

MULTI-WAVELENGTH SPECTRAL FITTING AND VARIABILITY STUDIES OF SUPERSOFT X-RAY SOURCES

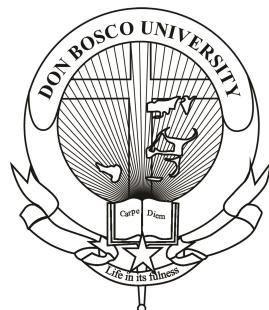
By

PARAG BHATTACHARYA

DEPARTMENT OF PHYSICS

A THESIS
SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN PHYSICS

To



ASSAM DON BOSCO UNIVERSITY

TAPESIA GARDENS, TAPESIA,

SONAPUR – 782 402

ASSAM – INDIA

SEPTEMBER, 2024

MULTI-WAVELENGTH SPECTRAL FITTING AND VARIABILITY STUDIES OF SUPERSOFT X-RAY SOURCES

By

PARAG BHATTACHARYA

DEPARTMENT OF PHYSICS

A THESIS
SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN PHYSICS

To



ASSAM DON BOSCO UNIVERSITY

TAPESIA GARDENS, TAPESIA,

SONAPUR – 782 402

ASSAM – INDIA

DECLARATION

I, Parag Bhattacharya, hereby declare that the subject matter of this thesis is the record of the work done by me, that the content of this thesis did not form the basis of the award of any previous degree to me or to the best of my knowledge to anybody else, and that the thesis has not been submitted by me for any research degree in any other university or institute.

Signature

(Parag Bhattacharya)

Research Scholar

Signature

(Dr. Monmoyuri Baruah)

Supervisor

Signature

(Prof. Ranjeev Misra)

Co-Supervisor

Signature with date and seal

(Dr. Monmoyuri Baruah)

Head, Department of Physics

CERTIFICATE

This is to certify that the thesis entitled, “*Multi-wavelength Spectral Fitting and Variability Studies of Supersoft X-ray Sources*” being submitted by Mr. Parag Bhattacharya for the award of the degree of Doctor of Philosophy in Physics of the Assam Don Bosco University is the result of his investigation under my supervision. He has fulfilled all the requirements for the submission of the thesis under the Ph.D. ordinance of the Assam Don Bosco University, and thesis presented is worthy of being considered for the Ph.D. degree in Physics.

The work described in this thesis is original and has not been submitted for any other degree or diploma in this or any other university.

Signature

Dr. Monmoyuri Baruah
Supervisor

Signature

Prof. Ranjeev Misra
Co-Supervisor

Signature with date and seal

Dr. Monmoyuri Baruah
Head, Department of Physics

ACKNOWLEDGEMENT

I am deeply grateful to Rev. Fr. (Dr.) Stephen Mavely, the Founder Vice-Chancellor of Assam Don Bosco University, whose unwavering belief in me has been a source of immense encouragement. I owe him my sincere thanks for choosing to appoint me as an Assistant Professor in the Department of Physics over more qualified candidates. Throughout my association with ADBU, he has consistently taken a keen interest in the progress of my research, for which I am profoundly thankful.

I extend my heartfelt thanks to the Vice-Chancellor of Assam Don Bosco University, Rev. Fr. Jose Palely, for his promptness and support in ensuring that every process has been smooth and hassle-free during my journey.

I am equally thankful to Prof. Jai Narayan Vishwakarma, Director of Research at Assam Don Bosco University, for creating and sustaining a healthy research atmosphere at the university. His availability and willingness to address any concerns have been instrumental in my progress, and his appreciation for my efforts has meant a great deal to me.

My deepest gratitude goes to my research supervisor, Dr. Monmoyuri Baruah, whose commitment to my academic growth has been unmatched. She has always ensured that I have access to the necessary research facilities at IUCAA, Pune, while placing me on a path that both challenges and nurtures my potential.

I also wish to sincerely thank Prof. Ranjeev Misra, Scientist-G at IUCAA, Pune, and my research co-supervisor, for his invaluable guidance during my visits to Pune. His support and introductions to numerous experts have helped me overcome challenges and move forward in my research.

I will forever cherish the inspiration I have received from Dr. Basil Koikara, the Founder Registrar of Assam Don Bosco University, Rev. Fr. Joseph Nellanat, the Pro-Vice-Chancellor of Assam Don Bosco University, and Rev. Fr. Biju Michael, former Registrar of Assam Don Bosco University. Their leadership and dedication have left a lasting impact on me.

I am deeply thankful to my colleagues at ADBU for their constant encouragement and belief in me. A special mention goes to Dr. Samrat Dey, Dr. Ngangom Aomoa, Dr. Debajyoti Dutta, Mr. Dipjyoti Sarma, Dr. Kaustubh Bhattacharyya, Dr. Pratikshya Bezbaruah, Dr. Sumita K. Sharma, Dr. Aibanjop Pyngrope, Dr. Bikash Borah, Dr. Indrani Dakua, Dr. C.

Satyanaanda Singh, Dr. Yajnya Sapkota, Dr. Nazneen Hussain, Dr. Eliyash Ahmed, Ms. Nairika Deka, Mr. Ankon Borah, Mr. Deepjyoti Mahanta, and Mr. Mintu Hazarika. Your friendship and encouragement have played a crucial role in my journey.

I would also like to express my sincere gratitude to all the researchers and scholars whose work I have cited throughout this thesis. Their contributions have laid the foundation for my own research and have been a guiding light in shaping my understanding of this field. The collective wisdom embedded in their writings has inspired and enriched my work, and I am deeply indebted to them for their insights, methodologies, and discoveries, which have propelled the advancement of knowledge across generations.

I must also acknowledge all of my students over the past decade, whose curiosity and enthusiasm have been infectious and have continuously inspired me.

Finally, my deepest gratitude is reserved for my family, who hold the most special place in this journey. Their unwavering love, patience, and support have been the bedrock upon which this entire endeavour has rested. Their quiet resilience and faith in me have fuelled my determination. This thesis is as much a testament to their support as it is to my efforts.

In the unfolding of my academic pursuit, I often find myself contemplating upon the nature of π – the enigmatic constant that governs circles, yet never reveals its full form. Much like the arc of π , the path of knowledge curves toward deeper understanding without ever reaching a final resolution. Each decimal of π mirrors the layers of discovery within my research, wherein every insight leads to another, and certainty seems to lie just beyond the horizon. It is in this endless quest for completeness, knowing that some mysteries may remain forever irrational, that I have found meaning in both the struggle and the pursuit.

(Parag Bhattacharya)
September, 2024
Department of Physics
Assam Don Bosco University.

ABSTRACT

This thesis presents a comprehensive study of Supersoft X-ray Sources (SSS), utilizing advanced spectral fitting techniques and variability analysis. By analyzing high-resolution X-ray spectra from multiple observatories, we have developed robust models to characterize the spectral properties and physical processes governing SSS. The study explores the diversity of SSS types, their spectral features, luminosity, and variability characteristics. Key findings include differences in interstellar absorption between Milky Way and Large Magellanic Cloud SSS, a correlation between luminosity and mass accretion rate, and the presence of periodic modulations. These results provide insights into the evolutionary pathways of SSS and their role in astrophysical phenomena. Future research directions include expanding the dataset, refining theoretical models, and conducting high-resolution timing studies to further unravel the complexities of these fascinating objects.

Keywords: X-rays: binaries, X-rays: individual: CAL 83, X-rays: individual: RS Oph, X-rays: individual: RX J0019.8+2156, X-rays: individual: RX J0527.8-6954, X-rays: individual: RX J0925.7-4758, methods: observational, techniques: spectroscopic, techniques: photometric

Contents

1 INTRODUCTION	1
1.1 Introduction	3
1.1.1 X-ray Astronomy	3
1.1.2 Supersoft X-ray Sources	4
1.2 Background	4
1.3 Current Status	5
1.3.1 Supersoft X-ray Sources as a New Class of Luminous X-ray Sources	5
1.3.2 Currently Accepted Model for SSS	6
1.3.3 Possible Companions to a White Dwarf in SSS	8
1.3.4 Classification Scheme for SSS	9
1.3.5 Classical Novae	10
1.4 Statement of the Research Problem	12
1.5 Hypothesis	12
1.6 Objectives of the Study	12
1.7 Significance of the Study	13
1.8 Contribution of the Thesis	14
1.9 Organization of the Thesis	15
2 METHODOLOGY	17
2.1 Spectral Study of Supersoft X-ray Sources	19
2.1.1 Basics of Spectral Curve Fitting	19

2.1.2	Software Implementation	22
2.1.3	Implementation of Spectral Fitting using XSPEC	26
2.2	Variability Study of Supersoft X-ray Sources	28
2.2.1	Introduction to Lomb-Scargle Periodogram	28
2.2.2	Theory of Lomb-Scargle Periodogram	29
2.2.3	Implementation of Lomb-Scargle Periodogram	36
3	MULTI-OBSERVATORY SPECTRAL ANALYSIS OF SUPERSOFT X-RAY SOURCES	39
3.1	Literature Review	41
3.2	Journal of Observations	44
3.3	Data Reduction and Analysis	45
3.3.1	XMM-Newton EPIC-pn Data Reduction	46
3.3.2	ASCA SIS1 Data Reduction	47
3.3.3	Chandra ACIS Data Reduction	48
3.3.4	NICER XTI Data Reduction	49
3.4	Results	50
3.4.1	Observed Count Rates	50
3.4.2	NLTE Continuum Model	52
3.4.3	Unfolded Spectra from Best-fit Model	53
3.4.4	Luminosity versus Effective Temperature	54
3.4.5	Presence of Elemental Absorption Edges	54
3.5	Discussion	58

3.5.1	Best-fit Continuum Model	58
3.5.2	NLTE Pure H Model Atmosphere	59
3.5.3	Inferences from Results	65
3.5.4	Relative Strengths of Absorption Edges	69
4	FITTING THE HIGH RESOLUTION SPECTRUM OF RX J0925.7-4758 USING COMPOSITE XSPEC MODEL	70
4.1	RGS Spectra from XMM-Newton	72
4.1.1	RGS Design and Functionality	73
4.1.2	Light Path and Spectral Orders	74
4.2	Models for Data Fitting	75
4.2.1	Model Grid	76
4.2.2	Model Components	77
4.3	Analysis of High-resolution RGS Spectra	84
4.3.1	Fitting of RGS1 Spectra	84
4.3.2	Fitting of RGS2 Spectra	88
4.3.3	Comparison with Current Literature	89
4.3.4	Supersoft X-ray Physics using Best-fit RGS Model	91
5	TOOL FOR IDENTIFICATION OF SPECTRAL LINES IN XMM-NEWTON RGS SPECTRA	94
5.1	Introduction	96
5.2	Working with RGS Data Files	96
5.2.1	Extraction of RGS Spectra	98

5.2.2	Accessing Spectral Data from FITS Files	99
5.2.3	Collecting Atomic Lines	100
5.2.4	Demonstration	101
5.2.5	Distribution	109
5.2.6	Future scope	110
5.3	Stellar Winds from Hot Luminous Massive Stars	112
5.3.1	Mass-Loss Rate	112
5.3.2	Terminal Velocity	112
5.4	Relevance of Stellar Winds	113
5.5	Radiative Line-Driving: Mechanism of Stellar Winds	114
6	RESULTS OF SPECTRAL FITTING AND VARIABILITY STUDIES	115
6.1	Source list	117
6.1.1	Journal of observations	118
6.1.2	Supersoft X-ray count rates	119
6.2	Spectral fitting	121
6.2.1	Best-fit statistics	121
6.2.2	Stellar parameters	121
6.2.3	Unfolded spectra from best-fit models	122
6.2.4	Presence of elemental absorption edges	126
6.3	Variability study of RX J0925.7-4758	130
6.3.1	Observed NICER and XMM-Newton lightcurves	130
6.3.2	Timing analysis of NICER lightcurves	131

6.3.3	Timing analysis of XMM-Newton lightcurves	136
6.3.4	Summary of timing analysis of RX J0925.7-4758	142
7	CONCLUSIONS AND DISCUSSION	144
7.1	Physical Interpretation	146
7.2	Implications	146
7.3	Future Scope	147
Bibliography		149
Appendix A	Bio-data of the Scholar	160
Appendix B	Transcript of the Ph.D. Coursework	161
Appendix C	List of Publications	163
C.1	Journal Publications	163
C.1.1	Published	163
C.1.2	Communicated to Journal	163
C.2	Conference Presentation	163
Appendix D	Python Code Listings	165
D.1	Line Identification Tool for XMM-Newton RGS Spectra	165
D.2	Multi-Observatory Spectral Analysis	185
D.2.1	Observed Count Rates	185
D.2.2	Unfolded Spectra	189
D.2.3	Luminosity versus Effective Temperature	191

D.2.4	Identification of Elemental Absorption Edges	193
D.2.5	Residual Statistics	201
D.3	Python Interface for Spectral Analysis	214
D.3.1	Phase I: PySAS Data Reduction	214
D.3.2	Phase II: Pileup Investigation	215
D.3.3	Phase III: Spectrum Extraction	217
D.3.4	Phase IV: PyXspec Analysis	221
D.4	Variability Analysis	227
D.4.1	Reading Lightcurves	228
D.4.2	Lomb-Scargle Periodogram	231
D.4.3	Timing Analysis	233

List of Figures

1.1	Regimes of nuclear burning in surface layers of a white dwarf [1]	7
3.1	Source and background extraction regions for the XMM-Newton observation of RX J0925.7-4758	47
3.2	Source and background extraction regions for the Chandra observation of RX J0925.7-4758	49
3.3	Comparison of flux from all observations	51
3.4	Unfolded spectra after model fitting	53
3.5	Uncertainties in L_* and T_{eff} calculations from multi-observatory observations of RX J0925.7-4758	54
3.6	Unfolded spectra with overlaid elemental absorption edges	55
3.7	Absorption edges for N, O, Ne and Fe overlaid on the unfolded spectrum for EPIC-pn observation	56
3.8	Residual statistics from best-fit model to all observations	59
3.9	Gaussian approximations of the KDE functions for residual distributions of all observations	60
3.10	Overlay of transition lines on the fluxed RGS spectrum of RX J0925.7-4758 for Obs. ID 0111150101	68
4.1	Schematic of the RGS instruments of XMM-Newton. Courtesy: Hardware Schematics at XMM-Newton SOC	73
4.2	χ^2_{red} trend of RGS1 spectra from RX J0925.7-4758	85
4.3	Model M07 fit to RGS1 spectra	86
4.4	Model M08 fit to RGS1 spectra	86

4.5	Model M09 fit to RGS1 spectra	86
4.6	Model M10 fit to RGS1 spectra	87
4.7	Model M11 fit to RGS1 spectra	87
4.8	Model M12 fit to RGS1 spectra	87
4.9	χ^2_{red} trend of RGS2 spectra from RX J0925.7-4758	89
4.10	Model M07 fit to RGS2 spectra	89
4.11	Model M08 fit to RGS2 spectra	90
4.12	Model M09 fit to RGS2 spectra	90
4.13	Model M10 fit to RGS2 spectra	91
4.14	Model M11 fit to RGS2 spectra	91
4.15	Model M12 fit to RGS2 spectra	92
5.1	Preview of fluxed spectrum	103
5.2	Line-overlaid spectrum using the <code>plot_spectrum()</code> method	105
5.3	Partitioning of the fluxed spectrum within specific regions of interest	106
5.4	Velocity profiles of the Lyman α , β , and γ lines for N VI (left) and N VII (right)	109
5.5	Velocity profiles of the Lyman α , β , and γ lines for O VII (left) and O VIII (right)	110
5.6	Velocity profiles of three Fe XVII (left) and Fe XVIII (right) lines	111
5.7	Principle of radiative line driving	114
5.8	Momentum transfer during radiative line driving	114
6.1	Sky positions of SSS chosen for the dataset	117

6.2	Count rates of individual sources in dataset in SSS regime	119
6.3	Count rates of all sources in dataset	120
6.4	L_* and T_{eff} computed for all SSS in dataset	122
6.5	Unfolded spectrum after model fitting for CAL 83	123
6.6	Unfolded spectrum after model fitting RS Oph	124
6.7	Unfolded spectrum after model fitting for RX J0019.8+2156	124
6.8	Unfolded spectrum after model fitting for RX J0527.8-6954	125
6.9	Unfolded spectrum after model fitting for RX J0925.7-4758	125
6.10	Identified absorption edges for RS Oph	127
6.11	Identified absorption edges for RX J0019.8+2156	128
6.12	Identified absorption edges from RX J0925.7-4758 spectrum	129
6.13	NICER lightcurves of RX J0925.7-4758 using 16 s time bins	130
6.14	XMM-Newton lightcurves of RX J0925.7-4758 using 50 s time bins	131
6.15	Lomb-Scargle periodogram of NICER lightcurves	132
6.16	Phase-folded NICER lightcurve over 0.2–1.0 keV with best-fit sinusoids	133
6.17	NICER lightcurve over 0.2–1.0 keV with best-fit sinusoids	134
6.18	Phase-folded NICER lightcurve over 0.2–12.0 keV with best-fit sinusoids	135
6.19	NICER lightcurve over 0.2–12.0 keV with best-fit sinusoids	136
6.20	Lomb-Scargle periodogram of XMM-Newton lightcurves	137
6.21	Phase-folded XMM-Newton EPIC-pn lightcurve over 0.2–1.0 keV with best-fit sinusoids	138
6.22	XMM-Newton EPIC-pn lightcurve over 0.2–1.0 keV with best-fit sinusoids	139

6.23 Phase-folded XMM-Newton EPIC-pn lightcurve over 0.2–12.0 keV with best-fit sinusoids	140
6.24 XMM-Newton EPIC-pn lightcurve over 0.2–12.0 keV with best-fit sinusoids	141
6.25 Phase-folded XMM-Newton RGS lightcurve over 0.2–12.0 keV with best-fit sinusoid	141
6.26 XMM-Newton RGS lightcurve over 0.2–12.0 keV with best-fit sinusoid	142

List of Tables

1.1	Classification of binary systems that may manifest as SSS [2]	10
3.1	Journal of observations	44
3.2	Parameters of RX J0925.7-4758 derived from continuum NLTE model from multi-observatory observations.	52
3.3	Absorption depth D of RX J0925.7-4758 derived from identified absorption edges of Ne, Fe, O and N.	57
4.1	In-orbit performance of RGS instruments	74
4.2	Wavelength and photon energy ranges covered by RGS	75
4.3	List of models used for RGS data fitting for RX J0925.7-4758	77
4.4	Model parameters for <code>ismabs</code>	79
4.5	Model parameters for <code>apec</code>	80
4.6	Model parameters for <code>mekal</code>	81
4.7	Model parameters for <code>swindl</code>	81
4.8	Elemental abundances for TMAP grid	83
4.9	Model parameters for <code>rauch</code>	83
4.10	Fitting statistics of RGS1 spectra from RX J0925.7-4758	84
4.11	Fitting statistics of RGS2 spectra from RX J0925.7-4758	88
4.12	Comparison of best-fit statistics of RGS spectra from RX J0925.7-4758 . .	92
5.1	List of lines along with transitions, considered for inspection with respect to the radial velocity	108

6.1	List of sources in dataset	117
6.2	SSS observation journal of dataset	118
6.3	Fitting statistics for all sources in dataset	121
6.4	Stellar parameters of all sources from continuum NLTE model	122
6.5	Absorption edges identified for RS Oph	127
6.6	Absorption edges identified for RX J0019.8+2156	128
6.7	Absorption edges identified for RX J0925.7-4758	129
6.8	Results of variability studies on RX J0925.7-4758	143

CHAPTER 1

INTRODUCTION

Contents

1.1	Introduction	3
1.1.1	X-ray Astronomy	3
1.1.2	Supersoft X-ray Sources	4
1.2	Background	4
1.3	Current Status	5
1.3.1	Supersoft X-ray Sources as a New Class of Luminous X-ray Sources	5
1.3.2	Currently Accepted Model for SSS	6
1.3.3	Possible Companions to a White Dwarf in SSS	8
1.3.4	Classification Scheme for SSS	9
1.3.5	Classical Novae	10
1.4	Statement of the Research Problem	12
1.5	Hypothesis	12
1.6	Objectives of the Study	12
1.7	Significance of the Study	13
1.8	Contribution of the Thesis	14
1.9	Organization of the Thesis	15

CHAPTER 1. INTRODUCTION

Abstract

This chapter introduces the field of X-ray astronomy, highlighting its significance in studying high-energy celestial phenomena. It covers the advancements from early X-ray detectors to contemporary missions like Chandra and XMM-Newton, which have provided detailed spectral data of various X-ray sources. A special focus is on luminous supersoft X-ray sources (SSS), a distinct category identified by their exceptionally soft spectra and high luminosities. The chapter explores the prevailing models for SSS, primarily involving accreting white dwarfs in binary systems, and their potential as progenitors of Type Ia supernovae. It also delves into the classifications and companion star types within SSS systems, and the significance of understanding classical novae in the context of X-ray observations. Finally, it outlines the research problem, hypothesis, and objectives, aiming to develop robust models for SSS spectra and understand their underlying physical mechanisms.

1.1 Introduction

The pursuit of cosmic knowledge has been propelled by advancements in observational techniques across the electromagnetic spectrum. X-ray astronomy, in particular, offers a unique window into the high-energy universe, revealing phenomena inaccessible to other wavelengths. By probing celestial objects in the X-ray band, astronomers gain profound insights into extreme astrophysical environments dominated by processes such as accretion, stellar explosions, and the interplay of matter and radiation.

1.1.1 X-ray Astronomy

X-ray astronomy has emerged as a pivotal domain within the broader field of astronomy, providing invaluable insights into high-energy phenomena across the Milky Way and beyond. Observations made in the X-ray window enable us to understand many high-energy events in the Milky Way and other galaxies. By studying celestial objects in the X-ray spectrum, many significant strides have been made in unraveling the complexities of stellar and galactic evolution [3]. Unlike visible light, however X-rays are entirely absorbed by Earth's atmosphere, necessitating observations from space-based platforms. Since the pioneering Uhuru mission in 1970, a succession of advanced X-ray telescopes has been deployed, leading to the discovery of a diverse array of exotic celestial X-ray sources, including black holes, neutron stars, white dwarfs and supernova remnants [4].

X-ray astronomy relies on the detection and analysis of individual X-ray photons to extract critical information, namely its *arrival direction*, *energy*, *time of arrival* and the *polarization angle* [5]. Advancements in space-based X-ray observatory instrumentation have significantly enhanced spectral resolution, providing deeper insights into celestial X-ray sources. While early X-ray detectors primarily employed proportional and scintillation counters, the later development of Wolter-type focusing optics and imaging detectors marked a pivotal turning point, enabling the creation of two-dimensional X-ray images [3]. Contemporary missions like Chandra and XMM-Newton utilize actively cooled, pixelated solid-state detectors as their focal plane instruments, offering superior energy resolution and broader energy coverage compared to their predecessors [6, 7]. Complementing imaging capabilities, these observatories also incorporate high-resolution grating spectrometers, providing exceptional spectral resolution and high throughput for detailed spectral analysis [8].

1.2. Background

1.1.2 Supersoft X-ray Sources

Luminous supersoft X-ray sources (SSS) were initially identified as a distinct category of exceptionally bright X-ray emitters by Trümper *et al.* [9], Greiner *et al.* [10], and Kahabka *et al.* [11]. These objects are characterized by extraordinarily soft X-ray spectra peaking within the 15–80 eV energy range, corresponding to blackbody temperatures of approximately 300,000–500,000 K [11]. Their immense X-ray luminosities, typically on the order of the Eddington limit ($\sim 10^{38}$ erg s $^{-1}$), place them among the most powerful X-ray sources known. The extreme softness of their spectra suggests a unique physical mechanism for their energy production, distinguishing SSS from other classes of luminous X-ray binaries.

The initial discovery of luminous SSS can be attributed to the pioneering observations conducted by the Einstein and ROSAT satellites, which employed low-resolution proportional counters. These early detections laid the groundwork for subsequent, more detailed investigations. Notably, the Chandra and XMM-Newton observatories have provided high-resolution X-ray spectra for a substantial number of SSS, enabling unprecedented insights into their physical properties. Among the most extensively studied SSS are RX J0925.7-4758 [12, 13], CAL 83 [14], CAL 87 [15], and RX J0019.8+2156 [16], which have yielded particularly remarkable spectral data.

The prevailing model for luminous SSS posits that the majority of these objects are accreting white dwarfs (WDs) within binary systems. In these systems, hydrogen-rich material is transferred from a companion star onto the WD's surface, where it accumulates and undergoes thermonuclear burning [17]. This sustained or intermittent hydrogen burning process is responsible for the observed intense, soft X-ray emission. While this model accounts for most SSS, its important to note that there are exceptions, such as isolated WDs undergoing final helium-shell flashes, which also exhibit supersoft X-ray characteristics.

1.2 Background

Once considered enigmatic, the study of SSS has evolved into a cornerstone for understanding the evolution of X-ray binaries. A significant fraction of the identified SSS exhibit transient behaviour. These transient SSS are categorized into two primary groups: classical and symbiotic novae with a supersoft phase, and systems without documented nova outbursts but

1.3. Current Status

characterized by alternating “on” and “off” supersoft X-ray states. While the precise evolutionary pathway leading to Type Ia supernovae (SNe Ia) remains an active area of research, the calculated rates of these events align with the predicted outcomes of two leading progenitor scenarios: merging double degenerate white dwarfs and carbon-oxygen white dwarfs accreting from a non-degenerate companion (CBSS). The intricate spectra of many SSS, as captured by high-resolution grating spectrometers, prominently feature P Cygni profiles. Although accurate modelling of these spectra presents a formidable challenge but is essential for deriving fundamental parameters of the SSS under investigation.

1.3 Current Status

This section provides an overview of the current state of knowledge regarding observational data obtained from SSS. It explores the key findings and challenges associated with analysing these data, drawing upon the latest research in the field.

1.3.1 Supersoft X-ray Sources as a New Class of Luminous X-ray Sources

The initial discovery of luminous SSS was hindered by the limitations of the Einstein satellite, which possessed a restricted spectral range and resolution. Consequently, the first four SSS identified by this mission were initially classified alongside other, more typical high-luminosity X-ray sources, such as accreting neutron stars or black holes within binary systems [18, 19].

It was the subsequent launch of the ROSAT satellite, equipped with enhanced spectral capabilities, that enabled SSS to be recognised as a distinct class of object. In contrast to the classical high-luminosity X-ray sources, SSS exhibited a characteristic spectral peak within the 15-80 eV energy band. This corresponds to blackbody temperatures approximately two orders of magnitude lower than those associated with accreting neutron stars or black holes [20].

1.3. Current Status

1.3.2 Currently Accepted Model for SSS

If the luminosity and effective temperature of a star are known, using the *Stefan-Boltzmann's law*, one can derive its radius. The Stefan-Boltzmann's law is as follows

$$L = 4\pi R^2 \sigma T^4 \quad (1.1)$$

Rearranging this equation yields:

$$R = 9 \times 10^8 (L_{37.5})^{1/2} (T_e / 40 \text{ eV})^{-2} \text{ cm}, \quad (1.2)$$

where $L_{37.5}$ is the X-ray luminosity in units of $10^{37.5}$ erg/s, and T_e is the effective temperature in electron volts.

Applying in equation (1.2) the characteristic values of SSS, i.e. $L_{37.5} = 1$ and $T_e = 40$, results in a radius of approximately 9000 km, closely resembling the size of a white dwarf. This similarity in radius, analogous to the relationship between accretion onto neutron stars or black holes in classical X-ray binaries, strongly suggests that SSS are also powered by accretion onto a white dwarf primary.

Van den Heuvel (1992) proposed a model in which white dwarfs with masses in the range $0.7 - 1.4 M_\odot$, accreting matter at rates of approximately $\sim 1 - 5 \times 10^{-7} M_\odot \text{ yr}^{-1}$, produce the observed supersoft X-ray emission [17]. These models are envisioned as binary systems where the accreting white dwarf is accompanied by either a main-sequence star with a mass in the range $1.4 - 1.5 M_\odot$ or a post-main-sequence star with a mass $1.5 - 2 M_\odot$. Theoretical calculations indicate that Roche lobe overflow in these systems would lead to mass transfer rates from the companion stars consistent with the observed values, being $\sim 10^{-7} M_\odot \text{ yr}^{-1}$ and $\sim 4 \times 10^{-7} M_\odot \text{ yr}^{-1}$ respectively. Furthermore, the estimated X-ray lifetimes for such systems are around $\sim 10^7$ years.

Nomoto's (1982) seminal work elucidated the diverse nature of nuclear burning processes occurring on the surfaces of accreting white dwarfs. Based on the interplay between accretion rate and white dwarf mass, he identified three distinct burning regimes [1]. These regimes, graphically illustrated in figure 1.1, encompass a spectrum of thermonuclear activity, ranging from stable hydrogen burning to explosive helium flashes and detonations.

1.3. Current Status

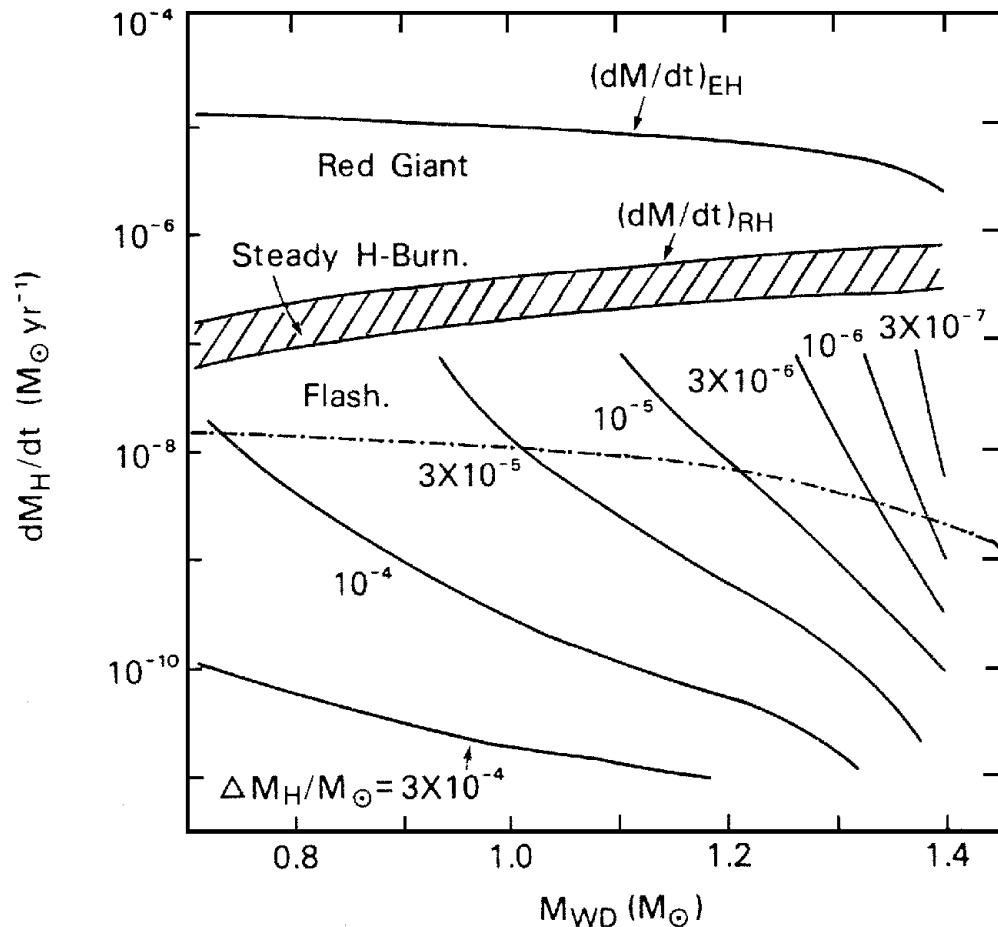


Figure 1.1: Regimes of nuclear burning in surface layers of a white dwarf [1]

These regimes are summarized as follows, for a white dwarf with solar mass:

1. **Stable hydrogen burning:** For accretion rates within a narrow range of $1 - 4 \times 10^{-7} M_{\odot} \text{ yr}^{-1}$, the accreted hydrogen undergoes a stable, continuous burning process on the white dwarf's surface. Crucially, this steady burning occurs without causing significant expansion of the white dwarf itself.
2. **Thermonuclear flashes:** When the accretion rate falls below the threshold for stable burning, the accreted hydrogen ignites in a series of thermonuclear flashes. These explosive events are intermittent, with the frequency of flashes decreasing as the accretion rate diminishes. Concurrently, the individual flashes become more energetic and violent under these conditions.

1.3. Current Status

3. **Stable hydrogen burning with radius expansion:** If the accretion rate surpasses the upper limit for stable burning, the white dwarf undergoes dramatic expansion, resembling a red giant star. Despite this expansion, hydrogen burning continues steadily within a thin shell surrounding the white dwarf's core. Intriguingly, this process mirrors the growth of degenerate cores in red giant stars, where hydrogen shell burning is also responsible for the expansion.

Hachisu et al. (1996) have calculated that when accretion rates onto a white dwarf significantly exceed the threshold for stable hydrogen burning, a fundamental transformation occurs. Instead of maintaining a static envelope, the accreting matter is expelled in a powerful stellar wind, reaching velocities of approximately 5000 km/s [21]. Crucially, this high-speed outflow is opaque to soft X-rays, preventing their escape. Consequently, to be observed as a luminous supersoft X-ray source (SSS), the white dwarf must maintain an accretion rate within the stable hydrogen burning regime.

Under these conditions of sustained hydrogen burning, the white dwarf gradually accretes helium while increasing its overall mass. Hachisu et al. suggest that this process can continue until the white dwarf reaches a critical mass of approximately $1.38 M_{\odot}$, at which point a Type Ia supernova is triggered. This theoretical framework establishes a compelling suggestion of *SSS being the potential progenitors of Type Ia supernovae*.

1.3.3 Possible Companions to a White Dwarf in SSS

Understanding the nature of the companion star in an SSS system is crucial for elucidating the system's evolutionary history and the mechanisms driving its X-ray emission. This section explores two primary categories of companion stars found in SSS: near main-sequence stars more massive than the white dwarf and symbiotic systems.

1.3.3.1 Near main-sequence stars more massive than WDs

The simplest configuration for a luminous SSS involves a binary system comprising a white dwarf and a more massive companion star with a radiative envelope, exceeding approximately $1.3 M_{\odot}$. Once mass transfer from the donor star to the white dwarf commences due to Roche lobe overflow, the binary orbit undergoes shrinkage. The donor star experiences

1.3. Current Status

thermal instability, leading to sustained mass transfer until its mass becomes lower than that of the white dwarf, at which point the orbital separation increases [22]. These systems are termed close-binary supersoft sources (CBSS). Greiner has identified and catalogued¹ nine such systems, providing observational support for this model [23].

1.3.3.2 Symbiotic systems

Sion et al. (1994) first identified symbiotic systems as another potential host environment for SSS. These systems consist of a white dwarf accreting matter from a red giant companion star. The specific characteristics of the mass transfer process in symbiotic systems can vary depending on the evolutionary stage of the red giant [24]. Two primary subtypes of symbiotic SSS are commonly recognized, that provide different modes of mass-transfer.

1. One subtype of symbiotic supersoft X-ray sources involves a $1 M_{\odot}$ red giant companion that is less massive than the white dwarf. These systems typically exhibit orbital periods exceeding 125 days. The red giant component in such systems possesses a degenerate helium core, and its evolutionary processes drive the mass transfer onto the white dwarf. This mass transfer is facilitated by Roche lobe overflow.
2. Symbiotic systems can also involve a red giant companion in the asymptotic giant branch (AGB) phase of its evolution. Unlike the first subtype, these AGB stars do not fill their Roche lobes. However, they exhibit intense stellar winds, characterized by mass loss rates of the order of $10^{-5} M_{\odot} \text{ yr}^{-1}$ and velocities around 30 km/s. A fraction of this ejected material is captured by the white dwarf companion, resulting in accretion rates of about $10^{-7} M_{\odot} \text{ yr}^{-1}$. These accretion rates are sufficient to power the white dwarf as a luminous supersoft X-ray source.

1.3.4 Classification Scheme for SSS

As per di Stefano *et al.* (1996) [2], binary systems that may appear as luminous SSS may be classified as per table 1.1.

In table 1.1, [A] only those systems are considered in which H-rich material is accreting

¹<http://www.mpe.mpg.de/~jcg/sss/ssscat.html>

1.3. Current Status

on the surface of a C-O white dwarf. Here CV = cataclysmic variable, CBSS = close-binary supersoft source and WBSS = wide-binary supersoft source. [B] Two prominent mass transfer mechanisms are mb = magnetic braking and gr = gravitational radiation. [C] \dot{M} is the mass accretion rate on the surface layers of the white dwarf. [D] P_{orb} refers to the orbital period of the binary. [D] When \dot{M} is in the correct range (about $\gtrsim 10^{-7} M_{\odot} \text{ yr}^{-1}$), the source burns nuclear fuel more or less steadily (S); for smaller values of \dot{M} , hydrogen burns sporadically (R).

Table 1.1: Classification of binary systems that may manifest as SSS [2]

[A] System Type	[B] Mass Transfer Mechanism	[C] Mass Accretion Rate \dot{M} ($M_{\odot} \text{ yr}^{-1}$)	[D] Orbital Period P_{orb} (days)	[E] Steady or Recurrent
CVs	mb and/or gr	$\lesssim 10^{-8}$	$\lesssim 0.2$	R
CBSSs	thermal time scale readjustment of donor	$\gtrsim 10^{-7}$	$\sim 0.2 - 3.0$	S
		$\lesssim 10^{-7}$	$\sim 0.2 - \mathcal{O}(10^2)$	R
WBSSs	nuclear evolution of donor	$\gtrsim 10^{-7}$	$\sim 3.0 - \mathcal{O}(10^2)$	S
		$\lesssim 10^{-7}$	$\sim 3.0 - \mathcal{O}(10^2)$	R
Symbiotics (Wind-driven)	stellar winds from evolved donor	$\gtrsim 10^{-7}$	$\mathcal{O}(10^2)$	S
		$\lesssim 10^{-7}$	$\mathcal{O}(10^2)$	R

1.3.5 Classical Novae

Classical novae rank as the third most cataclysmic explosions within a galaxy, surpassed only by supernovae and gamma-ray bursts in terms of energy release. While novae are far less energetic than supernovae, typically liberating luminosities of more than $10^{45} \text{ erg s}^{-1}$, they occur with significantly higher frequency, making them more commonly observed phenomena within galactic environments.

The prevailing model for classical novae, as outlined by Krautter (2008), posits their occur-

1.3. Current Status

rence within cataclysmic binary systems. These systems comprise a white dwarf primary accreting matter from a main-sequence or slightly evolved companion star. The mass transfer process, facilitated by Roche lobe overflow, channels material towards the white dwarf through the inner Lagrangian point, forming an accretion disk.

The conditions leading to a thermonuclear runaway on the white dwarf's surface are intricately linked to factors such as the white dwarf's mass and luminosity, the accretion rate, and the chemical composition of the accreted material [25]. As hydrogen-rich matter accumulates, it undergoes compression and heating. When temperatures within the accreted layer surpass 10^8 K, runaway thermonuclear fusion ignites, releasing vast amounts of energy. The peak luminosity during these events can approach the Eddington limit L_{Edd} , a critical threshold beyond which radiation pressure exceeds gravitational force, imposing an upper limit on the luminosity of a star.

X-ray observations have emerged as a pivotal tool for investigating the high-temperature phases associated with nova outbursts. While these observations have yielded valuable insights, the relatively small number of novae studied in the X-ray band compared to other spectral domains (optical, infrared, ultraviolet) has hindered the development of a comprehensive understanding of novae in this energy range. Nevertheless, two primary mechanisms have been proposed to explain X-ray emission from novae.

1.3.5.1 Thermal radiation from hot white dwarf

Following a nova eruption, the energy released from thermonuclear burning is rapidly transported to the white dwarf's surface. This influx of energy dramatically increases the star's effective temperature. For a white dwarf radiating at its Eddington luminosity L_{Edd} , the temperature could be expected to be several thousand kelvins, depending on the white dwarf's mass. Consequently, the nova enters a phase of intense soft X-ray emission, characterized by a spectral energy distribution (SED) resembling that of a hot stellar atmosphere.

As the nova ejecta expand outward, they form a cooling pseudo-photosphere. This expansion leads to a rapid decrease in temperature and a corresponding decline in X-ray flux due to the increasing opacity of the expanding envelope to X-rays [26].

1.4. Statement of the Research Problem

1.3.5.2 X-ray emission from circumstellar material

A second mechanism contributing to X-ray emission in nova systems involves interactions between the expanding nova ejecta and the surrounding circumstellar environment. This interaction can occur either within the ejected material itself or between the ejecta and pre-existing circumstellar matter. As proposed by Balman et al. (1998), shock waves generated by these collisions can reach temperatures of several keV, producing thermal *bremsstrahlung* radiation in the X-ray band [27].

1.4 Statement of the Research Problem

To understand the nature of binary systems emitting supersoft X-rays by the analysis of their spectra and the study of their variabilities in X-ray and visible/ultraviolet regimes.

1.5 Hypothesis

The radiative processes governing supersoft X-ray sources are much complex than previously assumed and could include non-steady states, NLTE and stellar winds.

1.6 Objectives of the Study

1. To develop a robust and consistent model that accurately fits the continuum spectra of supersoft X-ray sources (SSS) across different sources, and using multi-observatory science data.
2. To demonstrate that the spectra of SSS within the 0.2–1.0 keV photon energy range can be satisfactorily modeled using Non-Local Thermodynamic Equilibrium (NLTE) components, and to validate this approach across multiple sources.
3. To investigate and distinguish the absorption components due to the source atmosphere and the interstellar medium (ISM) in the spectra of SSS from both the Milky Way and the Large Magellanic Cloud (LMC).

1.7. Significance of the Study

4. To compute stellar parameters for the selected SSS using the best-fit models, thereby constraining their luminosities, effective temperature and column densities.
5. To identify and analyze the differences in spectral characteristics, such as absorption edges, between SSS in the Milky Way and those in the LMC, and to understand the underlying reasons for these differences.
6. To develop a software tool that accesses publicly available atomic data and overlays the relevant atomic lines on the grating spectra of SSS, thereby facilitating preliminary line identification for astronomers.
7. To study a grid of models for fitting the grating spectra of SSS, comparing their fitting statistics.

1.7 Significance of the Study

The study of SSS is crucial for understanding the physics and evolution of compact binary systems, such as white dwarfs and their interactions with companion stars. This research addresses several key challenges in modeling the spectra of SSS, providing significant advancements in the field of high-energy astrophysics.

Firstly, developing a robust and consistent model that accurately fits the continuum spectra of SSS across different sources and using multi-observatory data is fundamental to obtaining reliable measurements of their physical parameters. This helps in better understanding the nature of these sources, their life-cycle, and their role in the broader context of stellar evolution.

Secondly, demonstrating that the spectra of SSS within the 0.2–1.0 keV photon energy range can be satisfactorily modelled using Non-Local Thermodynamic Equilibrium (NLTE) components fills a crucial gap in current astrophysical models. NLTE modeling provides a more accurate representation of the physical conditions in the atmospheres of SSS, leading to more precise determinations of their effective temperatures, luminosities, and column densities. Additionally, the spectra fitted within this range can be extrapolated to lower energies (or higher wavelengths) and compared with observations made in the UV/optical regime. This cross-validation enhances the reliability of the models and provides a comprehensive understanding of the sources across different wavelengths.

1.8. Contribution of the Thesis

Thirdly, investigating the absorption components due to the source atmosphere and the interstellar medium (ISM) in the spectra of SSS from both the Milky Way and the Large Magellanic Cloud (LMC) adds another layer of understanding regarding the interaction of X-ray photons with intervening matter. This not only enhances our comprehension of the local environments of these sources but also provides insights into the composition and distribution of the ISM in different galactic environments.

Additionally, by computing stellar parameters using best-fit models, the study places constraints on the luminosities, effective temperatures, and column densities of SSS, contributing valuable data to the field of stellar astrophysics. Identifying and analyzing differences in spectral characteristics, such as absorption edges, between SSS in the Milky Way and those in the LMC, sheds light on the unique astrophysical processes at play in different galactic contexts.

Furthermore, the development of a software tool for preliminary line identification using publicly available atomic data is a significant contribution to the astronomical community. This tool facilitates the identification of key atomic lines in grating spectra, enabling more detailed and accurate spectral analysis.

Finally, by studying a grid of models for fitting the grating spectra of SSS and comparing their fitting statistics provides a comprehensive evaluation of different modeling approaches, ensuring the selection of the most appropriate models for accurate spectral fitting.

1.8 Contribution of the Thesis

The current thesis makes several important contributions to the field of high-energy astrophysics, particularly in the study of supersoft X-ray sources:

- I. **Development of a Robust Spectral Model:** The current thesis presents a robust and consistent model that accurately fits the continuum spectra of SSS across different sources using data from multiple observatories. This model provides a reliable framework for analyzing and interpreting SSS spectra.
- II. **Validation of NLTE Modeling:** By demonstrating that the spectra of SSS within the 0.2–1.0 keV photon energy range can be satisfactorily modeled using NLTE components, this research validates a crucial approach for studying the physical conditions

1.9. Organization of the Thesis

in the atmospheres of these sources. This contributes to more accurate determinations of their temperatures and luminosities. Moreover, the extrapolation of these models to UV/optical observations strengthen the consistency and reliability of the models.

- III. Analysis of Absorption Components:** The investigation into the absorption components due to the source atmosphere and the ISM in the spectra of SSS from both the Milky Way and the LMC provides a deeper understanding of the interactions between X-ray photons and intervening matter. This analysis enhances our knowledge of the local environments of SSS and the ISM's composition in different galaxies.
- IV. Computation of Stellar Parameters:** The thesis provides precise computations of stellar parameters for the selected SSS, including luminosities, effective temperatures, and column densities. These results are valuable for constraining models of these sources and understanding their physical properties.
- V. Identification of Spectral Absorption Edges:** By identifying and analyzing differences in absorption edges, between SSS in the Milky Way and those in the LMC, the research offers insights into the unique astrophysical processes in different galactic environments. This helps explain the variations in the observed spectra of SSS.
- VI. Development of a Line Identification Tool:** The development of a Python-based software tool² that accesses atomic data and overlays relevant atomic lines on the grating spectra of SSS is a practical contribution. This tool aids astronomers in performing preliminary line identification, facilitating further spectral analysis and interpretation.
- VII. Evaluation of Spectral Models:** By studying a grid of models for fitting the grating spectra of SSS and comparing their fitting statistics, the thesis provides a comprehensive evaluation of different modeling approaches. This ensures the selection of the most suitable models for accurate spectral fitting, enhancing the reliability of spectral analysis.

1.9 Organization of the Thesis

This thesis is organized into the following chapters, each addressing a specific aspect of the current work on SSS.

²https://github.com/pararover/xmmrgs_lines

1.9. Organization of the Thesis

The current chapter, numbered one, titled *Introduction* introduces the field of X-ray astronomy, focusing on supersoft X-ray sources (SSS). It explores the characteristics, models, and implications of SSS, including their potential connection to Type Ia supernovae. The chapter also outlines the research problem and objectives of the thesis, aiming to develop robust models for SSS spectra and understand their underlying physical mechanisms.

The second chapter titled *Methodology* presents a comprehensive framework for analyzing the spectral and variability characteristics of SSS. It outlines the fundamental principles of spectral curve fitting and the application of the Lomb-Scargle periodogram for identifying periodic signals.

The third chapter titled *Multi-Observatory Spectral Analysis of Supersoft X-ray Sources* focuses on the spectral characteristics of RX J0925.7-4758, utilizing a dataset from multiple space observatories. It aims to establish a robust NLTE spectral model and characterize the continuum spectrum of this SSS.

The fourth chapter titled *Fitting the High Resolution Spectrum of RX J0925.7-4758 Using Composite XSPEC Model* delves into a new analysis of high-resolution X-ray spectra of RX J0925.7-4758, employing spectral fitting techniques to identify the most accurate model for representing the observed data.

The fifth chapter titled *Tool for Identification of Spectral Line in XMM-Newton RGS Spectra* introduces a Python-based tool for identifying spectral lines in X-ray spectra, enabling astronomers to perform preliminary line identification before advanced analysis.

The sixth chapter titled *Results of Spectral Fitting and Variability Studies* explores a dataset of SSS, analyzing their properties, spectral features, and variability characteristics. It presents the results of spectral fitting, stellar parameter derivation, and timing analysis.

Finally, the seventh chapter titled *Conclusions and Discussion* provides a comprehensive synthesis of the findings, highlighting the key conclusions, implications, and future research directions. It discusses the diversity of SSS types, their spectral features, luminosity, and variability characteristics, and the insights gained into their evolutionary pathways.

CHAPTER 2

METHODOLOGY

Contents

2.1	Spectral Study of Supersoft X-ray Sources	19
2.1.1	Basics of Spectral Curve Fitting	19
2.1.2	Software Implementation	22
2.1.3	Implementation of Spectral Fitting using XSPEC	26
2.2	Variability Study of Supersoft X-ray Sources	28
2.2.1	Introduction to Lomb-Scargle Periodogram	28
2.2.2	Theory of Lomb-Scargle Periodogram	29
2.2.3	Implementation of Lomb-Scargle Periodogram	36

Abstract

This chapter presents a comprehensive methodology for analyzing the spectral and variability characteristics of Supersoft X-ray Sources (SSS). The spectral analysis section delves into the fundamentals of spectral curve fitting, including model-based fitting, the χ^2 statistic, and parameter estimation. The essential components and data preprocessing steps involved in spectral analysis are also outlined. The variability study section focuses on the application of the Lomb-Scargle periodogram for identifying and characterizing periodic signals within SSS lightcurves. The methodology includes data acquisition, preprocessing, periodogram calculation, statistical significance assessment, phase-folding, and sinusoidal fitting. By combining these techniques, this chapter provides a robust framework for extracting valuable insights into the spectral and variability properties of SSS.

2.1 Spectral Study of Supersoft X-ray Sources

2.1.1 Basics of Spectral Curve Fitting

Spectral curve fitting is a fundamental technique in data analysis, particularly in fields involving spectroscopy. It involves modeling observed spectral data using mathematical functions, often with adjustable parameters, to extract meaningful information about the underlying physical processes.

2.1.1.1 The Spectral Fitting Problem

Spectrometers do not directly measure the true spectrum of a source. Instead, they record photon counts in specific instrument channels, which are influenced by the instrument's response function. Mathematically, the observed spectrum $C(I)$ is related to the actual spectrum $f(E)$ through the integral equation (2.1).

$$C(I) = \int_0^{\infty} f(E)R(I, E)dE \quad (2.1)$$

In equation (2.1), $R(I, E)$ represents the instrument response function, which describes how photons of different energies are detected and recorded in the instrument channels. Inverting this equation to directly obtain the true spectrum from the observed data can be challenging due to non-uniqueness and instability issues. Therefore, an alternative approach of model selection and fitting is commonly adopted.

2.1.1.2 Model-Based Fitting

In spectral curve fitting, the initial step involves selecting a suitable model spectrum that can adequately represent the observed data. This model is often described by a set of parameters, and is of the form $f(E, p_1, p_2, p_3, \dots)$. These parameters act as adjustable variables that can be modified to fine-tune the shape and characteristics of the model spectrum.

The process of parameter adjustment involves modifying the values of p_1, p_2, p_3, \dots in an iterative manner to minimize the discrepancy between the model spectrum and the observed

2.1. Spectral Study of Supersoft X-ray Sources

data. This is essentially a trial-and-error approach where different combinations of parameter values are explored until a satisfactory fit is achieved. The goal is to find the optimal set of parameters that produces a model spectrum that closely aligns with the observed data, capturing the underlying physical processes or phenomena.

2.1.1.3 Fitting Methodology

Once a suitable model spectrum is chosen and its parameters are defined, the next step involves calculating a predicted count spectrum, denoted as $C_P(I)$. This predicted spectrum is generated by applying the chosen model to the instrument response function and integrating over the energy range. The resulting $C_P(I)$ represents the expected count spectrum based on the model and the instrument's characteristics.

To assess the agreement between the predicted count spectrum and the observed data, a statistical metric known as the fit statistic is employed. The χ^2 statistic is a commonly used fit statistic, which quantifies the deviation between the two spectra. It is calculated as the sum of the squared differences between the observed and predicted counts, normalized by the variance of the observed counts:

$$\chi^2 \equiv \sum \frac{[C(I) - C_P(I)]^2}{(\sigma(I))^2} \quad (2.2)$$

where $C(I)$ is the observed count in channel I , $C_P(I)$ is the predicted count in channel I , and $\sigma(I)$ is the estimated error in channel I . The error term is often approximated as the square root of the observed count, assuming Poisson statistics.

The goal of spectral curve fitting is to find the set of model parameters that minimizes the χ^2 value. This indicates a better agreement between the predicted and observed spectra, suggesting that the chosen model and its parameters accurately describe the underlying physical processes. To achieve this, the parameters are systematically varied and the corresponding χ^2 values are calculated. The set of parameters that yields the lowest χ^2 value is considered the best fit. This iterative process continues until a satisfactory minimum χ^2 value is reached.

2.1. Spectral Study of Supersoft X-ray Sources

2.1.1.4 Assessing the Goodness of Fit

The reduced χ^2 statistic, defined as:

$$\chi_{\text{red}}^2 = \frac{\chi^2}{\nu} \quad (2.3)$$

provides a valuable metric for evaluating the quality of the fit between a model and the observed data. Here, ν is the degrees of freedom, which is calculated as the difference between the number of channels and the number of model parameters. This effectively accounts for the constraints imposed by the model on the data.

The χ_{red}^2 , as a fit statistic, may be used to assess the goodness of the fit as follows:

- *Good Fit:* When the χ_{red}^2 value is close to 1, it indicates a satisfactory agreement between the model and the data. This suggests that the model captures the underlying physical processes effectively and that the estimated errors are reasonable.
- *Poor Fit:* A χ_{red}^2 value significantly greater than 1 suggests a poor fit. This could be due to several factors, such as an inadequate model, underestimated errors, or systematic biases in the data. A poor fit may necessitate revisiting the model choice or refining the error estimation.
- *Overestimated Errors:* A χ_{red}^2 value significantly less than 1 indicates that the errors on the data have been overestimated. This could be attributed to various factors, including underestimation of systematic uncertainties or the presence of correlations between data points.

The χ_{red}^2 statistic provides a quantitative measure of the goodness-of-fit. It allows for a systematic comparison of different models and helps identify potential shortcomings or inconsistencies in the data or the chosen model. By analyzing the deviation of the χ_{red}^2 value from 1, one can gain insights into the quality of the fit and make informed decisions regarding the suitability of the model and the reliability of the extracted information.

2.1. Spectral Study of Supersoft X-ray Sources

2.1.1.5 Error Estimation

The error width, which quantifies the uncertainty associated with the estimated parameters in spectral curve fitting, can be derived from the covariance matrix obtained during the fitting process. The covariance matrix is a square matrix that provides information about the relationships between the parameters and their associated uncertainties.

The diagonal elements of the covariance matrix represent the variances of the individual parameters. The square root of the diagonal elements gives the standard deviations, which can be interpreted as the error widths. These standard deviations indicate the range of values within which the true parameter value is likely to lie with a certain level of confidence.

The off-diagonal elements of the covariance matrix represent the covariances between pairs of parameters. These values indicate the degree of correlation between the uncertainties in the parameters. A high covariance between two parameters suggests that their uncertainties are interrelated, meaning that changes in one parameter may affect the uncertainty in the other. By examining the covariance matrix, it is possible to identify potential correlations between parameters and assess their impact on the overall uncertainty in the fitted model.

2.1.2 Software Implementation

2.1.2.1 Essential Components

Any generic software implementation of spectral curve fitting necessitates the presence of the following essential components for modeling and interpreting observed spectral data:

1. *Observed Spectra $D(I)$:* The core component of spectral curve fitting is the observed spectral data, denoted as $D(I)$. This dataset represents the measured intensity of radiation as a function of detector channels I . The detector channels correspond to specific energy ranges or wavelengths, capturing the spectral information of the observed phenomenon.
2. *Background Measurements $B(I)$:* In most practical scenarios, the observed spectra are contaminated by background radiation or noise. To accurately isolate the signal of interest, background measurements, denoted as $B(I)$, are typically acquired. These

2.1. Spectral Study of Supersoft X-ray Sources

measurements represent the instrumental background signal in the absence of the target source. By subtracting the background measurements from the observed spectra, it is possible to remove the unwanted noise and isolate the signal of interest.

3. *Instrument Response $R(I, E)$:* The instrument response function $R(I, E)$ maps the detector channels to the corresponding energy bins, providing a relationship between the observed spectral data and the underlying physical phenomenon. The instrument response function accounts for the instrument's sensitivity, resolution, and other characteristics, enabling the accurate interpretation of the measured spectral data.
4. *Model Spectra $M(E)$:* A set of model spectra $M(E)$ represent a variety of potential spectral shapes based on different physical scenarios or theoretical models. By comparing the observed spectra to these model spectra, one can identify the most suitable model and extract valuable information about the underlying physical processes. The choice of model spectra depends on the specific application and the expected characteristics of the observed phenomenon.

2.1.2.2 Data Processing and Model Construction

Background subtraction: The observed spectrum $C(I)$ is typically contaminated by background radiation or noise. To isolate the signal of interest, the background spectrum $B(I)$ is subtracted from the raw data. This process is represented by equation (2.4) as given below:

$$C(I) = \frac{D(I)}{a_D t_D} - \frac{b_D}{b_B} \frac{B(I)}{a_B t_B} \quad (2.4)$$

where the following scaling factors and exposure times are used to normalize the spectra and ensure accurate background subtraction:

- a_D, a_B : Area scaling factors to account for differences in the area of the detector or source regions
- b_D, b_B : Background scaling factors to adjust the relative contribution of the background
- t_D, t_B : Exposure times for the observed data and background measurements, respectively

2.1. Spectral Study of Supersoft X-ray Sources

Instrument response function: The instrument response function $R(I, E)$ maps the detector channels to the corresponding energy bins. It characterizes how the instrument responds to different energies. The instrument response function is typically a continuous function that can be discretized for computational efficiency. In order to do so, the energy range is divided into bins E_j . The corresponding response matrix elements $R_D(I, j)$ are calculated by integrating the continuous function $R(I, E)$ over each energy bin as given in equation (2.5):

$$R_D(I, j) = \frac{1}{E_j - E_{j-1}} \int_{E_{j-1}}^{E_j} R(I, E) dE \quad (2.5)$$

This discretization process transforms the continuous function into a matrix, which is suited for numerical calculations.

Auxiliary response function: In some cases, additional effects such as energy redistribution within the detector can be accounted for by introducing an auxiliary response function $A_D(j)$. This function modifies the instrument response matrix elements as follows:

$$R_D(I, j) \rightarrow R_D(I, j) \cdot A_D(j)$$

The auxiliary response function effectively redistributes the counts between energy bins, capturing the effects of energy-dependent phenomena within the detector. This represents the *redistribution matrix function* (RMF).

Model spectra discretization: Similar to the instrument response function, the model spectra $M(E)$ are also discretized to match the energy bins used for the analysis. This discretization is performed by integrating the model spectra over the corresponding energy bins:

$$M_D(j) = \int_{E_{j-1}}^{E_j} M(E) dE \quad (2.6)$$

The discretized model spectra $M_D(j)$ can then be used in conjunction with the instrument response matrix and observed data to calculate the predicted count spectrum.

2.1. Spectral Study of Supersoft X-ray Sources

2.1.2.3 Predicted Spectrum and Fit Statistic

The predicted count spectrum $C_P(I)$ represents the expected counts in each detector channel based on the chosen model and the instrument response. It is calculated by multiplying the instrument response matrix $R_D(I, j)$ with the model spectrum $M_D(j)$ as per equation (eqn:spec-predcounts):

$$C_P(I) = R_D(I, j) \cdot M_D(j) \quad (2.7)$$

This matrix multiplication effectively applies the instrument response to the model spectrum, accounting for the detector's sensitivity and energy resolution. The resulting predicted count spectrum provides a theoretical expectation of the observed counts, assuming the model accurately describes the underlying physical processes.

To quantify the agreement between the predicted count spectrum and the observed data, the fit statistic χ^2 is computed in a discretized form as given in equation ():

$$\chi^2 = \sum_I [C(I) - C_P(I)]^2 \quad (2.8)$$

2.1.2.4 General Fitting Framework

Spectral curve fitting can be formulated as an optimization problem, where the goal is to find the optimal parameters that minimize an objective function S . Such an optimization problem may be stated in general as follows:

Given a set of spectra $\mathbb{C}(I)$, each supplied as a function of detector channels, a set of theoretical models $\mathbb{M}(E)_j$, each expressed in terms of a vector of energies together with a set of functions $\mathbb{R}(I, E)_j$ that map the channels to the energies, minimize an objective function S of $\mathbb{C}, \mathbb{R}(I, E)_j, \mathbb{M}(E)_j$ using a fitting algorithm. The objective function S may be expressed as per equation ():

$$S = S(\mathbb{C}, \sum_k \mathbb{M}_j^k \cdot \mathbb{R}_j^k) \quad (2.9)$$

In the context of the spectral fitting of SSS, the objective function which is to be optimized is the χ^2 statistic. The *Levenberg-Marquardt algorithm* is a widely used optimization technique

2.1. Spectral Study of Supersoft X-ray Sources

for minimizing the χ^2 statistic. It combines the features of the steepest descent and Gauss-Newton methods, providing a robust and efficient approach to finding the optimal parameter values.

The Levenberg-Marquardt algorithm iteratively updates the parameter values based on the gradient of the χ^2 function and a damping parameter λ . The damping parameter controls the balance between the steepest descent and Gauss-Newton methods, allowing the algorithm to adapt to different optimization landscapes. By iteratively adjusting the parameters and evaluating the χ^2 statistic, the Levenberg-Marquardt algorithm effectively finds the parameter values that minimize the objective function and yield the best fit between the model and the observed data.

2.1.3 Implementation of Spectral Fitting using XSPEC

Here we outline the methodology employed for the spectral analysis of SSS using the XSPEC software. The analysis involves fitting spectral models to data extracted from various X-ray observatories to determine the physical properties of these sources. By following these steps, the spectral analysis of SSS can provide valuable insights into their physical properties, including effective temperatures, column densities, and the presence of elemental absorption features.

2.1.3.1 Data Reduction and Model Selection

- I. A grid of viable spectral models is established to explore different physical scenarios and potential fits to the data.
- II. The raw event files are processed to extract four (or three) FITS files:
 - i. the source spectrum
 - ii. the background spectrum
 - iii. the response matrix file
 - iv. the auxiliary response file (if available).
- III. The extracted FITS files are combined into a spectral group using the `grppha` command, specifying the minimum bin size for the spectral data.

2.1. Spectral Study of Supersoft X-ray Sources

2.1.3.2 Spectral Analysis in XSPEC

- I. The grouped data is loaded into XSPEC using the `data` command. The `cpd` command opens the plotting window, and the `plot` command displays the spectrum, allowing for visual inspection and identification of potential issues.
- II. Bad channels and energy bins outside the region of interest are excluded using the `ignore` command.
- III. A multi-component model is defined in XSPEC using the `model` command. This model can include additive and multiplicative components to represent various physical processes.
- IV. For non-local thermodynamic equilibrium (NLTE) components, the `atable` command is used to incorporate appropriate table models.
- V. Elemental abundances and initial values for model parameters are set based on prior knowledge or literature values.
- VI. The `renorm` command is used to renormalize the model, and the `fit` command is employed to fit the model to the grouped spectral data.

2.1.3.3 Model Evaluation and Parameter Estimation

- I. The χ^2 statistic and degrees of freedom are examined to calculate the reduced χ^2 value. Visual inspection of the data, model, residuals, and delchi values helps assess the suitability of the model fit.
- II. If necessary, the model parameters are iteratively modified using manual adjustments or the `steppar` command to achieve the best possible fit. Different models from the grid may be explored if the initial fit is unsatisfactory.
- III. The plot data is saved to ASCII files using the `WData` command of QDP-PLT by extracting directly from the plot window.

2.1.3.4 Result Reporting and Analysis

- I. The fitting statistic of the best-fit model is reported in tabular format.

2.2. Variability Study of Supersoft X-ray Sources

- II. The `error` command is used to calculate the uncertainties in the parameter values. The `parallel` command can be employed to simultaneously compute error ranges for multiple parameters.
- III. The best-fit values of parameters such as effective temperature and column density, along with their error ranges, are reported.
- IV. The absorbed and unabsorbed fluxes are computed using the `cflux` model component.
- V. The source luminosity is calculated using the unabsorbed flux and the constrained distance to the source, as per the Stefan-Boltzmann law.
- VI. The unfolded spectrum is visually inspected to identify the presence of elemental absorption edges. Literature searches are conducted to identify the corresponding elements and their spectral states.
- VII. The plot data is saved to ASCII files using the `WData` command of QDP-PLT by extracting directly from the plot window.
- VIII. The ASCII files containing the plot data are loaded into Python using NumPy and Matplotlib. Relevant plots are created and analyzed to extract additional insights from the spectral data.

2.2 Variability Study of Supersoft X-ray Sources

2.2.1 Introduction to Lomb-Scargle Periodogram

Periodograms are a versatile tool for identifying and quantifying periodic patterns within data. By transforming a time series into the frequency domain, periodograms reveal the frequencies at which the signal exhibits significant power. This spectral analysis is invaluable for detecting periodic components that might be obscured in the original time domain representation.

The height of peaks in a periodogram provides an estimate of the amplitude of the corresponding periodic component. This allows for a quantitative assessment of the strength of periodic signals. Additionally, the frequency of these peaks directly indicates the periodicity of the underlying patterns.

2.2. Variability Study of Supersoft X-ray Sources

While traditional periodograms are well-suited for uniformly sampled data, their applicability can be limited when dealing with irregularly spaced time series. In such scenario, the Lomb-Scargle periodogram emerges as a viable alternative [28, 29]. Specifically designed to handle unevenly spaced data, the Lomb-Scargle periodogram accounts for gaps in the time series, providing a robust and reliable method for detecting periodic signals.

2.2.2 Theory of Lomb-Scargle Periodogram

2.2.2.1 Uniform and non-uniform sampling of data

Uniform sampling of data: Suppose that $g(t)$ is a *signal function* and $W(t)$ represents the observation *window function*, then the observed signal is represented by the following function:

$$g_{\text{obs}}(t) = g(t)W(t) \quad (2.10)$$

Then, by the Fourier convolution theorem, we have,

$$\mathcal{F}\{g_{\text{obs}}\} = \mathcal{F}\{g(t)\} * \mathcal{F}\{W(t)\} \quad (2.11)$$

In the case of the discrete Fourier transform, where the continuous signal is sampled at regular intervals, the window function becomes a Dirac comb, which is a regular grid of Dirac delta functions with spacing of Δt , i.e.

$$W(t) = \text{III}_{\Delta t}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t) \quad (2.12)$$

Hence, equation 2.10 becomes

$$g_{\text{obs}}(t) = g(t)\text{III}_{\Delta t}(t) \quad (2.13)$$

2.2. Variability Study of Supersoft X-ray Sources

Taking the Fourier transform of equation (2.13), we can have

$$\begin{aligned}
\hat{g}_{\text{obs}}(f) &= \int_{-\infty}^{\infty} g_{\text{obs}}(t) e^{-2\pi i f t} dt \\
&= \int_{-\infty}^{\infty} g(t) \text{III}_{\Delta t}(t) e^{-2\pi i f t} dt \\
&= \int_{-\infty}^{\infty} g(t) \left\{ \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t) \right\} e^{-2\pi i f t} dt \\
&= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t - n\Delta t) g(t) e^{-2\pi i f t} dt \\
\implies \hat{g}_{\text{obs}}(f) &= \sum_{n=-\infty}^{\infty} g(n\Delta t) e^{-2\pi i f t}
\end{aligned} \tag{2.14}$$

In case of a finite number N of samples, we can express $g(n\Delta t) = g_n$ and so equation (2.14) becomes

$$\hat{g}_{\text{obs}}(f) = \sum_{n=0}^N g_n e^{-2\pi i f n \Delta t} \tag{2.15}$$

As per the Nyquist limit, the frequency range of relevance is $0 \leq f \leq \frac{1}{\Delta t}$. This range can be evenly spaced into N intervals with

$$\begin{aligned}
\Delta f &= \frac{1/\Delta t}{N} = \frac{1}{N\Delta t} \\
\implies \Delta f \Delta t &= \frac{1}{N}
\end{aligned}$$

Let the frequency range be spaced by Δf , i.e. $f = k\Delta f$, where k is an integer. Then from equation (2.15), we obtain

$$\begin{aligned}
\hat{g}_{\text{obs}}(k\Delta f) &= \sum_{n=0}^N g_n e^{-2\pi i k \Delta f n \Delta t} \\
\implies \hat{g}_k &= \sum_{n=0}^N g_n e^{-2\pi i k n / N}
\end{aligned} \tag{2.16}$$

Here, equation (2.16) gives the expression for the implementation of the discrete Fourier transform.

2.2. Variability Study of Supersoft X-ray Sources

From the DFT in equation (2.16), we can compute the *classical periodogram* or *Schuster periodogram* [30] as $\frac{1}{N}$ times the Fourier power spectrum for a continuous signal observed with uniform sampling using a Dirac comb:

$$P_s(f) = \frac{1}{N} \left| \sum_{n=1}^N g_n e^{-2\pi i f t_n} \right|^2 \quad (2.17)$$

It must be clarified here that the periodogram is a statistic which is computed from the data and it is used to estimate the power spectrum of the underlying continuous signal of interest.

In real-world observational fields such as astronomy, data collection is often subjected to a range of constraints and variability, leading to non-uniform sampling. Unlike controlled experiments where conditions can be held constant, astronomical observations by space-based observatories are influenced by a variety of factors:

- i. *Orbital constraints*: Space-based telescopes are often in specific orbits around the Earth, the Moon, or even the Sun. These orbits dictate observation windows and create periods when the telescope cannot point at certain objects due to the Sun, Earth, or other celestial bodies obstructing the line of sight.
- ii. *Sun avoidance regions*: To prevent damage to sensitive instruments, space-based telescopes typically have sun avoidance zones where they cannot observe. This limits the ability to track certain objects continuously, leading to gaps in data collection.
- iii. *Limited data storage and downlink bandwidth*: Space telescopes have finite onboard data storage and are dependent on ground stations for data downlink. Periods when the telescope is out of communication range or when storage reaches capacity can result in interruptions in data collection.
- iv. *Thermal constraints and cooling periods*: Space-based telescopes equipped with infrared sensors or other heat-sensitive instruments may require cooling periods or specific thermal management strategies. This can restrict observation times and introduce non-uniform sampling in the data.
- v. *Instrument maintenance and calibration*: Routine instrument calibration and maintenance activities can lead to temporary suspension of scientific observations, causing gaps in data collection.

2.2. Variability Study of Supersoft X-ray Sources

These factors create unevenly spaced data points that pose significant challenges for the discrete Fourier transform, which is built on the assumption of uniform sampling intervals.

Non-uniform sampling of data: In case the observed data is a signal that is sampled over a set of N times, i.e. $\{t_n\}$, and so the window function may be expressed as

$$W_{\{t_n\}}(t) = \sum_{n=1}^N \delta(t - t_n) \quad (2.18)$$

In this scenario, upon applying the window function to the underlying continuous signal, the observed signal is obtained to be of the form:

$$\begin{aligned} g_{\text{obs}}(t) &= g(t)W_{\{t_n\}}(t) \\ \implies g_{\text{obs}}(t) &= \sum_{n=1}^N g(t)\delta(t - t_n) \end{aligned} \quad (2.19)$$

The Fourier transform of the observed signal is obtained using the convolution theorem to be as follows

$$\mathcal{F}\{g_{\text{obs}}\} = \mathcal{F}\{g(t)\} * \mathcal{F}\{W_{\{t_n\}}(t)\} \quad (2.20)$$

However, in this case, the Fourier transform of the window function $W_{\{t_n\}}(t)$ would not be a Dirac comb. Rather, it would be noisier because the observation times have undergone randomization. Even though this noise may be reduced to a certain degree by increasing the number of observations, it would still persist. Also, increasing the number of observations might not be practically feasible in situation, such as in astronomy.

2.2. Variability Study of Supersoft X-ray Sources

2.2.2.2 Mathematical Formulation of Lomb-Scargle Periodogram

Using Euler's formula, the classical periodogram given in equation (2.17) can be re-cast in the following manner:

$$P(f) = \frac{1}{N} \left| \sum_{n=1}^N g_n e^{-2\pi i f t_n} \right|^2$$

$$\implies P(f) = \frac{1}{N} \left[\left\{ \sum_{n=1}^N g_n \cos(2\pi f t_n) \right\}^2 + \left\{ \sum_{n=1}^N g_n \sin(2\pi f t_n) \right\}^2 \right] \quad (2.21)$$

When the classical periodogram is applied to uniformly-sampled Gaussian noise, its distribution converges asymptotically to a χ^2 distribution with known degrees of freedom, this property being contingent upon the assumption of uniform sampling. For non-uniformly sampled data, the distribution of the periodogram becomes considerably more complex, and a closed-form analytical expression is generally intractable.

To address this issue, Scargle generalized the periodogram in equation (2.21) as follows [29]:

$$P(f) = \frac{A^2}{2} \left\{ \sum_{n=1}^N g_n \cos[2\pi f(t_n - \tau)] \right\}^2 + \frac{B^2}{2} \left\{ \sum_{n=1}^N g_n \sin[2\pi f(t_n - \tau)] \right\}^2 \quad (2.22)$$

where A , B and τ are arbitrary functions of the frequencies f and the observing times $\{t_i\}$. Scargle also showed that one can choose unique forms of A , B and τ in a manner such that we have the periodogram in equation (2.22)

- a) reducing to the classical form in equation (2.17) in case the observations are uniformly sampled.
- b) having statistics which are analytically computable.
- c) becoming insensitive to global time-shifts in data.

2.2. Variability Study of Supersoft X-ray Sources

The functions of A , B and τ that provide the above properties are

$$A = \frac{1}{\sqrt{\sum_n \cos^2 [2\pi f(t_n - \tau)]}}$$
$$B = \frac{1}{\sqrt{\sum_n \sin^2 [2\pi f(t_n - \tau)]}}$$
$$\tau = \frac{1}{4\pi f} \tan^{-1} \left\{ \frac{\sum_n \sin [4\pi f t_n]}{\sum_n \cos [4\pi f t_n]} \right\}$$

In the above, the function τ ensures the time-shift invariance of the Lomb-Scargle periodogram. A notable feature of the Lomb-Scargle periodogram is its equivalence to the result obtained from fitting a simple sinusoidal model to the data at each frequency f and constructing a periodogram from the χ^2 goodness-of-fit statistic at each frequency. This approach allows the periodogram to be viewed as a measure of how well a sine wave of a given frequency fits the observational data. In this context, the introduction of the τ shift is crucial, as it serves to orthogonalize the normal equations used in the least squares fitting process. This orthogonalization ensures that the fitting procedure is unbiased and more computationally stable. This deep connection between traditional Fourier analysis and least-squares analysis has led to the widespread adoption of the term *Lomb-Scargle periodogram* to describe this methodology, even though variants of this approach were utilized prior to Lomb and Scargle's formalization.

2.2.2.3 Uncertainties in Periodogram Results

A critical aspect of reporting results derived from the Lomb-Scargle periodogram is the quantification of uncertainty associated with the estimated period. While traditional approaches often rely on error bars to express uncertainty, such a metric may be less meaningful in the context of Lomb-Scargle periodograms. The primary concern with periodograms is often the presence of false peaks or aliases, rather than a continuous distribution of potential period values around a true estimate. This disjointed nature of uncertainty necessitates a more nuanced approach to quantifying the reliability of the estimated period.

Peak width and frequency precision: The presence of a peak within the Lomb-Scargle periodogram, characterized by its width and height, signifies the existence of a periodic signal. In the Fourier domain, the precision with which a peak's frequency can be identified

2.2. Variability Study of Supersoft X-ray Sources

is directly correlated with its width. A common metric for quantifying this width is the half-width at half-maximum, denoted by $f_{1/2} \approx T$.

A more rigorous framework for assessing uncertainty can be established within the context of least squares analysis. In this interpretation, the inverse of the peak's curvature is directly related to the uncertainty in the estimated frequency. This perspective aligns with the Bayesian approach, which involves fitting a Gaussian curve to the exponentiated peak. The Bayesian uncertainty in the frequency estimate is influenced by both the number of samples N , and the average signal-to-noise ratio Σ . The scaling of this uncertainty can be approximated by the standard deviation of the estimated frequency:

$$\sigma_f \approx \sqrt{\frac{2}{N\Sigma^2}} \quad (2.23)$$

This dependence arise from the fact that the Bayesian uncertainty is linked to the width of the exponentiated periodogram, which in turn is influenced by the height of the peak P_{\max} .

False alarm probability: A more pertinent metric for assessing the uncertainty associated with Lomb-Scargle periodogram results is the relative height of the signal peak to the background noise. The spurious peaks that arise in the periodogram are influenced by both the sample size N , and the signal-to-noise ratio. For scenarios with limited data and low signal-to-noise, these spurious peaks can rival the height of the true signal peak, making it challenging to distinguish between genuine and chance-driven features.

The False Alarm Probability (FAP) serves as a standard approach for quantifying the significance of a periodogram peak. It represents the probability that a dataset devoid of any true signal would, due to random fluctuations in the noise, produce a peak of comparable magnitude. Scargle (1982) demonstrated that for pure Gaussian noise at any given frequency f_0 , the unnormalized Lomb-Scargle periodogram statistic $Z = P(f_0)$, follows a χ^2 distribution with two degrees of freedom. Consequently, the cumulative probability of observing a periodogram value less than Z can be expressed as:

$$P_{\text{single}}(Z) = 1 - \exp(-Z) \quad (2.24)$$

Equation (2.24) provides a quantitative measure of the likelihood that a given peak is due to chance.

2.2. Variability Study of Supersoft X-ray Sources

The FAP measures the likelihood that a peak in the periodogram is due to chance, rather than a genuine periodic signal. A low FAP means that the observed peak is unlikely to be a false alarm. Therefore, for a periodic signal to be considered significant, the corresponding FAP should be small. This suggests that the peak is highly likely to represent a real periodic component within the data.

The Baluev method: Baluev (2008) introduced a refined approximation for the False Alarm Probability (FAP) in the context of Lomb-Scargle periodograms [31]. This approximation, given by:

$$\text{FAP}(z) \approx 1 - P_{\text{single}}(z)e^{-\tau(z)} \quad (2.25)$$

provides a conservative upper bound, even for time series characterized by highly structured window functions. Here, P_{single} denotes the cumulative probability of observing a periodogram value less than Z under the null hypothesis, while $\tau(z)$ is a correction factor defined as $\tau(z) \approx W(1 - z)^{(N-4)/2} \sqrt{z}$, where W represents the effective width of the observing window in units of the maximum frequency and N is the total number of data points.

While this approximation is not exact, it offers a reliable upper bound for alias-free periodograms. Notably, it maintains a reasonable degree of accuracy even for more realistic survey windows that exhibit non-trivial structure. The Baluev method has become the standard algorithm for computing the FAP in many software implementations of the Lomb-Scargle periodogram.

2.2.3 Implementation of Lomb-Scargle Periodogram

Here we outline the methodology we have employed to investigate the variability characteristics of lightcurves extracted from SSS across two energy ranges: 0.2-1.0 keV and 0.2-12.0 keV. The analysis leverages the Lomb-Scargle periodogram to identify and characterize periodic signals within the lightcurves. By systematically applying these steps to the lightcurves of SSS at the two specified energy ranges, this methodology enables a comprehensive characterization of their variability patterns. The combination of Lomb-Scargle periodogram analysis, statistical significance evaluation, phase-folding, and sinusoidal fitting provides a

2.2. Variability Study of Supersoft X-ray Sources

robust framework for uncovering periodic signals and quantifying their properties within the lightcurve data.

2.2.3.1 Data Acquisition and Preprocessing

- I. Google Drive is mounted within Google Colab to facilitate seamless access to the relevant FITS files containing the lightcurve data.
- II. The `astropy.io` module's `fits.open()` function is used to open the FITS files and extract the lightcurve data.
- III. The extracted lightcurve information is converted into NumPy arrays for efficient numerical operations.
- IV. Visual inspection of the lightcurve is performed to identify potential outliers. These outliers, if present, are removed using NumPy masks to ensure they don't influence the subsequent analysis. The cleaned lightcurve is then visualized.

2.2.3.2 Timing Analysis using Lomb-Scargle Periodogram

- I. A `LombScargle` object is created using the `astropy.timeseries` module. This object encapsulates the time and rate arrays extracted from the lightcurve to compute the Lomb-Scargle periodogram.
- II. The calculated Lomb-Scargle periodogram is utilized to retrieve the corresponding frequency and power arrays using the `autopower()` method of the `LombScargle` object.
- III. The Lomb-Scargle periodogram is displayed graphically. Prominent peaks are visually identified for further analysis.
- IV. Peaks corresponding to harmonics associated with the observation window and the time bin size are disregarded, as they do not represent genuine periodicities within the lightcurve data. Rather these are artefacts due to the observation cadence and the time binning process.

2.2. Variability Study of Supersoft X-ray Sources

2.2.3.3 Statistical Significance and Phase-Folded Lightcurves

- I. The `false_alarm_probability()` method of the `LombScargle` object is employed to compute the FAP for the identified peaks. This metric quantifies the probability of observing such peaks arising purely from random noise.
- II. Employing the periods corresponding to the identified significant peaks, phase-folded lightcurves are constructed. These lightcurves represent the binned data points as a function of their phase within a complete cycle (0 to 2π radians).
- III. The `curve_fit()` function from the `scipy.optimize` module is used to fit a sinusoidal function of the following form to each phase-folded lightcurve:

$$C = A \sin(2\pi\nu t + \phi) + C_0$$

Here, A represents the amplitude, ν the frequency, ϕ the phase offset, and C_0 the DC offset. The best-fit values for these parameters are obtained through the fitting process.

2.2.3.4 Visualization and Result Presentation

- I. The phase-folded lightcurve along with the superimposed best-fit sinusoid are plotted to visually assess if the calculated variability aligns with the observed data points.
- II. The analysis results are consolidated into a table, including the following:
 - i. Peak frequencies (in mHz)
 - ii. Corresponding periods (in ks)
 - iii. Power densities
 - iv. False alarm probabilities (FAP)
 - v. Best-fit parameters from sinusoidal fitting (A, ϕ, C_0)
 - vi. Amplitude-to-mean ratio (AMR): Calculated as $(A/C_0) \times 100\%$, indicating the strength of the periodic signal relative to the mean lightcurve level.

CHAPTER 3

MULTI-OBSERVATORY SPECTRAL ANALYSIS OF SUPERSOFT X-RAY SOURCES

Contents

3.1	Literature Review	41
3.2	Journal of Observations	44
3.3	Data Reduction and Analysis	45
3.3.1	XMM-Newton EPIC-pn Data Reduction	46
3.3.2	ASCA SIS1 Data Reduction	47
3.3.3	Chandra ACIS Data Reduction	48
3.3.4	NICER XTI Data Reduction	49
3.4	Results	50
3.4.1	Observed Count Rates	50
3.4.2	NLTE Continuum Model	52
3.4.3	Unfolded Spectra from Best-fit Model	53
3.4.4	Luminosity versus Effective Temperature	54
3.4.5	Presence of Elemental Absorption Edges	54
3.5	Discussion	58
3.5.1	Best-fit Continuum Model	58
3.5.2	NLTE Pure H Model Atmosphere	59
3.5.3	Inferences from Results	65
3.5.4	Relative Strengths of Absorption Edges	69

CHAPTER 3. MULTI-OBSERVATORY SPECTRAL ANALYSIS OF SUPERSOFT X-RAY SOURCES

Abstract

In this chapter, we present a comprehensive study on the spectral characteristics and parameters of the supersoft X-ray source RX J0925.7-4758. We utilize a dataset comprising of six observations conducted by four different space observatories, namely ASCA, Chandra, XMM-Newton and NICER, spanning a time period of 25 years. Our objective is to identify a robust NLTE spectral model that can provide an acceptable fit to the continuum spectrum of RX J0925.7-4758, which had not been performed earlier as per current literature, thereby enhancing our understanding of its astrophysical nature. The study employs rigorous spectral modeling and comparative analysis to establish a framework for characterizing the continuum spectrum of RX J0925.7-4758. The study also discusses the analysis of the supersoft spectrum of RX J0925.7-4758, including the challenges in reproducing its spectra using NLTE model atmospheres. The best-fit model consists of a pure hydrogen NLTE model with effective gravity of 7, in conjunction with photoelectric absorption, ISM absorption and edge components. The effective temperature is found to be approximately of the order of $\sim 10^5$ K for all six observations, suggesting the presence of a hot accretion disk surrounding the white dwarf, which accretes matter from a companion main-sequence star. The study also analyzes the relative strengths of absorption edges in the spectrum and notes inconsistencies in the NICER observations. The current work demonstrates that while the continuum spectrum of RX J0925.7-4758 can be fitted satisfactorily, the same cannot be said to be the case for its high-resolution grating spectra. Therefore, it makes a case for the need to explore and develop methods that study such latest grating spectra, so as to obtain a more nuanced view of the astrophysics involving RX J0925.7-4758.

3.1 Literature Review

Supersoft X-ray sources (SSS) represent an important class of celestial objects. They were initially recognized as a distinct class of intrinsically luminous X-ray sources by Trümper *et al.* [32], Greiner *et al.* [33], and Kahabka *et al.* [20]. These sources are classified based on their X-ray luminosities, which are typically around the Eddington limit ($\sim 10^{38}$ erg s $^{-1}$), indicating their exceptionally high brightness. However, their defining feature is their extremely soft X-ray spectra, with weak or negligible emission beyond ~ 1 keV and effective blackbody temperature no more than ~ 100 eV (or $\lesssim 1200$ kK) [11]. The understanding of the spectral characteristics of SSS promises to pave the way for a deeper exploration of their role in the broader astrophysical landscape.

SSS were initially observed by the Einstein Observatory during a soft X-ray survey of the Large Magellanic Cloud (LMC) by Long et al. in 1981 [18]. Over time, hundreds of objects exhibiting similar characteristics have been documented, many of which are considered candidates or confirmed members of this class. The catalog of SSS has expanded significantly, now encompassing sources not only within the LMC but also various other celestial bodies, including the Milky Way (MW), Small Magellanic Cloud (SMC), and Andromeda (M31) and numerous other galaxies [34, 35, 36, 37, 38, 39, 40, 41, 42].

The relatively fewer detections of SSS within our galaxy, compared to other galaxies, may be attributed to significant interstellar extinction of soft X-rays. Situated at the edge of the MW and along its galactic plane, soft X-rays emitted by galactic SSS face considerable interstellar absorption. Suggestions have been made that SSS within the galactic plane must be within a distance of approximately 1 kpc to remain observable. Beyond this distance, interstellar absorption becomes sufficiently pronounced to render them undetectable [43].

The optical spectra of the SSS in the Magellanic Cloud galaxies exhibit similarities with those of low-mass X-ray binaries – characterized by strong emission lines such as He II at 4686 Å and hydrogen Balmer lines. Subsequent numerical calculations suggested that SSS could result from near-Eddington accretion onto neutron stars [44], although subsequent analyses favoured white dwarfs as the likely compact objects responsible for the emission of supersoft X-rays [17].

Employing Stefan-Boltzmann's law with typical luminosity and effective temperature values for SSS, the estimated radius of the emitting object aligns with that of a white dwarf,

3.1. Literature Review

supporting the hypothesis of accretion onto white dwarfs as the source of supersoft X-rays, akin to accreting neutron stars and black holes in classical X-ray binaries. Van den Heuvel proposed that white dwarfs with masses in the range $0.7\text{--}1.4 M_{\odot}$ and mass accretion rates $\sim 1\text{--}5 \times 10^{-7} M_{\odot} \text{ yr}^{-1}$ produce supersoft X-rays, assuming the mass-accretor as the white dwarf and the companion as a main-sequence or post-main-sequence star within specific mass ranges [43]. Studies by various groups have explored different types of nuclear burning due to mass accretion on white dwarfs, depending on the thermal history of the white dwarf and conditions required for nuclear ignition, typically involving critical envelope masses sustaining high temperatures ($\sim 10^8 \text{ K}$) and pressures ($\gtrsim 10^{18}\text{--}10^{20} \text{ g cm}^{-3} \text{ s}^{-2}$) for nuclear burning via the CNO cycle [45, 46, 47, 48, 1, 49, 50, 51, 52, 53]. The steady state accretion rate, crucial for understanding the relationship between accretion and nuclear burning, has been investigated in early calculations [54, 51], providing insights into the dynamics of hydrogen-rich matter accretion on white dwarfs. For a hydrogen-rich system, the steady state accretion rate was given to be [55]

$$\dot{M}_{\text{steady}} \sim 3.7 \times 10^{-7} \left(\frac{M_{\text{WD}}}{M_{\odot}} - 0.4 \right) M_{\odot} \text{ yr}^{-1} \quad (3.1)$$

The particular luminous galactic SSS known as MR Vel, and referred to as RX J0925.7-4758, was discovered by Motch *et al.* [56] in the ROSAT Galactic Plane Survey (RGPS), which is defined as the $|b| \leq 20^\circ$ region of the ROSAT All Sky Survey. In the J2000 frame, the right ascension and declination of RX J0925.7-4758 is 141.44042 and -47.96972 ($\alpha=09^{\text{h}}25^{\text{m}}46.00^{\text{s}}$, $\delta=-47^{\circ}58'17.4'$: as resolved by Simbad¹).

RX J0925.7-4758 was the brightest SSS candidate source in RGPS, with ROSAT PSPC hardness ratios of $HR1 = 0.96 \pm 0.03$ and $HR2 = -0.69 \pm 0.03$. Fitting a blackbody to the ROSAT observations, a hydrogen column density in the range $n_H = (1.4\text{--}3.7) \times 10^{22} \text{ cm}^{-2}$ can be obtained. There is a considerable amount of uncertainty in current literature about its distance, with estimates ranging from 1 kpc to 10 kpc. Consulting Gaia Data Release 3², one can find negligible parallax. This fact along with its high luminosity suggests that its distance is likely to be > 5 kpc.

Hartmann *et al.* applied non-local thermodynamic equilibrium (NLTE) models, which included metal line opacities, to the spectrum extracted from the observations by BeppoSAX

¹<http://simbad.u-strasbg.fr/simbad/>

²<https://www.cosmos.esa.int/web/gaia/data-release-3>

3.1. Literature Review

LECS of RX J0925 on January 25-26 1997 [57]. They found that if a single model component is assumed for RX J0925.7-4758, a large discrepancy is observed between the model and data above 1.19 keV. The emission above ~ 1.2 kev (i.e. Ne IX edge) can be accounted for by adding another spectral model component, namely collisional ionization equilibrium.

Higher resolution data obtained using the grating instruments on-board the Chandra and XMM-Newton observatories revealed complex structures in the spectra of RX J0925.7-4758. Such spectra show the presence of P Cygni profiles of Fe XVII and O VIII, which typically arise in a wind. Earlier, Beard et al. (2002) and Motch et al. (2002) had come to the conclusion that the RX J0925.7-4758 spectra, as observed by the Chandra HETGS and the XMM-Newton RGS, cannot be reproduced by LTE or NLTE model atmospheres [58, 59], even though there is little clarity on the reason for this. Obtaining an acceptable fit for RX J0925.7-4758 spectrum assumes crucial importance at this juncture. In the absence of a proper model describing the emission spectrum, it becomes impossible to calculate its parameters such as effective temperature and luminosity.

In the present work, our primary objective was to identify a robust spectral model capable of providing an acceptable fit to the continuum spectrum of RX J0925.7-4758. To this end, we devised an approach wherein we analyse spectral data for RX J0925.7-4758 obtained using multiple observatories. A motivation for harnessing such multi-observatory science data over an extended period was to mitigate potential biases associated with individual instruments and temporal variations in observational conditions, thereby enabling a systematic examination of the source's spectral characteristics, in an attempt to constrain the underlying physical processes driving the observed supersoft X-ray emission.

The amalgamation of observations from diverse space observatories offered unique insights into the spectral evolution of the supersoft X-ray source over time, shedding light on its dynamic behavior and emission properties. Through rigorous spectral modeling and comparative analysis, we made an attempt to establish a robust framework for characterizing the continuum spectrum of the supersoft source, thereby enhancing our understanding of its astrophysical nature and evolutionary trends. A comprehensive investigation into recurrent SSS like RX J0925.7-4758 is paramount due to the emerging recognition of SSS as progenitors of type Ia supernovae, which serve as crucial standard candles in cosmology. By delving deeper into the astrophysical mechanisms governing SSS, we can potentially enhance our ability to make precise predictions regarding the occurrence of type Ia supernovae and subsequently refine cosmological distance measurements.

3.2 Journal of Observations

A summary of the observations used in this study, which included science data from Japan’s ASCA [60], NASA’s Chandra [58], ESA’s XMM-Newton [59] and ISS’ NICER [61] observatories, is presented in table 3.1.

Table 3.1: Journal of observations

Observation (Obs. ID)	Date (yyyy-mm-dd)	Observatory	Instrument	MJD [†] (ks)	Exposure [‡] (keV)	Region (keV)
43036000	1994-12-22	ASCA	SIS1	49708.56	20.58	0.20–1.00
644	2000-11-14	Chandra	ACIS	51862.92	57.40	0.40–1.00
0111150101	2000-12-16	XMM-Newton	EPIC-pn	51894.46	61.10	0.30–1.00
2611020101	2019-05-18	NICER	XTI	58621.90	2.53	0.40–1.00
2611020102	2019-05-19	NICER	XTI	58622.03	8.57	0.45–1.00
2611020103	2019-05-19	NICER	XTI	58623.00	9.91	0.45–1.00

[†]Modified Julian Date

[‡]From HEASARC archival query results

3.3. Data Reduction and Analysis

During the course of our investigation of the continuum spectrum of RX J0925.7-4758, we leveraged a comprehensive dataset comprising six observations conducted by four distinct space observatories spanning the years 1994 to 2019. We extracted spectral data within the range 0.2 – 1.0 keV from each observation in this dataset. This afforded us the opportunity to apply a consistent spectral analysis approach across a broad temporal range and disparate instrumentation platforms.

3.3 Data Reduction and Analysis

The raw data pertaining to all of the observations were downloaded using the online archival query interface at the *High Energy Astrophysics Science Archive Research Center* (HEASARC)³. The data was reduced finally to *Flexible Image Transport System* (FITS)⁴ format using the software tools for the corresponding observatory with recommended settings in the relevant data analysis threads, wherever available. For the final analysis, a set of four files were generated for each of the observations. As a standard practice, these files were given the extensions as per a convention adopted by the authors as follows:

1. Source spectrum, with file extension .src
2. Background spectrum, with file extension .bkg
3. Redistribution matrix file (RMF), with file extension .rmf
4. Ancillary response file (ARF), with file extension .arf

The above files were grouped using the FTOOLS task grphha⁵ to have an appropriate minimum number of counts per bin. The resulting spectrum sets were analysed using *XSPEC* version 12.13.1.

³<https://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl>

⁴https://fits.gsfc.nasa.gov/standard40/fits_standard40aa-1e.pdf

⁵https://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/docs/memos/cal_sw_93_010/cal_sw_93_010.pdf

3.3. Data Reduction and Analysis

3.3.1 XMM-Newton EPIC-pn Data Reduction

The SSS RX J0925.7-4758 was observed by all the instruments, viz. EPIC-MOS 1, EPIC-MOS 2, EPIC-pn and RGS, on-board ESA's XMM-Newton observatory for ~ 52 ks on 16 December 2000. Whereas, we had retrieved data from all the instruments, we decided to use only the EPIC-pn data. The reason for this being two-fold:

- i. The spectral region of interest is of the lowest energies detectable by EPIC, and the pn detector has a comparatively higher sensitivity than the MOS detectors at lower energies [62, 63].
- ii. Currently, the high resolution grating spectra (such as those produced by the RGS) yield unacceptable fits to atmosphere models of SSS. Also, no atmosphere model has yet been able to reproduce all the details in such grating spectra [64].

As per recommendations by the XMM-Newton SOC, the data analysis was restricted to energies above 0.2 keV⁶. The data reduction procedures were performed using the *XMM-Newton Science Analysis System* (SAS) version 21.0.0.

The Observation Data Files (ODF) were downloaded from HEASARC. In order to prepare the data for processing, we included the instrumental and calibration information by creating a Calibration Index File (CIF), which was up-to-date with the current calibration files (CCF)⁷, and an extended ODF summary file. These were done by running the SAS tasks `cifbuild` and `odfingest` respectively. The ODFs were then reprocessed to generate the calibrated event files using the `epproc`, using the default parameters. The event file for EPIC-pn was filtered using the canned screening set of flags, and by setting `PATTERN==0` to select only single-pixel events in order to maximise energy calibration and resolution. The procedure described by Jethwa et al. (2015) [65] was used to check and find that the spectral distortion and flux loss both $< 0.01\%$, which implied that the pile-up effects could be neglected.

The source photons were identified using DS9 and extracted from a circular region having a radius of 30" which is centred at the RX J0925.7-4758 centroid position, encompassing about

⁶<https://xmmweb.esac.esa.int/docs/documents/CAL-TN-0018.pdf>

⁷<https://www.cosmos.esa.int/web/xmm-newton/current-calibration-files>

3.3. Data Reduction and Analysis

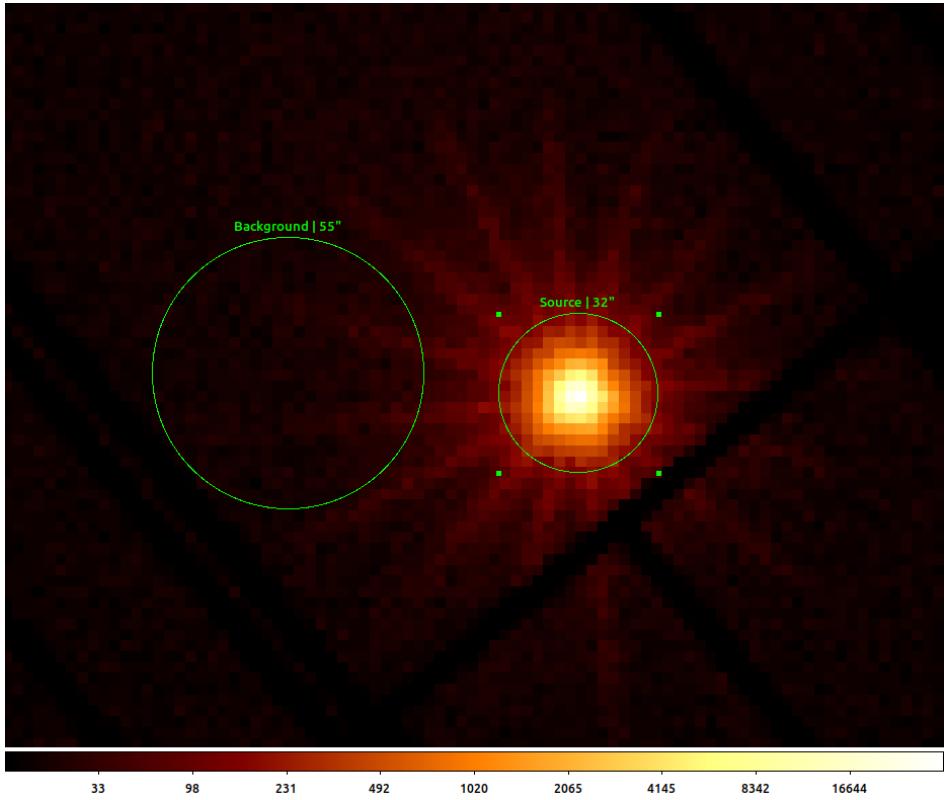


Figure 3.1: Source and background extraction regions for the XMM-Newton observation of RX J0925.7-4758

85-90% of XMM-Newton's telescope's on-axis PSF⁸. For the background photons, first the SAS task `ebkgreg` was executed to obtain an optimum circular background extraction region of radius $\sim 55''$. The resulting source and background extraction regions are displayed in figure 3.1. The RMF and ARF were generated using the standard SAS tasks `rmfgen` and `arfgen`. The spectrum was finally binned to have a minimum of 10 counts/bin using the task `grphha` in order to make it ready for analysis using XSPEC.

3.3.2 ASCA SIS1 Data Reduction

RX J0925.7-4758 was observed by the SIS1 instrument on-board Japan's ASCA observatory for a total duration of ~ 21 ks on 22 December 1994, with the purposes of characterization of its continuum spectral shape and the investigation of absorption edge features [60]. Because

⁸https://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/uhb/onaxisxraypsf.html

3.3. Data Reduction and Analysis

SIS1 has greater effective area for energies $< 1.5 \text{ keV}$ ⁹, the data analysis was restricted to an energy range of 0.2-1.0 keV. The procedures for the extraction of spectra were performed using *XSELECT*¹⁰ version 2.5, a multipurpose tool for filtering event files and generating images, spectra, and light curves made available as part of HEASoft.

The event files were downloaded from HEASARC. They were loaded into XSELECT and the source and background spectra were extracted using a circular region of radius 127" and an annular region of radii 129" and 210" respectively, centred at RX J0925.7-4758 centroid position. The RMF and ARF were generated using the FTOOL tasks *sisrmg*¹¹ and *ascaarf*¹² respectively. The spectrum set was grouped and binned to a minimum of 20 counts/bin using *grppha* so as to analyse using XSPEC.

3.3.3 Chandra ACIS Data Reduction

RX J0925.7-4758 was observed during 14 November 2000 for a duration of $\sim 57 \text{ ks}$ using the High-Energy Transmission Grating Spectrometer (HETGS) on-board NASA's Chandra X-ray Observatory [58]. The photons were detected with the ACIS-S CCD array at the focal plane. The data analysis was performed in the energy range 0.4–1.0 keV, with 0.4 keV being the lower limit of the HETGS+ACIS-S spectrometer combination. The extraction of the spectrum and response files was performed using *CIAO* version 4.10 and the Chandra calibration database *CALDB* version 4.7.7.

All the science data files were downloaded from HEASARC. DS9 was used to identify the source and background regions as being a circular region of radius 0.23' and an annular region of radii 0.28' and 0.57' respectively. The source and background spectra were extracted as FITS files using the task *specextract*¹³ using standard parameters as per the data analysis threads for point-like sources¹⁴. The relevant RMF and ARF were generated using the *mkacisrmf*¹⁵ and *mkarf*¹⁶ tasks respectively. The spectrum set was grouped and binned to have a minimum of 10 counts/bin using the task *grphha* for subsequent analysis using

⁹https://heasarc.gsfc.nasa.gov/docs/asca/newsletters/sis_overview.html

¹⁰<https://heasarc.gsfc.nasa.gov/ftools/xselect/>

¹¹<https://heasarc.gsfc.nasa.gov/lheasoft/ftools/fhelp/sisrmg.html>

¹²<https://heasarc.gsfc.nasa.gov/lheasoft/ftools/fhelp/ascaarf.html>

¹³<https://cxc.cfa.harvard.edu/ciao/ahelp/specextract.html>

¹⁴<https://cxc.cfa.harvard.edu/ciao/threads/pointlike/>

¹⁵<https://cxc.cfa.harvard.edu/ciao/ahelp/mkacisrmf.html>

¹⁶<https://cxc.cfa.harvard.edu/ciao/ahelp/mkarf.html>

3.3. Data Reduction and Analysis

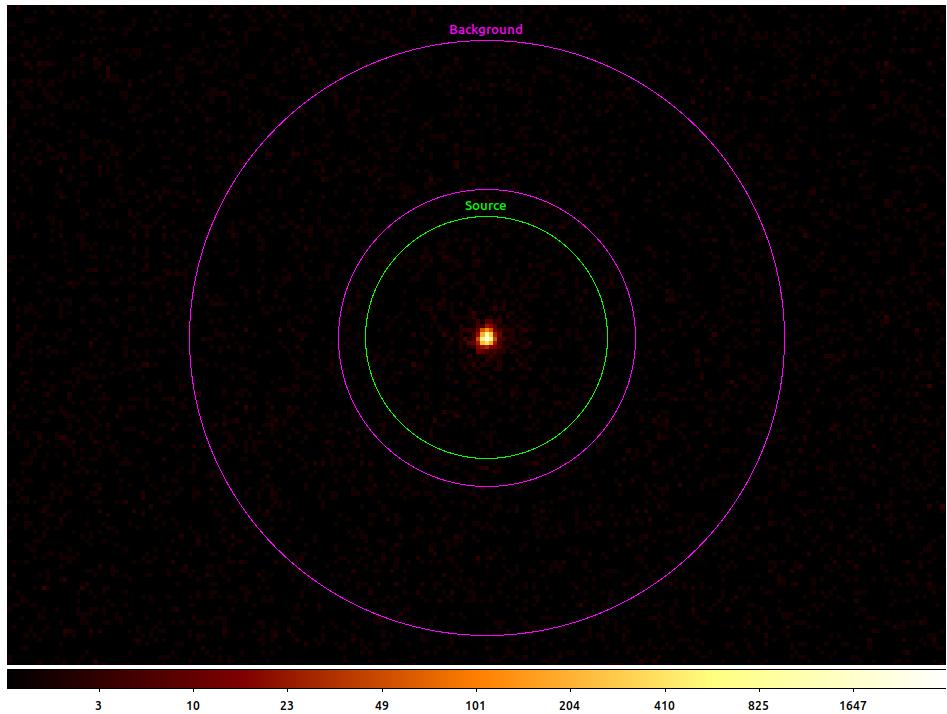


Figure 3.2: Source and background extraction regions for the Chandra observation of RX J0925.7-4758

XSPEC.

3.3.4 NICER XTI Data Reduction

RX J0925.7-4758 was observed by the XTI instrument on-board the NICER observatory on the International Space Station (ISS) for a total duration of ~ 21 ks on three occasions during 18–19 May 2019. All data reduction commands for the science data were performed using NICER-specific tasks are made available with the latest versions of HEASoft¹⁷.

The XTI observation dataset and auxiliary files were downloaded from HEASARC. In order to prepare the data for processing, we set up the remote access of the HEASARC CALDB by following the recommended procedure¹⁸. The cleaned event files were extracted using the `nicerl2` command¹⁹. They were then loaded into XSELECT. This produces the source

¹⁷<https://heasarc.gsfc.nasa.gov/docs/software/heasoft/>

¹⁸https://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/caldb_remote_access.html

¹⁹<https://heasarc.gsfc.nasa.gov/lheasoft/ftools/headas/nicerl2.html>

3.4. Results

and background spectrum files in FITS format. The ARF and RMF were generated with the extracted source spectrum files using the `nicerarf`²⁰ and the `nicerrmf`²¹ commands respectively. The spectrum set was finally grouped and binned to a minimum of 10 counts/bin using `grphha` and made ready for analysis using XSPEC.

3.4 Results

We present, in the following pages, a summary of the results of the analysis of the multi-observatory X-ray data for RX J0925.7-4758, including a comparison of the observed count rates, the stellar parameters calculated from the best-fit NLTE continuum models, the unfolded model spectra and the elemental absorption edges identified.

3.4.1 Observed Count Rates

In figures 3.3a and 3.3b, we present a comparison of count rates derived from the multi-observatory data listed in table 3.1. Figure 3.3a serves to illustrate the variability of flux across different observation epochs, with count rates plotted on a logarithmic scale. This visualization provides insights into the temporal evolution of X-ray emission from RX J0925.7-4758.

Concurrently, in figure 3.3b, the count rates are normalized to the range of 0 to 1 using *min-max normalization*. This visualization accentuates the relative sensitivity of more recent observations to supersoft X-ray photons emitted by RX J0925.7-4758. The sub-plot shows that in recent observations the supersoft X-ray features are discernibly enhanced. This might be suggestive of improved observational capabilities or heightened sensitivity to the emitted X-ray flux.

²⁰<https://heasarc.gsfc.nasa.gov/lheasoft/ftools/headas/nicerarf.html>

²¹<https://heasarc.gsfc.nasa.gov/lheasoft/ftools/headas/nicerrmf.html>

3.4. Results

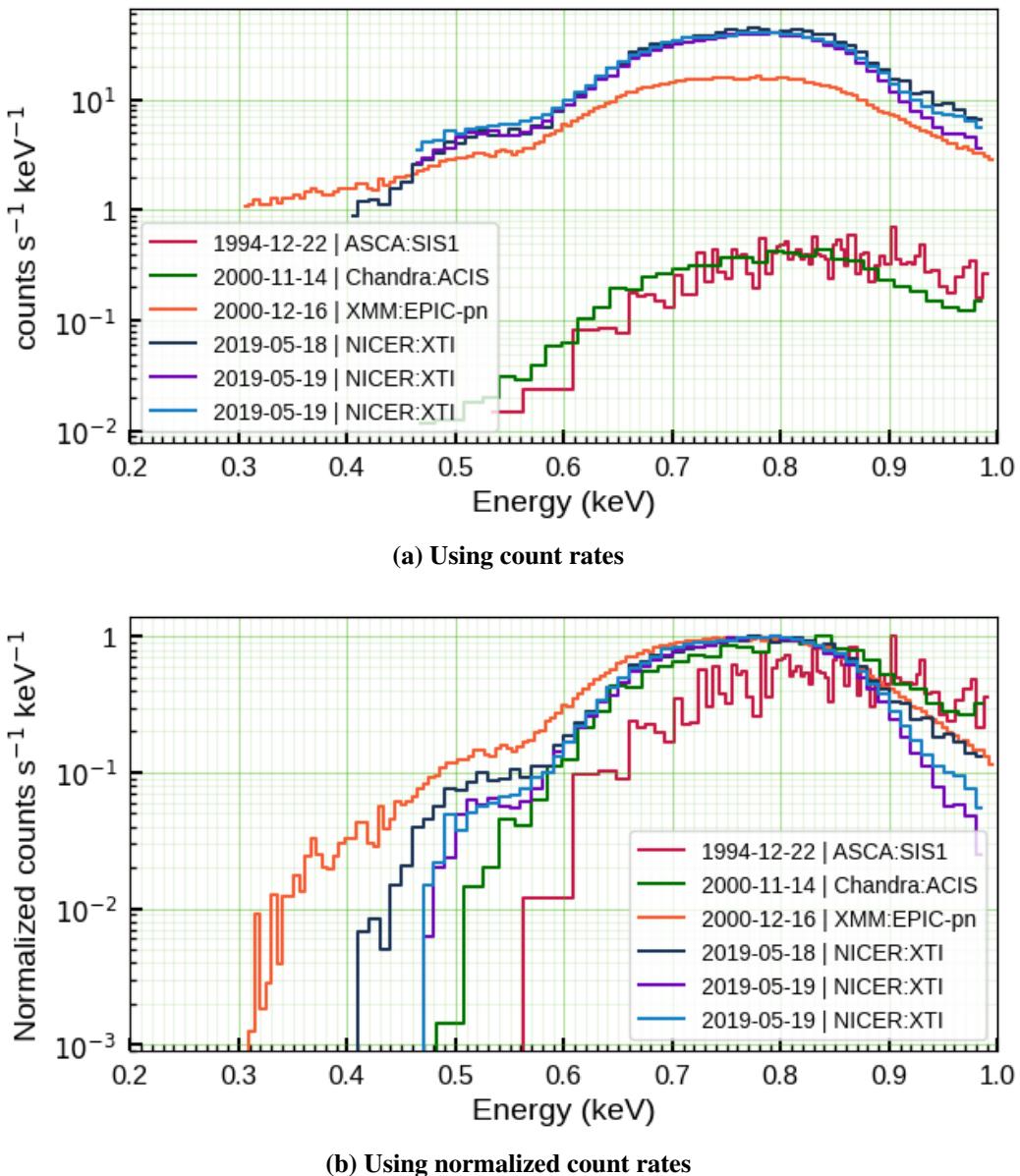


Figure 3.3: Comparison of flux from all observations

Together, these visualizations offer a nuanced understanding of the flux variability and observational sensitivity trends exhibited by RX J0925.7-4758 across different observation epochs, contributing to the broader understanding of its X-ray emission characteristics.

3.4.2 NLTE Continuum Model

In table 3.2, we present the luminosity, effective temperature and column density for RX J0925.7-4758 which are derived after performing the fitting to the NLTE continuum model described in §3.3. We also present the model fit statistics, namely the reduced χ^2 values for each fit. It can be noticed that the same continuum model has been able to provide a good fit to all of the observations in our chosen dataset.

Table 3.2: Parameters of RX J0925.7-4758 derived from continuum NLTE model from multi-observatory observations.

Observatory	Obs. ID	L_* (erg s ⁻¹)	T_{eff} (kK)	n_H ($\times 10^{22}$ cm ⁻²)	$\chi^2/\text{d.o.f}$	χ^2_{reduced}
ASCA	43036000	$2.07^{1.02}_{0.46} \times 10^{41}$	$100.2^{47.4}_{56.2}$	$0.13^{1.36}_{0.00}$	75.4/62	1.22
Chandra	644	$4.38^{24.34}_{2.02} \times 10^{41}$	$96.3^{120.5}_{81.6}$	$1.25^{1.37}_{1.17}$	30.5/28	1.09
XMM-Newton	0111150101	$2.19^{3.74}_{1.96} \times 10^{42}$	$91.7^{100.2}_{86.9}$	$1.17^{1.27}_{1.09}$	183.4/131	1.40
NICER	2611020101	$8.40^{19.16}_{6.77} \times 10^{41}$	$100.8^{10.1}_{90.0}$	$1.22^{1.28}_{1.11}$	65.4/52	1.26
NICER	2611020102	$1.68^{3.37}_{1.41} \times 10^{41}$	$107.9^{13.8}_{100.4}$	$0.46^{0.54}_{0.43}$	82.9/46	1.80
NICER	2611020103	$4.41^{6.81}_{3.42} \times 10^{40}$	$106.9^{10.4}_{101.6}$	$0.36^{0.41}_{0.34}$	83.3/46	1.81

3.4. Results

3.4.3 Unfolded Spectra from Best-fit Model

In figures 3.4a and 3.4b, the unfolded spectrum, which is obtained after fitting the data to the best-fit model, is displayed. The observations reveal features which are indicative of the presence of elemental absorption edges.

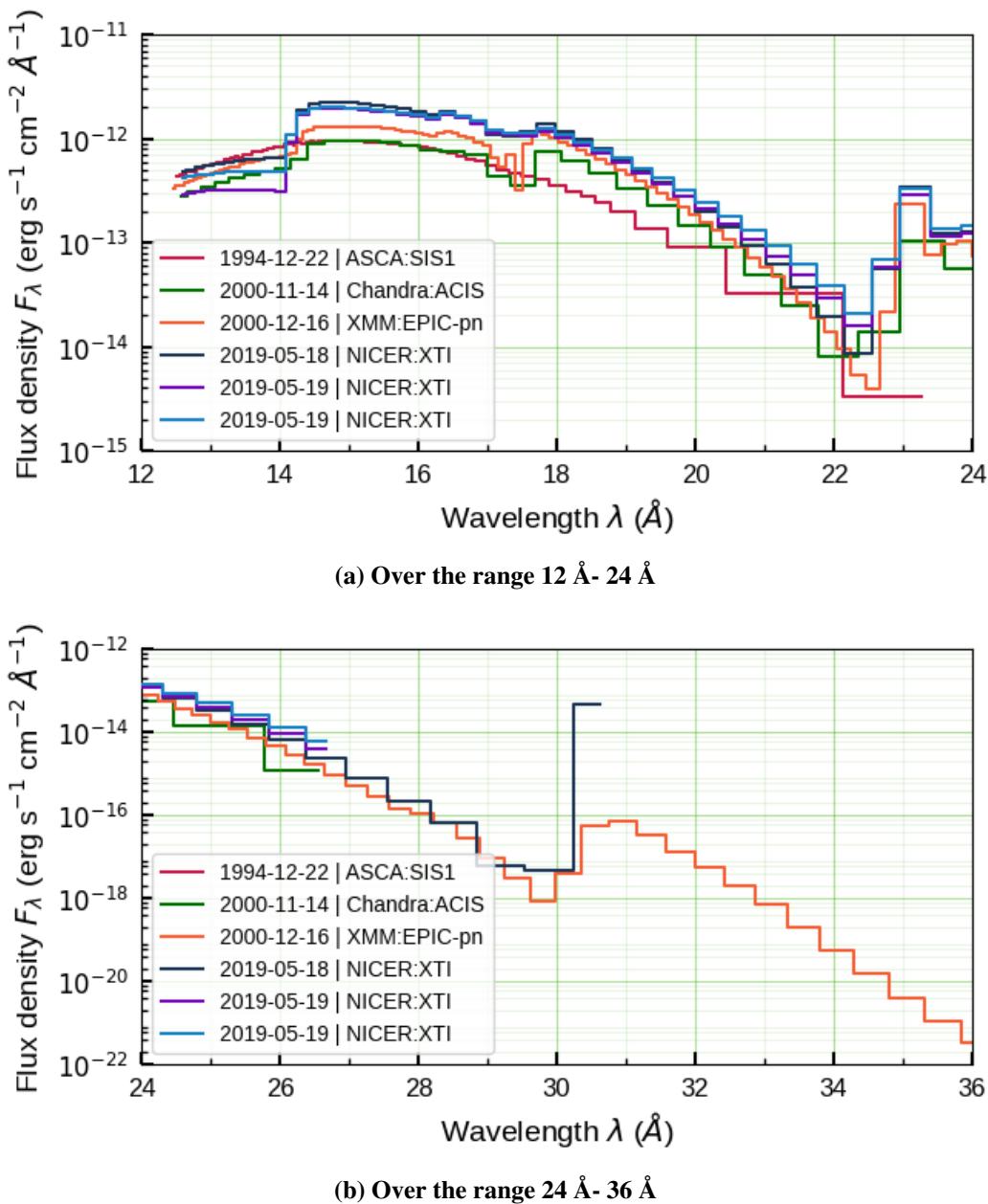


Figure 3.4: Unfolded spectra after model fitting

3.4. Results

3.4.4 Luminosity versus Effective Temperature

In figure 3.5, we present a visualization of the calculated values of the luminosity L_* and effective temperature T_{eff} for RX J0925.7-4758 using the best-fit model. One may note that the horizontal axis, containing the T_{eff} data is plotted as increasing from right to left, *à la* Hertzsprung-Russell diagram. This enables one to compare the overlap of the regions of uncertainty of these parameters as calculated using the data from multiple observatories.

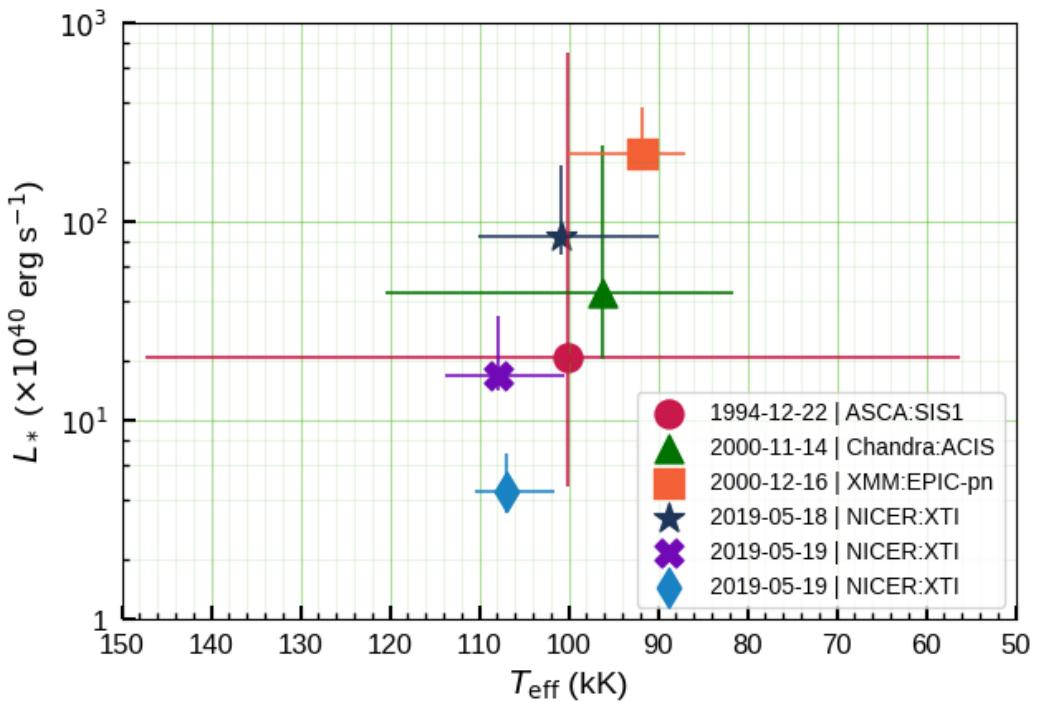


Figure 3.5: Uncertainties in L_* and T_{eff} calculations from multi-observatory observations of RX J0925.7-4758

3.4.5 Presence of Elemental Absorption Edges

During the preliminary fitting of the observations, the detection of residuals in absorption at energies of 0.402 keV, 0.532 keV, 0.708 keV and 0.867 keV suggested the presence of absorption edges. Consequently, by including relevant model components, four elemental absorption edges were detected from the best fitting model on all the observations. These absorption edges, belonging to N, O, Ne and Fe, are overlaid on the unfolded spectra and

3.4. Results

displayed in figure 3.6. The energies of the absorption edges were kept frozen during the fitting for all the observations.

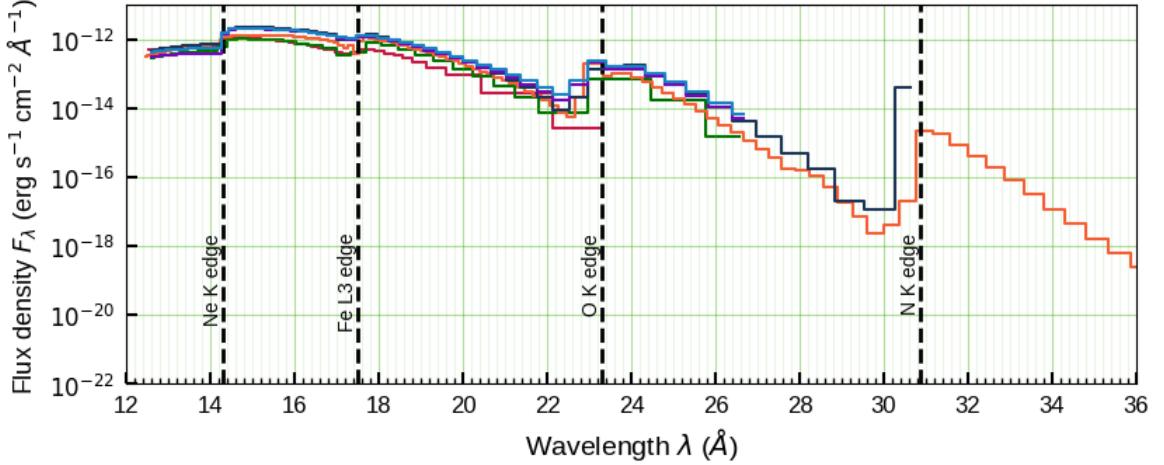


Figure 3.6: Unfolded spectra with overlaid elemental absorption edges

Because the best-fitted spectrum obtained from the EPIC-pn instrument of XMM-Newton spans the widest range of wavelengths, known absorption edges [66, 67] were identified in the vicinity of the absorption edges in figure 3.6. These are reported in table 3.3. Then they were overlaid on the unfolded spectrum obtained from the EPIC-pn observation. The four absorption edges that could be identified are the K edges of N, O and Ne and the L_3 edge of Fe at 30.873 Å, 23.305 Å, 14.302 Å and 17.509 Å respectively, as displayed in the subplots of figure 3.7. Three among these four edges, namely the K edges of O and Ne and the L_3 edge of Fe, were reported to be present in the spectrum of RX J0925.7-4758 using the CLOUDY photoionization code by Prodhani and Baruah (2018) [68].

3.4. Results

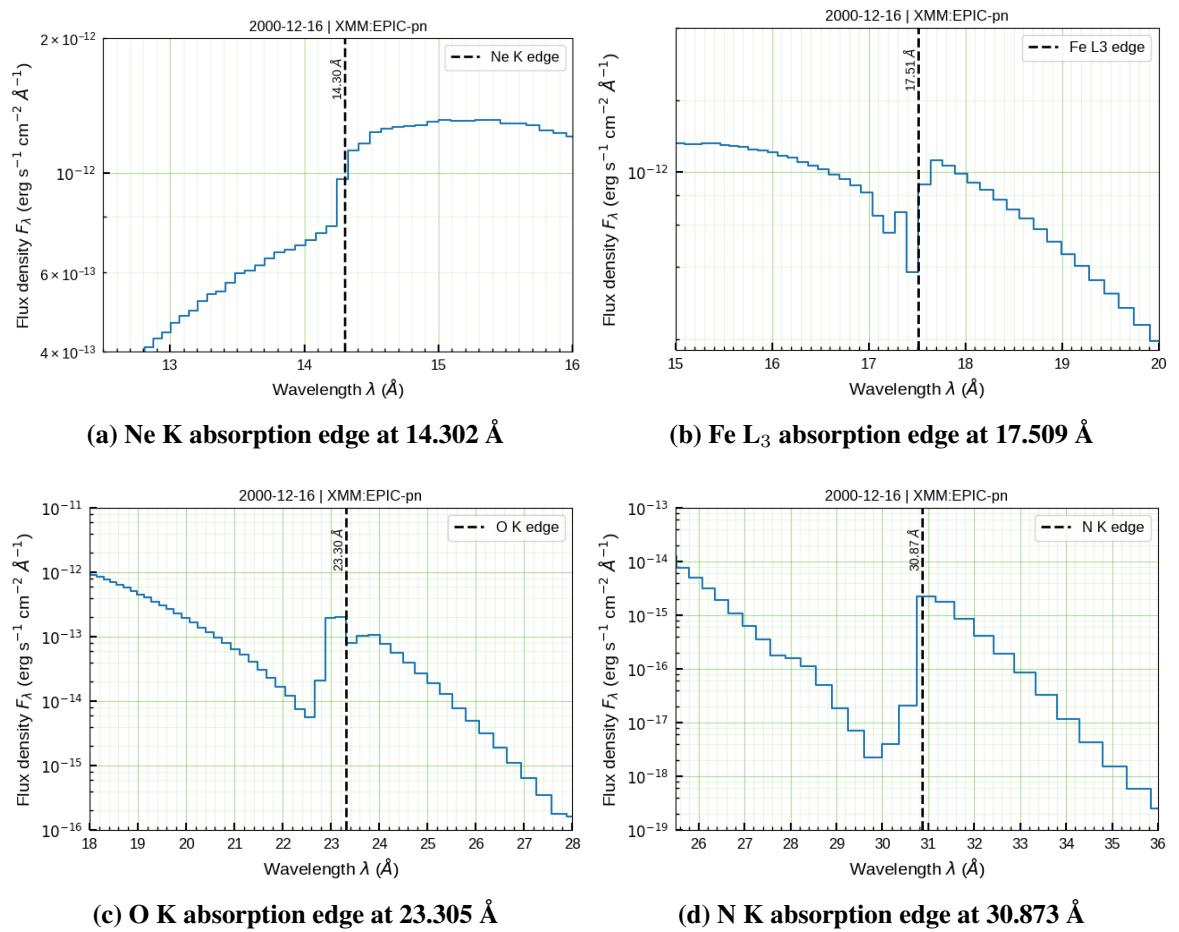


Figure 3.7: Absorption edges for N, O, Ne and Fe overlaid on the unfolded spectrum for EPIC-pn observation

3.4. Results

Table 3.3: Absorption depth D of RX J0925.7-4758 derived from identified absorption edges of Ne, Fe, O and N.

Observatory	Obs. ID	Absorption Depth D				
		Ne K edge	Fe L_3 edge	O K edge	N K edge	Relative Depth
		14.302 Å	17.509 Å	23.305 Å	30.873 Å	
ASCA	43036000	0.503	0.661	0.941	1.787	0.28 : 0.37 : 0.53 : 1.0
Chandra	644	0.497	0.673	1.823	5.171	0.10 : 0.13 : 0.35 : 1.0
XMM-Newton	0111150101	0.211	0.033	0.132	5.215	0.04 : 0.01 : 0.03 : 1.0
NICER	2611020101	0.819	0.013	1.630	0.185	0.50 : 0.01 : 1.0 : 0.11
NICER	2611020102	1.612	0.180	0.395	1.752	0.92 : 0.10 : 0.23 : 1.0
NICER	2611020103	1.271	0.315	0.132	0.006	1.0 : 0.25 : 0.10 : 0.01

3.5 Discussion

As per current literature on spectral fitting of the SSS RX J0925.7-4758, Bearda et al. (2002) and Motch et al. (2002) had arrived at the conclusion that its spectra cannot be reproduced by NLTE model atmospheres [58, 59]. Earlier, Hartmann et al. (1999) had applied non-local thermodynamic equilibrium (NLTE) models, which included metal line opacities, to the spectrum extracted from the observations by BeppoSAX LECS of RX J0925.7-4758 on 25–26 January 1997 [57], where the best fit was using a model consisting of two spectral components. But nevertheless, they obtained a reduced $\chi^2 > 2$. In the absence of a proper model describing the emission spectrum, it becomes impossible to derive fundamental parameters such as temperature, neutral hydrogen column density and luminosity [59]. Hence, obtaining an acceptable fit for RX J0925.7-4758 spectrum assumes crucial importance at this juncture.

3.5.1 Best-fit Continuum Model

The model, chosen for fitting the data from six different observations, consists of a publicly available²² NLTE (non-local thermal equilibrium) table model component computed from a grid of stellar model atmosphere fluxes for source emission, a photoelectric absorption model component²³, a model component for absorption by inter-stellar medium²⁴ and four model components to account for the presence of absorption edges²⁵.

Figure 3.8 presents the residual statistics from best-fit model to all observations. The values of the fit statistic used, i.e. the reduced χ^2 , were all within the acceptable range of $1 < \chi^2_{\text{reduced}} < 2$, which warrants a model to be considered a good fit for the observed data. An inspection of the distribution of the residual suggests an approximately normal distribution, which can be observed in the best-fit models for all the observations, further supporting the validity of the model fit. Such a distribution of the residuals is displayed in figure 3.8b for the observations of the EPIC-pn data.

As it can be seen in figure 3.8b, the kernel density estimate (KDE) function of the distribution

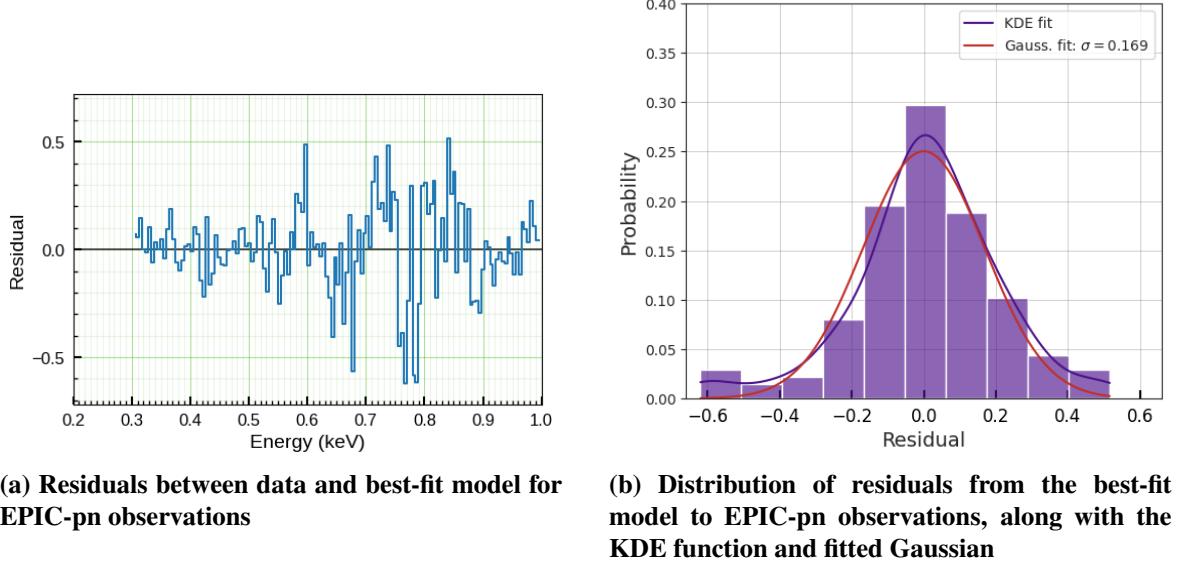
²²<http://astro.uni-tuebingen.de/~rauch/TMAF/TMAF.html>

²³<https://heasarc.gsfc.nasa.gov/xanadu/xspec/manual/XSmodelPhabs.html>

²⁴<https://heasarc.gsfc.nasa.gov/xanadu/xspec/manual/node255.html>

²⁵<https://heasarc.gsfc.nasa.gov/xanadu/xspec/manual/node247.html>

3.5. Discussion



(a) Residuals between data and best-fit model for EPIC-pn observations

(b) Distribution of residuals from the best-fit model to EPIC-pn observations, along with the KDE function and fitted Gaussian

Figure 3.8: Residual statistics from best-fit model to all observations

closely approximates a normal distribution centred about zero (with zero indicating a perfect fit) and with a standard deviation of 0.169, thereby indicating that the observed count rate data can be considered to be random fluctuations which are normally distributed about the best-fit model. Therefore, the normal distribution of the residuals indicates that they are random and do not have a systematic bias (as is expected of a good fit), thereby further validating that the model fitting performed is satisfactory for all six cases of the observations.

In figure 3.9, we find the Gaussian distribution fitted to all the observations. This figure shows that the quality of the fit is the best for the earlier Chandra, XMM-Newton and ASCA observations. For the recent NICER observations, the residuals show a wider spread about the perfect fit.

3.5.2 NLTE Pure H Model Atmosphere

In a classical stellar atmosphere, which is a plane-parallel, horizontally homogenous atmosphere in hydrostatic and radiative equilibrium, a non-LTE (or NLTE) description refers to a scenario where the energy levels of some selected species may be allowed to depart from their local thermodynamic equilibrium (LTE) values [69].

3.5. Discussion

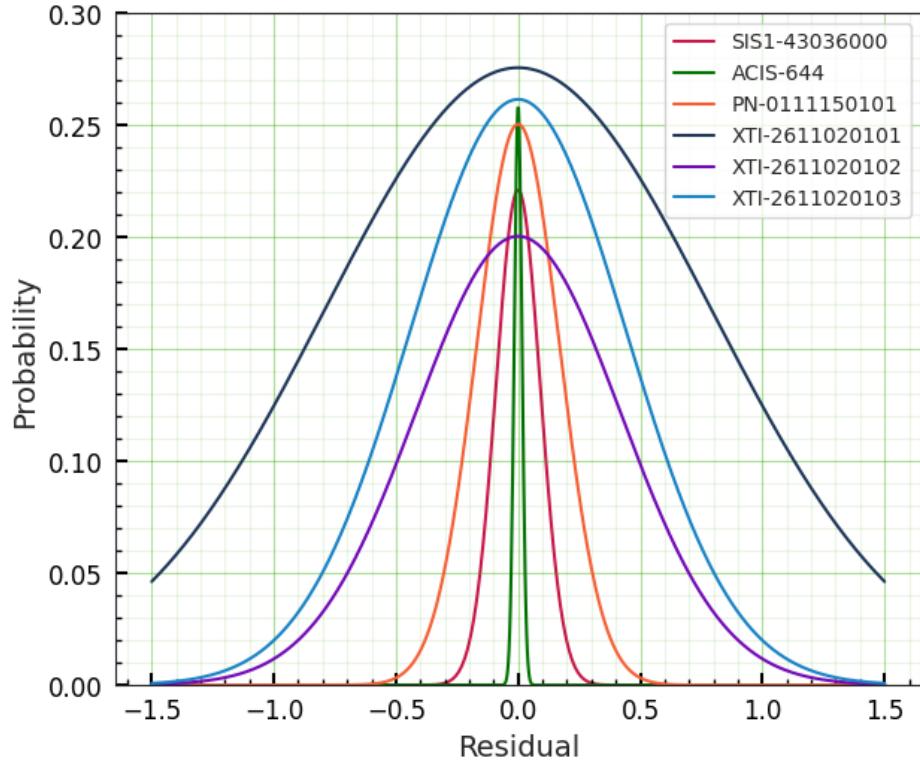


Figure 3.9: Gaussian approximations of the KDE functions for residual distributions of all observations

3.5.2.1 Structural Equations

Given below are the structural equations that are used to model an NLTE atmosphere.

In its second-order form, the *radiative transfer equation* is

$$\frac{\partial^2(f_\nu J_\nu)}{\partial \tau_\nu^2} = J_\nu - S_\nu \quad (3.2)$$

where J_ν is the *mean intensity* over all solid angles, τ_ν the monochromatic *optical depth*, f_ν the variable *Eddington factor* and S_ν the *source function*. The upper and lower boundary conditions on equation (3.2) are respectively

$$\left[\frac{\partial(f_\nu J_\nu)}{\partial \tau_\nu} \right]_0 = g_\nu J_\nu(0) - H_\nu^{\text{ext}} \quad (3.3)$$

$$\left[\frac{\partial(f_\nu J_\nu)}{\partial \tau_\nu} \right]_{\tau_{\max}} = H_\nu^+ - \frac{1}{2} J_\nu \quad (3.4)$$

3.5. Discussion

where g_ν is the *surface Eddington factor*, H_ν^{ext} , H_ν^+ being the *first angular moment* of specific intensity taken at the respective upper and lower boundaries of the current stellar atmospheric layer.

The *hydrostatic equilibrium equation* is re-cast as

$$\frac{dP_{\text{gas}}}{dm} = g - \frac{4\pi}{c} \int_0^\infty \frac{dK_\nu}{dm} d\nu = g - \frac{4\pi}{c} \int_0^\infty \frac{\chi_\nu}{\rho} H_\nu d\nu \quad (3.5)$$

where K_ν is the *second angular moment* of specific intensity, χ_ν is the *absorption coefficient* and m is the *Lagrangian mass*. In equation (3.5), the total pressure is composed of three parts: the *gas pressure* P_{gas} , the *radiation pressure* represented by the integrals on the right-hand side and the *microturbulent pressure* P_{turb} being ignored.

The *radiative equilibrium equation* is an expression of the conservation of total radiant flux. It may be written in an *integral form* – more accurate at upper atmospheric layer, or in a *differential form* – more accurate in deeper layers. To improve the accuracy, numerical algorithms implement a linear combination of both form as

$$\begin{aligned} & \alpha \left[\int_0^\infty (\kappa_\nu J_\nu - \eta_\nu) d\nu \right] \\ & + \beta \left[\int_0^\infty \frac{d(f_\nu J_\nu)}{d\tau_\nu} d\nu - \frac{\sigma_R}{4\pi} T_{\text{eff}}^4 \right] = 0 \end{aligned} \quad (3.6)$$

where σ_R is the Stefan-Boltzmann constant, κ_ν , η_ν are the thermal absorption and emission coefficients respectively. In equation (3.6) above, the term within the first pair of brackets contains the integral form and that within the second pair contains the differential form. The empirical coefficients α , β enable a transition from upper layers ($\alpha \rightarrow 1$, $\beta \rightarrow 0$) to lower layers ($\alpha \rightarrow 0$, $\beta \rightarrow 1$) – such a transition is taken to be around the depth where the Rosseland mean optical depth is ~ 1 .

The *kinetic equilibrium equations* (or *rate equations*) are written for each chemical species, after having first selected those species which are to be considered to deviate from LTE. For any species, these equations may be represented by a matrix equation as

$$\mathbf{A} \cdot \mathbf{n} = \mathbf{b} \quad (3.7)$$

3.5. Discussion

where the elements of the *rate matrix* \mathcal{A} are given as

$$\begin{aligned}\mathcal{A}_{ij} &= \sum_{j \neq i} (R_{ij} + C_{ij}) \\ \mathcal{A}_{ij} &= -(R_{ji} + C_{ji}), \text{ for } j \neq i \text{ and } i \neq k \\ \mathcal{A}_{kj} &= 1 + S_j\end{aligned}$$

with k being the index of the *characteristic level* and R_{ij} , C_{ij} the radiative and collisional rates respectively between transition levels i and j . It must be noted that R_{ij} 's implicitly involve the *Saha-Boltzmann factor* $\Phi_i(T)$ which comes into play for *bound-free transitions*, i.e. absorption edges

$$\Phi_i(T) = \frac{g_i}{2g_1^+} \left(\frac{h^2}{2\pi m_e kT} \right)^{3/2} e^{(E_I - E_i)/kT} \quad (3.8)$$

where E_I is the ionization potential of the ion to which level i belongs, E_i the excitation energy of level i and g_1^+ the statistical weight of the ground state of the next ion and m_e the electronic mass. In equation (3.7), \mathbf{n} is the vector of populations of levels and the vector \mathbf{b} has a single non-zero element corresponding to the characteristic level k as

$$b_i = (N - n_e)\alpha_I \delta_{ki}$$

where N is the net populations of all levels, n_e the number density of electrons and α_I the fractional abundance of the species I .

The overall electrical neutrality of the medium is expressed by the *charge conservation equation* as

$$\sum_i n_i Z_i - n_e = 0 \quad (3.9)$$

where Z_i is the charge associated with level i .

The *mass density*, which is related to the atomic level populations, can be expressed as

$$\rho = (N - n_e)\mu m_H \quad (3.10)$$

where μ is the mean molecular weight and m_H is the mass of the H atom. A quantity known

3.5. Discussion

as the *fictitious massive particle density* is defined as

$$n_m \equiv (N - n_e)\mu \quad (3.11)$$

which then enables the mass density to be simply written as $\rho = n_m m_H$.

The absorption coefficient included in equation (3.5) is written as the combination

$$\chi_\nu = \kappa_\nu + \kappa_\nu^{\text{sc}} \quad (3.12)$$

where κ_ν is also known as the *extinction coefficient* and κ_ν^{sc} the *scattering coefficient*. The extinction coefficient is given as

$$\begin{aligned} \kappa_\nu = & \sum_i \sum_{j>i} [n_i - n_j G_{ij}(\nu)] \sigma_{ij}(\nu) \\ & + \sum_l n_e n_l \sigma_{ll}(\nu, T) (1 - e^{-h\nu/kT}) + \kappa_\nu^{\text{addl}} \end{aligned} \quad (3.13)$$

where the first term represents the contribution from *bound-bound transitions* (i.e. absorption lines), the second represents that from *bound-free transitions* (i.e. absorption edges) and the last term lumps together any additional opacity not written as detailed transitions. Here $G_{ij} = \frac{w_i g_i}{w_j g_j}$, with g_i, g_j and w_i, w_j being the respective degeneracies and statistical weights of levels i and j respectively. Also σ_{ij} is the opacity of the transition between levels i and j .

The scattering coefficient in equation (3.12) is given as

$$\kappa_\nu^{\text{sc}} = n_e \sigma_e + \sum_i n_{\text{Ray},i} \sigma_{\text{Ray},i} \quad (3.14)$$

where $\sigma_e, \sigma_{\text{Ray},i}$ are the Thomson and Rayleigh scattering cross-sections respectively. For pure-hydrogen atmospheres with high temperatures, the Rayleigh scattering is negligible because the hydrogen atoms are expected to be fully ionized.

In a similar manner, the *total emission coefficient* is also expressed as a sum of thermal and scattering contributions as

$$\eta_\nu^{\text{total}} = \eta_\nu + \eta_\nu^{\text{sc}} \quad (3.15)$$

3.5. Discussion

where

$$\begin{aligned}\eta_\nu = \left(\frac{2h\nu^3}{c^2} \right) & \left[\sum_i \sum_{j>i} n_j G_{ij}(\nu) \sigma_{ij}(\nu) \right. \\ & \left. + \sum_l n_e n_l \sigma_{ll}(\nu, T) e^{-h\nu/kT} \right] + \eta_\nu^{\text{addl}}\end{aligned}\quad (3.16)$$

in which, again, any additional emissivity is lumped together and given using the Planck function as $\eta_\nu^{\text{addl}} = \kappa_\nu^{\text{addl}} B_\nu$.

Finally, taking into account the *convective flux* F_{conv} (if any) within the differential form of the radiative equilibrium equation (3.6) as

$$\int_0^\infty \frac{d(f_\nu J_\nu)}{d\tau_\nu} d\nu + \frac{F_{\text{conv}}}{4\pi} = \frac{\sigma_R}{4\pi} T_{\text{eff}}^4 \quad (3.17)$$

Differentiating equation (3.17) with respect to the Lagrangian mass m finally gives the *radiative+convective equilibrium equation* as

$$\int_0^\infty (\kappa_\nu J_\nu - \eta_\nu) d\nu + \frac{\rho}{4\pi} \frac{dF_{\text{conv}}}{dm} = 0 \quad (3.18)$$

3.5.2.2 Numerical Solution using ALI

Accelerated lambda iteration (ALI) is a technique used to numerically solve the radiative transfer problem, particularly in the context of stellar atmosphere modeling [70]. It accelerates the computational process by introducing a correction term that takes into account the influence of previous iterations. A simplified procedural overview of this scheme is as follows:

1. *Initial guess:* Start with an initial guess for the specific intensity of radiation at different frequencies within the atmosphere.
2. *Lambda iteration:* Calculate how the radiation interacts with the atmosphere using the current guess for intensity, considering the absorption and emission by the species in the atmosphere.
3. *Correction term:* Instead of simply accepting the result from step 2, ALI calculates a

3.5. Discussion

correction term based on the difference between the current guess and the result.

4. *Update guess*: Add this correction term to the initial guess to obtain a new estimate for the intensity.
5. *Iterate*: Repeat steps 2–4 until the solution converges, meaning the intensity values no longer change significantly between iterations.

In order to numerically solve the structural equations for NLTE, they are first discretized over all three continuous variables: optical depth τ , frequency ν and angle θ . These discretized structural equations, together with a number of auxiliary relations, form a system of highly coupled, non-linear equations. Then the *complete linearization scheme* [71] is used to treat all equations on the same footing, thus solving all structural equations simultaneously. The Tübingen NLTE Model Atmosphere Package (TMAP)²⁶ code uses ALI to calculate the stellar atmosphere by taking into account metal line-blanketing. With an extended grid of such atmospheres of pure-hydrogen models, synthetic spectral were calculated and these are eventually used as table models in XSPEC in the present work to simulate the source emission.

3.5.3 Inferences from Results

Here we are trying to understand the supersoft spectrum of RX J0925.7-4758, across observations from 1994 to 2019. The idea is to develop a robust model that explains its supersoft spectrum over different time scales and different observing instruments. The best-fit model consists of XSPEC model components which consider various factors affecting the observed spectrum:

- a) `atable{pure-H NLTE with log g = 7}`: for the actual radiation emitted by RX J0925.7-4758
- b) `edge`: for specific wavelengths where bound-free absorption is particularly strong, leading to absorption edges
- c) `phabs`: for describing the photoelectric absorption at the source itself.

²⁶<http://astro.uni-tuebingen.de/~rauch/TMAP/TMAP.html>

3.5. Discussion

- d) `ismabs`: for describing how light interacts with the intervening inter-stellar medium before reaching us

Initially, a simple blackbody model failed to fit the spectrum. Blackbodies emit a characteristic spectrum based on temperature, and it seems this doesn't accurately represent the complex emission from RX J0925.7-4758. Then the supersoft X-ray emission of RX J0925.7-4758 was modelled using an NLTE stellar atmosphere model [72]. Some of these are made available as tables in appropriate FITS format to be used with the XSPEC table model component `atable` [73, 74]. These models, which are essentially pre-calculated grids, vary from one another in terms of the elemental abundances, the effective temperature T_{eff} and the effective gravity $\log g$ (where the gravitational acceleration g is expressed in cgs units). Choosing a particular FITS file fixed the specific elemental abundance and effective gravity. Then the fitting of the model to the data is performed by using the effective temperature T_{eff} and the redshift z as parameters.

During the analysis of the spectra from all six observations of RX J0925.7-4758, we found that only an NLTE model with a pure hydrogen atmosphere for which $\log g = 7$ was able to provide a common model continuum spectrum that gave acceptable fit quality for all six observations. This model²⁷, developed and made publicly available by Rauch et al., has, as the core of its computational framework, the detailed representation of hydrogen atoms within stellar atmospheres. Using the TMAP, plane-parallel, static, pure-hydrogen models were calculated, employing several simplifying assumptions to create a tractable model of the stellar atmosphere. Here, hydrogen is represented as a model atom, focussing on energy levels up to $n = 14$ under NLTE conditions, neglecting the rapid depopulation of higher levels due to a compact star's strong gravity. Certain complexities like the exact treatment of light absorption are approximated for efficiency. Finally, the model only extends to a specific optical depths of

$$\log \tau = -8 \cdots + 4$$

within the atmosphere where light can travel freely, and utilizes increased detail for calculations in specific wavelength range to ensure accuracy in that region. For the synthetic spectra, the line-broadening tables of Lemke (1997) were used [75].

As it can be seen from table 3.2 and figure 3.5, the effective temperature is obtained to be $\sim 10^5$ K for all six observations and luminosity is computed to be of the order of 10^{41}

²⁷http://astro.uni-tuebingen.de/~rauch/TMAF/flux_H.html

3.5. Discussion

erg s⁻¹, which is about 3 orders of magnitude higher than what had been reported earlier [57]. Such order of magnitude of T_{eff} is also observed recently in similar fitting of CAL 83 spectrum using a pure-hydrogen NLTE model [62]. The high values of T_{eff} and L_* from a good fit using a pure-hydrogen model seem counterintuitive for an SSS, which is typically a white dwarf accreting from a companion star. This is because white dwarfs are usually much cooler, and the accreted material might not be pure hydrogen.

However, we need to take into account the currently accepted model for SSS which is the steady nuclear burning on the surface of a white dwarf. It is likely that the companion star to RX J0925.7-4758 is a main-sequence star, thereby hydrogen-rich. Consequently, it is possible that only hydrogen-rich matter is able to accrete around the white dwarf. This matter, upon reaching the surface of the white dwarf, owing to its large gravity, undergoes steady nuclear burning. This may explain the success of the pure-hydrogen NLTE model in fitting its spectrum. It must be borne in mind that the high T_{eff} likely characterizes the hot accretion disk surrounding the white dwarf, and not the white dwarf itself. Friction and compression may cause the accreting matter to attain the temperatures required to undergo steady nuclear burning.

There are a couple of limitations to drawing such an inference regarding RX J0925.7-4758:

- i. The NLTE model assumes a static source, which might not be entirely accurate for a dynamic system like an accretion disk.
- ii. The model assumes a pure-hydrogen composition, which might be a simplification of the actual accreting material.

Future modeling could explore relaxing some assumptions, such as including additional elements or allowing for a dynamic disk structure. One way to do so could be by utilising the high-resolution grating spectra from observatories, such as the reflection grating spectrometer (RGS) on-board XMM-Newton, to study the absorption line profiles observed in the spectrum of RX J0925.7-4758 and analyzing the corresponding transitions can provide valuable insights into the dynamics of the accreting material.

Absorption lines in the spectrum of RX J0925.7-4758 arise due to the interaction of the emitted radiation with material along the line of sight. These lines exhibit diverse profiles, including shapes, widths, and shifts, which reflect the physical properties and dynamics of the absorbing material. Each absorption line corresponds to a specific transition of an atom or

3.5. Discussion

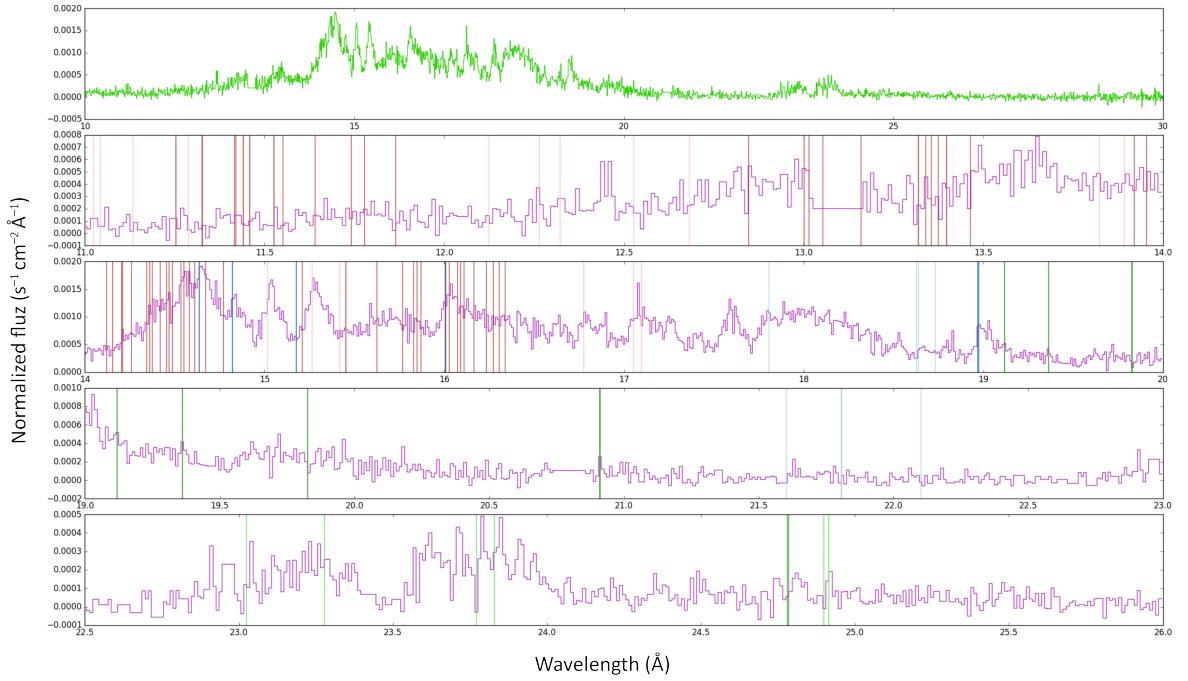


Figure 3.10: Overlay of transition lines on the fluxed RGS spectrum of RX J0925.7-4758 for Obs. ID 0111150101

ion in the intervening material. Some of such lines are overlaid on the fluxed RGS spectrum of RX J0925.7-4758 for the observation ID 0111150101, as presented in figure 3.10 [76]. Here, the colour scheme is: N VI and N VII lines in light and dark green respectively, O VII and O VIII lines in light and dark blue respectively, Fe XVII and Fe XVIII lines in light and dark red respectively.

By identifying the atomic species and transitions associated with each absorption feature, one can gain insight into the composition and kinematics of the absorbing medium. While optical spectra have been extensively studied by categorizing them based on the presence or absence of certain spectral lines and identifying anomalies such as unusually strong or weak lines, similar approaches have been challenging to apply to X-ray spectra due to their limited spectral resolution. However, with nearly two decades of high-resolution grating spectra available from X-ray observations, it is now opportune to explore and develop methods that leverage these data to categorize and analyze X-ray spectra in more nuanced ways, such as that by Ness et al. (2013) [77].

3.5.4 Relative Strengths of Absorption Edges

Absorption edges are included in an XSPEC model using the multiplicative component named `edge`. On a continuum model, an absorption edge may be modelled as follows:

$$M(E) = \begin{cases} 1; & E \leq E_{\text{th}} \\ \exp \left[-D \left(\frac{E}{E_{\text{th}}} \right)^{-3} \right]; & E > E_{\text{th}} \end{cases} \quad (3.19)$$

In equation (3.19), E_{th} is the *threshold energy* and D is the *absorption depth*. The model component is implemented with these two quantities being its parameters. The relative values of the absorption depths enables a comparison of the strengths of the absorption edges.

The absorption depths calculated from the unfolded spectra, after obtaining the best fit to the model, are presented in table 3.3. In all six observations, the same absorption edges were identified. For the observations made by ASCA, Chandra and XMM-Newton, the identified edges show the same trend with respect to the relative strengths of the absorption depths of these edges, i.e. the N K absorption edge is the strongest, followed by the O K edge, the Fe L_3 edge and the Ne K edge with similar strengths.

However, this is not the case for the three NICER observations, each of which show different edges to be the strongest. The reasons for such an inconsistency might range from instrumental effects (such as variations in the detector response with time, or changes in the gain calibration between observations) to issues with data reduction (such as inconsistencies in background subtraction, or inaccuracies in deadtime correction). Because NICER is a relatively new mission, it is a worthwhile exercise to investigate this particular inconsistency in absorption depth strength, which would include a detailed review of the NICER calibration documents, analysis of data from different detector regions, comparison with published data on similar sources and submission of relevant science proposals for new observations.

CHAPTER 4

FITTING THE HIGH RESOLUTION SPECTRUM OF RX J0925.7-4758 USING COMPOSITE XSPEC MODEL

Contents

4.1	RGS Spectra from XMM-Newton	72
4.1.1	RGS Design and Functionality	73
4.1.2	Light Path and Spectral Orders	74
4.2	Models for Data Fitting	75
4.2.1	Model Grid	76
4.2.2	Model Components	77
4.3	Analysis of High-resolution RGS Spectra	84
4.3.1	Fitting of RGS1 Spectra	84
4.3.2	Fitting of RGS2 Spectra	88
4.3.3	Comparison with Current Literature	89
4.3.4	Supersoft X-ray Physics using Best-fit RGS Model	91

**CHAPTER 4. FITTING THE HIGH RESOLUTION SPECTRUM OF RX J0925.7-4758
USING COMPOSITE XSPEC MODEL**

Abstract

This chapter presents a new analysis of high-resolution X-ray spectra of the astronomical object RX J0925.7-4758, obtained using the XMM-Newton observatory's RGS instrument. We employ spectral fitting using a model grid to ascertain the one that accurately represents the observed data. Our analysis focuses on a subset of pre-defined models, evaluating their fit quality using the reduced chi-squared statistic χ^2_{reduced} . We find that model M11 provides the best fit ($1 < \chi^2_{\text{reduced}} < 2$) for spectra from both RGS1 and RGS2 instruments, across all observed spectral orders. This signifies a significant improvement compared to previous studies on RX J0925.7-4758, which faced challenges in modeling the object's X-ray emission using a non-local thermodynamic equilibrium (NLTE) atmosphere model. The model M11 incorporates several components, including interstellar medium (ISM) absorption, intrinsic X-ray emission with separate contributions from cold gas absorption and an NLTE atmosphere, thermal plasma emission from potentially two different mechanisms, and a wind or outflow component. This comprehensive model suggests a more accurate capture of the physical processes governing the X-ray emission from RX J0925.7-4758.

4.1 RGS Spectra from XMM-Newton

X-ray astronomy provides a powerful tool for studying the universe’s most energetic phenomena. However, a complete understanding often requires detailed information about the elemental composition and physical conditions within the emitting region. This is where high-resolution X-ray spectroscopy shines, allowing us to dissect the X-ray emission line by line.

The Reflection Grating Spectrometers (RGS) onboard the XMM-Newton satellite offer exceptional capabilities in this domain¹. These instruments achieve resolving powers (ability to distinguish close-lying spectral lines) in the range of 100 to 500 FWHM (Full Width at Half Maximum) within the energy band of 0.33-2.5 keV. This specific energy range is particularly rich in diagnostic information due to the presence of numerous emission lines:

- **K-shell transitions and He-like triplets:** These lines originate from the innermost electron shells (K-shell) of lighter elements like such as C, N, O, Ne, Mg and Si.
- **L-shell transitions:** Heavier elements like Iron (Fe) and Nickel (Ni) also contribute to the X-ray spectrum through their L-shell transitions within this energy range.

The wealth of emission lines within the RGS spectra may be considered to be like a “fingerprint”, revealing the elemental composition and physical conditions within the emitting plasma. By analyzing the line strengths, profiles, and shifts, one can:

1. *Determine abundances:* The relative intensity of specific lines reflects the abundance of the emitting element within the source.
2. *Measure temperatures:* The broadening of emission lines can be attributed to the thermal Doppler effect, where hotter plasmas exhibit broader lines.
3. *Investigate densities:* Certain lines are sensitive to the density of the emitting gas.
4. *Identify redshift:* The systematic shift of all emission lines towards lower energies (redshift) indicates the object’s distance from the observatory.

¹<https://www.cosmos.esa.int/web/xmm-newton/technical-details-rgs>

4.1. RGS Spectra from XMM-Newton

Therefore, XMM-Newton RGS observations provide a powerful diagnostic tool for studying a wide range of astrophysical phenomena, which include stellar coronae and winds, accretion disks and supernova remnants.

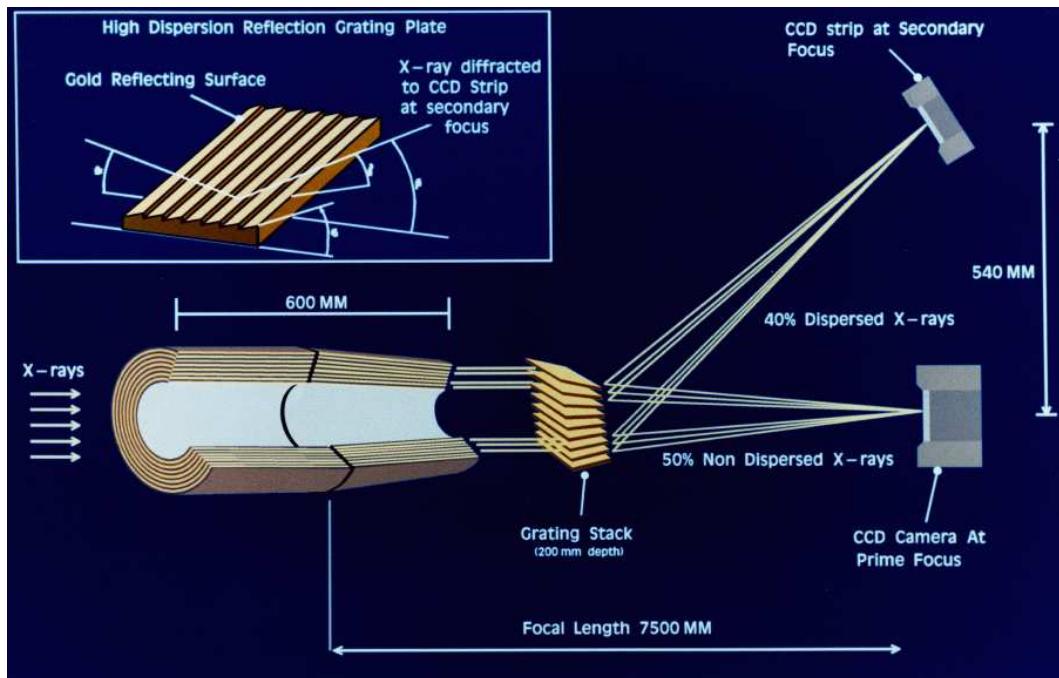


Figure 4.1: Schematic of the RGS instruments of XMM-Newton. Courtesy: Hardware Schematics at XMM-Newton SOC

4.1.1 RGS Design and Functionality

Each of the two XMM-Newton telescopes feeds light to a dedicated RGS instrument, referred to as RGS1 and RGS2, as illustrated in figure 4.1². These instruments consist of two main parts:

1. **Reflection Grating Assemblies (RGAs):** These assemblies act as the heart of the RGS instruments. They comprise arrays of precisely manufactured grating plates with microscopic grooves etched onto their surfaces, which play a crucial role in diffracting incoming X-ray photons. In figure 4.1, these are referred to as *grating stack*.
2. **RGS Focal Cameras (RFCs):** Downstream from the RGAs lie the RFCs. These consist of long, linear Charge-Coupled Device (CCD) detectors. The diffracted X-

²https://heasarc.gsfc.nasa.gov/docs/xmm/xmmhp_gal_hard_schem.html

4.1. RGS Spectra from XMM-Newton

rays from the gratings fall onto these CCDs, where they are converted into electronic signals and ultimately into a digital spectrum. They are referred to as *CCD strip* in figure 4.1.

Table 4.1: In-orbit performance of RGS instruments

Parameter		RGS1			RGS2		
		10 Å	15 Å	35 Å	10 Å	15 Å	35 Å
Effective area (cm^2)	1 st order	51	61	21	53	68	25
	2 nd order	29	15	–	31	19	–
Resolution (km s^{-1})	1 st order	1700	1200	600	1900	1400	700
	2 nd order	1000	700	–	1200	800	–
Wavelength range	1 st order	5 – 38 Å(0.35 – 2.5 keV)					
	2 nd order	5 – 20 Å(0.62 – 2.5 keV)					
Wavelength accuracy	1 st order	$\pm 5 \text{ m}\text{\AA}$			$\pm 5 \text{ m}\text{\AA}$		
	2 nd order	$\pm 4 \text{ m}\text{\AA}$			$\pm 3 \text{ m}\text{\AA}$		
Bin size [$3 \times 3 (27 \mu)^2$ pixels]		2.5 arcsec (cross dispersion direction)			7 – 14 mÅ (dispersion direction, first order)		

4.1.2 Light Path and Spectral Orders

While the primary focus of the XMM-Newton telescopes is occupied by the European Photon Imaging Camera (EPIC) MOS detectors, designed for high-resolution X-ray imaging, the RGAs ingeniously intercept about 58% of the incoming light before it reaches the EPIC cameras. This light then interacts with the grating plates in the RGAs. The gratings cause the X-ray photons to diffract, separating them according to their energy (wavelength). This phenomenon produces the two prominent first and second order spectra with dispersion of 8.3 and 12.7 mm Å⁻¹ at 15 Å.

These spectral orders are dispersed across the length of the CCD strips, with higher energy (shorter wavelength) photons landing closer to the beginning of the CCD and lower energy

4.2. Models for Data Fitting

(longer wavelength) photons landing towards the end. This creates a one-to-one mapping between position on the CCD and the energy (or wavelength) of the detected X-ray photon.

The performance characteristics of the RGS instruments, as measured in orbit, are summarized in table 4.1, which highlights key parameters namely the *effective area*, *resolution*, *wavelength range*, *wavelength accuracy* and *bin size*.

The effective area represents the collecting power of the instrument at a specific wavelength. As expected, the effective area is higher for the first order compared to the second. Resolution refers to the instrument's ability to distinguish between closely spaced spectral lines. Evidently, the resolution degrades at longer wavelengths (lower energies) within each order. The wavelength range covered by the RGS instruments allows them to probe a vast array of emission lines crucial for understanding the elemental composition and physical conditions within the X-ray emitting source.

While higher-order spectra (beyond the first and second) are technically present, their count rates are significantly lower (~ 8 times lower than that in the second order), rendering them less useful for scientific analysis. Consequently, processed scientific data products from the XMM-Newton Science Operations Centre (SOC) typically only include spectra from the first and second orders. Table 4.2 provides a concise overview of the wavelength and corresponding photon energy ranges covered by these two primary spectral orders.

Table 4.2: Wavelength and photon energy ranges covered by RGS

Order	Wavelength range (Å)	Photon energy range (keV)
1	6 – 38	0.3 – 2.0
2	6 – 20	0.6 – 2.0

4.2 Models for Data Fitting

The analysis of high-resolution X-ray spectra, such as those obtained with XMM-Newton RGS, involves a meticulous process of spectral fitting. This process aims to construct a mathematical model that accurately represents the observed spectrum, allowing us to extract meaningful physical information about the source.

4.2. Models for Data Fitting

4.2.1 Model Grid

Table 4.3 presents a sequence of XSPEC models progressively employed to fit the RGS spectrum of the source RX J0925.7-4758. Each model builds upon the previous one, incorporating additional complexity or refining existing components. For instance, consider the initial model, i.e. M01. This model incorporates two components:

- i. `tbabs`³: This is a multiplicative model component that accounts for the absorption of X-ray photons by the interstellar medium (ISM) along the line of sight to the source, by calculating the cross section for X-ray absorption as the sum of the cross sections for X-ray absorption due to the gas-phase ISM, the grain-phase ISM, and the molecules in the ISM.
- ii. `bbody`⁴: This is an additive model component which represents the intrinsic continuum emission from the source, modeled as a blackbody. Blackbody radiation describes the emission spectrum of a perfect thermal radiator, characterized by a single effective temperature.

By multiplying `tbabs` with `bbody`, the model M01 essentially takes the intrinsic blackbody emission (`bbody`) and modifies it according to the absorption characteristics of the intervening ISM (`tbabs`). This allows for a preliminary fit to the observed spectrum, accounting for both the source's emission and the dimming effect of the interstellar medium. While this model provides a basic framework, it's often too simplistic for complex X-ray sources. Real celestial objects might exhibit additional features like emission lines, multiple thermal components, or non-thermal emission processes. Subsequent models incorporate these complexities to achieve a more accurate representation of the observed spectrum.

The subsequent models become progressively more complex, incorporating additional components to capture the intricate features of the RGS spectrum. The final models (M11 and M12) exhibit the most intricate structure, potentially accounting for multiple sources of absorption (represented by the combination of `rauch` and `tbabs` within M11) and a complex interplay of thermal emission processes (involving both `mekal` and `apec` components).

By systematically refining the model through this iterative process of adding and adjusting components, we have tried to achieve the best possible fit to the observed spectrum. The

³<https://heasarc.gsfc.nasa.gov/xanadu/xspec/manual/node273.html>

⁴<https://heasarc.gsfc.nasa.gov/xanadu/xspec/manual/node136.html>

4.2. Models for Data Fitting

Table 4.3: List of models used for RGS data fitting for RX J0925.7-4758

S. No.	Model ID	Xspec model
1	M01	tbabs*bbbody
2	M02	ismabs*bbbody
3	M03	ismabs*(gauss+bbbody)
4	M04	ismabs*edge ³ *(gauss+bbbody)
5	M05	ismabs*edge ³ *(mekal+bbbody)
6	M06	ismabs*(apec+mekal)*swind1
7	M07	ismabs*(gauss+mekal+bbbody)
8	M08	ismabs*(gauss+mekal+bbbody)*swind1
9	M09	ismabs*(rauch+mekal)*swind1
10	M10	ismabs*(rauch+apec)*swind1
11	M11	ismabs*(rauch*tbabs+apec+mekal)*swind1
12	M12	ismabs*(rauch+apec+mekal)*swind1

quality of the fit is typically evaluated using statistical tests, and the resulting model parameters provide crucial insights into the physical properties of the X-ray emitting source and its environment.

4.2.2 Model Components

XSPEC allows one to construct complex models of astronomical spectra by combining simpler ones. These simpler models represent various physical processes that contribute to the overall observed spectrum. This section describes some the model components utilised to construct the model grid, as described in § 4.2.1.

4.2. Models for Data Fitting

4.2.2.1 X-ray photoabsorption model: `ismabs`

The `ismabs` multiplicative model in Xspec provides a way to simulate X-ray photoabsorption [78]. This model incorporates variable columns for both neutral and ionized species from H, He, N, O, Ne, Mg, Si, S, Ar, Ca, Fe, Ni and Zn.

Because of an inherent degeneracy between the relative columns of H, He I, He II, the column density of He I is not included as a free parameter in the model.

In this model, the absorption cross-sections for various species are sourced as follows:

- Neutral states of Si, S, Ar and Ca from Verner et al. [79]
- Singly and doubly ionized states of Si, S, Ar and Ca from Witthoeft et al. [80, 81]
- Neutral, singly and doubly ionized states of N from Garcia et al. [82]
- Neutral states of O from Gorczyca et al. [83]
- Singly and doubly ionized states of O from Garcia et al. [84], including corrections applied by Gatuzz et al. [85]
- Neutral state of Ne from Gorczyca et al. [86]
- Singly and doubly ionized states of Ne from Gorczyca et al. [87]
- For the Fe-L edge region we use the measurement of metallic iron by Kortright & Kim [88]
- Neutral, singly and doubly ionized states of Mg from Hasoglu et al. (2014).

The parameters for the `ismabs` model are given in table 4.4.

4.2.2.2 Astrophysical plasma emission code: `apec`

The `apec` additive model in Xspec simulates an emission spectrum that is obtained from a collisionally-ionized diffuse gas [89]. The atomic data for collisional and radiative rates, recombination cross sections, dielectronic recombination rates, and satellite line wavelengths are taken from the Astrophysical Plasma Emission Database (APED).

4.2. Models for Data Fitting

Table 4.4: Model parameters for `ismabs`

Parameter	Quantity	Parameter	Quantity
par1	H col. (in 10^{22} cm^{-2})	par2	He II col. (in 10^{22} cm^{-2})
par3	C I col. (in 10^{22} cm^{-2})	par4	C II col. (in 10^{22} cm^{-2})
par5	C III col. (in 10^{22} cm^{-2})	par6	N I col. (in 10^{22} cm^{-2})
par7	N II col. (in 10^{22} cm^{-2})	par8	N III col. (in 10^{22} cm^{-2})
par9	O I col. (in 10^{22} cm^{-2})	par10	O II col. (in 10^{22} cm^{-2})
par11	O III col. (in 10^{22} cm^{-2})	par12	Ne I col. (in 10^{22} cm^{-2})
par13	Ne II col. (in 10^{22} cm^{-2})	par14	Ne III col. (in 10^{22} cm^{-2})
par15	Mg I col. (in 10^{22} cm^{-2})	par16	Mg II col. (in 10^{22} cm^{-2})
par17	Mg III col. (in 10^{22} cm^{-2})	par18	Si I col. (in 10^{22} cm^{-2})
par19	Si II col. (in 10^{22} cm^{-2})	par20	Si III col. (in 10^{22} cm^{-2})
par21	S I col. (in 10^{22} cm^{-2})	par22	S II col. (in 10^{22} cm^{-2})
par23	S III col. (in 10^{22} cm^{-2})	par24	Ar I col. (in 10^{22} cm^{-2})
par25	Ar II col. (in 10^{22} cm^{-2})	par26	Ar III col. (in 10^{22} cm^{-2})
par27	Ca I col. (in 10^{22} cm^{-2})	par28	Ca II col. (in 10^{22} cm^{-2})
par29	Ca III col. (in 10^{22} cm^{-2})	par30	Fe col. (in 10^{22} cm^{-2})
par31	Redshift z		

The `apec` model provides a way to create emission models for plasma, which can be used to analyse spectral data from high-resolution X-ray spectrometers, as in the case of XMM-Newton or Chandra. The current version of the code stores the atomic data in FITS files, thereby separating it from the code. This optimizes limitations on the speed and memory across different computers.

The `apec` model simulates a hot, optically thin plasma which is in a collisional ionization equilibrium, and computes both resulting continuum and line emissivities. Here, the *emissivity* of a spectral line is defined as the total number of radiative transitions per unit volume divided by the product of the electron density n_e and the hydrogen (neutrals and protons) density n_H in the astrophysical plasma, resulting in line emissivities having units of $\text{cm}^3 \text{ s}^{-1}$.

4.2. Models for Data Fitting

The parameters for the `apec` model are given in table 4.5.

Table 4.5: Model parameters for `apec`

Parameter	Quantity
par1	Plasma temperature (in keV)
par2	Abundances of the metals C, N, O, Ne, Mg, Al, Si, S, Ar, Ca, Fe, Ni
par3	Redshift z
norm	Normalization of the component computed as $\frac{10^{-14}}{4\pi[D_A(1+z)]^2} \int n_e n_H dV$, where D_A is the angular diameter distance to the source (in cm), n_e and n_H are the electron densities (in cm^{-3}) respectively

4.2.2.3 Model for emission due to optically-thin plasma: `mekal`

The additive model `mekal` in Xspec allows the simulation of an emission spectrum due to a diffuse plasma, whose electrons have a Maxwellian energy distribution. This model uses the spectral line list as calculated by Mewe and Kaastra [90], additional calculations for L-shell of Fe ions by Liedahl et al. [91]. The model provides the option to either calculate the spectrum by running the `mekal` code, or by interpolation on a pre-calculated `mekal` table, or simply by using the AtomDB data.

The models `mekal` and `apec` both simulate emission due to optically-thin plasma, the difference being in the methodology of calculation of the line lists.

The parameters for the `mekal` model are given in table 4.6.

4.2.2.4 Velocity shear absorption: `swind1`

Originally meant for AGN spectra, the `swind1` multiplicative model fits the soft excess in partially ionized absorbing material with a large velocity shear. This is approximated by the model component by using XSTAR kn5 photoionization absorption model grids, which

4.2. Models for Data Fitting

Table 4.6: Model parameters for `mekal`

Parameter	Quantity
par1	Plasma temperature (in keV)
par2	H density (in cm ⁻³)
par3	Metal abundances for the elements C, N, O, Ne, Na, Mg, Al, Si, S, Ar, Ca, Fe, Ni
par4	Redshift z
par5	Switch between <code>mekal</code> calculation (0), interpolation (1) and interpolation using AtomDB data (2)
norm	Normalization of the component computed as $\frac{10^{-14}}{4\pi[D_A(1+z)]^2} \int n_e n_H dV$, where D_A is the angular diameter distance to the source (in cm), n_e and n_H are the electron densities (in cm ⁻³) respectively

were calculated assuming a micro-turbulence of 100 km/s, and subsequently convolving with Gaussian smearing [92].

In this work, the `swind1` component is used as a proxy model for possible stellar wind from the source RX J0925.7-4758, which may be indicated by the presence of P Cygni profiles in its spectrum.

The parameters for the `swind1` model are given in table 4.7.

Table 4.7: Model parameters for `swind1`

Parameter	Quantity
par1	Column density (in 10 ²² cm ⁻²)
par2	$\log \xi$ where $\xi = L/nr^2$
par3	σ : Gaussian σ for velocity smearing (v/c)
par4	Redshift z

4.2. Models for Data Fitting

4.2.2.5 Tübingen NLTE Model-Atmosphere Package: `rauch`

The Tübingen NLTE Model-Atmosphere Package (TMAP) is a tool to calculate stellar atmospheres in spherical or plane-parallel geometry in hydrostatic and radiative equilibrium allowing departures from local thermodynamic equilibrium (LTE) for the population of atomic levels [93]. TMAP is based on the so-called Accelerated Lambda Iteration (ALI) method and is able to account for line blanketing by metals [94]. All elements from hydrogen to nickel may be included in the calculation with model atoms which are tailored for the aims of the user [95].

The web-link to a set of theoretical spectral energy distributions (SEDs) of TMAP NLTE model atmospheres were provided by Rauch [96], which contained a grid of 10 FITS files for varying temperatures. The abundances of various elements for this grid are given in table 4.8. The TMAP model series refer to the files corresponding to the model atmosphere grid, with each column named after the last three characters of the FITS filename. The grid is calculated for effective temperatures in the range $4.50 \times 10^5 \text{ K} \leq T_{\text{eff}} \leq 1.05 \times 10^6 \text{ K}$ in steps $\Delta T = 10^4 \text{ K}$. The effective surface gravity is $\log_{10} g = 9$. The fluxes in the SEDs are calculated using the TMAP code from models with different elemental abundance ratios $[X]$.

The quantity $[X]$ is logarithmic and is calculated as

$$[X] = \log \left(\frac{\text{stellar mass fraction}}{\text{solar mass fraction}} \right) = \log \left(\frac{X_*}{X_\odot} \right) \quad (4.1)$$

For any element X denotes its mass fraction which is defined as $X \equiv \frac{m_X}{M}$, where m_X is the mass of the element and M is the total mass of the system. For example in table 4.8,

$$\begin{aligned} [X] &= -0.675 \\ \implies \log_{10} \left(\frac{X_*}{X_\odot} \right) &= -0.675 \\ \implies \frac{X_*}{X_\odot} &= 10^{-0.675} \\ \implies X_* &= 0.211 X_\odot \end{aligned}$$

That is, $[X]$ indicates that the stellar mass fraction X_* for that element is about 21% that of

4.2. Models for Data Fitting

Table 4.8: Elemental abundances for TMAP grid

[X]	TMAP model series									
	003	004	005	006	007	008	009	010	011	201
[H]	-0.688	-0.683	-0.677	-0.673	-0.672	-0.671	-0.670	-0.670	-0.669	-0.885
[He]	0.382	0.387	0.393	0.397	0.398	0.399	0.400	0.401	0.401	0.489
[C]	-1.513	-1.073	-0.772	-0.675	-0.596	-0.529	-0.471	-0.420	-0.374	-0.057
[N]	1.803	1.678	1.460	1.159	1.062	0.937	0.761	0.460	0.159	1.668
[O]	1.528	1.533	1.538	1.543	1.544	1.544	1.545	1.546	1.547	1.206
[Ne]	-0.474	-0.469	-0.464	-0.459	-0.459	-0.458	-0.457	-0.456	-0.456	-0.517
[Mg]	-0.454	-0.450	-0.444	-0.439	-0.439	-0.438	-0.437	-0.436	-0.436	-0.497
[Si]	0.167	0.172	0.178	0.182	0.183	0.184	0.185	0.186	0.186	0.125
[S]	-1.583	-1.578	-1.573	-1.568	-1.567	-1.567	-1.566	-1.565	-1.565	-1.625
[IG]	0.828	0.833	0.838	0.843	0.843	0.844	0.845	0.846	0.846	0.786

solar mass fraction X_{\odot} .

The parameters for the `rauch` model are given in table 4.9.

Table 4.9: Model parameters for `rauch`

Parameter	Quantity
par1	Effective temperature T (in K)
par2	Redshift z
norm	Normalization of the component computed as $\frac{10^{-14}}{4\pi[D_A(1+z)]^2} \int n_e n_H dV,$ where D_A is the angular diameter distance to the source (in cm), n_e and n_H are the electron densities (in cm^{-3}) respectively

4.3 Analysis of High-resolution RGS Spectra

The spectra of RX J0925.7-4758 obtained by both of the RGS instrument of XMM-Newton were analyzed using a subset of models from the list given in table 4.3. Models with IDs M07, M08, M09, M10, M11 and M12 were applied on the RGS data for both spectral orders.

4.3.1 Fitting of RGS1 Spectra

We present here the results of fitting a selection of spectral models to the RGS1 data from RX J0925.7-4758. We assess the quality of each fit using the values of the reduced chi-squared statistic (χ^2_{reduced}). Acceptable models typically have χ^2_{reduced} values in the range of 1 to 2. The fitting statistics are summarized in table 4.10. These values are also plotted in figure 4.2 to visually identify models with acceptable χ^2_{reduced} values.

Table 4.10: Fitting statistics of RGS1 spectra from RX J0925.7-4758

Model ID	RGS 1 Order 1		RGS 1 Order 2	
	$\chi^2/\text{d.o.f}$	χ^2_{reduced}	$\chi^2/\text{d.o.f}$	χ^2_{reduced}
M07	1168.4/446	2.62	386.8/174	2.22
M08	1176.8/443	2.66	346.4/171	2.03
M09	1126.5/445	3.57	318.4/173	1.84
M10	1194.4/445	2.68	372.4/173	2.15
M11	1044.4/442	2.36	282.8/170	1.66
M12	1087.5/443	2.45	424.2/171	2.48

As evident from table 4.10 or figure 4.2, the model M11 provides the best fit for both spectral orders. While other models do not yield the best fit according to the χ^2_{reduced} statistic, they still provide a reasonable fit that surpasses those found in previous literature. This trend of χ^2_{reduced} across all the models is illustrated in figure 4.2.

4.3. Analysis of High-resolution RGS Spectra

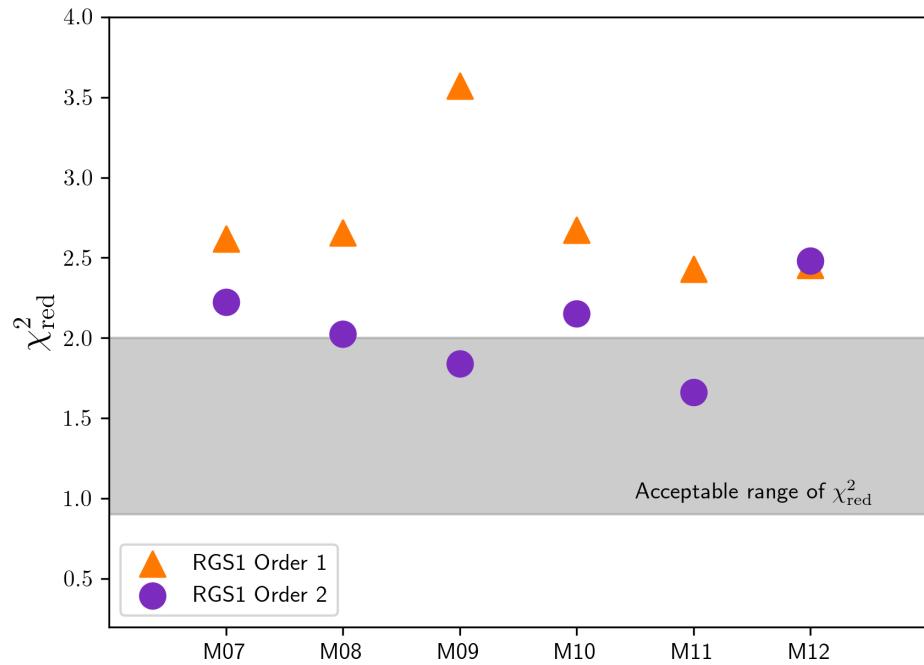


Figure 4.2: χ^2_{red} trend of RGS1 spectra from RX J0925.7-4758

Detailed spectral fits

The following set of figures present the fitted spectra along with their residuals for each model applied to RGS1 data. These figures allow for a visual inspection of the quality of the fit for each model.

4.3. Analysis of High-resolution RGS Spectra

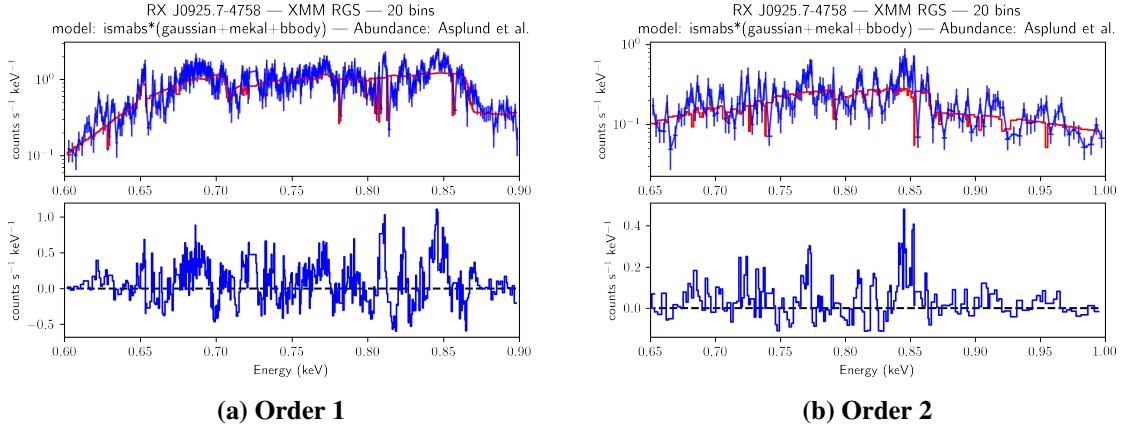


Figure 4.3: Model M07 fit to RGS1 spectra

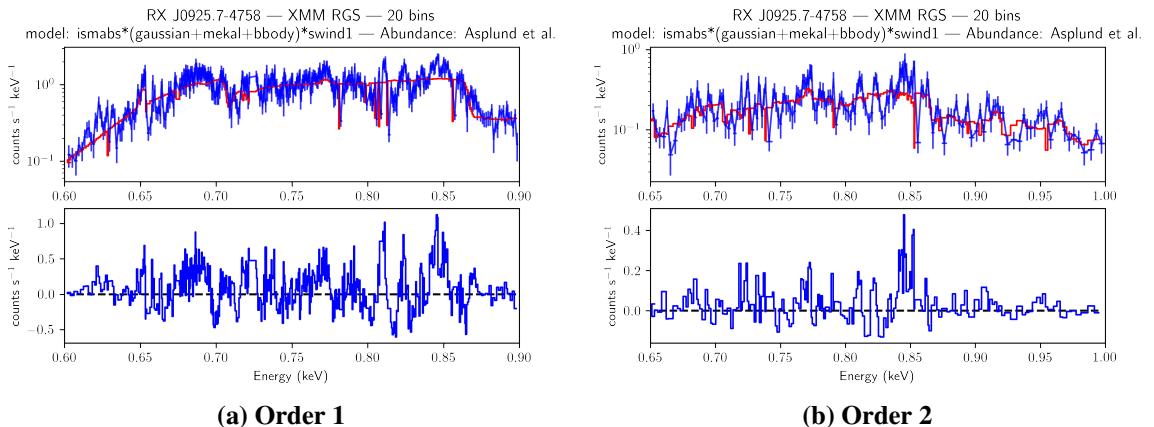


Figure 4.4: Model M08 fit to RGS1 spectra

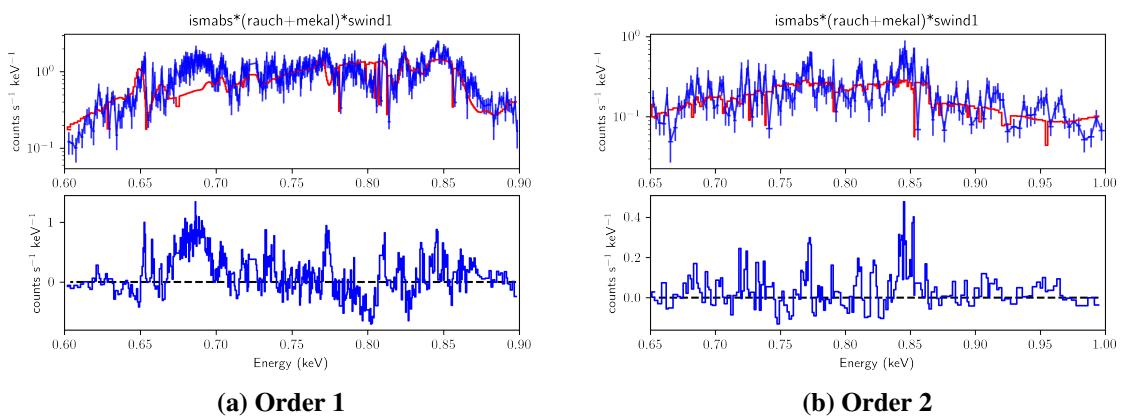


Figure 4.5: Model M09 fit to RGS1 spectra

4.3. Analysis of High-resolution RGS Spectra

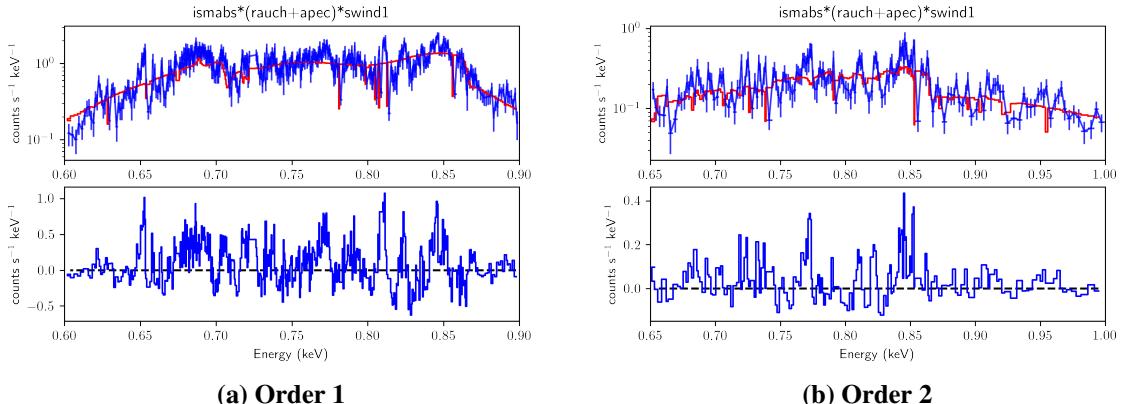


Figure 4.6: Model M10 fit to RGS1 spectra

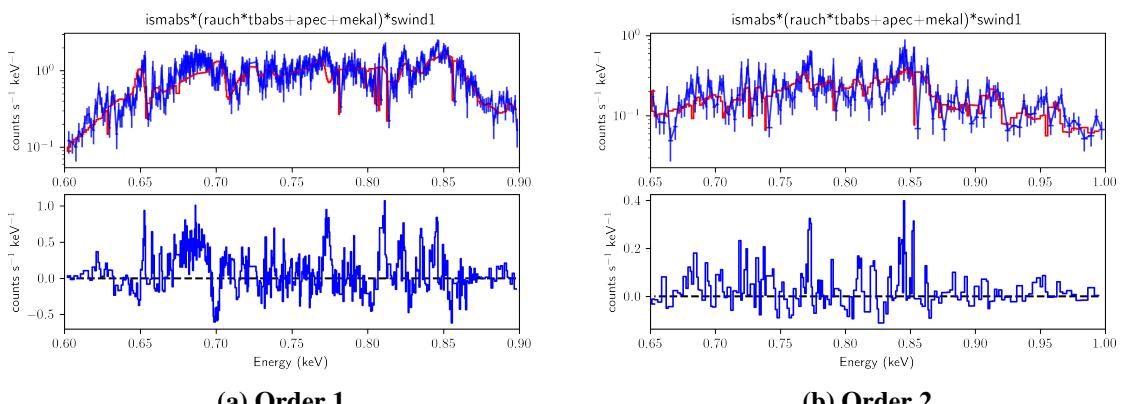


Figure 4.7: Model M11 fit to RGS1 spectra

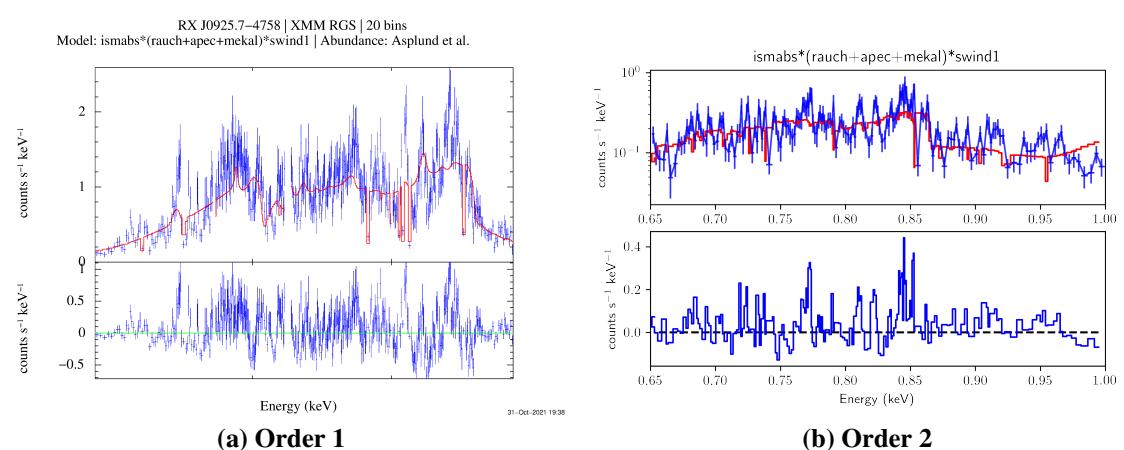


Figure 4.8: Model M12 fit to RGS1 spectra

4.3. Analysis of High-resolution RGS Spectra

4.3.2 Fitting of RGS2 Spectra

We present here the results of fitting a selection of spectral models to the RGS2 data from RX J0925.7-4758. We assess the quality of each fit using the values of the reduced chi-squared statistic (χ^2_{reduced}). Acceptable models typically have χ^2_{reduced} values in the range of 1 to 2. The fitting statistics are summarized in table 4.10. These values are also plotted in figure 4.2 to visually identify models with acceptable χ^2_{reduced} values.

Table 4.11: Fitting statistics of RGS2 spectra from RX J0925.7-4758

Model ID	RGS 1 Order 1		RGS 1 Order 2	
	$\chi^2/\text{d.o.f}$	χ^2_{reduced}	$\chi^2/\text{d.o.f}$	χ^2_{reduced}
M07	1506.9/553	2.73	497.3/201	2.47
M08	1353.8/550	2.46	462.0/198	2.33
M09	1576.8/552	2.86	434.6/200	2.17
M10	2176.7/552	3.94	423.8/200	2.12
M11	1173.8/549	2.14	355.2/197	1.80
M12	1465.1/552	2.65	445.9/198	2.25

As evident from table 4.11 or figure 4.9, the model M11 again provides the best fit for both spectral orders. While other models do not yield the best fit according to the χ^2_{reduced} statistic, with the exception of the models M09 and M10, which lies almost on the borderline for order 2 spectra. They still provide a reasonable fit that surpasses those found in previous literature. This trend of χ^2_{reduced} across all the models is illustrated in figure 4.9.

Detailed spectral fits

The following set of figures present the fitted spectra along with their residuals for each model applied to RGS2 data. These figures allow for a visual inspection of the quality of the fit for each model.

4.3. Analysis of High-resolution RGS Spectra

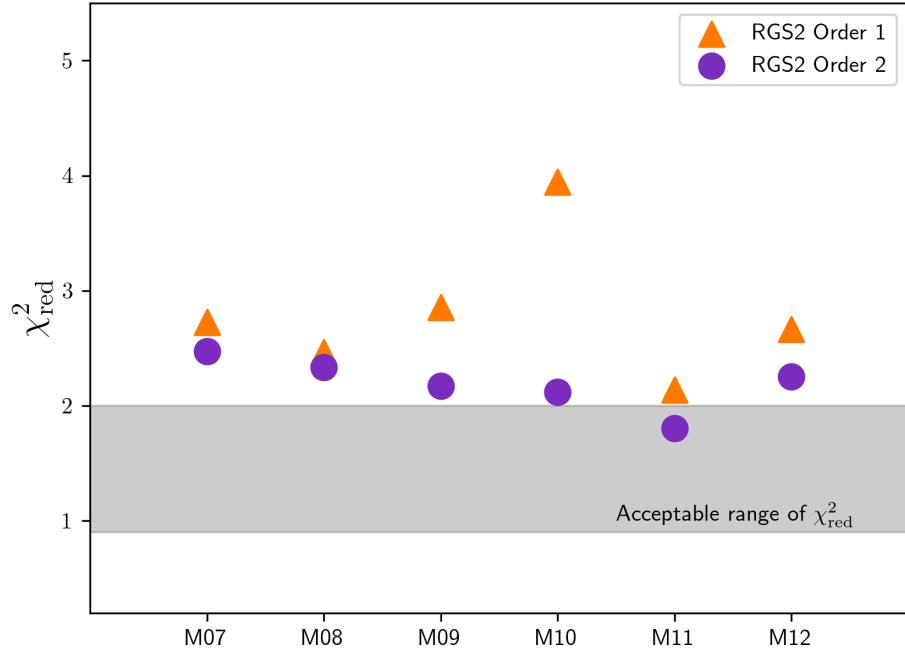


Figure 4.9: χ^2_{red} trend of RGS2 spectra from RX J0925.7-4758

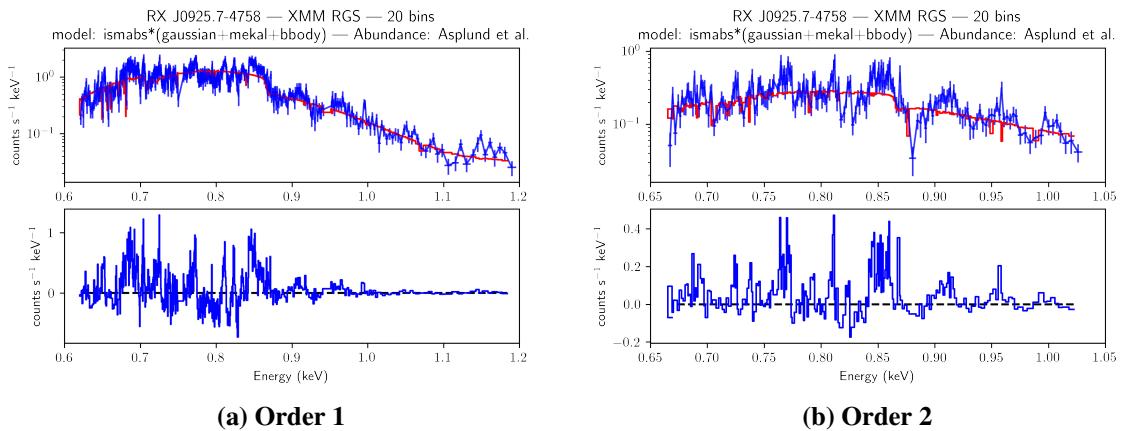


Figure 4.10: Model M07 fit to RGS2 spectra

4.3.3 Comparison with Current Literature

Here we present a comparison of the quality of the spectral fit achieved in this work with those reported in previous studies. Table 4.12 summarizes the reduced chi-squared (χ^2_{reduced})

4.3. Analysis of High-resolution RGS Spectra

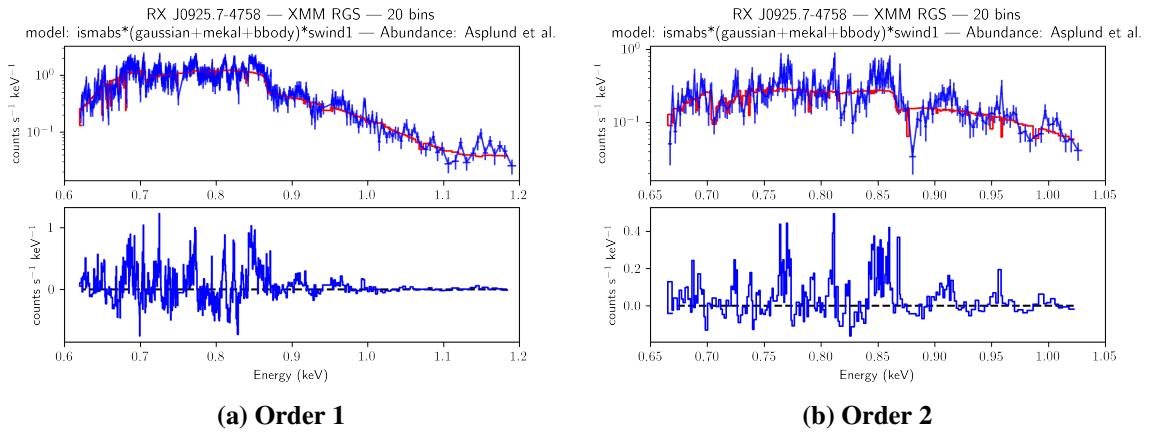


Figure 4.11: Model M08 fit to RGS2 spectra

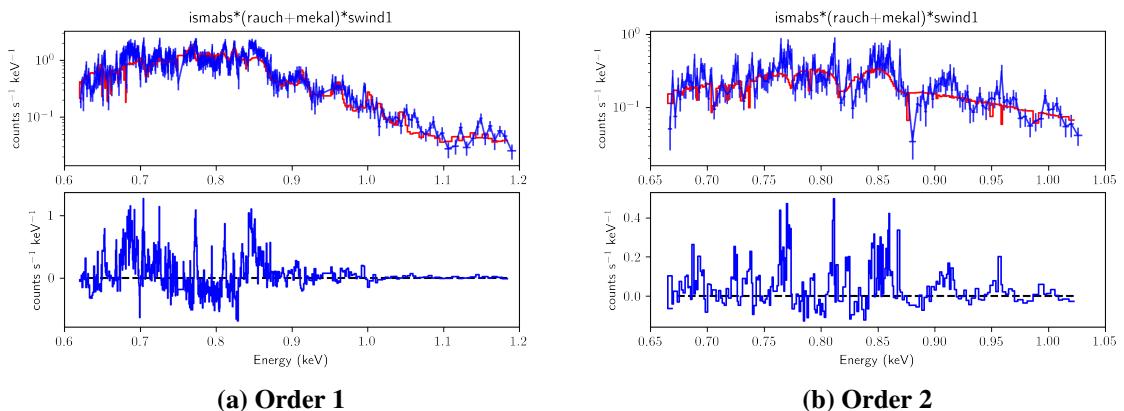


Figure 4.12: Model M09 fit to RGS2 spectra

values for the best-fitting model (M11) from our analysis, alongside results from existing literature.

As table 4.12 illustrates, the χ^2_{reduced} values obtained for model M11 in this work (ranging from 1.66 to 2.36) are significantly lower than the value reported by Ebisawa et al. (1996) [97]. Bearda et al. (2002) and Motch et al. (2002) do not provide χ^2_{reduced} values in their studies, merely reporting the near impossibility of modelling the RGS spectrum of RX J0925.7-4758 using an NLTE model atmosphere [98, 99]. The substantial improvement in fit quality achieved here suggests that model M11 seems to offer a more accurate representation of the physical processes governing the X-ray emission from RX J0925.7-4758.

Mathematically, the expression for model M11 can be written as,

4.3. Analysis of High-resolution RGS Spectra

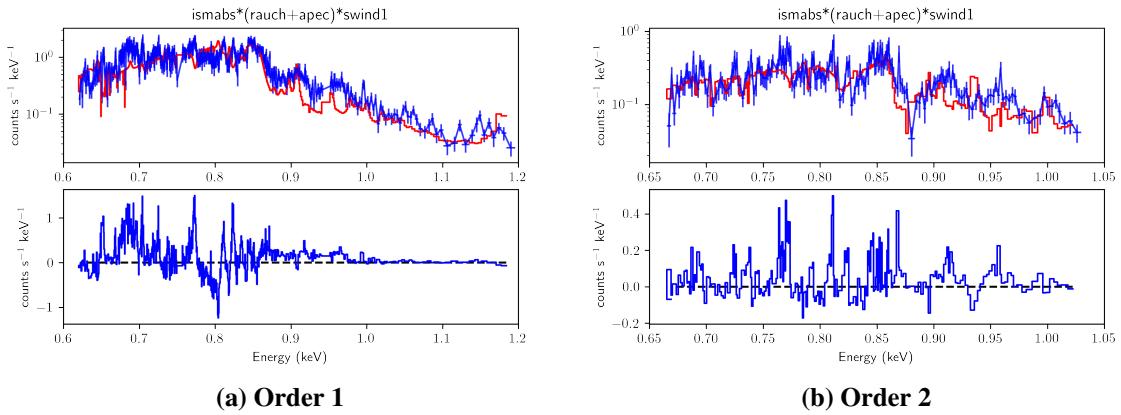


Figure 4.13: Model M10 fit to RGS2 spectra

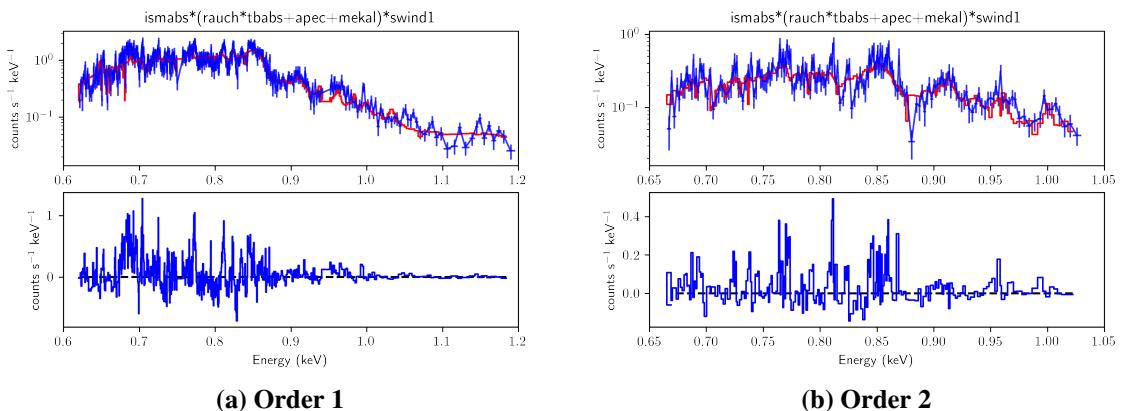


Figure 4.14: Model M11 fit to RGS2 spectra

ismabs*(rauch*tbabs+apec+mekal)*swindl

4.3.4 Supersoft X-ray Physics using Best-fit RGS Model

The model M11, by virtue of its individual components, sheds light on the physical mechanisms responsible for the X-ray emission observed from RX J0925.7-4758. The `ismabs` term specifically accounts for the partial absorption of X-rays along the line of sight due to the interstellar medium (ISM). The ISM, comprising primarily gas and dust, attenuates the incoming X-ray photons before they reach the object, effectively weakening the observed X-ray flux.

4.3. Analysis of High-resolution RGS Spectra

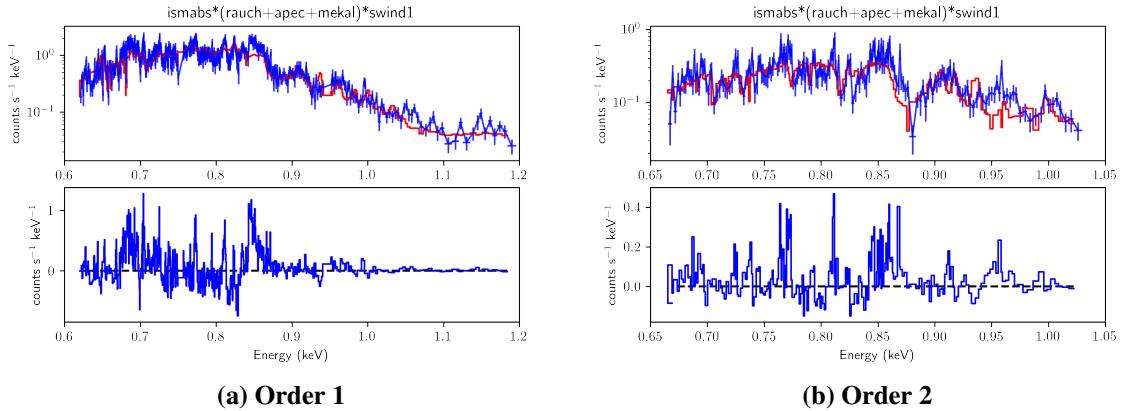


Figure 4.15: Model M12 fit to RGS2 spectra

Table 4.12: Comparison of best-fit statistics of RGS spectra from RX J0925.7-4758

Title of work	χ^2_{reduced}
Ebisawa et al. (1996)	10.0
Bearda et al. (2002)	Not quoted
Motch et al. (2002)	Not quoted
M11	RGS1 (order 1) 2.36
(current work)	RGS1 (order 2) 1.66
	RGS2 (order 1) 2.14
	RGS2 (order 2) 1.80

Delving deeper into RX J0925.7-4758, the `rauch*tbabs` component sheds light on the intrinsic X-ray emission. The `tbabs` element maintains its role as a standard model for photoelectric absorption by cold, denser gas within the object. However, the `rauch` term introduces a new layer of complexity. It represents an emission component arising from a non-local thermodynamic equilibrium (NLTE) atmosphere, which captures the effects of this specific atmospheric model on the X-ray emission. This suggests that the intrinsic X-ray emission from RX J0925.7-4758 might not be uniform but influenced by the properties of this NLTE atmosphere, potentially caused by deviations from thermal equilibrium within the emitting gas.

The narrative of X-ray emission from RX J0925.7-4758 takes a significant turn with the

4.3. Analysis of High-resolution RGS Spectra

inclusion of both `apec` and `mekal` terms. These terms represent distinct thermal plasma emission models. `Apec` is indicative of optically thin plasma, where the surrounding gas is relatively sparse. In such an environment, X-rays have a higher probability of traveling freely with minimal interactions before inducing radiative emission. `Mekal`, on the other hand, is suited for modeling thicker plasma regions. Here, X-rays are more likely to collide with particles within the gas, leading to a different emission mechanism. The presence of both models in the model M11 suggests that the hot gas surrounding RX J0925.7-4758 might exhibit variations in density. Consequently, the X-ray emission from this hot gas could potentially originate from two distinct processes depending on the local plasma conditions.

The final component, i.e. `swind1`, adds a layer of complexity by incorporating the possibility of deviations from a uniform velocity field within the emitting gas. This term suggests the presence of internal motions within the hot gas surrounding RX J0925.7-4758. These motions could be due to various factors, such as turbulence, bulk flows, or outflows. The inclusion of `swind1` acknowledges the dynamic nature of the emitting region and highlights the potential need for more sophisticated modeling approaches to fully characterize the physical processes at play.

CHAPTER 5

TOOL FOR IDENTIFICATION OF SPECTRAL LINES IN XMM-NEWTON RGS SPECTRA

Contents

5.1	Introduction	96
5.2	Working with RGS Data Files	96
5.2.1	Extraction of RGS Spectra	98
5.2.2	Accessing Spectral Data from FITS Files	99
5.2.3	Collecting Atomic Lines	100
5.2.4	Demonstration	101
5.2.5	Distribution	109
5.2.6	Future scope	110
5.3	Stellar Winds from Hot Luminous Massive Stars	112
5.3.1	Mass-Loss Rate	112
5.3.2	Terminal Velocity	112
5.4	Relevance of Stellar Winds	113
5.5	Radiative Line-Driving: Mechanism of Stellar Winds	114

CHAPTER 5. TOOL FOR IDENTIFICATION OF SPECTRAL LINES IN XMM-NEWTON RGS SPECTRA

Abstract

The identification of absorption and emission lines due to atomic species in the spectra of a X-ray binaries can reveal a wealth of information regarding the composition and physics of the stellar atmosphere. With the availability of high-resolution X-ray spectra from the RGS equipment on-board the XMM-Newton satellite, the study of such lines offers valuable diagnostics into the behaviour and evolution of the source object. Currently, data related to various atomic transitions, which lead to line formation, are made publicly available in the form of credible databases, one of them being the Atomic Spectra Database (ASD) at the National Institute of Standards and Technology (NIST). In this chapter, we present the development of a Python-based tool that accesses the relevant atomic data at NIST ASD for a given set of atomic species in a specific wavelength range and then overlays these lines on top of an X-ray spectrum obtained by the RGS spectrum of the XMM-Newton. With this tool, the astronomer can perform the important preliminary task of rudimentary line identification, before proceeding to advanced analysis of the X-ray data.

5.1 Introduction

XMM-Newton is an X-ray space observatory which was launched by the European Space Agency on December 10, 1999. It is designed to be a high throughput X-ray spectroscopy mission with spectral resolution of up to 0.025 Å at 1 keV from the *Reflection Grating Spectrometer* (RGS) detectors and an angular resolution of up to 1.1 arcsec from the *European Photon Imaging Cameras* (EPIC) [100, 6]. With a bandpass of 5–38 Å, corresponding to the energy range 0.33–2.5 keV, the spectra detected by RGS spans a substantial number of X-ray emission lines, which include the K-shell transitions and He-like triplets of light elements, such as C, N, O, Ne, Mg and Si, as well as the L-shell transitions of heavier elements like Fe and Ni. These factors enable the RGS spectra to be useful as diagnostic tools that can be used to investigate the physical conditions as well as the composition of the source of the spectra.

5.2 Working with RGS Data Files

The RGS spectra are available in the public domain in the *Flexible Image Transport System* (FITS) file format [101], which is an open standard describing the digital file format for the storage, transmission and processing of data – formatted as multi-dimensional columns of tables. In spite of having the word ‘image’ in its acronym, FITS files are most often used to store non-image data as well. This standard was designed, keeping in mind specifically astronomical data, namely images, spectra, lightcurves, photon lists, event lists, and source lists. In order to download the raw science data collected by XMM-Newton for any given source, there are two options:

- a) By accessing the *XMM-Newton Science Archive* (XSA) at the European Space Agency (ESA), using a web service (via a search form or a URL command access or methods from the `astroquery.esa.xmm_newton` Python module) or by *table access protocol* (TAP) queries to the XSA database [102].
- b) By accessing the *High Energy Astrophysics Science Archive Research Center*, or HEASARC, at the National Aeronautics and Space Administration (NASA) using a web service [103].

The XMM-Newton Science Operations Centre (SOC) provides a software package called the

5.2. Working with RGS Data Files

Science Analysis System (SAS)¹, which is a collection of tasks, scripts and libraries designed for the specific tasks of the reduction and analysis of the raw science data collected by the XMM-Newton observatory [104]. Using SAS, one can extract an RGS spectrum from the science data for a given X-ray source.

An X-ray spectrum from RGS is found to contain a multitude of absorption and emission lines corresponding to atomic transitions in elements heavier than H and He. One of the initial and crucial tasks of the observer is to make a preliminary identification of the elemental lines at the wavelengths where they appear. This line identification also needs to incorporate the shift in the apparent position of the wavelength due to the Doppler effect. Currently, such line identification is performed by spectral-fitting programs such as Xspec or Spex while fitting the spectrum to prior theoretical models in which one can include the desired elemental abundances. However, when the spectrum used is of very high-resolution (such as that obtained from XMM-Newton), it often leads to poor fits, especially when one includes non-LTE model atmospheres. A Python-based tool was developed to address these deficiencies with the following motivation:

- a) To upend the approach for fitting such high-resolution spectra by allowing the user to first quickly identify the prominent lines present in the spectrum as well the Doppler shift in the lines (if any).
- b) To estimate the elemental abundances (from the presence of specific lines) and the radial velocity of the emitting material (from the Doppler shifts).
- c) To allow the user to develop models for the individual lines and thereby engineer composite theoretical models which are more phenomenological as opposed to traditional approaches [64].

As present, while there are available codes that enable the parsing of atomic line data from NIST ASD (such as the `nist-asd` Python package on PyPI) or object-oriented interface for XSPEC (i.e. `PyXspec`), there is a lack of tools or codes that allow *a priori* identification of spectral lines identification.

In order get started, one first needs to obtain data of the atomic lines from credible databases, such as the *Atomic Spectra Database* (ASD)² at the *National Institute of Standards and Tech-*

¹<https://www.cosmos.esa.int/web/xmm-newton/sas>

²<https://www.nist.gov/pml/atomic-spectra-database>

5.2. Working with RGS Data Files

nology (NIST)³ [105] and *The Opacity Project online atomic database* (TOPbase)⁴ [106]. To obtain this data, one may either use the web service provided to send queries to these databases, or one may use the relevant methods in various Python modules.

Having identified the prominent lines in the RGS spectrum, one can then proceed to estimate the radial velocity of the source, the abundance of the elements in the emitting material and the interstellar absorption along the line-of-sight.

5.2.1 Extraction of RGS Spectra

5.2.1.1 Obtaining source-specific science data

XMM-Newton data is publicly available at the online multi-mission science data archive known as HEASARC⁵. The easiest way to access relevant data for any specific source is by using its browse service⁶ where one can enter the name of the source (or its right ascension and declination, if these are known) and the mission name (such as XMM-Newton or Chandra). One can supply additional information about the source as well, such as the observation dates. The more specific the query, the lesser would be the time taken to access and download the data.

HEASARC provides the relevant data products in the form of a `.tar` file, which may then be extracted to obtain the *Observation Data Files* (ODF), in case of XMM-Newton.

5.2.1.2 Reprocessing of science data

The downloaded ODF files are compressed in `.gz` format and need to be extracted before reprocessing them using the software package SAS. Two SAS commands are used during the reprocessing – `cifbuild` and `odfingest`. While the former produces an index file of all the calibration data relevant to the specific source under consideration, the latter creates a summary file of the ODF using the house-keeping data files and the calibration database [104].

³<https://www.nist.gov/>

⁴<https://cds.unistra.fr/topbase/topbase.html>

⁵<https://heasarc.gsfc.nasa.gov/>

⁶<https://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl>

5.2. Working with RGS Data Files

Subsequently, the data from both RGS instruments for the first and second spectral orders is extracted using the SAS task `rgsproc`. In five processing stages, this task extracts the events, the angles, the filter events, the spectra, the response matrices and the lightcurves.

5.2.2 Accessing Spectral Data from FITS Files

One of the outputs of the `rgsproc` task are two fluxed spectra – one for each spectral order. Each of these files contain the spectra fluxed from both the RGS instruments. One must bear in mind that these fluxed spectra are inherently just a qualitative summary of the data, and should not be used for quantitative analysis. However, in the current work, because the objective is to merely aid the user in the identification of atomic lines – which could be a preliminary step before serious spectral modeling, these fluxed spectra are more than sufficient as input files.

5.2.2.1 The `astropy` Python module

The fluxed spectrum file is saved in the FITS format. So the appropriate way to read its contents and extract the data is by using `astropy`⁷ [107], which is an open-source and community-developed Python package providing various functionalities that are important to astronomical applications.

The `astropy.io.fits` package provides the user with access to FITS files. One first reads the list of *header data units* (HDU), which is the highest level component of a FITS file – consisting of a header and a table (or data array). This can be done using the `astropy.io.fits.open()` function which takes the FITS filename as input and returns an object of the `HDUList` class – a list-like collection of HDUs.

5.2.2.2 Reading spectral data from FITS file

One can access the individual HDUs in a FITS file using the array indexing syntax – the first index pointing to the primary HDU by default. For instance, if the object returned by the `astropy.io.fits.open()` function is named `hdul`, then `hdul[0]` would be the

⁷<https://www.astropy.org/>

5.2. Working with RGS Data Files

primary HDU, `hdul[1]` the first extension HDU and so on.

In case of the fluxed spectra from RGS, the FITS files contain a single extension HDU corresponding to the spectrum – accessed using `hdul[1]`, which is a table containing 3600 rows and 3 columns. Then the data attribute of `hdul[1]` (or `hdul[1].data`) would return a NumPy numerical array containing the channels/wavelengths (in Å), the flux and the errors (both in $s^{-1}cm^{-2}\text{\AA}^{-1}$). The fluxes and errors corresponding to all bad channels contain NaN values, which can be filtered using the `isnan()` function and set to zero.

5.2.3 Collecting Atomic Lines

In order to be able to identify atomic lines in a spectrum, one needs to have a line list that contains information about the lines present within a given wavelength range for all the ion stages of the atoms under consideration. With tremendous improvement in computing power, the various online databases of atomic data provide more accurate and updated information that may be used to build better line lists.

So a code for qualitative atomic line identification in a spectrum becomes more reliable if it is able to access the current atomic data available online. This work achieves this objective by retrieving data for atomic transitions directly from ASD at NIST during runtime. So it is important to have an internet connection available while running the code for the first time or when one modifies the list of ions considered. As long as one keeps running the code subsequently on the same terminal session, using an unaltered ion list, the code would proceed to use the atomic data which is cached locally and one may work offline in that case.

5.2.3.1 The astroquery Python module

Atomic data from NIST can be accessed using the `astroquery` module⁸ [108], which uses the Python `requests` package for making HTTP requests and utilizes the data parsing functionality of the `astropy` package. The NIST ASD can be sent query requests and data retrieved therefrom using the `astroquery.nist` module.

⁸<https://astroquery.readthedocs.io/en/latest/>

5.2. Working with RGS Data Files

5.2.3.2 Reading atomic line data from NIST

The `astroquery.nist.Nist.query()` function provides an easy way to send HTML requests to NIST ASD. This function takes, as inputs, the lower and upper wavelength values and the name of the ion to be considered. One can also make use of the `astropy.units.AA` attribute to convert the wavelength value to Å units.

This function returns an object of the `astropy.table.Table` class. Then individual table rows can be accessed by the indexing syntax. The elements of a row can be subsequently accessed using the corresponding column header as a dictionary key. For instance, if the object named `ion_table` stores the output of the `query()` function, then one can access the Ritz wavelength on the 17th row as `ion_table[16]['Ritz']`.

5.2.4 Demonstration

What follows is a demonstration of the developed tool using the flux spectrum of the galactic luminous supersoft X-ray source RX J0925.7-4758, which was discovered in the Galactic Plane Survey of the ROSAT All-Sky Survey [56]. The specific observations were made during 16–17 December 2000 with the RGS exposures being carried out for a duration of 61.2 ks [99] under the observation ID 0111150101. The ODF files were downloaded from HEASARC and RGS data were processed using the SAS task `rgsproc` under default settings.

For the identification of atomic lines for specific ion stages present in the spectrum, the fluxed spectrum file named `P0111150101OBX000fluxed1000.FIT` (produced by `rgsproc`) is sufficient for the code to work. This specific file fluxes together the first-order spectra from both the RGS instruments. A copy of this file is provided with the distribution of the code for the user to reproduce the demonstrations that follow, as well as to try out variations of the run.

Three specific Python scripts are provided with the distribution, whose outputs are presented in this work. A study of these scripts provides the user with an experience of how to use the methods developed in this work. All the scripts take, as the second command-line argument, the name of the file containing the fluxed spectrum. Briefly, these scripts perform the following tasks:

5.2. Working with RGS Data Files

- i. `line-overlaid-flux.py`: computes all the atomic lines within the wavelength range chosen and produces a plot of the fluxed spectrum with these lines overlaid.
- ii. `flux-inspect.py`: allows the user to examine specific regions of interest, such as the vicinity of a known atomic line, within the chosen wavelength range.
- iii. `line-shift.py`: displays the Doppler shifts in the Lyman α , β , and γ lines for the H-like and He-like C, N and O ions, as well as some common Fe ion lines.

5.2.4.1 Reading fluxed spectrum from FITS file

The use of the `astropy.io.fits` module is wrapped up in a method named `get_flux`. This method takes a single argument, which is the name of the FITS file containing the fluxed spectrum – in this case, the file named `P0111150101OBX000fluxed1000.FIT`. It returns the wavelengths (in Å), the normalized flux and the error (both in $s^{-1}cm^{-2}\text{\AA}^{-1}$) in the form of three lists, which can be subsequently manipulated.

5.2.4.2 Preview of spectrum

When one is working with the spectrum of a new source, one may not be aware of the wavelength region which contains most of photon flux. Also, RGS spectra tend to have a very low SNR in the small wavelengths up to ~ 7 Å. So, in order to help the user to ascertain a more pragmatic range, the function named `preview_flux` quickly plots the entire spectrum. This method takes, as arguments, two lists – the wavelengths in Å and the flux. The preview of the first-order fluxed RGS spectrum for RX J0925.7-4758 is shown in figure 5.1.

As it can be seen from the preview in figure 5.1, there seems to be plenty of information in the vicinity of 6 Å. But this is not usable because it is just noise arising from the low effective area of the RGS instrument in the low wavelength region which severely limits the resolving power of the RGS instrument, as described in the XMM-Newton Users Handbook⁹ [109]. Also, the region above 26 Å has a very low SNR. Looking at the wavelength region between 10 Å and 26 Å one can clearly visualise a typical spectrum as expected from a luminous

⁹https://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/uhb/node1.html

5.2. Working with RGS Data Files

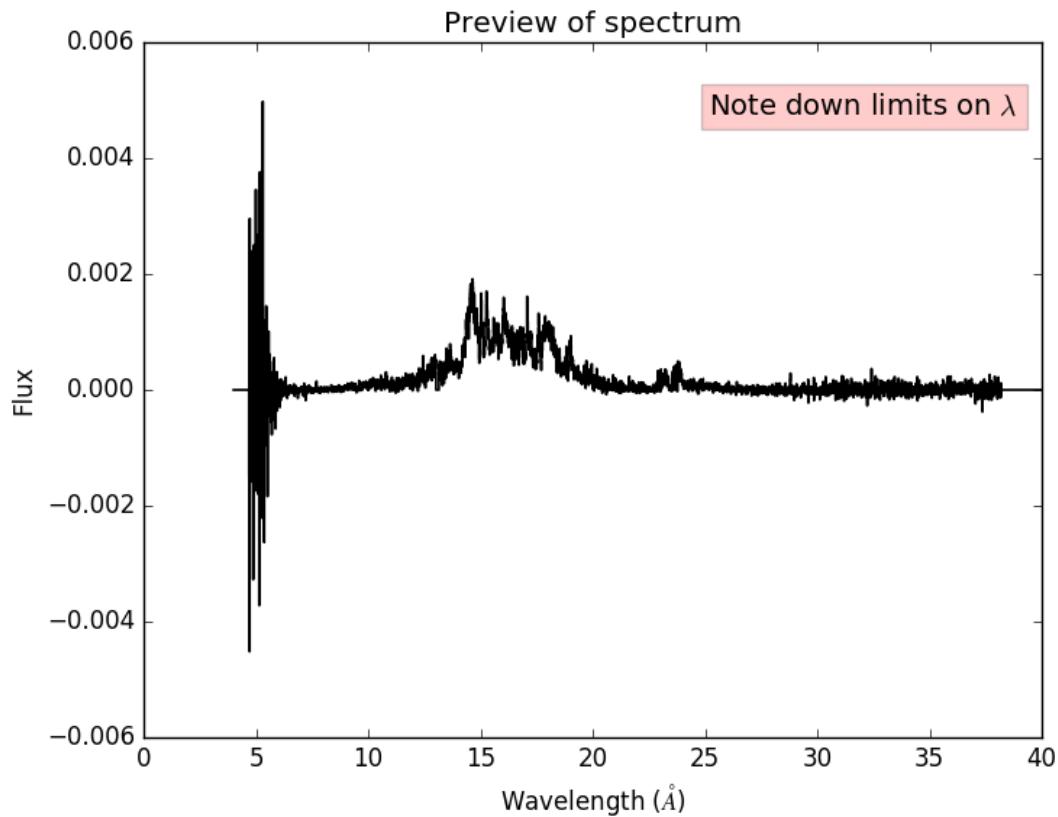


Figure 5.1: Preview of fluxed spectrum

X-ray binary source with a good SNR. These wavelength limits are subsequently taken as inputs from the user by the method `get_wave_limits()`.

5.2.4.3 Line list within chosen wavelength range

One needs the line list from the NIST database for the given list of ions within a chosen wavelength range. To achieve this, a method called `get_lines()` is provided. This takes, as input arguments, the lower and upper wavelength values (in Å). It uses the list of ions provided in the file `ion.list`, which is accessed by the script using the `ASTRODAT` environment variable. Then it queries the NIST database using the `astroquery.nist.Nist.query()` method.

The `get_lines()` method returns a list of dictionaries, with each dictionary containing four entries: the wavelength (in Å), the associated ion, a boolean flag which indicates whether that line needs to be included and the colour with which the line is to be plotted. The

5.2. Working with RGS Data Files

colour of a line for a specific ion is obtained from another method called `get_colour()`. This method uses a pre-defined dictionary of colour maps, from which a colour for a particular ion stage is sampled with the help of a NumPy array. One may add more colour maps corresponding to other ions which may not be included in the code at present.

This method has two more default arguments that allow the user to provide a Doppler shift and to filter out ground level transitions. These arguments are as follows:

- i. An argument `v_radial` to provide a radial velocity v_{rad} (in km/s), which is then used to compute the Doppler-shifted position λ of the line at λ_0 as

$$\lambda = \left(1 + \frac{v_{\text{rad}}}{c}\right) \lambda_0 \quad (5.1)$$

The default value of this argument is 0.

- ii. A flag named `ground_transitions_only` which ensures that only those lines are considered which are a consequence of transitions into/from the ground level of the individual ion. The default value of this argument is `True`, which greatly reduces the density of lines being taken into account.

5.2.4.4 Line-overlaid spectrum

Having read the FITS file containing the fluxed spectrum and queried the line list from NIST, one can then proceed to produce a plot of the spectrum, along with the lines overlaid on it. This can be done using the `plot_spectrum()` method. This method takes, as input arguments, the lists containing the wavelengths, the fluxes, the errors in the flux, the line list and the lower and upper limits of wavelength.

This method produces a set of three subplots: a plot of the spectrum only, a map of the lines within the wavelength region and a plot of the spectrum with the pertinent lines overlaid. The output of this code for the fluxed spectrum of RX J0925.7-4758 is shown in figure 5.2. As expected, one can observe a profusion of spectral lines due to K-shell transitions of N and O, and due to L-shell transitions of the heavier element Fe. Such an overlay of spectral lines provides a grid of markers of prominent lines. But the actual presence of any line in the spectrum may be latched on to after a closer inspection of sub-regions, as described in §5.2.4.5.

5.2. Working with RGS Data Files

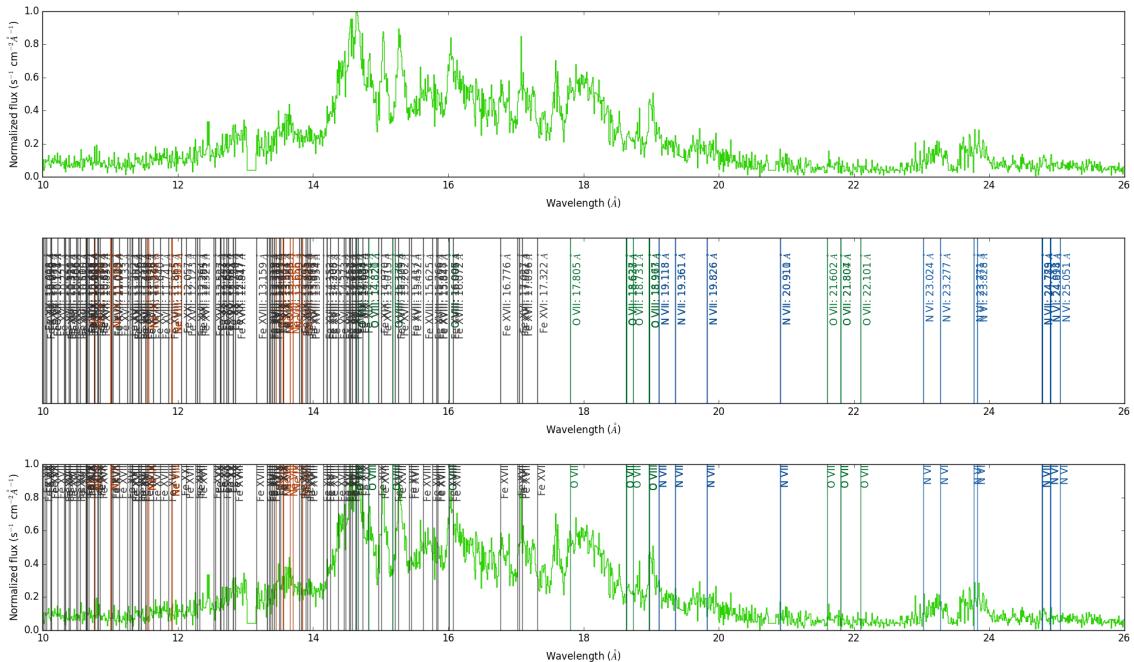


Figure 5.2: Line-overlaid spectrum using the `plot_spectrum()` method

Prior to running the `plot_spectrum()` method, one may also choose to display the wavelength values (in Å) along with the ion label, using the boolean flag `show_wavelength` input argument, whose default value is `False`.

One may also choose to normalize the fluxed spectrum to the range $[0, 1]$ in $\text{s}^{-1}\text{cm}^{-2}\text{\AA}^{-1}$ using the method `normalize()`, which takes a list of floats as input and returns another list of floats normalized appropriately. This is helpful in comparing the relative strengths of different lines.

5.2.4.5 Inspecting regions of interest

While the previous method described produces the entire spectrum with overlaid lines, it can be more helpful if one could precisely zoom into any specific region of interest, say between 16.5 Å and 18.4 Å. This can be done with the `examine_ROI()` method, which takes the lists of wavelengths, fluxes and errors, the line list, the chosen wavelength range and the lower and upper wavelengths (in Å) of the region of interest (e.g. 16.5 and 18.4).

Such an inspection of specific regions of interest was carried out for RX J0925.7-4758 and

5.2. Working with RGS Data Files

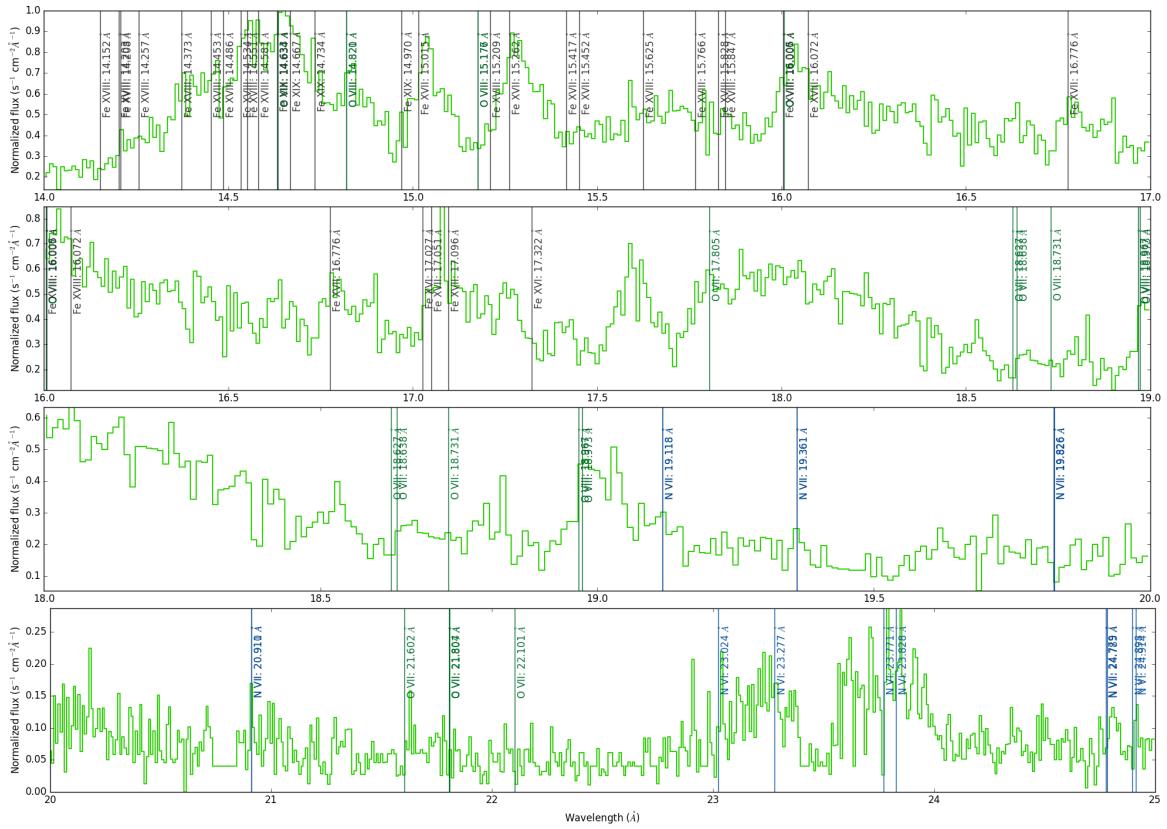


Figure 5.3: Partitioning of the fluxed spectrum within specific regions of interest

the resulting plots are shown in figure 5.3. One can observe here, for instance, two lines (one due to Fe XVII and the other due to O VIII) nearly overlapping with an emission line in the spectrum around 16.03 Å. Further analysis of the flux of other lines due to both atomic species would resolve this ambiguity.

5.2.4.6 Doppler shifts of lines

The emitting matter in stellar sources often have a radial velocity along the line-of-sight because of which their atomic lines often exhibit a Doppler shift, which translates the lines either blue-ward (i.e. towards smaller wavelengths) if the radial velocity is negative (moving towards the observer) or red-ward (i.e. towards larger wavelengths) if the radial velocity is positive (moving away from the observer).

This shift in the wavelength of a line can be obtained as

5.2. Working with RGS Data Files

$$\Delta\lambda = \lambda - \lambda_0 \quad (5.2)$$

where λ_0 is the central wavelength of the line and λ is the apparent position of the same line. The relation between these two is given by equation 5.1. By measuring the $\Delta\lambda$ value from the spectrum for some prominent and well-known atomic lines, one can estimate the radial velocity of the stellar matter which has produced the given spectrum.

One can make an initial estimate of the Doppler shift by plotting the vicinity of a specific line by transforming the horizontal axis from λ (in Å) to (in km/s). Then v_{rad} is obtained by inverting equation 5.1 as

$$v_{\text{rad}} = \left(\frac{c}{\lambda_0} \right) (\lambda - \lambda_0) \quad (5.3)$$

A method named `waveshift_to_velocity()` is defined which returns v_{rad} in km/s, given λ_0 and λ in Å as input arguments.

The method named `plot_doppler_lines()` can be used to plot the relative flux of different lines for a specific ion with respect to v_{rad} . At present, this method can handle the Lyman α , β , and γ lines for the H-like and He-like C, N and O ions, as well as some common lines of Fe XVII and Fe XVIII. The list of these lines is given in table 5.1.

Another method named `map_spec_to_vel()` maps the λ values to the values in the vicinity of a particular line using the previously described methods. It takes as input arguments the central wavelength of the line λ_0 , the lists containing the wavelengths, fluxes and errors, and an argument called `v_lim` which defines the bounds of v_{rad} values, the default value of which is ± 5000 km/s.

Finally, the method named `plot_doppler_lines()` wraps all of these and plots the line profiles for all the lines given in table 5.1. In case of RX J0925.7-4758, no lines of C V and C VI and also for N VI were detected within the chosen wavelength range. Shown in the figures 5.4, 5.5 and 5.6 are the line profiles obtained for the N VII, O and Fe ions. The `plot_doppler_lines()` method takes, as input arguments, the symbol of the element ('C', 'N', 'O' or 'Fe'), the list of wavelengths, fluxes and errors and the symmetric bound of the radial velocity about 0 (which corresponds to the line centre).

5.2. Working with RGS Data Files

Table 5.1: List of lines along with transitions, considered for inspection with respect to the radial velocity

Ion	Transition	Wavelength (in Å)
C V	Ly α	40.2678
	Ly β	34.9728
	Ly γ	33.4262
C VI	Ly α	33.7396
	Ly β	28.4663
	Ly γ	26.9901
N VI	Ly α	28.7870
	Ly β	24.8980
	Ly γ	23.7710
N VII	Ly α	24.7846
	Ly β	20.9106
	Ly γ	19.8261
O VII	Ly α	21.6020
	Ly β	18.6270
	Ly γ	17.7680*
O VIII	Ly α	18.9725
	Ly β	16.0067
	Ly γ	15.1765
Fe XVII	$2s^2 2p^6 \rightarrow 2s^2 2p^5 3d$	15.2620
	$2s^2 2p^6 \rightarrow 2s^2 2p^5 (2P_{3/2}^o) 3s$	17.0510
	$2s^2 2p^6 \rightarrow 2s^2 2p^5 (2P_{1/2}^o) 3s$	16.7760
Fe XVIII	$2s^2 2p^5 \rightarrow 2s^2 2p^4 (1D) 3d$	14.2080
	$2s^2 2p^5 \rightarrow 2s^2 2p^4 (3P) 3d$	14.3730*
	$2s^2 2p^5 \rightarrow 2s^2 2p^4 (3P) 3s$	16.0050*

* Obtained from the Chianti database [110]

It also takes another argument named mode which serves to draw a dotted vertical line marker either through the lowest flux or the highest flux or both (within the range of radial velocities considered). Either of these options may be chosen by setting `mode='abs'` (default value) or `mode='ems'` respectively. One can also set `mode='Pcyg'` to display both the markers simultaneously. However, one must be cautious while drawing any conclusions about the true position of the line centre from these markers because its position is merely ascertained

5.2. Working with RGS Data Files

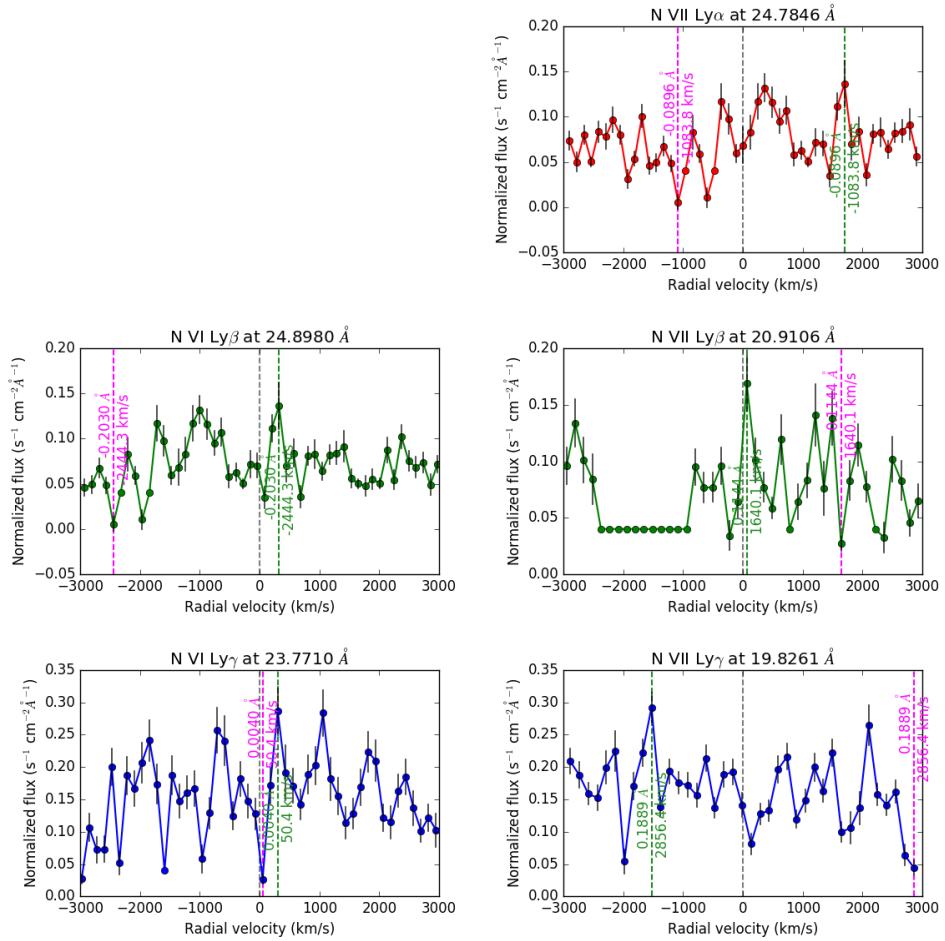


Figure 5.4: Velocity profiles of the Lyman α , β , and γ lines for N VI (left) and N VII (right)

from the flux values and are meant only to aid the user to get started with an approximate indication about the nature of the shift. For a more accurate evaluation, one needs to rely on a detailed quantitative analysis of the spectrum by using theoretical models.

5.2.5 Distribution

The current version of the entire Python tool is publicly available for download and free use on Github at

https://github.com/pararover/xmmrgs_lines

By making the code publicly available, critical evaluation and suggestions are desired, which

5.2. Working with RGS Data Files

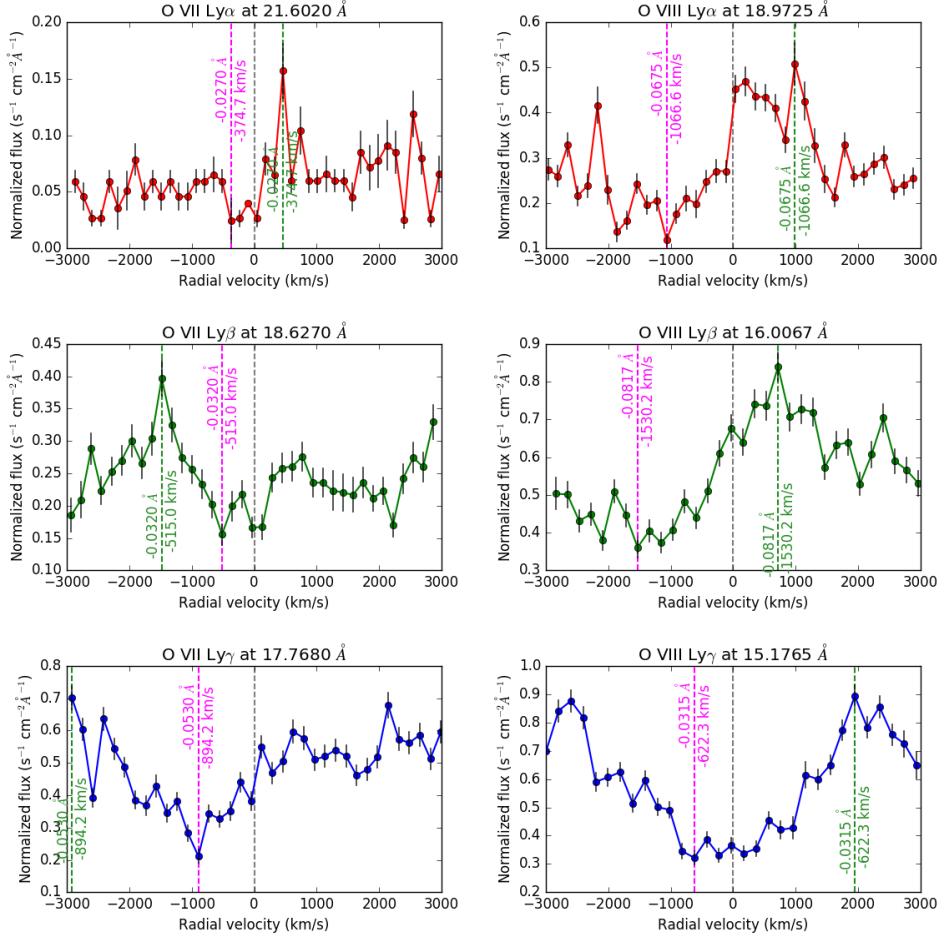


Figure 5.5: Velocity profiles of the Lyman α , β , and γ lines for O VII (left) and O VIII (right)

would enable further improvement to the code. Future versions of the code would evolve on the basis of such feedback from an open community.

5.2.6 Future scope

At present, the code is designed to work only with the RGS fluxed spectra from XMM-Newton. While there is no plan to extend its application to binned spectra, it is however planned that future versions of the code would be extended to work on the X-ray spectra from the EPIC MOS and PN instruments of XMM-Newton.

Currently, the range of velocity profiles produced by the code is highly restrictive – it provides the same only for the first three Lyman lines for the H-like and He-like ion stages of C,

5.2. Working with RGS Data Files

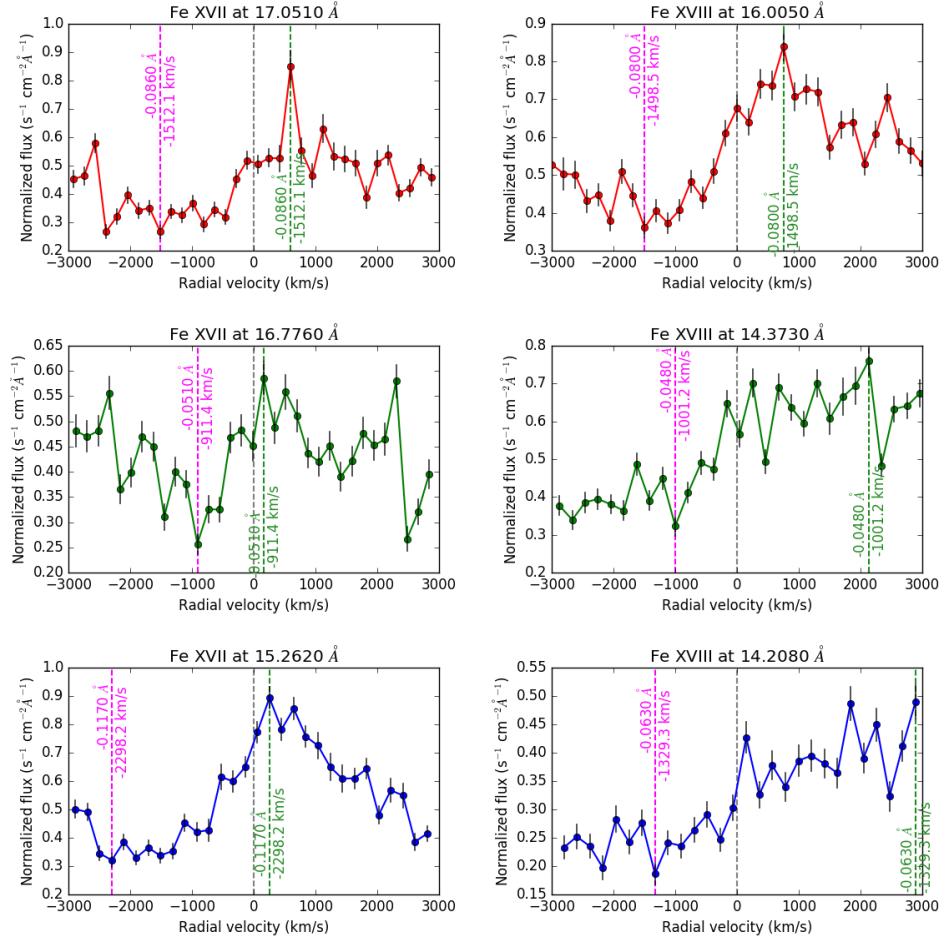


Figure 5.6: Velocity profiles of three Fe XVII (left) and Fe XVIII (right) lines

N, O and Fe only. In future versions, the authors intend to introduce further generalization by allowing the same for any ground transition of any ion, provided any line for the same exists within the chosen wavelength range. The filtering of lines, however, is planned to remain on ground transitions only. The reason for this being that transitions between excited states are much less likely. Also, it is planned that lines could be additionally filtered out on the basis of transition probability – this would help reduce the cluttering of Fe lines in the lower wavelengths (see the 10 Å – 16 Å range in figure 5.2).

5.3 Stellar Winds from Hot Luminous Massive Stars

Hot luminous massive stars are those with effective temperatures $\sim 10000 - 50000$ K (i.e. spectral type A0 - O2), luminosities $\sim 10^{37} - 10^{39}$ erg/s and masses $\sim 10 - 50 M_{\odot}$. At the end of their lifetime, such star could collapse as supernovae. If F is the energy flux received at the telescope and E is the energy radiated by the star, knowing the luminosity of such a star one can then use the ratio $\frac{F}{E}$ to determine their distances. This enables us to estimate the distances of the galaxies which the stars belong to.

All hot stars with mass $\gtrsim 15 M_{\odot}$ show a high velocity outflow[111]. Such an outflow is known as a stellar wind. Consequently, this leads to a mass loss from the star and the winds attaining a terminal velocity owing to Newton's first law of motion. Before proceeding further in the understanding of the driving mechanisms for stellar winds, one needs to be clear of these two important aspects.

5.3.1 Mass-Loss Rate

Due to the outflow of stellar material, the star suffers a mass loss, which is quantified by its *mass-loss rate* $\dot{M} = \frac{dM}{dt}$. Typical mass-loss rates of hot stars lie in the range $10^{-7} - 10^{-4} M_{\odot}/\text{yr}$, or approximately $\frac{1}{30} - 30 M_{\text{earth}}/\text{yr}$. In comparison, the sun also has a mass outflow, called *solar wind*, with $\dot{M} \approx 10^{-14} M_{\odot}/\text{yr}$.

5.3.2 Terminal Velocity

When stellar material is far from the surface of the star, it reaches its maximum velocity owing to the acceleration from the radiation pressure. At such large distances, due to the absence of any other external forces, the stellar velocity remains constant as per Newton's first law of motion. Therefore, this constant maximum velocity is known as the terminal velocity v_{∞} .

Typical values for v_{∞} lie between ~ 200 km/s (for A-type supergiants) and ~ 3000 km/s (for early O-type stars). Consequently, the speed of sounds in such atmospheres varies between

5.4. Relevance of Stellar Winds

10 km/s to 30 km/s as per the relation:

$$V(R) = v_\infty \left(1 - b \frac{R_*}{R}\right)^\beta$$

5.4 Relevance of Stellar Winds

The analysis of stellar winds gains importance due to the reasons outlined below:

- The lifetime of hot stars is approximately 10^{-3} that of the sun, i.e. $\sim 10^7$ yr. Assuming $\dot{M} \sim 10^{-6} M_\odot/\text{yr}$, the mass loss due to stellar winds is $\sim 10^7 \text{ yr} \times 10^{-6} M_\odot/\text{yr} \approx 10 M_\odot$ during its entire lifetime. This is a significant mass loss, e.g. for a star with an initial mass of $20 M_\odot$, half of its mass is lost due to stellar winds. Therefore, in order to have an accurate picture of the *evolution of hot stars*, it is essential to know its mass loss rate, and hence the stellar wind.
- In the interior of stellar cores, for most part of the star's lifetime energy is generated by the CNO bi-cycle. Therefore, the abundance ratio of the elements H : He : C : N : O is altered during the stellar lifetime, as compared to that during its birth. Furthermore, the nuclear processed matter is transported to the outer regions of the star by diffusion and convection, to be eventually carried to the surrounding space by the stellar wind. Because the hot stars are almost always clumped together in groups, these processes may significantly impact the chemical composition of the ISM, thereby affecting the *galactic evolution*.
- Far away from hot stars, where the stellar wind attains supersonic terminal velocities, it may collide with the surrounding material, leading to the development of shocks, which may lead to local compressions in the surrounding material. This may trigger the formation of proto-stars, leading to the birth of new stars. Therefore, the *formation of new stars* could be affected by the presence of stellar winds due to hot stars.

5.5 Radiative Line-Driving: Mechanism of Stellar Winds

The physical mechanism which initiates and drives the outflow of matter from hot stars is known as radiative line driving. Given below is a qualitative description of the physical mechanism behind this process.

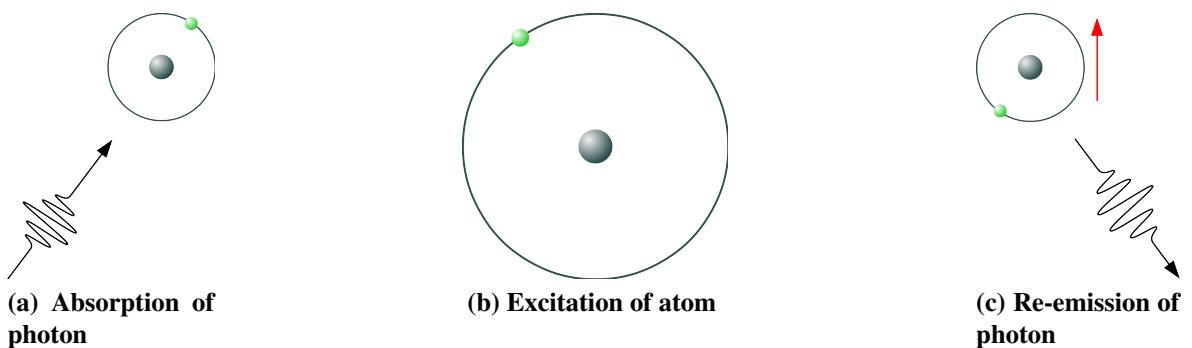


Figure 5.7: Principle of radiative line driving

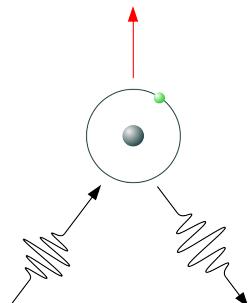


Figure 5.8: Momentum transfer during radiative line driving

Figure 5.7 depicts the interaction between the wind material (comprising of ions) and the photons emitted from the deeper layers of the stellar atmosphere, known as the photosphere. The transfer of momentum from the photons to the ions takes place in three steps:

- i. The absorption of a photon by an ion
- ii. The excitation of the ion
- iii. The re-emission of a photon by the ion

CHAPTER 6

RESULTS OF SPECTRAL FITTING AND VARIABILITY STUDIES

Contents

6.1	Source list	117
6.1.1	Journal of observations	118
6.1.2	Supersoft X-ray count rates	119
6.2	Spectral fitting	121
6.2.1	Best-fit statistics	121
6.2.2	Stellar parameters	121
6.2.3	Unfolded spectra from best-fit models	122
6.2.4	Presence of elemental absorption edges	126
6.3	Variability study of RX J0925.7-4758	130
6.3.1	Observed NICER and XMM-Newton lightcurves	130
6.3.2	Timing analysis of NICER lightcurves	131
6.3.3	Timing analysis of XMM-Newton lightcurves	136
6.3.4	Summary of timing analysis of RX J0925.7-4758	142

Abstract of chapter 6

This chapter explores a dataset of Supersoft X-ray Sources (SSS), analyzing their properties and spectral features. We begin by identifying the SSS and plotting their sky positions. An observation journal details the telescopes, instruments, and exposures used to collect data for each source. We then examine the observed count rates of the SSS, both individually and collectively, including a normalized view for better comparison. The best-fit model using the continuum NLTE approach is applied to all SSS, and the resulting fit statistics are presented. Stellar parameters like luminosity, effective temperature, and hydrogen column density are derived for each SSS using this model. We further explore the unfolded spectra obtained after applying the best-fit model, which reveal the intrinsic properties of the SSS after accounting for instrumental effects and absorption. Additionally, the presence of elemental absorption edges within the unfolded spectra is investigated, providing insights into the composition of the material surrounding these SSS objects. Notably, such edges were only found in SSS located in the Milky Way, not the Large Magellanic Cloud. Finally, we perform a detailed timing analysis of the SSS RX J0925.7-4758 using lightcurves from the NICER and XMM-Newton observatories. The Lomb-Scargle periodograms are used to identify periodic signals, with notable peaks observed at 0.055 mHz and 0.077 mHz, and their significance is discussed. Sinusoidal fits to the phase-folded lightcurves confirm the presence of periodic modulations, with variability characteristics reported across different energy ranges.

6.1. Source list

6.1 Source list

The following table 6.1 lists all the SSS included in the dataset studied in the present work. It provides information on the source name, SSS type, the host galaxy – either Milky Way (MW) or Large Magellanic Cloud (LMC), right ascension α , and declination δ in degrees.

Table 6.1: List of sources in dataset

S. No.	Source	Type	Galaxy	α (degrees)	δ (degrees)
1	CAL 83	XRB	LMC	85.89233	-68.37283
2	RS Oph	SS	MW	267.55483	-6.70791
3	RX J0019.8+2156	CV	MW	4.95802	21.94783
4	RX J0527.8-6954	CV	LMC	81.95345	-69.90247
5	RX J0925.7-4758	XRB	MW	141.44167	-47.97148
6	V3890 Sgr	CN	MW	277.68037	-24.01915

The SSS mentioned in table 6.1 are plotted at their respective sky positions using Aladin Sky Atlas¹ and presented using figure 6.1.

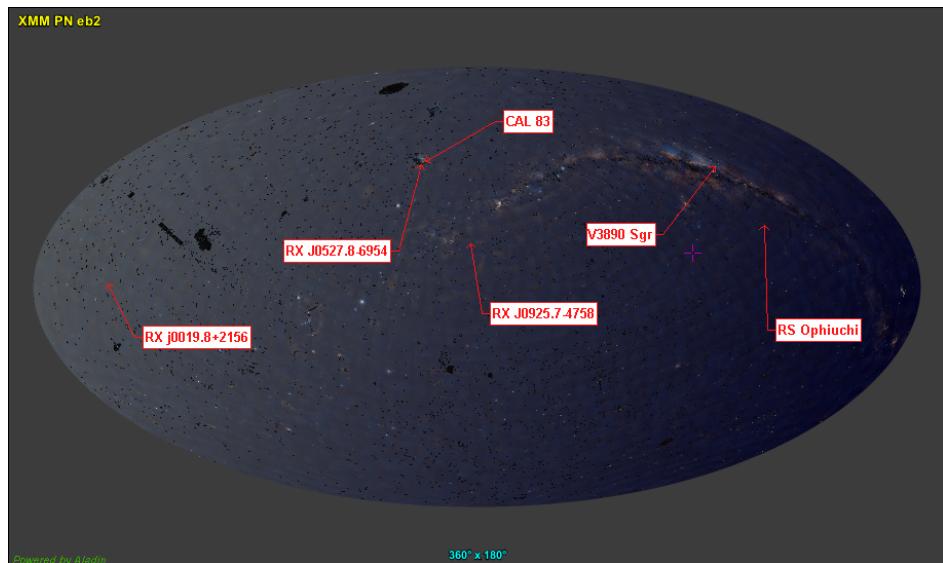


Figure 6.1: Sky positions of SSS chosen for the dataset

¹<https://aladin.cds.unistra.fr/>

6.1.1 Journal of observations

The table 6.2 presents a journal of the SSS observations for each source in the dataset. It includes the source name, the name of the observatory, the instrument used for the observation, the observation ID, the date of observation, the Modified Julian Day (in MJD), exposure time (in kiloseconds), and the observed photon energy range (in keV).

Table 6.2: SSS observation journal of dataset

Source	Observatory	Instrument	Observation (Obs. ID)	Date (yyy-mm-dd)	MJD	Exposure (ks)	Region (keV)
CAL 83	XMM-Newton	EPIC-pn	0506531501	2008-08-12	54690.62	6.9	0.2-1.0
RS Oph	XMM-Newton	EPIC-pn	0410180501	2006-10-09	54017.98	49.2	0.2-2.0
RX J0019.8+2156	XMM-Newton	EPIC-pn	0047940101	2001-12-31	52274.77	58.1	0.2-0.5
RX J0527.8-6954	XMM-Newton	EPIC-pn	0086770101	2000-12-19	51897.70	49.3	0.2-0.6
RX J0925.7-4758	XMM-Newton	EPIC-pn	0111150101	2000-12-16	51894.46	61.1	0.3-1.0
V3890 Sgr	XMM-Newton	EPIC-MOS1	0821560201	2019-09-14	58740.98	25.0	0.1-2.0

6.1. Source list

6.1.2 Supersoft X-ray count rates

The observed count rates of the sources listed in table 6.2 are presented in separate sub-figures within figure 6.2. Each sub-figure shows the count rate for a specific source across the corresponding energy range specified in table 6.2.

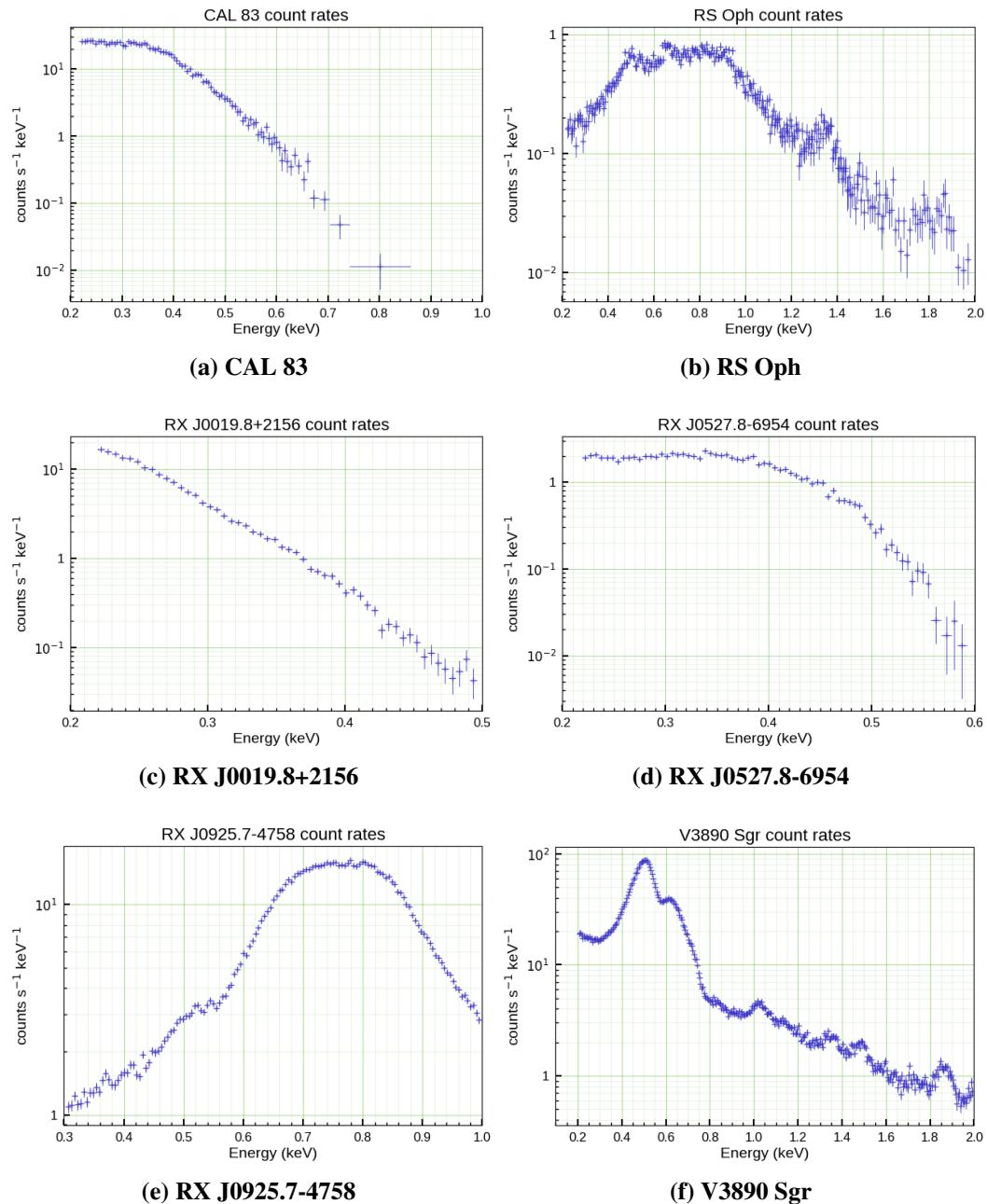


Figure 6.2: Count rates of individual sources in dataset in SSS regime

6.1. Source list

Figure 6.3 presents the observed count rates of all sources in the dataset.

- The sub-figure 6.3a presents the raw count rates for each source across the observed energy range, highlighting potential differences in their intrinsic luminosities and/or absorption properties.
- For a more direct comparison between the source spectra that may have intrinsically different luminosities, the sub-figure 6.3a presents the count rates scaled to a range of 0 to 1 using min-max normalization.

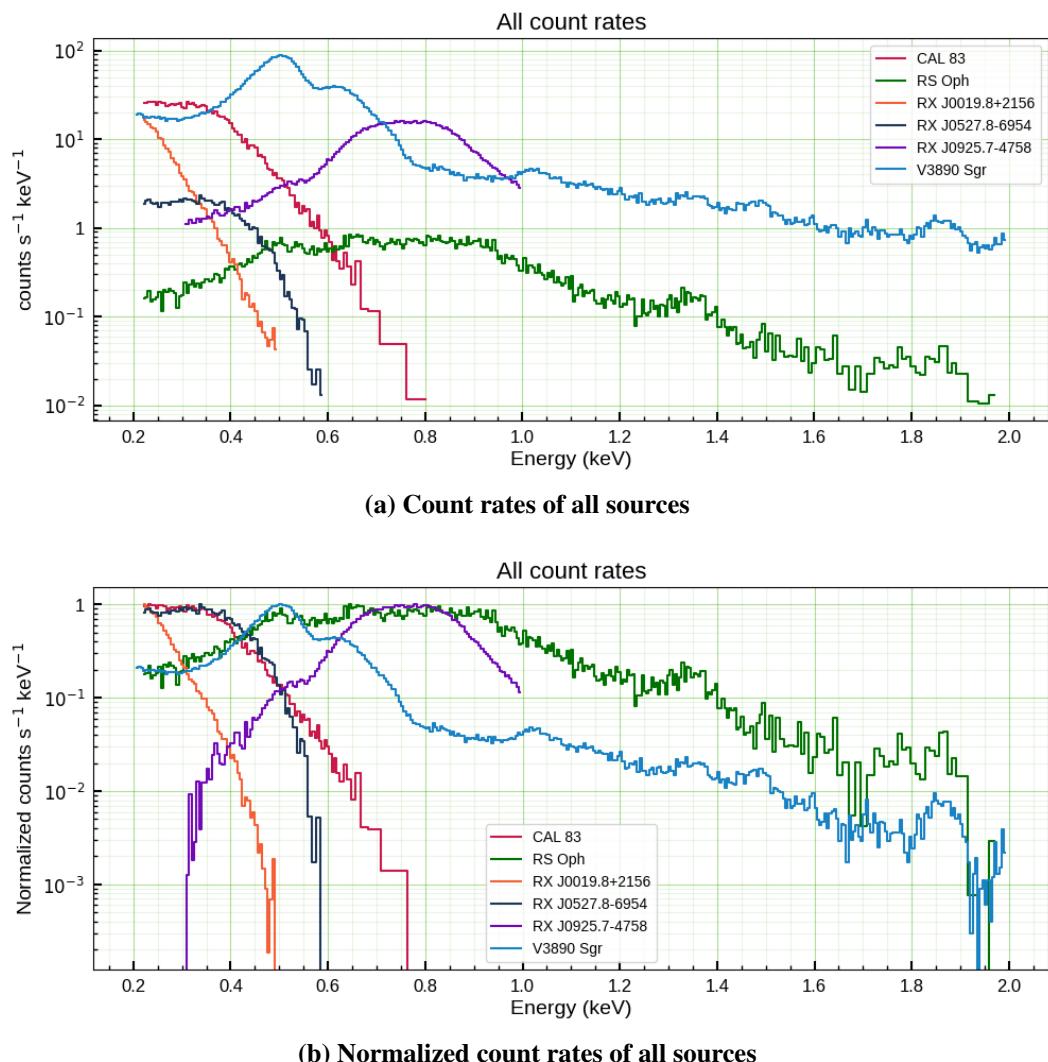


Figure 6.3: Count rates of all sources in dataset

6.2 Spectral fitting

6.2.1 Best-fit statistics

Table 6.3 presents the fit statistics for the best-fit model applied to all sources in the dataset using the continuum NLTE model. The table includes the chi-squared value (χ^2), degrees of freedom (d.o.f.), reduced chi-squared value (χ^2_{reduced}), and the null hypothesis probability.

Table 6.3: Fitting statistics for all sources in dataset

Source	$\chi^2/\text{d.o.f}$	χ^2_{reduced}	Null hyp. prob.
CAL 83	108.0/83	1.30	1.50×10^{-2}
RS Oph	189.7/135	1.40	1.35×10^{-3}
RX J0019.8+2156	68.3/54	1.26	9.16×10^{-2}
RX J0527.8-6954	135.1/87	1.55	7.28×10^{-4}
RX J0925.7-4758	183.4/131	1.40	1.73×10^{-3}
V3890 Sgr			

6.2.2 Stellar parameters

The stellar parameters for all sources in the dataset, derived using the continuum NLTE model, are presented in table 6.4. The table provides the values of the luminosity (L_*), effective temperature (T_{eff}), and hydrogen column density (n_H) for each source, calculated after obtaining the best-fit.

In figure 6.4, the calculated values of the luminosities L_* and effective temperature T_{eff} for the SSS in our dataset using their respective best-fit models, as presented in table 6.4.

6.2. Spectral fitting

Table 6.4: Stellar parameters of all sources from continuum NLTE model

Source	L_* (erg s $^{-1}$)	T_{eff} (kK)	n_H ($\times 10^{22}$ cm $^{-2}$)
CAL 83	$5.86_{4.54}^{7.20} \times 10^{38}$	$102.9_{100.8}^{110.0}$	$0.087_{0.079}^{0.093}$
RS Oph	$2.67_{2.35}^{8.64} \times 10^{35}$	$181.6_{167.4}^{200.1}$	$0.067_{0.018}^{0.170}$
RX J0019.8+2156	$1.23_{0.39}^{2.37} \times 10^{35}$	$60.5_{46.5}^{70.9}$	$0.046_{0.014}^{0.805}$
RX J0527.8-6954	$1.21_{0.60}^{2.13} \times 10^{40}$	$50.6_{43.7}^{54.1}$	$0.333_{0.246}^{0.338}$
RX J0925.7-4758	$1.69_{0.54}^{3.58} \times 10^{44}$	$91.6_{86.9}^{100.2}$	$1.175_{1.086}^{1.274}$
V3890 Sgr			

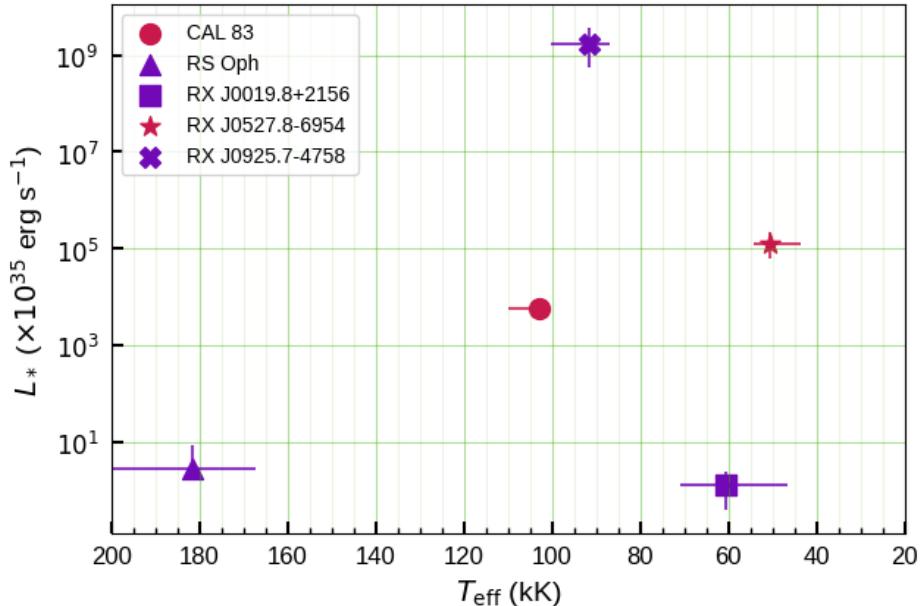


Figure 6.4: L_* and T_{eff} computed for all SSS in dataset

6.2.3 Unfolded spectra from best-fit models

Unfolded spectra represent the intrinsic spectral energy distribution (SED) of an astronomical object after removing the effects of instrumental response and absorption along the line of sight. The best-fit model accounts for these factors, allowing us to recover the underlying source spectrum. By analyzing the unfolded spectrum, we can identify prominent features such as emission lines and absorption edges. Also, the overall shape of the unfolded spec-

6.2. Spectral fitting

trum in the continuum region can provide insights into the physical processes responsible for the emission mechanism within the source.

In this section, we present the unfolded spectra obtained after applying the best-fit model derived using the continuum NLTE approach.

Unfolded spectrum of CAL 83

Figure 6.5 presents the unfolded spectrum of the SSS CAL 83 in the LMC.

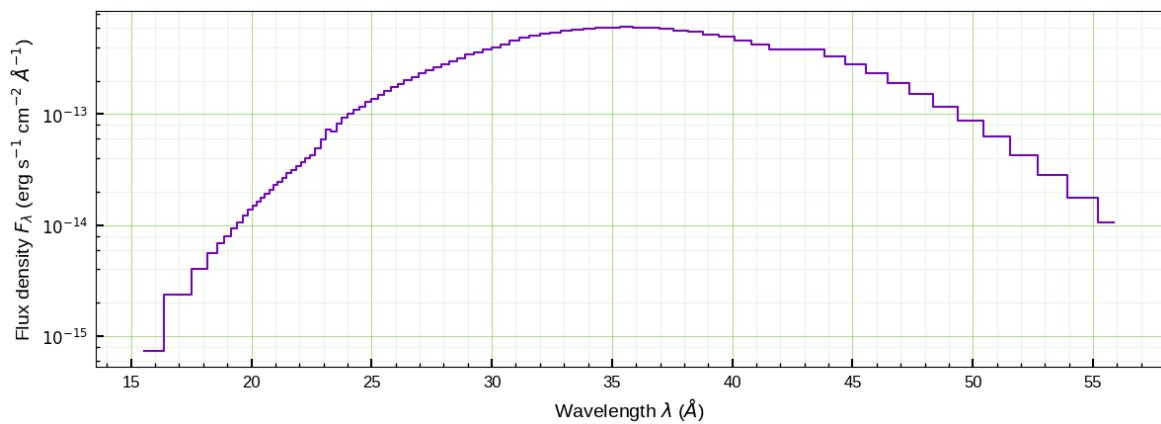


Figure 6.5: Unfolded spectrum after model fitting for CAL 83

6.2. Spectral fitting

Unfolded spectrum of RS Oph

Figure 6.6 presents the unfolded spectrum of the SSS RS Oph in the MW.

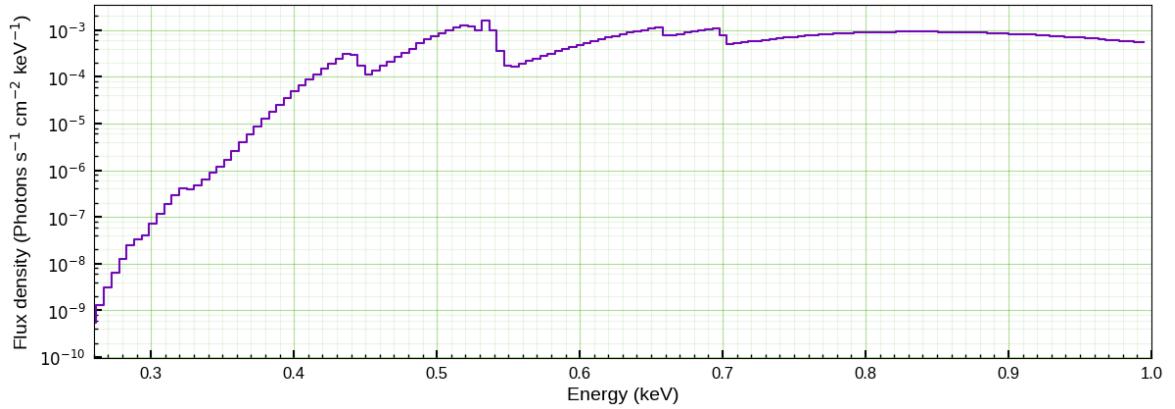


Figure 6.6: Unfolded spectrum after model fitting RS Oph

Unfolded spectrum of RX J0019.8+2156

Figure 6.7 presents the unfolded spectrum of the SSS RX J0019.8+2156 in the MW.

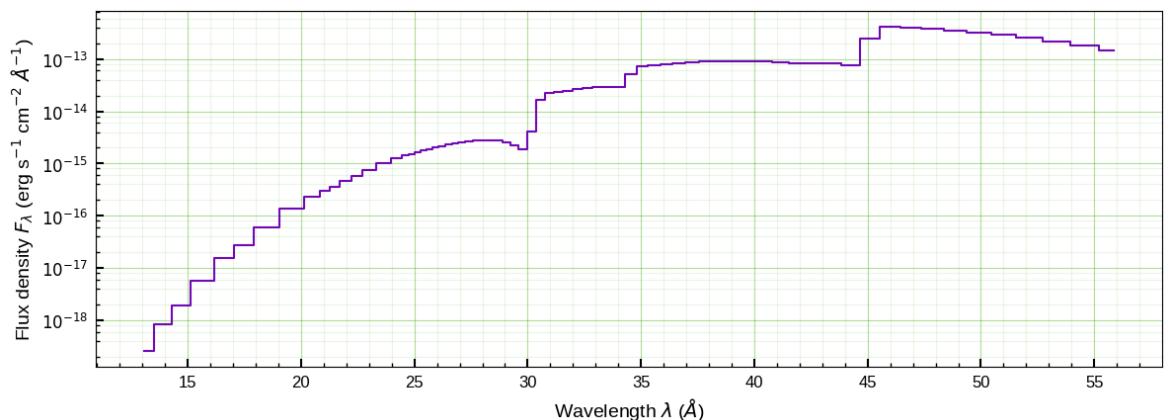


Figure 6.7: Unfolded spectrum after model fitting for RX J0019.8+2156

6.2. Spectral fitting

Unfolded spectrum of RX J0527.8-6954

Figure 6.8 presents the unfolded spectrum of the SSS RX J0527.8-6954 in the LMC.

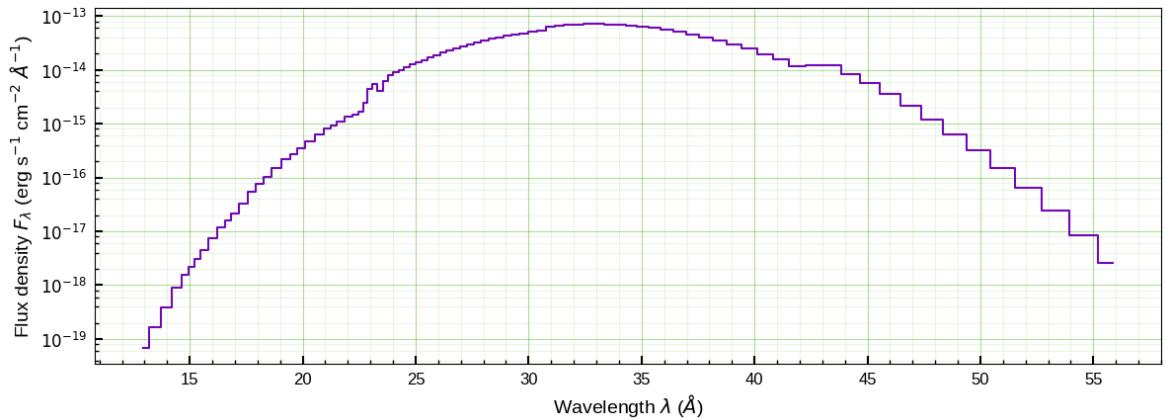


Figure 6.8: Unfolded spectrum after model fitting for RX J0527.8-6954

Unfolded spectrum of RX J0925.7-4758

Figure 6.9 presents the unfolded spectrum of the SSS RX J0925.7-4758 in the MW.

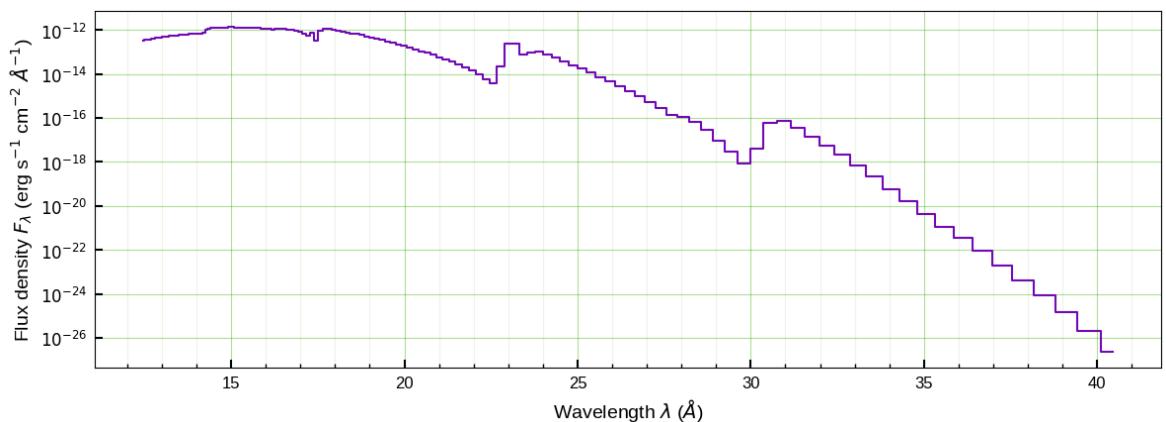


Figure 6.9: Unfolded spectrum after model fitting for RX J0925.7-4758

6.2.4 Presence of elemental absorption edges

An absorption edge is a spectral feature which appears as a sharp decrease in intensity at a specific energy in the spectrum. It resembles a step function, with a sudden drop followed by a plateau at a lower intensity level. This occurs when the energy of a photon exactly matches the ionization potential required to remove an electron from a specific energy level (shell) in an atom. The sudden drop happens because a majority of photons at this energy are absorbed. So, while an absorption line represents a *bound-bound transition*, an absorption edge represents a *bound-free transition*.

The presence of various absorption edges provides information about the elemental composition of the absorbing material. The energy at which the edge appears is characteristic of the element responsible for the absorption. In this section, we present the elemental absorption edges identified within the unfolded spectra of specific sources in the dataset. It is noteworthy that identifiable absorption edges were only found within the unfolded spectra of SSS objects located in the Milky Way (MW). No such edges were discernible in the spectra of SSS objects residing in the Large Magellanic Cloud (LMC).

6.2. Spectral fitting

Absorption edges in RS Oph spectrum

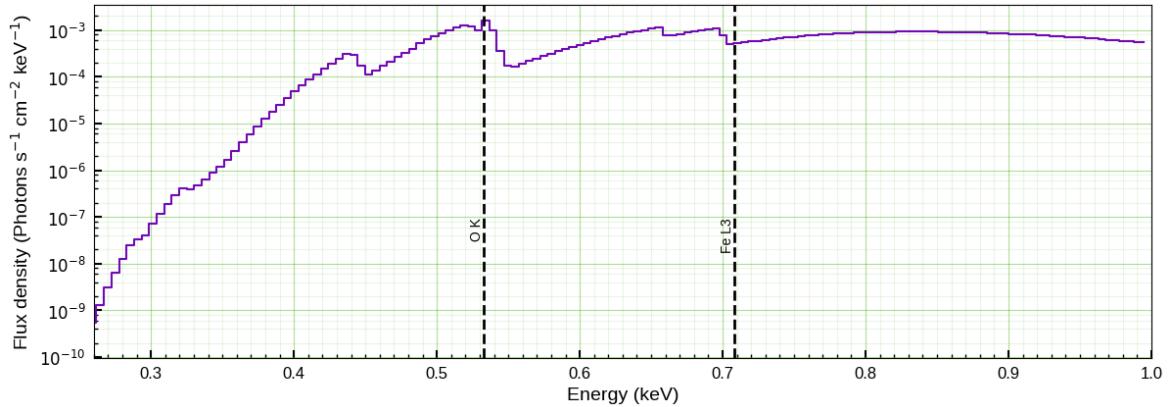


Figure 6.10: Identified absorption edges for RS Oph

Table 6.5: Absorption edges identified for RS Oph

Source	RS Oph
Observatory	XMM-Newton
Obs. ID	0410180501
Instrument	EPIC-pn
No. of edges identified	2
Relative depth	0.52 : 1.0
Absorption edge	O K Edge Fe L ₃ Edge
Edge energy	0.533 keV 0.708 keV
Edge depth	0.457 0.872

6.2. Spectral fitting

Absorption edges in RX J0019.8+2156 spectrum

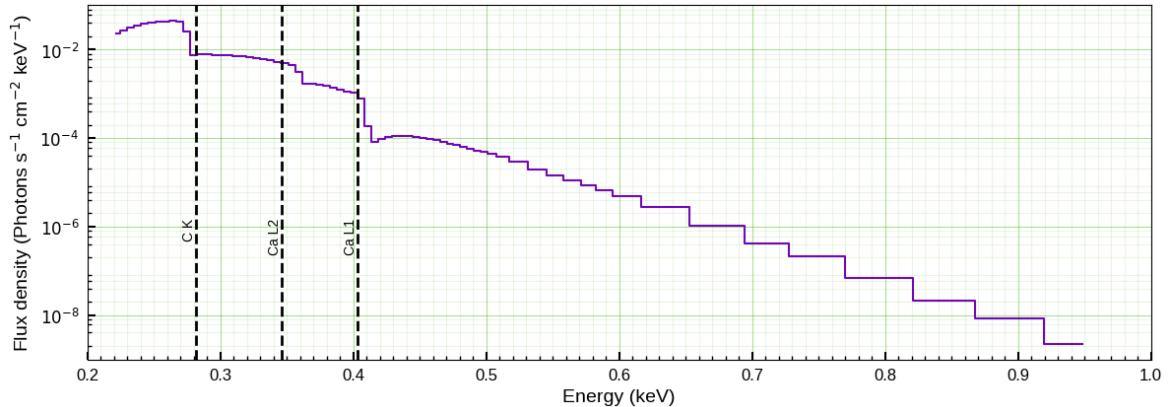


Figure 6.11: Identified absorption edges for RX J0019.8+2156

Table 6.6: Absorption edges identified for RX J0019.8+2156

Source	RX J0019.8+2156		
Observatory	XMM-Newton		
Obs. ID	0047940101		
Instrument	EPIC-pn		
No. of edges identified	3		
Relative depth	0.93 : 0.03 : 1.0		
Absorption edge	C K Edge	Ca L ₂ Edge	Ca L ₁ Edge
Edge energy	0.282 keV	0.346 keV	0.403 keV
Edge depth	0.809	0.026	0.867

6.2. Spectral fitting

Absorption edges in RX J0925.7-4758 spectrum

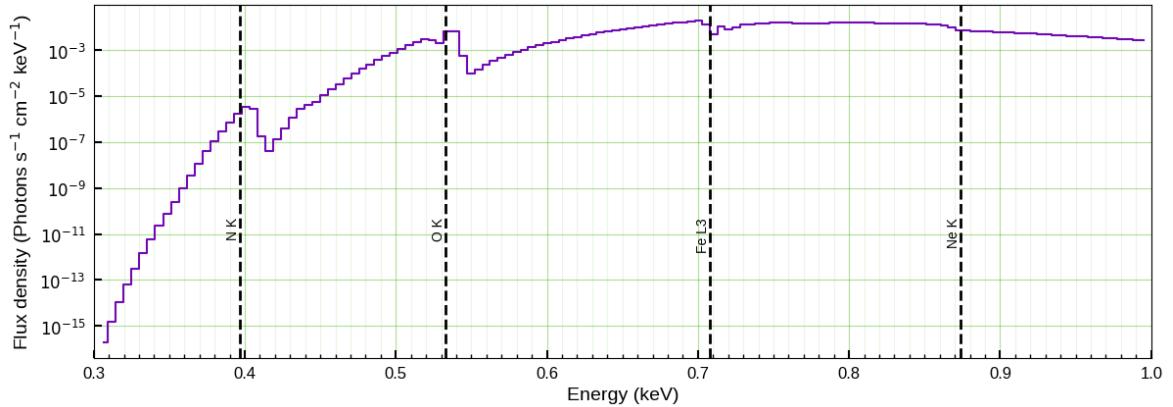


Figure 6.12: Identified absorption edges from RX J0925.7-4758 spectrum

Table 6.7: Absorption edges identified for RX J0925.7-4758

Source	RX J0925.7-4758
Observatory	XMM-Newton
Obs. ID	0111150101
Instrument	EPIC-pn
No. of edges identified	4
Relative depth	1.0 : 0.03 : 0.01 : 0.04
Absorption edge	N K Edge O K Edge Fe L ₃ Edge Ne K Edge
Edge energy	0.397 keV 0.533 keV 0.708 keV 0.874 keV
Edge depth	5.215 0.132 0.033 0.211

6.3 Variability study of RX J0925.7-4758

In this section, we present the results obtained from a detailed timing analysis of the NICER lightcurves for the source RX J0925.7-4758. The analysis was conducted using the *Lomb-Scargle periodogram*, a well-established method for identifying periodic signals in unevenly sampled time-series data.

6.3.1 Observed NICER and XMM-Newton lightcurves

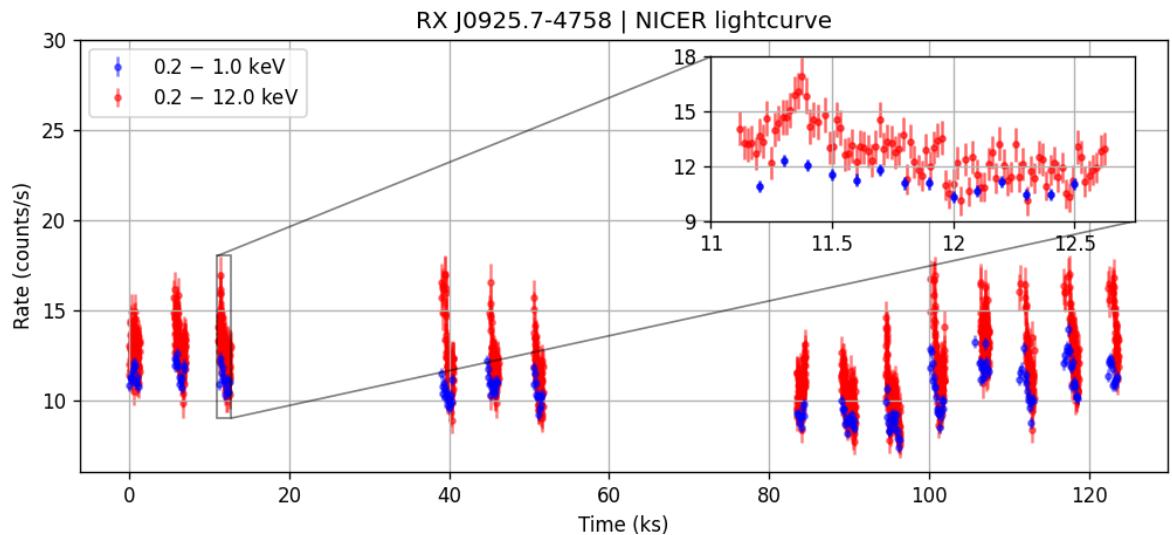


Figure 6.13: NICER lightcurves of RX J0925.7-4758 using 16 s time bins

The lightcurves of the SSS RX J0925.7-4758 as observed by the observatories NICER and XMM-Newton are presented here. For the NICER observations, the lightcurves are extracted by combining timing data from all three observations mentioned in table 3.1 using time bins of 16 s, and for XMM-Newton observations, the lightcurves are extracted from the timing data in observation ID 0111150101 using time bins of 50 s. Figures 6.13 and 6.14 respectively display the lightcurves extracted from NICER and XMM-Newton data over 0.2–1.0 keV and 0.2–12.0 keV photon energy ranges (the former is the range analysed in the multi-observatory spectral analysis of RX J0925.7-4758 described in chapter 3, whereas the latter is the full range for SSS).

As expected, the count rates for the lightcurve extracted over the full SSS photon energy range is higher than that for the 0.2–1.0 keV range. This can be seen in the insets in both

6.3. Variability study of RX J0925.7-4758

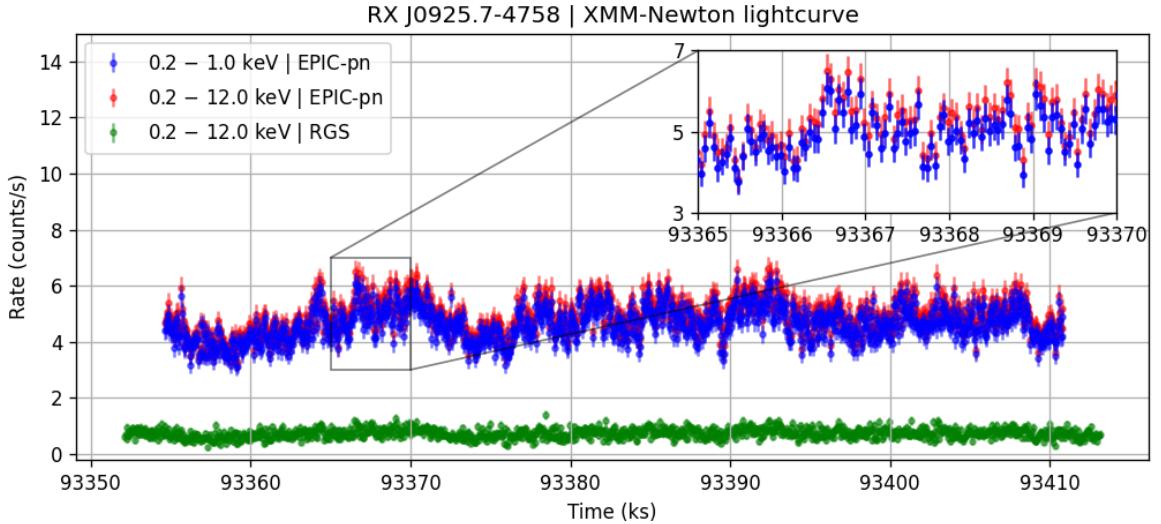


Figure 6.14: XMM-Newton lightcurves of RX J0925.7-4758 using 50 s time bins

figures 6.13 and 6.14 for the chosen sub-intervals of observations. In figure 6.14, we can also see that the count rates of the RGS observations are much smaller than the corresponding EPIC-pn observations.

6.3.2 Timing analysis of NICER lightcurves

We present here the results of the timing analysis by applying the Lomb-Scargle periodogram on the NICER lightcurves of RX J0925.7-4758. The use of this technique allows us to extract reliable periodic information from the NICER observations, overcoming potential issues related to gaps in data acquisition or variable observational cadence.

The lightcurves were examined in two distinct photon energy ranges: 0.2–1.0 keV and 0.2–12.0 keV. The former range aligns with the energy band used in the earlier spectral analysis of the source, providing a direct comparison of the timing characteristics within the same spectral window. The latter range encompasses the full energy band for SSS, thereby offering a comprehensive view of the timing behaviour across the entire spectrum accessible to NICER.

6.3. Variability study of RX J0925.7-4758

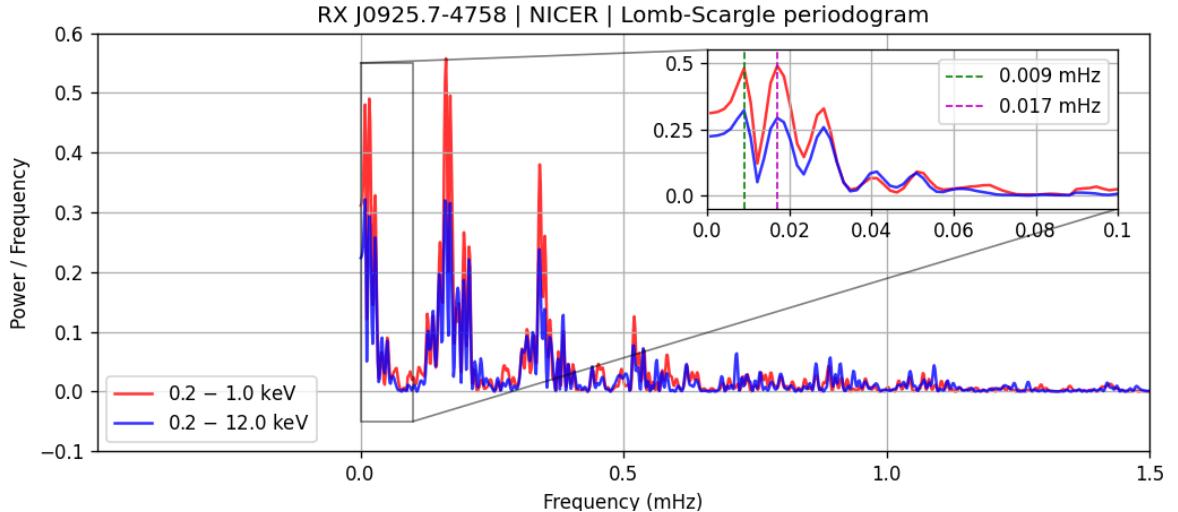


Figure 6.15: Lomb-Scargle periodogram of NICER lightcurves

6.3.2.1 Lomb-Scargle periodograms

The Lomb-Scargle periodograms were calculated for the lightcurves across both photon energy ranges. The resulting periodograms are depicted in figure 6.15. Upon examination of the inset, we observe closely spaced peaks at frequencies of 0.009 mHz and 0.017 mHz, which appear consistently in both energy ranges. These frequencies correspond to periods of 112.3 ks and 58.8 ks respectively, with the latter possibly being a harmonic. So we may consider 0.009 mHz to be an intrinsic variation in the source signal.

Additionally, peaks are identified at higher frequencies, specifically at 0.163 mHz, 0.341 mHz, and 0.521 mHz. However, it is important to note that these higher frequency peaks are attributed to the observational cadence of the NICER XTI, i.e. they correspond to the NICER observation windows, which are separated by approximately 6144 seconds, and represent harmonics of this fundamental period. These peaks do not necessarily indicate intrinsic periodicity in the source but rather reflect the instrumental observational pattern.

6.3.2.2 Best-fit sinusoids for 0.2–1.0 keV lightcurve

Fitting with phase-folded lightcurve: Utilising the frequencies corresponding to the peak power densities at 0.009 mHz and 0.017 mHz, we performed sinusoidal fits to the NICER lightcurve data. The resulting phase-folded lightcurve, spanning a single cycle from 0 to 2π radians, is presented in figure 6.16 for the photon energy range of 0.2–1.0 keV. This phase-

6.3. Variability study of RX J0925.7-4758

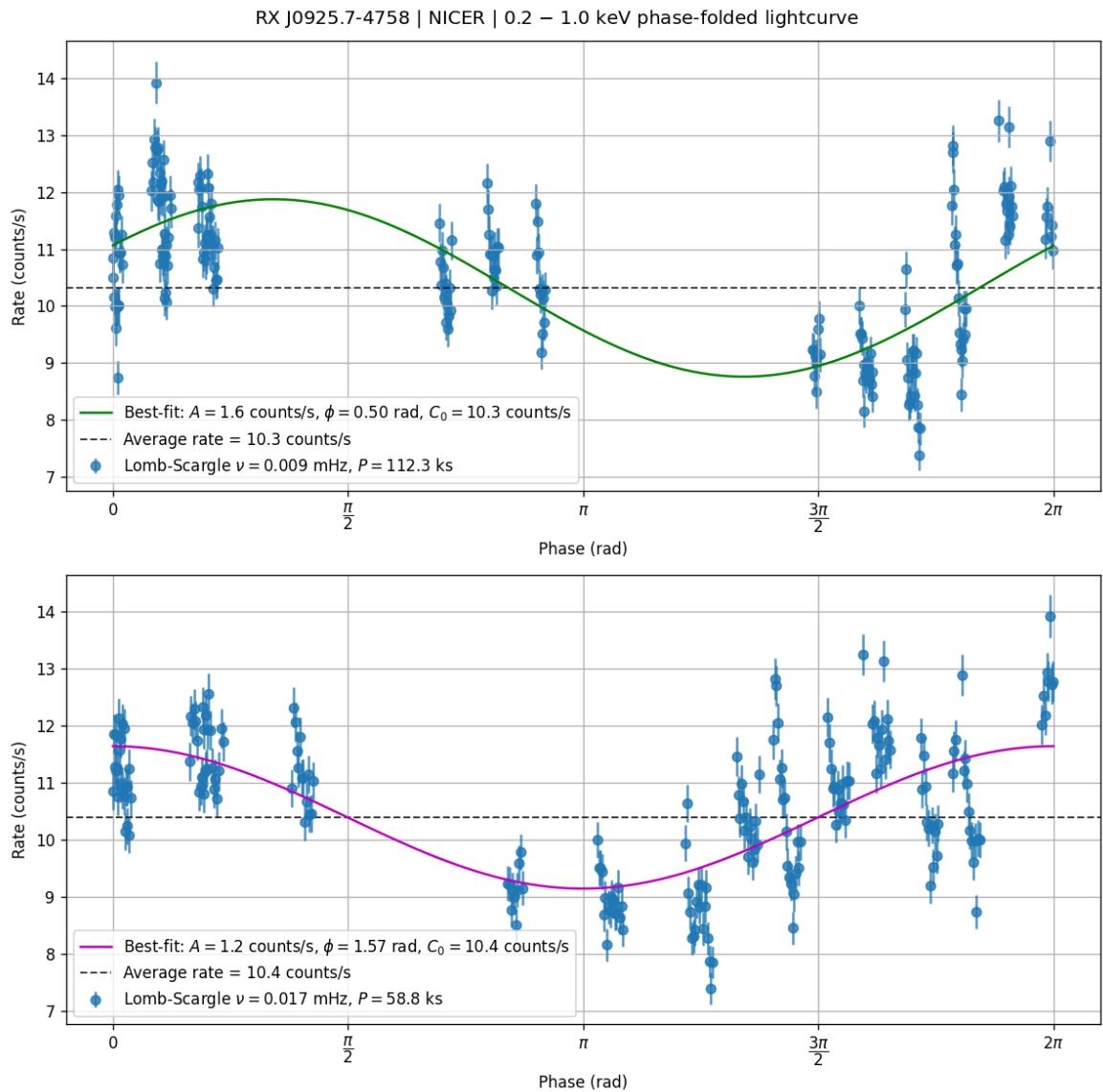


Figure 6.16: Phase-folded NICER lightcurve over 0.2–1.0 keV with best-fit sinusoids

folded representation provides a clearer view of the periodic variations in the lightcurve and highlights the sinusoidal nature of the observed modulation within this specific energy band. Additionally, the best-fit sinusoids are overlaid on top of the phase-folded data, illustrating the close agreement between the model and the observed lightcurve, thereby affirming the periodicity at these frequencies.

6.3. Variability study of RX J0925.7-4758

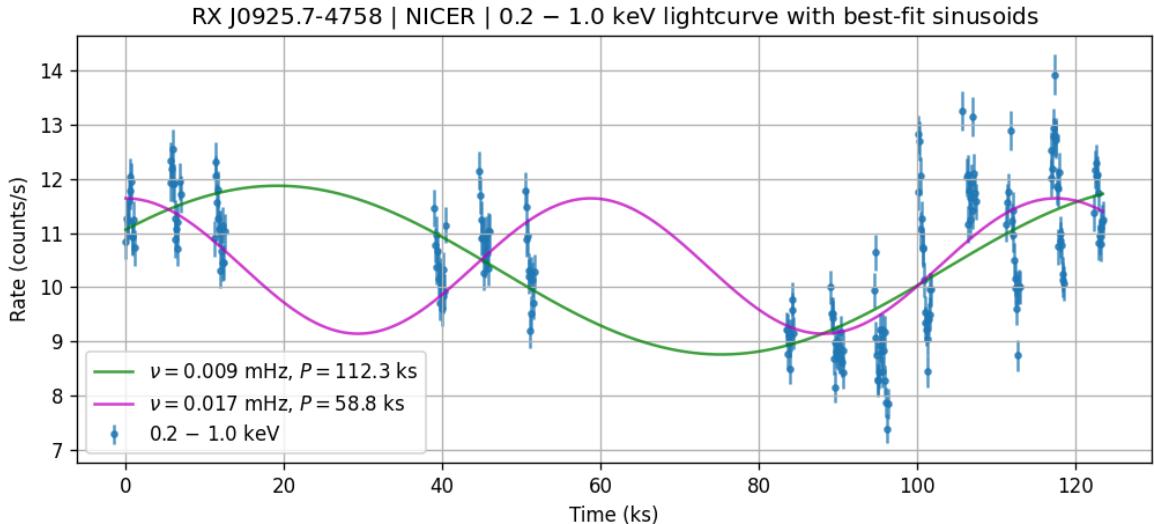


Figure 6.17: NICER lightcurve over 0.2–1.0 keV with best-fit sinusoids

Fitting with complete lightcurve: To further illustrate the periodic nature of the emission from RX J0925.7-4758, the best-fit sinusoids have also been overlaid on the complete NICER lightcurve, as shown in figure 6.17. This figure provides a comprehensive view of the model’s fit over the entire observation period, rather than just a phase-folded representation. By overlaying the best-fit sinusoids directly on the unmodified lightcurve, we can visually assess the alignment of the observed data points with the fitted model across different time intervals.

The strong correspondence between the sinusoidal fits and the actual lightcurve further corroborates the presence of these low-frequency modulations and supports the reliability of the Lomb-Scargle periodogram results in characterising the periodic variability of RX J0925.7-4758.

6.3.2.3 Best-fit sinusoids for 0.2–12.0 keV lightcurve

Fitting with phase-folded lightcurve: For the broader 0.2–12.0 keV photon energy range, sinusoidal fits were also applied using the identified frequencies of 0.009 mHz and 0.017 mHz. The resulting phase-folded lightcurve, spanning one full cycle from 0 to 2π radians, is displayed in figure 6.18. The phase-folded data, with the best-fit sinusoids overlaid, demonstrate periodic modulations consistent with those observed in the narrower 0.2–1.0 keV range.

6.3. Variability study of RX J0925.7-4758

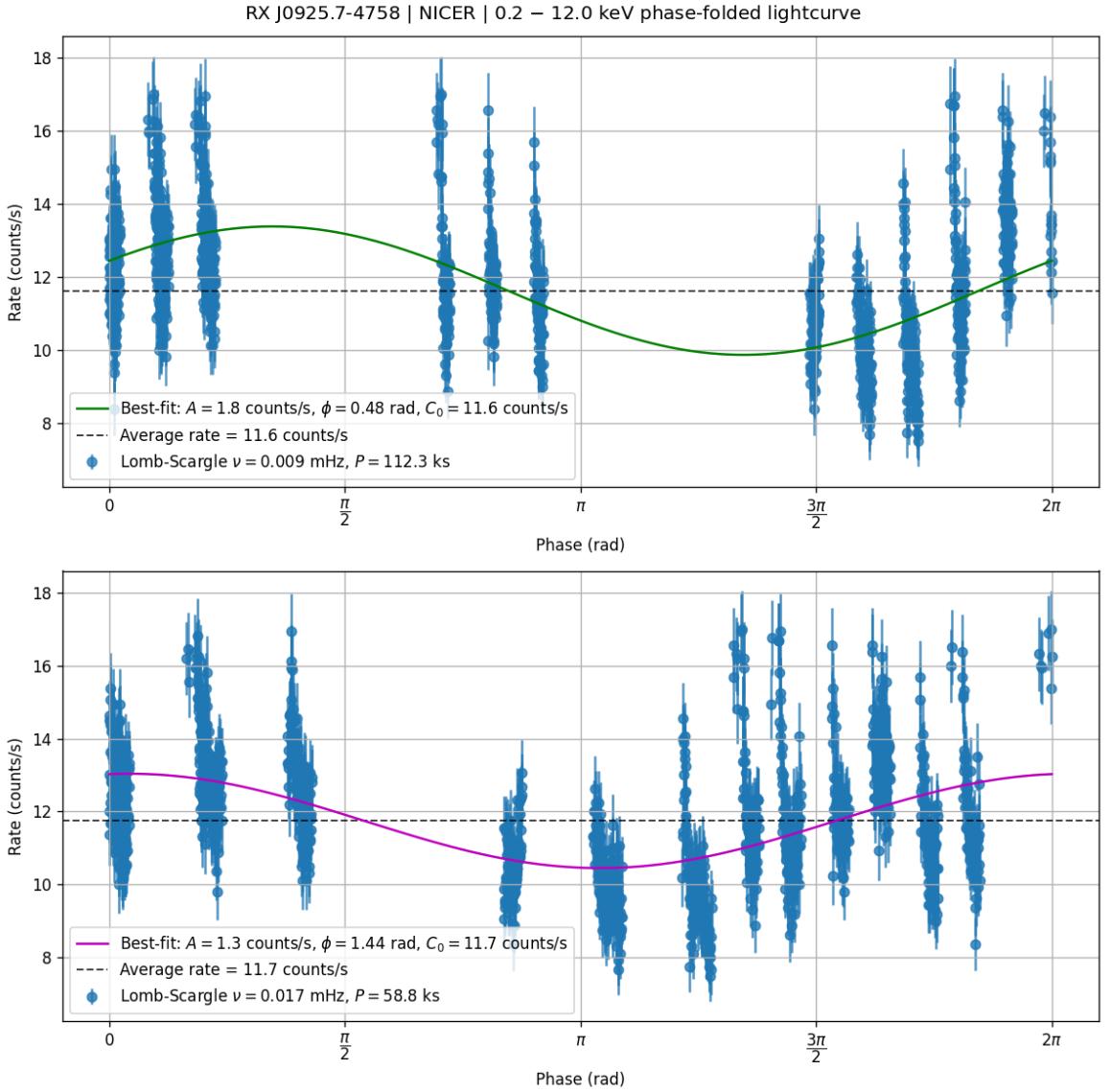


Figure 6.18: Phase-folded NICER lightcurve over 0.2–12.0 keV with best-fit sinusoids

Fitting with complete lightcurve: For the 0.2–12.0 keV photon energy range, the best-fit sinusoids were similarly overlaid on the complete NICER lightcurve, as depicted in figure 6.19. This visualisation provides an overall view of the model’s fit across the entire dataset, enabling a direct comparison of the modelled periodic variations with the observed lightcurve over extended periods. The alignment between the best-fit sinusoids and the actual data points consistently supports the presence of low-frequency modulations in this broader energy band.

6.3. Variability study of RX J0925.7-4758

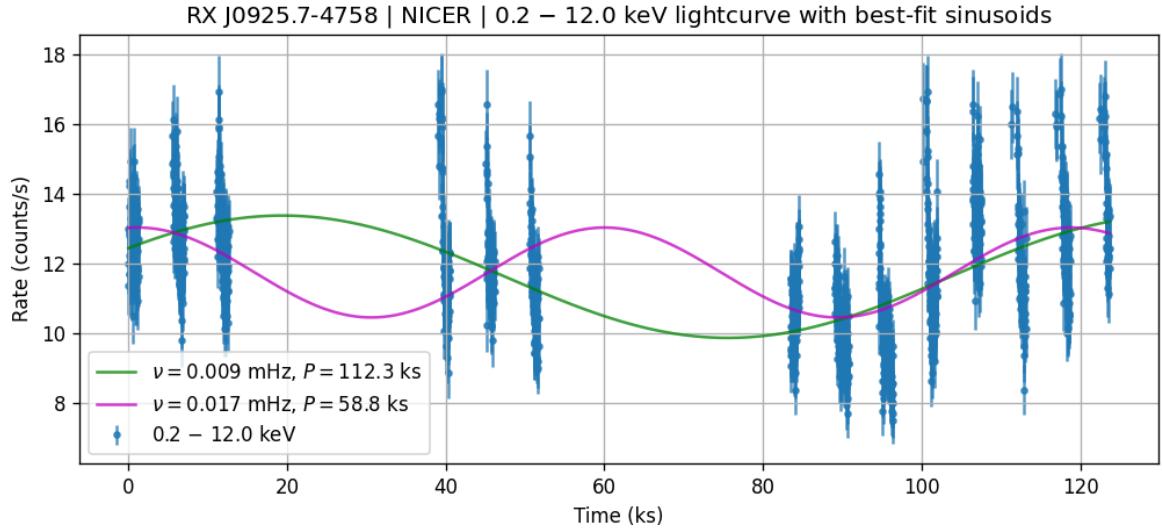


Figure 6.19: NICER lightcurve over 0.2–12.0 keV with best-fit sinusoids

6.3.3 Timing analysis of XMM-Newton lightcurves

In addition to the NICER observations, we conducted a timing analysis using lightcurves obtained from the EPIC-pn and RGS instruments onboard the XMM-Newton observatory. Similar to the NICER data, we applied the Lomb-Scargle periodogram to these lightcurves to detect and characterise periodic signals. The analysis was performed for the same photon energy ranges: 0.2–1.0 keV and 0.2–12.0 keV. Notably, the XMM-Newton observations were conducted over a continuous observation window of approximately 55 ks, reducing the impact of gaps in data acquisition and improving the reliability of the periodicity detection.

6.3.3.1 Lomb-Scargle periodograms

The Lomb-Scargle periodograms were also calculated for the lightcurves obtained from the EPIC-pn and RGS instruments of the XMM-Newton observatory, analysed across the same photon energy ranges of 0.2–1.0 keV and 0.2–12.0 keV. The resulting periodograms are depicted in figure 6.20. As it can be noticed in the inset, the periodograms reveal two prominent peaks at 0.055 mHz and 0.077 mHz, consistently appearing in both energy ranges and for both instruments. These frequencies correspond to periods of approximately 18.1 ks and 13.1 ks, respectively, indicating potential intrinsic periodicities in the source RX J0925.7-4758. The peaks observed with the RGS instrument are less pronounced, likely due to the lower flux levels in the RGS data compared to EPIC-pn.

6.3. Variability study of RX J0925.7-4758

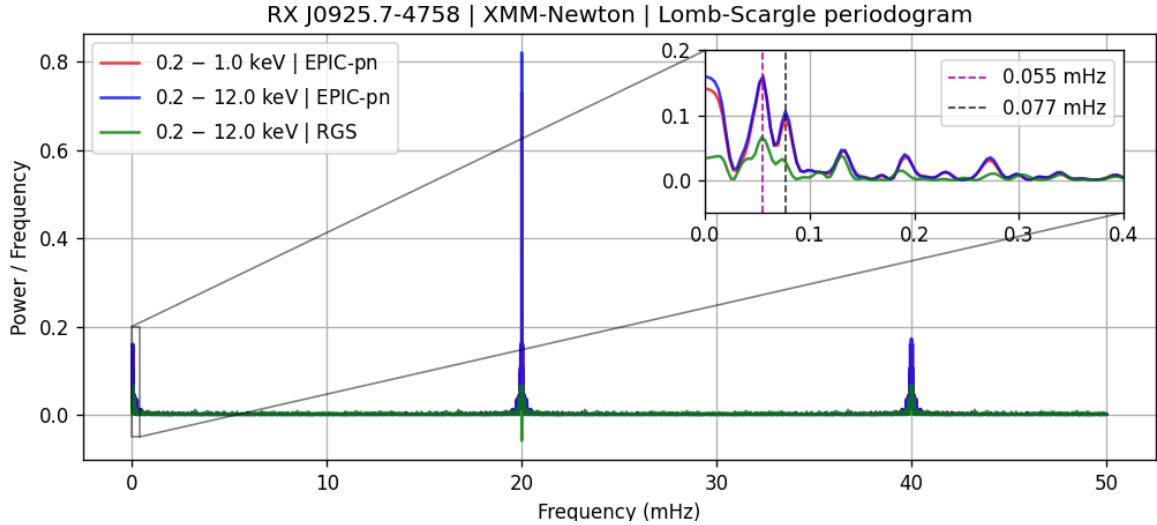


Figure 6.20: Lomb-Scargle periodogram of XMM-Newton lightcurves

In addition, higher frequency peaks are observed at 20 mHz and 40 mHz. However, these are attributed to harmonics introduced by the binning of the lightcurve data, with the 20 mHz peak corresponding to a time period of 50 seconds, which matches our chosen bin size. Therefore, these high-frequency peaks are artefacts of the temporal resolution rather than indicative of intrinsic variability in the source.

6.3.3.2 Best-fit sinusoids for 0.2–1.0 keV EPIC-pn lightcurve

Fitting with phase-folded lightcurve: For the 0.2–1.0 keV photon energy range obtained from the EPIC-pn instrument of the XMM-Newton observatory, sinusoidal fits were performed using the frequencies with peak power densities observed at 0.055 mHz and 0.077 mHz. The resulting phase-folded lightcurve, covering a single cycle from 0 to 2π radians, is shown in figure 6.21. The best-fit sinusoids are overlaid on the phase-folded data, demonstrating a strong correspondence between the model and the observed lightcurve.

6.3. Variability study of RX J0925.7-4758

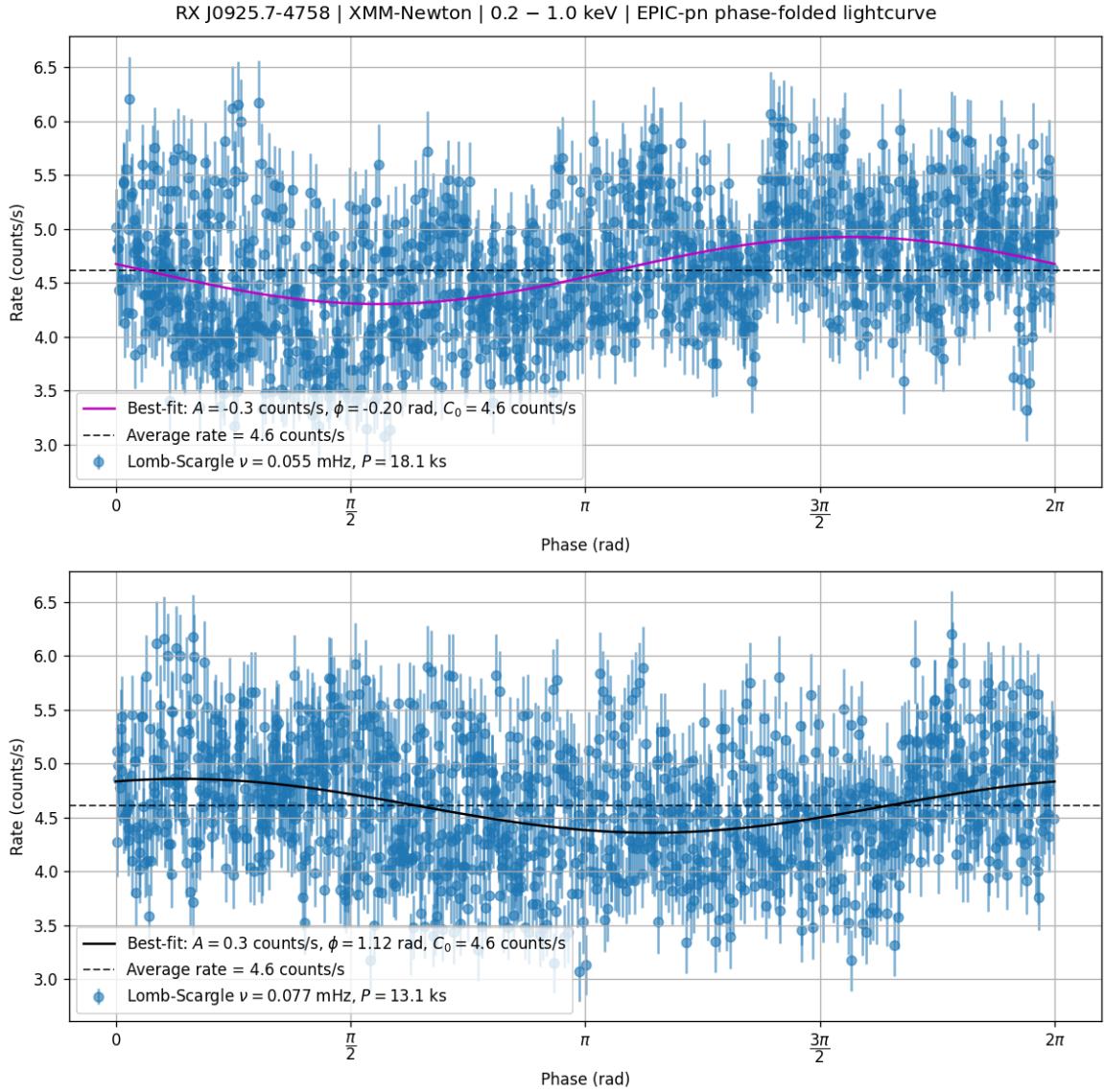


Figure 6.21: Phase-folded XMM-Newton EPIC-pn lightcurve over 0.2–1.0 keV with best-fit sinusoids

Fitting with complete lightcurve: The best-fit sinusoids have also been overlaid on the complete lightcurve for the 0.2–1.0 keV photon energy range using the EPIC-pn instrument of the XMM-Newton observatory. From figure 6.22, we can perform a visual assessment of the agreement between the observed data points with the fitted sinusoid models.

6.3. Variability study of RX J0925.7-4758

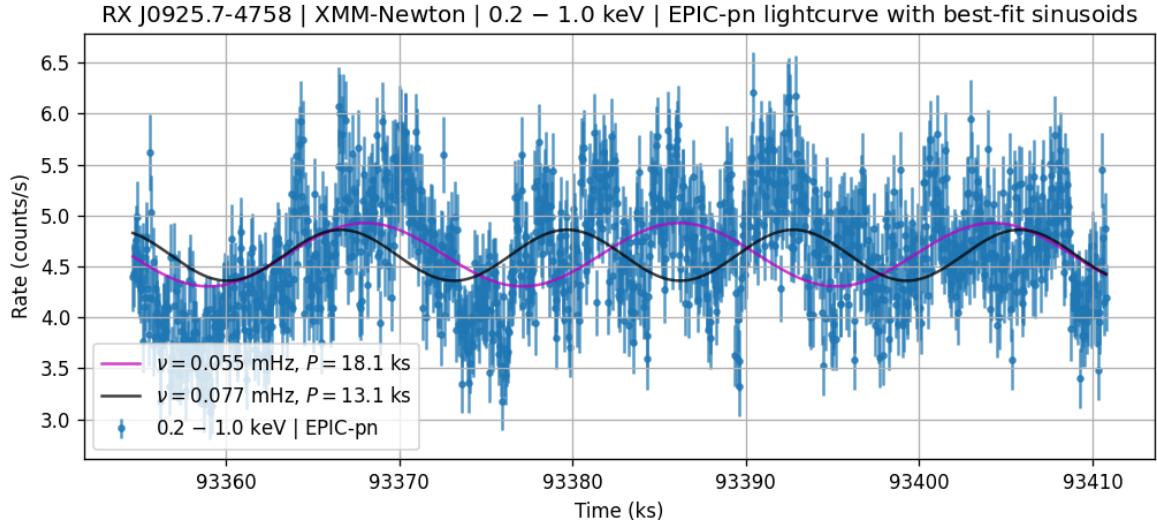


Figure 6.22: XMM-Newton EPIC-pn lightcurve over 0.2–1.0 keV with best-fit sinusoids

6.3.3.3 Best-fit sinusoids for 0.2–12.0 keV EPIC-pn lightcurve

Fitting with phase-folded lightcurve: For the 0.2–12.0 keV photon energy range using the EPIC-pn instrument of the XMM-Newton observatory, sinusoidal fits were also performed at the frequencies of 0.055 mHz and 0.077 mHz. The phase-folded lightcurve, covering a full cycle from 0 to 2π radians, is depicted in figure 6.23. The overlaid best-fit sinusoids show a strong correlation with the observed data, confirming the presence of periodic modulations in this broader energy band as well.

Fitting with complete lightcurve: The best-fit sinusoids have been overlaid on the complete lightcurve for the 0.2–12.0 keV photon energy range using the EPIC-pn instrument of the XMM-Newton observatory, as shown in figure 6.24. Here as well, this allows for a visual assessment of how well the sinusoidal models align with the observed data across the entire energy range.

6.3. Variability study of RX J0925.7-4758

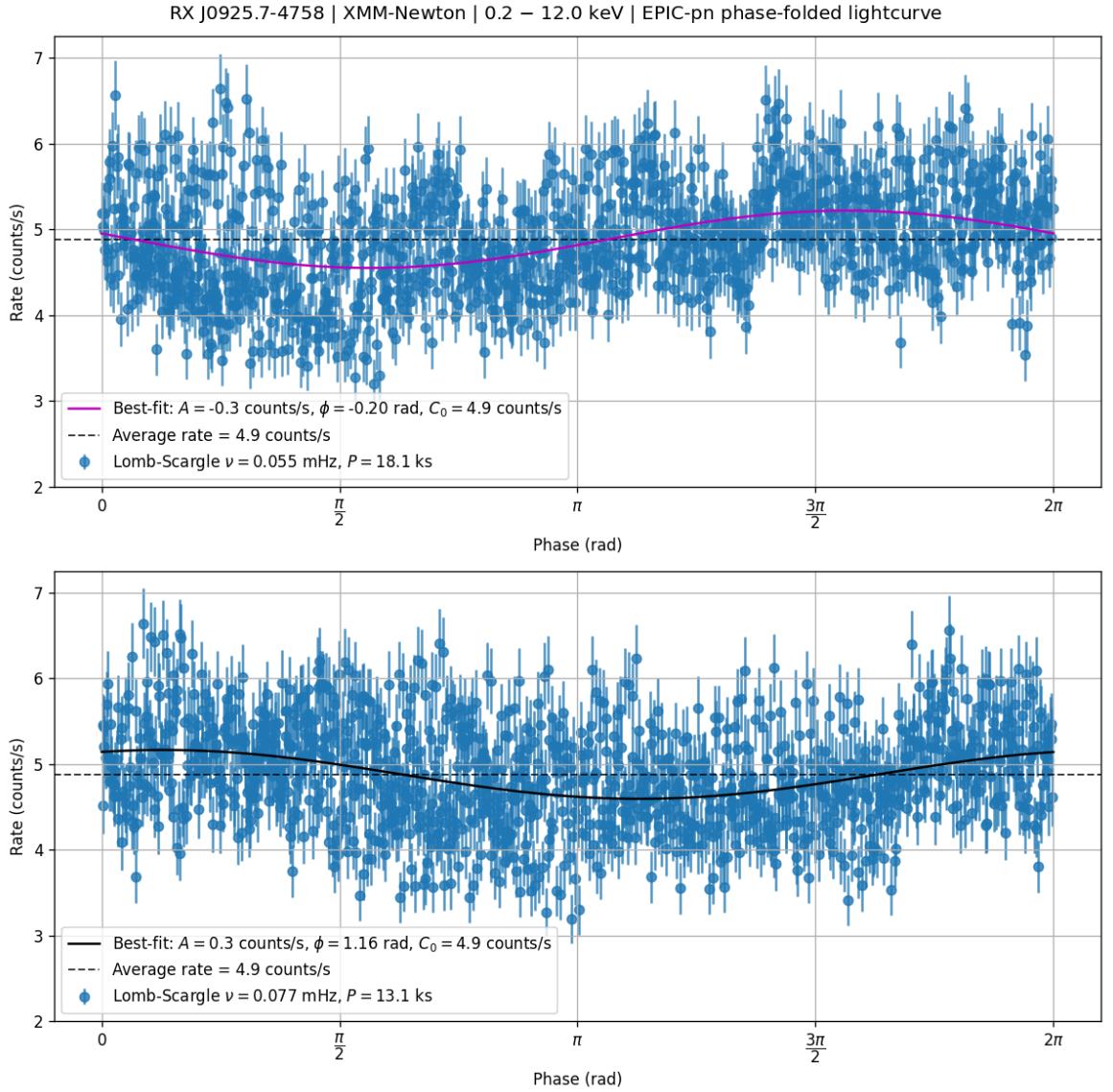


Figure 6.23: Phase-folded XMM-Newton EPIC-pn lightcurve over 0.2–12.0 keV with best-fit sinusoids

6.3.3.4 Best-fit sinusoid for 0.2–12.0 keV RGS lightcurve

Fitting with phase-folded lightcurve: For the 0.2–12.0 keV photon energy range using the RGS instrument of XMM-Newton, sinusoidal fits were applied using only the frequency of 0.055 mHz, as the peak at 0.077 mHz was relatively much weaker (as it can be observed in figure 6.20). The resulting phase-folded lightcurve, covering a full cycle from 0 to 2π radians, is presented in figure 6.25. The best-fit sinusoid at 0.055 mHz is overlaid on the

6.3. Variability study of RX J0925.7-4758

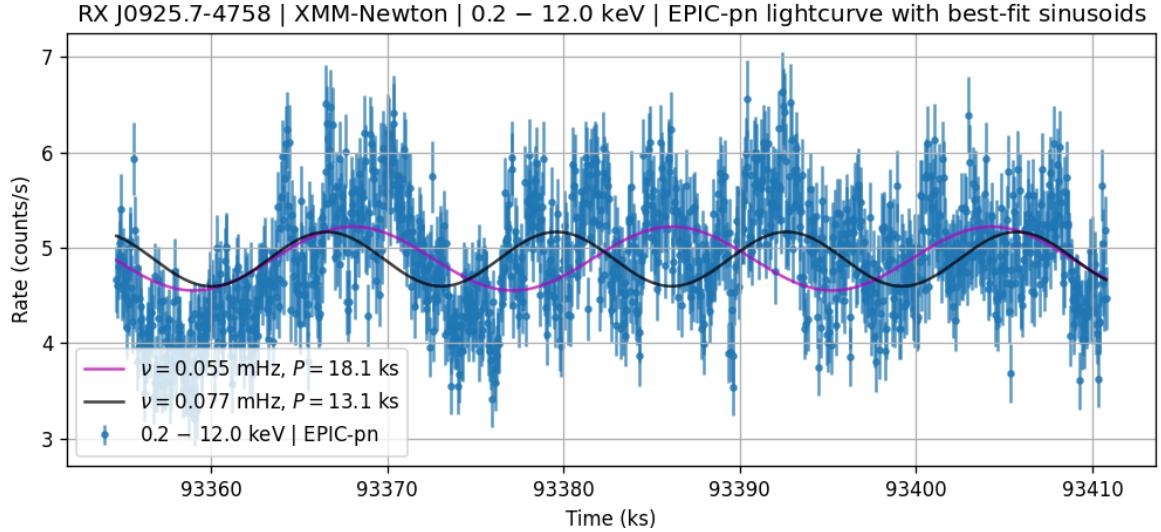


Figure 6.24: XMM-Newton EPIC-pn lightcurve over 0.2–12.0 keV with best-fit sinusoids

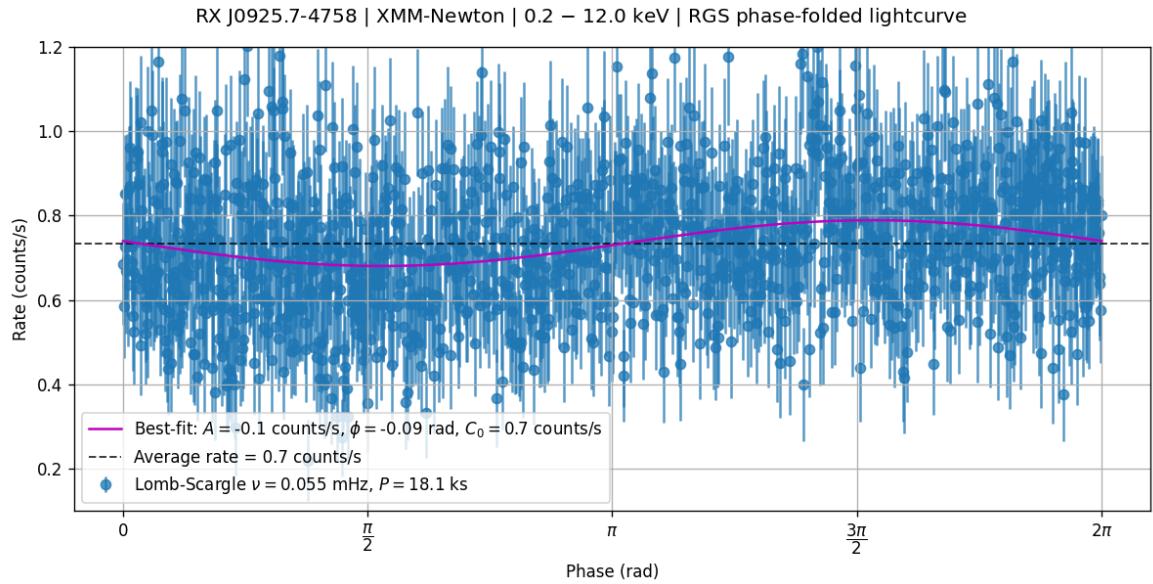


Figure 6.25: Phase-folded XMM-Newton RGS lightcurve over 0.2–12.0 keV with best-fit sinusoid

data, demonstrating a notable correlation with the observed lightcurve.

Fitting with complete lightcurve: The best-fit sinusoid for the 0.2–12.0 keV photon energy range from the RGS instrument of XMM-Newton is overlaid on the complete lightcurve, as shown in figure 6.26. This visualisation enables a direct assessment of how well the sinu-

6.3. Variability study of RX J0925.7-4758

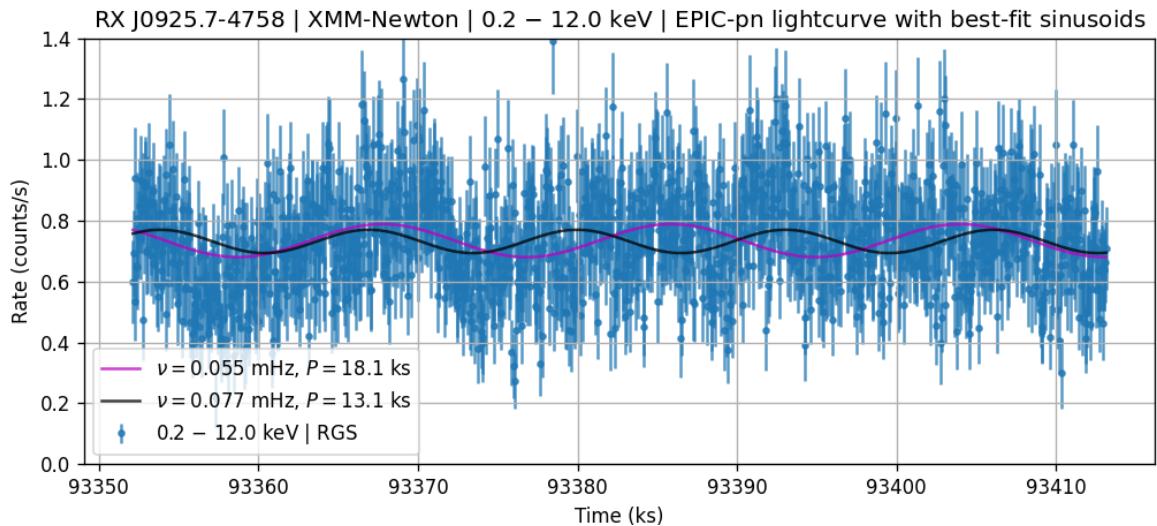


Figure 6.26: XMM-Newton RGS lightcurve over 0.2–12.0 keV with best-fit sinusoid

soidal model aligns with the observed data across the entire energy range.

6.3.4 Summary of timing analysis of RX J0925.7-4758

Table 6.8 summarizes the timing analysis results for RX J0925.7-4758 based on NICER and XMM-Newton light curves. Lomb-Scargle periodograms were employed to identify significant periodicities, with corresponding peak frequencies, periods, and power densities tabulated.

In order to characterize the sinusoidal modulation, phase-folded light curves were fitted with a simple sinusoid model. The derived parameters include amplitude (A), average signal level (C_0), and initial phase (ϕ). Additionally, an *amplitude-to-mean ratio* (AMR) was calculated as a percentage, quantifying the variability strength relative to the mean signal level. AMR provides a valuable metric for assessing the significance of the observed periodic behavior.

6.3. Variability study of RX J0925.7-4758

Table 6.8: Results of variability studies on RX J0925.7-4758

Observatory and Instrument	Photon energy range	Lomb-Scargle periodogram				Best-fit sinusoid			
		Freq. (MHz)	Period (ks)	Power per freq.	FAP	A (counts/s)	ϕ (rad)	C_0 (counts/s)	AMR
NICER XTI	0.2 – 1.0 keV	0.009	112.3	0.480	3.9×10^{-25}	1.56	0.50	10.32	15.1%
	0.2 – 12.0 keV	0.017	58.8	0.490	6.1×10^{-26}	1.25	1.57	10.39	12.0%
XMM- Newton EPIC-pn	0.2 – 1.0 keV	0.055	18.1	0.157	7.1×10^{-38}	0.31	-0.20	4.62	6.7%
	0.2 – 12.0 keV	0.077	13.1	0.095	4.3×10^{-23}	0.25	1.12	4.61	5.4%
XMM- Newton RGS	0.2 – 12.0 keV	0.055	18.1	0.160	8.9×10^{-21}	0.33	-0.20	4.89	6.8%
		0.077	13.1	0.106	1.2×10^{-18}	0.28	1.16	4.88	5.8%

CHAPTER 7

CONCLUSIONS AND DISCUSSION

Contents

7.1 Physical Interpretation	146
7.2 Implications	146
7.3 Future Scope	147

Abstract

This chapter offers a comprehensive synthesis of findings from the study of Supersoft X-ray Sources (SSS), focusing on their physical interpretation, implications, and future research directions. The chapter highlights the diversity of SSS types in both the Milky Way and the Large Magellanic Cloud, noting differences in spectral features, luminosity, and X-ray characteristics that suggest varying environmental and intrinsic properties. The detection of absorption edges in Milky Way SSS versus their absence in LMC counterparts implies significant interstellar medium absorption effects. Higher luminosity in sources like RX J0925.7-4758 is linked to elevated mass accretion rates, which are also associated with distinctive spectral features. The application of a multi-component non-local thermodynamic equilibrium (NLTE) model effectively captures the continuum spectra, while low-frequency modulations suggest complex mechanisms such as orbital variations and stellar pulsations. These findings provide insight into the evolutionary pathways of SSS, influenced by factors like companion type, accretion processes, and magnetic field strength. The successful use of the Lomb-Scargle periodogram underscores its value in characterising periodic variability in SSS. Looking forward, the chapter calls for extended spectral analysis, broader sample studies, high-resolution timing observations, enhanced theoretical modelling, and cross-instrument data comparisons to deepen our understanding of these enigmatic systems.

7.1 Physical Interpretation

- *Diverse SSS Types:* We have assembled a unique dataset encompassing a variety of SSS types (X-ray binaries, cataclysmic variables, and a symbiotic system) from both the Milky Way and the Large Magellanic Cloud, providing a diverse sample for our investigation.
- *Spectral Features:* Our analysis has revealed that the SSS in the Milky Way exhibit discernible absorption edges, suggesting differences in local environmental conditions or interstellar absorption. These findings are novel and contribute to our understanding of the interstellar medium in different galactic environments.
- *Luminosity and Distance:* We have discovered that RX J0925.7-4758, despite its greater distance, exhibits significantly higher luminosity than RS Oph. This implies a higher mass accretion rate in RX J0925.7-4758, which is a novel finding and provides valuable insights into the accretion processes in SSS.
- *Softer X-ray Characteristics:* Our analysis has identified softer X-ray emissions from certain SSS (namely CAL 83, RX J0527.8-6954, and RX J0019.8+2156), indicating differences in their effective temperatures and accretion processes. These findings are new and contribute to our understanding of the diversity among SSS.
- *Spectral Modelling:* We have successfully employed a multi-component NLTE model to accurately capture the continuum spectrum of RX J0925.7-4758. This novel approach demonstrates the potential of similar modeling techniques to constrain the stellar parameters of other SSS, providing valuable insights into their physical properties.

7.2 Implications

- *Galactic vs. Extragalactic SSS:* The absence of clear absorption edges in LMC SSS, compared to Milky Way SSS, suggests differences in local environmental conditions or the impact of interstellar absorption, potentially offering insights into the ISM properties of different galaxies.
- *Mass Accretion Insights:* Higher luminosity and the presence of multiple absorption

7.3. Future Scope

edges in RX J0925.7-4758 highlight the role of high accretion rates in shaping the observed spectral characteristics, which could help model the evolution of such systems.

- *Periodic Variability:* The detection of low-frequency modulations in RX J0925.7-4758 points to possible mechanisms such as orbital variations, accretion dynamics, or stellar pulsations, which can influence the emission properties of SSS.
- *Evolutionary Pathways:* Differences in spectral hardness and variability between the SSS suggest diverse evolutionary pathways, potentially influenced by factors like companion type, accretion rate, and magnetic field strength.
- *Diagnostic Tools:* The successful application of the Lomb-Scargle periodogram to identify periodic signals reinforces its utility in the timing analysis of variable X-ray sources, providing a robust method for detecting and characterising periodicity.

7.3 Future Scope

- *Extended Spectral Analysis:* Further analysis of high-resolution spectra from instruments like RGS could help identify more subtle spectral features, enhancing our understanding of the atmospheres and environments of these SSS.
- *Broader Sample Studies:* Expanding the dataset to include more SSS from different galaxies could improve statistical robustness and help uncover universal properties or distinct characteristics linked to specific galactic environments.
- *High-Resolution Timing Studies:* More precise timing observations using advanced instruments can help distinguish between intrinsic variability and observational artefacts, leading to a clearer understanding of the periodic phenomena in SSS.
- *Theoretical Modelling:* Development of more sophisticated NLTE models incorporating complex absorption and emission processes could provide better fits to observed spectra and yield more accurate stellar parameters.
- *Cross-Instrument Comparisons:* Combining data from different observatories like NICER, XMM-Newton, and future missions could refine our understanding of SSS variability and emission mechanisms, leveraging each instrument's strengths.

7.3. Future Scope

- *Study of Circumstellar Material:* Investigating the role of circumstellar material, such as stellar winds or disks, in shaping the observed X-ray characteristics could provide insights into the accretion dynamics and evolution of these systems.

Bibliography

- [1] K. Nomoto, “Accreting white dwarf models for type I supernovae. I-Presupernova evolution and triggering mechanisms,” *The Astrophysical Journal*, vol. 253, pp. 798–810, 1982.
- [2] R. di Stefano and L. Nelson, “On the evolution, numbers and characteristics of close-binary supersoft sources,” in *Supersoft X-Ray Sources* (J. Greiner, ed.), pp. 3–14, Springer, 1996.
- [3] R. Giacconi, H. Gursky, F. R. Paolini, and B. B. Rossi, “Evidence for x-rays from sources outside the solar system,” *Physical Review Letters*, vol. 9, no. 11, p. 439, 1962.
- [4] Y. Tanaka and N. Shibasaki, “X-ray novae,” *Annual Review of Astronomy and Astrophysics*, vol. 34, no. 1, pp. 607–644, 1996.
- [5] R. Staubert and J. Trümper, “Overview,” in *The Universe in X-rays* (J. Trümper and G. Hasinger, eds.), ch. 1, pp. 3–4, Springer Science & Business Media, 2008.
- [6] F. Jansen, D. Lumb, B. Altieri, J. Clavel, M. Ehle, C. Erd, C. Gabriel, M. Guainazzi, P. Gondoin, R. Much, *et al.*, “Xmm-newton observatory-i. the spacecraft and operations,” *Astronomy & Astrophysics*, vol. 365, no. 1, pp. L1–L6, 2001.
- [7] M. C. Weisskopf, H. D. Tananbaum, L. P. Van Speybroeck, and S. L. O’Dell, “Chandra x-ray observatory (cxo): overview,” *X-ray optics, instruments, and missions iii*, vol. 4012, pp. 2–16, 2000.
- [8] P. Bhattacharya and M. Baruah, “Supersoft X-ray Sources: A Review Study,” *Journal of Applied and Fundamental Sciences*, vol. 3, no. 1, pp. 30–41, 2017.
- [9] J. Trümper, G. Hasinger, B. Aschenbach, H. Braeuninger, U. Briel, W. Burkert, H. Fink, E. Pfeffermann, W. Pietsch, P. Predehl, J. Schmitt, W. Voges, U. Zimmermann, and K. Beuermann, “X-ray survey of the Large Magellanic Cloud by ROSAT,” *Nature*, vol. 349, pp. 579–583, February 1991.
- [10] J. Greiner, G. Hasinger, and P. Kahabka, “ROSAT observation of two supersoft sources in the Large Magellanic Cloud,” *Astronomy and Astrophysics*, vol. 246, pp. L17–L20, April 1991.

BIBLIOGRAPHY

- [11] P. Kahabka and E. van den Heuvel, “Super-soft sources,” in *Compact Stellar X-Ray Sources*, ch. 11, pp. 461–474, Cambridge University Press, 2006.
- [12] H. Bearda, W. Hartmann, K. Ebisawa, J. Heise, J. Kaastra, R. van der Meer, F. Verbunt, and C. Motch, “The complex X-ray spectrum of the supersoft source MR Vel,” *Astronomy and Astrophysics*, vol. 385, pp. 511–516, 2002.
- [13] C. Motch, H. Bearda, and C. Neiner, “XMM-Newton observations of MR Vel/RX J0925.7-4758,” *Astronomy and Astrophysics*, vol. 393, pp. 913–920, 2002.
- [14] T. Lanz, G. A. Telis, M. Audard, F. Paerels, A. P. Rasmussen, and I. Hubeny, “Non-LTE model atmosphere analysis of the Large Magellanic Cloud supersoft X-ray source CAL 83,” *The Astrophysical Journal*, vol. 619, no. 1, p. 517, 2005.
- [15] M. Orio, K. Ebisawa, J. Heise, and W. Hartmann, “A new view of the supersoft X-ray source CAL 87 observed with XMM-Newton,” *arXiv preprint astro-ph/0402040*, 2004.
- [16] R. Schwarz and K. Reinsch, “XMM-Newton observations of the galactic supersoft X-ray source QR And,” *Astronomische Nachrichten Supplement*, vol. 325, p. 107, 2004.
- [17] E. van den Heuvel, D. Bhattacharya, K. Nomoto, and S. Rappaport, “Accreting white dwarf models for CAL 83, CAL 87 and other ultrasoft X-ray sources in the LMC,” *Astronomy and Astrophysics*, vol. 262, pp. 97–105, 1992.
- [18] K. S. Long, D. J. Helfand, and D. A. Grabelsky, “A soft X-ray study of the Large Magellanic Cloud,” *The Astrophysical Journal*, vol. 248, pp. 925–944, 1981.
- [19] F. Seward and M. Mitchell, “X-ray survey of the Small Magellanic Cloud,” *The Astrophysical Journal*, vol. 243, pp. 736–743, 1981.
- [20] P. Kahabka and E. van den Heuvel, “Luminous Supersoft X-ray Sources,” *Annual Review of Astronomy and Astrophysics*, vol. 35, pp. 342–351, 1997.
- [21] I. Hachisu, M. Kato, and K. Nomoto, “A new model for progenitor systems of Type Ia supernovae,” *The Astrophysical Journal Letters*, vol. 470, no. 2, p. L97, 1996.
- [22] B. Paczynski, “Evolutionary processes in close binary systems,” *Annual Review of Astronomy and Astrophysics*, vol. 9, p. 183, 1971.

BIBLIOGRAPHY

- [23] J. Greiner, “Catalog of supersoft X-ray sources,” *New Astronomy*, vol. 5, no. 3, pp. 137–141, 2000.
- [24] E. M. Sion and S. G. Starrfield, “Evolutionary sequences of very hot, low-mass, accreting white dwarfs with application to symbiotic variables and ultrasoft/supersoft low-luminosity X-ray sources,” *The Astrophysical Journal*, vol. 421, pp. 261–268, 1994.
- [25] S. Starrfield, C. Iliadis, and W. R. Hix, “Thermonuclear processes,” in *Classical Novae, 2nd ed.* (M. Bode and A. Evans, eds.), pp. 77–101, Cambridge, 2008.
- [26] J. Krautter, H. Ögelman, S. Starrfield, R. Wichmann, and E. Pfeffermann, “ROSAT X-ray observations of nova V1974 Cygni: The rise and fall of the brightest supersoft X-ray source,” *The Astrophysical Journal*, vol. 456, p. 788, 1996.
- [27] S. Balman, J. Krautter, and H. Oegelman, “The X-ray spectral evolution of classical nova V1974 Cygni 1992: A reanalysis of the ROSAT data,” *The Astrophysical Journal*, vol. 499, no. 1, p. 395, 1998.
- [28] N. R. Lomb, “Least-squares frequency analysis of unequally spaced data,” *Astrophysics and space science*, vol. 39, pp. 447–462, 1976.
- [29] J. D. Scargle, “Studies in astronomical time series analysis. ii-statistical aspects of spectral analysis of unevenly spaced data,” *Astrophysical Journal, Part 1*, vol. 263, Dec. 15, 1982, p. 835-853., vol. 263, pp. 835–853, 1982.
- [30] A. Schuster, “On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena,” *Terrestrial Magnetism*, vol. 3, no. 1, pp. 13–41, 1898.
- [31] R. V. Bulavov, “Assessing the statistical significance of periodogram peaks,” *Monthly Notices of the Royal Astronomical Society*, vol. 385, no. 3, pp. 1279–1285, 2008.
- [32] J. Trümper, G. Hasinger, B. Aschenbach, H. Braeuninger, U. Briel, W. Burkert, H. Fink, E. Pfeffermann, W. Pietsch, P. Predehl, *et al.*, “X-ray survey of the Large Magellanic Cloud by ROSAT,” *Nature*, vol. 349, no. 6310, pp. 579–583, 1991.
- [33] J. Greiner, G. Hasinger, and P. Kahabka, “ROSAT observation of two supersoft sources in the Large Magellanic Cloud,” *Astronomy and Astrophysics (ISSN 0004-6361)*, vol. 246, no. 1, June 1991, p. L17-L20., vol. 246, pp. L17–L20, 1991.

BIBLIOGRAPHY

- [34] P. Kahabka and J. Trumper, “Supersoft ROSAT Sources in the Galaxies,” in *Compact Stars in Binaries* (J. van Paradijs, E. P. J. van den Heuvel, and E. Kuulkers, eds.), vol. 165, p. 425, 1996.
- [35] J. E. Steiner and M. P. Diaz, “The V Sagittae Stars,” *PASP*, vol. 110, no. 745, pp. 276–282, 1998.
- [36] J. Greiner, “Catalog of supersoft X-ray sources,” *New Astronomy*, vol. 5, pp. 137–141, May 2000.
- [37] W. Pietsch, M. Ehle, F. Haberl, Z. Misanovic, and G. Trinchieri, “Deep XMM-Newton survey of M33,” *Astronomische Nachrichten: Astronomical Notes*, vol. 324, no. 1-2, pp. 85–88, 2003.
- [38] R. Di Stefano and A. Kong, “Luminous supersoft X-ray sources in external galaxies,” *The Astrophysical Journal*, vol. 592, no. 2, p. 884, 2003.
- [39] M. Orio, T. Nelson, A. Bianchini, F. Di Mille, and D. Harbeck, “A census of the supersoft X-ray sources in M31,” *The Astrophysical Journal*, vol. 717, no. 2, p. 739, 2010.
- [40] M. Henze, W. Pietsch, F. Haberl, G. Sala, M. Hernanz, D. Hatzidimitriou, A. Rau, D. H. Hartmann, J. Greiner, M. Orio, *et al.*, “Recent discoveries of supersoft X-ray sources in M 31,” *Astronomische Nachrichten: Astronomical Notes*, vol. 331, no. 2, pp. 193–196, 2010.
- [41] R. Sturm, F. Haberl, W. Pietsch, M. Coe, S. Mereghetti, N. La Palombara, R. Owen, and A. Udalski, “A new super-soft X-ray source in the Small Magellanic Cloud: Discovery of the first Be/white dwarf system in the SMC?,” *Astronomy & Astrophysics*, vol. 537, p. A76, 2012.
- [42] I. Galiullin and M. Gilfanov, “Populations of super-soft X-ray sources in galaxies of different morphological types,” *Astronomy & Astrophysics*, vol. 646, p. A85, 2021.
- [43] E. Van den Heuvel, D. Bhattacharya, K. Nomoto, and S. Rappaport, “Accreting white dwarf models for CAL 83, CAL 87 and other ultrasoft X-ray sources in the LMC,” *Astronomy and Astrophysics*, vol. 262, pp. 97–105, 1992.

BIBLIOGRAPHY

- [44] N. Kylafis and E. Xilouris, “Low-mass X-ray binary models for the supersoft X-ray sources CAL 83, CAL 87 and RX J0527. 8-6954 in the Large Magellanic Cloud,” *Astronomy and Astrophysics*, vol. 278, pp. L43–L46, 1993.
- [45] B. Paczyński and A. Zytkow, “Hydrogen shell flashes in a white dwarf with mass accretion,” *The Astrophysical Journal*, vol. 222, pp. 604–611, 1978.
- [46] D. Prialnik, M. Shara, and G. Shaviv, “The evolution of a slow nova model with a Z= 0.03 envelope from pre-explosion to extinction,” *Astronomy and Astrophysics*, vol. 62, pp. 339–348, 1978.
- [47] E. Sion, M. Acierno, and S. Tomczyk, “Hydrogen shell flashes in massive accreting white dwarfs,” *The Astrophysical Journal*, vol. 230, pp. 832–838, 1979.
- [48] R. Sienkiewicz, “Stability of white dwarfs undergoing spherically symmetric steady-state accretion,” *Astronomy and Astrophysics*, vol. 85, pp. 295–301, 1980.
- [49] M. Fujimoto, “A Theory of Hydrogen Shell Flashes on Accreting White Dwarfs. I. Their Progress and the Expansion of the Envelope,” *The Astrophysical Journal*, vol. 257, pp. 752–766, 1982.
- [50] M. Fujimoto, “A Theory of Hydrogen Shell Flashes on Accreting White Dwarfs. II. The Stable Shell Burning and the Recurrence Period of Shell Flashes,” *The Astrophysical Journal*, vol. 257, pp. 767–79, 1982.
- [51] I. Iben Jr, “Hot accreting white dwarfs in the quasi-static approximation,” *The Astrophysical Journal*, vol. 259, pp. 244–266, 1982.
- [52] D. Prialnik and A. Kovetz, “An extended grid of multicycle nova evolution models,” *The Astrophysical Journal*, vol. 445, pp. 789–810, 1995.
- [53] J. MacDonald, “CNO abundances and the strengths of nova outbursts and hydrogen flashes on accreting white dwarfs,” *The Astrophysical Journal*, vol. 267, pp. 732–746, 1983.
- [54] B. Paczynski and B. Rudak, “Symbiotic stars: Evolutionary considerations,” *Astronomy and Astrophysics*, vol. 82, pp. 349–351, 1980.
- [55] I. Hachisu and M. Kato, “Recurrent novae as a progenitor system of type Ia supernovae. I. RS Ophiuchi subclass: systems with a red giant companion,” *The Astrophysical Journal*, vol. 558, no. 1, 2001.

BIBLIOGRAPHY

- [56] C. Motch, G. Hasinger, and W. Pietsch, “Discovery of a luminous galactic supersoft X-ray source in the ROSAT all-sky survey,” *Astronomy and Astrophysics*, vol. 284, pp. 827–838, 1994.
- [57] H. Hartmann, J. Heise, P. Kahabka, C. Motch, and A. Parmar, “Constraining the spectral parameters of RXJ0925. 7-4758 with the BeppoSAX LECS,” *Astronomy and Astrophysics*, vol. 346, pp. 125–133, 1999.
- [58] H. Beard, W. Hartmann, K. Ebisawa, J. Heise, J. Kaastra, R. van der Meer, F. Verbunt, and C. Motch, “The complex X-ray spectrum of the supersoft source MR Vel,” *Astronomy & Astrophysics*, vol. 385, no. 2, pp. 511–516, 2002.
- [59] C. Motch, H. Beard, and C. Neiner, “XMM-Newton observations of MR Vel/RX J0925. 7-4758,” *Astronomy & Astrophysics*, vol. 393, no. 3, pp. 913–920, 2002.
- [60] K. Ebisawa, K. Mukai, T. Kotani, K. Asai, T. Dotani, F. Nagase, H. Hartmann, J. Heise, P. Kahabka, and A. Van Teeseling, “X-Ray Energy Spectra of the Supersoft X-Ray Sources CAL 87 and RX J0925. 7-4758 Observed with ASCA,” *The Astrophysical Journal*, vol. 550, no. 2, p. 1007, 2001.
- [61] M. Orio, K. Gendreau, M. Giese, G. J. M. Luna, J. Magdolen, S. Pei, B. Sun, E. Behar, A. Dobrotka, J. Mikolajewska, D. R. Pasham, and T. E. Strohmayer, “NICER Monitoring of Supersoft X-Ray Sources,” *Astrophysical Journal*, vol. 932, p. 45, June 2022.
- [62] P. E. Stecchini, M. Perez Diaz, F. D’Amico, and F. Jablonski, “Revisiting multiwavelength data on the supersoft X-ray source CAL 83,” *Monthly Notices of the Royal Astronomical Society*, vol. 522, no. 3, pp. 3472–3480, 2023.
- [63] S. Mateos, R. Saxton, A. Read, and S. Sembay, “Statistical evaluation of the flux cross-calibration of the XMM-Newton EPIC cameras,” *Astronomy & Astrophysics*, vol. 496, no. 3, pp. 879–889, 2009.
- [64] J.-U. Ness, “The complications of learning from Super Soft Source X-ray spectra,” *Advances in Space Research*, vol. 66, no. 5, pp. 1202–1208, 2020.
- [65] P. Jethwa, R. Saxton, M. Guainazzi, P. Rodriguez-Pascual, and M. Stuhlinger, “When is pile-up important in the XMM-Newton EPIC cameras?,” *Astronomy & Astrophysics*, vol. 581, p. A104, 2015.

BIBLIOGRAPHY

- [66] J. A. Bearden and A. Burr, “Reevaluation of X-ray atomic energy levels,” *Reviews of Modern Physics*, vol. 39, no. 1, p. 125, 1967.
- [67] A. M. Juett, N. S. Schulz, D. Chakrabarty, and T. W. Gorczyca, “High-resolution X-ray spectroscopy of the interstellar medium. II. Neon and iron absorption edges,” *The Astrophysical Journal*, vol. 648, no. 2, p. 1066, 2006.
- [68] N. Prodhani and M. Baruah, “The galactic luminous supersoft X-ray source RXJ0925.7-4758/MR Vel,” *Pramana*, vol. 90, pp. 1–5, 2018.
- [69] I. Hubeny and D. Mihalas, *Theory of stellar atmospheres: An introduction to astrophysical non-equilibrium quantitative spectroscopic analysis*, vol. 26. Princeton University Press, 2014.
- [70] I. Hubeny, “Accelerated lambda iteration: An overview,” *Stellar Atmosphere Modeling*, vol. 288, p. 17, 2003.
- [71] L. H. Auer and D. Mihalas, “Non-lte model atmospheres. iv. results for multi-line computations,” *Astrophysical Journal*, vol. 160, p. 233, vol. 160, p. 233, 1970.
- [72] K. Werner and S. Dreizler, “The classical stellar atmosphere problem,” *Journal of Computational and Applied Mathematics*, vol. 109, no. 1-2, pp. 65–93, 1999.
- [73] T. Rauch, “A grid of synthetic ionizing spectra for very hot compact stars from NLTE model atmospheres,” *Astronomy & Astrophysics*, vol. 403, no. 2, pp. 709–714, 2003.
- [74] T. Rauch, M. Orio, R. Gonzales-Riestra, T. Nelson, M. Still, K. Werner, and J. Wilms, “NLTE model atmospheres for the hottest white dwarfs: Spectral analysis of the compact component in nova V4743 Sgr,” *arXiv preprint arXiv:1006.2918*, 2010.
- [75] M. Lemke, “Extended VCS Stark broadening tables for hydrogen–Lyman to Brackett series,” *Astronomy and Astrophysics Supplement Series*, vol. 122, no. 2, pp. 285–292, 1997.
- [76] P. Bhattacharya, B. M. Lyngdoh, J. I. Nongrum, R. Misra, and M. Baruah, “A Python-based tool for spectral line identification in RGS spectra from XMM-Newton,” *ADBU Journal of Engineering Technology*, vol. 9, 2020.
- [77] J.-U. Ness, J. P. Osborne, M. Henze, A. Dobrotka, J. Drake, V. Ribeiro, S. Starrfield, E. Kuulkers, E. Behar, M. Hernanz, *et al.*, “Obscuration effects in super-soft-source X-ray spectra,” *Astronomy & Astrophysics*, vol. 559, p. A50, 2013.

BIBLIOGRAPHY

- [78] E. Gatuzz, J. García, T. R. Kallman, C. Mendoza, and T. W. Gorczyca, “ISMabs: A Comprehensive X-Ray Absorption Model for the Interstellar Medium,” *The Astrophysical Journal*, vol. 800, no. 1, p. 29, 2015.
- [79] D. A. Verner and D. G. Yakovlev, “Analytic FITS for partial photoionization cross sections.,” *Astronomy and Astrophysics Supplement*, vol. 109, pp. 125–133, 1995.
- [80] M. C. Witthoeft, M. A. Bautista, C. Mendoza, T. R. Kallman, P. Palmeri, and P. Quinet, “K-Shell Photoionization and Photoabsorption of Ne, Mg, Si, S, Ar, and Ca,” *The Astrophysical Journal Supplement*, vol. 182, no. 1, pp. 127–130, 2009.
- [81] M. C. Witthoeft, J. García, T. R. Kallman, M. A. Bautista, C. Mendoza, P. Palmeri, and P. Quinet, “K-shell Photoionization of Na-like to Cl-like Ions of Mg, Si, S, Ar, and Ca,” *The Astrophysical Journal Supplement*, vol. 192, no. 1, p. 7, 2011.
- [82] J. García, T. R. Kallman, M. Witthoeft, E. Behar, C. Mendoza, P. Palmeri, P. Quinet, M. A. Bautista, and M. Klapisch, “Nitrogen K-Shell Photoabsorption,” *The Astrophysical Journal Supplement*, vol. 185, no. 2, pp. 477–485, 2009.
- [83] T. W. Gorczyca, M. A. Bautista, M. F. Hasoglu, J. García, E. Gatuzz, J. S. Kaastra, T. R. Kallman, S. T. Manson, C. Mendoza, A. J. J. Raassen, C. P. de Vries, and O. Zatsarinny, “A Comprehensive X-Ray Absorption Model for Atomic Oxygen,” *The Astrophysical Journal*, vol. 779, no. 1, p. 78, 2013.
- [84] J. García, C. Mendoza, M. A. Bautista, T. W. Gorczyca, T. R. Kallman, and P. Palmeri, “K-Shell Photoabsorption of Oxygen Ions,” *The Astrophysical Journal Supplement*, vol. 158, no. 1, pp. 68–79, 2005.
- [85] E. Gatuzz, J. García, C. Mendoza, T. R. Kallman, M. Witthoeft, A. Lohfink, M. A. Bautista, P. Palmeri, and P. Quinet, “Photoionization Modeling of Oxygen K Absorption in the Interstellar Medium: The Chandra Grating Spectra of XTE J1817-330,” *The Astrophysical Journal*, vol. 768, no. 1, p. 60, 2013.
- [86] T. W. Gorczyca, “Auger decay of the photoexcited $1s^{-1}np$ Rydberg series in neon,” *Physical Review A*, vol. 61, no. 2, p. 024702, 2000.
- [87] T. W. Gorczyca, “Production and Decay of Atomic Inner-Shell Vacancy States,” in *X-ray Diagnostics of Astrophysical Plasmas: Theory, Experiment, and Observation* (R. Smith, ed.), vol. 774 of *American Institute of Physics Conference Series*, pp. 223–235, 2005.

BIBLIOGRAPHY

- [88] J. B. Kortright and S.-K. Kim, “Resonant magneto-optical properties of Fe near its 2p levels: Measurement and applications,” *Physical Review B*, vol. 62, no. 18, pp. 12216–12228, 2000.
- [89] R. K. Smith, N. S. Brickhouse, D. A. Liedahl, and J. C. Raymond, “Collisional Plasma Models with APEC/APED: Emission-Line Diagnostics of Hydrogen-like and Helium-like Ions,” *The Astrophysical Journal*, vol. 556, no. 2, p. L91–L95, 2001.
- [90] R. Mewe, E. H. B. M. Gronenschild, and G. H. J. van den Oord, “Calculated X-radiation from optically thin plasmas. V.” *Astronomy and Astrophysics Supplement*, vol. 62, pp. 197–254, 1985.
- [91] D. A. Liedahl, A. L. Osterheld, and W. H. Goldstein, “New Calculations of Fe L-Shell X-Ray Spectra in High-Temperature Plasmas,” *The Astrophysical Journal*, vol. 438, p. L115, 1995.
- [92] M. Gierliński and C. Done, “Energy-dependent variability and the origin of the soft X-ray excess in active galactic nuclei,” *Monthly Notices of the Royal Astronomical Society*, vol. 371, no. 1, pp. L16–L20, 2006.
- [93] K. Werner and S. Dreizler, “The classical stellar atmosphere problem.” *Journal of Computational and Applied Mathematics*, vol. 109, no. 1, pp. 65–93, 1999.
- [94] K. Werner, J. L. Deetjen, S. Dreizler, T. Nagel, T. Rauch, and S. L. Schuh, “Model Photospheres with Accelerated Lambda Iteration,” in *Stellar Atmosphere Modeling* (I. Hubeny, D. Mihalas, and K. Werner, eds.), vol. 288 of *Astronomical Society of the Pacific Conference Series*, p. 31, 2003.
- [95] K. Werner, T. Rauch, and S. Dreizler, “A users’s guide to the Tübingen NLTE Model Atmosphere Package,” *Institut für Astronomie und Astrophysik, Astronomie, Tübingen*, 1998.
- [96] T. Rauch. http://astro.uni-tuebingen.de/~rauch/TMAF/flux_HHeCNONEMgSiS_gen.html. email communication.
- [97] K. Ebisawa, K. Asai, K. Mukai, A. Smale, T. Dotani, H. Hartmann, and J. Heise, “The X-ray energy spectrum of RX J0925. 7-4758 with ASCA,” in *Supersoft X-Ray Sources*, pp. 91–98, Springer, 1996.

BIBLIOGRAPHY

- [98] H. Bearda, W. Hartmann, K. Ebisawa, J. Heise, J. Kaastra, R. van der Meer, F. Verbunt, and C. Motch, “The complex x-ray spectrum of the supersoft source mr vel,” *Astronomy and Astrophysics*, vol. 385, pp. 511–516, 2002.
- [99] C. Motch, H. Bearda, and C. Neiner, “Xmm-newton observations of mr vel/rx j0925.7-4758,” *Astronomy and Astrophysics*, vol. 393, pp. 913–920, 2002.
- [100] M. Ehle, M. Breitfellner, M. Dahlem, M. Guainazzi, P. Rodriguez, M. Santos-Lleo, N. Schartel, and L. Tomas, “Xmm-newton users’ handbook,” *Issue*, vol. 2, p. 2003, 2003.
- [101] L. Chiappetti, M. J. Currie, S. Allen, A. Dobrtyzcki, W. D. Pence, A. Rots, R. Shaw, and W. D. Thompson, “Definition of the flexible image transport system (fits) the fits standard version 4.0: updated 2016 july 22 by the iaufwg original document publication date: 2016 july 22 language-edited document publication date: 2018 august 13,” 2018.
- [102] C. Arviset, M. Guainazzi, J. Hernandez, J. Dowson, P. Osuna, and A. Venet, “The xmm-newton science archive,” *arXiv preprint astro-ph/0206412*, 2002.
- [103] P. Barrett, “Heasarc the high energy astrophysics science archive research center,” *Cataclysmic Variables and Related Physics*, vol. 10, p. 276, 1993.
- [104] I. de la Calle and N. Loiseau, “Users guide to the xmm-newton science analysis system,” *Issue 15.0 (ESA: XMM-Newton SOC)*, 2019.
- [105] Y. Ralchenko, A. Kramida, and J. Reader, “Nist atomic spectra database (version 3.1.5)(national institute of standards and technology, gaithersburg, md, 2008),” *National Institute of Standards and Technology, Gaithersburg, MD*, 2008.
- [106] W. Cunto, C. Mendoza, F. Ochsenbein, and C. Zeippen, “Topbase at the cds,” *Astronomy and Astrophysics*, vol. 275, p. L5, 1993.
- [107] 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, *et al.*, “Astropy: A community python package for astronomy,” *Astronomy & Astrophysics*, vol. 558, p. A33, 2013.

BIBLIOGRAPHY

- [108] A. Ginsburg, B. M. Sipőcz, C. Brasseur, P. S. Cowperthwaite, M. W. Craig, C. Deil, A. M. Groener, J. Guillochon, G. Guzman, S. Liedtke, *et al.*, “Astroquery: an astronomical web-querying package in python,” *The Astronomical Journal*, vol. 157, no. 3, p. 98, 2019.
- [109] “XMM-Newton Users Handbook,” 2020.
- [110] K. Dere, E. Landi, H. Mason, B. M. Fossi, and P. Young, “Chianti-an atomic database for emission lines-i. wavelengths greater than 50 Å,” *Astronomy and Astrophysics Supplement Series*, vol. 125, no. 1, pp. 149–173, 1997.
- [111] R.-P. Kudritzki and J. Puls, “Winds from hot stars,” *Annual Review of Astronomy and Astrophysics*, vol. 38, no. 1, pp. 613–666, 2000.

APPENDIX A

Bio-data of the Scholar

Name: Parag Bhattacharya

Gender: Male

Father's name: Late Surendranath Bhattacharya

Present address: Department of Physics, Rangapara College, Rangapara, Dist. Sonitpur, Assam – 784 505.

Home address: House Ho. 3, Byelane Mathura Boro Path, Nagarik Path, Chandan Nagar, Ghoramara, P.O. Beltola, P.S. Hatigaon, Guwahati, Dist. Kamrup (M), Assam – 781 028.

Nationality: Indian

Date of Birth: 25 February, 1984

Age on 1st January 2024: 39 years

Educational Qualifications: M.Sc. (Physics), M.Tech. (Applied Optics), SLET

Service Experience:

1. Assistant Professor, Department of Physics, Rangapara College, Rangapara, Dist. Sonitpur, Assam. 24 Sep. 2022 – present
2. Assistant Professor, Department of Physics, Assam Don Bosco University, Tapesia Gardens, Sonapur, Assam. 6 Jul. 2015 – 23 Sep. 2022
3. Faculty (Physics), Aakash Institute and Concept Educations, Guwahati, Dist. Kamrup (M), Assam. 19 Oct. 2012 – 4 Jul. 2015
4. PGT (Physics), Delhi Public School Guwahati, Guwahati, Dist. Kamrup (M), Assam. 4 Jan. 2012 – 18 Oct. 2012
5. Guest Faculty, Gauhati University, Guwahati, Dist. Kamrup (M), Assam. 2 Jan. 2012 – 31 Dec. 2012

APPENDIX B

Transcript of the Ph.D. Coursework

Form No. : ADBU|gs|F8-09-04

No.G1600673

Date: 14 December 2016



ASSAM DON BOSCO UNIVERSITY
Azara, Guwahati - 781017, Assam, INDIA

GRADE SHEET

Programme : Ph. D
Semester : Course Work (Autumn 2016)
Name : Parag Bhattacharya
Reg. No. : -

Student ID : DU2016PHD0015

has secured grades as detailed below.

Sl. No.	Course Code	Course	Grade	Grade Points	Credits Allotted	Credits Secured
1.	PDGCLS1003	Literature Survey	E	9	4	4
2.	PDGCRM1002	Research Methodology, Quantitative Techniques and Computer Applications for Research	E	9	4	4
3.	PDPHAP1004	Physics	A+	8	4	4

Total credits secured : 12
GPA : 8.67

(Signature) **Asst. Registrar (Examinations)** **Verified**

(Signature) **Controller of Examinations**

GUWAHATI-781017

Please see overleaf for the grading system

APPENDIX B. TRANSCRIPT OF THE PH.D. COURSEWORK

GRADES, GRADEPOINTS, GPA/SGPA/TGPA and CGPA

The correspondence between letter grades and grade points is given in the table below:

Grade	Description	Grade Points
O	Outstanding	10
E	Excellent	9
A+	Very Good	8
A	Good	7
B	Average	6
C	Below Average	5
F	Failed	0
P	Passed	Not allotted
NP	Not passed	Not allotted
X	Not permitted	Not allotted

The Grade Point Average (GPA), Semester Grade Point Average (SGPA) or Trimester Grade Point Average (TGPA) is calculated using the following formula:

$$GPA/SGPA/TGPA = \frac{\sum_i^n GP_i \times NC_i}{\sum_i^n NC_i}$$

Where GP_i = Grade points earned in the i^{th} course

NC_i = Number of credits for the i^{th} course and

n = the number of courses in the semester/trimester

The Cumulative Grade Point Average (CGPA) is calculated using the following formula:

$$CGPA = \frac{\sum_i^n GPA_i \times NSC_i}{\sum_i^n NSC_i}$$

Where GPA_i = Semester or Trimester Grade Point Average of the i^{th} semester

NSC_i = Number of credits for the i^{th} semester/trimester

n = Number of semesters/trimesters completed

APPENDIX C

List of Publications

C.1 Journal Publications

C.1.1 Published

1. Rabindra Mahato, **Parag Bhattacharya**, Monmoyuri Baruah, “*A Study of ISM in the Line of Sight of NS Transient LMXB MXB 1659-298*”, Galaxies, ISSN: 2075-4434, vol. 12 no. 4 (July 2024), Indexing: Scopus, ESCI (Web of Science).
2. **Parag Bhattacharya**, Bansy M. Lyngdoh, Jessica I. Nongrum, Ranjeev Misra and Monmoyuri Baruah, “*A Python-based tool for spectral line identification in RGS spectra from XMM-Newton*”, ADBU Journal of Engineering & Technology, ISSN: 2348-7305, vol. 9 no. 2 (December 2020), Indexing: UGC-CARE list.
3. **Parag Bhattacharya**, Monmoyuri Baruah, “*Supersoft X-ray Sources: A Review Study*”, Journal of Fundamental & Applied Sciences, ISSN 2395-5554 (Print), 2395-5562 (Online), vol. 3 no. 2 (June 2017), Indexing: UGC list.

C.1.2 Communicated to Journal

1. **Parag Bhattacharya**, Rabindra Mahato, Ranjeev Misra and Monmoyuri Baruah, “*Multi-Observatory Spectral Analysis of the Supersoft X-ray Source RX J0925.7-4758*”, Pramana: Journal of Physics, ISSN: 0973-7111 (Communicated on 30 April 2024), Indexing: Scopus, UGC-CARE list.

C.2 Conference Presentation

1. Parag Bhattacharya, Monmoyuri Baruah and Ranjeev Misra, “*A continuum spectral model for RX J0925.7-4758 using ASCA observations*”, 2nd national conference on

C.2. Conference Presentation

Trends in Modern Physics 2020 (TiMP 2020), Assam Don Bosco University, Sonapur, Assam.

2. Parag Bhattacharya, Flossie B. F. C. Marak and Monmoyuri Baruah, “*Comparative Study of the H β and HeI 4922Å Absorption Lines in the Synthetic Spectra of O-Type Sub-Dwarf Stars*”, North-East Meet of Astronomers 2019 (NEMA-5), Tezpur University, Tezpur, Assam.
3. Parag Bhattacharya, Washim Akram and Monmoyuri Baruah, “*Theoretical Modelling of Stellar Atmospheres using TLUSTY/SYNSPEC*”, North-East Meet of Astronomers 2018 (NEMA-4), Assam University, Silchar, Assam.
4. Parag Bhattacharya, Mizanur Rahman and Monmoyuri Baruah, “*Spectral Fitting of the supersoft X-ray sources RXJ0925.7-4758*”, North-East Meet of Astronomers 2017 (NEMA-3), St. Anthony’s College, Shillong, Meghalaya.
5. Parag Bhattacharya, and Monmoyuri Baruah, “*Spectral Fitting of Supersoft X-ray Sources*”, North-East Meet of Astronomers 2016 (NEMA-2), Tezpur University, Tezpur, Assam.

APPENDIX D

Python Code Listings

In this section, we provide detailed Python code listings that were developed throughout the course of the current work. These scripts were integral to the analysis and processing of the observational data, particularly for line identification, spectral fitting, lightcurve extraction, and timing analysis. The code has been meticulously documented to ensure reproducibility of the results and to facilitate further research in this area.

D.1 Line Identification Tool for XMM-Newton RGS Spectra

Filename: `setup.py`

```
1 from os import getcwd, makedirs, path, chdir, system
2 import site
3 import os
4 from shutil import copy2
5 from sys import argv
6
7 def create_folder(directory):
8     try:
9         if not os.path.exists(directory):
10             makedirs(directory)
11     except OSError:
12         print('Error :: Creating directory. ' + directory)
13
14 site_package_folder = site.getusersitepackages()
15 #site_package_folder = os.environ['HOME']+'/Desktop/site-packages'
16 data_folder = os.environ['HOME']+'/astro-data'
17 #data_folder = os.environ['HOME']+'/Desktop/astro-data'
18
19 source_folder = getcwd()
20
21 print('\nRemoving tar file...')
22 system('rm '+source_folder+'/xmmrgs_lines.tar')
23
24 req_packages = ['numpy', 'matplotlib', 'pandas', 'mendeleev', 'sqlalchemy',
25                 'astropy', 'astroquery']
```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

26 print('Upgrading existing version of pip...')
27 system('python3 -m pip install --upgrade pip')
28
29 for package in req_packages:
30     install_command = 'python3 -m pip install '+package
31     print('\nInstalling package '+package)
32     system(install_command)
33
34 print('\nCopying user site packages...')
35 target_folder = site_package_folder+'/constant'
36 create_folder(target_folder)
37 copy2(source_folder+'/electrodynamics.py', target_folder+/
       'electrodynamics.py')
38 target_folder = site_package_folder+'/astro_da'
39 create_folder(target_folder)
40 copy2(source_folder+'/xmmrgs_lines.py', target_folder+/'xmmrgs_lines.py')
41
42 print('\nCopying data files...')
43 target_folder = data_folder
44 create_folder(target_folder)
45 copy2(source_folder+'/ion.list', target_folder+/'ion.list')
46 copy2(source_folder+'/nist_5-40_gnd.csv', target_folder+/'nist_5-40_gnd.
       csv')
47
48 print('\nRemoving redundant files...')
49 system('rm '+source_folder+'/electrodynamics.py')
50 system('rm '+source_folder+'/xmmrgs_lines.py')
51 system('rm '+source_folder+'/ion.list')
52 system('rm '+source_folder+'/nist_5-40_gnd.csv')
53
54 print('\nYou may now test the Python code in the following files:\n\tline
       -overlaid-flux.py\n\tflux-inspect.py\n\tline-shift.py\nUse the flux
       files provided in the folder /fluxes. The flux file is to be used as
       an argument while running the Python code.')
55
56 os.remove(argv[0])

```

Filename: **electrodynamics.py**

```

1 from math import pi
2 epsilon_0 = 8.8541878128E-12      # permitt. of free space (C^2/Nm^2)
3 mu_0 = (4*pi)*1E-7                 # permeab. of free space (N/A^2)
4 e = 1.602176634E-19                # electronic charge (C)
5 c = 299792458                      # speed of light (m/s)
6 k_E = 1/(4*pi* epsilon_0)          # Coulomb's law const. (Nm^2/C^2)
7 k_M = 1E-7                         # Biot-Savart's law const. (N/A^2)

```

Filename: **flux-inspect.py**

```

1 from astro_da.xmmrgs_lines import *

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

2 from sys import argv
3 import os
4
5 def main():
6     flux_filename = str(argv[1])
7
8     #os.environ['ASTRODAT'] = os.environ['HOME']+'/AstroDA/data-files'
9     ,
10    os.environ['ASTRODAT'] = os.environ['HOME']+'/astro-data'
11    print('You have chosen: '+flux_filename)
12    print('Using ion list: '+os.environ['ASTRODAT']+'/ion.list')
13    print('Using NIST ground levels: '+os.environ['ASTRODAT']+'/'
14      nist_5-40_gnd.csv')
15
16    mission_name = 'XMM-Newton'
17    instrument_name = 'RGS'
18    observation_ID = '0111150101'
19    source_name = 'RX J0925.7-4758'
20
21    [angstrom, flux, error] = get_flux(flux_filename)
22    preview_flux(angstrom, flux)
23    [low_wl, high_wl] = get_wave_limits()
24    all_lines = get_lines(low_wl, high_wl)           # Comment this to
25    remove filter
26    # all_lines = get_lines(low_wl, high_wl, False) # Uncomment this
27    to remove filter
28    [angstrom, flux, error] = data_ROI(angstrom, flux, error, low_wl,
29    high_wl)
30    flux = normalize(flux)
31    plot_spectrum(angstrom, flux, error, all_lines, low_wl, high_wl,
32    True)
33
34    # ANG_A1, ANG_A2 = 12.0, 14.0
35    # ANG_B1, ANG_B2 = 14.0, 16.5
36    # ANG_C1, ANG_C2 = 16.0, 18.0
37    # ANG_D1, ANG_D2 = 18.0, 24.0
38
39    # plot_partitions(angstrom, flux, all_lines, low_wl, high_wl,
40    [12.0, 14.0, 16.5, 18.0, 24.0])
41    while input('\nDo you wish to examine any specific region? (Y/N)
42    ').upper() == 'Y':
43        [wl_i, wl_f] = obtain_ROI(low_wl, high_wl)
44        examine_ROI(angstrom, flux, error, all_lines, low_wl,
45        high_wl, wl_i, wl_f)
46
47 if __name__ == "__main__":
48     main()

```

Filename: **line-overlaid-flux.py**

```

1 from astro_da.xmmrgs_lines import *

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

2 from sys import argv
3 import os
4
5 def main():
6     flux_filename = str(argv[1])
7
8     #os.environ['ASTRODAT'] = os.environ['HOME']+ '/AstroDA/data-files'
9     ,
10    os.environ['ASTRODAT'] = os.environ['HOME']+ '/astro-data'
11    print('You have chosen: '+flux_filename)
12    print('Using ion list: '+os.environ['ASTRODAT']+ '/ion.list')
13    print('Using NIST ground levels: '+os.environ['ASTRODAT']+ '/nist_5-40_gnd.csv')
14
15    mission_name = 'XMM-Newton'
16    instrument_name = 'RGS'
17    observation_ID = '0111150101'
18    source_name = 'RX J0925.7-4758'
19
20    [angstrom, flux, error] = get_flux(flux_filename)
21    preview_flux(angstrom, flux)
22    [low_wl, high_wl] = get_wave_limits()
23    all_lines = get_lines(low_wl, high_wl)           # Comment this to
24    remove filter
25    # all_lines = get_lines(low_wl, high_wl, False) # Uncomment this
26    to remove filter
27    plot_spectrum(angstrom, flux, error, all_lines, low_wl, high_wl)
28
29 if __name__ == "__main__":
30     main()

```

Filename: line-shift.py

```

1 from astro_da.xmmrgs_lines import *
2 from sys import argv
3 import os
4
5 def main():
6     flux_filename = str(argv[1])
7
8     #os.environ['ASTRODAT'] = os.environ['HOME']+ '/AstroDA/data-files'
9     ,
10    os.environ['ASTRODAT'] = os.environ['HOME']+ '/astro-data'
11    print('You have chosen: '+flux_filename)
12    print('Using ion list: '+os.environ['ASTRODAT']+ '/ion.list')
13    print('Using NIST ground levels: '+os.environ['ASTRODAT']+ '/nist_5-40_gnd.csv')
14
15    mission_name = 'XMM-Newton'
16    instrument_name = 'RGS'
17    observation_ID = '0111150101'

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

17     source_name = 'RX J0925.7-4758'
18
19     [angstrom, flux, error] = get_flux(flux_filename)
20     preview_flux(angstrom, flux)
21     [low_wl, high_wl] = get_wave_limits()
22     [angstrom, flux, error] = data_ROI(angstrom, flux, error, low_wl,
23     high_wl)
23     all_lines = get_lines(low_wl, high_wl)           # Comment this to
24     remove filter
24
25     error = ref_normalize(error, flux)
26     flux = normalize(flux)
27
28     plot_doppler_lines('C', angstrom, flux, error, 3000.0, 'Pcyg')
29     plot_doppler_lines('N', angstrom, flux, error, 3000.0, 'Pcyg')
30     plot_doppler_lines('O', angstrom, flux, error, 3000.0, 'Pcyg')
31     plot_doppler_lines('Fe', angstrom, flux, error, 3000.0, 'Pcyg')
32
33 if __name__ == "__main__":
34     main()

```

Filename: `xmmrgs_lines.py`

```

1 class InvalidInputError(Exception):
2     # ABOUT:      A class for catching custom exceptions.
3     def __init__(self, data):
4         self.data = data
5     def __str__(self):
6         return repr(self.data)
7
8 def write_roman(num):
9     # ABOUT:      A function that returns the roman numeral
10    # equivalent
11    #           for a given integer
12    # INPUT:       An integer.
13    # OUTPUT:      A string containing the equivalent roman numerals
14    #
15    # SYNTAX:      <string> = write_roman(<integer>)
16    from collections import OrderedDict
17    roman = OrderedDict()
18    roman[1000] = "M"
19    roman[900] = "CM"
20    roman[500] = "D"
21    roman[400] = "CD"
22    roman[100] = "C"
23    roman[90] = "XC"
24    roman[50] = "L"
25    roman[40] = "XL"
26    roman[10] = "X"
27    roman[9] = "IX"
28    roman[5] = "V"

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

27     roman[4] = "IV"
28     roman[1] = "I"
29     def roman_num(num):
30         for r in roman.keys():
31             x, y = divmod(num, r)
32             yield roman[r] * x
33             num -= (r * x)
34             if num <= 0:
35                 break
36     return "".join([a for a in roman_num(num)])
37
38 def write_decimal(rom_num):
39     # ABOUT:      A function that returns the integer equivalent
40     #           for given roman numerals.
41     # INPUT:       A string containing the roman numerals.
42     # OUTPUT:      An integer containing the equivalent value.
43     # SYNTAX:     <integer> = write_decimal(<string>)
44     from collections import OrderedDict
45     num = OrderedDict()
46     num["M"] = 1000
47     num["D"] = 500
48     num["C"] = 100
49     num["L"] = 50
50     num["X"] = 10
51     num["V"] = 5
52     num["I"] = 1
53
54     result = 0
55     i = 0
56     while i < len(rom_num):
57         sym1 = num[rom_num[i]]
58         if i+1 < len(rom_num):
59             sym2 = num[rom_num[i+1]]
60             if sym1 >= sym2:
61                 result += sym1
62                 i += 1
63             else:
64                 result += (sym2 - sym1)
65                 i += 2
66         else:
67             result += sym1
68             i += 1
69     return result
70
71 def is_element(name):
72     # ABOUT:      A function that checks if a given input string
73     #           corresponds to any known element.
74     #           Returns True if element exists, and False
75     #           if it does not.
76     # INPUT:       A string containing the symbol of the proposed
#          element.
#     # OUTPUT:      A boolean value.

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

77     # SYNTAX:      <boolean> = is_element(<string>)
78     from mendeleev import element
79     from sqlalchemy.orm.exc import NoResultFound
80     try:
81         elem = element(name.capitalize())
82     except NoResultFound as e:
83         return False
84     else:
85         return True
86
87 def is_roman(rom_num):
88     # ABOUT:          A function that checks if a given input string
89     #                  corresponds to any valid roman numerals.
90     #                  Returns True if roman numerals exist, and
91     # False if it does not.
92     # INPUT:           A string containing the proposed roman numerals.
93     # OUTPUT:          A boolean value.
94     # SYNTAX:         <boolean> = is_roman(<string>)
95     roman_letters = ['I','V','X','L','C','D','M']
96     flag = True
97     for char in rom_num:
98         if not char.upper() in roman_letters:
99             flag = False
100            break
101    return flag
102
102 def get_ion(input_str):
103     # ABOUT:          A function that returns the ion stage in standard
104     # format,
105     #                  i.e. as {Symbol} {Stage (roman numerals)}
106     #                  }, for a given string.
107     #                  For example, C III, N II, O VI, Fe XVII
108     #                  The function returns a list of strings,
109     # with first element
110     #                  containing the symbol and the second
111     # element containing ion stage.
112     # INPUT:           A string containing proposed ion stage.
113     # OUTPUT:          A list of two formatted strings -- element symbol
114     # and ion stage.
115     # SYNTAX:         <list 'string'> = get_ion(<string>)
116     from mendeleev import element
117     from sqlalchemy.orm.exc import NoResultFound
118     try:
119         raw_parts = input_str.split(' ')
120         parts = []
121         for part in raw_parts:
122             if not part == "":
123                 parts.append(part)
124         if not len(parts) == 2:
125             raise InvalidInputError("Input must be in the
format <Z> <Ion Stage>.")
126     except InvalidInputError as e:

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

122         print("Invalid input:", e.data)
123         return "NULL"
124     else:
125         try:
126             parts[0] = parts[0].capitalize()
127             parts[1] = parts[1].upper()
128             if not is_element(parts[0]):
129                 raise InvalidInputError("No such element
as "+parts[0]+" exists.")
130             if not is_roman(parts[1]):
131                 raise InvalidInputError("Ion stage must
be in roman numerals.")
132         except InvalidInputError as e:
133             print("Invalid input:", e.data)
134             return ["NULL", "NULL"]
135     else:
136         return parts
137
138 def check_wave_presence(wl, low_wl, high_wl):
139     if wl >= low_wl and wl <= high_wl:
140         return True
141     else:
142         return False
143
144 def read_ions(ion_filename):
145     ion_file = open(ion_filename, 'r')
146     ions = ion_file.readlines()
147     ion_file.close()
148     for i in range(len(ions)):
149         ions[i] = ions[i].strip()
150     return ions
151
152 def get_level(conf, term, J):
153     level = conf.ljust(10, ' ')+'| '+term.ljust(5,' ')+'| '+J
154     return level
155
156 def get_nums(ion):
157     from mendeleev import element
158     ion_parts = get_ion(ion)
159     return [element(ion_parts[0]).atomic_number, write_decimal(
160     ion_parts[1])]
161
162 def retrieve_ground_index(ion, df):
163     nums = get_nums(ion)
164     rec = df[(df['at_num']==nums[0]) & (df['sp_num']==nums[1])]
165     gnd_idx = rec.loc[:, 'ground_conf'].index[0]
166     return gnd_idx
167
168 def get_ground_level(ion, df):
169     idx = retrieve_ground_index(ion, df)
170     ground_level = get_level(df['ground_conf'][idx], df['ground_term',
171     ][idx], df['ground_J'][idx])

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

170         return ground_level
171
172 def truncate_colormap(cmap, minval=0.0, maxval=1.0, n=100):
173     from matplotlib.colors import LinearSegmentedColormap
174     from numpy import linspace
175     new_cmap = LinearSegmentedColormap.from_list(
176         'trunc({n}, {a:.2f}, {b:.2f})'.format(n=cmap.name, a=
177         minval, b=maxval),
178         cmap(linspace(minval, maxval, n)))
179     return new_cmap
180
181 def get_atom_from_ion(ion):
182     atom = ion[0:2]
183     return atom.replace(' ', '')
184
185 def get_colour(ion):
186     from matplotlib.pyplot import get_cmap
187     from numpy import linspace
188     cmap_dict = {'H':'bone_r', \
189                  'He':'pink_r', \
190                  'C':'Purples', \
191                  'N':'Blues', \
192                  'O':'Greens', \
193                  'Ne':'Oranges', \
194                  'Mg':'Reds', \
195                  'Fe':'Greys'}
196     [at_num, sp_num] = get_nums(ion)
197     low_cm_val = 0.5
198     high_cm_val = 1.0
199     cm_vals = linspace(low_cm_val, high_cm_val, at_num)
200     cm = get_cmap(cmap_dict[get_atom_from_ion(ion)])
201     cm = truncate_colormap(cm, 0.5, 0.9)
202
203     return cm(cm_vals[sp_num - 1])
204
205 def show_lines(all_lines, ax, a=1.0):
206     y_min, y_max = ax.get_ylim()
207     for record in all_lines:
208         if record['Included']:
209             ax.plot([record['Wavelength'], record['Wavelength']],
210                     [y_min, y_max], color=record['Color'], lw=1, alpha=a)
211
212 def show_labeled_lines(all_lines, ax, show_wavelength, a=1.0, linewidth=1.0):
213     :
214     y_min, y_max = ax.get_ylim()
215     for record in all_lines:
216         if record['Included']:
217             ax.plot([record['Wavelength'], record['Wavelength']],
218                     [y_min, y_max], color=record['Color'], lw=linewidth, alpha=a)
219             if show_wavelength == False:
220                 ax.text(record['Wavelength'], 0.95*y_max,
221                         record['Ion'], color=record['Color'], rotation=90)

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

217             else:
218                 ax.text(record['Wavelength'], 0.85*y_max,
219                     record['Ion']+': '+%6.3f'%record['Wavelength']+'
220                     $AA$', color=
221                     record['Color'], rotation=90)
222                     ax.set_xlabel('Wavelength ($AA$)')
223                     #ax.axes.yaxis.set_visible(False)
224                     # ax.text(record['Wavelength'], 0.95*y_max,
225                     record['Ion']+': '+str(record['Wavelength'])+r'$AA$', color=record['
226                     Color'], rotation=90)
227
228
229 def show_filtered_lines(low_wl, high_wl, ion_filename, gnd_level_datafile
230 ):
231     import matplotlib.pyplot as plt
232     import pandas as pd
233     from astroquery.nist import Nist
234     import astropy.units as u
235
236     nist_gnd = pd.read_csv(gnd_level_datafile)
237
238     ground_levels = []
239     for i in range(nist_gnd.shape[0]):
240         ground_levels.append({'Ion':nist_gnd.loc[i,'sp_name'], 'Ground level':get_level(nist_gnd.loc[i,'ground_conf'], nist_gnd.loc[i,
241             'ground_term'], nist_gnd.loc[i,'ground_J'])})
242
243     ground_transitions_only = True
244
245     ions = read_ions(ion_filename)
246     ion_table = []
247
248     for ion in ions:
249         try:
250             table = Nist.query(low_wl*u.AA, high_wl*u.AA,
251             linename=ion)
252             except Exception as e:
253                 print(ion+':')
254                 print(e)
255             else:
256                 ion_table.append({'Ion':ion, 'Data':table})
257
258     all_lines = []
259
260     for record in ion_table:
261         ion = record['Ion']
262         ground_level = get_ground_level(ion, nist_gnd)
263         ion_table = record['Data']
264         NREC = len(ion_table)
265         col = get_colour(ion)
266         for i in range(NREC):
267             if ground_transitions_only:
268                 level = ion_table[i]['Lower level']
269                 level_info = level.replace(' ', '').split()

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

261 ('|')
262             level = get_level(level_info[0],
263                                     level_info[1], level_info[2])
264             if level == ground_level:
265                 if isinstance(ion_table[i]['Ritz'],
266                               float):
267                     line = {'Wavelength':
268                             ion_table[i]['Ritz'], 'Ion':ion, 'Included':True, 'Color':col}
269             else:
270                 line = {'Wavelength':
271                         float(ion_table[i]['Ritz'].replace('+','')), 'Ion':ion, 'Included':
272                         True, 'Color':col}
273             else:
274                 if isinstance(ion_table[i]['Ritz'],
275                               float):
276                     line = {'Wavelength':
277                             ion_table[i]['Ritz'], 'Ion':ion, 'Included':False, 'Color':col}
278             else:
279                 line = {'Wavelength':
280                         float(ion_table[i]['Ritz'].replace('+','')), 'Ion':ion, 'Included':
281                         False, 'Color':col}
282             else:
283                 if isinstance(ion_table[i]['Ritz'], float):
284                     line = {'Wavelength':ion_table[i]
285                             ['Ritz'], 'Ion':ion, 'Included':True, 'Color':col}
286                 else:
287                     line = {'Wavelength':float(
288                             ion_table[i]['Ritz'].replace('+','')), 'Ion':ion, 'Included':True, 'Color':col}
289             all_lines.append(line)
290
291     fig = plt.figure(figsize=(20, 3))
292     ax = fig.add_subplot(1,1,1)
293     ax.set_ylim(0, 1)
294
295     show_lines(all_lines, ax)
296     plt.show()
297
298 def get_flux(flux_filename):
299     from astropy.io import fits
300     from numpy import isnan
301     hdul = fits.open(flux_filename)
302     data = hdul[1].data
303     columns = hdul[1].columns
304
305     angstrom = []
306     flux = []
307     error = []
308     for record in data:
309         a = record[0]
310         f = record[1]
311
312         if a > 0:
313             angstrom.append(a)
314             flux.append(f)
315             error.append(error)
316
317     return angstrom, flux, error

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

299         e = record[2]
300         angstrom.append(float(a))
301         if isnan(f):
302             flux.append(0.0)
303         else:
304             flux.append(float(f))
305         if isnan(e):
306             error.append(0.0)
307         else:
308             error.append(float(e))
309     hdul.close()
310     return [angstrom, flux, error]
311
312 def preview_flux(angstrom, flux):
313     from matplotlib.pyplot import step, show, xlabel, ylabel, title,
314     text
315     # step(angstrom, flux, color = 'r', lw = 1.5)
316     step(angstrom, flux, color = 'k', lw = 1.2)
317     xlabel(r'Wavelength ($\AA$)')
318     ylabel('Flux')
319     title('Preview of spectrum')
320     text((max(angstrom)-min(angstrom))*0.7, (max(flux)-min(flux))
321           *0.5, r'Note down limits on $\lambda$', fontsize = 14, bbox=dict(
322           facecolor='red', alpha=0.2))
323     print('Displaying a preview of the spectrum.\nNote down the lower
324     and upper limits on wavelength (in angstroms)\n\tbefore closing the
325     preview plot.')
326     show()
327
328 def get_wave_limits():
329     low_wl = float(input('Enter lower wavelength (in angstroms): '))
330     high_wl = float(input('Enter upper wavelength (in angstroms): '))
331     return [low_wl, high_wl]
332
333 def get_lines(low_wl, high_wl, v_radial=0, ground_transitions_only=True):
334     import pandas as pd
335     import astropy.units as u
336     from astroquery.nist import Nist
337     from os import environ
338     from constant.electrodynamics import c
339
340     v_c = c/1E3
341
342     ion_filename = environ['ASTRODAT']+ '/ion.list'
343     gnd_level_datafile = environ['ASTRODAT']+ '/nist_5-40_gnd.csv'
344
345     nist_gnd = pd.read_csv(gnd_level_datafile)
346     ground_levels = []
347     for i in range(nist_gnd.shape[0]):
348         ground_levels.append({'Ion':nist_gnd.loc[i,'sp_name'], 'Ground level':get_level(nist_gnd.loc[i,'ground_conf'], nist_gnd.loc[i, 'ground_term'], nist_gnd.loc[i,'ground_J'])})

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

344     ions = read_ions(ion_filename)
345     ion_table = []
346     for ion in ions:
347         try:
348             table = Nist.query(low_wl*u.AA, high_wl*u.AA,
349             linename=ion)
350         except Exception as e:
351             print(ion+' : '+str(e))
352         else:
353             ion_table.append({'Ion':ion, 'Data':table})
354
355     all_lines = []
356     for record in ion_table:
357         ion = record['Ion']
358         ground_level = get_ground_level(ion, nist_gnd)
359         ion_table = record['Data']
360         NREC = len(ion_table)
361         col = get_colour(ion)
362         for i in range(NREC):
363             if ground_transitions_only:
364                 level = ion_table[i]['Lower level']
365                 level_info = level.replace(' ', '').split
366                 ('|')
367                 level = get_level(level_info[0],
368                 level_info[1], level_info[2])
369                 if level == ground_level:
370                     if isinstance(ion_table[i]['Ritz'],
371                     float):
372                         line = {'Wavelength':
373                             ion_table[i]['Ritz']*(1-(v_radial/v_c)), 'Ion':ion,
374                             'Included':True, 'Color':col}
375                     else:
376                         line = {'Wavelength':
377                             float(ion_table[i]['Ritz'].replace('+',''))*(1-(v_radial/v_c)), 'Ion':
378                             ion, 'Included':True, 'Color':col}
379                     else:
380                         if isinstance(ion_table[i]['Ritz'],
381                         float):
382                             line = {'Wavelength':
383                             ion_table[i]['Ritz']*(1-(v_radial/v_c)), 'Ion':ion,
384                             'Included':False, 'Color':col}
385                     else:
386                         if isinstance(ion_table[i]['Ritz'], float
387                             ):
388                             line = {'Wavelength':ion_table[i]
389                             ['Ritz']*(1-(v_radial/v_c)), 'Ion':ion,
390                             'Included':True, 'Color':col}
391                         else:

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

381             line = {'Wavelength':float(
382                 ion_table[i]['Ritz'].replace('+',''))*(1-(v_radial/v_c)), 'Ion':ion, 'Included':True, 'Color':col}
383             all_lines.append(line)
384         return all_lines
385
385 def find_flux_limits(angstrom, flux, low_wl, high_wl):
386     subset = []
387     for i in range(len(angstrom)):
388         if check_wave_presence(angstrom[i], low_wl, high_wl):
389             subset.append(flux[i])
390     return [min(subset), max(subset)]
391
392 def data_ROI(angstrom, flux, error, low_wl, high_wl):
393     a = []
394     f = []
395     e = []
396     for i in range(len(angstrom)):
397         if angstrom[i] >= low_wl and angstrom[i] <= high_wl:
398             a.append(angstrom[i])
399             f.append(flux[i])
400             e.append(error[i])
401     return [a, f, e]
402
403 def normalize(data):
404     min_val = min(data)
405     max_val = max(data)
406     for i in range(len(data)):
407         data[i] = (data[i]-min_val)/(max_val-min_val)
408     return data
409
410 def ref_normalize(data, reference):
411     min_val = min(reference)
412     max_val = max(reference)
413     for i in range(len(data)):
414         data[i] = (data[i])/(max_val-min_val)
415         # data[i] = abs(data[i]/reference[i])
416     return data
417
418 def wavelength_partition(low_wl, high_wl, npart=4):
419     parts = []
420     delta_wl = (high_wl - low_wl)/npart
421     wl = low_wl
422     for i in range(npart):
423         part_low = wl
424         part_high = wl + delta_wl
425         parts.append([part_low, part_high])
426         wl = part_high
427     return parts
428
429 def wavelength_partition_2(low_wl, high_wl, markers):
430     try:

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

431         markers.sort()
432         if low_wl > markers[0] or high_wl < markers[len(markers)
433             -1]:
434             raise Exception('The wavelength markers should
435             lie within the wavelength range chosen')
436             npart = len(markers)-1
437             parts = []
438             wl = markers[0]
439             for i in range(npart):
440                 part_low = wl
441                 part_high = markers[i+1]
442                 parts.append([part_low, part_high])
443                 wl = markers[i+1]
444             return parts
445         except Exception as e:
446             print('Error: '+str(e))
447
448 def error_bars(angstrom, flux, error, ax):
449     for i in range(len(angstrom)):
450         ax.plot([angstrom[i], angstrom[i]], [flux[i]-error[i]
451             ]*0.5, flux[i]+error[i]*0.5, lw=1.2, color='k', alpha=0.8)
452
453 def show_line_overlaid_spec(angstrom, flux, all_lines, low_wl, high_wl,
454     min_flux, max_flux, ax, spec_width=1.0, spec_col="#2ed006", lwidth
455     =1.0):
456     ax.set_xlim(low_wl, high_wl)
457     ax.set_ylim(min_flux, max_flux)
458     ax.step(angstrom, flux, color=spec_col, lw=spec_width)
459     # show_lines(all_lines, ax, 0.6)
460     show_labeled_lines(all_lines, ax, False, 0.8, lwidth)
461     ax.set_xlabel('Wavelength ($\AA$)')
462     ax.set_ylabel('Normalized flux (s$^{-1}$ cm$^{-2}$\AA$^{-1}$)')
463
464 def show_line_detail(angstrom, flux, error, all_lines, low_wl, high_wl,
465     min_flux, max_flux, ax, spec_width=1.0, spec_col="#2ed006", lwidth
466     =1.0):
467     ax.set_xlim(low_wl, high_wl)
468     ax.set_ylim(min_flux-min(error), max_flux+max(error))
469     #ax.plot(angstrom, flux, 'o-', color=spec_col, lw=spec_width)
470     ax.step(angstrom, flux, color=spec_col, lw=spec_width)
471     #error_bars(angstrom, flux, error, ax)
472     # show_lines(all_lines, ax, 0.6)
473     show_labeled_lines(all_lines, ax, 0.8, lwidth)
474     ax.set_xlabel('Wavelength ($\AA$)')
475     ax.set_ylabel('Normalized flux (s$^{-1}$ cm$^{-2}$\AA$^{-1}$)')
476
477 def plot_spectrum(angstrom, flux, error, all_lines, low_wl, high_wl,
478     show_wavelength=False):
479     import matplotlib.pyplot as plt
480     fig = plt.figure(figsize=(20, 12))
481     ax_spec = fig.add_subplot(3,1,1)

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

475
476     ax_lines = fig.add_subplot(3,1,2)
477     ax_overlay = fig.add_subplot(3,1,3)
478
479     ax_spec.step(angstrom, flux, color='#2ed006', lw=1.0)
480     [min_flux, max_flux] = find_flux_limits(angstrom, flux, low_wl,
481     high_wl)
482     ax_spec.set_xlim(low_wl, high_wl)
483     ax_spec.set_ylim(min_flux, max_flux)
484     ax_spec.set_xlabel('Wavelength ($\AA$)')
485     ax_spec.set_ylabel('Normalized flux (s$^{-1}$ cm$^{-2}$\AA$^{-1}$)')
486
487
488     ax_lines.set_ylim(0, 1)
489     ax_lines.set_xlim(low_wl, high_wl)
490     ax_lines.axes.yaxis.set_visible(False)
491     # show_lines(all_lines, ax_lines)
492     show_labeled_lines(all_lines, ax_lines, show_wavelength)
493
494
495 def plot_partitions(angstrom, flux, all_lines, low_wl, high_wl, markers):
496     import matplotlib.pyplot as plt
497     parts = wavelength_partition_2(low_wl, high_wl, markers)
498     nparts = len(parts)
499     fig = plt.figure(figsize=(20,14))
500     axes = []
501     for i in range(nparts):
502         axes.append(fig.add_subplot(nparts,1,i+1))
503
504     for k in range(len(axes)):
505         ang = []
506         flx = []
507         for i in range(len(angstrom)):
508             if check_wave_presence(angstrom[i], parts[k][0],
509             parts[k][1]):
510                 ang.append(angstrom[i])
511                 flx.append(flux[i])
512             [min_flux, max_flux] = find_flux_limits(ang, flx, parts[k][0],
513             parts[k][1])
514             show_line_overlaid_spec(ang, flx, all_lines, parts[k][0],
515             parts[k][1], min_flux, max_flux, axes[k], 1.5, 'k', 1.5)
516
517     plt.show()
518
519 def bandpass_spec(angstrom, flux, error, wl_i, wl_f):
520     ang = []
521     flx = []
522     err = []
523     for i in range(len(angstrom)):

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

521             if check_wave_presence(angstrom[i], wl_i, wl_f):
522                 ang.append(angstrom[i])
523                 flx.append(flx[i])
524                 err.append(error[i])
525         return [ang, flx, err]
526
527 def bandpass_lines(all_lines, wl_i, wl_f):
528     band_lines = []
529     for line in all_lines:
530         if check_wave_presence(line['Wavelength'], wl_i, wl_f):
531             band_lines.append(line)
532     return band_lines
533
534 def obtain_ROI(low_wl, high_wl):
535     ROI_acceptable = False
536     while not ROI_acceptable:
537         wl_i = float(input('Enter lower wavelength (in angstrom')
538         for region of interest: '))
539         wl_f = float(input('Enter upper wavelength (in angstrom')
540         for region of interest: '))
541         try:
542             if wl_i >= wl_f:
543                 raise ValueError('You have entered lower
wavelength to be greater than upper wavelength.')
544             if not check_wave_presence(wl_i, low_wl, high_wl)
545             and not check_wave_presence(wl_f, low_wl, high_wl):
546                 raise ValueError('You have entered
wavelengths that are outside acceptable range of ['+str('%4.1f'%low_wl)
547             +', '+str('%4.1f'%high_wl)+'] angstrom.')
548             if not check_wave_presence(wl_i, low_wl, high_wl):
549                 raise ValueError('You have entered lower
wavelength that is outside acceptable range of ['+str('%4.1f'%low_wl)+'
550             ', '+str('%4.1f'%high_wl)+'] angstrom.')
551             if not check_wave_presence(wl_f, low_wl, high_wl):
552                 raise ValueError('You have entered upper
wavelength that is outside acceptable range of ['+str('%4.1f'%low_wl)+'
553             ', '+str('%4.1f'%high_wl)+'] angstrom.')
554         except ValueError as e:
555             print('Error: '+str(e))
556             print('\nEnter again... ')
557             continue
558         else:
559             ROI_acceptable = True
560     return [wl_i, wl_f]
561
562 def examine_ROI(angstrom, flux, error, all_lines, low_wl, high_wl, wl_i,
563 wl_f):
564     import matplotlib.pyplot as plt
565     spec = bandpass_spec(angstrom, flux, error, wl_i, wl_f)
566     lines = bandpass_lines(all_lines, wl_i, wl_f)

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

560
561     fig = plt.figure(figsize=(20,4))
562     ax = fig.add_subplot(1,1,1)
563
564     # [min_flux, max_flux] = find_flux_limits(spec[0], spec[1], wl_i,
565     wl_f)
566     show_line_detail(spec[0], spec[1], spec[2], lines, wl_i, wl_f,
567     min(spec[1]), max(spec[1]), ax, 1.5)
568     fig.tight_layout(pad = 1.0)
569     plt.show()
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

586                     return record['angstrom']
587
588 def get_lyman_wave_limits(lmda_0, v_lim=5000.0):
589     lmda_min = velocity_to_waveshift(lmda_0, -v_lim)
590     lmda_max = velocity_to_waveshift(lmda_0, v_lim)
591     return [lmda_min, lmda_max]
592
593 def map_spec_to_vel(lmda_0, angstrom, flux, error, v_lim=5000.0):
594     wave_limit = get_lyman_wave_limits(lmda_0, v_lim)
595     a = []
596     f = []
597     e = []
598     v = []
599     for i in range(len(angstrom)):
600         if angstrom[i] >= wave_limit[0] and angstrom[i] <=
601             wave_limit[1]:
602                 v.append(waveshift_to_velocity(lmda_0, angstrom[i])
603                           )
604                 a.append(angstrom[i])
605                 f.append(flux[i])
606                 e.append(error[i])
607     return [v, a, f, e]
608
609 def plot_doppler_lines(element, angstrom, flux, error, v_lim=5000.0, mode
610 = 'abs'):
611     import matplotlib.pyplot as plt
612     # flux = normalize(flux)
613     ion_species = [{element:'C', 'sp_1':'V', 'sp_2':'VI'}, \
614                     {'element':'N', 'sp_1':'VI', 'sp_2':'VII'}, \
615                     {'element':'O', 'sp_1':'VII', 'sp_2':'VIII'}, \
616                     {'element':'Fe', 'sp_1':'XVII', 'sp_2':'XVIII'}]
617     transitions = ['alpha', 'beta', 'gamma']
618     trans_symbol = {'alpha':r'$\alpha$', 'beta':r'$\beta$', 'gamma':r'$\gamma$'}
619     trans_color = {'alpha':'r', 'beta':'g', 'gamma':'b'}
620     plot_series = []
621     for member in ion_species:
622         if member['element'] == element:
623             plot_series.append(member['element']+member['sp_1'])
624             plot_series.append(member['element']+member['sp_2'])
625
626         num_rows = len(transitions)
627         num_cols = len(plot_series)
628         num_plots = num_rows*num_cols
629         axis = []
630
631         fig = plt.figure(figsize=(12,12))

```

D.1. Line Identification Tool for XMM-Newton RGS Spectra

```

629     k = 1
630     for i in range(num_rows):
631         for j in range(num_cols):
632             axis.append({'ion':plot_series[j], 'transition':
transitions[i], 'axis':fig.add_subplot(num_rows,num_cols,k)})
633             k += 1
634
635     print('\nDisplaying '+element+' lines...')
636     for ax in axis:
637         lmda_0 = get_lyman_line(ax['ion'], ax['transition'])
638         [v, a, f, e] = map_spec_to_vel(lmda_0, angstrom, flux,
error, v_lim)
639         ax['axis'].axvline(linewidth=1.5, color='gray', dashes
=(5, 2))
640         if f:
641             if mode == 'abs':
642                 ax['axis'].axvline(v[f.index(min(f))], 0,
linewidth=1.5, color='magenta', dashes=(5, 2))
643                 ax['axis'].text(v[f.index(min(f))]+50,
1.0*max(f), '%.1f'%v[f.index(min(f))]+'' km/s'', color='magenta',
rotation=90)
644                 ax['axis'].text(v[f.index(min(f))]-300,
1.0*max(f), '%.4f'%(a[f.index(min(f))]-lmda_0)+'' $\AA$'', color='
magenta', rotation=90)
645             elif mode == 'ems':
646                 ax['axis'].axvline(v[f.index(max(f))], 0,
linewidth=1.5, color='forestgreen', dashes=(5, 2))
647                 ax['axis'].text(v[f.index(max(f))]+50,
0.6*max(f), '%.1f'%v[f.index(min(f))]+'' km/s'', color='forestgreen',
rotation=90)
648                 ax['axis'].text(v[f.index(max(f))]-300,
0.5*max(f), '%.4f'%(a[f.index(min(f))]-lmda_0)+'' $\AA$'', color='
forestgreen', rotation=90)
649             elif mode == 'Pcyg':
650                 ax['axis'].axvline(v[f.index(min(f))], 0,
linewidth=1.5, color='magenta', dashes=(5, 2))
651                 ax['axis'].text(v[f.index(min(f))]+50,
1.0*max(f), '%.1f'%v[f.index(min(f))]+'' km/s'', color='magenta',
rotation=90)
652                 ax['axis'].text(v[f.index(min(f))]-300,
1.0*max(f), '%.4f'%(a[f.index(min(f))]-lmda_0)+'' $\AA$'', color='
magenta', rotation=90)
653                 ax['axis'].axvline(v[f.index(max(f))], 0,
linewidth=1.5, color='forestgreen', dashes=(5, 2))
654                 ax['axis'].text(v[f.index(max(f))]+50,
0.6*max(f), '%.1f'%v[f.index(min(f))]+'' km/s'', color='forestgreen',
rotation=90)
655                 ax['axis'].text(v[f.index(max(f))]-300,
0.5*max(f), '%.4f'%(a[f.index(min(f))]-lmda_0)+'' $\AA$'', color='
forestgreen', rotation=90)
656
657             ax['axis'].plot(v, f, 'o-', color=trans_color[ax['

```

D.2. Multi-Observatory Spectral Analysis

```
658     transition)], lw=1.5)
659         for i in range(len(v)):
660             ax['axis'].plot([v[i], v[i]], [f[i]-e[i]*0.5, f[i]
661                             +e[i]*0.5], lw=1.2, color='k', alpha=0.8)
662             if get_atom_from_ion(ax['ion'])=='Fe':
663                 ax['axis'].set_title(ax['ion']+ ' at +'+'%7.4f'%lmda_0+r' $\AA$')
664             else:
665                 ax['axis'].set_title(ax['ion']+ ' Ly'+trans_symbol
666 [ax['transition']]+' at +'+'%7.4f'%lmda_0+r' $\AA$')
667             ax['axis'].set_xlim(-v_lim, v_lim)
668             ax['axis'].set_xlabel('Radial velocity (km/s)')
669             ax['axis'].set_ylabel('Normalized flux (s$^{-1}$ cm$^
670 ^{-2} \AA^{-1}$)')
671
672     fig.tight_layout(pad = 2.5)
673     plt.show()
```

D.2 Multi-Observatory Spectral Analysis

D.2.1 Observed Count Rates

```
1 ## This cell is required while on Google Colab
2 from google.colab import drive
3 drive.mount('/content/drive')

1 ## This cell is required while on Google Colab
2 import sys
3 from glob import glob
4 sys.path.append('/content/drive/MyDrive/analysis')
5 from plot_helpers import *
6 analysis_path = '/content/drive/MyDrive/analysis/'
7 data_path = '/content/drive/MyDrive/analysis/plots-data/'

1 ldata_files = sorted(glob(data_path+'*ldata*.qdp')) # To be modified
2           while on Google Colab
2 ldata_files

1 output_folder = analysis_path+'outputs/' # To be modified while on Google
2           Colab
2 pdf_folder = 'pdf'
3 png_folder = 'png'
4 try:
5     os.mkdir(output_folder)
6 except FileExistsError as error:
7     print(f"The folder 'outputs' already exists.")
8 try:
9     os.mkdir(os.path.join(output_folder, pdf_folder)) # To be modified
2           while on Google Colab
```

D.2. Multi-Observatory Spectral Analysis

```
10 except FileExistsError as error:
11     print(f"The folder '{pdf_folder}' already exists.")
12 try:
13     os.mkdir(os.path.join(output_folder, png_folder)) # To be modified
14     while on Google Colab
15 except FileExistsError as error:
16     print(f"The folder '{png_folder}' already exists.")

1 ## This cell is required while on Google Colab
2 mpl.rcParams['pdf.fonttype'] = 42
3 mpl.rcParams['ps.fonttype'] = 42
4 mpl.rcParams['font.family'] = 'Liberation Sans'

1 energy_lo = 0.2
2 energy_hi = 1.0
3 for file in ldata_files:
4     data = np.loadtxt(file, skiprows=3)
5     plot_ldata(data,
6                 os.path.basename(file),
7                 (os.path.join(output_folder, pdf_folder), os.path.join(
8                     output_folder, png_folder)), # To be modified while on Google Colab
9                 energy_lo=energy_lo, energy_hi=energy_hi)

1 plot_colors = ['#c9184a', '#007200', '#f46036', '#1d3557', '#7209b7', '
2     #1982c4']
3 obs_dates = {'SIS1-43036000': '1994-12-22',
4             'ACIS-644': '2000-11-14',
5             'PN-0111150101': '2000-12-16',
6             'XTI-2611020101': '2019-05-18',
7             'XTI-2611020102': '2019-05-19',
8             'XTI-2611020103': '2019-05-19'}
```

D.2. Multi-Observatory Spectral Analysis

Count rates from all observatories

```
1 group_plot = group_plots_labels(ldata_files, obs_dates)
2 sort_order = list(obs_dates.keys())
3
4 energy_lo = 0.2
5 energy_hi = 1.0
6
7 fig = plt.figure(figsize=(6,3), dpi=100)
8 ax = fig.add_axes((0.1,0.1,0.9,0.9))
9
10 for val in sort_order:
11     data = np.loadtxt(group_plot[val][0], skiprows=3)
12     energy = data[:,0]
13     rate = data[:,2]
14     ax.step(energy, rate, c=plot_colors[sort_order.index(val)],
15             where='mid', zorder=sort_order.index(val), label=group_plot[
16               val][1])
17
18 ax.set_yscale('log')
19 ax.set_xlabel(r'Energy (keV)', fontsize=14)
20 ax.set_ylabel(r'counts s$^{-1}$ keV$^{-1}$', fontsize=14)
21 ax.set_xlim(energy_lo, energy_hi)
22
23 ax.xaxis.set_major_locator(MultipleLocator(0.1))
24 ax.xaxis.set_minor_locator(MultipleLocator(0.01))
25 ax.tick_params(axis='x', which='major', direction='in',
26                 length=6, width=1.5, color='k', pad=6,
27                 labelsize=12, labelcolor='k')
28 ax.tick_params(axis='x', which='minor', direction='in',
29                 length=3, width=1.0, color='k', pad=6,
30                 labelsize=12, labelcolor='k')
31
32 ax.yaxis.set_major_locator(LogLocator(base=10))
33 ax.yaxis.set_minor_locator(LogLocator(base=10, subs
34             =(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9), numticks=12))
35 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
36 ax.tick_params(axis='y', which='major', direction='in',
37                 length=6, width=1.5, color='k', pad=6,
38                 labelsize=12, labelcolor='k')
39 ax.tick_params(axis='y', which='minor', direction='in',
40                 length=3, width=1.0, color='k', pad=6,
41                 labelsize=12, labelcolor='k')
42
43 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
44 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
45
46 op_filename = 'mr-vel-counts_all-obs'
47 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
48             bbox_inches='tight') # To be modified while on Google Colab
```

D.2. Multi-Observatory Spectral Analysis

```
48 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
    bbox_inches='tight') # To be modified while on Google Colab
```

Normalized count rates from all observatories

```
1 group_plot = group_plots_labels(ldata_files, obs_dates)
2 sort_order = list(obs_dates.keys())
3
4 energy_lo = 0.2
5 energy_hi = 1.0
6
7 fig = plt.figure(figsize=(6,3), dpi=100)
8 ax = fig.add_axes((0.1,0.1,0.9,0.9))
9
10 for val in sort_order:
11     data = np.loadtxt(group_plot[val][0], skiprows=3)
12     energy = data[:,0]
13     rate = data[:,2]
14     rate_normalized = (rate - min(rate)) / (max(rate) - min(rate))
15     ax.step(energy, rate_normalized, c=plot_colors[sort_order.index(val)],
16             where='mid', zorder=sort_order.index(val), label=group_plot[
17               val][1])
18
19 ax.set_yscale('log')
20 ax.set_xlabel(r'Energy (keV)', fontsize=14)
21 ax.set_ylabel(r'Normalized counts s$^{-1}$ keV$^{-1}$', fontsize=14)
22 ax.set_xlim(energy_lo, energy_hi)
23
24 ax.xaxis.set_major_locator(MultipleLocator(0.1))
25 ax.xaxis.set_minor_locator(MultipleLocator(0.01))
26 ax.tick_params(axis='x', which='major', direction='in',
27                 length=6, width=1.5, color='k', pad=6,
28                 labelsize=12, labelcolor='k')
29 ax.tick_params(axis='x', which='minor', direction='in',
30                 length=3, width=1.0, color='k', pad=6,
31                 labelsize=12, labelcolor='k')
32
33 ax.yaxis.set_major_locator(LogLocator(base=10))
34 ax.yaxis.set_minor_locator(LogLocator(base=10, subs
35             =(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9), numticks=12))
36 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
37 ax.tick_params(axis='y', which='major', direction='in',
38                 length=6, width=1.5, color='k', pad=6,
39                 labelsize=12, labelcolor='k')
40 ax.tick_params(axis='y', which='minor', direction='in',
41                 length=3, width=1.0, color='k', pad=6,
42                 labelsize=12, labelcolor='k')
43 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
44 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
```

D.2. Multi-Observatory Spectral Analysis

```
44 ax.legend(loc='lower right')
45
46 op_filename = 'mr-vel-normcounts_all-obs'
47
48 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
49             bbox_inches='tight') # To be modified while on Google Colab
50 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
51             bbox_inches='tight') # To be modified while on Google Colab
```

D.2.2 Unfolded Spectra

```
1 ## This cell is required while on Google Colab
2 from google.colab import drive
3 drive.mount('/content/drive')

1 ## This cell is required while on Google Colab
2 import sys
3 from glob import glob
4 sys.path.append('/content/drive/MyDrive/analysis')
5 from plot_helpers import *
6
7 analysis_path = '/content/drive/MyDrive/analysis/'
8 data_path = '/content/drive/MyDrive/analysis/plots-data/'

1 output_folder = analysis_path+'outputs/' # To be modified while on Google
    Colab
2 pdf_folder = 'pdf'
3 png_folder = 'png'
4 try:
5     os.mkdir(output_folder)
6 except FileExistsError as error:
7     print(f"The folder '{output_folder}' already exists.")
8 try:
9     os.mkdir(os.path.join(output_folder, pdf_folder)) # To be modified
    while on Google Colab
10 except FileExistsError as error:
11     print(f"The folder '{pdf_folder}' already exists.")
12 try:
13     os.mkdir(os.path.join(output_folder, png_folder)) # To be modified
    while on Google Colab
14 except FileExistsError as error:
15     print(f"The folder '{png_folder}' already exists.")

1 ## This cell is required while on Google Colab
2 mpl.rcParams['pdf.fonttype'] = 42
3 mpl.rcParams['ps.fonttype'] = 42
4 mpl.rcParams['font.family'] = 'Liberation Sans'

1 eufspec_files = sorted(glob(data_path+'*eufspec.qdp')) # To be modified
    while on Google Colab
2 eufspec_files
```

D.2. Multi-Observatory Spectral Analysis

```

1 wave_lo = 12.0
2 wave_hi = 42.0
3 for file in eufspec_files:
4     data = np.loadtxt(file, skiprows=3)
5     plot_eufspec(data,
6                     os.path.basename(file),
7                     (os.path.join(output_folder, pdf_folder), os.path.join(
8                         output_folder, png_folder)), # To be modified while on Google Colab
9                     wave_lo=wave_lo, wave_hi=wave_hi)

1 plot_colors = ['#c9184a', '#007200', '#f46036', '#1d3557', '#7209b7', '#1982c4']
2 obs_dates = {'SIS1-43036000': '1994-12-22',
3               'ACIS-644': '2000-11-14',
4               'PN-0111150101': '2000-12-16',
5               'XTI-2611020101': '2019-05-18',
6               'XTI-2611020102': '2019-05-19',
7               'XTI-2611020103': '2019-05-19'}

1 group_plot = group_plots_labels(eufspec_files, obs_dates)
2 sort_order = list(obs_dates.keys())
3
4 fig = plt.figure(figsize=(6,5), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 for val in sort_order:
8     data = np.loadtxt(group_plot[val][0], skiprows=3)
9     wavelength = data[:,0]
10    eufspec = data[:,4]
11    ax.step(wavelength, eufspec, c=plot_colors[sort_order.index(val)],
12             where='mid', zorder=sort_order.index(val), label=group_plot[
13                 val][1])
14
15 ax.set_yscale('log')
16 ax.set_xlabel(r'Wavelength $\lambda$ ($\AA$)', fontsize=14)
17 ax.set_ylabel(r'Flux density $F_\lambda$ (erg s$^{-1}$ cm$^{-2}$ $\AA$  
$^{-1}$)', fontsize=14)
18 #ax.set_ylim(1E-22, 1E-12)
19
20 ax.xaxis.set_major_locator(MultipleLocator(2.0))
21 ax.xaxis.set_minor_locator(MultipleLocator(1.0))
22 ax.tick_params(axis='x', which='major', direction='in',
23                 length=6, width=1.5, color='k', pad=6,
24                 labelsize=12, labelcolor='k')
25 ax.tick_params(axis='x', which='minor', direction='in',
26                 length=3, width=1.0, color='k', pad=6,
27                 labelsize=12, labelcolor='k')
28
29 ax.yaxis.set_major_locator(LogLocator(base=10))
30 ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2,0.4,0.6,0.8),
31                           numticks=12))

```

D.2. Multi-Observatory Spectral Analysis

```

31 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
32 ax.tick_params(axis='y', which='major', direction='in',
33                 length=6, width=1.5, color='k', pad=6,
34                 labelsize=12, labelcolor='k')
35 ax.tick_params(axis='y', which='minor', direction='in',
36                 length=3, width=1.0, color='k', pad=6,
37                 labelsize=12, labelcolor='k')
38
39 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
40 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
41
42 ax.legend(loc='lower left')
43
44 op_filename = 'mr-vel-uf_all-obs'
45 #op_filename = 'mr-vel-uf_all-obs_12-24'
46 #op_filename = 'mr-vel-uf_all-obs_24-36'
47
48 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
49             bbox_inches='tight') # To be modified while on Google Colab
50 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
51             bbox_inches='tight') # To be modified while on Google Colab

```

D.2.3 Luminosity versus Effective Temperature

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import astropy as ast
4 from astropy import units as u
5 from astropy import constants as const

1 from decimal import Decimal
2
3 def fexp(number):
4     (sign, digits, exponent) = Decimal(number).as_tuple()
5     return len(digits) + exponent - 1
6
7 def fman(number):
8     return Decimal(number).scaleb(-fexp(number)).normalize()

1 nH_bb = [1.4E22, 3.7E22]
2 nH_optical = [1.0E22, 1.9E22]
3 R = 425 * u.parsec
4
5 flux_data = np.loadtxt('fluxes.dat', skiprows=1)
6 log10Flux = flux_data[0:,1:]
7 Flux = np.power(10, log10Flux) * u.erg / (u.s * (u.cm)**2)
8 Luminosity = (4 * np.pi * R**2 * Flux).to(u.erg / u.s)

1 pows = []
2 for L_list in Luminosity:
3     pows.append(fexp(L_list[3].value))

```

D.2. Multi-Observatory Spectral Analysis

```

4 common_pow = min(pows)
5
6 L = []
7 L_errmin, L_errmax = [], []
8 for L_list in Luminosity:
9     L_val = L_list[3].value / (10**common_pow)
10    del_L_lo = L_val - L_list[4].value / (10**common_pow)
11    del_L_hi = L_list[5].value / (10**common_pow) - L_val
12    L.append(L_val)
13    L_errmin.append(del_L_lo)
14    L_errmax.append(del_L_hi)
15 L_err = [L_errmin, L_errmax]

1 Teff_data = np.loadtxt('teff.dat', skiprows=1)[:,1:]
2 T = Teff_data[:,0]
3 T_errmin, T_errmax = Teff_data[:,1], Teff_data[:,2]
4 T_errmin = T - Teff_data[:,1]
5 T_errmax = Teff_data[:,2] - T
6 T_err = [T_errmin, T_errmax]

1 import matplotlib as mpl
2 from matplotlib.ticker import MultipleLocator, LogLocator,
3     FormatStrFormatter, MaxNLocator, FuncFormatter
4
5 mpl.rcParams['pdf.fonttype'] = 42
6 mpl.rcParams['ps.fonttype'] = 42
7 mpl.rcParams['font.family'] = 'Liberation Sans'
8
9 def log_formatter(x, pos):
10     if x == 1:
11         return '1'
12     else:
13         return f'$10^{{{{int(np.log10(x))}}}}$'

1 plot_colors = ['#c9184a', '#007200', '#f46036', '#1d3557', '#7209b7', '#1982c4']
2 plot_markers = ["o", "^", "s", "*", "x", "d"]
3 plot_markersize = [150, 150, 150, 180, 150, 150]
4 plot_labels = ["1994-12-22 | ASCA:SIS1",
5                 "2000-11-14 | Chandra:ACIS",
6                 "2000-12-16 | XMM:EPIC-pn",
7                 "2019-05-18 | NICER:XTI",
8                 "2019-05-19 | NICER:XTI",
9                 "2019-05-19 | NICER:XTI"]
10
11 fig = plt.figure(figsize=(6,4), dpi=100)
12 ax = fig.add_axes((0.1,0.1,0.9,0.9))
13
14 for i in range(len(T)):
15     ax.errorbar(T[i], L[i], xerr = [np.array([T_errmin[i]]), np.array([
16         T_errmax[i]])], yerr = [np.array([L_errmin[i]]), np.array([L_errmax[i]])], fmt='.', c=plot_colors[i], alpha=0.8)

```

D.2. Multi-Observatory Spectral Analysis

```
16     ax.scatter(T[i], L[i], c=plot_colors[i], marker=plot_markers[i], s=
17         plot_markersize[i], label=plot_labels[i])
18 ax.set_yscale('log')
19 ax.set_xlim(150, 50)
20 ax.set_ylim(1E0,1E3)
21 ax.set_xlabel(r"$T_{\mathrm{eff}}$ (kK)", fontsize=14)
22 ax.set_ylabel(r"$L_{*}$ ($\times 10^{f \{\mathrm{common\_pow}\} +}$) erg s$^{-1}$)", 
23     fontsize=14)
24
25 ax.xaxis.set_major_locator(MultipleLocator(10))
26 ax.xaxis.set_minor_locator(MultipleLocator(2))
27 ax.tick_params(axis='x',which='major',direction='in', length=6, width=1.5,
28     color='k', pad=6, labelsize=12, labelcolor='k')
29 ax.tick_params(axis='x',which='minor',direction='in', length=3, width=1.0,
30     color='k', pad=6, labelsize=12, labelcolor='k')
31
32 ax.yaxis.set_major_locator(LogLocator(base=10))
33 ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2,0.4,0.6,0.8,1.0),
34     numticks=12))
35 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
36 ax.tick_params(axis='y',which='major',direction='in', length=6, width=1.5,
37     color='k', pad=6, labelsize=12, labelcolor='k')
38 ax.tick_params(axis='y',which='minor',direction='in', length=3, width=1.0,
39     color='k', pad=6, labelsize=12, labelcolor='k')
40 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
41 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
42
43 ax.legend(loc="lower right")
44
45 plt.savefig('L-Teff_all-obs.pdf', bbox_inches='tight')
46 plt.savefig('L-Teff_all-obs.png', bbox_inches='tight')
```

D.2.4 Identification of Elemental Absorption Edges

```
1 ## This cell is required while on Google Colab
2 from google.colab import drive
3 drive.mount('/content/drive')

1 ## This cell is required while on Google Colab
2 import sys
3 from glob import glob
4 sys.path.append('/content/drive/MyDrive/analysis')
5 from plot_helpers import *
6
7 analysis_path = '/content/drive/MyDrive/analysis/'
8 data_path = '/content/drive/MyDrive/analysis/plots-data/'

1 output_folder = analysis_path+'outputs/' # To be modified while on Google
2                               Colab
3 pdf_folder = 'pdf'
4 png_folder = 'png'
```

D.2. Multi-Observatory Spectral Analysis

```

4 try:
5     os.mkdir(output_folder)
6 except FileExistsError as error:
7     print(f"The folder 'outputs' already exists.")
8 try:
9     os.mkdir(os.path.join(output_folder, pdf_folder)) # To be modified
10    while on Google Colab
11 except FileExistsError as error:
12     print(f"The folder '{pdf_folder}' already exists.")
13 try:
14     os.mkdir(os.path.join(output_folder, png_folder)) # To be modified
15    while on Google Colab
16 except FileExistsError as error:
17     print(f"The folder '{png_folder}' already exists.")

1 ## This cell is required while on Google Colab
2 mpl.rcParams['pdf.fonttype'] = 42
3 mpl.rcParams['ps.fonttype'] = 42
4 mpl.rcParams['font.family'] = 'Liberation Sans'

1 url = 'https://docs.google.com/spreadsheets/d/1iWk96c9yzq_7e-
      nQG7Mu87Ist8Ye6lBVbAiTzjW2GIg/export?format=csv'
2 df = pd.read_csv(url)
3 edgedf_keV, edgedf_ang = get_edgedf(df)

1 eufspec_files = sorted(glob(data_path+'*eufspec-edges.qdp')) # To be
      modified while on Google Colab
2 eufspec_keV_files = sorted(glob(data_path+'*eufspec-keV.qdp')) # To be
      modified while on Google Colab

1 plot_colors = ['#c9184a', '#007200', '#f46036', '#1d3557', '#7209b7', '#1982c4']
2 obs_dates = {'SIS1-43036000': '1994-12-22',
3              'ACIS-644': '2000-11-14',
4              'PN-0111150101': '2000-12-16',
5              'XTI-2611020101': '2019-05-18',
6              'XTI-2611020102': '2019-05-19',
7              'XTI-2611020103': '2019-05-19'}
8 sort_order = list(obs_dates.keys())
9
10 elements = ['N', 'O', 'Fe', 'Ne']

1 energy_lo, energy_hi = 0.3, 1.0
2 group_plot = group_plots_labels(eufspec_keV_files, obs_dates)
3
4 fig = plt.figure(figsize=(8,3), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 for val in sort_order:
8     data = np.loadtxt(group_plot[val][0], skiprows=3)
9     energy      = data[:,0]
10    eufspec     = data[:,4]

```

D.2. Multi-Observatory Spectral Analysis

```

11     ax.step(energy, eufs, c=plot_colors[sort_order.index(val)],
12             where='mid', zorder=sort_order.index(val), label=group_plot[
13               val][1])
14
15 ax.set_yscale('log')
16 ax.set_xlabel(r'Energy (keV)', fontsize=14)
17 ax.set_ylabel(r'Flux density (Photons $^{-1} \text{ cm}^{-2} \text{ keV}^{-1}$)', fontsize=12.2)
18 ax.set_xlim(energy_lo, energy_hi)
19 ax.set_ylim(1E-11, 1E-1)
20
21 ax.xaxis.set_major_locator(MultipleLocator(0.1))
22 ax.xaxis.set_minor_locator(MultipleLocator(0.02))
23 ax.tick_params(axis='x', which='major', direction='in',
24                 length=6, width=1.5, color='k', pad=6,
25                 labelsize=12, labelcolor='k')
26 ax.tick_params(axis='x', which='minor', direction='in',
27                 length=3, width=1.0, color='k', pad=6,
28                 labelsize=12, labelcolor='k')
29
30 ax.yaxis.set_major_locator(LogLocator(base=10))
31 ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2, 0.4, 0.6, 0.8),
32                           numticks=12))
33 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
34 ax.tick_params(axis='y', which='major', direction='in',
35                 length=6, width=1.5, color='k', pad=6,
36                 labelsize=12, labelcolor='k')
37 ax.tick_params(axis='y', which='minor', direction='in',
38                 length=3, width=1.0, color='k', pad=6,
39                 labelsize=12, labelcolor='k')
40
41 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
42 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
43
44 ax.legend(loc='lower right')
45
46 op_filename = 'mr-vel-uf-keV_abs-edge'
47
48 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
49             bbox_inches='tight') # To be modified while on Google Colab
50 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
51             bbox_inches='tight') # To be modified while on Google Colab

```



```

1 energy_lo, energy_hi = 0.3, 1.0
2 group_plot = group_plots_labels(eufs_keV_files, obs_dates)
3
4 fig = plt.figure(figsize=(8,3), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 for val in sort_order:
8     data = np.loadtxt(group_plot[val][0], skiprows=3)
9     energy      = data[:,0]

```

D.2. Multi-Observatory Spectral Analysis

```

10     eufspec      = data[:,4]
11     ax.step(energy, eufspec, c=plot_colors[sort_order.index(val)], where=
12         'mid', zorder=sort_order.index(val), label=group_plot[val][1])
13
14     ax.set_yscale('log')
15     ax.set_xlabel(r'Energy (keV)', fontsize=14)
16     ax.set_ylabel(r'Flux density (Photons $^{-1} \text{ cm}^{-2} \text{ keV}^{-1}$)', 
17                   fontsize=12.2)
18     ax.set_xlim(energy_lo, energy_hi)
19     ax.set_ylim(1E-11, 1E-1)
20
21     ax.xaxis.set_major_locator(MultipleLocator(0.1))
22     ax.xaxis.set_minor_locator(MultipleLocator(0.02))
23     ax.tick_params(axis='x', which='major', direction='in', length=6, width=1.5,
24                     color='k', pad=6, labelsize=12, labelcolor='k')
25     ax.tick_params(axis='x', which='minor', direction='in', length=3, width=1.0,
26                     color='k', pad=6, labelsize=12, labelcolor='k')
27
28     ax.yaxis.set_major_locator(LogLocator(base=10))
29     ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2, 0.4, 0.6, 0.8),
30                                   numticks=12))
31     ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
32     ax.tick_params(axis='y', which='major', direction='in', length=6, width=1.5,
33                     color='k', pad=6, labelsize=12, labelcolor='k')
34     ax.tick_params(axis='y', which='minor', direction='in', length=3, width=1.0,
35                     color='k', pad=6, labelsize=12, labelcolor='k')
36
37     ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
38     ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
39
40     ax.legend(loc='lower right')
41
42     op_filename = 'mr-vel-uf-keV_abs-edge'
43
44     plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
45                 bbox_inches='tight') # To be modified while on Google Colab
46     plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
47                 bbox_inches='tight') # To be modified while on Google Colab

```



```

1 obs_edges_keV = [0.42, 0.55, 0.72, 0.88]
2 epsilon = 0.02
3 edge_df = identify_edges(edgedf_keV, (energy_lo, energy_hi), elements,
                           obs_edges_keV, epsilon=epsilon)
4 temp_df = edge_df.drop(axis='index', index=3)

```

Overlay of Absorption Edges in keV

```

1 energy_lo, energy_hi = 0.3, 1.0
2 group_plot = group_plots_labels(eufspec_keV_files, obs_dates)
3
4 fig = plt.figure(figsize=(8,3), dpi=100)

```

D.2. Multi-Observatory Spectral Analysis

```

5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 for val in sort_order:
8     data = np.loadtxt(group_plot[val][0], skiprows=3)
9     energy      = data[:,0]
10    eufspec     = data[:,4]
11    ax.step(energy, eufspec, c=plot_colors[sort_order.index(val)], where='mid', zorder=sort_order.index(val), label=group_plot[val][1])
12
13 for i in temp_df.index:
14     energy_edge = edge_df.loc[i, 'energy']
15     label_string = edge_df.loc[i, 'element'] + ' ' + edge_df.loc[i, 'edge'][:-5]
16     ax.axvline(energy_edge, c='k', ls='--', lw=2)
17     ax.text(energy_edge, 0.4, label_string, c='k', ha='right', va='top',
18             rotation=90, transform=ax.get_xaxis_transform())
19
20 ax.set_yscale('log')
21 ax.set_xlabel(r'Energy (keV)', fontsize=14)
22 ax.set_ylabel(r'Flux density (Photons s$^{-1}$ cm$^{-2}$ keV$^{-1}$)', fontsize=12.2)
23 ax.set_xlim(energy_lo, energy_hi)
24 ax.set_ylim(1E-11, 1E-1)
25
26 ax.xaxis.set_major_locator(MultipleLocator(0.1))
27 ax.xaxis.set_minor_locator(MultipleLocator(0.02))
28 ax.tick_params(axis='x', which='major', direction='in', length=6, width=1.5,
29                 color='k', pad=6, labelsize=12, labelcolor='k')
30 ax.tick_params(axis='x', which='minor', direction='in', length=3, width=1.0,
31                 color='k', pad=6, labelsize=12, labelcolor='k')
32
33 ax.yaxis.set_major_locator(LogLocator(base=10))
34 ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2,0.4,0.6,0.8),
35                               numticks=12))
36 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
37 ax.tick_params(axis='y', which='major', direction='in', length=6, width=1.5,
38                 color='k', pad=6, labelsize=12, labelcolor='k')
39 ax.tick_params(axis='y', which='minor', direction='in', length=3, width=1.0,
40                 color='k', pad=6, labelsize=12, labelcolor='k')
41
42 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
43 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
44
45 ax.legend(loc='lower right')
46
47 op_filename = 'mr-vel-uf-keV_abs-edge'
48
49 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
50             bbox_inches='tight') # To be modified while on Google Colab
51 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
52             bbox_inches='tight') # To be modified while on Google Colab

```

D.2. Multi-Observatory Spectral Analysis

Overlay of Absorption Edges in Angstrom

```

1 obs_edges_ang = keV_to_angstrom(obs_edges_keV)
2 wave_lo, wave_hi = 12.0, 36.0
3 group_plot = group_plots_labels(eufspec_files, obs_dates)

1 fig = plt.figure(figsize=(8,3), dpi=100)
2 ax = fig.add_axes((0.1,0.1,0.9,0.9))

3
4 for val in sort_order:
5     data = np.loadtxt(group_plot[val][0], skiprows=3)
6     wavelength = data[:,0]
7     eufspec    = data[:,4]
8     ax.step(wavelength, eufspec, c=plot_colors[sort_order.index(val)],
9     where='mid', zorder=sort_order.index(val), label=group_plot[val][1])

10 for i in temp_df.index:
11     wavelength_edge = edge_df.loc[i, 'wavelength']
12     label_string = edge_df.loc[i, 'element'] + ' ' + edge_df.loc[i, 'edge']
13     label_string.replace('_', ' ')
14     ax.axvline(wavelength_edge, c='k', ls='--', lw=2)
15     ax.text(wavelength_edge, 0.4, label_string, c='k', ha='right', va='top',
16             rotation=90, transform=ax.get_xaxis_transform())

17 ax.set_yscale('log')
18 ax.set_xlabel(r'Wavelength $\lambda$ (\AA)', fontsize=14)
19 ax.set_ylabel(r'Flux density $F_\lambda$ (erg s$^{-1}$ cm$^{-2}$ $\AA$)
20 ^{-1}$)', fontsize=13)
21 ax.set_xlim(wave_lo, wave_hi)
22 ax.set_ylim(1E-22, 1E-11)

23 ax.xaxis.set_major_locator(MultipleLocator(2.0))
24 ax.xaxis.set_minor_locator(MultipleLocator(0.2))
25 ax.tick_params(axis='x', which='major', direction='in', length=6, width=1.5,
26     color='k', pad=6, labelsize=12, labelcolor='k')
27 ax.tick_params(axis='x', which='minor', direction='in', length=3, width=1.0,
28     color='k', pad=6, labelsize=12, labelcolor='k')

29 ax.yaxis.set_major_locator(LogLocator(base=10))
30 ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2,0.4,0.6,0.8),
31     numticks=12))
32 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
33 ax.tick_params(axis='y', which='major', direction='in', length=6, width=1.5,
34     color='k', pad=6, labelsize=12, labelcolor='k')
35 ax.tick_params(axis='y', which='minor', direction='in', length=3, width=1.0,
36     color='k', pad=6, labelsize=12, labelcolor='k')

37 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
38 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')

op_filename = 'mr-vel-uf-ang_abs-edge'

```

D.2. Multi-Observatory Spectral Analysis

```
38 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
39             bbox_inches='tight') # To be modified while on Google Colab
40 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
41             bbox_inches='tight') # To be modified while on Google Colab
```

Absorption Edges on XMM-Newton EPIC-pn Spectrum

```
1 idx = 2
2
3 fig = plt.figure(figsize=(8,3), dpi=100)
4 ax = fig.add_axes((0.1,0.1,0.9,0.9))
5
6 op_filename = plot_edge_overlay('mr-vel', idx, 'all', (wave_lo, wave_hi),
7                                 ax, group_plot, sort_order, temp_df, flux_lo=1E-22, flux_hi=1E-11,
8                                 text_yposn=0.4)
9
10 ax.xaxis.set_major_locator(MultipleLocator(2.0))
11 ax.xaxis.set_minor_locator(MultipleLocator(0.2))
12
13 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
14             bbox_inches='tight')
15 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
16             bbox_inches='tight')

1 idx = 2
2 elem_list = ['N']
3
4 fig = plt.figure(figsize=(6,4), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 op_filename = plot_edge_overlay('mr-vel', idx, elem_list, (25.5, 36), ax,
8                                 group_plot, sort_order, edge_df[edge_df['element'].isin(elem_list)],
9                                 flux_lo=1E-19, flux_hi=1E-13, text_yposn=0.94)
10
11 ax.xaxis.set_major_locator(MultipleLocator(1.0))
12 ax.xaxis.set_minor_locator(MultipleLocator(0.2))
13
14 ax.legend(loc='best', fontsize=12)
15
16 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
17             bbox_inches='tight')
18 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
19             bbox_inches='tight')

1 idx = 2
2 elem_list = ['O']
3
4 fig = plt.figure(figsize=(6,4), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
```

D.2. Multi-Observatory Spectral Analysis

```
7 op_filename = plot_edge_overlay('mr-vel', idx, elem_list, (18, 28), ax,
8     group_plot, sort_order, edge_df[edge_df['element'].isin(elem_list)],
9     flux_lo=1E-16, flux_hi=1E-11, text_yposn=0.94)
10
11 ax.xaxis.set_major_locator(MultipleLocator(1.0))
12 ax.xaxis.set_minor_locator(MultipleLocator(0.2))
13
14 ax.legend(loc='best', fontsize=12)
15
16 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
17             bbox_inches='tight')
18 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
19             bbox_inches='tight')

1 idx = 2
2 elem_list = ['Ne']
3
4 fig = plt.figure(figsize=(6,4), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 op_filename = plot_edge_overlay('mr-vel', idx, elem_list, (12.5, 16), ax,
8     group_plot, sort_order, edge_df[edge_df['element'].isin(elem_list)],
9     flux_lo=4E-13, flux_hi=2E-12, text_yposn=0.94)
10
11 ax.xaxis.set_major_locator(MultipleLocator(1.0))
12 ax.xaxis.set_minor_locator(MultipleLocator(0.1))
13
14 ax.legend(loc='best', fontsize=12)
15
16 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
17             bbox_inches='tight')
18 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
19             bbox_inches='tight')

1 idx = 2
2 elem_list = ['Fe']
3
4 fig = plt.figure(figsize=(6,4), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 op_filename = plot_edge_overlay('mr-vel', idx, elem_list, (15, 20), ax,
8     group_plot, sort_order, temp_df[edge_df['element'].isin(elem_list)],
9     flux_lo=1.8E-13, flux_hi=4E-12, text_yposn=0.94)
10
11 ax.xaxis.set_major_locator(MultipleLocator(1.0))
12 ax.xaxis.set_minor_locator(MultipleLocator(0.1))
13
14 ax.legend(loc='best', fontsize=12)
15
16 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
17             bbox_inches='tight')
18 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
```

D.2. Multi-Observatory Spectral Analysis

```
bbox_inches='tight')
```

D.2.5 Residual Statistics

Residues

```
1 ## This cell is required while on Google Colab
2 from google.colab import drive
3 drive.mount('/content/drive')

1 ## This cell is required while on Google Colab
2 import sys
3 from glob import glob
4 sys.path.append('/content/drive/MyDrive/analysis')
5 from plot_helpers import *
6
7 analysis_path = '/content/drive/MyDrive/analysis/'
8 data_path = '/content/drive/MyDrive/analysis/plots-data/'

1 resid_files = sorted(glob(data_path+'*resid*.qdp')) # To be modified
    while on Google Colab
2 output_folder = analysis_path+'outputs/' # To be modified while on Google
    Colab
3 pdf_folder = 'pdf'
4 png_folder = 'png'
5 try:
6     os.mkdir(output_folder)
7 except FileExistsError as error:
8     print(f"The folder '{output_folder}' already exists.")
9 try:
10     os.mkdir(os.path.join(output_folder, pdf_folder)) # To be modified
        while on Google Colab
11 except FileExistsError as error:
12     print(f"The folder '{pdf_folder}' already exists.")
13 try:
14     os.mkdir(os.path.join(output_folder, png_folder)) # To be modified
        while on Google Colab
15 except FileExistsError as error:
16     print(f"The folder '{png_folder}' already exists.")

1 ## This cell is required while on Google Colab
2 mpl.rcParams['pdf.fonttype'] = 42
3 mpl.rcParams['ps.fonttype'] = 42
4 mpl.rcParams['font.family'] = 'Liberation Sans'
5
6 plot_colors = ['#c9184a', '#007200', '#f46036', '#1d3557', '#7209b7', '#1982c4']
7 obs_dates = {'SIS1-43036000': '1994-12-22', 'ACIS-644': '2000-11-14', 'PN
    -0111150101': '2000-12-16', 'XTI-2611020101': '2019-05-18', 'XTI
    -2611020102': '2019-05-19', 'XTI-2611020103': '2019-05-19'}
```

D.2. Multi-Observatory Spectral Analysis

```

1 group_plot = group_plots_labels(resid_files, obs_dates)
2 sort_order = list(obs_dates.keys())
3
4 idx = 5
5 maj_tick, min_tick = 0.5, 0.1
6
7 energy_lo = 0.2
8 energy_hi = 1.0
9
10 data = np.loadtxt(group_plot[sort_order[idx]][0], skiprows=3)
11 plot_resid(data, os.path.basename(group_plot[sort_order[idx]][0]), (os.
12     path.join(output_folder, pdf_folder), os.path.join(output_folder,
13         png_folder)), energy_lo=energy_lo, energy_hi=energy_hi, major_tick=
14         maj_tick, minor_tick=min_tick)

```

Residual Histogram

```

1 sigma_values = []
2 A_values = []
3 B_values = []
4
5 idx = 5
6 data = np.loadtxt(group_plot[sort_order[idx]][0], skiprows=3)
7 resid = data[:,2]
8
9 sns.set_style('ticks')
10 fig, ax = plt.subplots(figsize=(6,5), dpi=100)
11
12 hist = sns.histplot(resid, kde=True, bins=10, #'doane',
13                      color='#4e148c', stat='probability',
14                      ax=ax, alpha=0.65, line_kws={'label': "KDE fit"})
15 # Ref: https://stats.stackexchange.com/questions/798/calculating-optimal-number-of-bins-in-a-histogram
16
17 resid_values = hist.get_lines()[0].get_data()[0]
18 kde_values = hist.get_lines()[0].get_data()[1]
19 (A, B), covariance = curve_fit(gauss, resid_values, kde_values)
20 sigma = 1 / np.sqrt(2*B)
21
22 ax.plot(resid_values, gauss(resid_values, A, B), c='#c32f27', label=r'
23 Gauss. fit: $\sigma=$'+f'{sigma:.3f}')
24
25 ax.set_xlabel('Residual', fontsize=14)
26 ax.set_ylabel('Probability', fontsize=14)
27
28 ax.tick_params(axis='x', which='major', direction='in',
29                 length=6, width=1.5, color='k', pad=6,
30                 labelsize=12, labelcolor='k')
31
32 ax.set_xlim(0, 0.4)
33 hist_lim = max([abs(min(resid)), abs(max(resid))])

```

D.2. Multi-Observatory Spectral Analysis

```
29 ax.set_xlim(-(round(hist_lim, 1)*1.1), (round(hist_lim, 1)*1.1))
30
31 ax.grid(True, which='major', axis='both', alpha=0.4, c='#8f857d', lw=0.7)
32
33 plt.legend(loc='upper right', fontsize=10)
34
35 data_file = os.path.basename(group_plot[sort_order[idx]][0])
36
37 print(os.path.join(output_folder, pdf_folder, data_file)[-4:]+'-hist.pdf')
    )
38 print(os.path.join(output_folder, png_folder, data_file)[-4:]+'-hist.png')
    )
39
40 plt.savefig(os.path.join(output_folder, pdf_folder, data_file)[-4:]+'-
    hist.pdf', bbox_inches='tight')
41 plt.savefig(os.path.join(output_folder, png_folder, data_file)[-4:]+'-
    hist.png', bbox_inches='tight')
```

Kernel Density Estimate

```
1 A_values.append(A)
2 B_values.append(B)
3 sigma_values.append(sigma)

1 resid_lim = 1.5
2 r = np.linspace(-resid_lim, resid_lim, 1000)
3
4 fig = plt.figure(figsize=(6,5), dpi=100)
5 ax = fig.add_axes((0.1,0.1,0.9,0.9))
6
7 for i in range(len(sigma_values)):
8     Pr = gauss(r, A_values[i], B_values[i])
9     ax.plot(r, Pr, label=sort_order[i], zorder=i, c=plot_colors[i])
10
11 ax.set_xlabel('Residual', fontsize=14)
12 ax.set_ylabel('Probability', fontsize=14)
13 ax.set_ylim(0, 0.3)
14
15 ax.xaxis.set_major_locator(MultipleLocator(0.5))
16 ax.xaxis.set_minor_locator(MultipleLocator(0.1))
17 ax.tick_params(axis='x',which='major',direction='in', length=6, width=1.5,
    color='k', pad=6, labelsize=12, labelcolor='k')
18 ax.tick_params(axis='x',which='minor',direction='in', length=3, width=1.0,
    color='k', pad=6, labelsize=12, labelcolor='k')
19
20 ax.yaxis.set_major_locator(MultipleLocator(0.05))
21 ax.yaxis.set_minor_locator(MultipleLocator(0.01))
22 ax.tick_params(axis='y',which='major',direction='in', length=6, width=1.5,
    color='k', pad=6, labelsize=12, labelcolor='k')
23 ax.tick_params(axis='y',which='minor',direction='in', length=3, width=1.0,
    color='k', pad=6, labelsize=12, labelcolor='k')
```

D.2. Multi-Observatory Spectral Analysis

```
24 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
25 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
26
27 plt.legend(loc='best')
28
29 op_filename = 'mr-vel-resid-gaussfit_all-obs'
30
31 plt.savefig(os.path.join(output_folder, pdf_folder, op_filename)+'.pdf',
32             bbox_inches='tight')
33 plt.savefig(os.path.join(output_folder, png_folder, op_filename)+'.png',
34             bbox_inches='tight')
```

Helper Functions

```
1 import os
2 import numpy as np
3 import matplotlib as mpl
4 import matplotlib.pyplot as plt
5 from matplotlib.ticker import MultipleLocator, LogLocator,
6     FormatStrFormatter, MaxNLocator, FuncFormatter
7 import seaborn as sns
8 import pandas as pd
9 from scipy.optimize import curve_fit
10 from glob import glob
11 import sys
12 from astropy import units as u
13 from astropy import constants as const
14
15 mpl.rcParams['pdf.fonttype'] = 42
16 mpl.rcParams['ps.fonttype'] = 42
17 # mpl.rcParams['font.family'] = 'Carlito'
18 # mpl.rcParams['font.family'] = 'Liberation Sans'
19 mpl.rcParams['font.family'] = 'Calibri'
20
21 pn_dir = 'pn'
22 acis_dir = 'ACIS'
23 sis_dir = 'SIS1'
24 xti_dir = 'XTI'
25
26 def gauss(x, A, B):
27     y = A*np.exp(-1*B*x**2)
28     return y
29
30 def log_formatter(x, pos):
31     if x == 1:
32         return '1'
33     else:
34         return f'$10^{{{{int(np.log10(x))}}}}$'
35
36 def keV_to_angstrom(keV_list):
```

D.2. Multi-Observatory Spectral Analysis

```

36     ang_list = []
37     temp = ((const.h * const.c) / (keV_list*u.keV)).to(u.Angstrom)
38     for i in range(len(temp)):
39         ang_list.append(round(temp[i].value, 2))
40     return sorted(ang_list)
41
42 def keV_to_ang(keV_val, sigfig=3):
43     if not keV_val == 0:
44         ang_val = ((const.h * const.c) / (keV_val*u.keV)).to(u.Angstrom).value
45     else:
46         ang_val = np.NaN
47     return round(ang_val, sigfig)
48
49 def ang_to_keV(ang_val, sigfig=3):
50     if not ang_val == 0:
51         keV_val = ((const.h * const.c) / (ang_val*u.Angstrom)).to(u.keV).value
52     else:
53         keV_val = np.NaN
54     return round(keV_val, sigfig)
55
56 def find_files(folder, file_type, file_ext):
57     files = []
58     all_files = os.listdir(folder)
59     for file in all_files:
60         if not file.find(file_type)==-1:
61             if file.endswith(file_ext):
62                 files.append(os.path.join(os.path.abspath(file)))
63     return files
64
65 def plot_ldata(data, data_file, op_folders, energy_lo=0.2, energy_hi=1.0):
66     :
67     energy = data[:,0]
68     ebins = data[:,1]
69     rate = data[:,2]
70     rerr = data[:,3]
71     model = data[:,4]
72
73     fig = plt.figure(figsize=(6,4), dpi=100)
74     ax = fig.add_axes((0.1,0.1,0.9,0.9))
75
76     ax.errorbar(energy, rate, xerr=ebins, yerr=rerr, fmt='+', c='#3f37c9',
77     , label='Data')
78     ax.step(energy, model, c='#ef233c', where='mid', zorder=3, label='Model')
79     ax.set_yscale('log')
80     ax.set_xlabel(r'Energy (keV)', fontsize=14)
81     ax.set_ylabel(r'counts s$^{-1}$ keV$^{-1}$', fontsize=14)
82     ax.set_xlim(energy_lo, energy_hi)
83
84     ax.xaxis.set_major_locator(MultipleLocator(0.1))
85     ax.xaxis.set_minor_locator(MultipleLocator(0.01))
86     ax.tick_params(axis='x', which='major', direction='in',

```

D.2. Multi-Observatory Spectral Analysis

```
85             length=6, width=1.5, color='k', pad=6,
86             labelsize=12, labelcolor='k')
87     ax.tick_params(axis='x', which='minor', direction='in',
88                     length=3, width=1.0, color='k', pad=6,
89                     labelsize=12, labelcolor='k')
90
91     ax.yaxis.set_major_locator(LogLocator(base=10))
92     ax.yaxis.set_minor_locator(LogLocator(base=10, subs
93 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9), numticks=12))
94     ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
95     ax.tick_params(axis='y', which='major', direction='in',
96                     length=6, width=1.5, color='k', pad=6,
97                     labelsize=12, labelcolor='k')
98     ax.tick_params(axis='y', which='minor', direction='in',
99                     length=3, width=1.0, color='k', pad=6,
100                    labelsize=12, labelcolor='k')
101
102     ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
103     ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
104
105     ax.legend(loc='upper right')
106
107     pdf_folder, png_folder = op_folders
108
109     plt.savefig(os.path.join(pdf_folder, data_file)[-3:]+'pdf',
110                 bbox_inches='tight')
111     plt.savefig(os.path.join(png_folder, data_file)[-3:]+'png',
112                 bbox_inches='tight')
113
114 def plot_counts(data, data_file, source_name, op_folders, x_ticks=(0.1,
115 0.01), energy_lo=0.2, energy_hi=1.0):
116     energy = data[:,0]
117     ebins = data[:,1]
118     rate = data[:,2]
119     rerr = data[:,3]
120
121     fig = plt.figure(figsize=(6,4), dpi=100)
122     ax = fig.add_axes((0.1,0.1,0.9,0.9))
123
124     ax.errorbar(energy, rate,
125                 xerr=ebins, yerr=rerr,
126                 fmt='+', elinewidth=0.8, alpha=0.8,
127                 c='#3f37c9', label='Counts')
128
129     ax.set_yscale('log')
130     ax.set_xlabel(r'Energy (keV)', fontsize=14)
131     ax.set_ylabel(r'counts s$^{-1}$ keV$^{-1}$', fontsize=14)
132     ax.set_title(f'{source_name} count rates', fontsize=16)
133     ax.set_xlim(energy_lo, energy_hi)
134
135     x_majticks, x_minticks = x_ticks
136     ax.xaxis.set_major_locator(MultipleLocator(x_majticks))
137     ax.xaxis.set_minor_locator(MultipleLocator(x_minticks))
```

D.2. Multi-Observatory Spectral Analysis

```

133     ax.tick_params(axis='x', which='major', direction='in',
134                     length=6, width=1.5, color='k', pad=6,
135                     labelsize=12, labelcolor='k')
136     ax.tick_params(axis='x', which='minor', direction='in',
137                     length=3, width=1.0, color='k', pad=6,
138                     labelsize=12, labelcolor='k')
139
140     ax.yaxis.set_major_locator(LogLocator(base=10))
141     ax.yaxis.set_minor_locator(LogLocator(base=10, subs
142 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9), numticks=12))
143     ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
144     ax.tick_params(axis='y', which='major', direction='in',
145                     length=6, width=1.5, color='k', pad=6,
146                     labelsize=12, labelcolor='k')
147     ax.tick_params(axis='y', which='minor', direction='in',
148                     length=3, width=1.0, color='k', pad=6,
149                     labelsize=12, labelcolor='k')
150
151     ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
152     ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
153
154     #ax.legend(loc='upper right')
155
156     pdf_folder, png_folder = op_folders
157
158     plt.savefig(os.path.join(pdf_folder, data_file)[-3:]+'.pdf',
159                 bbox_inches='tight')
160     plt.savefig(os.path.join(png_folder, data_file)[-3:]+'.png',
161                 bbox_inches='tight')
162
163 def plot_eufspec(data, data_file, op_folders, wave_lo=12.0, wave_hi=62.0)
164 :
165     wavelength = data[:,0]
166     wbins      = data[:,1]
167     rate       = data[:,2]
168     rerr       = data[:,3]
169     eufspec    = data[:,4]
170
171     fig = plt.figure(figsize=(6,4), dpi=100)
172     ax = fig.add_axes((0.1,0.1,0.9,0.9))
173
174     ax.errorbar(wavelength, rate, xerr=wbins, yerr=rerr, fmt='+', c='#3
175 f37c9')
176     ax.step(wavelength, eufspec, c='#ef233c', where='mid', zorder=3,
177             label='Unfolded spectrum')
178     ax.set_yscale('log')
179     ax.set_xlabel(r'Wavelength $\lambda$ (\AA)', fontsize=14)
180     ax.set_ylabel(r'Flux density $F_\lambda$ (erg s$^{-1}$ cm$^{-2}$ $\AA
181 ^{-1}$)', fontsize=14)
182     ax.set_xlim(wave_lo, wave_hi)
183
184     ax.xaxis.set_major_locator(MultipleLocator(2.0))

```

D.2. Multi-Observatory Spectral Analysis

```

178 ax.xaxis.set_minor_locator(MultipleLocator(1.0))
179 ax.tick_params(axis='x', which='major', direction='in',
180                 length=6, width=1.5, color='k', pad=6,
181                 labelsize=12, labelcolor='k')
182 ax.tick_params(axis='x', which='minor', direction='in',
183                 length=3, width=1.0, color='k', pad=6,
184                 labelsize=12, labelcolor='k')
185
186 ax.yaxis.set_major_locator(LogLocator(base=10))
187 ax.yaxis.set_minor_locator(LogLocator(base=10, subs
188 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9), numticks=12))
189 ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
190 ax.tick_params(axis='y', which='major', direction='in',
191                 length=6, width=1.5, color='k', pad=6,
192                 labelsize=12, labelcolor='k')
193 ax.tick_params(axis='y', which='minor', direction='in',
194                 length=3, width=1.0, color='k', pad=6,
195                 labelsize=12, labelcolor='k')
196
197 ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
198 ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
199
200 ax.legend(loc='upper right')
201
202 pdf_folder, png_folder = op_folders
203
204 plt.savefig(os.path.join(pdf_folder, data_file)[-3:]+'pdf',
205 bbox_inches='tight')
206 plt.savefig(os.path.join(png_folder, data_file)[-3:]+'png',
207 bbox_inches='tight')
208
209 def plot_eufspec_keV(data, data_file, op_folders, energy_lo=0.2,
210                      energy_hi=1.0):
211     energy    = data[:,0]
212     ebins    = data[:,1]
213     rate     = data[:,2]
214     rerr     = data[:,3]
215     eufspec  = data[:,4]
216
217     fig = plt.figure(figsize=(6,4), dpi=100)
218     ax = fig.add_axes((0.1,0.1,0.9,0.9))
219
220     ax.errorbar(energy, rate, xerr=ebins, yerr=rerr, fmt='+', c='#3f37c9')
221     ax.step(energy, eufspec, c='ef233c', where='mid', zorder=3, label='Unfolded spectrum')
222     ax.set_yscale('log')
223     ax.set_xlabel(r'Energy (keV)', fontsize=14)
224     ax.set_ylabel(r'Flux density (Photons s$^{-1}$ cm$^{-2}$ keV$^{-1}$)', fontsize=14)
225     ax.set_xlim(energy_lo, energy_hi)

```

D.2. Multi-Observatory Spectral Analysis

```

223     ax.xaxis.set_major_locator(MultipleLocator(0.2))
224     ax.xaxis.set_minor_locator(MultipleLocator(0.02))
225     ax.tick_params(axis='x', which='major', direction='in',
226                     length=6, width=1.5, color='k', pad=6,
227                     labelsize=12, labelcolor='k')
228     ax.tick_params(axis='x', which='minor', direction='in',
229                     length=3, width=1.0, color='k', pad=6,
230                     labelsize=12, labelcolor='k')
231
232     ax.yaxis.set_major_locator(LogLocator(base=10))
233     #ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2, 0.4, 0.6, 0.8),
234     , numticks=12))
235     ax.yaxis.set_minor_locator(LogLocator(base=10, subs
236     =(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9), numticks=12))
237     ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
238     ax.tick_params(axis='y', which='major', direction='in',
239                     length=6, width=1.5, color='k', pad=6,
240                     labelsize=12, labelcolor='k')
241     ax.tick_params(axis='y', which='minor', direction='in',
242                     length=3, width=1.0, color='k', pad=6,
243                     labelsize=12, labelcolor='k')
244
245     ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
246     ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')
247
248     ax.legend(loc='lower right')
249
250     pdf_folder, png_folder = op_folders
251
252     plt.savefig(os.path.join(pdf_folder, data_file)[-3:]+'.pdf',
253     bbox_inches='tight')
254     plt.savefig(os.path.join(png_folder, data_file)[-3:]+'.png',
255     bbox_inches='tight')
256
257 def plot_resid(data, data_file, op_folders, energy_lo=0.2, energy_hi=1.0,
258                 major_tick=1.0, minor_tick=0.5):
259     energy = data[:,0]
260     resid = data[:,2]
261
262     fig, ax = plt.subplots(figsize=(6,4), dpi=100)
263
264     plt.axhline(0, c='k', alpha=0.8, lw=1.2)
265     plt.step(energy, resid)
266
267     resid_lim = max([abs(min(resid)), abs(max(resid))])
268     ax.set_xlim(-(round(resid_lim, 1)+0.2), (round(resid_lim, 1)+0.2))
269     ax.set_xlabel(r'Energy (keV)', fontsize=14)
270     ax.set_ylabel('Residual', fontsize=14)
271     ax.set_xlim(energy_lo, energy_hi)
272
273     resid_lim = max([abs(min(resid)), abs(max(resid))]); limfactor = 1.2
274     ax.set_xlim(-(round(resid_lim, 1)*limfactor), (round(resid_lim, 1)*)
```

D.2. Multi-Observatory Spectral Analysis

```
270     limfactor))
271     #ax.set_ylim(-4, 4)
272
273     ax.xaxis.set_major_locator(MultipleLocator(0.1))
274     ax.xaxis.set_minor_locator(MultipleLocator(0.01))
275     ax.tick_params(axis='x', which='major', direction='in',
276                     length=6, width=1.5, color='k', pad=6,
277                     labelsize=12, labelcolor='k')
278     ax.tick_params(axis='x', which='minor', direction='in',
279                     length=3, width=1.0, color='k', pad=6,
280                     labelsize=12, labelcolor='k')
281
282     ax.yaxis.set_major_locator(MultipleLocator(major_tick))
283     ax.yaxis.set_minor_locator(MultipleLocator(minor_tick))
284     ax.tick_params(axis='y', which='major', direction='in',
285                     length=6, width=1.5, color='k', pad=6,
286                     labelsize=12, labelcolor='k')
287     ax.tick_params(axis='y', which='minor', direction='in',
288                     length=3, width=1.0, color='k', pad=6,
289                     labelsize=12, labelcolor='k')
290
291     ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000', lw=0.8)
292     ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000', lw=0.8)
293
294     pdf_folder, png_folder = op_folders
295
296     plt.savefig(os.path.join(pdf_folder, data_file)[-3:]+'.pdf',
297                 bbox_inches='tight')
298     plt.savefig(os.path.join(png_folder, data_file)[-3:]+'.png',
299                 bbox_inches='tight')
300
301 def group_plots_labels(data_files, obs_dates):
302     plot_labels = []
303     group_plot = {}
304     for file in data_files:
305         instrument = os.path.basename(file).split('-')[3].split('_')[0].upper()
306         if instrument == 'ACIS':
307             satellite = 'Chandra'
308         elif instrument == 'PN':
309             satellite = 'XMM'
310         elif instrument == 'SIS1':
311             satellite = 'ASCA'
312         elif instrument == 'XTI':
313             satellite = 'NICER'
314         obs_id = os.path.basename(file).split('-')[2]
315         obs_date = (obs_dates[f'{instrument}-{obs_id}'])
316
317         if instrument == 'PN':
318             plot_label = f'{obs_date} | {satellite}:EPIC-{instrument}.
```

D.2. Multi-Observatory Spectral Analysis

```

    lower())
    plot_labels.append(f'{obs_date} | {satellite}:EPIC-{instrument.lower() }')
else:
    plot_label = f'{obs_date} | {satellite}:{instrument}'
    plot_labels.append(f'{obs_date} | {satellite}:{instrument}')
plot_labels.append(plot_label)

group_plot[f'{instrument}-{obs_id}'] = (file, plot_label)
return group_plot

def overlay_edges(ax, url, energy_lims, elements, obs_edges, epsilon=0.02):
    df = pd.read_csv(url)
    energy_lo, energy_hi = energy_lims
    abs_edge = df.columns.tolist()[3:]
    for idx in df.index:
        element = df.loc[idx, 'element']
        if element in elements:
            for ae in abs_edge:
                edge_value = df.loc[idx, ae]
                if edge_value >= energy_lo and edge_value <= energy_hi:
                    for oe in obs_edges:
                        if edge_value >= oe-epsilon and edge_value <= oe+epsilon:
                            ax.axvline(edge_value, c='k', ls='--', lw=2)
                            label_string = f'{element} {ae[:-5]} {edge_value:.3f} keV'
                            ax.text(edge_value, 0.78, label_string, c='k', ha='right',
                                    va='top', rotation=90,
                                    transform=ax.get_xaxis_transform())
def get_edgedf(df):
    edgedf_keV = df.copy() # DF containing edge data in keV
    edgedf_ang = pd.DataFrame() # Create an empty DF for storing edge data in angstroms
    collist = df.columns.tolist() # Store the DF column names in a list
    for i in range(len(df)): # Run a loop over all rows
        row = {} # Create an empty dictionary that will store values for current row
        for col in collist: # Run a loop over all columns
            if col in collist[3:]: # Only for those columns which contain edge values...
                edge_ang = keV_to_ang(df.loc[i, col], sigfig=2) # Convert the value from keV to angstrom
                row[col] = edge_ang # Store the angstrom value as a dictionary entry
            else: # For those columns which do not contain edge values...
                row[col] = df.loc[i, col] # Store the DF value as is
    edgedf_ang = pd.concat([edgedf_ang, pd.DataFrame([row])], ignore_index=True) # Append a row into the DF as a new row
    return edgedf_keV, edgedf_ang

```

D.2. Multi-Observatory Spectral Analysis

```

357 def identify_edges_keV(edgedf_keV, energy_lims, elements, obs_edges_keV,
358     epsilon=0.02):
359     edge_keV = pd.DataFrame(columns=['element', 'edge', 'energy'])
360     for i in edgedf_keV.index:
361         element = edgedf_keV.loc[i, 'element']
362         if element in elements:
363             abs_edge = edgedf_keV.columns.tolist()[3:]
364             for ae in abs_edge:
365                 edge_val = edgedf_keV.loc[i, ae]
366                 if edge_val >= energy_lims[0] and edge_val <= energy_lims[1]:
367                     for oe in obs_edges_keV:
368                         if edge_val >= oe-epsilon and edge_val <= oe+epsilon:
369                             edge_keV.loc[len(edge_keV.index)] = [element, ae, edge_val]
370     return edge_keV
371
372 def identify_edges(edgedf_keV, energy_lims, elements, obs_edges_keV,
373     epsilon=0.02):
374     edge_df = pd.DataFrame(columns=['element', 'edge', 'energy',
375         'wavelength'])
376     for i in edgedf_keV.index:
377         element = edgedf_keV.loc[i, 'element']
378         if element in elements:
379             abs_edge = edgedf_keV.columns.tolist()[3:]
380             for ae in abs_edge:
381                 edge_keV = edgedf_keV.loc[i, ae]
382                 if edge_keV >= energy_lims[0] and edge_keV <= energy_lims[1]:
383                     for oe in obs_edges_keV:
384                         if edge_keV >= oe-epsilon and edge_keV <= oe+epsilon:
385                             edge_df.loc[len(edge_df.index)] = [element, ae, edge_keV,
386                             keV_to_ang(edge_keV)]
387     return edge_df
388
389 def plot_edge_overlay(source_name, idx, elem_list, wave_lims, ax,
390     group_plot, sort_order, edge_df, flux_lo=1E-22, flux_hi=1E-11,
391     text_yposn=0.78):
392     wave_lo, wave_hi = wave_lims
393     data = np.loadtxt(group_plot[sort_order[idx]][0], skiprows=3)
394     wavelength = data[:,0]
395     eufspec = data[:,4]
396     #ax.plot(wavelength, eufspec)
397     ax.step(wavelength, eufspec, where='mid')
398
399     if isinstance(elem_list, list):
400         edge_df = edge_df[edge_df['element'].isin(elem_list)]
401
402     for i in edge_df.index:
403         wavelength_edge = edge_df.loc[i, 'wavelength']
404         #label_string = edge_df.loc[i, 'element'] + ' ' + edge_df.loc[i, 'edge'][:-5]
405         label_string = edge_df.loc[i, 'element'] + ' ' + edge_df.loc[i, 'edge']
406         label_string.replace('_', ' ')
407         #ax.axvline(wavelength_edge, c='k', ls='--', lw=2, label='Literature'

```

D.2. Multi-Observatory Spectral Analysis

```

        ')
    #.axvline(wavelength_edge, c='k', ls='--', lw=2, label=f'{label_string}')
    #ax.text(wavelength_edge, text_yposn, label_string, c='k', ha='right',
    ', va='top', rotation=90,
    #           transform=ax.get_xaxis_transform())
    if len(elem_list)>1:
        ax.axvline(wavelength_edge, c='k', ls='--', lw=2)
        ax.text(wavelength_edge, text_yposn, label_string, c='k', ha='right',
        va='top', rotation=90,
        transform=ax.get_xaxis_transform())
    else:
        ed_val = edge_df.loc[i,'wavelength']
        ax.axvline(wavelength_edge, c='k', ls='--', lw=2, label=f'{label_string}')
        ax.text(wavelength_edge, text_yposn, f'{ed_val:.2f} '+r'$\AA$', c='k',
        ha='right', va='top', rotation=90,
        transform=ax.get_xaxis_transform())

    ax.set_yscale('log')
    ax.set_xlabel(r'Wavelength $\lambda$ ($\AA$)', fontsize=14)
    ax.set_ylabel(r'Flux density $F_\lambda$ (erg s$^{-1}$ cm$^{-2}$) $\AA$',
    '^{-1}$)', fontsize=14)
    ax.set_title(group_plot[sort_order[idx]][1])
    ax.set_xlim(wave_lo, wave_hi)
    ax.set_ylim(flux_lo, flux_hi)

    ax.xaxis.set_major_locator(MultipleLocator(2.0))
    ax.xaxis.set_minor_locator(MultipleLocator(1.0))
    ax.tick_params(axis='x', which='major', direction='in',
    length=6, width=1.5, color='k', pad=6,
    labelsize=12, labelcolor='k')
    ax.tick_params(axis='x', which='minor', direction='in',
    length=3, width=1.0, color='k', pad=6,
    labelsize=12, labelcolor='k')

    ax.yaxis.set_major_locator(LogLocator(base=10))
    ax.yaxis.set_minor_locator(LogLocator(base=10, subs=(0.2,0.4,0.6,0.8),
    numticks=12))
    ax.yaxis.set_major_formatter(FuncFormatter(log_formatter))
    ax.tick_params(axis='y', which='major', direction='in',
    length=6, width=1.5, color='k', pad=6,
    labelsize=12, labelcolor='k')
    ax.tick_params(axis='y', which='minor', direction='in',
    length=3, width=1.0, color='k', pad=6,
    labelsize=12, labelcolor='k')

    ax.grid(True, which='major', axis='both', alpha=0.4, c='#38b000')
    ax.grid(True, which='minor', axis='both', alpha=0.1, c='#38b000')

instr = group_plot[sort_order[idx]][1].split(' |')[1][1:].replace(':', '-')

```

D.3. Python Interface for Spectral Analysis

```
444
445     if isinstance(elem_list, str):
446         op_filename = f'{source_name}-{instr}-uf-all-edges'
447     else:
448         op_filename = f'{source_name}-{instr}-uf-'
449         for e in elem_list:
450             op_filename += e + '-'
451         op_filename += 'edges'
452     return op_filename
```

D.3 Python Interface for Spectral Analysis

D.3.1 Phase I: PySAS Data Reduction

```
1 from astroda_utilities import *
2
3 source_name = 'rx-j0925.7-4758'
4 dataset_name = 'parag-dataset'
5
6 obs_id       = get_obsid()
7 odf_dir, work_dir, epic_dir, mos_dir, pn_dir, rgs_dir = get_dirs(
8     source_name, dataset_name)
9 os.chdir(odf_dir)
9 rev = get_rev()

1 !gunzip -v *.gz

1 from pysas.wrapper import Wrapper as w
2
3 os.environ['SAS_ODF'] = odf_dir
4 t = w('sasver', [])
5 t.run()

1 !cifbuild

1 os.environ['SAS_CCF'] = f'{odf_dir}/ccf.cif'

1 !odfingest

1 os.environ['SAS_ODF'] = f'{odf_dir}/{rev}_{obs_id}_SCX00000SUM.SAS'
2 t = w('sasver', [])
3 t.run()

1 os.mkdir(work_dir)
2 os.mkdir(epic_dir)
3 os.mkdir(mos_dir)
4 os.mkdir(pn_dir)
5 os.mkdir(rgs_dir)
```

D.3. Python Interface for Spectral Analysis

```

1 os.chdir(pn_dir)
2
3 eproc = 'epproc'
4 in_args = []
5 pnevt_list = run_epicproc(eproc, in_args)

```

D.3.2 Phase II: Pileup Investigation

```

1 pn_qflag      = '#XMMEA_EP' # quality flag for EPIC pn
2 pn_pattern    = 0           # pattern selection
3 pn_flag       = 0           # FLAG
4
5 filt_file     = f'{source_name}_filt.fits'
6 filtimg_file  = f'{source_name}_filt-img.fits'

1 sas_task = "evselect" # SAS task to be executed
2
3 # Arguments of SAS Command
4 filter_expression = f'{pn_qflag} && (PATTERN=={pn_pattern}) && (FLAG=={pn_flag})'
5 inargs      = [f'table={pn_dir}/{pnevt_list[0]}', 'keepfilteroutput=yes',
   'withfilteredset=yes', f'filteredset={filt_file}', 'writedss=yes',
   'updateexposure=yes', 'filterexposure=yes', 'energycolumn=PHA', f'
   expression={filter_expression}']
6 print(f'Filter expression to use: {filter_expression} \n')
7 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
8 inargs

1 w(sas_task, inargs).run()

1 sas_task = "evselect" # SAS task to be executed
2
3 # Arguments of SAS Command
4 inargs      = [f'table={pn_dir}/{filt_file}', 'withimageset=yes', f'
   imageset={filtimg_file}', 'xcolumn=X', 'ycolumn=Y']
5 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
6 inargs

1 w(sas_task, inargs).run()

1 xpa_id = ds9_linking(f'{pn_dir}/{filtimg_file}')

1 # Ref.: https://het.as.utexas.edu/HET/Software/pyDS9/
2 d = pyds9.DS9(xpa_id)
3 d.set('scale log')
4 d.set('cmap bb');

1 ds9_region_instructions()
2 src_reg = ds9_region(d)

```

D.3. Python Interface for Spectral Analysis

```

1 src_coord_file = 'src.xy'
2 with open(src_coord_file, 'w+') as f:
3     for i in range(len(src_reg)-1):
4         f.write('%f\n' %src_reg[i])
5     print(f"Source region successfully saved in file {src_coord_file}!")
6 f.close()
7
8 srcevt_file = f'{source_name}_filt_{int(px_to_arcsec(src_reg[2]))}arcsec.
9     fits'
```



```

1 sas_task = "evselect" # SAS task to be executed
2
3 # Arguments of SAS Command
4 filter_expression = f'(X,Y) IN circle({src_reg[0]},{src_reg[1]},{src_reg
5 [2]})'
6 inargs      = [f'table={pn_dir}/{filt_file}', 'keepfilteroutput=yes', '
7     withfilteredset=yes', f'filteredset={srcevt_file}', f'expression={
8     filter_expression}']
9 print(f'Filter expression to use: {filter_expression} \n')
10 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
11 inargs
```



```

1 w(sas_task, inargs).run()
```



```

1 srcimg_file = f'{source_name}_filt-img_{int(px_to_arcsec(src_reg[2]))}_
2     arcsec.fits'
```



```

1 sas_task = "evselect" # SAS task to be executed
2
3 # Arguments of SAS Command
4 inargs      = [f'table={pn_dir}/{srcevt_file}', 'withimageset=yes', f'
5     imageset={srcimg_file}', 'xcolumn=X', 'ycolumn=Y', f'expression={
6     filter_expression}']
7 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
8 inargs
```



```

1 w(sas_task, inargs).run()
```



```

1 # https://heasarc.gsfc.nasa.gov/lheasoft/heasoftpy/
2 import heasoftpy as hsp
3
4 fstat = hsp.fstatistic(infile=srcevt_file, colname='PHA', rows='-')
5 total_counts = fstat.params['numb']
6 print(total_counts)
```



```

1 hdu_list = fits.open(srcevt_file)
2
3 value_ONTIME = hdu_list['EVENTS'].header['ONTIME']
4 print(value_ONTIME)
```



```

1 observed_cps = total_counts / value_ONTIME
2 print(observed_cps)
```

D.3. Python Interface for Spectral Analysis

```
1 frame_time = 5.7E-3
2 observed_cpf = observed_cps * frame_time
3 print(observed_cpf)

1 flux_loss = float(input(f'Using obs. counts per frame of {"%.3f"%
    observed_cpf},\nrefer to the above plot to estimate an upper limit of
    flux loss (in %): '))
2 spectral_distortion = float(input(f'Using obs. counts per frame of {"%.3f"
    "%observed_cpf},\nrefer to the above plot to estimate an upper limit
    of spectral distortion (in %): '))
```

D.3.3 Phase III: Spectrum Extraction

```
1 from astroda_utilities import *
2
3 source_name = 'rx-j0925.7-4758'
4 dataset_name = 'parag-dataset'
5
6 obs_id       = get_obsid()
7 odf_dir, work_dir, epic_dir, mos_dir, pn_dir, rgs_dir = get_dirs(
        source_name, dataset_name)
8 os.chdir(odf_dir)
9 rev = get_rev()

1 rev = get_rev()
2 os.environ['SAS_CCF'] = f'{odf_dir}/ccf.cif'
3 os.environ['SAS_ODF'] = f'{odf_dir}/{rev}_{obs_id}_SCX00000SUM.SAS'
4
5 t = w('sasver', [])
6 t.run()

1 os.chdir(pn_dir)
2 os.listdir()
```

Source Spectrum Extraction

```
1 src_reg = []
2 f = open('src.xy', 'r')
3 for x in f.readlines():
4     src_reg.append(float(x))
5 f.close()
6 src_reg.append(arcsec_to_px(float(input("Enter radius of source region (
    in arcsec): "))))
7 print(src_reg)

1 src_reg_file = 'src.reg'
2 with open(src_reg_file, 'w+') as f:
3     for i in range(len(src_reg)):
4         f.write('%f\n' %src_reg[i])
5     print(f"Source region successfully saved in file {src_reg_file}!")
6 f.close()
```

D.3. Python Interface for Spectral Analysis

```

1 srcspec_file = f'{source_name}.src'
2
3 filt_file = f'{source_name}_filt.fits'
4 bin_size = 5
5 pattern = 'PATTERN==0'
6
7 sas_task = "evselect" # SAS task to be executed
8
9 # Arguments of SAS Command
10 filter_expression = f'({pattern}) && ((X,Y) IN circle({src_reg[0]}, {
11     src_reg[1]}, {src_reg[2]}))'
12 inargs      = [f'table={pn_dir}/{filt_file}', 'withspectrumset=yes', f'
13     spectrumset={srcspec_file}', 'energycolumn=PI', f'spectralbinsize={
14         bin_size}', 'withspecranges=yes', 'specchannelmin=0', 'specchannelmax
15         =20479', f'expression={filter_expression}']
16 print(f'Filter expression to use: {filter_expression} \n')
17 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
18 inargs
19
20 w(sas_task, inargs).run()

```

Background Spectrum Extraction

```

1 x = float(input("Enter the RA of the source (degrees): "))
2 y = float(input("Enter the Dec of the source (degrees): "))
3 r = float(input("Enter the radius of the source region (arcsec): "))

4 filtimg_file = f'{source_name}_filt-img.fits'
5
6 sas_task = "ebkgreg" # SAS task to be executed
7
8 # Arguments of SAS Command
9 inargs      = [f'imageset={pn_dir}/{filtimg_file}', 'withsrclist=no', '
10     withcoords=yes', 'coordtype=EQPOS', f'x={x}', f'y={y}', f'r={r}']
11 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
12 inargs
13
14 w(sas_task, inargs).run()

15 xpa_id = ds9_linking_silent()
16 d = pyds9.DS9(xpa_id)

17 bkg_reg = ds9_region(d)
18 print(bkg_reg)

19 bkg_reg_file = 'bkg.reg'
20 with open(bkg_reg_file, 'w+') as f:
21     for i in range(len(bkg_reg)):
22         f.write('%f\n' %bkg_reg[i])
23     print(f"Background region successfully saved in file {bkg_reg_file}!")
24
25 f.close()

```

D.3. Python Interface for Spectral Analysis

```

1 bkg_reg = []
2 f = open('bkg.reg', 'r')
3 for x in f.readlines():
4     bkg_reg.append(float(x))
5 f.close()

1 bkgspec_file = f'{source_name}.bkg'
2
3 filt_file = f'{source_name}_filt.fits'
4 bin_size = 5
5 pattern = 'PATTERN==0'
6
7 sas_task = "evselect" # SAS task to be executed
8
9 # Arguments of SAS Command
10 filter_expression = f'({pattern}) && ((X,Y) IN circle({bkg_reg[0]}, {
11     bkg_reg[1]},{bkg_reg[2]}))'
12 inargs      = [f'table={pn_dir}/{filt_file}', 'withspectrumset=yes', f'
13     spectrumset={bkgspec_file}', 'energycolumn=PI', f'spectralbinsize={
14     bin_size}', 'withspecranges=yes', 'specchannelmin=0', 'specchannelmax
15     =20479', f'expression={filter_expression}']
16 print(f'Filter expression to use: {filter_expression} \n')
17 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
18 inargs
19
20 w(sas_task, inargs).run()

```

Backscaling Source and Background Spectra

```

1 pnevt_list = run_epicproc('epproc', [])
2
3 sas_task = "backscale" # SAS task to be executed
4
5 # Arguments of SAS Command
6 inargs = [f'spectrumset={pn_dir}/{srcspec_file}', f'badpixlocation={
7     pn_dir}/{pnevt_list[0]}']
8
9 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
10 inargs
11
12 w(sas_task, inargs).run()
13
14 sas_task = "backscale" # SAS task to be executed
15
16 # Arguments of SAS Command
17 inargs = [f'spectrumset={pn_dir}/{bkgspec_file}', f'badpixlocation={
18     pn_dir}/{pnevt_list[0]}']
19
20 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
21 inargs
22
23 w(sas_task, inargs).run()

```

D.3. Python Interface for Spectral Analysis

Generating RMF

```
1 rmf_file = f'{source_name}.rmf'
2
3 sas_task = "rmfgen" # SAS task to be executed
4
5 # Arguments of SAS Command
6 inargs = [f'spectrumset={pn_dir}/{srcspec_file}', f'rmfset={rmf_file}']
7
8 print(f'SAS command to be executed: {sas_task}, with arguments: \n')
9 inargs

10 w(sas_task, inargs).run()
```

Generating ARF

```
1 arf_file = f'{source_name}.arf'
2
3 sas_task = "arfgen" # SAS task to be executed
4
5 print("Checking for Response File ..... \n")
6 # Check if RESP file is available.
7 if os.path.isfile(rmf_file):
8     print (f"File {rmf_file} exists. \n")
9     inargs = [f'spectrumset={pn_dir}/{srcspec_file}', f'arfset={arf_file}',
10             'withrmfset=yes', f'rmfset={pn_dir}/{rmf_file}', f'badpixlocation={pn_dir}/{pnev_list[0]}',
11             'detmaptpe=psf']
12     print(f'SAS command to be executed: {sas_task}, with arguments: \n')
13     print(inargs)
14 else:
15     print (f'File {rmf_file} does not exist, please run the rmfgen task
16 first, or check. \n')

17 w(sas_task, inargs).run()
```

Spectrum Grouping

```
1 # Run this cell if rebinning is required during a later session
2 from astroda_utilities import *
3
4 source_name = 'rx-j0925.7-4758'
5 dataset_name = 'parag-dataset'
6
7 obs_id      = get_obsid()
8 odf_dir, work_dir, epic_dir, mos_dir, pn_dir, rgs_dir = get_dirs(
9     source_name, dataset_name)
10 os.chdir(odf_dir)
11 rev = get_rev()
12 os.environ['SAS_CCF'] = f'{odf_dir}/ccf.cif'
```

D.3. Python Interface for Spectral Analysis

```

13 os.environ['SAS_ODF'] = f'{odf_dir}/{rev}_{obs_id}_SCX00000SUM.SAS'
14
15 t = w('sasver', [])
16 t.run()
17
18 os.chdir(pn_dir)
19
20 srcspec_file = f'{source_name}.src'
21 bkgspec_file = f'{source_name}.bkg'
22 rmf_file = f'{source_name}.rmf'
23 arf_file = f'{source_name}.arf'

min_counts = 10
grp_file = source_name + f'_{min_counts:02d}c.grp'
sas_task = "specgroup" # SAS task to be executed
# Arguments of SAS Command
inargs = [f'mincounts={min_counts}', f'spectrumset={pn_dir}/{srcspec_file}',
          f'rmfset={pn_dir}/{rmf_file}', f'arfset={pn_dir}/{arf_file}', f'',
          f'backgndset={pn_dir}/{bkgspec_file}', f'groupedset={grp_file}']

print(f'SAS command to be executed: {sas_task}, with arguments: \n')
inargs
w(sas_task, inargs).run()

```

D.3.4 Phase IV: PyXspec Analysis

```

1 from astroda_utilities import *
2
3 source_name = 'rx-j0925.7-4758'
4 dataset_name = 'parag-dataset'
5
6 obs_id       = get_obsid()
7 odf_dir, work_dir, epic_dir, mos_dir, pn_dir, rgs_dir = get_dirs(
        source_name, dataset_name)
8 os.chdir(odf_dir)
9 rev = get_rev()

# Clear all data and models
AllModels.clear() # Clear all models
AllData.clear()    # Clear all data
#
# Load the group spectral file
grp_file = find_specgroup_file(min_counts=10)
data = Spectrum(f'{pn_dir}/{grp_file}') # load spectra grouped file

lo_E = 0.2 # keV
hi_E = 1.1 # keV

```

D.3. Python Interface for Spectral Analysis

```
4 ignore_string = f'**-{lo_E} {hi_E}-**'
5 data.ignore(ignore_string)
```

Initial Inspection

```
1 Plot.device='/null'
2 Plot.xAxis='keV'
3 Plot('ldata')
4
5 energies = Plot.x()
6 edeltas = Plot.xErr()
7 rates = Plot.y()
8 errors = Plot.yErr()
9
10 labels = Plot.labels()

1 print(f'Number of data points: {len(energies)}')
2 matplotlib.rcParams['text.usetex'] = True
3 fig, ax = plt.subplots(figsize=(6,4), dpi=150)
4 ax.errorbar(energies, rates, xerr=edeltas, yerr=errors, fmt='.', ms=6,
     elinewidth=1.0)
5 ax.set_xlabel(labels[0])
6 ax.set_ylabel(labels[1])
7 ax.set_xscale('linear')
8 ax.set_yscale('log')
9 ax.grid()
10 ax.set_title(f'{source_name.upper()} ${obs_id} $ Observed count rate
      ')
11 ax.ticklabel_format(axis='x', style='plain')
```

Model Fitting

```
1 modelstats_df = pd.DataFrame(columns=['additive_model', 'log_g', '
    reduced_chisq', 'T_eff'])
2 os.mkdir(os.path.join(pn_dir, 'fits'))
3 os.mkdir(os.path.join(pn_dir, 'resids'))

1 Xset.chatter = 5
2 Xset.abund = 'wilm'
3 Xset.xsect = 'vern'
4
5 Fit.query = 'yes'
```

```
1 Xset.chatter = 10
2 comp_add = 'bbody'
3 comp_mult1 = 'ismabs'
4
5 mod_1 = f'{comp_add}*{comp_mult1}'
6
7 AllModels.clear()
8 model1 = Model(mod_1)
```

D.3. Python Interface for Spectral Analysis

```
1 NH_galactic = 4.42E21 # cm-2 (nh task of ftools)
2
3 Xset.chatter = 10
4 Xset.show()
5
6 modell.show()

1 Fit.renorm(setting='auto')
2 Fit.perform()
3
4 Plot.device='/null'
5 Plot.xAxis='keV'
6 Plot('ldata resid')
7 energies = Plot.x()
8 edeltas = Plot.xErr()
9 rates = Plot.y(plotWindow = 1)
10 errors = Plot.yErr()
11
12 foldedmodel = Plot.model()
13 residuals = Plot.y(plotWindow = 2)
14
15 labels = Plot.labels()

1 matplotlib.rcParams['text.usetex'] = True
2 fig, ax = plt.subplots(2, 1, figsize=(6,6), dpi=150)
3 fig.suptitle(f'{source_name.upper()} ${} \{obs_id\} ${} Model: {mod_1}')
4
5 ax[0].errorbar(energies, rates, xerr=edeltas, yerr=errors, fmt='.', ms=3,
6                 elinewidth=1.0, label='Data')
6 ax[0].step(energies, foldedmodel, where='mid', c='r', label='Model')
7 ax[0].set_ylabel(Plot.labels(plotWindow=1)[1])
8 ax[0].set_xscale('linear')
9 ax[0].set_yscale('log')
10 ax[0].grid()
11 ax[0].ticklabel_format(axis='x', style='plain')
12 ax[0].set_title(f"Data and folded model ${} \chi^2={reduced_chisq(Fit)}",
13                  fontsize=10)
13 ax[0].legend(loc='best')
14
15 ax[1].axhline(0, c='k', lw=0.8)
16 ax[1].errorbar(energies, residuals, xerr=edeltas, yerr=errors, fmt='.', 
17                 ms=4, elinewidth=1.0)
17 ax[1].set_xlabel(Plot.labels(plotWindow=2)[0])
18 ax[1].set_ylabel(Plot.labels(plotWindow=2)[1])
19 ax[1].set_title("Residuals", fontsize=10)
20 ax[1].grid()

1 print(f'Reduced Chi-squared: {reduced_chisq(Fit)}')
2
3 T_bb = keV_to_K(modell.bbody.kT.values[0])
4 print(f'Best-fit blackbody temperature: {display_temp(T_bb)[0]} {
5       display_temp(T_bb)[1]}')



```

D.3. Python Interface for Spectral Analysis

Parameter Adjustment

```
1 model1.show()
2
3 bbody_kT = float(input("Enter bbody.kT (in keV): "))
4 model1.bbody.kT = bbody_kT
5 model1.bbody.kT.frozen = False

1 Fit.renorm(setting='auto')
2 Fit.perform()
3
4 print(f'Reduced Chi-squared: {reduced_chisq(Fit)}')
5 T_bb = keV_to_K(model1.bbody.kT.values[0])
6 print(f'Best-fit blackbody temperature: {display_temp(T_bb)[0]} {display_temp(T_bb)[1]}')
7
8 Plot.device='/null'
9 Plot.xAxis='keV'
10 Plot('ldata resid')
11 energies = Plot.x()
12 edeltas = Plot.xErr()
13 rates = Plot.y(plotWindow = 1)
14 errors = Plot.yErr()
15
16 foldedmodel = Plot.model()
17 residuals = Plot.y(plotWindow = 2)
18
19 labels = Plot.labels()

1 matplotlib.rcParams['text.usetex'] = True
2 fig, ax = plt.subplots(2, 1, figsize=(6,6), dpi=150)
3 fig.suptitle(f'{source_name.upper()} ${}^{}_{\circ} \{obs_id\} \${}^{}_{\circ} \{Model: mod_1\}')
4
5 ax[0].errorbar(energies, rates, xerr=edeltas, yerr=errors, fmt='.', ms=3,
6                 elinewidth=1.0, label='Data')
7 ax[0].step(energies, foldedmodel, where='mid', c='r', label='Model')
8 ax[0].set_ylabel(Plot.labels(plotWindow=1)[1])
9 ax[0].set_xscale('linear')
10 ax[0].set_yscale('log')
11 ax[0].grid()
12 ax[0].ticklabel_format(axis='x', style='plain')
13 ax[0].set_title(f"Data and folded model $\chi^2={reduced_chisq(Fit)}"
14                 , fontsize=10)
15 ax[0].legend(loc='best')
16
17 ax[1].axhline(0, c='k', lw=0.8)
18 ax[1].errorbar(energies, residuals, xerr=edeltas, yerr=errors, fmt='.', 
19                 ms=4, elinewidth=1.0)
20 ax[1].set_xlabel(Plot.labels(plotWindow=2)[0])
21 ax[1].set_ylabel(Plot.labels(plotWindow=2)[1])
22 ax[1].set_title("Residuals", fontsize=10)
23 ax[1].grid()
```

D.3. Python Interface for Spectral Analysis

```

21 plt.savefig(f'{pn_dir}/fits/{source_name}-{obs_id}_{comp_add}_specfit.png
22     , dpi=200)

1 import seaborn as sns
2 from scipy.stats import norm
3
4 resid_hist = sns.displot(x=residuals, bins=20, kde=True)
5 resid_hist.set(title=f'{source_name.upper()} ${} {obs_id} ${} Model: {
6     mod_1} ${} Residual Histogram')
7 kde_curve = resid_hist.ax.lines[0].get_ydata()
8
9 mu, std = norm.fit(residuals)
10 xmin, xmax = plt.xlim()
11 x = np.linspace(xmin, xmax, 100)
12 p = norm.pdf(x, mu, std)
13
14 scale_factor = max(kde_curve) / (1 / (std * np.sqrt(2 * np.pi)))
15 p_scaled = p * scale_factor
16
17 plt.plot(x, p_scaled, 'r', linewidth=2)
18 plt.xlabel(f'Residual ({Plot.labels(plotWindow=2)[1]})')
19 plt.ylabel('Counts')
20 plt.savefig(f'{pn_dir}/resids/{source_name}-{obs_id}_{comp_add}_resid-
dist.png', dpi=200)

```

NLTE Model Fitting

```

1 Xset.chatter = 5
2 Xset.abund = 'wilm'
3 Xset.xsect = 'vern'
4
5 Fit.query = 'yes'

1 NH_galactic = 6.5E20 # cm-2 (Gansicke et al. 1998)
2 XH_rolleston = {'He':0.04, 'C':-0.34, 'N':-0.30, 'O':-0.32, 'Ne':0.00, 'Na':
    :0.00, 'Mg':-0.47, 'Al':0.00, 'Si':-0.30, 'S':0.00, 'Cl':0.00, 'Ar':
    :0.00, 'Ca':0.00, 'Cr':0.00, 'Fe':-0.17, 'Co':0.00, 'Ni':0.00} # (
    Rolleston et al. 2002)

1 nlte_dir = '/home/pararover/Documents/Research/2.0/data-analysis/nlte-
    models'
2 pureH_dir = os.path.join(nlte_dir, 'pure-H')
3 HCa_dir = os.path.join(nlte_dir, 'H-Ca')
4 HNi_dir = os.path.join(nlte_dir, 'H-Ni')

1 # Extracting log g values from the NLTE filenames
2 files = os.listdir(pureH_dir)
3 files.sort()
4 logg_values = []
5 nlte_files = {}

1 # Extracting log g values from the NLTE filenames
2 files = os.listdir(pureH_dir)
3 files.sort()
4 logg_values = []
5 nlte_files = {}

```

D.3. Python Interface for Spectral Analysis

```

6 for f in files:
7     logg_values.append(find_float(f))
8     nlte_files.update({find_float(f):f})

1 # Set the log g value here
2 logg = 7.0
3 nlte_type = 'Pure H'
4 nlte_file = os.path.join(pureH_dir, nlte_files[logg])

1 Xset.chatter = 10
2 comp_add = 'atable{' + f'{nlte_file}' + '}'
3 comp_mult1 = 'TBabs'
4 comp_mult2 = 'ismabs'
5 comp_mult3 = 'redden'
6
7 mod_2 = f'{comp_mult1}*{comp_mult2}*{comp_mult3}*{comp_add}'
8
9 AllModels.clear()
10 model2 = Model(mod_2)

1 Xset.chatter = 5
2 set_TBabs(model2, 1.5E22, freezeFlag=False)
3 set_vphabs(model2, XH_rolloston)
4
5 Xset.chatter = 10
6 Xset.show()
7
8 model2.show()

1 Xset.chatter = 5
2 Xset.restore('model-ism-rauch-05.xcm')
3 modl = AllModels(1)

1 Xset.chatter = 10
2 Fit.renorm(setting='auto')
3 Fit.perform()
4
5 Plot.device='/null'
6 Plot.xAxis='keV'
7 Plot('ldata resid')
8 energies = Plot.x()
9 edeltas = Plot.xErr()
10 rates = Plot.y(plotWindow = 1)
11 errors = Plot.yErr()
12
13 foldedmodel = Plot.model()
14 residuals = Plot.y(plotWindow = 2)
15
16 labels = Plot.labels()

1 matplotlib.rcParams['text.usetex'] = True
2 fig, ax = plt.subplots(2, 1, figsize=(6,6), dpi=150)

```

D.4. Variability Analysis

```
3 fig.suptitle(f'{source_name.upper()} ${obs_id} Model: {nlte_type}(\n    log g={logg})*vphabs*tbabs')\n4\n5 ax[0].errorbar(energies, rates, xerr=edeltas, yerr=errors, fmt='.', ms=3,\n    elinewidth=1.0, label='Data')\n6 ax[0].step(energies, foldedmodel, where='mid', c='r', label='Model')\n7 ax[0].set_ylabel(Plot.labels(plotWindow=1)[1])\n8 ax[0].set_xscale('linear')\n9 ax[0].set_yscale('log')\n10 ax[0].grid()\n11 ax[0].ticklabel_format(axis='x', style='plain')\n12 ax[0].set_title(f"Data and folded model $\backslash\chi^2=\${reduced_chisq(Fit)}"\n    , fontsize=10)\n13 ax[0].legend(loc='best')\n14\n15 ax[1].axhline(0, c='k', lw=0.8)\n16 ax[1].errorbar(energies, residuals, xerr=edeltas, yerr=errors, fmt='.',\n    ms=4, elinewidth=1.0)\n17 ax[1].set_xlabel(Plot.labels(plotWindow=2)[0])\n18 ax[1].set_ylabel(Plot.labels(plotWindow=2)[1])\n19 ax[1].set_title("Residuals", fontsize=10)\n20 ax[1].grid()\n21\n22 plt.savefig(f'{pn_dir}/fits/{source_name}-{obs_id}_pureH-{logg}_specfit.\n    png', dpi=200)\n\n1 print(f'Reduced Chi-squared: {reduced_chisq(Fit)}')\n2\n3 T_bb = modl.rauch.T.values[0]\n4 print(f'Best-fit blackbody temperature: {display_temp(T_bb)[0]} {\n    display_temp(T_bb)[1]}'')
```

D.4 Variability Analysis

```
1 import sys\n2 from google.colab import drive\n3 drive.mount('/content/drive')\n\n1 analysis_path = '/content/drive/MyDrive/analysis/dataset/timing'\n2 data_path = '/content/drive/MyDrive/analysis/dataset/timing/lc-data'\n3 op_path = '/content/drive/MyDrive/analysis/dataset/timing/op-plots'\n4\n5 sys.path.append(analysis_path)\n\n1 import numpy as np\n2 import matplotlib.pyplot as plt\n3 import astropy.units as u\n4 import pandas as pd\n5 import os\n6
```

D.4. Variability Analysis

```
7 from astropy.io import fits
8 from astropy.timeseries import LombScargle
9
10 from scipy.optimize import curve_fit
```

D.4.1 Reading Lightcurves

```
1 fname01 = "rx-j0925-4758_NICER_LC.fits"
2 fname02 = "rx-j0925-4758_NICER_full-SSS-LC.fits"

1 hdulist01 = fits.open(os.path.join(data_path, fname01))
2 hdulist01.info()

1 hdulist02 = fits.open(os.path.join(data_path, fname02))
2 hdulist02.info()

1 def sinusoid(t, nu, A, phi, C_0):
2     C = A * np.sin(2 * np.pi * nu * t + phi) + C_0
3     return C
4
5 def get_output_path(op_path, output_basename, output_type, output_detail)
6     :
7     output_fname = output_basename + output_type + output_detail + '.png'
8     output_path = os.path.join(op_path, output_fname)
9     return output_path

1 lc_data_01 = np.array(hdulist01[1].data)
2 t01 = np.array([row[0] for row in lc_data_01]) * u.second
3 rate01 = np.array([row[1] for row in lc_data_01]) * u.s**-1
4 rateError01 = np.array([row[2] for row in lc_data_01]) * u.s**-1
5
6 mask01 = ~np.isnan(rate01)      # Masking all array elements with NaN
7     values
8
9 t01 = t01[mask01]
10 rate01 = rate01[mask01]
11 rateError01 = rateError01[mask01]

1 lc_data_02 = np.array(hdulist02[1].data)
2 t02 = np.array([row[0] for row in lc_data_02]) * u.second
3 rate02 = np.array([row[1] for row in lc_data_02]) * u.s**-1
4 rateError02 = np.array([row[2] for row in lc_data_02]) * u.s**-1
5
6 mask02 = ~np.isnan(rate02)      # Masking all array elements with NaN
7     values
8
9 t02 = t02[mask02]
10 rate02 = rate02[mask02]
11 rateError02 = rateError02[mask02]
```

D.4. Variability Analysis

```

1 source = 'RX J0925.7-4758'
2 observatory = 'NICER'
3 output_basename = 'rx-j0925_'

1 energy_lo, energy_hi = 0.2, 1.0
2 band01 = str(energy_lo) + r' $-$ ' + str(energy_hi) + ' keV'
3
4 energy_lo, energy_hi = 0.2, 12.0
5 band02 = str(energy_lo) + r' $-$ ' + str(energy_hi) + ' keV'

1 output_type = 'lc'
2 output_detail = ''
3
4 plt.figure(figsize=(10, 4), dpi=120)
5 plt.errorbar(t01.to(u.ks), rate01, yerr=rateError01, fmt='b.', alpha=0.5,
    label=band01)
6 plt.errorbar(t02.to(u.ks), rate02, yerr=rateError02, fmt='r.', alpha=0.5,
    label=band02)
7 plt.xlabel(f"Time ({t01.to(u.ks)[0].unit})")
8 plt.ylabel(f"Rate (counts/{t01[0].unit})")
9 plt.title(source + ' | ' + observatory + ' lightcurve')
10 plt.legend()
11 plt.grid()
12
13 plt.savefig(get_output_path(op_path, output_basename, output_type,
    output_detail), dpi=120)

1 thresh_hi = 15
2 thresh_lo = 0
3
4 mask01 = (rate01.value >= thresh_lo) & (rate01.value <= thresh_hi)
5
6 thresh_hi = 17
7 thresh_lo = 0
8
9 mask02 = (rate02.value >= thresh_lo) & (rate02.value <= thresh_hi)

1 t01 = t01[mask01]
2 rate01 = rate01[mask01]
3 rateError01 = rateError01[mask01]
4
5 t02 = t02[mask02]
6 rate02 = rate02[mask02]
7 rateError02 = rateError02[mask02]

1 output_type = 'lc'
2 output_detail = '-bothbands'
3
4 plt.figure(figsize=(10, 4), dpi=120)
5 plt.errorbar(t01.to(u.ks), rate01, yerr=rateError01, fmt='b.', alpha=0.5,
    label=band01, zorder=2)
6 plt.errorbar(t02.to(u.ks), rate02, yerr=rateError02, fmt='r.', alpha=0.5,
    label=band02, zorder=1)

```

D.4. Variability Analysis

```

7 plt.xlabel(f"Time ({t01.to(u.ks)[0].unit})")
8 plt.ylabel(f"Rate (counts/{t01[0].unit})")
9 plt.title(source + ' | ' + observatory + ' lightcurve')
10 plt.legend()
11 plt.grid()
12
13 print(get_output_path(op_path, output_basename, output_type,
14 output_detail))
15 plt.savefig(get_output_path(op_path, output_basename, output_type,
16 output_detail), dpi=120)

```



```

1 output_type = 'lc'
2 output_detail = '_bothbands_11-13ks'
3
4 plt.figure(figsize=(10, 4), dpi=120)
5 plt.errorbar(t01.to(u.ks), rate01, yerr=rateError01, fmt='b.', alpha=0.8,
6 label=band01, zorder=2)
7 plt.errorbar(t02.to(u.ks), rate02, yerr=rateError02, fmt='r.', alpha=0.8,
8 label=band02, zorder=1)
9 plt.xlabel(f"Time ({t01.to(u.ks)[0].unit})")
10 plt.ylabel(f"Rate (counts/{t01[0].unit})")
11 plt.title(source + ' | ' + observatory + ' lightcurve')
12
13 plt.xlim(11, 13)
14 plt.xticks(np.arange(11, 13.5, 0.5))
15
16 plt.legend()
17 plt.grid()
18
19 print(get_output_path(op_path, output_basename, output_type,
20 output_detail))
21 plt.savefig(get_output_path(op_path, output_basename, output_type,
22 output_detail), dpi=120)

```



```

1 output_type = 'nicer-lc'
2 output_detail = '_bothbands-inset'
3
4 fig, ax1 = plt.subplots(figsize=(10, 4), dpi=120)
5
6 ax1.errorbar(t01.to(u.ks), rate01, yerr=rateError01, fmt='b.', alpha=0.5,
7 label=band01, zorder=2)
8 ax1.errorbar(t02.to(u.ks), rate02, yerr=rateError02, fmt='r.', alpha=0.5,
9 label=band02, zorder=1)
10 ax1.set_xlim(6, 30)
11 ax1.set_xlabel(f"Time ({t01.to(u.ks)[0].unit})")
12 ax1.set_ylabel(f"Rate (counts/{t01[0].unit})")
13 ax1.set_title(source + ' | ' + observatory + ' lightcurve')
14 ax1.legend()
15 ax1.grid()
16

```

D.4. Variability Analysis

```
15 x1, x2, y1, y2 = 11, 12.75, 9, 18 # subregion of the original image
16 axins = ax1.inset_axes(
17     [0.58, 0.58, 0.39, 0.38],
18     xlim=(x1, x2), ylim=(y1, y2), xticklabels=[], yticklabels[])
19 axins.errorbar(t01.to(u.ks), rate01,
20                 yerr=rateError01, fmt='b.', alpha=0.8, label=band01, zorder
21                 =2)
22 axins.errorbar(t02.to(u.ks), rate02,
23                 yerr=rateError02, fmt='r.', alpha=0.6, label=band02, zorder
24                 =1)
25 axins.set_xticks([11, 11.5, 12, 12.5], [11, 11.5, 12, 12.5])
26 axins.set_yticks([9, 12, 15, 18], [9, 12, 15, 18])
27 axins.grid()
28
29 ax1.indicate_inset_zoom(axins, edgecolor="black")
30
31 print(get_output_path(op_path, output_basename, output_type,
32                     output_detail))
33 plt.savefig(get_output_path(op_path, output_basename, output_type,
34                     output_detail), dpi=120)
```

D.4.2 Lomb-Scargle Periodogram

```
1 ls01 = LombScargle(t01, rate01, rateError01)
2 frequency01, power01 = ls01.autopower()
3
4 ls02 = LombScargle(t02, rate02, rateError02)
5 frequency02, power02 = ls02.autopower()
6
7 print(ls01.false_alarm_probability(max(power01)))
8 print(ls02.false_alarm_probability(max(power01)))
9
10 output_type = 'nicer-ls'
11 output_detail = '_bothbands-inset'
12
13 signal_freq = []
14 signal_period = []
15 signal_power01, signal_power02 = [], []
16
17 fig, ax1 = plt.subplots(figsize=(10, 4), dpi=120)
18
19 ax1.plot(frequency01.to(u.mHz), power01, c='r', alpha=0.8, label=band01,
20           zorder=1)
21 ax1.plot(frequency02.to(u.mHz), power02, c='b', alpha=0.8, label=band02,
22           zorder=2)
23 ax1.set_xlim(-0.5, 1.5)
24 ax1.set_ylim(-0.1, 0.6)
25 ax1.set_xticks([0.0, 0.5, 1.0, 1.5], [0.0, 0.5, 1.0, 1.5])
26 ax1.set_xlabel('Frequency (mHz)')
27 ax1.set_ylabel('Power / Frequency')
```

D.4. Variability Analysis

```

17 ax1.set_title(source + ' | ' + observatory + ' | Lomb-Scargle periodogram
   ')
18 ax1.legend(loc="lower left")
19 ax1.grid()
20
21 x1, x2, y1, y2 = 0.0, 0.1, -0.05, 0.55 # subregion of the original image
22 axins = ax1.inset_axes([0.58, 0.58, 0.39, 0.38], xlim=(x1, x2), ylim=(y1,
   y2), xticklabels=[], yticklabels[])
23 axins.plot(frequency01.to(u.mHz), power01, c='r', alpha=0.8, zorder=1)
24 axins.plot(frequency02.to(u.mHz), power02, c='b', alpha=0.8, zorder=2)
25
26 idx=5; axins.axvline(x=frequency01[idx].to(u.mHz).value, c='g', alpha
   =1.0, ls='--', lw=1, label=f'{frequency01[idx].to(u.mHz).value:.3f} mHz')
27 signal_freq.append(frequency01[idx])
28 signal_period.append(1/frequency01[idx])
29 signal_power01.append(power01[idx])
30 signal_power02.append(power02[idx])
31
32 idx=10; axins.axvline(x=frequency01[idx].to(u.mHz).value, c='m', alpha
   =1.0, ls='--', lw=1, label=f'{frequency01[idx].to(u.mHz).value:.3f} mHz')
33 signal_freq.append(frequency01[idx])
34 signal_period.append(1/frequency01[idx])
35 signal_power01.append(power01[idx])
36 signal_power02.append(power02[idx])
37
38 axins.set_xticks(np.arange(0, 0.11, 0.02), np.arange(0, 0.11, 0.02))
39 axins.set_yticks([0.0, 0.25, 0.50], [0.0, 0.25, 0.50])
40 axins.legend()
41 axins.grid()
42
43 ax1.indicate_inset_zoom(axins, edgecolor="black")
44
45 print('Freq.(Hz)\tPeriod(s)\tPower/Freq (01)\tPower/Freq (02)')
46 for i in range(len(signal_freq)):
47     print(str(signal_freq[i].value) + '\t' + str(signal_period[i].value)
48           + '\t' + str(signal_power01[i].value) + '\t' + str(signal_power02[i].value))
49
50 print(get_output_path(op_path, output_basename, output_type,
   output_detail))
51 plt.savefig(get_output_path(op_path, output_basename, output_type,
   output_detail), dpi=120)

1 num_peaks = 100 # Set this to a high enough value so that sufficient
   red dots
2                                     # mark the peaks of interest, compared with above plot
3 peak_indices = np.argpartition(power01, -num_peaks)[-num_peaks:]
4
5 plt.plot(peak_indices, power01[peak_indices], 'ro')
6 for i in range(len(peak_indices)):
7     plt.text(peak_indices[i], power01[peak_indices[i]], str(peak_indices[

```

D.4. Variability Analysis

```
    i]))  
8  
9 plt.title('Note down index values of peaks');  
  
1 peak_idx = [5, 10]      # Make a list of index values of peaks  
2                      # from above plot  
3  
4 sine_cols = ['g', 'm']
```

D.4.3 Timing Analysis

Phase-folded Lightcurve

```
1 T01 = t01 % signal_period[0]  
2 phase01 = 2*np.pi*(T01) / signal_period[0]  
3  
4 T02 = t01 % signal_period[1]  
5 phase02 = 2*np.pi*(T02) / signal_period[1]  
  
1 output_type = 'nicer-lc_200-1000'  
2 output_detail = '_phasefold'  
3  
4 fig, axs = plt.subplots(2, 1, figsize=(10, 10), dpi=120, tight_layout=  
    True)  
5  
6 fig.suptitle(source + ' | ' + observatory + ' | ' + band01 + ' phase-  
    folded lightcurve')  
7  
8 axs[0].errorbar(phase01, rate01, rateError01, label='Lomb-Scargle '+r'$\nu=$'+f'{signal_freq[0].to(u.mHz):.3f}, '+r'$P=$'+f'{signal_period[0].to(u.ks):.1f}', fmt='o', alpha=0.75)  
9 axs[0].set_xlabel('Phase (rad)')  
10 axs[0].set_ylabel('Rate (counts/s)')  
11 axs[0].set_xticks([0, np.pi/2, np.pi, 3*np.pi/2, 2*np.pi], labels=[r'$0$',  
    r'$\frac{\pi}{2}$', r'$\pi$', r'$\frac{3\pi}{2}$', r'$2\pi$'])  
12 axs[0].legend(loc="lower left")  
13 axs[0].grid()  
14  
15 axs[1].errorbar(phase02, rate01, rateError01, label='Lomb-Scargle '+r'$\nu=$'+f'{signal_freq[1].to(u.mHz):.3f}, '+r'$P=$'+f'{signal_period[1].to(u.ks):.1f}', fmt='o', alpha=0.8)  
16 axs[1].set_xlabel('Phase (rad)')  
17 axs[1].set_ylabel('Rate (counts/s)')  
18 axs[1].set_xticks([0, np.pi/2, np.pi, 3*np.pi/2, 2*np.pi], labels=[r'$0$',  
    r'$\frac{\pi}{2}$', r'$\pi$', r'$\frac{3\pi}{2}$', r'$2\pi$'])  
19 axs[1].legend(loc="lower left")  
20 axs[1].grid()  
21  
22 print(get_output_path(op_path, output_basename, output_type,  
    output_detail))
```

D.4. Variability Analysis

```
23 plt.savefig(get_output_path(op_path, output_basename, output_type,
    output_detail), dpi=120)
```

Sinusoid Fitting

```
1 initial_guess = [3, 0, 11]
2
3 A_fit, phi_fit, C_0_fit = [], [], []
4
5 idx = T01.value.argsort()
6 params, covariance = curve_fit(lambda t01, A, phi, C_0: sinusoid(t01,
7     signal_freq[0].value, A, phi, C_0), T01.value[idx], rate01.value[idx],
8     p0=initial_guess)
9 A_fit.append(params[0])
10 phi_fit.append(params[1])
11 C_0_fit.append(params[2])
12
13 idx = T02.value.argsort()
14 params, covariance = curve_fit(lambda t01, A, phi, C_0: sinusoid(t01,
15     signal_freq[1].value, A, phi, C_0), T02.value[idx], rate01.value[idx],
16     p0=initial_guess)
17 A_fit.append(params[0])
18 phi_fit.append(params[1])
19 C_0_fit.append(params[2])
20
21 T01_fit = np.linspace(min(T01.value), max(T01.value), 100)
22 phase01_fit = np.linspace(0, 2*np.pi, 100)
23 rate01_fit = sinusoid(T01_fit, signal_freq[0].value, A_fit[0], phi_fit
24     [0], C_0_fit[0])
25
26 T02_fit = np.linspace(min(T02.value), max(T02.value), 100)
27 phase02_fit = np.linspace(0, 2*np.pi, 100)
28 rate02_fit = sinusoid(T02_fit, signal_freq[1].value, A_fit[1], phi_fit
29     [1], C_0_fit[1])
30
31 output_type = 'nicer-lc_200-1000'
32 output_detail = '_phasefold-bestfit'
33
34 fig, axs = plt.subplots(2, 1, figsize=(10, 10), dpi=120, tight_layout=True)
35
36 fig.suptitle(source + ' | ' + observatory + ' | ' + band01 + ' phase-
37     folded lightcurve')
38
39 axs[0].errorbar(phase01, rate01, rateError01, label='Lomb-Scargle '+r'$\nu=$'+f'{signal_freq[0].to(u.mHz):.3f}, '+r'$P=$'+f'{signal_period[0].to(u.ks):.1f}', fmt='o', alpha=0.75, zorder=1)
40 axs[0].plot(phase01_fit, rate01_fit, c=sine_cols[0], label='Best-fit: '+r'$A=$'+f'{A_fit[0]:.1f} counts/s, '+r'$\phi=$'+f'{phi_fit[0]:.2f} rad, '+r'$C_0=$'+f'{C_0_fit[0]:.1f} counts/s', zorder=3)
41 axs[0].axhline(y=C_0_fit[0], c='k', ls='--', lw=1.2, alpha=0.75, label=f'Average rate = {C_0_fit[0]:.1f} counts/s', zorder=2)
```

D.4. Variability Analysis

```

11 axs[0].set_xlabel('Phase (rad)')
12 axs[0].set_ylabel('Rate (counts/s)')
13 axs[0].set_xticks([0, np.pi/2, np.pi, 3*np.pi/2, 2*np.pi], labels=[r'$0$',
14 , r'$\frac{\pi}{2}$', r'$\pi$', r'$\frac{3\pi}{2}$', r'$2\pi$'])
15 axs[0].legend(loc="lower left")
16
17 axs[1].errorbar(phase02, rate01, rateError01, label='Lomb-Scargle '+r'$\nu$'
18 nu=$'+f'{signal_freq[1].to(u.mHz):.3f}', '+r'$P=$'+f'{signal_period[1].
19 to(u.ks):.1f}', fmt='o', alpha=0.8)
20 axs[1].plot(phase02_fit, rate02_fit, c=sine_cols[1], label='Best-fit: '+r
21 '$A=$'+f'{A_fit[1]:.1f} counts/s, '+r'$\phi=$'+f'{phi_fit[1]:.2f} rad
22 , '+r'$C_0=$'+f'{C_0_fit[1]:.1f} counts/s', zorder=3)
23 axs[1].axhline(y=C_0_fit[1], c='k', ls='--', lw=1.2, alpha=0.75, label=f'
24 Average rate = {C_0_fit[1]:.1f} counts/s', zorder=2)
25 axs[1].set_xlabel('Phase (rad)')
26 axs[1].set_ylabel('Rate (counts/s)')
27 #axs[1].set_xlim(0, 2*np.pi)
28 axs[1].set_xticks([0, np.pi/2, np.pi, 3*np.pi/2, 2*np.pi], labels=[r'$0$',
29 , r'$\frac{\pi}{2}$', r'$\pi$', r'$\frac{3\pi}{2}$', r'$2\pi$'])
30 axs[1].legend(loc="lower left")
31 axs[1].grid()
32
33 print('Freq. (Hz) \t A (cts/s) \t $\phi$ (rad) \t C_0 (cts/s)')
34 for i in range(len(signal_freq)):
35     print(str(signal_freq[i].value)+'\t'+str(A_fit[i])+'\t'+str(phi_fit[i])
36     +'\t'+str(C_0_fit[i]))
37
38 print(get_output_path(op_path, output_basename, output_type,
39         output_detail))
40 plt.savefig(get_output_path(op_path, output_basename, output_type,
41         output_detail), dpi=120)

```

```

1 T = np.linspace(min(t01.value), max(t01.value), 1000)
2 rate01_fit = sinusoid(T, signal_freq[0].value, A_fit[0], phi_fit[0],
3     C_0_fit[0])
4 rate02_fit = sinusoid(T, signal_freq[1].value, A_fit[1], phi_fit[1],
5     C_0_fit[1])

```

```

1 output_type = 'nicer-lc_200-1000'
2 output_detail = '_bestfit'
3
4 plt.figure(figsize=(10, 4), dpi=120)
5 plt.errorbar(t01.to(u.ks), rate01, rateError01, label=band01, fmt='.',
6     alpha=0.75, zorder=1)
7 plt.plot(T*u.ks/1E3, rate01_fit, c=sine_cols[0], alpha=0.75, label=r'$\nu$'
8     =''+f'{signal_freq[0].to(u.mHz):.3f}', '+r'$P=$'+f'{signal_period[0].to
9     (u.ks):.1f}', zorder=2)
10 plt.plot(T*u.ks/1E3, rate02_fit, c=sine_cols[1], alpha=0.75, label=r'$\nu$'
11     =''+f'{signal_freq[1].to(u.mHz):.3f}', '+r'$P=$'+f'{signal_period[1].to
12     (u.ks):.1f}', zorder=3)
13 plt.xlabel(f"Time ({t01[0].to(u.ks).unit})")

```

D.4. Variability Analysis

```
9 plt.ylabel(f"Rate (counts/{t01[0].unit})")
10 plt.legend(loc='lower left', fontsize=10)
11 plt.grid()
12
13 plt.title(source + ' | ' + observatory + ' | ' + band01 + ' lightcurve
14     with best-fit sinusoids')
15 print(get_output_path(op_path, output_basename, output_type,
16     output_detail))
16 plt.savefig(get_output_path(op_path, output_basename, output_type,
    output_detail), dpi=120)
```