow  type Type of norm          0.000000     frozen    0.0       1.0
   1    3 pow  elow Low flux limit (keV)  2.000000     frozen   1.00E-20  1.00E+10
   1    3 pow  eupp Upp flux limit (keV)  10.00000     frozen   1.00E-20  1.00E+10
   1    3 pow  lum  Luminosity (1E30 W)   1.000000     frozen    0.0      1.00E+20

--------------------------------------------------------------------------------
Fluxes and restframe luminosities between   2.0000     and    10.000     keV

 sect comp mod   photon flux   energy flux nr of photons    luminosity
          (phot/m**2/s)      (W/m**2)   (photons/s)           (W)
    1    3 pow    0.00000       0.00000       0.00000       0.00000
```

We can set the parameters using the `par` command. The first "1" in column "sect" can usually be ignored. The commands then look like this:

```
SPEX> par 1 z val 0.1
SPEX> par 2 nh val 2.E-3
SPEX> par 3 norm val 1.E+10
SPEX> par gamm val 1.5
```

The last component number used is saved, so in the last line we can skip typing the number 3 after `par`. Then, we run `par show` again to see what happened:

```
SPEX> par show
----------------------------------------------------------------------------------
sect comp mod  acro parameter with unit     value      status    minimum   maximum

   1    1 reds z    Redshift               0.100000     frozen   -1.0      1.00E+10

   1    2 absm nh   Column (1E28/m**2)   2.0000001E-03 thawn     0.0      1.00E+20
   1    2 absm f    Covering fraction     1.000000     frozen    0.0       1.0

   1    3 pow  norm Norm (1E44 ph/s/keV)  1.000000E+10 thawn     0.0      1.00E+20
   1    3 pow  gamm Photon index          1.500000     thawn    -10.       10.
   1    3 pow  dgam Photon index break    0.000000     frozen   -10.       10.
   1    3 pow  e0   Break energy (keV)   1.0000000E+10 frozen    0.0      1.00E+20
   1    3 pow  b    Break strength        0.000000     frozen    0.0       10.
   1    3 pow  type Type of norm          0.000000     frozen    0.0       1.0
   1    3 pow  elow Low flux limit (keV)  2.000000     frozen   1.00E-20  1.00E+10
   1    3 pow  eupp Upp flux limit (keV)  10.00000     frozen   1.00E-20  1.00E+10
   1    3 pow  lum  Luminosity (1E30 W)  5.6014867E+08 frozen    0.0      1.00E+20
```

```
--------------------------------------------------------------------------------
Fluxes and restframe luminosities between   2.0000     and    10.000     keV

 sect comp mod   photon flux   energy flux nr of photons    luminosity
            (phot/m**2/s)      (W/m**2)   (photons/s)           (W)
    1    3 pow    0.00000       0.00000       0.00000       0.00000
```

Finding the right initial values for the parameters is a game of trial and error. To see whether you are going in the right direction, you can calculate the model with the command `calc` and `plot` again. If you see the model appear in your screen, then the model is close enough to be fitted. Especially the normalization of the powerlaw (`3 norm`) can vary a lot depending on the countrate of the source.

1.6 Fitting the data

We are ready to fit the data! SPEX has a nice feature to look at the progress of the fit. To activate this feature you have to give the command `fit print 1`. If your initial parameters were acceptable, you can see the model converge to the data in the plot window after you entered the `fit` command. When the fit is done, then the parameters and χ^2 are printed on screen. If the χ^2 value is close to the number of degrees of freedom, then your fit is acceptable. Sometimes more runs of the command `fit` are necessary after changing some initial parameters. This is especially true when using complex models. Again this is a game of trial and error.

You also might want to fix or free certain parameters to see if they can be constrained. In SPEX fixing is `f` (frozen) and freeing is `t` (thawn). You can free the redshift and fix the N_{H} by the following commands:

```
SPEX> par 1 z stat t
SPEX> par 2 nh stat f
```

1.7 Calculating errors

When the fit is acceptable, you might want to know the uncertainties on your fitted parameters. Errors are determined one-by-one by fixing the parameter to some value and calculate the $\Delta\chi^2$ with respect to the best fit. If you want to know the 1σ error on the parameter, you need to know its values at $\Delta\chi^2 = 1$. This is done by the `error` command. First you have to set the desired $\Delta\chi^2$ in SPEX: `error dchi 1.` After this you can calculate the error for each parameter. For example redshift:

```
SPEX> error 1 z
```

1.8 Making life easier...

In this short manual you have seen a lot of commands, but to avoid typing too much you want to use some identical series of commands every time you fit a certain spectrum. For example, you don't want to type all plot commands again when making a plot. Therefore, the program has a command to solve this problem. With the command `log exe filename` you can execute a number of commands at the same time. The numbers are read from a normal text file with (in this case) the name `filename.com`. Again the extension `.com` should not be typed explicitly. Below is an example to setup a plot for an EPIC spectrum (range 0.2–10.0 keV) with a small frame that shows residuals. Note that you can put any command in such a command file and you can make comment lines by putting a # sign in front of the line.

```
# This is a command file that creates a plot with residuals.
plot dev xs
plot type data
plot x log
plot y log
plot rx 0.2 10.
plot ry 0.0001 10.
plot back disp t
plot set 1
plot data col 1
plot model col 2
plot back col 1
plot set all
plot frame new
plot frame 2
plot type chi
plot uy rel
plot x log
plot rx 0.2 10.
plot ry -0.5 0.5
plot view def f
plot view x 0.08 0.92
plot view y 0.1 0.3
plot cap y text "Rel. Error"
plot cap ut disp f
plot cap lt disp f
plot cap id disp f
plot frame 1
plot view def f
plot view x 0.08 0.92
plot view y 0.3 0.9
plot cap x disp f
plot cap id disp f
plot cap ut disp f
plot box numlab bot f
```

1.9 Saving your work

There are several ways in SPEX you can save your work. Below you find a few examples to save your commands, output or plots.

1.9.1 Saving a plot

These commands open a PostScript plot device with filename `filename.ps`, then they plot your figure in the PS file and closes the device:

```
SPEX> plot dev cps filename.ps
SPEX> plot
SPEX> plot close 2
```

1.9.2 Saving commands

If you want to save all commands that you execute to an ASCII file (`filename.com`), then type `log save filename`. Do not forget to close the file at the end of the session by typing `log close save`. The saved commands in the textfile can be executed again by the `log exe filename` command.

1.9.3 Saving output

In the same way as in the previous example, you can also save the output on your screen by typing `log out filename` (the file will be an ASCII file called `filename.out`). You can close the file with `log close out`. This command is very useful to save your parameters and errors.

1.10 Quitting the program

Just type `quit`. . .

1.11 Tips & Tricks

- If you make a typo in a command or you want to do the same command again, then push the `arrow-up` button on your keyboard. There is an entire history of your commands there.
- The `Tab` key is able to automatically complete the command you are typing. In case there are more possibilities, it shows them all.

1.12 References

- The full SPEX manual can be found here:
  `http://www.sron.nl/files/HEA/SPEX/manuals/manual.pdf`
- FTOOLS software package to open and manipulate FITS files:
  `http://heasarc.nasa.gov/lheasoft/ftools/ftools_menu.html`