# HiXUV 2022 workshop threads
(SPEX edition)

Junjie Mao

Latest update: August 4, 2022

## Contents

# 1 Introduction to SPEX

SPEX is an X-ray spectral fitting package optimized for the analysis and interpretation of high-resolution cosmic X-ray spectra. The software is especially suited for fitting spectra obtained by current and future X-ray observatories like XMM-Newton, Chandra, XRISM, Athena, and HUBS. Like other astrophysical plasma codes (e.g., APEC, CHIANTI, Cloudy, XSTAR), SPEX is built on a large and ever-increasing atomic database. Continuous development of the SPEX code and its atomic database is essential to advance our knowledge of the warm and hot astrophysical plasmas in the Universe.

SPEX is currently developed by Prof. Jelle S. Kaastra, Dr. Jelle de Plaa, Dr. Liyi Gu at SRON Netherlands Institute for Space Research and collaborators around the world. The latest version of SPEX 3.07.00 was just released on August 1st, 2022.

Note: Starting in the 1970s, Prof. Rolf Mewe and his colleagues (at the Space Research Laboratory in Utrecht) started the development of a spectral analysis code to study optically thin X-ray emitting plasmas (Mewe, 1972). On Jan. 17 – 19 2022, an online workshop celebrated the 50th year of astronomical X-ray spectroscopy in the Netherlands (see the workshop website for talk slides and recordings).

## 1.1 Why use SPEX?

Here are a few reasons to start with,

- Making the "black box" more transparent via the on-the-fly calculation. This allows users to check the details of the current calculation, including ionization balance, level population, collision rate, recombination rate, etc.
- No plasma models (and atomic databases) are perfect. When we find mismatches between the plasma model and observed data, digging into the black box makes it handy to figure out possible issues.
- Different plasma models in SPEX are built upon the same atomic database.
- Plasma models are relatively simple (ideal) but our Universe is much more sophisticated.
  - For the collisional ionized equilibrium (CIE) plasma in SPEX, the multi-temperature differential emission measure feature is likely more close to the nature of many astrophysical plasmas.
  - Plasma models in SPEX not only depend on plasma temperature but also number density.
  - For photoionization modeling in SPEX, it simultaneously fits the broadband SED (all other photoionization models require the SED to be fixed).

# 2 Install SPEX

## 2.1 Overview

SPEX is a software package optimized for high-resolution spectroscopy analysis. In this thread, we show how to install SPEX.

Reference: https://spex-xray.github.io/spex-help/getstarted/runspex.html

### 2.1.1 Download

The SPEX package can be downloaded via one of the following links:

- The SPEX Zenodo page provides the install files for all the versions. Proper citations to different versions of SPEX can also be found here (e.g., DOI: 10.5281/zenodo.6948884 for v3.07.00), as well as the list of papers using the SPEX code.
- The latest version of the packages is available in the SPEX folder of the workshop cloud drive (pwd: hixuv):
  - For Mac users, one can download SPEX-3.07.00-MacOS.dmg (admin install) and SPEX-3.07.00-MacOS.tar.gz (non-admin install)
  - For Linux users, one can download SPEX-3.07.00-Linux-x86_64.tar.gz

### 2.1.2 Install (for Linux users)

Unpack the tar file in the destination directory (for example /path/to/folder), where you should have the write access.

```
user$ cd /path/to/folder
user$ mv ~/Download/spex-3.07.00-Linux-Intel.tar.gz ./
user$ tar -xvfz spex-3.07.00-Linux-Intel.tar.gz
```

The tar file will create a directory called SPEX-3.07.00-Linux where the entire program is installed.

**It is necessary to set the environment variable SPEX90 properly before running SPEX**. This is achieved by modifying the spexdist.sh (for bash shell) or spexdist.csh (for C-type shell). This script can be found at /path/to/folder/SPEX-3.07.00-Linux. Please check whether you have the writing access. If not, use chown and/or chmod command to gain access.

```
user$ # for Linux user
user$ cd /path/to/folder/SPEX-3.07.00-Linux
user$ ls
NEWS bin data lib pgplot python spexdist.csh spexdist.sh
```

The following command can help you identify which shell you are using:

```
user$ which $SHELL
```

The screen output will be /bin/bash for bash shell and /bin/csh for C-type shell.
Note: Lines starting with the # symbol are comments.
The following command can help you to change shell:

```
user$ # Switch to bash shell
user$ chsh -s /bin/bash
user$ # Switch to C-type shell
user$ chsh -s /bin/csh
```

Open the shell script (spexdist.sh or spexdist.csh) with your favorite text editor (e.g., vim, nano, Emacs). Modify the file to set the environment variable SPEX90 properly:

```
user$ # for bash shell
user$ export SPEX90= /path/to/folder/SPEX-3.07.00-Linux/
user$ # for C-type shell
user$ setenv SPEX90 /path/to/folder/SPEX-3.07.00-Linux/
```

An exemplary spexdist.sh file where a Linux user has already set the SPEX90 environment correctly:

```
#!/bin/bash
export SPEX90=/path/to/folder/SPEX-3.07.00-Linux
export PGPLOT_DIR=$SPEX90/pgplot
export PATH=$PATH:$SPEX90/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SPEX90/lib
export PYTHONPATH=$PYTHONPATH:$SPEX90/python
# Readline functionality
if [ ! -d "$HOME/.spex_d" ]
then
    mkdir $HOME/.spex_d
    echo "-readline" > $HOME/.spex_d/spexrc
fi
```

An exemplary spexdist.csh file where a Linux user has already set the SPEX90 environment correctly:

```
#!/bin/csh
setenv PGPLOT_DIR $SPEX90/pgplot
setenv PATH $PATH:$SPEX90/bin
setenv LD_LIBRARY_PATH
$LD_LIBRARY_PATH:$SPEX90/lib
setenv PYTHONPATH $PYTHONPATH:$SPEX90/python
# Readline functionality
if ( ! -d $HOME/.spex_d ) then
    mkdir $HOME/.spex_d
    echo "-readline" > $HOME/.spex_d/spexrc
endif
limit stacksize unlimited
```

#### 2.1.3 Install (for Mac admin install)

Open the downloaded .dmg file and follow the instructions on the screen. The application is typically installed at /opt/spex/.

Note: If MacOS complain about not being able to verify the developer of the software, please right-click the .pkg file and then choose "Open with" → "Installer" → "Open". Now, you can proceed with the installation.

The ${SPEX90} environment variable has been written in the /opt/spex/spexdist.sh or /opt/spex/spexdist.csh file.

For M1 chip users, it might be necessary to change the value for ulimit (the last line in the spexdist.sh or spexdist.csh file). The default value is 65532, which will run into error when sourcing the spexdist.sh or spexdist.csh file (see this ticket on Github for details).

The following command can help you to the proper stack size to use:

```
user$ ulimit -a
core file size (blocks, -c) 0
data seg size (kbytes, -d) unlimited
file size (blocks, -f) unlimited
max locked memory (kbytes, -l) unlimited
max memory size (kbytes, -m) unlimited
open files (-n) 256
pipe size (512 bytes, -p) 1
stack size (kbytes, -s) 8176
cpu time (seconds, -t) unlimited
max user processes (-u) 2666
virtual memory (kbytes, -v) unlimited
```

Modify the stack size in spexdist.sh or spexdist.csh to the value obtained above with

your favorite text editor (e.g., vim, nano, Emacs). Please check whether you have the writing access. If not, use chown and/or chmod command to gain access.

```
#!/bin/bash
export SPEX90=/path/to/folder/SPEX-3.07.00-Linux
export PGPLOT_DIR=$SPEX90/pgplot
export PATH=$PATH:$SPEX90/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SPEX90/lib
export PYTHONPATH=$PYTHONPATH:$SPEX90/python
# Readline functionality
if [ ! -d "$HOME/.spex_d" ]
then
    mkdir $HOME/.spex_d
    echo "-readline" > $HOME/.spex_d/spexrc
fi
ulimit -s 8176
```

#### 2.1.4 Install SPEX (for Mac non-admin install)

Compared to the admin install, non-admin install allows users to specify where SPEX is installed and multiple versions of SPEX can be installed in different folders.

Unpack the tar file in the destination directory (for example /path/to/folder), where you should have the write access.

```
user$ cd /path/to/folder
user$ mv ~/Download/spex-3.07.00-MacOS.tar.gz ./
user$ tar -xvfz spex-3.07.00-MacOS.tar.gz
```

The tar file will create a directory called SPEX-3.07.00-Darwin where the entire program is installed.

**It is necessary to set the environment variable SPEX90 properly before running SPEX**. This is achieved by modifying the spexdist.sh (for bash shell) or spexdist.csh (for C-type shell). This script can be found at /path/to/folder/SPEX-3.07.00-Linux. Please check whether you have the writing access. If not, use chown and/or chmod command to gain access.

```
user$ # for Linux user
user$ cd /path/to/folder/SPEX-3.07.00-Linux
user$ ls
NEWS bin data lib pgplot python spexdist.csh spexdist.sh
```

The following command can help you identify which shell you are using:

```
user$ which $SHELL
```

The screen output will be /bin/bash for bash shell and /bin/csh for C-type shell.
Note: Lines starting with the # symbol are comments.
The following command can help you to change shell:

```
user$ # Switch to bash shell
user$ chsh -s /bin/bash
user$ # Switch to C-type shell
user$ chsh -s /bin/csh
```

Open the shell script (spexdist.sh or spexdist.csh) with your favorite text editor (e.g., vim, nano, Emacs). Modify the file to set the environment variable SPEX90 properly:

```
user$ # for bash shell
user$ export SPEX90= /path/to/folder/SPEX-3.07.00-Linux/
user$ # for C-type shell
user$ setenv SPEX90 /path/to/folder/SPEX-3.07.00-Linux/
```

An exemplary spexdist.sh file where a Linux user has already set the SPEX90 environment correctly:

```
#!/bin/bash
export SPEX90=/path/to/folder/SPEX-3.07.00-Linux
export PGPLOT_DIR=$SPEX90/pgplot
export PATH=$PATH:$SPEX90/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SPEX90/lib
export PYTHONPATH=$PYTHONPATH:$SPEX90/python
# Readline functionality
if [ ! -d "$HOME/.spex_d" ]
then
    mkdir $HOME/.spex_d
    echo "-readline" > $HOME/.spex_d/spexrc
fi
ulimit -s 65532
```

An exemplary spexdist.csh file where a Linux user has already set the SPEX90 environment correctly:

```
#!/bin/csh
setenv PGPLOT_DIR $SPEX90/pgplot
setenv PATH $PATH:$SPEX90/bin
setenv LD_LIBRARY_PATH
$LD_LIBRARY_PATH:$SPEX90/lib
setenv PYTHONPATH $PYTHONPATH:$SPEX90/python
# Readline functionality
if ( ! -d $HOME/.spex_d ) then
    mkdir $HOME/.spex_d
    echo "-readline" > $HOME/.spex_d/spexrc
endif
limit stacksize 65532
```

For M1 chip users, it might be necessary to change the value for ulimit (the last line in the spexdist.sh or spexdist.csh file). The default value is 65532, which will run into error when sourcing the spexdist.sh or spexdist.csh file (see this ticket on Github for details).

The following command can help you to the proper stack size to use:

```
user$ ulimit -a
core file size (blocks, -c) 0
data seg size (kbytes, -d) unlimited
file size (blocks, -f) unlimited
max locked memory (kbytes, -l) unlimited
max memory size (kbytes, -m) unlimited
open files (-n) 256
pipe size (512 bytes, -p) 1
stack size (kbytes, -s) 8176
cpu time (seconds, -t) unlimited
max user processes (-u) 2666
virtual memory (kbytes, -v) unlimited
```

Modify the stack size in spexdist.sh or spexdist.csh to the value obtained above with your favorite text editor (e.g., vim, nano, Emacs). Please check whether you have the writing access. If not, use chown and/or chmod command to gain access.

```
#!/bin/bash
export SPEX90=/path/to/folder/SPEX-3.07.00-Linux
export PGPLOT_DIR=$SPEX90/pgplot
export PATH=$PATH:$SPEX90/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SPEX90/lib
export PYTHONPATH=$PYTHONPATH:$SPEX90/python
# Readline functionality
if [ ! -d "$HOME/.spex_d" ]
then
    mkdir $HOME/.spex_d
    echo "-readline" > $HOME/.spex_d/spexrc
fi
ulimit -s 8176
```

#### 2.1.5 Acitative the SPEX environment and run SPEX

After successful installation of SPEX (Linux, MacOS admin or non-admin install) and proper setting of the ${SPEX90} environment variable, one needs to acitative this environment before running SPEX.

The best practice is to add the following line to ~/.bashrc (for bash shell) or ~/.cshrc (for C-type shell) using your favorite text editor (e.g., vim, nano, Emacs).

```
# for bash shell
alias spex307="source /path/to/folder/SPEX-3.07.00-Linux/spexdist.sh"
# for C-type shell
alias spex307 "source /path/to/folder/SPEX-3.07.00-Darwin/spexdist.csh"
```

Then source the script as following:

```
# for bash shell
user$ source ~/.bashrc
# for C-type shell
user$ source ~/.cshrc
```

Note: If you open another terminal (or screen) session, you need to source the ~/.bashrc or ~/.cshrc script again.

Note: Please navigate to a path where you have the write access to run SPEX. Then you can do the following to start SPEX from the terminal:

```
user$ # Activate the SPEX environment
user$ spex307
user$ # Run SPEX
user$ spex
```

You should see an exemplary welcome message below:

```
Welcome user to SPEX version 3.07.00
NEW in this version of SPEX:
10-08-2021 Added tpho model for time-dependent photo-ionisation modeling
18-11-2021 Added spherically symmetric cluster model
13-05-2022 Included resonant scattering in cluster model
13-05-2022 Added aerror command to calculate atomic data errors
24-06-2022 Double precision now also for data handling and plotting
25-07-2022 Final release candidate
Currently using SPEXACT version 2.07.00. Type help var calc for details.

SPEX>
```

Note: If MacOS complains about not being able to verify SPEX, please select "Cancel" (not "Move to bin"). Open your "Finder" and navigate to the path where SPEX is installed. Right-click /path/to/folder/SPEX-3.07.00-Darwin/bin/spex: "Open with" → "Terminal". Close the terminal window after once the line "[Process completed]" appears. Now you should be able to run SPEX in your terminal smoothly.

Note: Similar issues might occur for plotting, in that case, open your "Finder" and navigate to the path where SPEX is installed. Right-click /path/to/folder/SPEX-3.07.00-Darwin/pgplot/pgxwin_server: "Open with" → "Terminal". Close the terminal window after once the line "[Process completed]" appears. Now you should be able to use the plotting functions in SPEX smoothly.

# 3 Visualization of the instrument effective area

## 3.1 Overview

In this thread, we use SPEX to visualize the effective area (or photon collecting area) of a given instrument.

## 3.2 Getting started

Install the SPEX software package following this thread. We use Chandra/ACIS-I as an example here, but the method applies to all other instruments.

## 3.3 Running SPEX

In your terminal session, navigate to the main working directory, which should contain the response and spectral data files. Start SPEX from the terminal:

```
user$ cd /path/to/folder
user$ ls
user$ acisi.res acisi.spo
user$ spex307
user$ spex
Welcome user to SPEX version 3.07.00
NEW in this version of SPEX:
10-08-2021 Added tpho model for time-dependent photo-ionisation modeling
18-11-2021 Added spherically symmetric cluster model
13-05-2022 Included resonant scattering in cluster model
13-05-2022 Added aerror command to calculate atomic data errors
24-06-2022 Double precision now also for data handling and plotting
25-07-2022 Final release candidate
Currently using SPEXACT version 2.07.00. Type help var calc for details.

SPEX>
```

## 3.4 Load data

```
SPEX> data acisi acisi
```

## 3.5 Plot the effective area

Comment lines start with "#", which can be safely deleted. Note that the plot settings can be readjusted. In this case, it is essential to type the "plot" command again to see the updated figure.

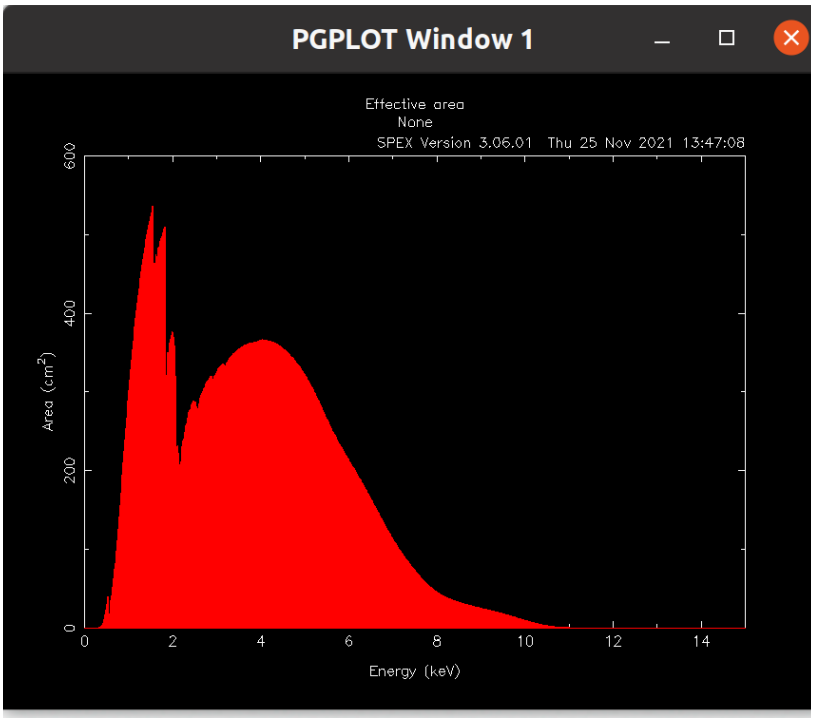


Figure 1: Effective area of Chandra/ACIS-I.

```
SPEX> # Setting the plotting device to X-window
SPEX> plot dev xw
SPEX> # Setting the plotting type to area
SPEX> plot type area
SPEX> # Setting the plotting X-axis unit to keV
SPEX> plot ux keV
SPEX> # Setting the plotting X-axis range to 0:15 keV
SPEX> plot rx 0:15
SPEX> # Setting the plotting Y-axis unit to cm^2
SPEX> plot uy cm
SPEX> # Setting the plotting Y-axis range to 0:600 cm^2
SPEX> plot ry 0:600
SPEX> # All the above lines are settings
SPEX> # The command "plot" below will create/update the plot
SPEX> plot
```

The following effective area plot (Figure 1) shall appear on your screen.

### 3.6 Save the effective area curve (optional)

It is possible to save the effective area curve in the ASCII format. In fact, I use python to plot and compare effective area curves of different instruments as shown in the lecture slides.

```
SPEX> # Save the effective area curve in the ASCII format to aeff_acisi.qdp.
SPEX> The extension .qdp will be added automatically by SPEX.
SPEX> The last keyword "overwrite" is essential if there is an existing file.
SPEX> plot adum aeff_acisi overwrite
SPEX> # The above line is again a plot setting (where the save the data)
SPEX> # The commend "plot" will write data to the file
SPEX> plot
SPEX> # Now we can quit SPEX
SPEX> quit
```

#### 3.6.1 Column-by column description of the ASCII file

Note that the description below corresponds to the units setting above. If the plot setting uses Angstrom or $m^2$, the content of the data (and description) shall also change.

- Col. #0: X-axis value (energy in keV)
- Col. #1 & #2: Energy bin half width (positive and negative both in keV)
- Col. #3: Y-axis value (effective area in $cm^2$)

```
user $ head aeff_acisi.qdp
READ Terr 1
.... (multiple lines above)
1.4931178E+00 3.1177998E-03 -3.1177998E-03 5.2280481E+02
1.4981178E+00 1.8821955E-03 -1.8821955E-03 5.2280481E+02
1.5050000E+00 4.9999952E-03 -4.9999952E-03 5.2544513E+02
1.5117806E+00 1.7806292E-03 -1.7806292E-03 5.2775830E+02
1.5167806E+00 3.2193661E-03 -3.2193661E-03 5.2775830E+02
... ... (multiple lines below)
```

### 3.7 Summary

In this thread, we showed how to plot the effective area of the Chandra/ACIS-I instrument using SPEX. Effective area curves of other instruments can be obtained in a similar way.

# 4 Examining the number of counts per bin

## 4.1 Overview

In this thread, we use SPEX to examine the number of counts per bin.

## 4.2 Getting started

We use a simulated XRISM/Resolve spectrum (with a spectral resolution $\Delta E = 7$ eV) here, but the method applies to all other spectra. The spectral data files (plik.spo and xr7ev.res) are available in the SPEX/counts_per_bin folder of the workshop cloud drive (pwd: hixuv).

## 4.3 Running SPEX

In your terminal session, navigate to the main working directory, which should contain the response and spectral data files. Start SPEX from the terminal:

```
user$ cd /path/to/folder
user$ ls
user$ xr7ev.res plik.spo
user$ spex307
user$ spex
Welcome user to SPEX version 3.07.00
NEW in this version of SPEX:
22-07-2020 Bugfix: neij gives line emission while abundance is zero.
18-08-2020 Changed reference density from electron density to hydrogen density.
04-09-2020 Added pyroxene back to the amol model.
23-10-2020 Update of H I collision strengths (IMPORTANT)
24-11-2020 Added ascdump and par_show functions for the Python interface.
Currently using SPEXACT version 2.07.00. Type help var calc for details.

SPEX> # Read spectral data files (.res file comes first without the extension)
SPEX> data xr7ev plik
SPEX> # Obtain basic info of this data set (e.g., exposure time)
SPEX> data show
```

## 4.4 Plot the data spectrum in different units

SPEX users can specify X- and Y-axis units for the data spectrum, please see a full list of available units here. Caution that available plotting units might vary for different plotting types, please see a full list of available plotting types here.

```
SPEX> # Set the plotting device to the running X Window System
SPEX> plot dev xw
SPEX> # Set the plotting type to data spectrum
SPEX> plot type data
SPEX> plot
```

The following plot of the data spectrum (Figure 2) shall appear on your screen, which is certainly not pretty. This is partly due to the fact that we have not rebinned the spectrum properly. When dealing with low-resolution X-ray spectra, it is common to rebin the spectrum to have $\gtrsim 20$ counts per bin so that $\chi^2$ statistics can be used. However, $\chi^2$ statistics won't work well with most high-resolution X-ray spectra. Readers are referred to Kaastra (2017) on why we should use $C$-statistics instead.

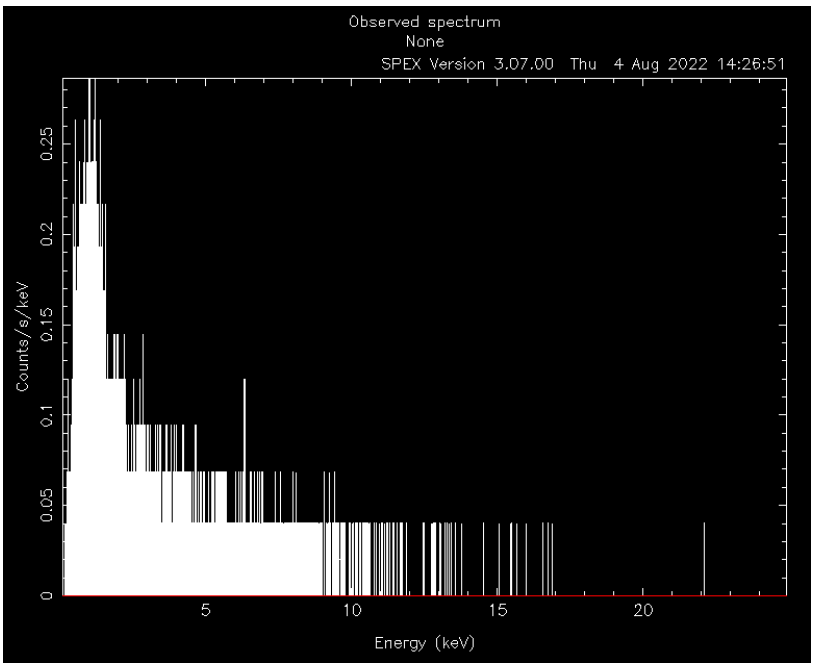


Figure 2: Data spectrum with default X- and Y-axis units.

How do we probably rebin the spectrum? Before answering this question, let's first examine the number of counts per bin in our exemplary spectrum. This can be achieved by setting the Y-axis units to "cou" (short for "Counts"). Furthermore, we zoom into the energy range between 6.2 and 6.5 keV to have a better view of potential line features.

```
SPEX> # Set the Y-axis units to "Counts"
SPEX> plot uy co
SPEX> # Zoom into the energy range of 6.2 − 6.5 keV
SPEX> plot rx 6.2:6.5
SPEX> plot
```

The following plot (Figure 3) shall appear on your screen. Apparently, for the majority of spectral bins between 6.2 and 6.5 keV, the number of counts per bin is either 0 or 1. One can further zoom into an energy range with a few data points to find out the default bin size (0.5 eV), which is much smaller than the spectral resolution $\Delta E = 7$ eV of the XRISM/Resolve instrument.

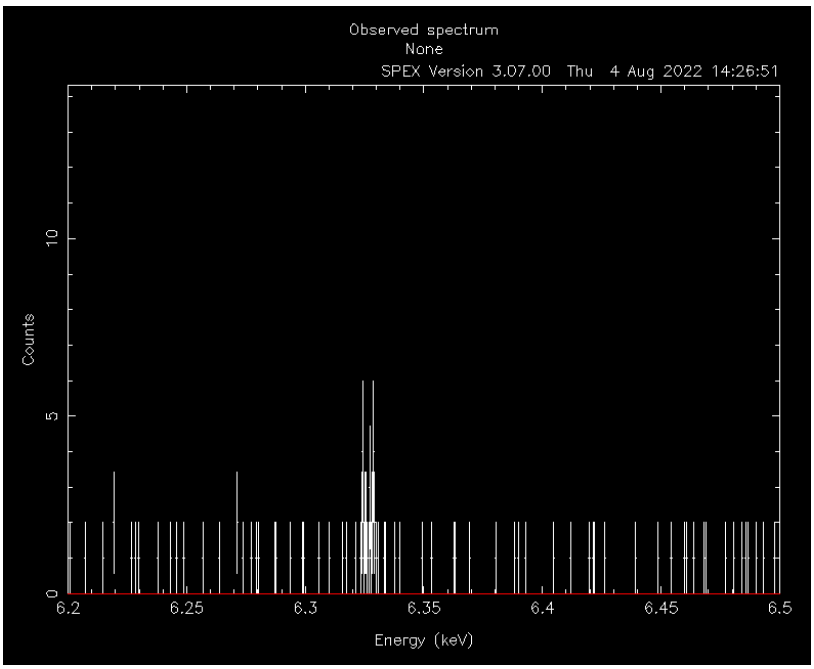


Figure 3: Number of counts per bin in the energy range of 6.2 and 6.5 keV.

There is a cluster of bins with up to 4 counts per bin around 6.325 keV or so in Figure 3. Is this a line or a blend of several lines? Here comes a "magical" function called "optimal binning" in SPEX (i.e., the obin command). Readers are referred to Kaastra & Bleeker (2016) to unveil the magic!

```
SPEX> # Rebin the entire spectrum
SPEX> obin inst 1 reg 1 0:1.E3 unit kev
SPEX> plot
```

Now the following plot (Figure 3) shall appear on your screen. A narrow line is now clearly visible! What is the bin size now?

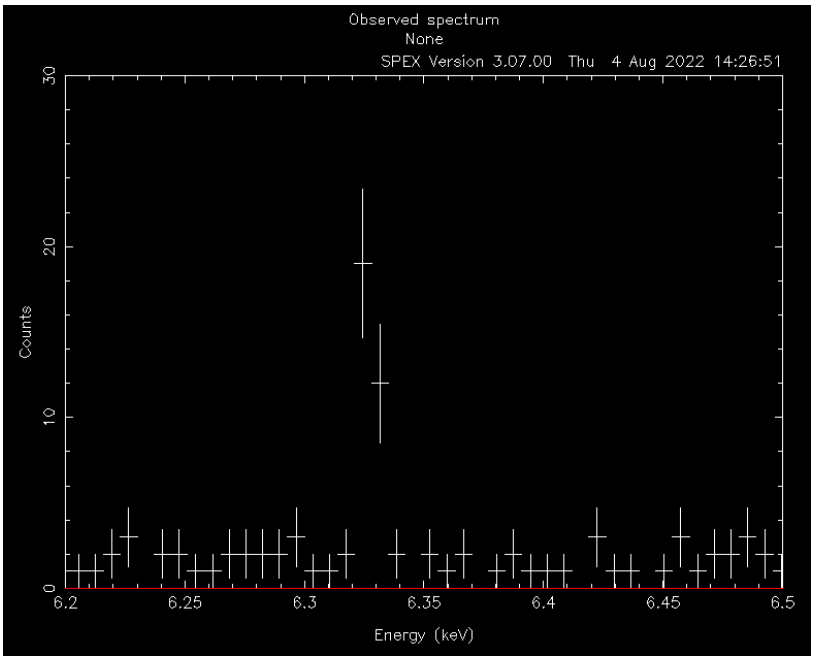


Figure 4: Number of counts per bin in the energy range of 6.2 and 6.5 keV.

## References

Mewe, R. 1972, Solar Physics, 22, 459. doi:10.1007/BF00148711

Kaastra, J. S. & Bleeker, J. A. M. 2016, A&A, 587, A151. doi:10.1051/0004-6361/201527395

Kaastra, J. S. 2017, A&A, 605, A51. doi:10.1051/0004-6361/201629319