



DATA PIPELINE

JUDE (Jayant's UVIT Data Explorer) pipeline user manual

P. T. RAHNA^{1,2,*} , JAYANT MURTHY² and MARGARITA SAFONOVA²

¹CAS Key Laboratory for Research in Galaxies and Cosmology, Shanghai Astronomical Observatory, Shanghai 200030, China.

²Indian Institute of Astrophysics, Bengaluru 560 034, India.

*Corresponding Author. E-mail: 7rehanrenzin@gmail.com

MS received 28 October 2020; accepted 13 December 2020

Abstract. We have written a reference manual to use *JUDE* (Jayant's UVIT Data Explorer) data pipeline software for processing and reducing the Ultraviolet Imaging Telescope (UVIT) Level 1 data into event lists and images—Level 2 data. The *JUDE* pipeline is written in the GNU Data Language (GDL) and released as an open-source which may be freely used and modified. GDL was chosen because it is an interpreted language allowing interactive analysis of data; thus in the pipeline, each step can be checked and run interactively. This manual is intended as a guide to data reduction and calibration for the users of the UVIT data.

Keywords. Ultraviolet—data reduction—pipeline.

1. Introduction

This manual describes the *JUDE* software system for processing and understanding the Ultra-Violet Imaging Telescope (UVIT) data. UVIT was launched on 28 September 2015 on a PSLV-C30 into a 650 km, 6° inclination low Earth orbit by Indian Space Research Organization (ISRO), as part of the multi-wavelength Indian AstroSat mission. The UVIT has seen its first light on 30 November 2015 by observing the open cluster NGC188. UVIT operates in three channels: visible, near-ultraviolet (NUV), and far-ultraviolet (FUV), covering a broad range of UV to the visible spectral region from about 1250 to 5500 Å, dividing it into 15 narrow and broad wavelength bands.

The UVIT instrument has been continuously observing since 2015, and the data, including the PV (performance verification) phase observations used to characterize the instrument (Tandon *et al.* 2017b) with in-flight calibration and verification (Rahna *et al.* 2017; Tandon *et al.* 2017a), are being released to the observers. The Announcement of Opportunity (AO)

Cycle was initiated in April 2017 for the Indian astronomy community. Though at present (2020) only the FUV and VIS channels are operating (NUV channel has failed in March 2018), the archive is large and open to the public—more than 1000 datasets of UVIT data were made public. The data may be accessed at the AstroSat data archive¹ (Astrobrowse) at the Indian Space Science Data Centre (ISSDC), Bangalore. Further data will be released upon completion of the respective proprietary periods (12 months). AstroSat has now completed its 5 years of operation.

The official UVIT pipeline is developed and available for download at the Science Support Cell (<http://astrosat-ssc.iucaa.in>). It is designed to produce scientifically usable images from the raw data—Level 2 data, which are flat-field, distortion, and drift-corrected images with absolute calibration. Canadian UVIT pipeline called CCDLAB is also available for UVIT data reduction (Postma & Leahy 2017). However, the source codes of these two pipelines are proprietary and difficult to modify.

This article is part of the Special Issue on “AstroSat: Five Years in Orbit”.

¹https://astrobrowse.issdc.gov.in/astro_archive/archive/Home.jsp—one needs to register and create an account there.

We have written the software package *JUDE* (Jayant’s UVIT data Explorer) entirely in the GNU Data Language (GDL) (Coulais *et al.* 2010) to reduce Level 1 UVIT data into Level 2 image files and event files. *JUDE* is released under an Apache License², which may be freely used and modified. GDL was chosen for the development environment because it is an interpreted language which lends itself to the interactive analysis of data. This was invaluable in the development of the pipeline, where we could check each step and run commands interactively. GDL is an open-source version of the Interactive Data Language³ (IDL) and runs on a wide variety of systems, just as IDL. GDL will run most IDL programs without modification (and *vice versa*), allowing access to the rich library of utilities developed for IDL over the last four decades, thus easing the development of *JUDE*. We have tested *JUDE* using both GDL (version 0.9.6) and IDL with identical results on multiple operating systems. In the remainder of the manual, GDL and IDL may be freely interchanged. *JUDE* starts with the Level 1 data, provided by the Indian Space Science Data Centre (ISSDC), and produces photon lists and images suitable for the scientific analysis. *JUDE* is archived at the Astrophysics Source Code Library (Murthy *et al.* 2016), and the latest version is available on GitHub at: <https://github.com/jaymurthy/JUDE>.

We will continue to update the pipeline as we find errors and also welcome feature requests to improve the scientific utility of the programs.⁴

1.1 UVIT instrument details

The UVIT is one of the five payloads on AstroSat spacecraft. It consists of two identical 37.5 cm Ritchey–Chrétien telescopes, one is reserved for FUV channel, and the second one sharing NUV and visible (VIS) channels through a beam splitter (Table 1). The VIS channel is not intended for science purposes and is used solely to track the spacecraft (s/c) motion. Each channel is equipped with a 5-filter filter wheel, providing spectral coverage in several passbands from the FUV to the visible. The basic properties of UVIT filters are given in Table 2. The

²<http://www.apache.org/licenses/LICENSE-2.0>.

³<https://idlastro.gsfc.nasa.gov/>.

⁴As was communicated to us by a user, the use of a /notime keyword in the *interactive.com* code was giving an error message about exposure times. We have corrected this in the code; Nov. 2020.

effective bandwidth $\Delta\lambda$ in Table 2 is the integral of the normalized effective area, and λ_0 is the central (or ‘mean’) source-independent wavelength, weighted by the normalized effective areas; these quantities were calculated from the ground calibration values (Rahna *et al.* 2017). See Tandon *et al.* (2020) for the latest updated corrections to the ground calibration values.

1.2 Status of *JUDE*

- The development of *JUDE* began in June 2016.
- The production release of *JUDE* is June 2017.
- *JUDE* update: June 2019.
- The last update of *JUDE*: November 2020.

2. UVIT data definitions

2.1 Level 0 data

Level 0 data is the raw data from the AstroSat spacecraft. This data is sent to the AstroSat Data Center at the ISSDC, where the data is separated by instrument and written into Level 1 data.

2.2 Level 1 data

The Level 1 data from ISSDC is sent to the payload operations center (POC) at the Indian Institute of Astrophysics (IIA), Bengaluru. The Level 1 data are distributed as a single zipped archive for each observation. The files within the archive are in a number of sub-directories organised by the orbit number and the type of file, all under a single top-level directory. All data files are FITS binary tables. More details about the data are described in Murthy *et al.* (2017).

2.3 Level 2 data

Once the refined coordinate information is brought into the FITS binary extension tables, it is ready to be converted to an image of processed data from a single scan of the sky. These images are termed Level 2 data. This is the final data product from the pipeline software. Level 2 data are FITS images, readable by any of the standard astronomical data processing packages.

Table 1. Summary of characteristics of UVIT instrument

Parameter	Value
Type ^a	Ritchey–Chrétien
Diameter ^a	37.5 cm
Field of View ^a	28′
Detector ^a	CMOS, 512 × 512
Mode of operation ^a	Photon counting mode (FUV and NUV)
Channels ^a	FUV (1300–1800 Å), NUV (2000–3000 Å)
Spatial resolution ^b	≲1.5″
Frame rate ^a	29 frames/s
Sensitivity ^a	20 AB mag. in FUV for 200 s
Time resolution ^c	<35 ms
Pixel scale	3.28″ (for 512 × 512 window size)
<i>JUDE</i> pipeline image format	4096 × 4096 pixels with 0.41″ pixel size

^aKumar *et al.* (2012); ^bRahna *et al.* (2017); ^cMurthy *et al.* (2017).

Table 2. Properties of different UVIT UV filters

Filter channel	Type	Filter name	Passband (nm)	Effective bandwidth $\Delta\lambda$ (nm)	Central λ_0 (nm)	CF ^a (erg cm ⁻² Å ⁻¹ cnt ⁻¹)	ZP ^b (mag)
FUV	CaF ₂ -1	F148W	125–179	44.1	150.94	3.8689e-15	17.73
	BaF ₂	F154W	133–183	37.8	154.96	4.2036e-15	17.58
	Sapphire	F169M	145–181	27.4	160.7	5.4399e-15	17.22
	Silica	F172M	160–179	13.13	170.3	1.4273e-14	16.05
NUV	Silica	N242W	194–304	76.9	241.8	1.9459e-16	19.95
	NUV15	N219M	190–240	27.1	218.5	5.8360e-15	16.48
	NUV13	N245M	220–265	28.17	243.6	1.0327e-15	18.12
	NUVB4	N263M	220–265	28.23	262.8	6.9611e-16	18.39
	NUVN2	N279N	273–288	8.95	279.0	2.7988e-15	16.75

^aConversion factors from Rahna *et al.* (2017). ^bZero-point magnitudes derived from conversion factors.

3. Installing and setting up the *JUDE* pipeline

- (1) *Download and install GDL/IDL*: *JUDE* runs under either IDL or GDL. You have to download and install either of them to run *JUDE* software system.
- (2) *Download the IDL libraries*: *JUDE* uses many library routines from the IDL Astronomy Users’ Library⁵ and from the Markwardt IDL Library⁶ that can be downloaded and installed. These routines should be placed in one of the directories pointed to by the !PATH variable. Our recommendation is to create a separate directory for each library and unzip the files into that directory. Update the !PATH system variable to include these directories.

⁵<https://idlastro.gsfc.nasa.gov/ftp/astron.zip>.

⁶<http://www.physics.wisc.edu/~craig/idl/down/cmtotal.zip>.

- (3) *Download *JUDE* programs from GitHub*.⁷ Add the *JUDE* library to the IDL path i.e.: !path = path to jude/: + !path
- (4) *Download Level 1 data of your target from ISSDC Astrobrowse*. Unzip and save the downloaded raw data in a folder called Level1(dname). Note that *JUDE* will find all relevant files under the dname directory. The individual files may be gzipped to save considerable space.

4. Running *JUDE* pipeline

4.1 Automatic mode

JUDE comprises a number of individual routines that together can be run as a pipeline. A program

⁷<https://github.com/jaymurthy/JUDE>.

(*process_uvfit.com*) is included in the *JUDE* distribution, which may be modified for local use and can be run using the syntax:

Step 1. @process_uvfit.com (from IDL) or idl process_uvfit.com (from terminal). This program chains together the following commands (Note that, if desired, these statements may be entered in sequence at the GDL prompt). You may have to set the path before running *process_uvfit.com*:

(1) Setting path (as per above setup):

- `!path = "< path to jude folder > :"+!path;`
Add *JUDE* routines to path.
- `!path = "< path to cmlib folder > :"+ !path;`
Add Markwardt routines.
- `!path = "< path to idl libraries > :"+!path;`
Add IDLASTRO routines.
- `dname = "< path to level1 folder >";`
Location of Level 1 data.

(2) Produce Level 2 data:

- `jude_driver_vis, dname;`
- `jude_driver_uv, dname,/nuv,/notime`
- `jude_driver_uv, dname,/fuv,/notime`
- `jude_verify_files, dname`
- `jude_uv_cleanup, /nuv`
- `jude_uv_cleanup, /fuv`

Description: *jude_driver_vis* will process the visible files (VIS) and *jude_driver_uv* will process the UV files to produce event lists and images. When running *jude_uv_driver* for the first time, it is faster to use the `/notime` keyword. The time per pixel will be approximate at the edges. Program *jude_verify* was introduced because there are crashes in GDL for long runs, and it will go through each of the files and ensure that they are, at least, readable. *jude_uv_cleanup* will process the event lists and images, merge the data, and register the images. Finally, it creates Level 2 data.

There are random crashes caused by memory issues which can affect datasets with many files. These errors may occur in any module and at different locations. It was found that 4 GB RAM is enough but, obviously, more RAM is better. In most cases, the program may be restarted and will pick up from where it was left off, skipping files already processed. If the memory problem occurs at a critical point, it is possible that the FITS file may not be properly written. We have added a program (*jude_verify_files*) which should check the integrity of each FITS file and delete it if there is a problem. If this does not work, it would be best to

delete any suspect files and start *process_uvfit.com* again. Files already completed will be skipped. If the verification (from *jude_verify_files*) is successful, a file named *jude_VERIFY_FILES_DONE* will be created. There will be no further Level 1 processing if this file exists. It should be deleted if processing from the scratch is really desired.

4.2 Interactive mode

After the initial run of *JUDE* using *process_uvfit.com*, you can run it interactively with *interactive.com* using the syntax:

Step 2. @interactive.com (from IDL) or idl interactive.com (from terminal). It will open a window with the image and ask a few questions:

- Enter new image scaling (Enter if ok)?
If image is too messy or crowded, you can change the contrast (scale), e.g. 0.001.
- Parameter to change? `-1` to continue, `-2` to exit, `-3` to debug:
It will display default parameters on the screen: The explanation of these parameters is given in Table C.5. in the appendix to the *JUDE* paper (Murthy *et al.* 2016). You can change the parameters based on your data by entering the corresponding parameter number and change the values. If you don't want to change the default parameters, type `-1` and hit Enter.
- Run centroid (y/n/v/r)? (y – yes, n – no, v to use VIS, r to reset offsets):
 - If you want it to run automatically using *JUDE* centroid program, press 'y' and Enter.
 - If you want to correct for s/c motion using VIS data interactively, press 'v' and Enter.
 - If you choose 'n', it will run with default offset values and gives you the final output with default s/c correction.
 - If you press 'r', it will reset offsets (make them full nulls).
- Run using *JUDE* centroid: First, it will ask to change the image scaling, then it will ask to choose the individual star from the displayed image. Select the isolated bright star in the field within the binned image by clicking the cursor

on the display image. It is better to avoid stars on the edges of the field.

- ‘Enter new box size (or press Enter):’ You can specify a box size here, and the program will perform the s/c correction by following the selected star within the box size in each set of frames, and display the co-added image. It will also display the PSF of the image.
- ‘Write files out (this may take some time)?’: If you are satisfied with the PSF and the image, you can write the FITS file by pressing y to this question.
- ‘Do you want to run with different parameters?’: If you want to improve the images, you can run the code again by entering y, otherwise it will automatically go to the next image. Follow the same procedure.

After finishing all NUV, it automatically goes to FUV. Follow the same procedure.

Important note: When you run *interactive.com*, you can use `/notime` keyword with *jude_interactive.pro* in *interactive.com* to run fast. But, in order to get the correct exposure time and flux (CPS) in the final output, remove the `/notime` keyword from *interactive.com* and run it again, or run *jude_apply_time.pro* afterwards for all the files.

Calling sequence: *jude_apply_time*, *L2_data_file*, *uv_base_dir*, *params = params*
where *L2_data_file* is the directory of Level 2 events list folder (`/events`) and *uv_base_dir* is The top level UV directory (`/fuv` or `/nuv`).

5. Data products

- *Events list:* A FITS binary table with N extensions, where N is the number of frames in the data. Each extension has a list of all the photons detected in that frame. The format of the file is described here: in the first stage of processing, one event file will be created for each Level 1 data file. Duplicate files will be removed in the second stage (*jude_uv_cleanup*) and the data will be merged if they are continuous. There will thus be fewer event files than Level 1 data files. *Display events:* *make_movie.pro* will provide a movie of the frames from the events files, which can be run using *make_movie*, *events_file*.
- *Images:* A FITS image file will be created for every event file. Each frame will be shifted and

added to correct for spacecraft motion.

The first extension in the FITS image file will be the flux in counts/second (CPS).

The second extension in the FITS image file will be the exposure time for each pixel in seconds. If `/notime` keyword was set, then the array contains the total number of frames. The final image from *JUDE* will be 4096×4096 pixels ($28' \times 28'$) with the pixel size of $0.41''$. The total exposure time will be given in the fits header of each images.

Changing exposure time: The exposure time of a final image can be changed by changing parameters “`minframe`” and “`maxframe`” when running *interactive.com*. This can be helpful if the user is looking for flux variability at different time scales.

- *Diagnostic files:* Corresponding quick-look PNG image will be created for every event file. This image will additionally contain the following plots: a plot of the data quality index (DQI) for each frame, a histogram of the number of counts per frame, and a plot of spacecraft motion derived from calculating shifts between the frames in both X and Y . For details on DQI and calculation of shifts see Murthy *et al.* (2017).
- *CSV files:*
final_obslog.csv: Observation log for Level 2 data;
obs_times.csv: Observation log for Level 1 data with duplications;
merge.csv: Log of files merged to produce the final Level 2 data. After running *interactive.com*, *jude_obs_log.pro* can be run to create the observation log of final images.
Calling sequence: *jude_obs_log*, *data_dir*, *output_file*, *params*.

6. Verifying the output

In principle, *JUDE* should automatically produce usable science products: photon lists and images. However, there are times when the registration fails, and it is best to verify the images manually. Program *interactive.com* will run through each Level 2 file and give options for reprocessing the data, if desired. The main purpose of the program is to run *jude_interactive.pro* for each event file. As a general rule, there are three parameters to check:

- The integration time of the image should be close to that of events file. If it is much less than the actual time observed, it is worth checking why.
- The PSF depends on the quality of the correction for the *s/c* motion, and is typically 1.2–1.5 arcseconds. If the PSF is much greater than that, it means that automatic registration failed.
- The appearance of the image depends on the *s/c* motion, and this may be seen as tails in the sources.

jude_checkpsf.pro can be used to check for the PSF of a star in the field. The program gives PSF in *x* and *y* direction (unit of both pixels and arcsec) and also displays 3D surface profile and a contour plot of the star.

Calling sequence: jude_checkpsf, im, res
where *im* is the Level2 image file of UVIT, and *res* is the resolution (number of sub-pixels in one physical pixel) of the image. By default, *JUDE* produce images with 4096×4096 (res=8) pixels.

7. Data registration

Data registration corrects for spacecraft motion. The first estimate comes from the VIS images. The program *jude_match_vis_offsets.pro* only samples every 1 sec in time, and thus the derived spatial resolution is poorer. The second estimate comes from self registration—program *jude_centroid.pro*. As long as there is a bright star in the UV channel, this will give better resolution than getting offsets from the VIS channel. Most fields have stars that are bright enough to follow along. If the automated registration fails, there are several steps to follow:

- Increase the number of bins (the amount of time) to be binned over. The default is 20 frames (0.6 s). The box size usually has to be increased because the *s/c* motion is larger in longer times.
- Decrease the resolution, which effectively increases the S/N per pixel. It is usually easier to find bright sources suitable for registration in the NUV channel. We have written *jude_match_nuv_offsets* code to match the NUV offsets to the FUV channel.

8. Astrometric corrections

We have found that the astrometric information provided by the *s/c* star sensor is not accurate enough for good astrometric calibration of the UVIT images, and we have implemented several alternatives.

8.1 *Astrometry.net*

This is our recommendation and it gives excellent solutions for most NUV fields. Though it is less likely to work well for the FUV files because the stellar magnitudes are much more different from the optical. There are several different ways to use *Astrometry.net*.

8.1.1 *Directly on the website*

- Submit images to the website (<http://nova.astrometry.net>). Anonymous uploads (without signing in) are allowed but then the data becomes public. Creating an account allows more customization.
- The solution will be faster if we restrict the field size to the UVIT FoV.
- The FITS file with the astrometric solution will be on the “SUCCESS” page and is named new-image.fits.
- Note that this file will take the first extension from the original file (the data) and will not include the exposure time per pixel.
- We have written a program (*jude_copy_astrometry_hdr.pro*) to merge the astrometry from the new-image.fits into the original file.
Calling sequence: jude_copy_astrometry_hdr, astrometric, l2_file

8.1.2 *Python script to upload data files to Astrometry.net*

- Python and *client.py* are required to run this. Typing `python client.py` will print the arguments to the program.
- A key (API_KEY) to *Astrometry.net* (obtained through creating an account) is also required.
- *jude_call_client.py* is the interface to *client.py*. The key may be stored in the system variable AN_API_KEY; entered as a keyword, or entered when requested. The path to *client.py* should be changed in the program (if it does not exist, the user is prompted for input). The *client.py* is called with preselected parameters. The Level 2 image is uploaded to the *Astrometry.net* website, processed online, and then downloaded. Finally, We update the astrometry in the Level 2 file. The call would be (with default parameters) *jude_call_client.py,inp_dir,out_dir*. *inp_dir* is the directory containing

image files to be corrected and *out_dir* is the directory to put files from *astrometry.net*.

8.2 Independent astrometric corrections

There are times when the astrometric correction fails, more often in the case of the FUV where there are not enough recognizable stars. Program *jude_fix_astrometry* takes an image with known astrometry (typically the NUV image) and compares it with the test image. We have tried to automate the procedure as much as possible, but user intervention is necessary at times. At least two points are required for an astrometric solution. If two or three points have been defined, the code *starast.pro* is used. If 6 or more points have been defined, the code *solve_astro.pro* is used. In case not enough points are found, they can be added by hand to a reference file. The name of the reference file is the same as the image file with *.FITS.gz* replaced by *.ref*. The reference file has 4 lines, each with at least two elements. The lines contain the *x* and *y* positions (in pixels) of each star with its RA and DEC (in degrees). The order of the lines is unimportant, but the variable names (*newxp*, *newyp*, *newra*, *newdec*) are required, e.g.,

```
newxp = [3494.0319, 3099.7382, 1987.0005],
newyp = [1579.9278, 1483.6838, 2929.903],
newra = [9.1011347, 9.1398875, 9.4078713],
newdec = [39.927001, 39.963891, 39.913863].
```

Calling sequence: *jude_fix_astrometry*, *new_file*, *ref_file*, where *new_file* is the Level 2 image file to be corrected astrometrically, and *ref_file* is file with accurate astrometry.

8.3 Coordinates for every photon

jude_apply_astrometry.pro will take an astrometrically corrected image file and apply the astrometry back to the event list. The *x* and *y* of each photon (in the detector plane) will have an associated RA and DEC.

9. Co-addition of images

The final step is the co-addition of the images from different orbits. We used *jude_coadd.pro* routine to produce final co-added image. Program

jude_coadd.pro will take a set of images, place them on the same coordinate frame, and add together. The images must have astrometric information and the quality of the addition will depend on the correction. The result is an exposure time-weighted sum of the inputs. Note that the output pixel is larger than the input pixel, so the counts/pixel will be necessarily larger.

Calling sequence: *jude_coadd*, *images_dir*, *out_file*, *ra_cent*, *dec_cent*, *fov*, *pixel_size*, where *ra_cent* and *dec_cent* are coordinates of the image center, *fov* is the image field of view in degrees, and *pixel_size* is in arcsec.

10. Flux calibration

The flux calibration of UVIT is described in Rahna *et al.* (2017). The flux unit of the final image derived from *JUDE* is counts/second (CPS). CPS can be converted into physical flux in $\text{erg s}^{-1} \text{cm}^{-2} \text{\AA}^{-1}$ by multiplying by the conversion factor CF ($\text{erg s}^{-1} \text{cm}^{-2} \text{\AA}^{-1} \text{cps}^{-1}$), given in Table 2 (Rahna *et al.* 2017):

$$F(\lambda) = \text{CF} \times \text{CPS}. \quad (1)$$

Errors in CPS can be calculated from the square root of the total number of counts divided by square root of the total exposure time,

$$\text{CPS}_{\text{err}} = \frac{\sqrt{\text{CPS}}}{\sqrt{\text{exposure time}}}. \quad (2)$$

To obtain errors in flux $F(\lambda)$, multiply CPS errors by the corresponding conversion factors.

11. Saturation correction

At high count rates, intensified CMOS detectors are subject to non-linearity and saturation. Corrections for saturated bright stars in the UVIT field are discussed in Section 3.2.3 of Rahna *et al.* (2017). We modeled the non-linearity using the formulation of Kuin and Rosen (2008). The observed UVIT counts are non-linear after 15 CPS (9.7% roll-off) and saturated count rates above this value (till 29 CPS) can be corrected using empirical relation,

$$C_{\text{obs}} = 29 \times (1 - e^{(-\alpha C_{\text{inc}}/29)}), \quad (3)$$

where $\alpha = 1.24$ (determined empirically), C_{inc} is the number of events incident on the detector, C_{obs} is the number of events detected, and there are 29 frames in

a second. After 29 CPS, the counts saturate and the true value cannot be recovered.

12. Conclusion

We have described here how to install and operate the independent UVIT data pipeline *JUDE*. The pipeline is free and open-source and consists of modules where users can control and change the parameters. We have successfully used *JUDE* in the last three years; first to independently characterize the in-flight performance of the UVIT (Rahna *et al.* 2017) and, secondly, in our scientific programs, of which several are completed (Rahna *et al.* 2018; Kameswara Rao *et al.* 2018a, b; Rubinur *et al.* 2020) and few are still running on (Safonova *et al.* 2020; Rahna *et al.* 2021; Yadav *et al.* 2021).

Attribution and Acknowledgements

If you use this software, please cite Murthy *et al.* (2017). We have used GDL very heavily and here is the attribution for that: Coulais *et al.* (2010). The IDL Astronomy User's Library reference is Landsman (1995), the Markwardt IDL Library: Markwardt (2009), and the Astronomy.net reference is Lang *et al.* (2010). This publication uses the data from the AstroSat mission of the Indian Space Research Organisation (ISRO), archived at the Indian Space Science Data Centre (ISSDC). The UVIT is built in collaboration between IIA, IUCAA, TIFT, ISRO and CSA. The authors gratefully thank all the members of various teams of UVIT project. RPT acknowledge IIA (Indian Institute of Astrophysics, India) and SHAO (Shanghai Astronomical Observatory, China) for providing the computational facilities. This research has been supported by the Department of Science and Technology under Grant No. EMR/2016/00145 to IIA.

References

Coulais A., Schellens M., Gales J. *et al.* 2010, in Mizomoto Y., Morita K.-I., Ohishi M., eds, Astronomical Society of

- the Pacific Conference Series, Vol. 434, Astronomical Data Analysis Software and Systems XIX, 187
- Kameswara Rao N., De Marco O., Krishna S. *et al.* 2018a, A&A, 620, A138
- Kameswara Rao N., Sutaria F., Murthy J. *et al.* 2018b, A&A, 609, L1
- Kuin N. P. M., Rosen S. R. 2008, MNRAS, 383, 383
- Kumar A., Ghosh S. K., Hutchings J. *et al.* 2012, in Space Telescopes and Instrumentation 2012: Ultraviolet to Gamma Ray, Vol. 8443, 84431N
- Landsman W. B. 1995, in Shaw R. A., Payne H. E., Hayes J. J. E., eds, Astronomical Society of the Pacific Conference Series, Vol. 77, Astronomical Data Analysis Software and Systems IV, 437
- Lang D., Hogg D. W., Mierle K., Blanton M., Roweis S. 2010, AJ, 139, 1782
- Markwardt C. B. 2009, in Bohlender D. A., Durand D., Dowler P., eds, Astronomical Society of the Pacific Conference Series, Vol. 411, Astronomical Data Analysis Software and Systems XVIII, 251
- Murthy J., Rahna P. T., Safonova M. *et al.* 2016, *JUDE: An Ultraviolet Imaging Telescope pipeline*, Astrophysics Source Code Library, ascl:1607.007
- Murthy J., Rahna P. T., Sutaria F. *et al.* 2017, *Astronomy and Computing*, 20, 120
- Postma J. E., Leahy D. 2017, PASP, 129, 115002
- Rahna P. T., Das M., Murthy J., Gudennavar S. B., Bubbly S. G. 2018, MNRAS, 481, 1212
- Rahna P. T., Murthy J., Safonova M. *et al.* 2017, MNRAS, 471, 3028
- Rahna P. T., Zhenya Z., Das M., Murthy J. 2021, FUV excess emission in the outer disk of SO/ETS galaxies, manuscript in preparation
- Rubinur K., Kharb P., Das M. *et al.* 2020, MNRAS, 500, 3908
- Safonova M., Kashyap U., Rahna P. T. *et al.* 2020, Testing the nature of the ULX by AstroSat: UV–X-ray variability in ultra-luminous X-ray source Holmberg II X-1, manuscript in preparation
- Tandon S. N., Subramaniam A., Girish V. *et al.* 2017a, E-print arXiv:1705.03715, arXiv:1705.03715
- Tandon S. N., Hutchings J. B., Ghosh S. K. *et al.* 2017b, JAA, 38, 28
- Tandon S. N., Postma J., Joseph P. *et al.* 2020, AJ, 159, 158
- Yadav J., Das M., Patra N. *et al.* 2021, Comparing the inner and outer star forming complexes in the nearby Spiral galaxies Ngc 628, Ngc 5457 and Ngc 6946 using UV observations, manuscript in preparation