

Using The XSPEC Models Library In Other Programs

For those who wish to incorporate the standard XSPEC model functions library into their own programs, XSPEC provides a set of functions and wrappers that can be called from external C, C++ or Fortran programs.

Calling Model Functions From C And Fortran

An increasing number of XSPEC model functions are written in C++, and have the C++-style function interface described in Appendix C. XSPEC provides function wrappers for each of these to make them callable from Fortran or C programs. The wrappers are stored in the files `funcWrappers.h` and `funcWrappers.cxx` in the `XSFunctions` directory.

For each C++ model function there are 2 wrappers: one for passing single precision arrays and one for double precision, with the interfaces as shown in Appendix C for single precision Fortran-style and C-style respectively. The single precision wrapper function name will be the original C++ function name appended with a “f_” prefix, while the double precision wrapper will have a “C_” prefix”.

For example, XSPEC’s `model.dat` entry for the power law model lists the function name `C_powerLaw`. This shows that the actual function name is “powerLaw” and the “C_” indicates it has a C++ interface inside XSPEC. `funcWrappers.cxx` defines the following 2 wrappers:

```
void f_powerLaw(const float* energy, int nFlux, const float* params, int
               spectrumNumber, float* flux, float* fluxError)

void C_powerLaw(const double* energy, int nFlux, const double* params, int
               spectrumNumber, double* flux, double* fluxError, const char*
               initStr)
```

The second function is intended to be called from C programs, while Fortran programs may call either (`funcWrappers.cxx` also includes CERN `<cfortran.h>` definitions to make these accessible to Fortran).

Interface Routines

XSPEC also provides a set of functions for accessing some of the model functions’ internal data. The C++ functions are listed in the file `FunctionUtility.h` in the `XSUtil/FunctionUtils` directory. For C and Fortran access, equivalent wrapper functions are listed in the same directory in `xsFortran.h`. The wrapper functions have C-style function declarations, and are also made available to Fortran calling routines via the CERN `<cfortran.h>` interface.

The currently provided C/Fortran wrapper functions are (see `xsFortran.h` for the function signatures):

FNINIT	Initializes data directory locations needed by the models. See below for a fuller description.
FGABND	Get an element abundance.

FGCHAT	Get current chatter level setting for model functions' output verbosity.
FPCHAT	Set the chatter level. Default is 10, higher chatter levels produce more output.
FGDATD	Get the model .dat files path.
FPDATD	Set the model .dat files path.
FGMODF	Get the model ion data path.
FGMSTR	Get a model string value (see XSPEC xset command).
FPMSTR	Set a model string value.
FPSLFL	Load values of a "file" solar abundance table (see abund command).
FGSOLR	Get the solar abundance table setting.
FPSOLR	Set the solar abundance table.
FGXSCT	Get the cross section table setting.
FPXSCT	Set the cross section table.
csmgh0	Get the cosmology H_0 setting (see the cosmo command).
csmph0	Set H_0 .
csmgl0	Get Λ_0 .
csmpl0	Set Λ_0 .
csmgq0	Get q_0 .
csmppq0	Put q_0 .
fzsq	Computes the luminosity distance, $(c/H_0)*fzsq$. The function is valid for small values of q_0*z for the case of no cosmological constant and uses the approximation of Pen (1999 ApJS 120, 49) for the case of a cosmological constant and a flat Universe. The function is not valid for non-zero cosmological constant if the Universe is not flat.
xs_getVersion (or xgvers)	Retrieve XSPEC's version string.

Initializing the Models Library

The external program should always call the FNINIT routine prior to any other call into the models library. This initializes the locations of the various data files needed by the models, and also sets the abundance and cross-section tables. Unless the user has overridden the model ion data directory location with the XSPEC_MDATA_DIR environment variable, the initial settings are:

Model ion data location	\$HEADAS/./spectral/modelData
Abundance and cross-section .dat files location	\$HEADAS/./spectral/manager
Solar abundance table	angr
Photoelectric cross-section table	bcmc

Building with the Models Library

The XSFunctions library depends on three lower-level XSPEC libraries, XS, XSUtil, and XSModel, and also the CCfits and cfitsio libraries distributed with HEASOFT. A Makefile for a small Fortran program linking with the models library therefore may look like this on Linux:

```
myprog : myprog.o
        g77 -g myprog.o -o myprog \
            -L/path/to/headas/installed/location/lib \
            -lXSFunctions -lXSModel -lXSUtil -lXS -lCCfits_2.1 -lcfitsio_3.11
myprog.o: myprog.f
        g77 -g -c myprog.f
```