ixpeobssim: a Simulation and Analysis Framework for the Imaging X-ray Polarimetry Explorer

Baldini L.[a,b], Bucciantini N.[c], Di Lalla N.[1], Ehlert S. R.[e], Manfreda A.[b], Omodei N.[1], Pesce-Rollins M.[b], Sgrò C.[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 Fermi 5, I-50125, Firenze, Italy*
[d]*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*
[e]*NASA Marshall Space Flight Center, Huntsville, AL 35812, USA*

Abstract

ixpeobssim is a simulation and analysis framework, based on the Python programming language and the associated scientific ecosystem, 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 super-set of the information provided by standard IXPE level-2 files. The core *ixpeobssim* simulation capabilities are complemented by a full suite of post-processing applications, allowing for the implementation of complex, polarization-aware analysis pipelines, and facilitating the inter-operation with the standard visualization and analysis tools traditionally in use by the X-ray community.

We emphasize that, although a significant part 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.

Keywords: X-ray polarimetry
PACS: 95.55.Ka, 95.55.Qf

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, 2, 3], and the first ever to provide position-resolved polarimetric capabilities in the 2–8 keV energy band.

IXPE recovers the linear polarization of the source on a statistical basis, by measuring the azimuthal distribution of the photo-electrons generated by X-rays absorbed in the detector, and complements the polarization sensitivity with timing, spectral and imaging capabilities [1]. From the standpoint of high-level science analysis, each event is characterized by five independent quantities: the arrival time, the energy, two sky-coordinates and the azimuthal angle ϕ_i of the photo-electron in the tangent-plane projection. For a number of reasons[1], we encode the polarization information into the Stokes parameters

$$q_i = 2\cos 2\phi_i \quad \text{and} \quad u_i = 2\sin 2\phi_i, \tag{1}$$

following the formalism in [5][2]. It should be emphasized that q_i and u_i are expressed in detector space, i.e., they represent the measured modulation, as opposed to the source polarization; we shall discuss how to account for the detector response in section 4. The information included in the publicly-distributed files is complemented by an additional quantity representing the estimated quality of the direction reconstruction (or *weight*) w_i, that can be exploited in an ensemble analysis to enhance the polarization sensitivity [6].

In this paper we describe *ixpeobssim*, a framework designed to simulate IXPE observations of celestial sources, producing event lists in FITS format that is intended to be a strict super-set of that of the standard level-2 files. In addition to the simulation facilities, *ixpeobssim* provides a set of applications to filter, reduce, analyze and visualize both simulated and real data, which we anticipate will be a useful resource for the community engaged in the analysis of IXPE data.

2. Architectural Overview

The *ixpeobssim* framework is based on the Python programming language and makes extensive use of the associated scientific ecosystem, most notably *numpy* [8], *SciPy* [9] and *mat-*

[1]This is primarily due to the approach we use for the calibration of the detector response to un-polarized radiation [4], but we also point out that a formulation in Stokes parameter space is less prone to the pitfalls associated to the fact that the polarization angle and degree are not statistically independent.

[2]We note the extra factor of 2 with respect to the original paper. We also emphasize that, after the calibration of the detector response, the corrected Stokes parameters are generally not properly normalized, and cannot be readily interpreted in terms of an effective azimuthal angle. As pointed out in [4], the effect averages out on any large ensemble of events, and has essentially no practical implications.

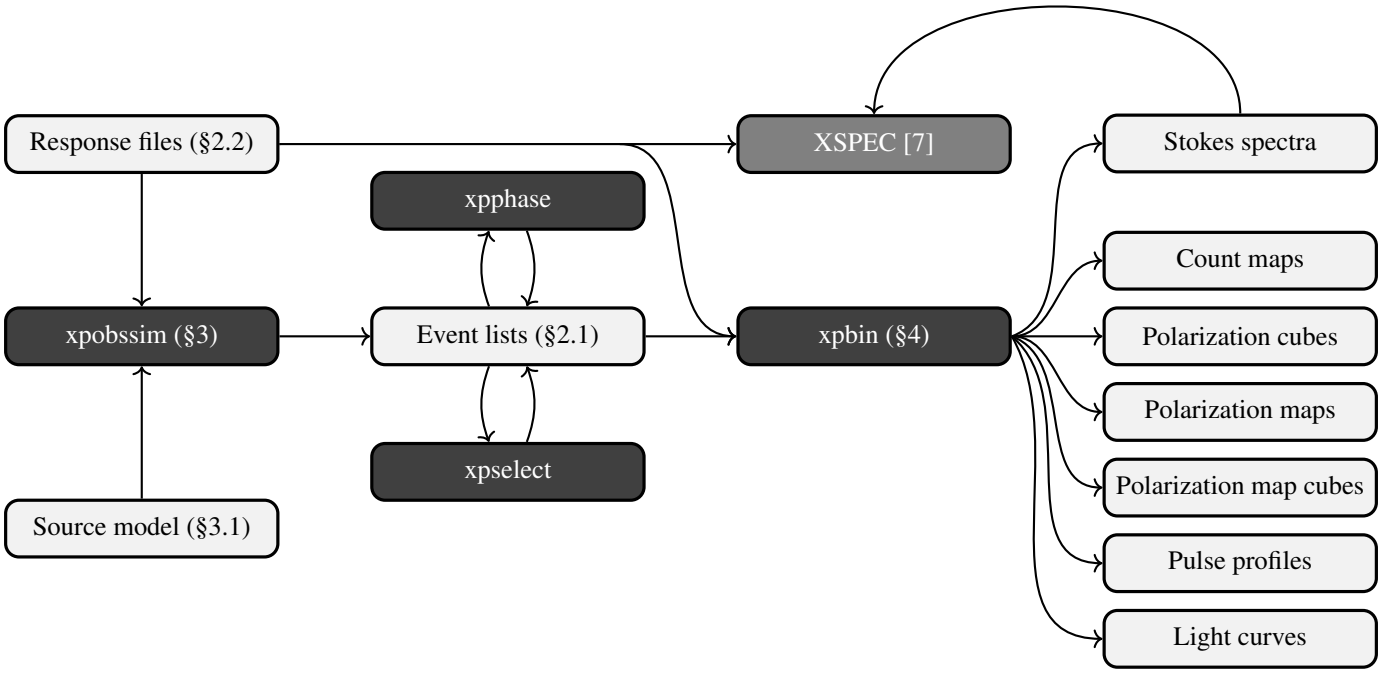

Figure 1: *ixpeobssim* architectural overview. The dark boxes identify specific *ixpeobssim* applications, whereas light boxes represent different types of data products. (XSPEC, as an external program, is rendered in a different style.) The numbers in parentheses provide the reference to the proper section of the paper where each item is discussed.

plotlib [10], as well as the de-facto standard package for numerical analysis in astronomy: *Astropy* [11, 12]. Leveraging over the Python intuitive syntax, extensibility and introspection capabilities, *ixpeobssim* is streamlined for speed and modularity, with the ultimate goal of making it easy for the end-user to create complex simulations and analysis workflows.

Figure 1 shows a simplified architectural overview of the framework. The main application, *xpobssim*, takes a complete source model (including the temporal, morphological, spectral and polarimetric characteristics) and a coherent set of parameterized response files (most notably the effective area and associated vignetting function, the response matrix, the modulation factor as a function of the energy, as well as a model of the PSF) to produce an event list that is germane to an actual level-2 file from a celestial observation. The construction of the source models, the response files and the format of the event lists will be covered in more detail in sections 3.1, 2.2 and 2.1, respectively.

ixpeobssim provides a number of tools for modifying and post-processing event lists, among which *xpselect* allows to apply arbitrary selections on time or phase, energy and position in the sky. Filtered event lists are intended to be functionally identical to their parents to be able to inter-operate with all the analysis tools in exactly the same fashion.

xpbin provides the capability of reducing event lists, generating binned data products in a number of different fashions, including *Stokes spectra* (i.e., standard .pha1 files and their generalizations for the Q and U Stokes parameters, suitable to perform spectro-polarimetric fits) and several different data structures encapsulating the results of ensemble analyses a la [5].

2.1. Output data format

Event lists generated by *ixpeobssim* provide the two extensions in the standard IXPE level-2 data: `EVENTS`, containing the event data, and `GTI`, containing the good time intervals (GTI) for the observation. Compared to the standard format definition [13], *ixpeobssim* simulated event lists include a few additional columns in the `EVENTS` extension, either for user convenience (e.g., energy and sky-direction in physical units) or for diagnostics purposes (e.g., event positions in detector coordinates); none of these additional columns is used by the high-level analysis tools.

The ground truth, including the true photon energy, the true direction in the sky and a unique identifier of the particular source (or source component) in the field of view that originated the event, is captured in a dedicated `MONTE_CARLO` extension. The latter is used for debugging purposes and, more importantly, because it readily provides a way to evaluate the effect of the detector response on any given high-level observable. A few additional binary extensions can be optionally generated, and will be briefly described in the following sections.

2.2. Response functions: the pseudo calibration database

The instrument response functions (IRFs) are a fundamental part of *ixpeobssim*, and they are used, in identical form, for both the simulation and the science analysis. Notably, this allows for a comprehensive verification of the full analysis workflow under controlled conditions. All the response files are OGIP-compliant and are intended to be inter-operable with the analysis tools provided by HEASARC.

More specifically, We identify six different types of response functions: on-axis effective area and response matrix (stored in standard .arf and .rmf FITS files), vignetting, point-spread function, modulation factor and modulation response function. The modulation factor represents the response of the detector to 100% polarized radiation, and serves the purpose of converting the azimuthal modulation measured by the detector into the actual source polarization. The modulation response function is the product of the modulation factor and the on-axis effective area, and is used as the proper ancillary response files for Q and U spectra in polarimetric fits, as explained in section 4.1.

ixpeobssim provides facilities for generating, reading, displaying and using IRFs. Due to the specific needs of the simulation facilities, *ixpeobssim* is distributed with its own, self-contained calibration database, that we shall refer to as the *pseudo-CALDB* and is structurally equivalent to the real database distributed through HEASARC.

3. Simulating Observations

3.1. Source model definition

The basic characteristics of each model *component* are specified as ordinary Python functions: the photon spectrum can be an arbitrary function of energy and time (or phase, for periodic sources), an in principle also on position, while the polarization degree and angle can be an arbitrary function of energy, time (or phase), and sky-direction:

$$\begin{cases} \mathcal{S}(E,t) & [\mathrm{cm}^{-2}\ \mathrm{s}^{-1}\ \mathrm{keV}^{-1}] \\ P_D(E,t,x,y) & \\ P_A(E,t,x,y) & [\mathrm{rad}]. \end{cases} \tag{2}$$

This approach allows for a large degree of flexibility, as the function bodies can contain, e.g., complex analytic functions or interpolators build from numerical tables, provided that the signature is correct. We emphasize that, since the input model does not have associated errors by its nature, our treatment is strictly equivalent to a formulation in Stokes parameter space, and none of the nuisances connected with the fact that *measured* polarization degree and angle are not independent applies to the simulation process.

ixpeobssim allows to control the spatial morphology of the model through a rich hierarchy of classes describing point sources, simple geometrical shapes like disks or annuli, and arbitrary extended sources based on intensity maps in FITS format. Additionally, a dedicated code path allows to define model components based on Chandra ACIS-I/S event lists, leveraging on Chandra's superior energy and spatial resolution This provides an alternative simulation strategy fully preserving the correlations between the position in the sky and the spectral shape, overcoming the seemingly restrictive constraints on the signature of the photon spectrum (2).

It is important to note that the concept of model component is used throughout *ixpeobssim* to describe a few slightly different, but related, objects: a simple celestial source that cannot be further decomposed, different physical components of the same source (e.g., the thermal and non-thermal emission), or different physical sources in the same field (e.g., a PWN and its pulsar). The basic simulation unit, that we refer to as a a *region of interest* (ROI), is a collection of an arbitrary number of model components within the IXPE field of view, encapsulated in a Python module that can be fed into *xpobssim*.

3.2. Simulation workflow

Considering for simplicity an on-axis point source, the basic flow of the simulation for a single model component starts with the calculation of the count spectrum, as a function of energy and time (or phase), given the photon spectrum $\mathcal{S}(E,t)$ and the on-axis effective area $A_{\mathrm{eff}}(E)$:

$$C(E,t) = \mathcal{S}(E,t) \times A_{\mathrm{eff}}(E) \quad [\mathrm{s}^{-1}\ \mathrm{keV}^{-1}]. \tag{3}$$

A simple model for the phase-resolved count spectrum of the Crab pulsar is shown in Figure 2 for illustrative purposes.

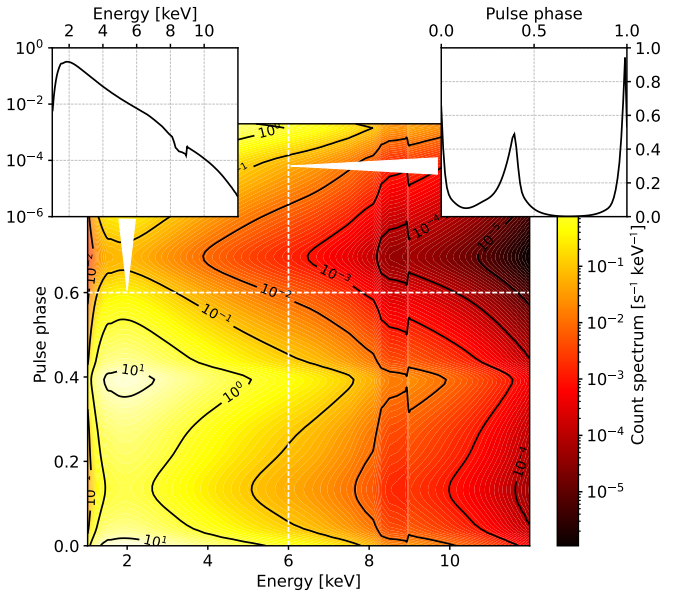

Figure 2: Representation of a simple model for the count spectrum of the Crab pulsar (note that, being the source periodic, the y-axis represents the pulse phase). Any horizontal slice represents the photon spectrum, convolved with the on-axis effective area of the telescope, at a given phase, while any vertical slice represents the pulse profile at a given energy.

The light curve (or the pulse profile) in counts space is readily obtained by integrating over the energy

$$\mathcal{L}(t) = \int_{E_{\mathrm{min}}}^{E_{\mathrm{max}}} C'(E,t)\,\mathrm{d}E \quad [\mathrm{s}^{-1}], \tag{4}$$

which in turn allows to calculate the total number of expected events N_{exp} by integrating over the observation time:

$$N_{\mathrm{exp}} = \int_{t_{\mathrm{min}}}^{t_{\mathrm{max}}} \mathcal{L}(t)\,\mathrm{d}t. \tag{5}$$

We extract the number of observed events N_{obs} according to a Poisson distribution with mean N_{exp} and treat the light curve

as a one-dimensional probability density function (pdf) to extract the initial vector of event times t_i (or phases p_i, for periodic sources). For each event we use the count spectrum $C(E, t_i)$, calculated at the proper time or phase, as a one-dimensional pdf from which we extract the true energy E_i.

The sky-direction (x_i, y_i) is sampled independently, based on the source type, and the photo-electron emission angle, extracted at the proper modulation

$$m_i = P_D(E_i, t_i, x_i, y_i) \times \mu(E_i) \tag{6}$$

and position angle, completes the ground truth for the simulation, which is then properly convolved with the detector response—particularly, the PSF and the response matrix—to provide the measured energy and direction in the sky.

The process is repeated independently for all the source components in the ROI, and the partial event lists are then merged and sorted in time. All the additional instrumental corrections necessary to account for the effect of the vignetting, the dead time and the fiducial area of the focal-plane detectors, are applied in dedicated filtering stages before the final, consolidated event list is written to file.

3.3. Low-level implementation

The vast majority of the probability density functions involved in the simulation of celestial sources can only be sampled via numerical methods—even a simple power-law photon spectrum, when convolved with the telescope effective area, cannot be sampled by analytical means. We largely rely on interpolating splines as an effective mean to sample arbitrary random variables.

For one-dimensional pdfs, we calculate the cumulative function on a suitable, regular grid and use the values to build an interpolated spline representing the percent point function (ppf), that can in turn be used to sample the underlying random variable r via inverse transform:

$$r = \text{ppf}(\text{Unif}[0, 1]). \tag{7}$$

Provided that the order of the spline and grid spacing are selected properly for the situation at hand, this allows for a formulation of the problem that can be naturally vectorized in an efficient fashion through the *numpy* facilities. Implementation details aside, this is the basic strategy that we use to sample the event times from the light curve (or the phase values from the pulse profile for periodic sources).

The problem of extracting the event energies given a count spectrum is intrinsically more complex, due to the fact that the spectral shape, in general, depends on the event time or phase. As re-computing the one-dimensional pdf for each event would be overly computationally-intensive, we resort to precalculating the ppf values on a regular grid of time (or phase) values and use them to build an interpolated bi-variate spline, that we refer to as the *horizontal* ppf (hppf), as illustrated in Figure 3. The latter can then be used to sample the underlying

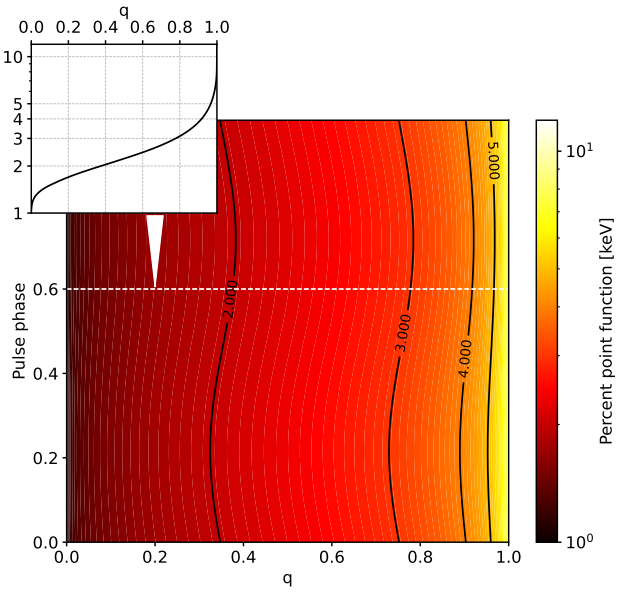

Figure 3: Bi-dimensional ppf corresponding to the count Crab spectrum in Figure 2. Each horizontal slice represents the actual ppf at a given pulse phase, and can be used to sample the underlying event energy in a fully vectorized fashion.

2-dimensional distribution in a vectorized fashion[3]:

$$E = \text{hppf}(\text{Unif}[0,1], t). \tag{8}$$

This is the basic approach that, properly re-cast in azimuthal angle-modulation space, we also use to sample the emission direction of the photo-electrons.

3.4. Event weights

Since the the *ixpeobssim* simulation facilities are entirely based on a parametrization of the detector response, as opposed to an actual microscopic simulation of particle interactions, one of the intrinsic limitations is the inability to generate track images. This, in turn, makes it non trivial to properly simulate the event-by-event weights and provide the complete set of information that one would find in real level-2 data.

To this end, the IXPE Collaboration has been largely relying on the *ixpeobssim* interface to the microscopic Monte Carlo simulation of the detectors, based on the Geant 4 toolkit [14], developed to support the design and implementation of the mission and inform the generation of the response files. This provides the ultimate fidelity (at the expense of a much longer simulation time) and has been extensively used to test the use of weights in the tools that *ixpeobssim* provides to analyze real observations. However, due to its inherent complexity, as well as the dependence of various large external libraries, packaging the detector simulation into a form that could be publicly distributed and supported was deemed to be too resource consuming, given the relatively niche use case.

[3] It is worth noticing that being able to specify a source model in terms of a position-dependent spectrum $\mathcal{S}(E, t, x, y)$ would require generalizing this approach from two to four dimensions, which is the main reason for the corresponding limitation in (2).

At the time of writing, *ixpeobssim* simply sets the weights to unity for all the events. This allows to gauge the sensitivity of a weighted ensemble analysis for any given source, provided that the proper response files are used at simulation time, and guarantees at the same time that simulated data are transparently inter-operable with all the high-level analysis tools. On the minus side, this simplistic approach does not allow to perform end-to-end tests of weighted polarimetric analyses with pure *ixpeobssim* simulations. While future versions may offer realistic event weights through a hybrid approach (e.g., a static look-up table generated with the full detector simulation), this is currently one of the main limitations, and the user should be aware of the implications.

3.5. Advanced source models

ixpeobssim provides a comprehensive set of high-level interfaces to facilitate the coding of complex source models. Dedicated Python decorators allow to compose spectral models with time- or phase-dependent parameters, e.g., a power law where the normalization and/or the spectral index are functions of time or pulse phase. Arbitrary *XSPEC* spectral models can be automatically wrapped into the proper function signature and fed natively into *xpobssim*, building on top of the *PyXSPEC* Python interface. In addition, a fully-fledged interface to OGIP-compliant libraries of tabular fitting models, with interpolation capabilities, is available—and a real-life example, interfacing to the magnetar models described in [15], is provided. Finally, complex polarization patterns for extended sources can be specified via a dedicated data structure encapsulating a collection of sky-maps of Stokes parameters in different energy layers, leveraging on the *SciPy* capability of interpolating on a regular grid in an arbitrary number of dimensions.

A comprehensive review of the *ixpeobssim* modeling facilities is beyond the scope of this paper, and the subject is largely covered in the documentation, to which the reader is referred.

3.6. Good time intervals

IXPE operates in a approximately circular low Earth orbit at 601.1 km altitude and 0.23° inclination. *ixpeobssim* captures the main features of the motion of the spacecraft around the Earth by means of a representative two-line element (TLE) set based on observations of the spacecraft taken soon after launch

There are three main ways in which the spacecraft orbit needs to be accounted for in simulated data: the Earth can obstruct the line of sight between the spacecraft and observation target; no observations can take place while the spacecraft is above the South Atlantic Anomaly (SAA); and certain celestial positions are only available for observations at certain times of the year due to Sun angle constraints. These values can be of crucial importance when trying to coordinate observations with other telescopes or accounting for the expected gaps in coverage for a time or pulse phase-dependent polarimetry signal. We use the *Skyfield* [16] package to geo-locate the ground footprint of the spacecraft as a function of time – which, in turn, enables all three of these effects to be incorporated into the production of Good Time Intervals (GTI) for simulated observations. Although they are not identical to those expected in a real observation, they are statistically representative of the latter.

3.7. Dithering and pointing history

As explained in [3], the IXPE focal plane detectors feature systematic deviations from a flat azimuthal response to unpolarized radiation, characterized by variations over small spatial scales, that need to be accounted for to reach the design polarization sensitivity. In order to average out this spurious modulation over the detector surface and make the correction practically viable [4], the observatory is dithered around the pointing direction. The dithering pattern has the basic form

$$\begin{cases} \delta x = a\cos(\omega_a t)\,\cos(\omega_x t) \\ \delta y = a\sin(\omega_a t)\,\sin(\omega_y t) \end{cases} \tag{9}$$

with a default amplitude of $a = 1.6$ arcmin and the three periods corresponding to the angular pulsations ω_a, ω_x and ω_y being 907 s, 101 s and 449 s, respectively (see Figure 4).

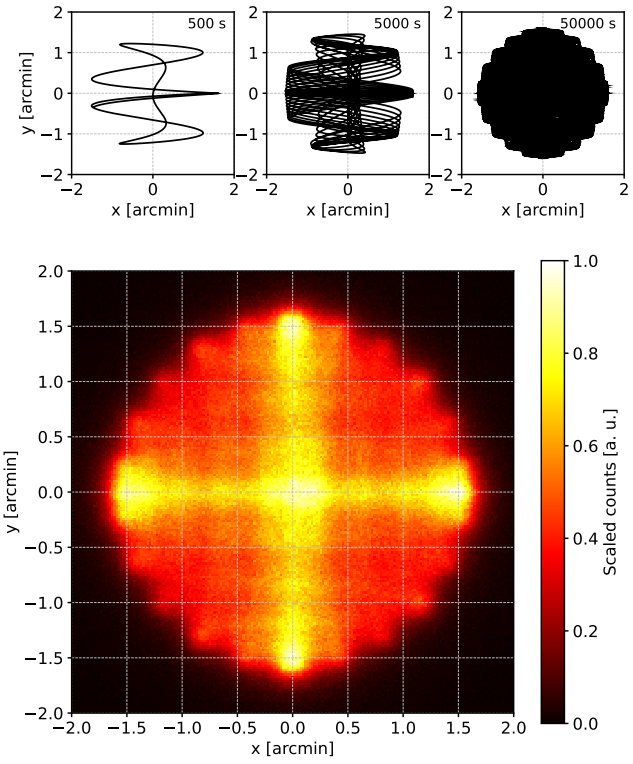

Figure 4: Representation of the default dithering pattern. The three small panels on the top represent the dithering path around the target position for 500, 5000 and 50000 s, while the histogram on the bottom represent the normalized counts after the convolution with the PSF of the telescope.

While the effect of the dithering is removed in the ground processing pipeline through the knowledge of the aspect solution, and does not affect the source image in sky coordinates, the specifics of the observation strategy need to be captured by the simulation in order to reproduce the morphology of the energy flux in detector coordinates (which is in turn relevant for

some instrumental effects) and for a correct calculation of the exposure. *ixpeobssim* keeps track of the effect of the dithering and stores the pointing history, sampled on a fixed-step, user-selectable, time grid in the (optional) `SC_DATA` extension.

4. Analysis Tools

ixpeobssim comes with a set of facilities for binning event lists in several different flavors. From an architectural standpoint, each binning algorithm comes with its own interface classes for output (i.e., creating binned files from event lists) and input (i.e., reading, visualizing and manipulating binned FITS files).

4.1. Basic polarization analysis

The simplest approach that *ixpeobssim* provides for polarization analysis is largely borrowed from [5]. More specifically, for each event we define the three additive reconstructed quantities

$$\begin{cases} \tilde{i}_i = \dfrac{w_i}{A_{\rm eff}(E_i)} \\ \tilde{q}_i = \dfrac{w_i q_i}{A_{\rm eff}(E_i)\mu(E_i)} \\ \tilde{u}_i = \dfrac{w_i u_i}{A_{\rm eff}(E_i)\mu(E_i)}, \end{cases} \tag{10}$$

where w_i represents the (optional) event weights introduced in section 1. The on-axis effective area term in equation (10) acts as an acceptance correction guaranteeing that the relevant quantities are summed over a proxy of the input source spectrum, as opposed to the measured count spectrum; note that q_i and u_i need to be divided by the proper modulation factor to transform the detector modulation into the actual polarization of the source.

We emphasize that the effective area and modulation factor in equation (10) are calculated by default at the measured energy, i.e., the effect of the energy dispersion is neglected. Experience shows that the effect is generally small, but *ixpeobssim* allows to verify it on a case-by-case basis, by using the ground truth for reference.

The measured Stokes parameters over a generic subset $\mathcal{S}$ of the events (be that a specific energy range, or a spatial bin in sky coordinates), is obtained by simply summing the event-by-event quantities over $\mathcal{S}$. The polarization degree and angle can be recovered with the usual formulæ, and the formalism to propagate the statistical uncertainties is thoroughly described in [5]. *ixpeobssim* provides facilities to calculate the broadband polarization properties over an arbitrary energy binning, integrated over a given sub-region of the field of view, or binned in the sky, as illustrated in Figure 5.

Among the additional analysis tools that are impossible to cover in the limited scope of this paper, we mention in passing *xpstokesalign*, that allows to align the Stokes parameters, on an event-by-event basis, to a given polarization model, facilitating the search for large-scale polarization signatures (e.g., radial or tangential) in extended sources.

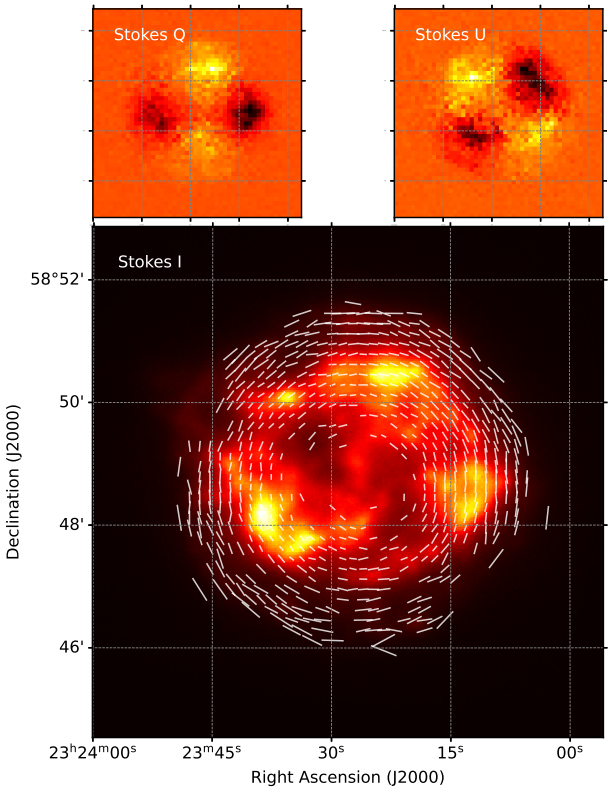

Figure 5: Polarization maps in the 2–8 keV energy band for a simulated, 2 Ms Cassiopea A observation, with a simple composite model including a thermal, un-polarized component, and a non-thermal, polarized component based on a simple geometric tangential pattern. The two panels on the top show the maps of the Q and U Stokes parameters in sky coordinates, while the main panel is a count map with the polarization direction for the pixels with a significance larger than 3σ overlaid. (The length of the arrows is proportional to the measured polarization degree.)

4.2. Spectro-polarimetric fitting

xpbin provides dedicated algorithms to create spectra of Stokes parameters, binned in pulse invariant channels, that can be readily used in conjunction with the standard fitting tools used by the X-ray community, e.g., *XSPEC* [7], *ThreeML* [17] and *Sherpa* [18], to perform spectro-polarimetric fits. More specifically, *xpbin* can write standard PHA type-I files (with specific header keywords for polarization analysis) containing the relevant quantities

$$\begin{cases} I_k = \varepsilon_k \displaystyle\sum_{\rm PI=k} w_i & \sigma_{I_k} = \varepsilon_k \sqrt{\displaystyle\sum_{\rm PI=k} w_i^2} \\ Q_k = \varepsilon_k \displaystyle\sum_{\rm PI=k} w_i q_i & \sigma_{Q_k} = \varepsilon_k \sqrt{\displaystyle\sum_{\rm PI=k} (w_i q_i)^2} \\ U_k = \varepsilon_k \displaystyle\sum_{\rm PI=k} w_i u_i & \sigma_{U_k} = \varepsilon_k \sqrt{\displaystyle\sum_{\rm PI=k} (w_i u_i)^2} \end{cases} \tag{11}$$

where the index i runs over the single events, k runs over the pulse invariant values, and ε_k is defined in terms of the total

observation livetime T as

$$\varepsilon_k = \frac{1}{T}\frac{\sum_{\mathrm{PI}=k} w_i}{\sum_{\mathrm{PI}=k} w_i^2}. \tag{12}$$

For an un-weighted analysis, i.e., when $w_i = 1$ for all the events, the above expressions reduce to the more familiar form

$$\begin{cases} I_k \xrightarrow[w_i=1]{} \dfrac{N_k}{T} & \sigma_{I_k} \xrightarrow[w_i=1]{} \dfrac{\sqrt{N_k}}{T} \\ Q_k \xrightarrow[w_i=1]{} \dfrac{1}{T}\displaystyle\sum_{\mathrm{PI}=k} q_i & \sigma_{Q_k} \xrightarrow[w_i=1]{} \dfrac{1}{T}\sqrt{\displaystyle\sum_{\mathrm{PI}=k} q_i^2} \\ U_k \xrightarrow[w_i=1]{} \dfrac{1}{T}\displaystyle\sum_{\mathrm{PI}=k} u_i & \sigma_{U_k} \xrightarrow[w_i=1]{} \dfrac{1}{T}\sqrt{\displaystyle\sum_{\mathrm{PI}=k} u_i^2}. \end{cases} \tag{13}$$

It is important to notice that the binned spectra follow a pure counting statistics only for the I Stokes parameter in the unweighted case, and care should be taken into selecting the correct fitting statistics in the general case.

We also emphasize that I_k, Q_k and U_k are expressed in detector space, and the detector response is taken into account by setting the proper response matrix and ancillary response files—the effective area for the I and the modulation response function for Q and U. The *ixpeobssim* pseudo-CALDB provides response functions in both weighted and un-weighted fashion; a few simple, multiplicative polarimetric models are provided by HEASARC through the page hosting XSPEC additional models[4], and shipped with *ixpeobssim* for convenience.

4.3. Interface to XSPEC

Although the *xpbin* output can be used directly in *XSPEC* with the proper response files, *ixpeobssim* provides a lightweight Python wrapper, dubbed *xpxspec* that facilitates combined spectral and spectro-polarimetric fits using the full data set from the three IXPE detector units. Figure 6 shows an example of such a combined fit for a simulated point source with a power-law spectrum and a constant polarization degree and angle, displayed using the *ixpeobssim* visualization facilities.

4.4. Analysis pipelines

As mentioned in the previous sections, one of the *ixpeobssim* design goals since the very beginning was to allow the user to develop simulation and analysis pipelines with minimal effort. To this end, every single *ixpeobssim* application is wrapped into a dedicated, top-level module so that it can be effectively called from within a generic Python script with the exact same keyword arguments that one would pass to the command line version of the same script. These wrappers typically return the list of all the files that the function call has created, which makes it very easy to chain application calls one after the other. The user is referred to the documentation for more information on this functionality, that we deem as one of the most powerful of the entire framework.

[4] `https://heasarc.gsfc.nasa.gov/docs/xanadu/xspec/newmodels.html`

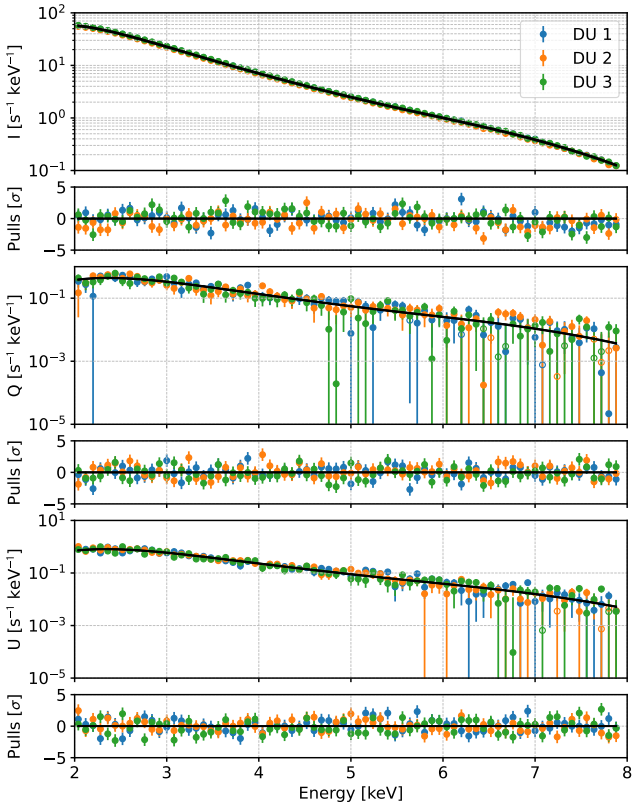

Figure 6: Spectro-polarimetric fit in *XSPEC* to simulated data of a point source with a power-law spectrum and constant polarization degree and angle. This is a simultaneous, combined fit to 9 independent data sets (I, Q and U for each of the three detector units) using a `pollin * powerlaw` model. The empty markers in the Q and U spectra represent negative values, that could not otherwise be rendered in logarithmic scale.

5. Conclusions

Initially conceived as a limited simulation tool with polarimetric capabilities, *ixpeobssim* has been expanded over the years to include a number of advanced simulation and analysis facilities and support the preparation of the IXPE mission.

With the IXPE data approaching public release, we decided to change our development model and release the codebase under an OSI-approved license, with the twofold purpose of benefiting the community engaged in the data analysis and encourage reuse for future X-ray missions. The ability to easily run nearly-identical analysis pipelines (including up-to-date calibration products) to both simulated and real IXPE observations should greatly enhance the ability of the scientific community to interpret this new and complex frontier of X-ray observations.

Acknowledgements

The Italian contribution to the IXPE mission is supported by the Italian Space Agency (ASI) through the contract ASI-OHBI-2017-12-I.0, the agreements ASI-INAF-2017-12-H0 and ASI-INFN-2017.13-H0, and its Space Science Data Center (SSDC), and by the Istituto Nazionale di Astrofisica (INAF) and the Istituto Nazionale di Fisica Nucleare (INFN) in Italy.

This work was supported by the EU Horizon 2020 Research and Innovation Program under the Marie Sklodowska-Curie Grant Agreement 734303.

We acknowledge useful contributions by (and/or stimulating discussions with) a number of members of the IXPE Collaboration, that are too many to be listed here.

References

[1] Martin C. Weisskopf, Paolo Soffitta, Luca Baldini, Brian D. Ramsey, Stephen L. O'Dell, Roger W. Romani, Giorgio Matt, William D. Deininger, Wayne H. Baumgartner, Ronaldo Bellazzini, Enrico Costa, Jeffery J. Kolodziejczak, Luca Latronico, Herman L. Marshall, Fabio Muleri, Stephen D. Bongiorno, Allyn Tennant, Niccolo Bucciantini, Michal Dovciak, Frederic Marin, Alan Marscher, Juri Poutanen, Pat Slane, Roberto Turolla, William Kalinowski, Alessandro Di Marco, Sergio Fabiani, Massimo Minuti, Fabio La Monaca, Michele Pinchera, John Rankin, Carmelo Sgro', Alessio Trois, Fei Xie, Cheryl Alexander, D. Zachery Allen, Fabrizio Amici, Jason Andersen, Angelo Antonelli, Spencer Antoniak, Primo Attina', Mattia Barbanera, Matteo Bachetti, Randy M. Baggett, Jeff Bladt, Alessandro Brez, Raffaella Bonino, Christopher Boree, Fabio Borotto, Shawn Breeding, Daniele Brienza, H. Kyle Bygott, Ciro Caporale, Claudia Cardelli, Rita Carpentiero, Simone Castellano, Marco Castronuovo, Luca Cavalli, Elisabetta Cavazzuti, Marco Ceccanti, Mauro Centrone, Saverio Citraro, Fabio D' Amico, Elisa D'Alba, Laura Di Gesu, Ettore Del Monte, Kurtis L. Dietz, Niccolo' Di Lalla, Giuseppe Di Persio, David Dolan, Immacolata Donnarumma, Yuri Evangelista, Kevin Ferrant, Riccardo Ferrazzoli, MacKenzie Ferrie, Joseph Footdale, Brent Forsyth, Michelle Foster, Benjamin Garelick, Shuichi Gunji, Eli Gurnee, Michael Head, Grant Hibbard, Samantha Johnson, Erik Kelly, Kiranmayee Kilaru, Carlo Lefevre, Shelley Le Roy, Pasqualino Loffredo, Paolo Lorenzi, Leonardo Lucchesi, Tyler Maddox, Guido Magazzu, Simone Maldera, Alberto Manfreda, Elio Mangraviti, Marco Marengo, Alessandra Marrocchesi, Francesco Massaro, David Mauger, Jeffrey McCracken, Michael McEachen, Rondal Mize, Paolo Mereu, Scott Mitchell, Ikuyuki Mitsuishi, Alfredo Morbidini, Federico Mosti, Hikmat Nasimi, Barbara Negri, Michela Negro, Toan Nguyen, Isaac Nitschke, Alessio Nuti, Mitch Onizuka, Chiara Oppedisano, Leonardo Orsini, Darren Osborne, Richard Pacheco, Alessandro Paggi, Will Painter, Steven D. Pavelitz, Christina Pentz, Raffaele Piazzolla, Matteo Perri, Melissa Pesce-Rollins, Colin Peterson, Maura Pilia, Alessandro Profeti, Simonetta Puccetti, Jaganathan Ranganathan, Ajay Ratheesh, Lee Reedy, Noah Root, Alda Rubini, Stephanie Ruswick, Javier Sanchez, Paolo Sarra, Francesco Santoli, Emanuele Scalise, Andrea Sciortino, Christopher Schroeder, Tim Seek, Kalie Sosdian, Gloria Spandre, Chet O. Speegle, Toru Tamagawa, Marcello Tardiola, Antonino Tobia, Nicholas E. Thomas, Robert Valerie, Marco Vimercati, Amy L. Walden, Bruce Weddendorf, Jeffrey Wedmore, David Welch, Davide Zanetti, and Francesco Zanetti. The imaging x-ray polarimetry explorer (ixpe): Pre-launch, 2021.

[2] Paolo Soffitta, Luca Baldini, Ronaldo Bellazzini, Enrico Costa, Luca Latronico, Fabio Muleri, Ettore Del Monte, Sergio Fabiani, Massimo Minuti, Michele Pinchera, Carmelo Sgro', Gloria Spandre, Alessio Trois, Fabrizio Amici, Hans Andersson, Primo Attina', Matteo Bachetti, Mattia Barbanera, Fabio Borotto, Alessandro Brez, Daniele Brienza, Ciro Caporale, Claudia Cardelli, Rita Carpentiero, Simone Castellano, Marco Castronuovo, Luca Cavalli, Elisabetta Cavazzuti, Marco Ceccanti, Mauro Centrone, Stefano Ciprini, Saverio Citraro, Fabio D'Amico, Elisa D'Alba, Sergio Di Cosimo, Niccolo' Di Lalla, Alessandro Di Marco, Giuseppe Di Persio, Immacolata Donnarumma, Yuri Evangelista, Riccardo Ferrazzoli, Asami Hayato, Takao Kitaguchi, Fabio La Monaca, Carlo Lefevre, Pasqualino Loffredo, Paolo Lorenzi, Leonardo Lucchesi, Carlo Magazzu, Simone Maldera, Alberto Manfreda, Elio Mangraviti, Marco Marengo, Giorgio Matt, Paolo Mereu, Alfredo Morbidini, Federico Mosti, Toshio Nakano, Hikmat Nasimi, Barbara Negri, Seppo Nenonen, Alessio Nuti, Leonardo Orsini, Matteo Perri, Melissa Pesce-Rollins, Raffaele Piazzolla, Maura Pilia, Alessandro Profeti, Simonetta Puccetti, John Rankin, Ajay Ratheesh, Alda Rubini, Francesco Santoli, Paolo Sarra, Emanuele Scalise, Andrea Sciortino, Toru Tamagawa, Marcello Tardiola, Antonino Tobia, Marco Vimercati, and Fei Xie. The instrument of the imaging x-ray polarimetry explorer. *The Astronomical Journal*, 162(5):208, oct 2021.

[3] L. Baldini, M. Barbanera, R. Bellazzini, R. Bonino, F. Borotto, A. Brez, C. Caporale, C. Cardelli, S. Castellano, M. Ceccanti, S. Citraro, N. Di Lalla, L. Latronico, L. Lucchesi, C. Magazzù, G. Magazzù, S. Maldera, A. Manfreda, M. Marengo, A. Marrocchesi, P. Mereu, M. Minuti, F. Mosti, H. Nasimi, A. Nuti, C. Oppedisano, L. Orsini, M. Pesce-Rollins, M. Pinchera, A. Profeti, C. Sgrò, G. Spandre, M. Tardiola, D. Zanetti, F. Amici, H. Andersson, P. Attinà, M. Bachetti, W. Baumgartner, D. Brienza, R. Carpentiero, M. Castronuovo, L. Cavalli, E. Cavazzuti, M. Centrone, E. Costa, E. D'Alba, F. D'Amico, E. Del Monte, S. Di Cosimo, A. Di Marco, G. Di Persio, I. Donnarumma, Y. Evangelista, S. Fabiani, R. Ferrazzoli, T. Kitaguchi, F. La Monaca, C. Lefevre, P. Loffredo, P. Lorenzi, E. Mangraviti, G. Matt, T. Meilahti, A. Morbidini, F. Muleri, T. Nakano, B. Negri, S. Nenonen, S.L. O'Dell, M. Perri, R. Piazzolla, S. Pieraccini, M. Pilia, S. Puccetti, B.D. Ramsey, J. Rankin, A. Ratheesh, A. Rubini, F. Santoli, P. Sarra, E. Scalise, A. Sciortino, P. Soffitta, T. Tamagawa, A.F. Tennant, A. Tobia, A. Trois, K. Uchiyama, M. Vimercati, M.C. Weisskopf, F. Xie, F. Zanetti, and Y. Zhou. Design, construction, and test of the gas pixel detectors for the ixpe mission. *Astroparticle Physics*, 133:102628, 2021.

[4] John Rankin, Fabio Muleri, Allyn F. Tennant, Matteo Bachetti, Enrico Costa, Alessandro Di Marco, Sergio Fabiani, Fabio La Monaca, Paolo Soffitta, Antonino Tobia, Alessio Trois, Fei Xie, Luca Baldini, Niccolò Di Lalla, Alberto Manfreda, Stephen L. O'Dell, Matteo Perri, Simonetta Puccetti, Brian D. Ramsey, Carmelo Sgrò, and Martin C. Weisskopf. An algorithm to calibrate and correct the response to unpolarized radiation of the x-ray polarimeter onboard IXPE. *The Astronomical Journal*, 163(2):39, jan 2022.

[5] F. Kislat, B. Clark, M. Beilicke, and H. Krawczynski. Analyzing the data from x-ray polarimeters with stokes parameters. *Astroparticle Physics*, 68:45–51, 2015.

[6] Alessandro Di Marco, Enrico Costa, Fabio Muleri, Paolo Soffitta, Sergio Fabiani, Fabio La Monaca, John Rankin, Fei Xie, Matteo Bachetti, Luca Baldini, Wayne Baumgartner, Ronaldo Bellazzini, Alessandro Brez, Simone Castellano, Ettore Del Monte, Niccolò Di Lalla, Riccardo Ferrazzoli, Luca Latronico, Simone Maldera, Alberto Manfreda, Stephen L. O'Dell, Matteo Perri, Melissa Pesce-Rollins, Simonetta Puccetti, Brian D. Ramsey, Ajay Ratheesh, Carmelo Sgrò, Gloria Spandre, Allyn F. Tennant, Antonino Tobia, Alessio Trois, and Martin C. Weisskopf. A weighted analysis to improve the x-ray polarization sensitivity of ixpe, 2022.

[7] K. A. Arnaud. XSPEC: The First Ten Years. In George H. Jacoby and Jeannette Barnes, editors, *Astronomical Data Analysis Software and Systems V*, volume 101 of *Astronomical Society of the Pacific Conference Series*, page 17, January 1996.

[8] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[9] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[10] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[11] Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis, A. Ginsburg, A. M. Price-Whelan, W. E. Kerzendorf, A. Conley, N. Crighton, K. Barbary, D. Muna,

H. Ferguson, F. Grollier, M. M. Parikh, P. H. Nair, H. M. Unther, C. Deil, J. Woillez, S. Conseil, R. Kramer, J. E. H. Turner, L. Singer, R. Fox, B. A. Weaver, V. Zabalza, Z. I. Edwards, K. Azalee Bostroem, D. J. Burke, A. R. Casey, S. M. Crawford, N. Dencheva, J. Ely, T. Jenness, K. Labrie, P. L. Lim, F. Pierfederici, A. Pontzen, A. Ptak, B. Refsdal, M. Servillat, and O. Streicher. Astropy: A community Python package for astronomy. *Astronomy & Astrophysics*, 558:A33, October 2013.

[12] Astropy Collaboration, A. M. Price-Whelan, B. M. Sipőcz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe, M. W. Craig, N. Dencheva, A. Ginsburg, J. T. Vand erPlas, L. D. Bradley, D. Pérez-Suárez, M. de Val-Borro, T. L. Aldcroft, K. L. Cruz, T. P. Robitaille, E. J. Tollerud, C. Ardelean, T. Babej, Y. P. Bach, M. Bachetti, A. V. Bakanov, S. P. Bamford, G. Barentsen, P. Barmby, A. Baumbach, K. L. Berry, F. Biscani, M. Boquien, K. A. Bostroem, L. G. Bouma, G. B. Brammer, E. M. Bray, H. Breytenbach, H. Buddelmeijer, D. J. Burke, G. Calderone, J. L. Cano Rodríguez, M. Cara, J. V. M. Cardoso, S. Cheedella, Y. Copin, L. Corrales, D. Crichton, D. D'Avella, C. Deil, É. Depagne, J. P. Dietrich, A. Donath, M. Droettboom, N. Earl, T. Erben, S. Fabbro, L. A. Ferreira, T. Finethy, R. T. Fox, L. H. Garrison, S. L. J. Gibbons, D. A. Goldstein, R. Gommers, J. P. Greco, P. Greenfield, A. M. Groener, F. Grollier, A. Hagen, P. Hirst, D. Homeier, A. J. Horton, G. Hosseinzadeh, L. Hu, J. S. Hunkeler, Ž. Ivezić, A. Jain, T. Jenness, G. Kanarek, S. Kendrew, N. S. Kern, W. E. Kerzendorf, A. Khvalko, J. King, D. Kirkby, A. M. Kulkarni, A. Kumar, A. Lee, D. Lenz, S. P. Littlefair, Z. Ma, D. M. Macleod, M. Mastropietro, C. McCully, S. Montagnac, B. M. Morris, M. Mueller, S. J. Mumford, D. Muna, N. A. Murphy, S. Nelson, G. H. Nguyen, J. P. Ninan, M. Nöthe, S. Ogaz, S. Oh, J. K. Parejko, N. Parley, S. Pascual, R. Patil, A. A. Patil, A. L. Plunkett, J. X. Prochaska, T. Rastogi, V. Reddy Janga, J. Sabater, P. Sakurikar, M. Seifert, L. E. Sherbert, H. Sherwood-Taylor, A. Y. Shih, J. Sick, M. T. Silbiger, S. Singanamalla, L. P. Singer, P. H. Sladen, K. A. Sooley, S. Sornarajah, O. Streicher, P. Teuben, S. W. Thomas, G. R. Tremblay, J. E. H. Turner, V. Terrón, M. H. van Kerkwijk, A. de la Vega, L. L. Watkins, B. A. Weaver, J. B. Whitmore, J. Woillez, V. Zabalza, and Astropy Contributors. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *The Astronomical Journal*, 156(3):123, September 2018.

[13] A. Tennant K. Dietz and Stephen Odell. Ixpe soc: Data format of level-1, level-2 and caldb products, 2022.

[14] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J.J. Gómez Cadenas, I. González, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F.W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampén, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M.G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J.P. Wellisch, T. Wenaus, D.C. Williams, D. Wright, T. Yamada, H. Yoshida, and D. Zschiesche. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.

[15] R Taverna, R Turolla, V Suleimanov, A Y Potekhin, and S Zane. X-ray spectra and polarization from magnetar candidates. *Monthly Notices of the Royal Astronomical Society*, 492(4):5057–5074, 01 2020.

[16] Brandon Rhodes. Skyfield: High precision research-grade positions for planets and Earth satellites generator, July 2019.

[17] Giacomo Vianello, Robert J. Lauer, Patrick Younk, Luigi Tibaldo, James M. Burgess, Hugo Ayala, Patrick Harding, Michelle Hui, Nicola Omodei, and Hao Zhou. The Multi-Mission Maximum Likelihood framework (3ML). *arXiv e-prints*, page arXiv:1507.08343, July 2015.

[18] Peter Freeman, Stephen Doe, and Aneta Siemiginowska. Sherpa: a mission-independent data analysis application. In Jean-Luc Starck and Fionn D. Murtagh, editors, *Astronomical Data Analysis*, volume 4477 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 76–87, November 2001.