# Practical Session 3: SIXTE Workflow in Python

The SIXTE Team – Remeis Observatory

Sixte Workshop — March 2022

ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

WFI ATHENA

IFU ATHENA X-ray Integral Field Unit

# SIXTE Workflow (as an example in Python)

**idea: get familiar with the workflow from creating SIMPUT files manually, running simulations, and start analyzing the simulation products**

*No worries if you are not (very) familiar with python:*
*(1) you do not have to use python and*
*(2) no detailed python knowledge is required for the exercise*

Download the necessary material (solutions will be provided later):

`www.sternwarte.uni-erlangen.de/~sixte/downloads/sixte_python_exercise.tgz`

# Creating a SIMPUT file with multiple sources and a light curve

0. create a SIMPUT file of a point source with a Crab spectrum and a flux of 0.5 mCrab ($2 \cdot 10^{-11}$ erg/cm$^2$/s in the 2-10 keV energy band); note that you use the SIMPUT file `athenacrab_1mCrab.simput` given in the data package and set the proper flux values

1. add a second sources 2 arcmin away form the first one
   - use the script **exercise_1_create_simput.py** for this and fill in the blanks
   - create a simput with both sources, re-using the first SIMPUT created in 4.0
   - set the flux of the second source a factor 3 higher (to 1.5 mCrab)
   - calculate a relative light curve, starting at 1, decaying exponentially such that it is at a value of 0.5 at t = 500 sec (total length of the light curve should be 1300 sec)
   - add this light curve as timing extension to the SIMPUT file and attach it to the second source (the one starting at 1.5 mCrab)

2. simulate a 1300 sec observations with the WFI large chip: **exercise_2_runsixt.bash**

# Creating a SIMPUT file with multiple sources and a light curve

3. extract a light curve of each source with `makelc`:
   **exercise_2_makelc.bash**
   - use the extended filename syntax for this (see simulator manual), which should look like:

```
makelc EvtFile='sim_twosources_decay_evt.fits[EVENTS] \
  [sum(SRC_ID)==1]' [...]
```

   for the first source
   - best put in a shell script
   - the `sum(SRC_ID)==1` is necessary as `SRC_ID` can be an array, as in an unlikely case more than 1 photon can contribute to an array
   - alternatively, a selection with `[RA<?? && RA>?? && Dec<?? && Dec>??]` works as well

4. plot the light curve and compare it with the input from the SIMPUT file
   - you can use the script **exercise_4_analysis.py** for this and fill in the blanks
   - **Question:** why does the decaying light curve not follow the input?

# Creating a SIMPUT file with multiple sources and a light curve

5. Bonus 1: calculate the pile-up fraction
6. Bonus 2: redo the simulation with an appropriate detector/readout configuration
   - verify that the extracted light curve now follows the simulated one
   - what is the pile-up fraction now?

*As a very good reference, Jupyter Notebooks with all examples from the simulator manual can be found at* `https://github.com/mtceballos/SIXTE-tutorial`