ixpeobssim: a simulation and analysis framework for IXPE

Niccolò Di Lalla
niccolo.dilalla@stanford.edu

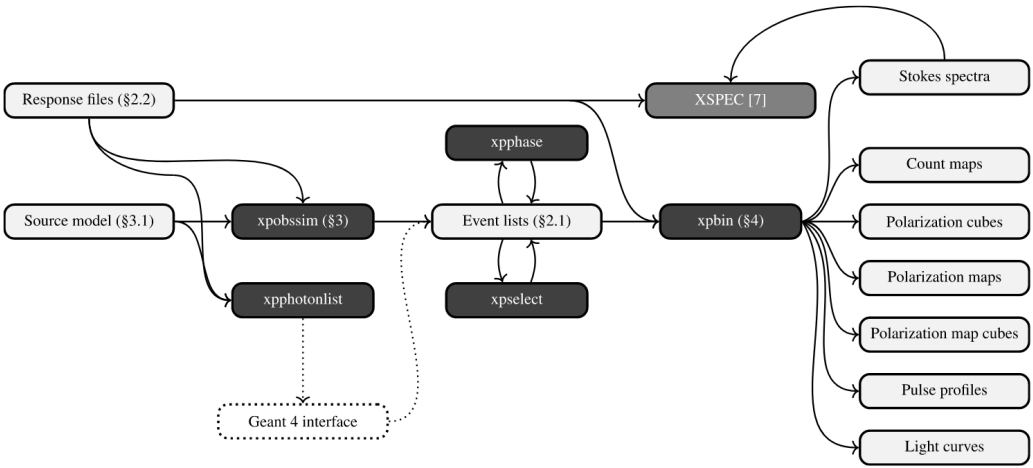
Stanford University

IXPE Splinter Session, HEAD21 Meeting

Introduction

- Project started in 2015 under the name of `XIMPOL`:
  - Initially not tied to any specific mission or instrument design
  - After IXPE selection in 2017, it was renamed and progressively tailored in preparation for the new mission
  - Publicly released in 2022 to support the analysis of public IXPE data and engage the broader community in anticipation of the General Observer program
- Simulation and analysis framework:
  - Based on `python` programming language and the associated scientific ecosystem
  - Designed to produce fast and realistic simulated IXPE observations
  - Complemented by a suite of post-processing applications to select, bin and analyze simulated and real IXPE data
- Output data are:
  - Event lists in FITS format, containing a strict superset of the information included in the publicly released IXPE data products
  - Fully compliant with the visualization and analysis tools commonly used by the X-ray community (`XSPEC`, `Sherpa`, `3ML`, `DS9`, `HENDRICS`).

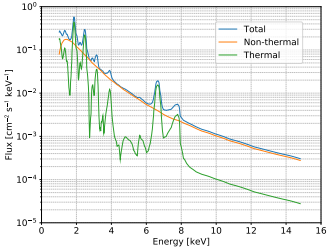
Architectural Overview

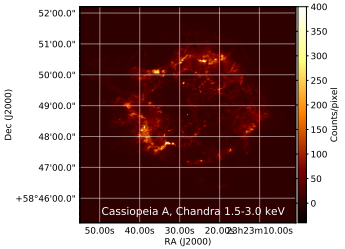


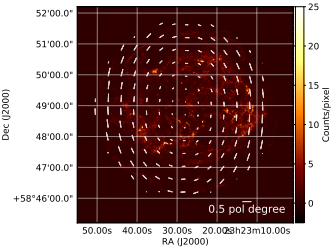
Source model definition

Cassiopeia A example

- Need to define essentially three source properties:
  - Morphology (point sources, disks, annuli, generic extended sources from FITS images)
  - Energy spectrum in units of $[\mathrm{cm}^{-1}\,\mathrm{s}^{-1}\,\mathrm{keV}^{-1}]$
  - Polarization model (degree and angle, or Stokes parameters Q and U)
- Can use a Chandra photon list in lieu of defining morphology and spectrum
- Can overlay several components in the same model
- Support for time-dependent transient and periodic sources

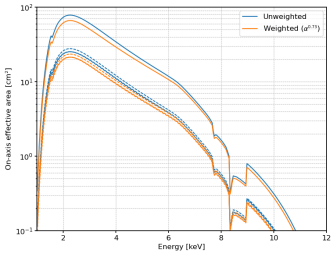


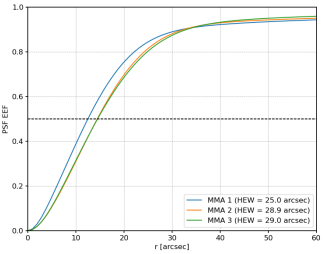


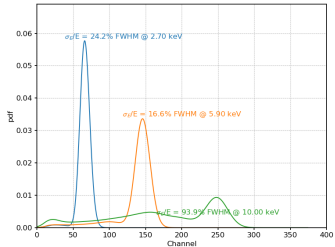


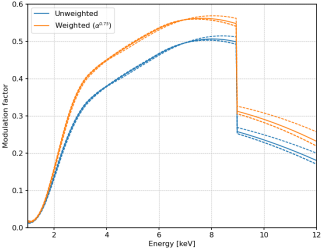
Instrument Response Functions

- Each of the three DUs has its own set of IRFs:
  - FITS files compliant with the OGIP format
  - Weighted and un-weighted flavors
- Generated and stored in a local CALDB:
  - Kept in sync with the official IXPE CALDB
- Latest version (v13) released a month ago:
  - Time-dependent, validity time binned in 6-month interval
  - User has to select the appropriate set of IRFs



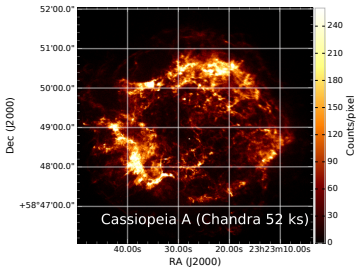


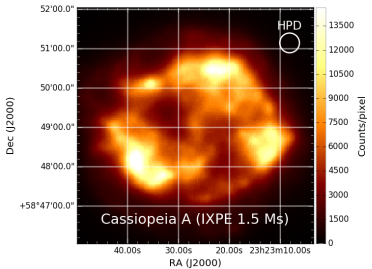


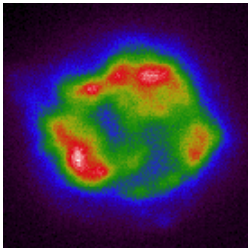


xpobssim

Simulation flow



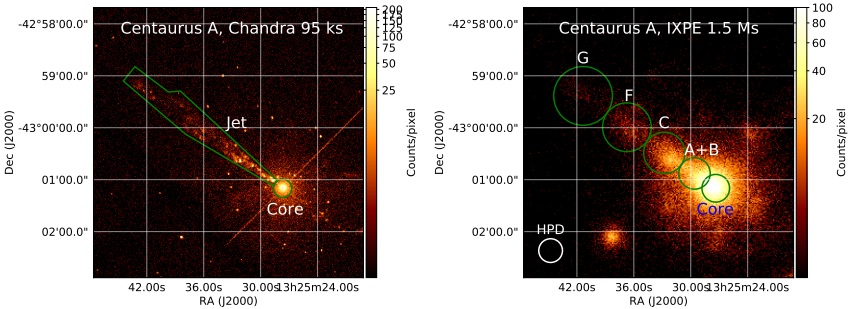




- Simulate an observation starting from an arbitrary source model:
  - Calculate the expected number of events by convolving the source spectrum with the effective area and extract the event times based on the light curve
  - Extract the true energies and sky positions and smear them with energy dispersion and PSF
  - Generate the angular distribution of the photoelectrons according to the polarization model
- With composite sources, the simulation is performed separately for each component and the resulting photon lists are then merged together

Conversion of a Chandra Observation

Simulation flow



- Process an actual archived Chandra photon list to produce an IXPE simulation:
  - Chandra measured energies, times and positions taken as MC truth
  - Events are down-sampled and smeared with the IXPE response functions
  - The angular distribution of the photoelectrons is generated according to the provided polarization model
- Preserve the full correlation between the morphology and the energy spectrum

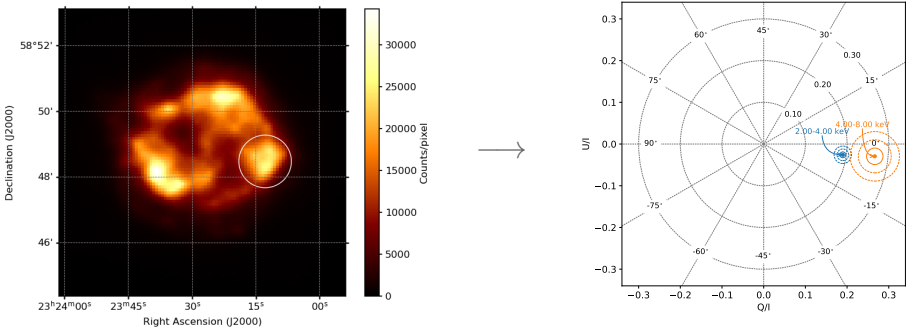
Analysis Tools

- *ixpeobssim* is distributed with its own set of analysis tools:
  - Provide an easy-to-use interface to manipulate simulated and real IXPE data
  - Fully configurable via command-line options
  - Can be either used as stand-alone applications or easily combined into complex analysis pipelines in `python`
  - Full support for weighted and un-weighted types of analysis
- *xpphase*:
  - Calculate the phase of a periodic source based on its ephemeris
- *xpselect*:
  - Filter event lists based on energy, direction, time or phase
- *xpbin*:
  - Bin the data using several different algorithms, producing binned events lists
  - HEASOFT `xselect` FTOOL provides support for part of the same functionalities since v6.30
- *xpbinview*:
  - Visualize the binned data products
- *xpstokesalign*:
  - Align the Stokes parameters to a given polarization model on an event-by-event basis

Polarization cubes

Binned Data Products

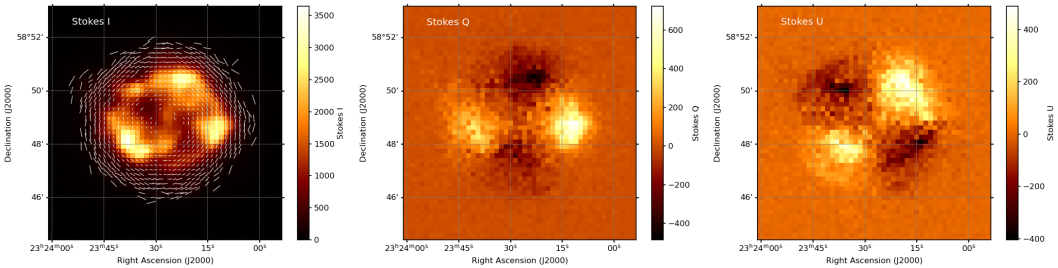
- The simplest possible data structure holding polarization information
- Table listing I, Q, U, polarization degree and angle with the associated uncertainties in multiple energy bins:
  - Provided with methods to rescale and subtract the background contribution (see talk by Stefano)



Polarization maps

Binned Data Products

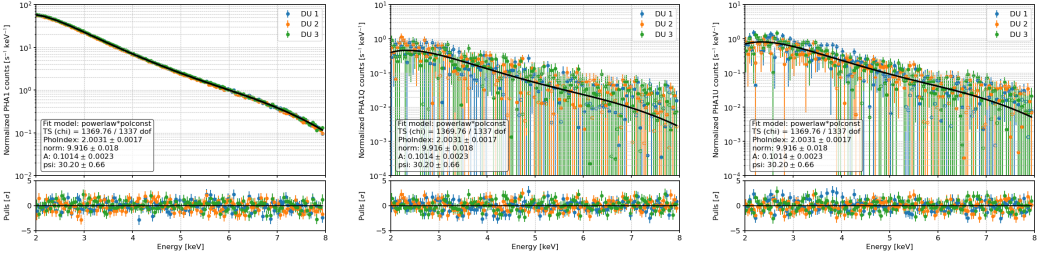
- Hold the exact same information as polarization cubes, but binned in sky-coordinates
- Maps of I, Q, U, polarization degree and angle in multiple energy layers:
  - Provided with methods to convolve the map with a generic binned kernel and overlay the arrows of polarization information



Stokes spectra

Binned Data Products

- Main interface to spectro-polarimetric fitting in `XSPEC`, `Sherpa` and `3ML`
- Generalization of the standard PHA spectra:
  - PHA1, PHA1Q and PHA1U
- Lightweight python wrapper dubbed *xpxspec* shipped with *ixpeobssim*:
  - Together with a few simple, multiplicative polarimetric models provided by HEASARC through the page hosting XSPEC additional models



References

- Github webpage: `https://github.com/lucabaldini/ixpeobssim`
- Software documentation: `https://ixpeobssim.readthedocs.io/en/latest/index.html`
- Paper: `https://www.sciencedirect.com/science/article/pii/S2352711022001169`
- Pip: `https://pypi.org/project/ixpeobssim/`
  - `pip install ixpeobssim`
  - Latest version: 31.0.1 released on March 8, 2024
- Tutorial @HEAD20: `https://drive.google.com/drive/folders/1AGixwB3TSLGvMeQ89ICE-Ww6FL2QSXbe?usp=sharing`

SoftwareX 19 (2022) 101194

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

ixpeobssim: A simulation and analysis framework for the imaging X-ray polarimetry explorer

Luca Baldini [a,b,*], Niccolò Bucciantini [c,d,e], Niccolò Di Lalla [f], Steven Ehlert [g], Alberto Manfreda [b], Michela Negro [h,i,j], Nicola Omodei [f], Melissa Pesce-Rollins [b], Carmelo Sgrò [b], Stefano Silvestri [a,b]

[a] Università di Pisa, Dipartimento di Fisica Enrico Fermi, Largo B. Pontecorvo 3, I-56127 Pisa, Italy
[b] Istituto Nazionale di Fisica Nucleare, Sezione di Pisa, Largo B. Pontecorvo 3, I-56127 Pisa, Italy
[c] Istituto Nazionale di Astrofisica, Osservatorio Astrofisico di Arcetri, Largo E. Fermi 5, I-50125, Firenze, Italy
[d] Dipartimento di Fisica & Astronomia, Univesità degli Studi di Firenze, Via G. Sansone 1, 50019, Sesto F.no, Italy
[e] Istituto Nazionale di Fisica Nucleare, Sezione di Firenze, Via G. Sansone 1, 50019 Sesto F.no, Italy
[f] W.W. Hansen Experimental Physics Laboratory, Kavli Institute for Particle Astrophysics and Cosmology, Department of Physics and SLAC National Accelerator Laboratory, Stanford University, Stanford, CA 94305, USA
[g] NASA Marshall Space Flight Center, Huntsville, AL 35812, USA
[h] University of Maryland, Baltimore County, Baltimore, MD 21250, USA
[i] NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA
[j] Center for Research and Exploration in Space Science and Technology, NASA/GSFC, Greenbelt, MD 20771, USA

ARTICLE INFO

Article history:
Received 12 March 2022
Received in revised form 21 July 2022
Accepted 20 August 2022

Keywords:
X-ray polarimetry

ABSTRACT

ixpeobssim is a simulation and analysis framework specifically developed for the Imaging X-ray Polarimetry Explorer (IXPE). Given a source model and the response functions of the telescopes, it is designed to produce realistic simulated observations, in the form of event lists in FITS format, containing a strict superset of the information included in the publicly released IXPE data products. The core simulation capabilities are complemented by a full suite of post-processing applications which support the spatial, spectral, and temporal models needed for analysis of typical polarized X-ray sources, allowing for the implementation of complex, polarization-aware analysis pipelines, and facilitating the interoperation with the standard visualization and analysis tools traditionally in use by the X-ray community. Although much of the framework is specific to IXPE, the modular nature of the underlying implementation makes it potentially straightforward to adapt it to different missions with polarization capabilities.

Code metadata

Current code version	29.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00065
Permanent link to Reproducible Capsule	
Legal Code License	GPL-3.0
Code versioning system used	git
Software code languages, tools, and services used	python
Compilation requirements, operating environments & dependencies	numpy, scipy, matplotlib, astropy, regions, skyfield
If available Link to developer documentation/manual	https://ixpeobssim.readthedocs.io
Support email for questions	luca.baldini@pi.infn.it

1. Introduction

Launched on December 9, 2021, the Imaging X-ray Polarimetry Explorer (IXPE) is a NASA Small Explorer Mission developed in collaboration with the Italian Space Agency [1–3], and the first

* Corresponding author at: Università di Pisa, Dipartimento di Fisica Enrico Fermi, Largo B. Pontecorvo 3, I-56127 Pisa, Italy.
E-mail address: luca.baldini@pi.infn.it (Luca Baldini).

https://doi.org/10.1016/j.softx.2022.101194

Conclusions

- *ixpeobssim* was developed to support the IXPE mission by providing advanced simulation and analysis facilities
- With the official public release of IXPE data, we decided to release the codebase under an OSI-approved license:
  - Support the community engaged in the analysis of IXPE data
  - Support the simulation effort required for the General Observer program
  - Encourage reuse for future X-ray (polarimetry?) missions
- *ixpeobssim* is stable but still under an active development phase:
  - A new releases every 1–2 months, on average (check the release notes!)
  - Please open a new issue on github if you find a bug or have something to propose/discuss
  - Everyone is very welcome to participate and help us with the development!
- Many areas can be improved and are currently in the works:
  - Improve the current simplistic, azimuthally-symmetric model for the PSF
  - Implement a tool to quickly evaluate the effect of the polarization leakage
  - Add new analysis tools and algorithms

SPARE SLIDES

Configuration File

Point source with background example

```
# Source coordinates, in decimal degrees.
SRC_NAME = 'Toy point source w/ bkg'
SRC_RA, SRC_DEC = 45., 45.

# Pointing coordinates
PNT_RA, PNT_DEC = SRC_RA, SRC_DEC

# Spectral and polarimetric parameters
PL_NORM = 1          # cm-1 s-1 keV-1 @ 1 keV
PL_INDEX = 2.        # -2
PD = 0.1             # 10%
PA = 30.             # 30 degrees
SPEC = power_law(PL_NORM, PL_INDEX)
POL_DEG = constant(PD)
POL_ANG = constant(numpy.radians(PA))

# Definition of the sources and the region of interest.
SRC = xPointSource(SRC_NAME, SRC_RA, SRC_DEC, SPEC, POL_DEG, POL_ANG)
BKG = xTemplateInstrumentalBkg()
ROI_MODEL = xROIModel(PNT_RA, PNT_DEC, SRC, BKG)
```

Simulation and Analysis

Point source with background example

```
def simulate(duration=100000):
    """Run the simulation.
    """
    pipeline.xpobssim(duration=duration, configfile='toy_point_source_bkg.py')

def select(src_rad=0.75, bkg_inner_rad=1.5, bkg_outer_rad=3.):
    """Select the photon lists.
    """
    file_list = pipeline.file_list()
    pipeline.xpselect(*file_list, rad=src_rad, suffix='src')
    pipeline.xpselect(*file_list, innerrad=bkg_inner_rad, rad=bkg_outer_rad, suffix='bkg')

def bin_(ebinning=[2, 4, 8]):
    """Create the necessary binned files.
    """
    pipeline.xpbin(*pipeline.file_list(), algorithm='CMAP')
    kwargs = dict(algorithm='PCUBE', ebinalg='LIST', ebinning=ebinning)
    pipeline.xpbin(*pipeline.file_list('src'), **kwargs)
    pipeline.xpbin(*pipeline.file_list('bkg'), **kwargs)
    for algorithm in ['PHA1', 'PHA1Q', 'PHA1U']:
        pipeline.xpbin(*pipeline.file_list('src'), algorithm=algorithm)
        pipeline.xpbin(*pipeline.file_list('bkg'), algorithm=algorithm)
```

Basic polarization analysis using Stokes parameters

Kislat et al., Astroparticle Physics **68**, 45–51 (2015)

- For each measured angle ϕ_k, a set of Stokes parameters can be defined as:

$$\begin{aligned} i_k &= 1, \\ q_k &= \cos 2\phi_k \\ u_k &= \sin 2\phi_k. \end{aligned}$$

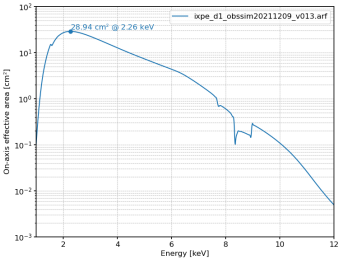
- Owing to their linearity, the analysis for a data-set consisting of N events reduces to:

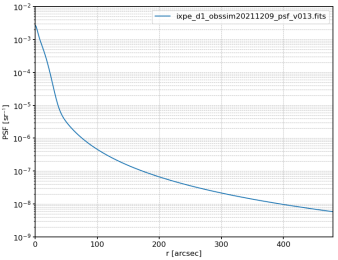
$$I = \sum_{k=1}^{N} i_k = N, \qquad Q = \sum_{k=1}^{N} q_k, \qquad U = \sum_{k=1}^{N} u_k.$$

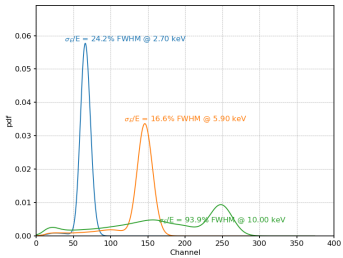
- Finally, the degree and angle of polarization can be estimated as:

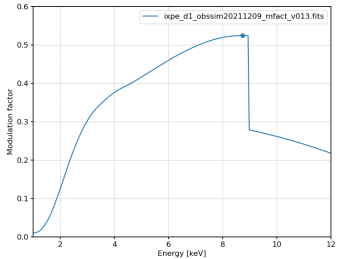
$$\begin{aligned} P &= \frac{2}{\mu} \frac{\sqrt{Q^2 + U^2}}{I} \\ \phi &= \frac{1}{2} \arctan \frac{U}{Q}. \end{aligned}$$

Instrument Response Functions



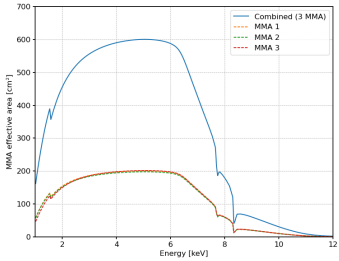


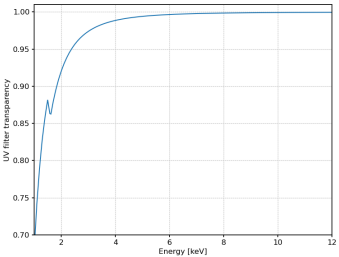


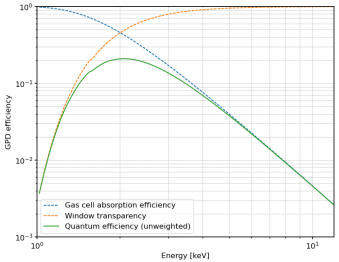


IXPE effective area

Relevant contributions







IXPE field of view