SpectralFitting.jl – A Julia Package For Spectroscopy

Eric M. Schlegel[1,5], Fergus Baker[2], Paul Barrett[3], Andrew Young[2], Joe Prendergast[4]
[1]University of Texas at San Antonio, [2]University of Bristol, [3]George Washington University, [4]American University

What is Julia?

- a programming language[1]
- released in stable version (1.0) in 2018[2];
- built over the prior decade at MIT by a group dis-satisfied with the available languages;
- multi-paradigm: combines functional, imperative, and object-oriented approaches;
- some attributes: multiple dispatch, inherently parallel, high-level yet as fast as C, no 'two language' problem (implemented via a LLVM = Low-Level Virtual Machine);
- number abstraction: propagate uncertainties of all measurements and units seamlessly

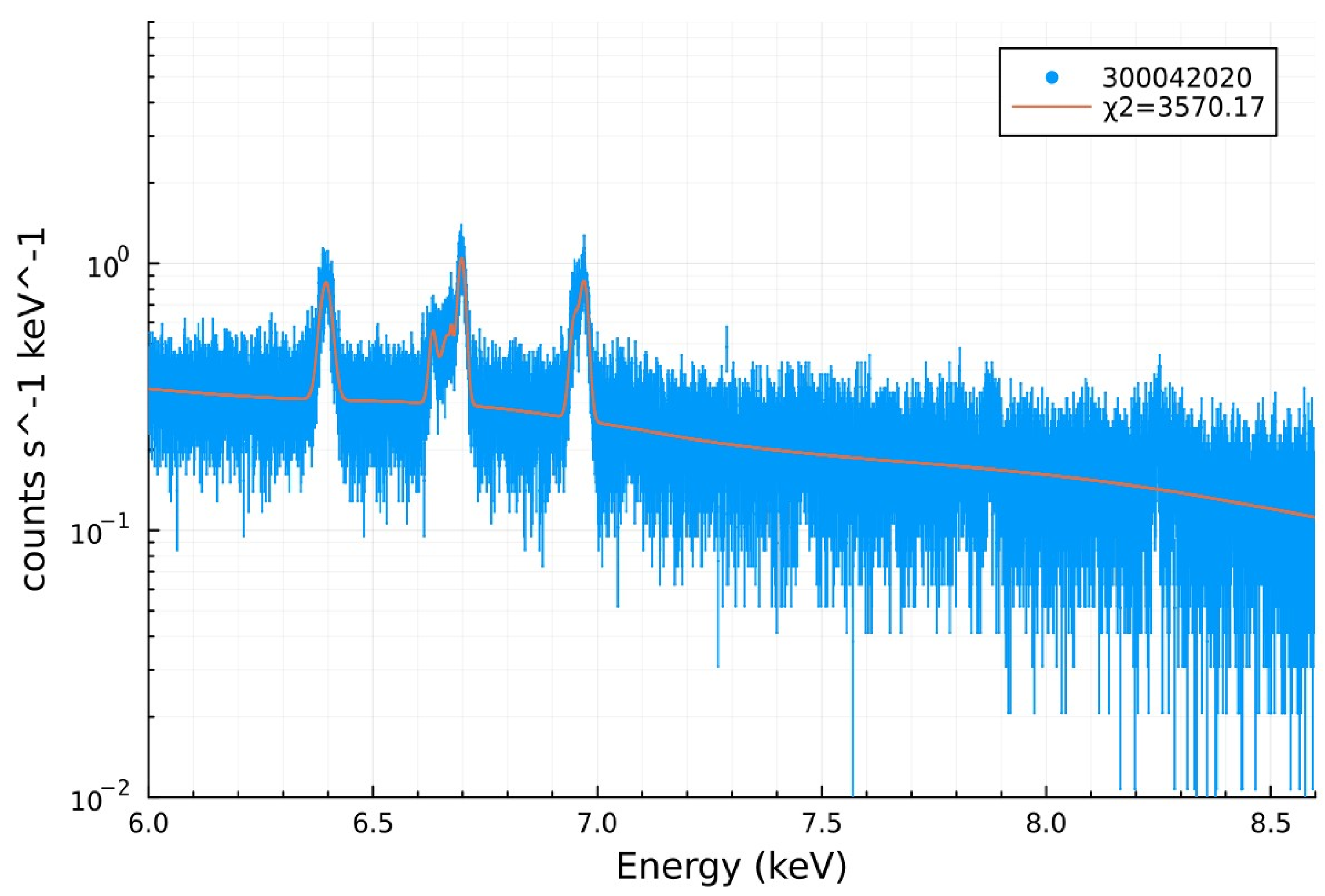
- current version: 1.11 (1.12 likely out spring 2025)
- purpose of poster: introduce Julia to astrophysical data analyses!

What is SpectralFitting.jl?

- a spectral fitting package[4] developed by F. Baker and A. Young at University of Bristol; first 'released' in 2023;
- ultimate intent: unify spectral fitting capabilities at *all* wavelengths + include radiative transfer and geometric models as options;
- currently outperforms XSPEC by ~3x on a single thread[3];
- multiple optimization routines via different packages: e.g., LsqFit.jl (least squares, Levenberg-Marquadt, etc.); optimization.jl (Nelder-Mead, BFGS, etc.); Turing.jl (MCMC, other Bayesian approaches).

- Current status: SpectralFitting.jl works with most X-ray spectra including *XRISM* data[5].
- Package does *not* work (as yet) with *Chandra* grating data *as packaged* by the CIAO software (i.e., the 'pha2' file is the roadblock)[5] – but a relatively 'easy' fix.
- Must also extend support for additional data formats; currently limited to HEASARC's OGIP format.
- Many models are wrapped XSPEC routines – these will be re-written using Julia in the 'near' future.

SpectralFitting.jl Fit to *XRISM* Resolve Spectrum



Fit Parameters

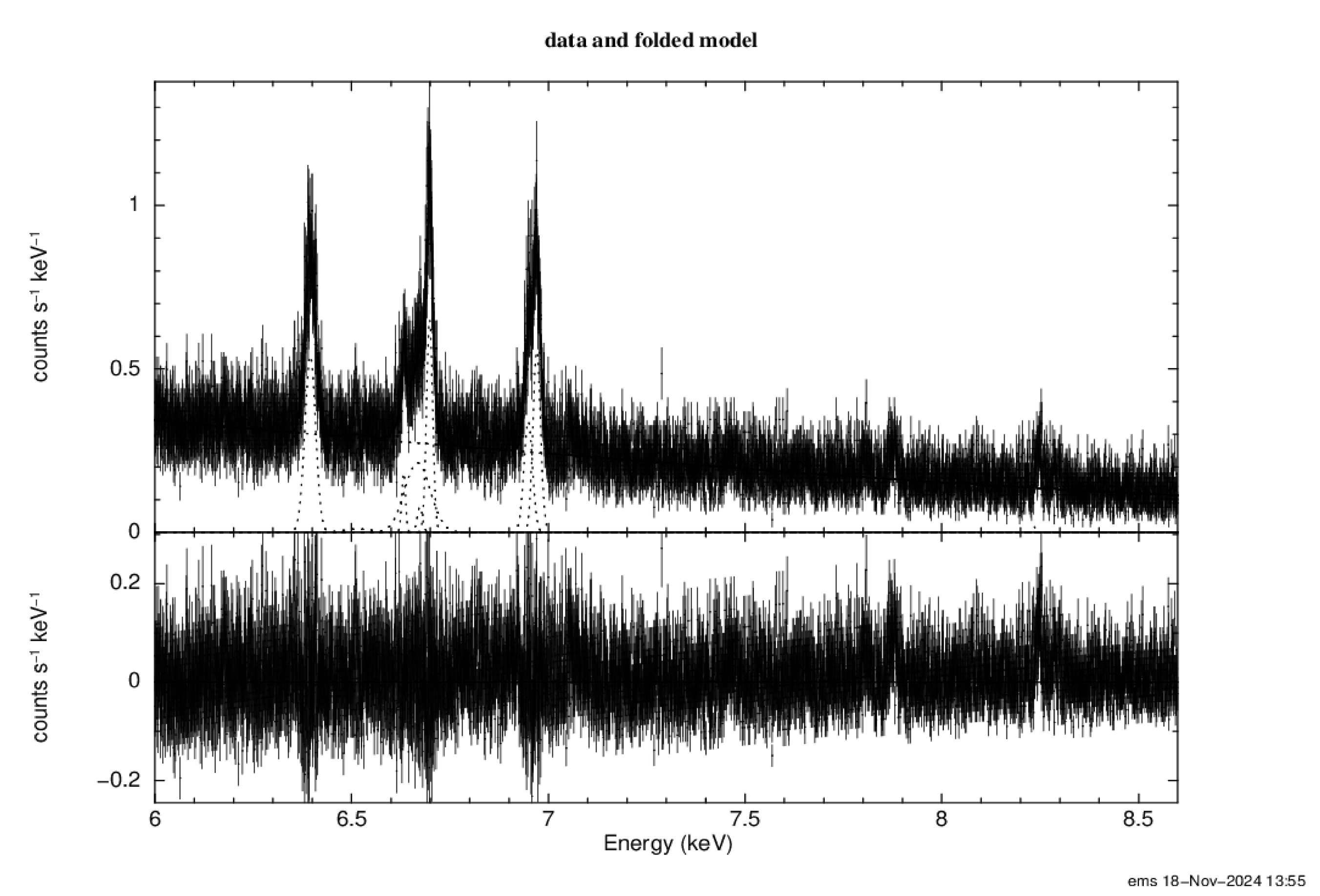
- Spectrum fit with power law continuum + nine Gaussians: SpectralFitting.jl

Line	Position (keV)		Line Flux	Line width (keV)
G1	6.3955		1.0206e-4	0.013238
G2	6.5101		1.3797e-6	0.000065
G3	6.6334	z	2.7139e-5	0.007577
G4	6.6661	y	4.8397e-5	0.013904
G5	6.6751	x	2.8204e-6	0.001693
G6	6.6981	w	9.0099e-5	0.008358
G7	6.9491	Lyα2	4.9580e-5	0.009448
G8	6.9711	Lyα1	4.5770e-5	0.008899
G9*	8.1046	Lyβ	2.3363e-4	0.51003

* see note 2 below at 'Experience Results'

G1 = Kα (Fe I to ~Fe XVII)
G3, G4, G5, G6 = Fe XXV;
G7,G8 = Fe XXVI

XSPEC Fit to *XRISM* Spectrum



Fit Parameters

- Spectrum fit with power law continuum + nine Gaussians: XSPEC

Line	Position (keV)		Line Flux	Line width (keV)
G1	6.3947		1.0405e-4	0.013483
G2	6.5162		4.9836e-6	0.034869
G3	6.6323	z	1.2395e-5	0.004296
G4	6.6673	y	9.8626e-5	0.032364
G5	6.6733	x	7.6998e-6	0.006544
G6	6.6982	w	7.2320e-5	0.007488
G7	6.9487	Lyα2	5.3937e-5	0.010703
G8	6.9706	Lyα1	7.7517e-5	0.009358
G9*	8.2350	Lyβ	7.0339e-7	0.000410

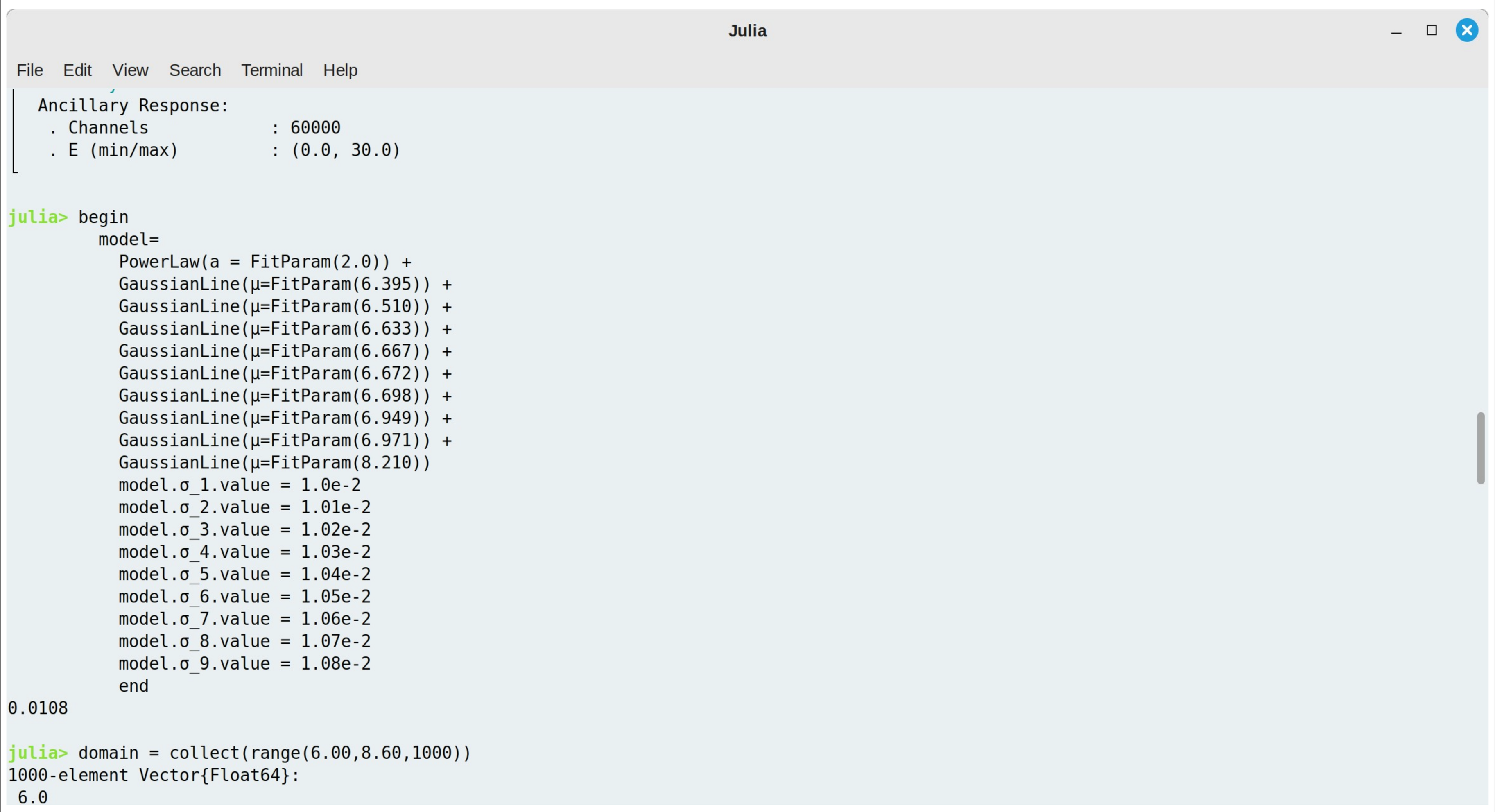
* = see note 2 below at 'Experience Results'

Experience Results

1. SpectralFitting.jl fit the spectrum in ~30-35 sec (timed by watch, not system clock); XSPEC, in about 2 minutes. While time is important, the one noteworthy item: both models started with the same initial guesses. SpectralFitting.jl was *not* further constrained while XSPEC was: line position and width were initially frozen, then the spectrum was fit, else the code went 'off the ranch'. The '2 minutes' did *not* count that initial fit but only the fit once constraints were removed. The cause of the speed difference? Very likely, because SpectralFitting.jl uses automatic differentiation[6].

2. G9: the differences here demonstrate that the Lyβ line likely does not exist. The XSPEC 'fit' occurs only because of the constraints on line center; the SpectralFitting fit shows a very broad and low height 'Gaussian'. The intent: test directly whether the line exists.

Model Definition – SpectralFitting.jl

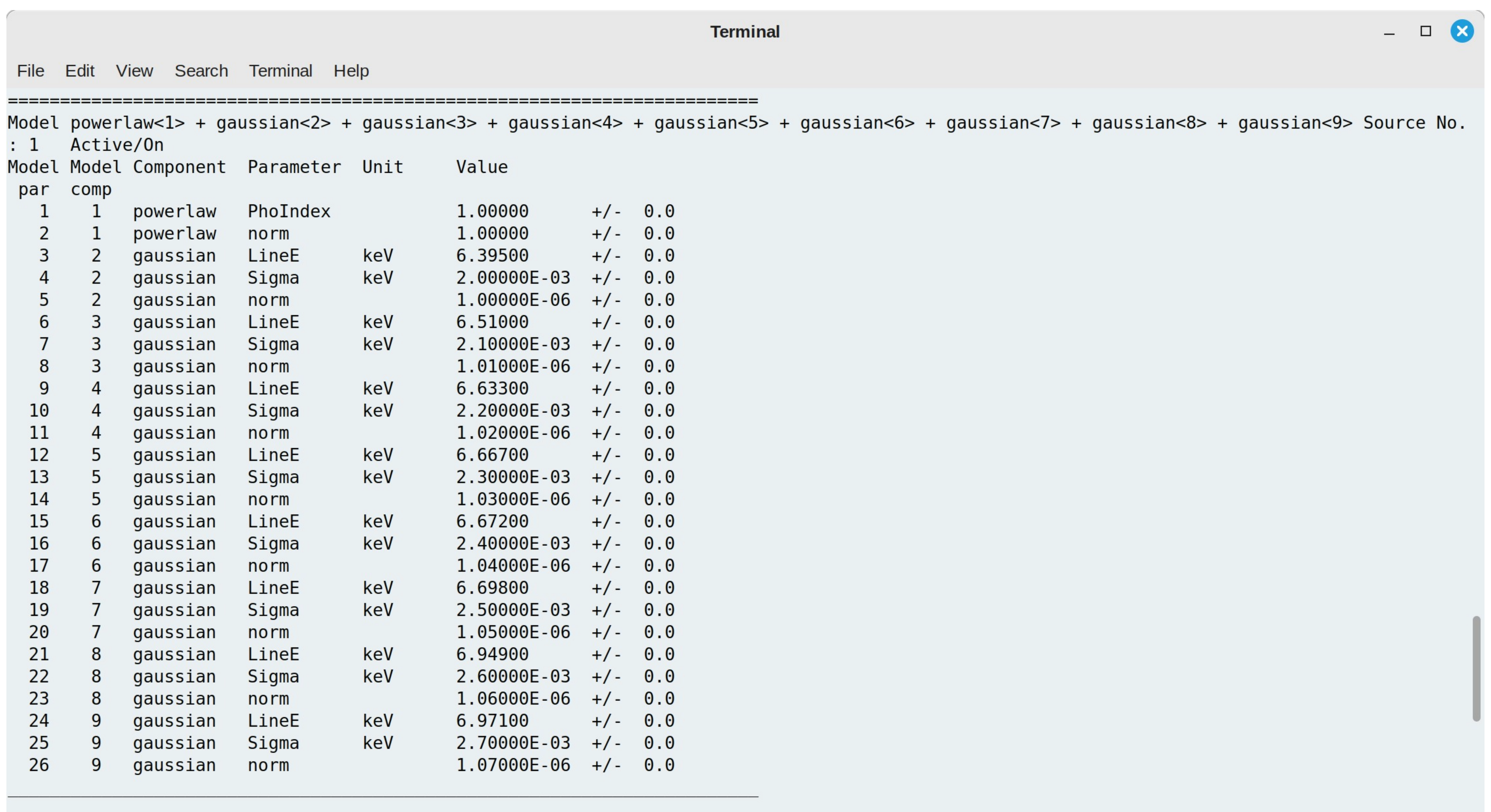


```
Julia
File Edit View Search Terminal Help
  Ancillary Response:
   . Channels            : 60000
   . E (min/max)         : (0.0, 30.0)

julia> begin
         model=
           PowerLaw(a = FitParam(2.0)) +
           GaussianLine(μ=FitParam(6.395)) +
           GaussianLine(μ=FitParam(6.510)) +
           GaussianLine(μ=FitParam(6.633)) +
           GaussianLine(μ=FitParam(6.667)) +
           GaussianLine(μ=FitParam(6.672)) +
           GaussianLine(μ=FitParam(6.698)) +
           GaussianLine(μ=FitParam(6.949)) +
           GaussianLine(μ=FitParam(6.971)) +
           GaussianLine(μ=FitParam(8.210))
           model.σ_1.value = 1.0e-2
           model.σ_2.value = 1.01e-2
           model.σ_3.value = 1.02e-2
           model.σ_4.value = 1.03e-2
           model.σ_5.value = 1.04e-2
           model.σ_6.value = 1.05e-2
           model.σ_7.value = 1.06e-2
           model.σ_8.value = 1.07e-2
           model.σ_9.value = 1.08e-2
           end
0.0108

julia> domain = collect(range(6.00,8.60,1000))
1000-element Vector{Float64}:
 6.0
```

Model Definition – XSPEC



```
Terminal
File Edit View Search Terminal Help
========================================================================
Model powerlaw<1> + gaussian<2> + gaussian<3> + gaussian<4> + gaussian<5> + gaussian<6> + gaussian<7> + gaussian<8> + gaussian<9> Source No.
: 1   Active/On
Model Model Component  Parameter  Unit     Value
 par  comp
   1    1   powerlaw   PhoIndex            1.00000      +/-  0.0
   2    1   powerlaw   norm                1.00000      +/-  0.0
   3    2   gaussian   LineE      keV      6.39500      +/-  0.0
   4    2   gaussian   Sigma      keV      2.00000E-03  +/-  0.0
   5    2   gaussian   norm                1.00000E-06  +/-  0.0
   6    3   gaussian   LineE      keV      6.51000      +/-  0.0
   7    3   gaussian   Sigma      keV      2.10000E-03  +/-  0.0
   8    3   gaussian   norm                1.01000E-06  +/-  0.0
   9    4   gaussian   LineE      keV      6.63300      +/-  0.0
  10    4   gaussian   Sigma      keV      2.20000E-03  +/-  0.0
  11    4   gaussian   norm                1.02000E-06  +/-  0.0
  12    5   gaussian   LineE      keV      6.66700      +/-  0.0
  13    5   gaussian   Sigma      keV      2.30000E-03  +/-  0.0
  14    5   gaussian   norm                1.03000E-06  +/-  0.0
  15    6   gaussian   LineE      keV      6.67200      +/-  0.0
  16    6   gaussian   Sigma      keV      2.40000E-03  +/-  0.0
  17    6   gaussian   norm                1.04000E-06  +/-  0.0
  18    7   gaussian   LineE      keV      6.69800      +/-  0.0
  19    7   gaussian   Sigma      keV      2.50000E-03  +/-  0.0
  20    7   gaussian   norm                1.05000E-06  +/-  0.0
  21    8   gaussian   LineE      keV      6.94900      +/-  0.0
  22    8   gaussian   Sigma      keV      2.60000E-03  +/-  0.0
  23    8   gaussian   norm                1.06000E-06  +/-  0.0
  24    9   gaussian   LineE      keV      6.97100      +/-  0.0
  25    9   gaussian   Sigma      keV      2.70000E-03  +/-  0.0
  26    9   gaussian   norm                1.07000E-06  +/-  0.0
________________________________________________________________________
```

References and Pointers

1. https://julialang.com
2. wikipedia (Julia)
3. SpectralFitting.jl link: github.com/fjebaker/SpectralFitting.jl
4. Baker et al. 2024, ADASS 2024
5. github.com/fjebaker/SpectralFitting.jl/issues
6. wikipedia(automatic differentiation)