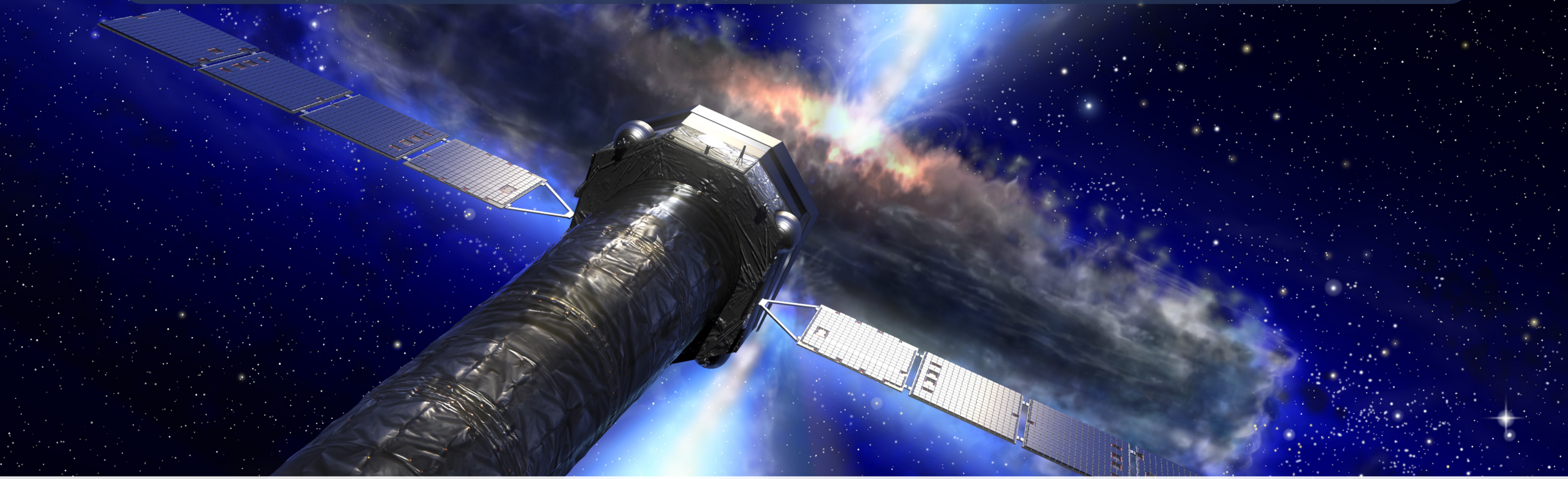
Tutorial for Simput and Sixte



Jörn Wilms, Christian Kirsch & Maximilian Lorenz
Remeis Observatory

Sixte Workshop IFCA, Santander — February 2019

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

WFI ATHENA

IFU ATHENA X-ray Integral Field Unit

Simput & Sixte — Workflow

(1) Preparation of the input of the simulation In this step we define what is to be observed and store it as a SIMPUT file. The input can be point sources, including time variability,extended sources, or the simulation of large catalogues of astronomical sources that can contain millions of X-ray sources on the whole sky.

(2) Running the simulation Run the simulation for a certain detector setup and exposure.

(3) Analyzing the simulation The output of the previous step are one or multiple standard FITS event files. SIXTE provides some tools that prepare standard data products such as spectra and images from event files.

Part 0: Set up the installation

this is part of the installation on the web

- set environment variable SIMPUT and SIXTE in your .bash or .cshrc
- source the correct install file `$SIMPUT/bin/simput-install.(c)sh` `(also in your .bash or .cshrc)`
- check SIMPUT by calling “`plist simputfile`”
- check SIXTE by calling “`plist runsixt`”

Part 1: Preparing the SIMPUT-file

- SIXTE bases the simulations on a FITS standard called SIMPUT. SIMPUT-files allow us to specify point sources, but they also scale well to very complicated setups.
- SIMPUT files can be generated from scratch using your own scripts, for many simulations it is easier to build the SIMPUT file using the program `simputfile`.

1.1 list all parameters of the tool `simputfile`

1.2 create a Crab-like spectral model with `xspec` and save it in a file (flux 1 mCrab)

1.3 construct a SIMPUT-file with this model (*note: write a little shell script*)

detailed commands in the Simulator Manual, Sect. 10.2.2

Part 1: Preparing the SIMPUT-file

1.1 `plist simputfile`

1.2 define model in xspec and save it with

```
XSPEC12>save model mcrab.xcm
```

1.3

```
$SIXTE/bin/simputfile Simput=mcrab.fits \
    RA=0.0 Dec=0.0 srcFlux=2.137e-11 Elow=0.1 Eup=15 \
    NBins=1000 logEgrid=yes Emin=2 Emax=10 \
    XSPECFile=mcrab.xcm
```

Part 2: Running the simulation

- All properties describing an instrument are set in XML-files.
- The general simulator is called with `runsixt`, and can perform a simulation for any given XML-file.

2.1 run a simulation of the 1000 sec long simulation of the SIMPUT file created in (1) for one large chip of the *Athena* WFI centered on the source

2.2 check if the Event File contains a significat fraction of pile-up

2.3 select a different mode of the WFI (have a look at the manual) to reduce pile-up and re-do the simulation

detailed commands in the Simulator Manual, Sect. 10.2.4

Part 2: Running the simulation

2.1 the shell script should look like:

```
#!/bin/bash

base=mcrab

xmldir=$SIXTE/share/sixte/instruments/athena-wfi/wfi_wo_filter_15row
xml=${xmldir}/ld_wfi_ff_large.xml

$SIXTE/bin/runsixt \
    XMLFile=${xml} \
    RA=0.000 Dec=0.000 \
    Prefix=sim_ \
    Simput=${base}.fits \
    EvtFile=evt_${base}.fits \
    Exposure=1000 \
    Mission=Athena \
    Instrument=WFI \
    Mode=large
```

Part 3: Analyzing the simulation

The output of the previous step are one or multiple standard FITS event files. SIXTE provides some tools that prepare standard data products such as spectra and images from event files.

3.1 Use `fstruct` and `fv` or `fdump` to take a look at the structure of the event file. How many events were simulated? Speculate on the meaning of the individual columns in the event file (see below for an explanation of their definition).

3.2 generate an image of the event file using `imgev`

3.3 generate a spectrum using `makespec`

detailed commands in the Simulator Manual, Sect. 10.2.5

Part 3: Analyzing the simulation

3.2 & 3.3 the relevant lines in the shell script could look like:

```
$SIXTE/bin/imgev \
    EvtFile=sim_evt_mcrab.fits \
    Image=img_mcrab.fits \
    CoordinateSystem=0 Projection=TAN \
    NAXIS1=512 NAXIS2=512 CUNIT1=deg CUNIT2=deg \
    CRVAL1=0.0 CRVAL2=0.0 CRPIX1=256.5 CRPIX2=256.5 \
    CDELT1=-6.207043e-04 CDELT2=6.207043e-04 \
    history=true clobber=yes

$SIXTE/bin/makespec \
    EvtFile=sim_evt_mcrab.fits \
    Spectrum=spec_mcrab.pha \
    EventFilter="(RA>359.95 || RA<0.05) && Dec>-0.05 && Dec<+0.05" \
    RSPPath=${xmldir} clobber=yes
```

Part 4: Deep Field Simulations (with the WFI)

This exercise shows how to create a small SIMPUT catalogue (by adding SIMPUTs with `simputmerge`) and simulating them.

4.1 Create a SIMPUT containing several sources within a few arcmin of each other

4.2 plot the chip geometry using `xml2svg` tool

4.3 observe your field with the full WFI large detector array, using `athenawfisim`

4.4 create an image of the whole field

4.5 extract the spectrum of the brightest source

detailed commands in the Simulator Manual, Sect. 10.4.1-10.4.2

Part 4: Deep Field Simulations (continued)

Now we use an existing, complex SIMPUT file. In the following we have to include *dithering* to observe the complete field.

4.6 simulate the Chandra Deep Field South (link to download the necessary files is in the manual) with a staring observation at RA=53.13 Dec=-27.8.

4.7 calculate an exposure map of the given Lissajous-Attitude file for 10 ksec with the `exposure_map` tool

4.8 now simulate the CDFS including dithering with the given Lissajous pattern

4.9 create images of the simulations from 5.6 and 5.8 and compare them

detailed commands in the Simulator Manual, Sect. 10.4.3-10.4.4

Part 5: Adding source variability

Source variability can be easily added in the SIMPUT file generation.

- As a (energy dependent) light curve, specified through an ASCII file (parameter `LCFile` with two columns, time and flux)
- As a stochastic process defined through its power spectrum
- As parameters of the power spectrum, which is defined by the sum of several Lorentzians and a zero-centered low frequency QPO.

5.1 add light curve with the `LCFile` parameter such that the source decreases in flux for you source `mcrab.fits`

5.2 re-run the simulation

5.3 create a light curve with 1 sec binning using `makelc`

detailed commands in the Simulator Manual, Sect. 10.3.1

Part 6: Extended Source Simulations

Images of extended sources (e.g., from observations) can be easily added to a SIMPUT file.

6.1 download a Cas A FITS image (from `https://www.sternwarte.uni-erlangen.de/research/sixte/downloads/Cas_A.img`) and extract the source position

6.2 create a simput file using this image and an `apec` spectral model (suggestion: $kT = 1.25\,\mathrm{keV}$, $N_H = 1 \cdot 10^{22}\,\mathrm{cm}^{-2}$) with a flux of $2.9 \times 10^{-9}\,\mathrm{erg/s/cm^2}$ between 0.3 and 10 keV

Building a simput file from an image is shown in the Simulator Manual, Sect. 10.5.1

Part 6: Extended Source Simulations (continued)

We now compare the results from the X-IFU and WFI

6.3 simulate a 100 s observation with the WFI for one chip

6.4 simulate a 100 s observation for the X-IFU using `xifupipeline`. Create a PixImpactList as well for Part 8 later (see the `plist`)

6.5 construct images from both event files using `imgev`. How do they compare?

6.6 create spectra for both observations via `makespec`. For this, use the same extraction region in RA and Dec in both (keyword: `EventFilter`) and compare them.

A xifupipeline command is shown in the Simulator Manual, Sect. 10.8

Part 7: Simulating Galaxy Clusters with the *Athena* X-IFU

The main difficulty of simulating a galaxy cluster observation lies in the building of a suitable SIMPUT file containing all the information probed by a high-resolution integrated field unit like the X-IFU.

7.1 download the `X-IFU_clusters_tutorial.tgz` file containing all data to simulate A2146

7.2 construct a point source XSPEC model file with an absorbed `apec` model

7.3 create a SIMPUT file of A2146 with `simputmultispec`, using the given temperature and abundance map such that the source spectrum changes over the source

7.4 look at the simput with `fstruct` and/or `fv` and try to understand what `simputmultispec` does

7.5 simulate the source with `xifupipeline`

detailed commands in the Simulator Manual, Sect. 10.7-10.8

Part 8: Simulating TES Pulses with `tessim`

The tessim tool is based on a detailed detector model and accurately simulates the physics of transition-edge sensor based microcalorimeters.

8.1 Use the photon impact list generated in Part 6 to run a `tessim` simulation. To save time, we will only simulate one pixel – how do we do this? Find the pixel with the highest number of impacts.

8.2 simulate the pulses for this impactlist using `tessim`, for a pixel type defined in a file `newpix_LPA75um.fits` (file is available in the X-IFU instrument files). Set the `triggertype` to `stream` and inspect the output file.

Commands on `tessim` are in the Simulator Manual, Sect. 10.9

Part 8: Simulating TES Pulses with `tessim` (continued)

The tessim tool is based on a detailed detector model and accurately simulates the physics of transition-edge sensor based microcalorimeters.

8.3 repeat the simulation, setting `triggersize=8192` and `triggertype=diff:4:50:8192` (a high-resolution pulse is defined to have 8192 samples). How does the number of `TESRECORDS` compare to the number of input photons?

The meaning of the `triggertype` is described in the Simulator Manual, Sect. 8.3.1

Part 9: Simulating a mosaic of M31

Individual observations can easily be combined to create images of larger astronomical objects.

9.1 Download a simput of M31 from `https://www.sternwarte.uni-erlangen.de/research/sixte/downloads/m31.simput`

9.2 Select coordinates for three WFI pointings along the length of the galaxy and simulate them (*Note: The FOV of the WFI is* $40' \times 40'$)

9.3 Combine the event files and create a single image using `imgev`