# Use of XSPEC for XRISM

Keith Arnaud

High Energy Astrophysics Science Archive Research Center

University of Maryland College Park,
NASA's Goddard Space Flight Center,
and CRESST

Craig Gordon

Bryan Irby

Kristin Rutkowski

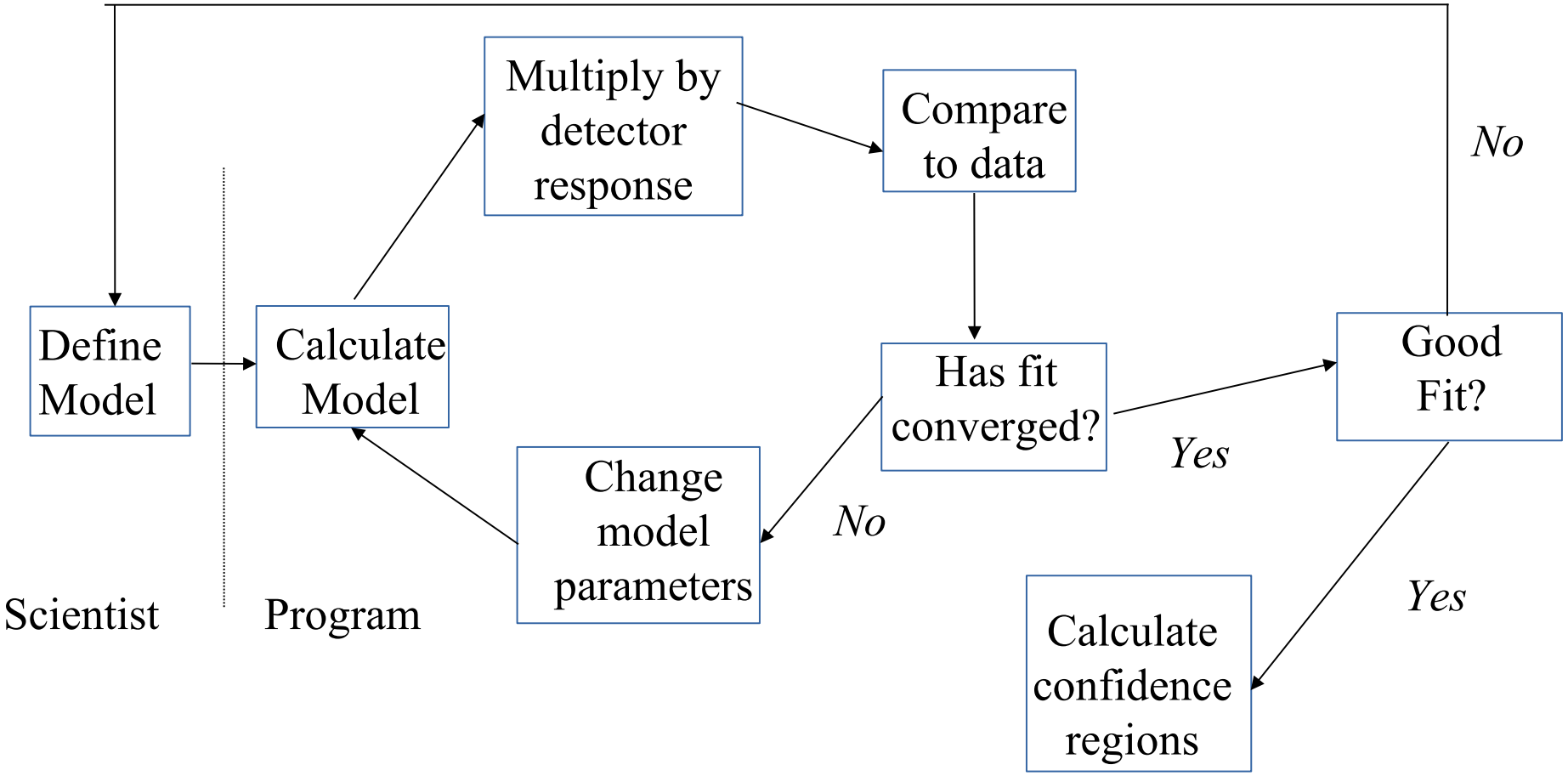
General comments

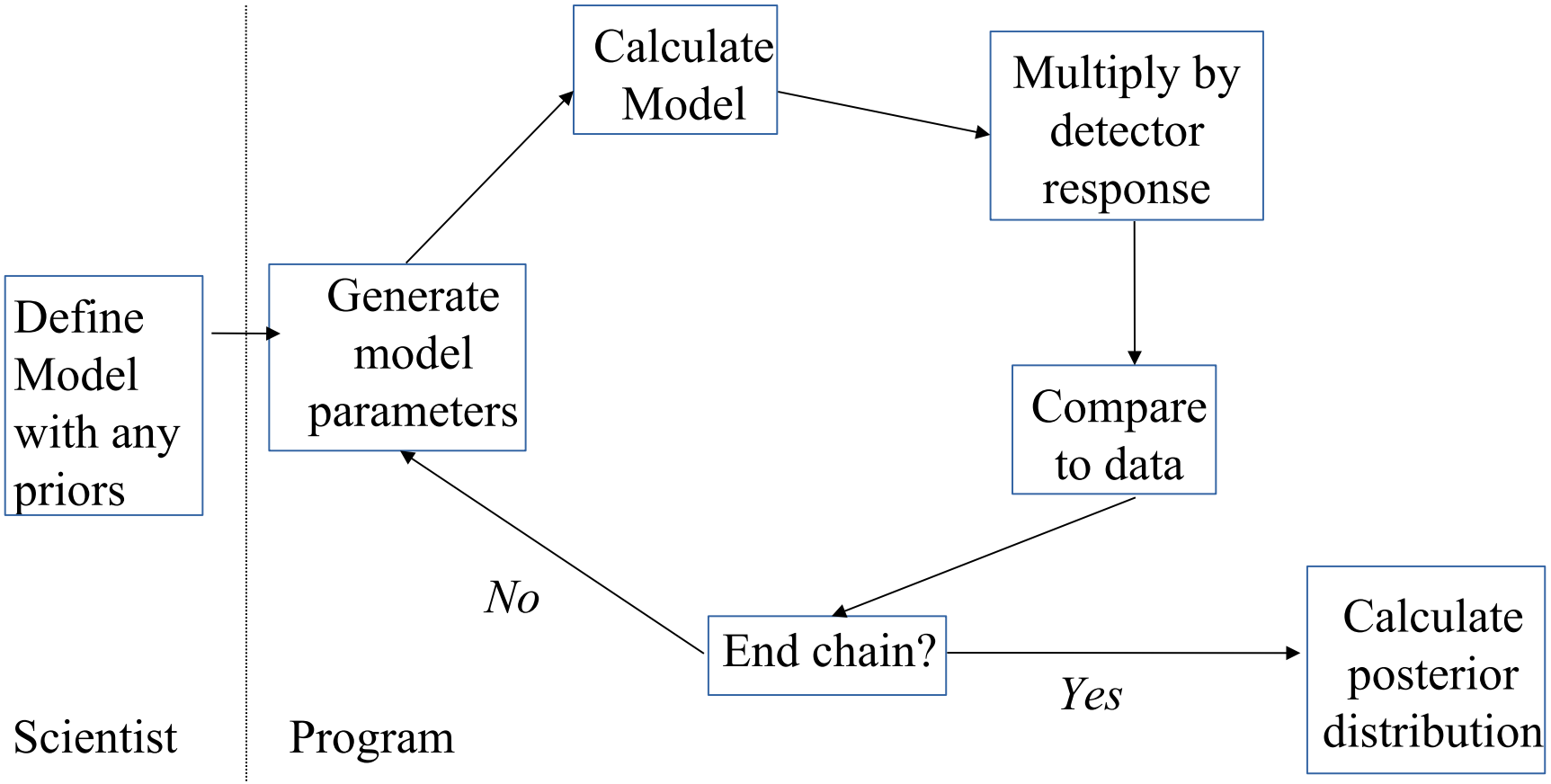
Coming attractions

Handling the Resolve response matrix

Parallelization (tomorrow)

Simultaneous fitting of Resolve and Xtend (tomorrow)


Define Model
Calculate Model
Multiply by detector response
Compare to data
Has fit converged?
Good Fit?
No
Yes
Change model parameters
No
Yes
Calculate confidence regions
Scientist
Program


Calculate Model
Multiply by detector response
Define Model with any priors
Generate model parameters
Compare to data
No
End chain?
Calculate posterior distribution
Yes
Scientist
Program

# Xspec

An X-Ray Spectral Fitting Package

**Downloads** | **Manual** (Updated 7/20/23) | **Issues/Patches** (Updated 10/10/23) | **Additional Models** | **Xspec In Python** | **Additional Python**

---

***Version 12.13.1 Now Available***

Helpful information available includes:

- **Frequently Asked Questions**
- **Download and installation instructions**
- **Xspec release notes**
- **Xspec wiki**
- **Xspec internal functions guide**
- **Xspec facebook group** [external]
- **External Resources**
- **Xspec v11**
- **Details of Xspec redesign**
  A site section displaying the design details and implementation of Xspec version 12.

# Known Issues



## New issues from Xspec 12.13.1

For **Mac OS** users attempting to build the **relxill local model package** with the **clang++ compiler**, you will first need to download and run **this shell script** from your `heasoft-[ver]/Xspec/src` directory. Note that this is the same directory from which you would normally install Xspec patch files.

**Frequently Asked Questions**

List of currently **outstanding issues.**

## Patches

Latest update **12.13.1e** (Oct 10, 2023)

Requires patch installer **v4.17**

For details about the patch installer, including options available with the latest v4.17, see the **installer notes**

To apply patches, you'll need to have downloaded and already built the **source code distribution** of XSPEC. Your HEADAS environment variables must also be initialized, such as with the headas-init script as you would if you were running XSPEC.

To update your XSPEC installation with **ALL** listed patches:

1. If you don't already have the **patch installer** tcl script (v4.17), download and place it in the `heasoft-[ver]/Xspec/src` directory in the source code distribution.
2. Download the **latest patchfile** into the same directory. There is **no need to unzip/untar** the patchfile since the installer will do that.
3. At the command line, launch the script by typing `'tclsh patch_install_[ver].tcl'`, and **that's all**.

The script will detect which patches you've installed previously, copy the new code files to the necessary directories, determine whether complete or only partial rebuilds are necessary, and finally perform the rebuilds. Numbers listed in red indicate that the patch modifies the interface to one or more of XSPEC's libraries. These patches will generally cause longer rebuilds. If you have a local model library, it is recommended (though not always necessary) that you rebuild it with **initpackage** after applying red patches.

---

**12.13.1a** Table models with both **redshift** and **escale** set are not handled correctly. Our thanks to Doug Burke for reporting this. Report added on Aug 23, 2023.

This patch modifies Xspec/src/XSModel/Model/Component/OGIPTable/ **OGIPTable.cxx**

**12.13.1b** When an **MCMC** chain file contains **response parameters**, Xspec is falsely reporting mismatches between the chain parameters and the current variable fit parameters. Therefore it doesn't allow the chain to be used for uncertainty estimates in place of fit covariance values. Our thanks to Kenji Hamaguchi for reporting this. Report added on Sep 08, 2023.

This patch modifies Xspec/src/XSFit/Fit/ **Fit.cxx**

**12.13.1c** In PyXspec's **AllModels.addPyMod** method, usage of a deprecated Python function is preventing execution with the latest **Python v3.11**. Our thanks to Fabio Muleri for reporting this. Report added on Sep 11, 2023.

This patch modifies Xspec/src/XSUser/Python/xspec/ **model.py**

**12.13.1d** There is an error in the calculation of the **pgstat** statistic. Our thanks to Yanqui Zhang for reporting this. Report added on Sep 11, 2023.

This patch modifies Xspec/src/XSStat/ **Cstat.h**

**12.13.1e** The **vvapec, bvvapec, vvtapec,** and **bvvtapec** models all deal with the **H abundance parameter** incorrectly. If it is not zero, it is reset to one. So, e.g. setting the H abundance to 0.5 in vvapec actually produces a spectrum with the H abundances as 1.0. If the H abundance is set to 1.0 (the default and usual setting) the results are correct. We thank Denis Leahy and Adam Foster for pointing this out. Report added on Oct 10, 2023.

This patch modifies Xspec/src/XSFunctions/ **bvvapec.cxx**, **bvvtapec.cxx**, **vvapec.cxx** and **vvtapec.cxx**

### *Local models*

- **A variable-Comptonization spectral-timing model for QPOs: vKompth**
  Model added on 8/11/23.

- **Relativistic reflection model from Dauser and Garcia which combines relconv and xillver: relxill.**
  For **Mac OS** users attempting to build the **relxill local model package** (using Xspec's 'initpackage' command) with the **clang++ compiler**, you will first need to download and run **this shell script** from your `heasoft-[ver]/Xspec/src` directory. Note that this is the same directory from which you would normally install Xspec patch files.

  Model added 1/23/14. Updated 7/15/14. Updated (v0.3a) 10/15/15. Updated (v0.4a) 1/18/15. Updated (v1.1.0) 8/1/18. Updated (v1.2.0) 8/9/18. Updated (v1.4.0) 10/8/20. Updated (v2.0) 7/5/22. Updated (v2.1) 8/1/22. Updated (v2.2) 9/6/22.

- **Spectral models for Chandra ACIS particle-induced background from Suzuki et al.: mkacispback**
  Model added on 5/24/22.

- **A library of synthetic spectra for fitting tidal disruption events from Wen et al.: slimdisk**
  Model added on 4/7/22.

- **A python script to extract spectra from SPEX in a form that can be used by xspec**
  Model added 7/10/06. Updated 10/24/08. Updated 8/1/18. Updated 9/22/20.

- **Collisional plasmas with non-Maxwellian distributions of electrons from Cui et al.: kappa.**
  Model added on 8/6/20.

- **AtomDB electron-electron bremsstrahlung model: eebrems.**
  Model added on 8/6/20.

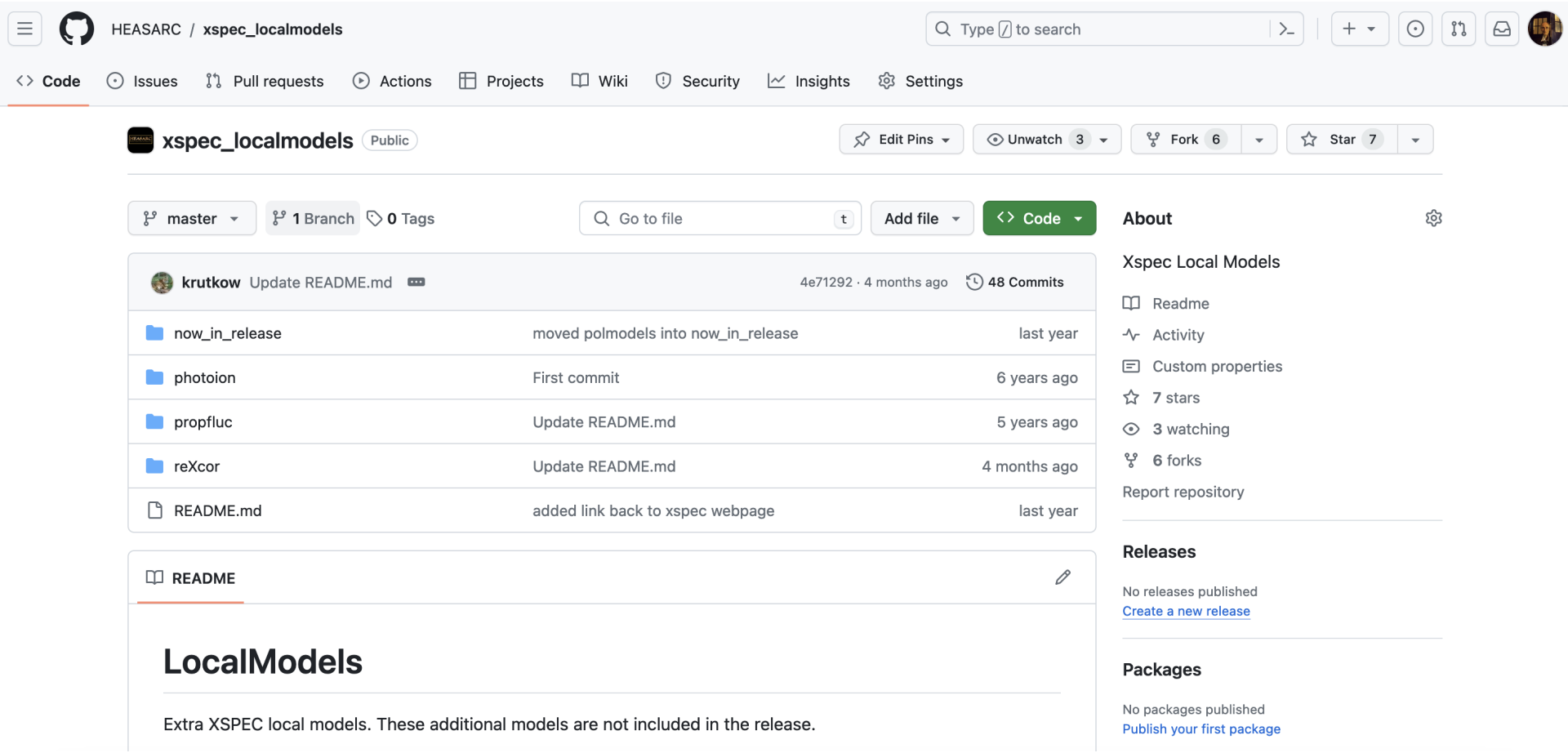
- **Relativistic reflection model from Niedzwiecki et. al.: reflkerr.**
  Model added 4/18/19.

- **Model for high-resolution X-ray absorption from material in ionization equilibrium from Gatuzz and Churazov:ioneq.**
  Model added 11/9/17.

- **Propagating fluctuations in the mass accretion rate of a precessing flow as a power spectral model for black hole binaries: propfluc**
  This is not a model for energy spectra but for power spectrum from timing data. Model added 8/2/11. New much faster version with more features 9/16/14. Updated 06/12/2019.

- **Line profiles from hot star winds; absorption of X-rays from hot star winds: windprof, hwind, hewind, windtabs.**
  Model added 1/30/13. Updated on 8/4/15. Installation instructions updated 8/28/15.

- **Photoionization models from Ali Kinkhabwala produced as part of a PhD thesis at Columbia Astrophysics Laboratory : photoion, phai, phxi, miabs, siabs, xiabs, neutral, vneutral, addext, mulext, tauext.**
  Model added 4/7/06. Updated 11/7/06 to include some input data files that were accidentally left out of the tar file. Updated 03/23/2018.




Models can use gsl but please avoid routines from Numerical Recipes.

# Quick Tutorial

This assumes the user already has a basic familiarity with both XSPEC and Python. Everything in PyXspec is accessible by importing the package **xspec** into your Python script.

PyXspec can be utilized in a Python script or from the command line of the plain interactive Python interpreter. PyXspec does not implement its own command handler, so it is NOT intended to be run as the Python equivalent of a traditional interactive XSPEC session (which is really an *enhanced* interactive Tcl interpreter). In other words you launch an interactive PyXspec session with:

```
LINUX> python
>>> import xspec
>>>
```

rather than:

```
LINUX> xspec
XSPEC12>
```

Note that in all the tutorial examples the **xspec** package name qualifier is left off. You must either include the **xspec** qualifier:

```
s = xspec.Spectrum("file1.pha")
```

or use a variation of the Python `import` or `from...import` commands:

```
from xspec import *
s = Spectrum("file1.pha")
```

## Jumping In: The *Really* Quick Tutorial

A simple Xspec load-fit-plot Python script may look something like this:

```
#!/usr/bin/python
from xspec import *

Spectrum("file1.pha")
Model("phabs*pow")
Fit.perform()
Plot.device = "/xs"
Plot("data")
```




## Python packages for PyXspec

The following packages either build on PyXspec or have been proved to be very useful with PyXspec.

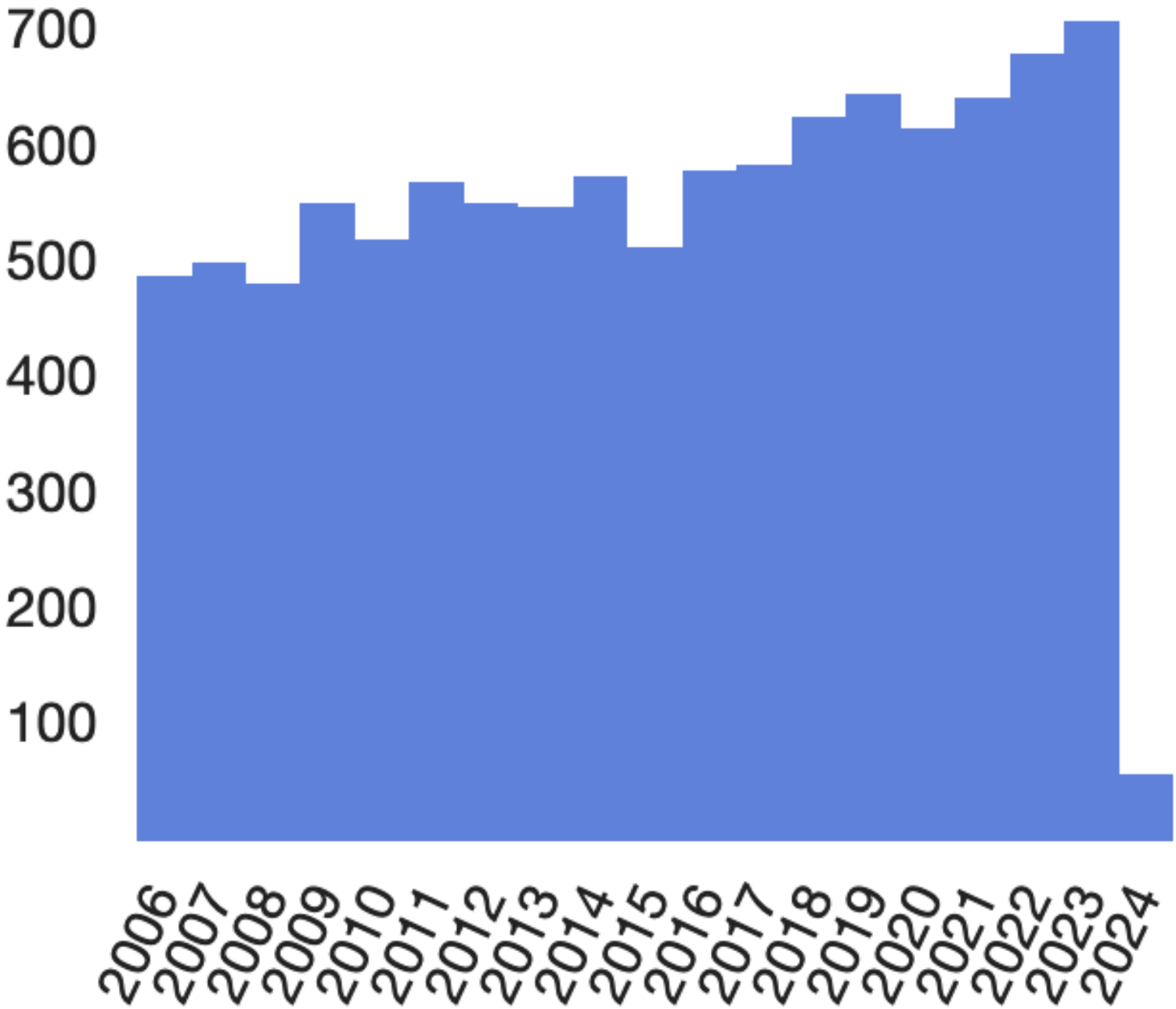
If you have packages that you would like to advertise here, please contact Keith Arnaud (Keith.A.Arnaud-at-nasa.gov).

- **Bayesian X-ray Analysis** by Johannes Buchner running on top of PyXspec or Sherpa. See **Buchner et al. 2014.**
- **A simple example of using PyXspec in Jupyterlab from Ivan Valtchanov.**
- **Example PyXspec Jupyter notebooks.**
- **An example using virtual observatory methods to acquire a spectrum and PyXspec to analyze it.**
- **Jeremy Sanders' package to run xspec with emcee. Supports running multiple copies of xspec simultaneously on different machines.**
- **A python tool from Federico Garcia to make interactive corner plots based on xspec mcmc chains saved to FITS files.**


Xspec Home Page

HEASARC Home | Observatories | Archive | Calibration | Software | Tools | Students/Teachers/Public

Refereed publications with XSPEC from ADS



Top authors in 2023.

| | |
|---|---|
| Garcia, J | 41 |
| Bianchi, S | 39 |
| Tombesi, F | 39 |
| Enoto, T | 38 |
| Doroshenko, V | 36 |
| Vink, J | 35 |
| Dovciak, M | 34 |
| Petrucci, P | 34 |
| Matt, G | 33 |
| De Rosa, A | 32 |

# HEASP and HEASPTOOLS

heasp is a C++ library of routines to manipulate spectra, responses, table models.

A python module called heasp provides an interface to these C++ routines.

heasptools are a set of tools built on top of heasp and replace many old tools in caltools and heasarc.

ftgrouppha is a replacement for grppha with many more grouping options including optimal binning.

```
from heasp import *
import xspec as xsp
import numpy as np

responseName = “input.rsp”
inputRSP = rmf(responseName)

# energyBins needs to be the standard internal xspec energy array
numEnergies = inputRSP.NumberEnergyBins() + 1
energies = np.empty((numEnergies))
energies[0] = inputRSP.getLowEnergyElement(0)
for j in range(1,numEnergies):
      energies[j] = inputRSP.getHighEnergyElement(j-1)

# set up parameters as a list
params = [ 1.7 ]
# and output flux as a list
fluxlist = []
xsp.callModelFunction(“powerlaw”,energies.tolist(), params, fluxlist)
flux = np.array(fluxlist)
phaValues = inputRSP.multiplyByModel(flux)
```

```
channel = np.arange(phaValues.size).astype(np.int32)
exposure = 10000.0
phaValues *= exposure

phaOut = pha()
phaOut.setFirstChannel(0)
phaOut.setPha(phaValues)
phaOut.setChannel(channel)
phaOut.setExposure(exposure)
phaOut.setDetChans(phaValues.size)
phaOut.setPoisserr(True)
phaOut.setDatatype("COUNT")
phaOut.setSpectrumType("TOTAL")
phaOut.setResponseFile(responseName)
phaOut.setTelescope(inputRSP.getTelescope())
phaOut.setInstrument(inputRSP.getInstrument())
phaOut.setFilter(inputRSP.getFilter())

status = phaOut.write("output.pha")
```

# In the next XSPEC release

Electron-electron bremsstrahlung added to apec models.

All models using apec have a “b” option (eg bvrnei) which adds a parameter for velocity broadening.

All models which have a switch parameter between mekal and apec default to apec.

Rationalized some names. For instance:

coolflow == mkcflow

vcoolflow == vmcflow

with bcoolflow and bvcoolflow added.

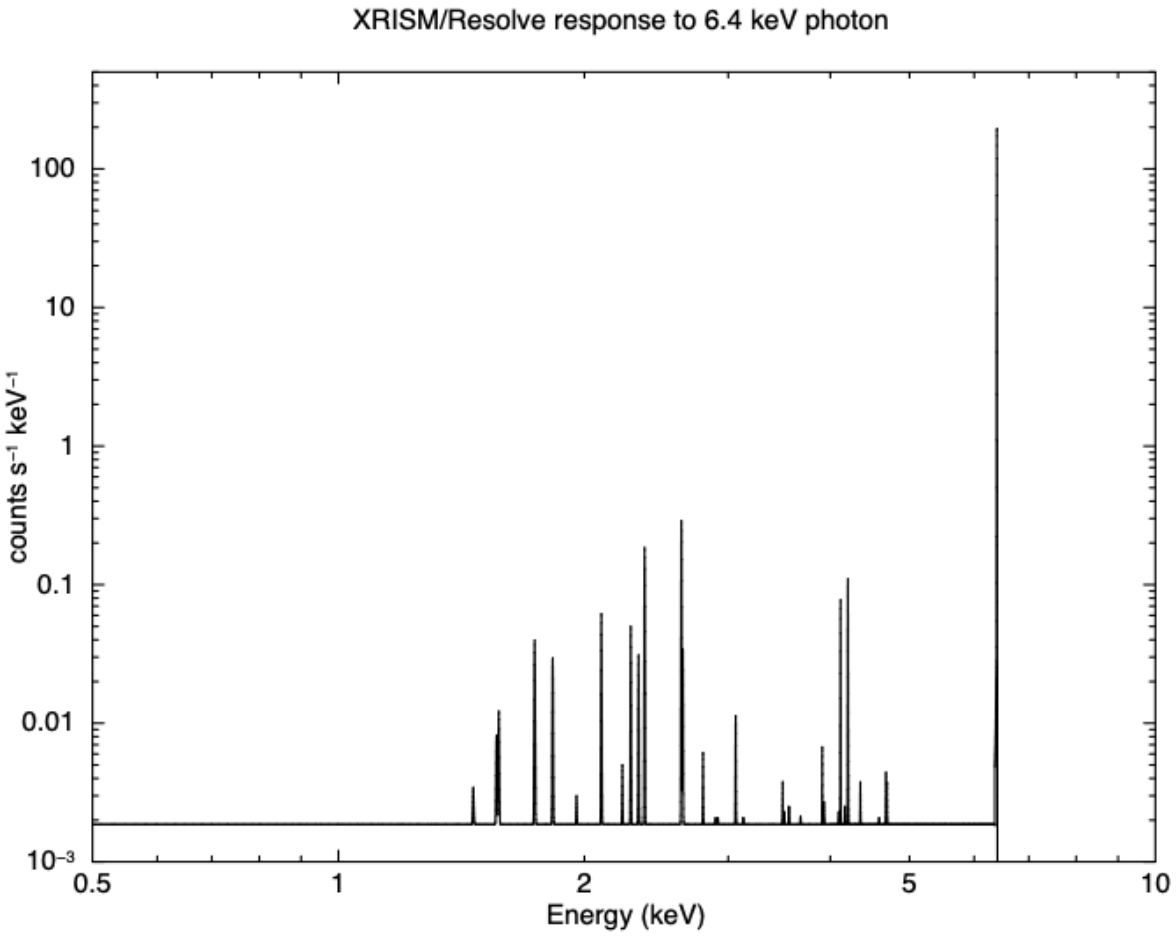
# XSPEC speed

XSPEC speed is dominated by two steps: calculating the model and multiplying the model by the response matrix.

Model calculation is usually $O(N_E)$ or $O(N_E \log N_E)$ where $N_E$ is the number of energy bins.

Response matrix calculation is $O(N_R)$ where $N_R$ is the number of non-zero elements in the matrix.

XRISM/Resolve has 60,000 energy bins and 60,000 channels. Since the full response matrix is triangular it is >7 Gb in size.



Calculate model on 60,000 energy bins

Bin up to 3,750 energy bins