Universidad Zaragoza
1542

Trabajo Fin de Máster

Título del trabajo:
RadioPy: un paquete en Python para analizar emisión radio sincrotrón

English tittle:
RadioPy: a synchrotron radio package in Python

Autor/es

Miguel González San Emeterio

Director/es

Miguel Ángel Pérez Torres

FACULTAD DE CIENCIAS
2017

UNIVERSIDAD DE ZARAGOZA

RadioPy: a Python tool for synchrotron radio emission

Máster en Física y Tecnologías Físicas

Author

Miguel González San Emeterio

Supervisor

Miguel A. Pérez-Torres

Departamento de Física Teórica

Contents

Abstract

RadioPy is an open source Python package for radio astrophysics capable of fitting spectra of synchrotron radio emission. Adapted for supernovae in its first version, one of its main features is that it can perform a complete an automatic analysis for a sequence of supernova spectra and calculate the shockwave radius (R), the intensity of the magnetic field (B) and the circumstellar density (A) and their evolution over time. Its other main feature is its object-oriented architecture that sets a standard on how to record the data regarding the non-thermal emission of radio sources and at the same time provides an easily extendible structure designed to evolve. The design of the architecture makes it easy to use and to extend, allowing researchers all over the world to download RadioPy and implement their own models and methods easily.

The first version of this package contains 15 classes and 126 methods (version: June 2017) and provides robust algorithms for fitting spectra to an approximation of the Synchrotron Self-Absorption (SSA) model. The development of those algorithms and the goodness of the approximation are reasonable and they are discussed in the present document. A comparison between the results in the literature and those obtained with RadioPy for a case study of the radio evolution of supernova SN2011dh proved the capabilities of the software tool.

1 Introduction

1.1 Background

The field of radio astrophysics was born after the first detection of radio waves from the Milky Way by Karl Jansky in 1932 [1]. Thereafter, radio astrophysics studies astronomical objects at radio frequencies. At first its purpose was to give another point of view to the celestial objects (such as stars, galaxies or super novae) that can be seen in other frequencies, like in the visible spectrum, adding new information. Soon new cosmic entities were discovered and were identified as new sources of radio emission: radiogalaxies, quasars, masers and others. Those discoveries, along with the cosmic microwave radiation, boosted the reach and importance of radio astrophysics.

With time, researchers discovered and improved methods to gather and analyse the information written in the sky. In order to understand the radio emission and to be capable of inferring information about it, research focused on the theoretical basis of the emission itself: its origin and what factors influence it. Several authors made their contributions like the work of Ginzburg and Syrovatskii [2] on the origin of cosmic radiation during the 1950s. A major publication in 1962 shed light on this issue (Kardashev 1962 [3]), proposing several models for synchrotron radio emission. A few years later, and following the line of Kardashev's work, a great contribution in that regard was made by the work of Andrzej G. Pacholczyk. Based on the previous work of other authors like Ginzburg and Syrovatskii summarized in the annual review of astronomy and astrophysics [2][4], he published his book "Radio Astrophysics" [5] in 1970: an extensive, unprecedented work on the radio astrophysical measurements of synchrotron radiation produced by ultrarelativistic electrons accelerated by the magnetic fields present in plasma. Pacholczyk had brilliant skills in mathematics that allowed him to develop really complex models and equations in his extended, thorough work. The complexity of the non-linear equations and the amount of work accomplished in the book made his work unparalleled.

A major breakthrough was led by the technological advances in the fields of computers and numerical analysis. Those technologies allowed researchers to numerically solve the complex equations theoretically stated by their predecessors and opened a new access to knowledge. Papers appeared in such direction as the ones from Chevalier (1998) [6], treating the Synchrotron Self-Absorption (SSA) model for radio supernovae with numerical analysis. Years passed and researchers embraced the computer techniques, especially in the fields based on complex non-linear equations, like this one. Close in time, Murgia et al. [7] published a studio of compact radio sources emitting a steep spectrum, using a model based on the theory proposed by Kardashev (1962)[3] with some modifications. Later, the thesis work from di Gennaro (2015) [8] focused on the study of radio galaxies using a model derived from the works of Kardashev[3], Pacholzyk[5] and others. Resuming Chevalier's work on supernovae, a remarkable investigation for the framework of this project is the article from Krauss et al. (2012) [9] in which they study the radio evolution of a supernova using the SSA model.

As common denominator, the research needs of a computer tool to process the information and run the models. Moreover each researcher had to build their own numerical methods. This happens because of the lack of a standardized, free, open source software tool for radio emission.

1.2 Motivation

RadioPy was conceived as a software tool for nonthermal radio emission analysis. That field operates on the basis of complex equations and models that depend on several variables with exponential and logarithmic treatments that are nearly impossible to work with analytically. Usual fit methods for widespread languages in data processing like Python's LMFIT package [10] are not fully useful since in many cases their algorithms do not converge while solving such complex equations. Thus researchers are in need of developing their own fitting algorithms in order to interpret the information coming from the sky. This drives lots of time away from the actual progress of the research.

RadioPy was conceived to fill that gap by providing a standard, user-friendly and robust software tool that, given data of flux density and frequency, would perform a complete analysis and automatically create a report on the fit and the physical parameters of the model that it would be capable to infer. Such a tool would be of great use and would save time.

As it was said, there are many models and variations and consequently the reach of RadioPy's initial version is limited from the beginning. For it to survive and evolve, the collaboration of others is needed. Therefore, one of the big challenges when developing this tool has been to design it to be easily extendable, to be adaptive to any needs the researchers of so varied field might encounter.

In order to demonstrate the capacities of this software tool we will implement a full case study centred in supernovae and using the model proposed by Krauss et al. [9].

1.3 Goals

Here we show a list of aims that the initial version of RadioPy is required to accomplish:

- Range: to develop a software tool implementing fitting algorithms for radio astrophysics. Starting from observations of flux at different frequencies and times the tool should provide the best parameters for the equations modelling synchrotron radiation phenomena.
- Widespread implementation language: RadioPy is a Python implementation. A well-known and popular language, free, easy to access, resourceful and open source.
- Robust: defensive programming with safe algorithms that ensure convergence involving complex models.
- User-friendly: Radiopy is intended to perform a complete analysis automatizing all processes between the input of data and output results, allowing the researcher to concentrate more on the physical aspects under investigation.
- Extensible: radio astrophysics is a large and complex field. Researchers need to use lots of different models and RadioPy must satisfy the necessity. Thus it needs to facilitate the extensibility of its methods, give the possibility of mutating and absorbing different models.
- Evolve: RadioPy is an ambitious project, far exceeding the reach of a master thesis. Its goal is to help a large and varied community and therefore it would need time and collaboration from others. The structure of RadioPy, including a well-enginereed architecture and complete documentation, should facilitate its evolution.

Demo: the software has to prove itself to be useful. One of the goals of the present work is to perform a demonstration of RadioPy's skills. We implement a full case study centred in supernovae and using the model proposed by Krauss et al. [9].

1.4 Structure of this document

This document is divided in six sections plus the bibliography.

1. Introduction. Review of the background on the field of radio astrophysics, the motivation of the project and the goals that define its reach.
2. Synchrotron radio emission in astrophysics. Theoretical review of the radio frequency emission, the radio window and its role in the astronomical scenario. The origin of synchrotron in radio frequencies and expected behaviour, the Synchrotron Self-Absorption model and how to extract information from the radio observations.
3. SSA model fit. In this section, we explain how we fit the SSA model to actual data, going through the algorithm and the error treatment.
4. RadioPy architecture. The keys for achieving the goals of the project are its design and its architecture. This section contains explanations in that regard.
5. Case studies. Two examples that demonstrate the software's capabilities by fitting actual data from the publication of Krauss et al. (2012) [9] on the evolution of the radio emission from a supernova.

6. Conclusions & future. Summarized conclusions and a little discussion on the future of the project.

2 Radio emission in astrophysics

2.1 Radio astrophysics

2.1.1 The radio window

The Earth's atmosphere protects us from the radiation of the space by absorbing almost all wavelengths. Nevertheless there are two windows, or bands (speaking of frequency) that are practically transparent and thus those bands gain importance: radio (3 KHz to 300 GHz) and visible (450 to 750 THz). This transparency allows researchers to place their measuring equipment on the surface of the planet instead of having to place it in orbit.

The measurement of incoming radiation at radio frequencies requires large diameter for telescopes. The reason can be easily grasped by analysing a simple example: the Rayleigh criterion for the angular resolution of a telescope that follows Equation (1) [11][12]:

$$\sin(\theta) = 1.22\frac{\lambda}{D} \tag{1}$$

Where θ is the angular resolution, λ is the wavelength of the radiation and D is the diameter of the telescope. If we want an angular resolution of 10 arcsec; for a typical radio emission wavelength of $\lambda = 3$ cm (10 GHz), the resulting diameter of a telescope of such resolution would be $D = 754.9$ m. So the radio detection of distant objects needs large areas in which to install arrays of radiotelescopes that work together.

Another motivation for the study of the radio frequency in astrophysics is that many radio sources have been discovered in the universe. These sources emit predominantly in the radio frequency as a result of the curvature of relativistic electrons in presence of strong magnetic fields: synchrotron radiation.

2.1.2 Scenario

Radio sources can be classified in two groups: discrete and diffuse radio sources.

Discrete radio sources [5][8] are those whose emission comes from an identifiable object that usually has an optical counterpart. Some of them, such as supernova remnants, pulsating radio sources (pulsars), quasi-stellar sources (quasars), and radio galaxies are characterized by their non-thermal radio emission. In the case of the last three (pulsars, quasars and radio galaxies) the synchrotron emission is produced by an accretion disk near a supermassive black hole. As such is the case of Active Galactic Nucleus (AGN) for radio galaxies.

Diffuse radio sources [8] have also been discovered via synchrotron radiation with no optical counterpart. These sources provide the best evidence of the presence of vast magnetic fields across the large distances between galaxies within a cluster. They are classified by their size and location into halos, relics and mini-halos. Their common denominator is their very low surface brightness and their very steep spectrum.

2.2 Radio emission mechanisms

2.2.1 Thermal radio emission

All objects with temperatures above absolute zero have some internal motion: the atoms in solids are vibrating and the molecules in gases are moving around and bumping into each other. The hotter the body, the faster the vibrations or the more collisions occur, accelerating and decelerating according to its kinetic energy. Electromagnetic waves are produced whenever charged particles are accelerated and therefore all bodies with temperature above absolute zero emit EM waves that depend on the temperature. This radiation emission associated to the temperature is called "thermal emission".

Thermodynamics sets a limitation to the temperature through a universal principle that states the conditions for the thermal equilibrium. The thermal emission is also limited by this principle as it depends on the kinetic energy of the emitting body. Famous examples of thermal radiation are the black-body spectrum and free-free radiation (bremsstrahlung).

2.2.2 Non-thermal radio emission

It's been said that all bodies emit thermal radiation and it depends on the kinetic energy of the components of such body. Nevertheless it occurs that charged particles can be accelerated by other means such as the interaction with an electromagnetic field. Then an electromagnetic radiation will be produced but its energy would not depend on the temperature –internal kinetic energy of the body- and hence the name of "non-thermal" emission. In radio astrophysics two main non-thermal emission processes are important:

Synchrotron radiation is produced by the energy loss of charged particles that move in presence of a magnetic field. The interaction between such charges and the magnetic field curves the trajectory of the particle's movement causing deceleration and thus the production of electromagnetic radiation. In radio astrophysics the synchrotron emission in radio frequency occurs when electrons accelerated to relativistic velocities interact with intense magnetic fields.

The inverse Compton effect explains the augment of the energy of photons through the collision with a very energetic electron–inversely than the "regular" Compton Effect-. In radio astrophysics this effect occurs when having relativistic electrons within a medium with very high radiation density such as the core of supernova or an AGN. Lower radiation densities would prevent the electron and photons to encounter and for so it is required a medium with an exceptional density of radiation.

2.3 Synchrotron radio emission of relativistic electrons

2.3.1 Relativistic electron losses

The bulk of the radio emission in many astrophysical scenarios is generated through the energy loss of accelerated electrons. The function φ for the energy loss is proportional to the energy itself: the more velocity the more energy is lost. For better understanding of the radiation mechanisms we will discuss the different terms of Equation (2) for the relativistic electron energy losses [3][5][8]:

$$\varphi(E) = -\zeta - \eta E - \xi E^2 \tag{2}$$

This equation describes the function for relativistic electron energy (E) losses. The three terms of the equation are explained below:

Energy losses caused by ionization of the surrounding medium:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\zeta$$

This term is a constant whose value depends on the composition of the surrounding medium as well as its thermodynamic properties. Nevertheless it presents no dependency with the energy of the electrons themselves.

Energy losses caused by bremsstrahlung/free-free radiation:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\eta E$$

The free-free radiation (also called bremsstrahlung) is caused by the generation of photons because of the interaction between the electrons and the nuclei of the surrounding medium. The deceleration of electrons due to the interaction with other charged particles (such as nuclei) generates this free-free radiation, also called "bremsstrahlung". The energy of the created photons is comparable to the energy of the electron itself and therefore these losses are not continuous: an electron loses practically all its energy in a single interaction with a nucleus. It is "thermal" radiation loss since it comes from the kinetic energy of the electrons.

The last term in the equation encompasses the radiation losses from both synchrotron radiation and the inverse Compton Effect (non-thermal emission) since they are both proportional to the square of the energy:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -(\xi_S + \xi_C)E^2$$

Synchrotron radiation losses (ξ_S) depend on the transversal component of the magnetic field whereas the inverse Compton (ξ_C) depends on the radiation energy density in the source. These are "non-thermal" energy losses, as the temperature associated with this emission is typically orders of magnitude larger than the temperature associated with the kinetic energy of the electron plasma where the emission is produced. In the framework of this project we focus in the synchrotron emission for the radio sources we study in the demonstration are extensive and thus they do not have high enough radiation density to produce a dominant inverse Compton effect.

The behaviour of the energy losses for relativistic electrons presents three stages: low frequencies where ionization losses are dominant; intermediate frequencies where free-free radiation is predominant and high frequencies where the non-thermal losses (synchrotron and inverse Compton) dominate.

2.3.2 Synchrotron radio emission

As mentioned before, the radio emission for many celestial objects is actually synchrotron radiation (non-thermal). Here we will limit ourselves to discuss the synchrotron self-absorption (SSA) in radio astrophysical scenarios and try to find an equation that relates flux density (a measurable physical magnitude) and observing frequencies (as indicator of the energies of the radiation).

A key magnitude is the so-called "brightness" or "intrinsic intensity" of the source. Since the spectra shows the behaviour at different energies it is interesting to treat monochromatic measurements, i.e., intrinsic intensity per frequency, I_ν. This magnitude is intrinsic to the source and thus it does not depend on the observer at all (nor on the distance between them). The relation between the intrinsic intensity and the intrinsic brightness detected by the observer is given by the equation of transfer of radiation that describes the change in energy of the radiation because of passing through an element of matter of thickness s. This change in energy must be equal to the difference in the rate of emission and absorption within the element.

In order to describe the synchrotron spectra caused by a distribution of relativistic electrons Pacholczyk [5] uses in his formalism the so called source function: a ratio between the emission ε_ν –Equation (4)- and absorption κ_ν –Equation (5)- coefficients. These two are in turn functions that depend on the distribution of

electrons itself. As assumed by Pacholzcyk [5], Kardashev [3], Murgia et al. [7], di Gennaro [8], Chevalier [6] and Krauss el al. [9] the non-thermal radiation of radio sources is well approximated as the synchrotron spectra of a power-law distribution of electrons:

$$N(E) = N_0 \cdot E^{-p} \tag{3}$$

Such a distribution describes the number of electrons N as function of the energy, E, the number of injected electrons, N_0, and the exponent, p.

It can be shown [5] that the expressions for the emission (ε_ν) and absorption coefficients (κ_ν) per frequency ν of a power-law distribution of electrons are the following:

$$\varepsilon_\nu = c_5(p) \cdot N_0 \cdot H_\perp^{\frac{p+1}{2}} \cdot \left(\frac{\nu}{2c_1}\right)^{\frac{1-p}{2}} \tag{4}$$

$$\kappa_\nu = c_6(p) \cdot N_0 \cdot H_\perp^{\frac{p+2}{2}} \cdot \left(\frac{\nu}{2c_1}\right)^{\frac{-(p+4)}{2}} \tag{5}$$

These Equations (4) and (5) depend on the transversal magnetic field $H_\perp = H \sin\vartheta$ (pitch angle ϑ is the angle between the direction of the magnetic field and the direction towards the observer) and the frequency ν. In the expression there are also dependencies on the functions $c_5(p)$ and $c_6(p)$ and constant c_1, all of them defined at Pacholczyk's book [5], appendix 2 as:

$$c_5(p) = \frac{\sqrt{3} \cdot e^3}{16\pi \cdot mc^2}\left(\frac{p+\frac{7}{3}}{p+1}\right) \cdot \Gamma\left(\frac{3p-1}{12}\right) \cdot \Gamma\left(\frac{3p+7}{12}\right)$$

$$c_6(p) = \frac{\sqrt{3}\pi}{72} e m^5 c^{10} \cdot \left(p+\frac{10}{3}\right) \cdot \Gamma\left(\frac{3p+2}{12}\right) \cdot \Gamma\left(\frac{3p+10}{12}\right)$$

$$c_1 = \frac{3e}{4\pi m^3 c^5}$$

Where p is the exponent of the power-law distribution of electrons, m is the electron rest mass, c is the speed of light, e is the elementary charge and Γ is the gamma function.

From the ratio between the coefficients we get the so called source function S_ν per frequency [5]:

$$S_\nu = \frac{\varepsilon_\nu}{\kappa_\nu} = \frac{c_5(p) N_0 H_\perp^{\frac{p+1}{2}} \left(\frac{\nu}{2c_1}\right)^{\frac{1-p}{2}}}{c_6(p) N_0 H_\perp^{\frac{p+2}{2}} \left(\frac{\nu}{2c_1}\right)^{\frac{-(p+4)}{2}}} = \frac{c_5(p)}{c_6(p)} \cdot (H_\perp)^{-\frac{1}{2}} \cdot \left(\frac{\nu}{2c_1}\right)^{\frac{5}{2}} \tag{6}$$

This expression is key to describing the transfer equation. It also depends on the optical thickness τ_ν defined as:

$$\tau_\nu = \int_0^s \kappa_\nu \, \mathrm{d}s$$

The optical thickness describes the absorption for a given frequency due to passing through an element of thickness s.

Given all the above stated elements now we are in position to write the transfer equation:

$$I_\nu(\tau_\nu) = S_\nu(1 - e^{-\tau_\nu}) \tag{7}$$

2.3.3 Synchrotron Self-Absorption (SSA)

Optically thick case

Condition $\tau_\nu \gg 1$.

$$I_\nu(\tau_\nu) \approx S_\nu$$

The source function for a power-law distribution of electrons is defined by Equation (6):

$$I_\nu(\tau_\nu) \approx S_\nu = \frac{\varepsilon_\nu}{\kappa_\nu} = \frac{c_5(p) N_0 H_\perp^{\frac{p+1}{2}} \left(\frac{\nu}{2c_1}\right)^{\frac{1-p}{2}}}{c_6(p) N_0 H_\perp^{\frac{p+2}{2}} \left(\frac{\nu}{2c_1}\right)^{\frac{-(p+4)}{2}}} = \frac{c_5(p)}{c_6(p)} \cdot (H_\perp)^{-\frac{1}{2}} \cdot \left(\frac{\nu}{2c_1}\right)^{\frac{5}{2}}$$

In conclusion we can see that in the optically thick case where $\tau_\nu \gg 1$ we have that

$$I_\nu(\tau_\nu) \propto \nu^{5/2}$$

This is a remarkable result. The intensity of the synchrotron radiation, in the optically thick case, does only depend on the frequency, and is independent of both the power-law distribution of relativistic electrons and the magnetic field.

Optically thin case

Condition $\tau_\nu \ll 1$. Expanding the exponential $e^{-\tau_\nu}$ term as a Taylor series Equation (7) can be written as follows:

$$I_\nu(\tau_\nu) \approx S_\nu \cdot \tau_\nu - \frac{S_\nu \cdot \tau_\nu^2}{2} + \frac{S_\nu \cdot \tau_\nu^3}{6} + \cdots$$

Since $\tau_\nu \ll 1$, higher powers of τ_ν result in smaller numbers so the successive terms of the expansion would wane on and on. Thus it is a good approximation to keep just the first of the terms, having:

$$I_\nu(\tau_\nu) \approx S_\nu \cdot \tau_\nu$$

The intensity of the synchrotron radiation is equal to the source function times the optical depth. Assuming in the expression above for the optical depth that the opacity κ_ν is constant through the region s, we can write $\tau_\nu = \kappa_\nu \cdot s$, and so the final expression for the synchrotron intensity becomes:

$$I_\nu(\tau_\nu) \approx S_\nu \cdot \tau_\nu = \frac{\varepsilon_\nu}{\kappa_\nu} \cdot \kappa_\nu \cdot s = \varepsilon_\nu \cdot s = c_5(p)\, s\, N_0\, H_\perp^\alpha \left(\frac{\nu}{2c_1}\right)^{-\alpha}$$

Where $\alpha = \frac{p-1}{2}$

We can see that in the optically thin case where $\tau_\nu \ll 1$ we have that

$$I_\nu(\tau_\nu) \propto \nu^{-\alpha}$$

This is a fundamental result. In order to recover the whole information about the population of relativistic electrons, one has to observe the source at several frequencies, and high enough so as to be sure the source is

optically thin. The observed spectral index α directly tells us the power-law index of the population of relativistic electrons.

When frequencies (i.e., energies) grow higher the losses start to rapidly increase (as shown in Equation (2) which states that $\mathrm{d}E/\mathrm{d}t = -\xi \cdot E^2$) causing a breakdown in the spectrum (its maximum) and then decreasing the emission intensity.

2.3.4 Flux vs frequency equation for SSA

Equation (7) describes the intrinsic intensity I_ν but what we need is an expression depending on the frequency $I_\nu(\nu)$. Pacholczyk [5] calculates the transfer equation over frequency for a power-law distribution of electrons as:

$$I_\nu = S(\nu_\tau) \cdot J\left(\frac{\nu}{\nu_\tau}, p\right) \tag{8}$$

Where

$$S(\nu_\tau) = \frac{c_5(p)}{c_6(p)}(H_\perp)^{-1/2}\left(\frac{\nu_\tau}{2c_1}\right)^{5/2}$$

$$J\left(\frac{\nu}{\nu_\tau}, p\right) = \left(\frac{\nu}{\nu_\tau}\right)^{\frac{5}{2}}\left(1 - e^{-\frac{\nu}{\nu_\tau}^{-\frac{p+4}{2}}}\right)$$

$$\nu_\tau = 2c_1 \cdot (s_x \cdot c_6(p))^{\frac{2}{p+4}} \cdot N_0^{\frac{2}{p+4}} \cdot H_\perp^{\frac{p+2}{p+4}}$$

These expressions depend on a frequency ν_τ defined as the frequency at which the optical thickness is equal to one: $\tau(\nu_\tau) = 1$. This frequency depends on the number of injected electrons N_0, the transversal magnetic field $H_\perp$, the exponent p for the power-law distribution of electrons and the extent of the radiating region s_x.

Moreover what researchers actually measure from the universe is flux density: a magnitude related to the intrinsic intensity but whose readings depend on the distance. From the expressions present at Pacholczyk's book [5] and also from the work of Kardashev [3] it is possible to develop such expression.

The relation between monochromatic flux density F_ν and intrinsic intensity I_ν is given by Equation (9) [5]:

$$F_\nu = \int I_\nu \cos\phi \, d\Omega \tag{9}$$

where ϕ is the angle between the source-observer direction and the normal to the detecting surface and $d\Omega$ is the solid angle differential. At almost every discrete radio sources we can approximate $\cos\phi \approx 1$ [5]. Substituting the solid angle differential for its alternate form we have the following Equation (10):

$$F_\nu = \frac{I_\nu}{d^2} \cdot \int_A dA \tag{10}$$

where d is the distance from the source to the observer, and A is the observed area of the source. Hereby we rename the observed area as θ, a measurable magnitude related to the angular size. Substituting the expression for intrinsic brightness given Equation (8) at Equation (10) we obtain:

$$F_\nu = \frac{\theta}{d^2} S(\nu_\tau) \cdot J\left(\frac{\nu}{\nu_\tau}, p\right)$$

In a more complete form, given the equivalences explained at Equation (8), we have Equation (11) below:

$$F_\nu = \frac{\theta c_5(p)}{d^2 c_6(p) \sqrt{H_\perp} (2c_1)^{\frac{5}{2}}} \nu^{\frac{5}{2}} \cdot \left[1 - e^{-\frac{\nu}{\nu_\tau}^{-\frac{p+4}{2}}}\right] \tag{11}$$

This expression essentially describes a flux density vs frequency function whose shape behaves as expected at high and low frequencies (as seen in 2.2.1):

$$F_\nu \propto \left\{ \begin{matrix} \nu^{\frac{5}{2}} \text{ for } \nu \downarrow \\ \nu^{-\alpha} \text{ for } \uparrow \nu \end{matrix} \right\} \text{ with } \alpha = \frac{p-1}{2}$$

Nevertheless the full developed form of this equation is more complicated since we should substitute the equivalence for ν_τ as well, having the full and complete flux vs frequency expression for the synchrotron self-absorption model, Equation (12):

$$F_\nu = \frac{\theta c_5(p)}{d^2 c_6(p) \sqrt{H_\perp} (2c_1)^{\frac{5}{2}}} \nu^{\frac{5}{2}} \cdot \left[1 - e^{-\frac{\nu}{2c_1 \cdot (s_x \cdot c_6(p))^{\frac{2}{p+4}} \cdot N_0^{\frac{2}{p+4}} \cdot H_\perp^{\frac{p+2}{p+4}}}^{-\frac{p+4}{2}}}\right] \tag{12}$$

Essentially a non-linear function with eight parameters, some of them presenting non-linear interdependencies among them (as it is the case of exponent p).

2.4 SSA approximation

Subsection 2.3 went through the theoretical basis of the non-thermal radiation of a power-law distribution of electrons, resulting in a complex expression (Equation (12)) for the synchrotron self-absorption (SSA) model. When applying this basis to actual measurements researchers elaborated their own models by making approximations.

Within this project we built an analysis tool and, in order to demonstrate its functioning we focused on one of the many models present in the formalism: the SSA model for supernovae radio evolution proposed by Krauss et al. (2012) [9]. The theoretical model is an approximation of Equation (12) based in turn on Chevalier's research of 1998 [6]. Literarily taken from Krauss et al.'s paper (reference [9]) Equation (13) presents a SSA approximated model:

$$S(\nu) = 1.582\, S_{\nu_\tau} \left(\frac{\nu}{\nu_\tau}\right)^{5/2} \cdot \left\{1 - e^{\left[-\left(\frac{\nu}{\nu_\tau}\right)^{\frac{-(p+4)}{2}}\right]}\right\} \tag{13}$$

In their formalism Krauss et al. named as S the monochromatic flux F_ν. For the remainder of the document we adopt that formalism so "S" will now be the flux density per frequency unit F_ν instead of the source function (Equation (6)).

Equation (13) has three free parameters: the exponent p of the power-law distribution of electrons, the flux density $S_{\nu\tau}$ which is the flux density at the frequency ν_τ for which the optical thickness is one (condition

$\tau_\nu(\nu_\tau) = 1$) and ν_τ, which is precisely that frequency. The most daring approach taken by this model is to encompass ν_τ as a free parameter obviating its dependencies (shown at Equation (8)). Nevertheless the results of fitting this model to the actual data were good [9], indicating that this is a reasonable approximation.

2.3.1 Notation for parameters of the SSA approximation

While implementing the model in Python we had to establish a different notation for the parameters of Equation (13). The elected ones were as follows:

- Exponent for the power-law distribution of electrons: named parameter "p", from the original p.
- Flux density for frequency ν_τ: named "st" as for S_τ, from the original S_{ν_τ}.
- Frequency at which the optical thickness is one: named "vt", from the original ν_τ

2.3.2 Goodness of the SSA approximation

Results from Krauss et al. [9] indicate that the approximation for Equation (12) made by them (Equation (13)) is a good one since it accurately describes the reality. Nevertheless, before taking for granted the goodness of the approximation and start implementing this model in Python we made a simple test by running the equation (13) for an array of frequency values (25 points between 1 GHz and 50 GHz) with parameters set to be exponent p = 3.0; flux density st = 7.0 mJy and frequency vt = 4.0 GHz. The model example is shown in Figure 1 below:

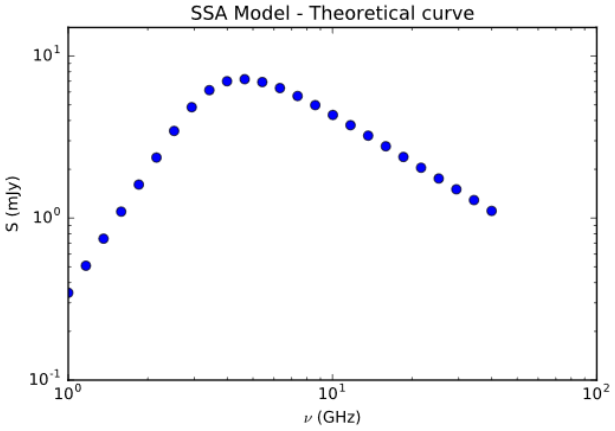

Figure 1. *Theoretical flux densities (in mJy) vs frequency (in GHz) represented at logarithmic scale. These theoretical points were obtained by creating a frequency array and applying Equation (13) using the following values for the free parameters: p = 3; st = 7; vt = 4.*

Figure 1 shows that Equation (13) behaves as expected with two well-defined regions: low frequencies with positive, constant slope (self-absorption) and high frequencies with constant, negative slope.

Raw calculations on the graphical showed that these slopes where:

- Low frequencies – slope: 2.49
- High frequencies – slope: 0.98 → p = 2.96

For our case with p = 3 slope α should be $\alpha = 1$. The slope for the self-absorption part (low frequencies) should be 5/2 (see subsection 2.2.1). Calculations were made removing 8 points in the area between low and high frequencies (next to the peak) and showed results according with the theory (section 2.2.1), proving that the approximation is reasonable.

2.4 Estimation of physical magnitudes for supernovae

In their paper Krauss et al. [9] state three expressions for physical magnitudes of supernovae that can be calculated from the fit results of their model (Equation (13)). The calculus of such magnitudes depends on two

parameters: $S_{\nu_{op}}$ and ν_{op}, respectively flux density of the observed maximum peak and the frequency for that maximum flux. The expressions are the following, literally taken from their paper [9]:

Equation (14) for the shockwave radius (R) of the supernova (in centimetres):

$$R = 3.9 \cdot 10^{14} \cdot \alpha^{-\frac{1}{19}} \cdot \left(\frac{f}{0.5}\right)^{-\frac{1}{19}} \cdot \left(\frac{d}{\text{Mpc}}\right)^{\frac{18}{19}} \cdot \left(\frac{S_{\nu_{op}}}{\text{mJy}}\right)^{\frac{9}{19}} \cdot \left(\frac{\nu_{op}}{5\ \text{GHz}}\right)^{-1}\ \text{cm} \tag{14}$$

Equation (15) for the intensity of the magnetic field (B) accelerating the electrons (in gauss):

$$B = 1.0 \cdot \alpha^{-\frac{4}{19}} \cdot \left(\frac{f}{0.5}\right)^{-\frac{4}{19}} \cdot \left(\frac{d}{\text{Mpc}}\right)^{\frac{-4}{19}} \cdot \left(\frac{S_{\nu_{op}}}{\text{mJy}}\right)^{\frac{-2}{19}} \cdot \frac{\nu_{op}}{5\ \text{GHz}}\ \text{G} \tag{15}$$

Equation (16) for the circumstellar wind density parametrized as $A_* = A/(5 \cdot 10^{-11}\ \text{g/cm})$:

$$A_* = 0.82 \cdot \alpha^{-\frac{8}{19}} \cdot \left(\frac{\epsilon_B}{0.1}\right)^{-1} \cdot \left(\frac{f}{0.5}\right)^{-\frac{8}{19}} \cdot \left(\frac{d}{\text{Mpc}}\right)^{\frac{-8}{19}} \cdot \left(\frac{S_{\nu_{op}}}{\text{mJy}}\right)^{\frac{-4}{19}} \cdot \left(\frac{\nu_{op}}{5\ \text{GHz}}\right)^{2} \cdot \left(\frac{t}{10\ \text{day}}\right)^{2} \tag{16}$$

These expressions depend on several variables that are explained here: α is in this case the ratio of electron-to-magnetic energy densities (do not mistake with exponent $\alpha = (p-1)/2$, section 2.2.1); f is the filling factor of the emitting material, d is again the distance (in Mpc), ϵ_B is the fraction of kinetic-to-magnetic converted energy density and t is the age of the supernova (in days)

Some explanation is needed for variables α, f and ϵ_B. Krauss et al. [9] gave them values based on both theoretical and empirical basis: if we assume equipartition then $\alpha = 1$; we assume $f = 0.5$ as it was found by Bartel et al. [13] and we assume $\epsilon_B = 0.1$ as Krauss et al. [9] did.

As a conclusion, we can see from Equations (14), (15) and (16) is that from the flux vs frequency curve the part that matters to the calculations is the one around the maximum, the intermediate frequencies between low and high.

The process for getting physical information about the supernova is to fit the theoretical curve (given by Equation (13)) to the measured data to get the best values for the free parameters "p", "st" and "vt". Once we know those values we calculate a theoretical curve (similar to the one shown in Figure 1 but with many more frequency points) and from that theoretical curve we extract its maximum flux $S_{\nu_{op}}$ (named in the implementation as "sm") and the frequency at which the curve attains its maximum ν_{op} (named "vm").

2.5 Brief comments on supernovae

As it has been said the current version 1 of RadioPy package implements algorithms for fitting synchrotron spectrum described by the SSA model and especially focuses in the case of supernovae (section 5.2).

A supernova is a gigantic explosion caused by the destruction of a massive star (more than 8 solar masses) and it is also the last stage of a star's life. Supernovae, as the catastrophic explosion of a star can occur for several reasons but the most common is the collapse of a massive star when the fuel to the thermonuclear reactions is lowering. Then the gravity comprises the material and it raises its temperature to an incredible level, triggering a titanic thermonuclear reaction that ignites an enormous explosion.

As a result a shockwave expands in all directions colliding with the surrounding medium and accelerating it to velocities of the order of 20000 km/s. The shockwave front sweeps the circumstellar medium and accelerates its components to high kinetic energies associated to high temperatures breaking up the atoms into a hot plasm. Thus within the shockwave front there are electrons accelerated to relativistic velocities and in the presence on intense magnetic fields which causes the synchrotron emission at the radio frequencies.

Understanding the origin of the synchrotron emission through the synchrotron self-absorption gives us direct information about the shockwave front as they are the radius, the intensity of the magnetic field and the circumstellar density -Equations (14), (15) and (16)-. The direct observation of the synchrotron spectrum gives also information on the distribution of electrons that cause the radio emission as we saw at section 2.4.

3 SSA model fit

3.0 Introduction to RadioPy's model fit

As it was said in the Introduction (section 1) the idea of RadioPy package limited to the scope of this project is to build an extensible and standard structure that considers and solves problems of synchrotron radio emission of astrophysical radio sources. In order to do so we took a case studio and the elected one was the case of supernova SN 2011dh, documented and studied in turn by Krauss et al. [9].

Thus we decided to use the SSA model approximation with three free parameters described at Equation (13), section 2.4, the first step was to search for an algorithm already implemented in python. The first decision was to look for programs that based their fitting algorithms on weighted least squares, not just regular mathematical regressions that do not consider the error of the measurements. Thus we tried the package called "LMFIT" [10] that uses the Levenberg-Marquardt algorithm [14] to fit data. The potential of this algorithm is that it considers several variables. Nevertheless, we found that this algorithm was not useful to us since it did not converge for cases where data and model were not strictly close. This happened for other methods implemented in the LMFIT package: Nelder-Mead, L-BFGS-B method, Powell's method, conjugate gradient, and COBYLA [10]. None of these methods would converge if the initial values of the parameters were not strictly close to the actual ones, nor if the theoretical model did not match the real data. It is common that the spectra is deformed by the actions of external effects and exhibit a shape like the theoretical curve but with distortion. Thus the need of a new method appeared, and the solution was to develop our own algorithm.

3.1 Dependencies among the model parameters

The first step on that path was to find out if there was any dependency between the three free parameters (parameter "p", flux "st" and frequency "vt"). Figure 2 shows three simulations in which we varied each parameter separately in order to see "if and how" they depended on each other. All simulations were performed on the same model example shown at section 2.4 (Figure 1).

Parameter "p" (left at Figure 2) dominates the slope of the right part of the curve (while the left part remains equal). It also shows that for a certain "p" the curve loses its maximum, turning into a monotonically increasing function. Parameter "st" (centre of Figure 2) dominates the ordinates axis (maximum flux) but it does not influence the slopes nor the abscises axis (frequency of the maximum flux). Parameter "vt" (right of Figure 2) dominates over the frequencies but does not influence the flux or the slope of the curve.

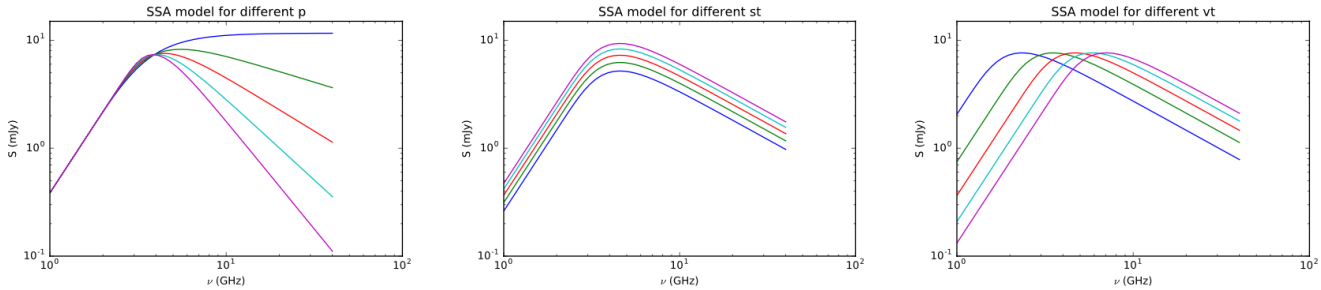

Figure 2. A *set of simulations of model example, section 2.4, where the free parameters p, st and vt were alternatively varied. Left – variation of exponent "p". Centre – variation of flux "st". Right – variation of frequency "vt". The three parameters seem to have a negligible dependence on each other.*

For what we can see in Figure 2 we have in conclusion that the interdependences of the free parameters are very small. A derived conclusion is that the approximation of frequency "vt" as a free parameter is well justified. In fact, the results shown by Figure 2 indicate that taking "vt" as a free parameter is a good approximation since its impact on the two other parameters is almost unnoticeable.

Since the model parameters present no appreciable interdependences we do not need an algorithm for simultaneously fitting several variables. We could search for a method capable of fitting a single independent variable. Thus the decision we called was to fit each parameter independently and alternatively. A sequence of iterations of that process would allow us to take into account any small interdependencies among the free parameters, and would lead to increasingly better results.

3.2 Least squares algorithm

The next call is to decide what the algorithm should minimize. It has been already discussed we searched for a weighted least squares algorithm and coherently with the reasoning we had been doing, it was decided to minimize the weighted chi-square function described by Equation (17) below [15][16][17]:

$$\chi^2 = \sum_N \frac{(M_i - T_i)^2}{\sigma_i^2} \tag{17}$$

Where N is the number of measurements, M_i are the measured values, T_i the theoretical ones and σ_i is the standard deviation of each point (absolute error). This function gets close to zero when measured and theoretical values are close. The error weighting makes points with larger uncertainty to add a lesser contribution to the summation than the ones with smaller error., which contribute more to the total. It might be interesting to add that the chi-square function that our method actually uses is calculated with a logarithmic scale. The justification of that decision is to give fairer importance to all the data, weighting them equally regardless of whether the measurements correspond to high or low frequencies.

So from the last paragraph we can summarize that a good fitting would be the function whose parameters "p", "st" and "vt" make the logarithmic weighted chi-square function (χ^2 function) attain its minimum. One way to calculate this minimum is through calculating the roots of the derivative of chi-square over the model equation. Since the parameters are not dependant on each other (as discussed at section 3.1) we can find the roots of the derivative of chi-square successively over each of the parameters. We calculated the differentiation with a numeric derivative method described by expression (18) below [18][16]:

$$\frac{\mathrm{d}\chi^2(x)}{\mathrm{d}x} = \frac{\chi^2(x+h) - \chi^2(x-h)}{2 \cdot h} \tag{18}$$

Where x is a variable, and in our case, a parameter and h is a very small number that tends to zero. In order to find the roots it is useful to have a hint about the behaviour of these derivatives. Figure 3 shows the differentiation of the chi-square function over the three parameters, separately.

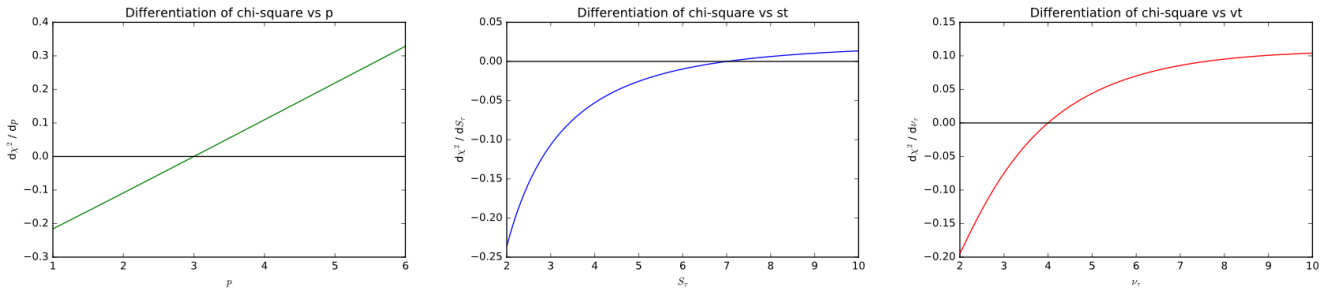

Figure 3. *A set of simulations of the differentiation of function chi-square over parameters "p", "st" and "vt" respectively. They all present a root and are continuous and derivable functions.*

Figure 3 shows that the χ^2 derivative over parameters "p", "st" and "vt" are all continuous and derivable functions in the regions of interest, which ensures a correct operation of the method to find the root of the non-linear equation.

3.3 Non-linear root solutions: the binary search algorithm

The next step was to implement a robust algorithm for finding roots of non-linear functions with one variable. There are several, well-known algorithms for that task and we decided to use the so-called binary search algorithm [15][18][16]. These methods numerically solve the equation $f(x) = 0$ for a the real variable x. Such function $f(x)$ can be non-linear as long as it presents at least one root in the interval of interest.

The binary search algorithm is also known as the bisection method, the interval halving method, the dichotomy method, binary chop or simply logarithmic search. It can be applied on a continuous function defined in an interval $[a, b*]$ where $f(a)$ and $f(b)$ have opposite signs. This algorithm works this way: assuming that values $f(a)$ and $f(b)$ have different signs, two initial values $f(a_1)$ and $f(b_1)$ are assigned. Then the algorithm calculates the half-point between them, $p_1 = (b_1 - a_1)/2$, and checks for the sign of the function there. The calculation of the second-iteration values a_2 and b_2 is as follows. If the sign at the half-point is the same as at value $f(a_1)$ then a_2 is made equal to the calculated half-point $a_2 = p_1$ while b_2 is kept equal to b_1. Oppositely, if the sign at the half-point is the same as at value $f(b_1)$ then b_2 is made equal to the calculated half-point $b_2 = p_1$ while a_2 is kept equal to a_1. An iteration of this operation will rapidly set the points a_n, b_n close to the root where $f(a_n) \cong f(b_n) \cong 0$. Figure 4 is a schematic representation of how does the bisection method work:

While programming this algorithm we must establish a condition in order to stop iterating the operation. Such condition would be that the normalized difference between points a_n, b_n is lower than a certain number that would be really small $(1 \cdot 10^{-5})$, considering that point to be the root of the equation. The result of this algorithm would be a value for a fit parameter that corresponds to the minimised χ^2, i.e., a fit parameter which makes the theoretical function to be closer to the actual measured data.

Starting from a set of initial guesses for the model parameters the binary search method is sequentially applied to fit the three parameters of the model: "st", "vt" and "p". In order to minimize the effects of small interdependencies between these free parameters we iterate this method alternatively switching the fit parameters until the value of χ^2 can no longer be minimised.

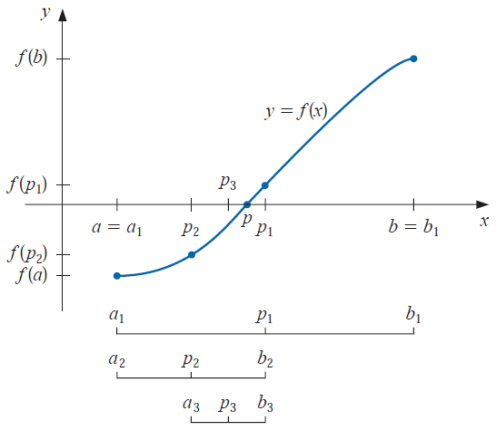

Figure 4. *Schematic functioning of the bisection method –or binary search algorithm-. Two points a_1, b_1 are defined with opposite function signs. The half-point between the two values is obtained and new a_2, b_2 values are defined depending the sign of the half-point. Iterating this operation would lead the points a_n, b_n to be as close to root of the function as we want. Source: Richard L. Burden, J. Douglas Faires (2011) [17].*

This method has proven to be robust since the convergence of the algorithm is assured if the function presents a negative and a positive part. In our case the function is the differentiation of χ^2 over the model parameters "p", "st" (flux S_τ) and "vt" (frequency ν_τ), represented at Figure 3. As we can see in the figure such function fulfils the requirements of this algorithm.

The following pseudocode shows a simplified view of the fitting algorithms that we have designed:

```
algorithm binary_search(measured_data, sought_parameter, model_params) returns fitted_param
    a1=value smaller than the initial guess of the sought_parameter
    b1=value larger than the initial guess of the sought_parameter
    while |(b1-a1)/a1|>desired_accuracy
        halfpoint = (a1+b1)/2
        modelParams[soughtParam]=halfpoint
        func = chiSquareDerivative(measured_data, soughtParam, model_params)
        if func>0
            b1 = halfpoint
        else
            a1 = halfpoint
        end if
    end while
    return soughtValue = halfpoint
end algorithm
```

To see next part of the algorithm turn the page.

To make the algorithm robust we have included a maximum number of iterations for the fitting algorithm. We have also included provisions to automatically change the initial values used in the binary search if they do not lead to different signs of the chi square derivative.

Other algorithms were considered, for instance Newton's method or the false position method [15][18]. Although these methods may be faster than the binary search, Newton's method dos not guarantee convergence for all cases, and the false position method may have slow convergence in some cases. Therefore, we chose the bisection method for its robustness.

```
algorithm RadioPyFitting(measured_data) returns fitted_model_params
    st= initial guess for st= maximum observed flux
    vt= initial guess for vt= frequency of max observed flux
    p= initial guess for p= slope of right part of the flux curve
    repeat
        st=binary_search(measured_data, 'st', model_params)
        vt=binary_search(measured_data, 'vt', model_params)
        p=binary_search(measured_data, 'p', model_params)
    until chiSquare changed very little
    return (st, vt, p)
end algorithm
```

3.4 Goodness of fit

A spread and well-known method to evaluate the goodness of a fitting process is the so called reduced chi-square per degree of freedom [15][17]:

$$\chi_V^2 = \frac{\chi^2}{N - L} \tag{19}$$

Where χ^2 is the chi-square statistics (Equation (17)), N is the number of measurements and L is the number of parameters. The difference between the number of measurements and the number of parameters $V = N - L$ is defined as the number of degrees of freedom.

3.5 Error treatment

3.5.1 Errors derived from the fitting

The errors on the measurements are already considered in the definition of the weighted chi-square statistics (section 3.2).

The absolute error is calculated through the standard deviation of the regression [15][17] as the square-root of the variance:

$$\sigma_{fit} = \sqrt{\frac{1}{N} \cdot \sum_{N} (M_i - T_i)^2} \tag{20}$$

Where N is the number of measurements, M_i are the measured values, T_i the theoretical ones. This standard error can be taken as an absolute error of the flux values of the theoretical curve. So the absolute error of flux parameter st (S_τ) would be the standard error. But this error is actually an account for the whole model, so it is fair to assume that the relative error of all the parameters calculated by the fit would be the same, as they all derive from the variance of the fit [15][17]. So relative errors of parameters "p" and "vt" are assumed to be equal to the relative uncertainty of parameter "st":

$$\delta p = \delta vt = \delta st = \frac{\sigma_{fit}}{st} \tag{21}$$

Where δf stands for relative error of function f and σ_{fit} represents the standard error of the regression (and absolute error on the theoretical flux).

3.5.2 Uncertainty of the physical parameters

The error of the calculated physical magnitudes is treated in a different way than the error of the fitting, yet they are related. Magnitudes R (shockwave radius), B (intensity of the magnetic field) and A (circumstellar density) are calculated from analytical expressions (Equations (14), (15) and (16)) that depend on several independent variables –fit parameters "p", "st" and "vt" are among them-. The method for obtaining the uncertainties would be to propagate the errors of each of the independent variables. For that we use another well-known expression [15][17] for deriving errors of a generic function:

$$\Delta f(x,y,z,\dots) = \sqrt{\left(\frac{\partial f}{\partial x}\cdot\Delta x\right)^2 + \left(\frac{\partial f}{\partial y}\cdot\Delta y\right)^2 + \left(\frac{\partial f}{\partial z}\cdot\Delta z\right)^2 + \cdots} \tag{22}$$

4 RadioPy architecture

In this section we explain the structure of the developed software. It is divided into 8 subsections. The first two are focused on a general view of the architecture and its most important features. Sections 4.3 to 4.6, inclusive, explain the different modules present in the package. Section 4.7 presents an extra module outside the package in which Appendix 2 from Pacholzcyk's book [5] has been implemented in python. That package includes some constants and functions used in the theoretical formalisms that can be of use to the extensibility. Finally, Section 4.8 contains a description on the deployment of the RadioPy software modules and documentation.

4.1 General comments on the structure

RadioPy was designed following the Object Oriented Programming (OOP) architecture [19] and consequently all data and methods are defined in the classes of the package. The encapsulation of data together with the methods that implement the operations on them allows us to easily handle large amounts of data grouped as objects that can be easily passed as parameters to the different methods, instead of passing individual values. This design plays a major role in accomplishing some of the project's goals (section 1.3) such as usability (user-friendly), extensibility (see section 4.2) and future evolution capabilities. A well-engineered architecture is key to setting a standard structure of data recording and a work methodology for the posterior evolution and development of the package.

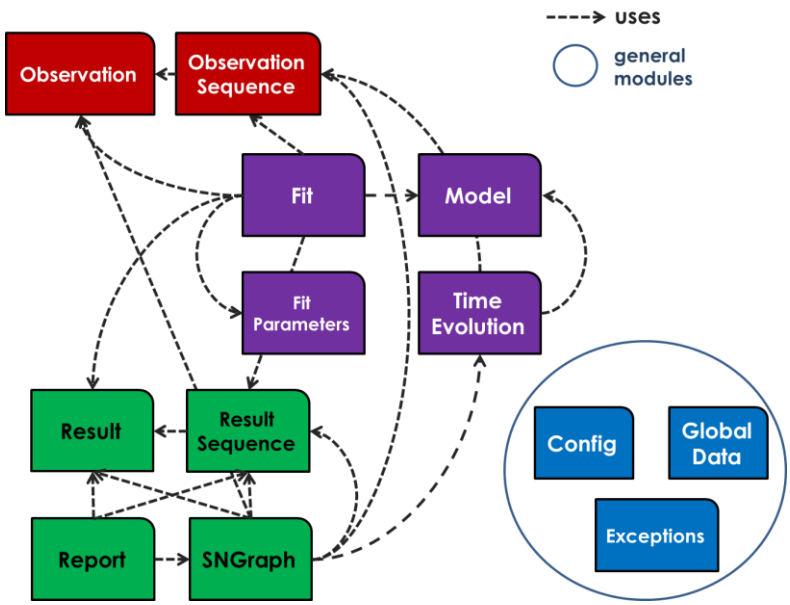

Figure 5. *Classes and modular structure of RadioPy package with the interrelations between classes. Represented in red are the "input modules", green for the "output modules", purple for the "fit modules" and blue for the general modules. Grey dotted lines indicate which class uses which other. The general classes are used virtually by all the others.*

Importing package RadioPy will grant access to all its 15 classes and 126 methods of current version 1. The use of the classes and methods gives RadioPy a great flexibility at the time of analysing and performing fits, as the package is customizable. A collaboration diagram showing the relations among the classes is shown in Figure 5. Sections 4.3 through 4.6 give more information on the modules and classes.

4.2 Extensibility

As it was said before, the package was designed for it to be extendible and to evolve. The key for achieving those goals is it architecture. Extension to other radio astrophysics models is easy because the classes containing the model itself and its fitting algorithms are designed to be written as extensions of existing classes through the OOP inheritance mechanism.

At the heart of a model fit we find a "Model" class, which contains the function that defines the theoretical curve to compare to the measured data, and a "Fit" class, in which the fitting algorithms and methods are implemented. For the sake of better extensibility two "abstract" classes were created:

- AbstractModel: a parent class for all models. It contains basic and general functions capable of calculating several important values and statistics such as the quadratic error, the weighted χ^2 and its derivatives. Any new extension of AbstractModel will inherit all the methods from the AbstractModel. The main function that defines the model itself (a $F_\nu(\nu)$ equation) is called fluxModel(). It is empty in the abstract class and any model that extends it must override it to implement its own equation. All the rest of the methods will be inherited. The concrete class that is provided in this version of RadioPy is:
  - KraussModel: an extension of AbstractModel that overrides the fluxModel() method implementing the equation proposed by Krauss et al. [9] (Equation (13)) representing the approximation to the SSA model focused on supernovae.
- AbstractFit: a parent class for all fit classes. It contains general functions on calculating parameters and performing fits (such as the binary search algorithm). Just as with the AbstractModel the main fitting method, called searchParams(), is empty and must be overridden. It also has methods to calculate the physical parameters by applying Equations (14), (15) and (16), and for fitting a sequence of epochs
  - SNFit: fitting process for supernovae (SN). It overrides method searchParams() to perform a fitting over an epoch using the algorithm presented at section 3.

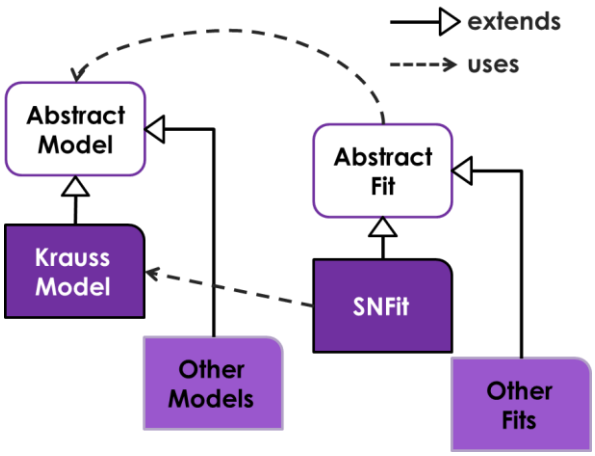

Figure 6. *RadioPy's extensibility. Each new model should be an extension of class AbstractModel. This parent class contains general methods that will be inherited, as for example the chi-square calculation and its derivative. For each extended model, there should be an extended fit class (from parent class AbstractFit) that will inherit methods such as the binary search algorithm and its iteration, or the calculus of physical magnitudes. The methods in the new subclasses should override the empty ones from their respective abstract classes. The figure shows KraussModel, and SNFit, which are the concrete classes already implemented in RadioPy's version 1. The bridge between these classes is established by invoking the init classmethod.*

As the fit uses the model to perform the adjustments those two classes need to be related. This is accomplished by invoking a special method of AbstractFit called "init", an initializer that sets which model the fit should use.

4.3 Configuration, Global Data & Exception modules

There are three general modules that are used by all the others in the package.

radiopy.config
Contains class SNConfig with configuration parameters for supernovae fitting that allow the user to modify the automatic behaviour of a fitting. Table 2 shows some of these parameters:

radiopy.global_data
This module contains the class GlobalData that contains some data that is general to all the experiment or set of observations but it has nothing to do with the fitting. Data such as the name of the event, the authors, title or the start date. The most important "global data" is the distance and its uncertainty, in megaparsec (Mpc).

radiopy.exceptions
This module contains several exception classes for notification of different contingencies that the methods may encounter.

Name	Default value	Description
enableErrCorr	True	Establish whether the error bars should be corrected in order to get a better fitting with reduced chi-square per degree of freedom close to 1.
enableFixedP	False	Establish whether the fitting algorithm (class SNFit) should fix parameter p or set it free.
fixedP	2.8	Default value for fixed p exponent of the power-law distribution of electrons.
fitAccuracy	0.0001	Accuracy factor for the root binary search. It might be useful to change it if fitting does not behave properly
alpha	1	Ratio of electron to magnetic energy densities. Assuming equipartition alpha = 1.
fillFactor	0.5	The filling factor of emitting material gives an idea of how many electrons of the total material contribute to the emission. We assume it to be 0.5 as found by Bartel et al. (2002) [13] for SN 1993J.
eB	0.1	The converted fraction of kinetic to magnetic energy density. eB was found to give consistent results with value eB = 0.1 at Krauss et al (2012) [9].
figurePath	'figures/'	Path where figures will be saved.
reportPath	'reports/'	Path where the reports will be saved.

Table 2. *Some attributes of class SNConfig. Changing the default value would lead to different ways of fitting and results.*

4.4 Input data structure modules

The input of the observed data is defined in two classes:

radiopy.observation
This module has class Observation, a data container for flux density and frequency points for an epoch of observation. It can record a list of frequencies, a list of fluxes and a list of uncertainties. It also contains the age

of the event, "t". It contains a couple of methods for "observing" the measured curve; a method to return the list-index for the observed peak and another one to calculate slopes at logarithmic scale between two indexes.

This class plays a major role since it is a data container. Fit methods that require flux and frequency measurements will receive as parameter an instance of class Observation instead of several arrays of data.

radiopy.observation_sequence
This module has class ObservationSequence that contains a list of Observation instances. Useful for sequence fittings and time evolution studies.

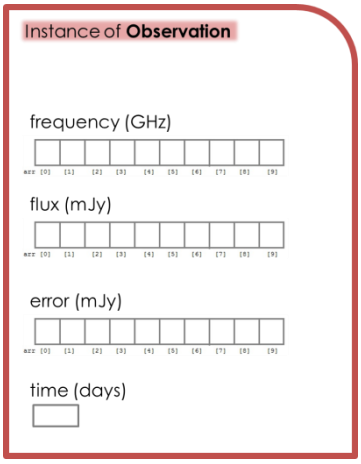

Figure 7. *Data structure of class Observation.*

Reading input data

Both clases Observation and ObservationSequence have methods for reading the data from a .csv file. This facilitates automatic reading of the data using a well-known file format. The .csv file has to be written in a special format. It must have 9 header lines with information on the experiment. The rest of the rows contain: time (in days); frequency (in GHz); flux density (in mJy) and flux density error (in mJy). A total of 4 values per row. Here we expose a general example for the accepted .csv format:

```
"TITLE: ","Name of the file/set of measurements"
"AUTHOR/S: ","Author1","Author2","Author3","+..."
"ORGANIZATION: ","Name of the organization/researcher affiliation"
"NAME OF THE EVENT: ","Type (supernova, radiogalaxy): code"
"START DATE OF THE EVENT: ","YYYY/MM/DD"
"DISTANCE TO THE EVENT (Mpc): ",float
"UNCERTAINTY ON THE DISTANCE (Mpc): ",float
"MEASURING EQUIPMENT: ","measuring equipment"
"t (days)","Freq (GHz)","Flux (mJy)","fluxErr (mJy)"
day1, freq1.1, flux1.1, fluxError1.1
day1, freq1.2, flux1.2, fluxError1.2
day1, freq1.3, flux1.3, fluxError1.3
day2, freq2.1, flux2.1, fluxError2.1
day2, freq2.2, flux2.2, fluxError2.2
.   ,    .   ,    .   ,      .
.   ,    .   ,    .   ,      .
.   ,    .   ,    .   ,      .
dayN, freqN.M, fluxN.M, fluxErrorN.M
```

4.5 Output data structure modules

The output modules category includes all those methods involved in the returning of data. There are four classes:

radiopy.result
Module for class Result. An homologous of container class "Observation", class Result records the values of the fit parameters, the fitting statistics and the values of the physical magnitudes for the observation at one epoch.

radiopy.result_sequence
Module for class ResultSequence. An homologous of "ObservationSequence" it contains a list of Result instances. In addition, an object of class ResultSequence can record the data with the time evolution of the physical parameters of the radio emission for its sequence of epochs and the average "p" exponent.

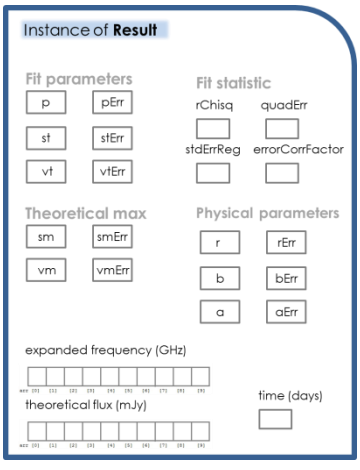

Figure 8. *Data structure of class Result.*

radiopy.report

Module for class Report. It contains methods for representing data in the console and also to generate .html and .pdf reports in a separate folder, given a ResultSequence as parameter.

radiopy.sn_graph

Module for class SNGraph. Its methods automatically represent in the Python console graphs of the fitting or the time evolution of the physical parameters. They can also save those figures in a separate folder under .svg or .png extensions.

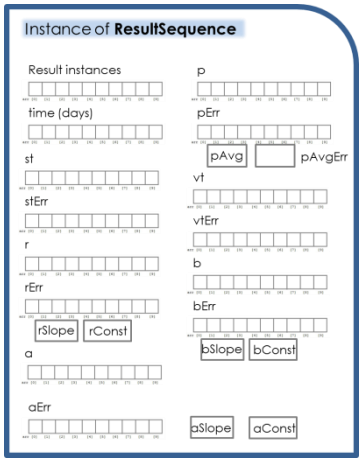

Figure 9. *Data structure of class ResultSequence.*

4.6 Fit modules

The modules involved in the algorithm and performing the fit:

radiopy.abstract_model

Module for class AbstractModel. The parent class for all models in RadioPy. It contains basic and general functions capable of calculating several important values and statistics such as the quadratic error, the weighted χ^2 and it derivatives.

radiopy.krauss_model

Module for class KraussModel. An extension of AbstractModel that overrides the empty model of AbstractModel with the equation proposed by Krauss et al. [9] (Equation (13)).

radiopy.abstract_fit

Module for class AbstractFit. The parent class for all fit classes in RadioPy. It contains general functions on calculating parameters, performing fits (such as the binary search algorithm) and calculating the physical parameters.

radiopy.sn_fit

Module for class SNFit. Contains the fitting process for supernovae (SN). It contains methods to perform a fitting over an epoch using the algorithm presented at section 3.

radiopy.time_evol

Module for class TimeEvol. It contains methods that, given a ResultSequence, calculate the time evolution for the physical parameters and the exponent "p". The time evolutions of the different parameters are fitted to straight lines using the LMFit python package.

4.7 Pacholczyk's constants and functions module

RadioPy has the objective of becoming a widespread tool for radio astrophysics. One of the most important references in the field is Pacholczyk's book [5] and thus many models would come from its equations. Pacholczyk uses in his formalism a series of constants and functions that serve as bridge and simplification for big, complex models. In order to make RadioPy more extendible we implemented Appendix 2 from Pacholzcyk (1970) [5] in a Python module containing the functions and constants following the nomenclature of the book. In addition, there are functions to simulate some of the figures appearing in the book.

While implementing this appendix a couple of errata were found in the book: see Appendix B.

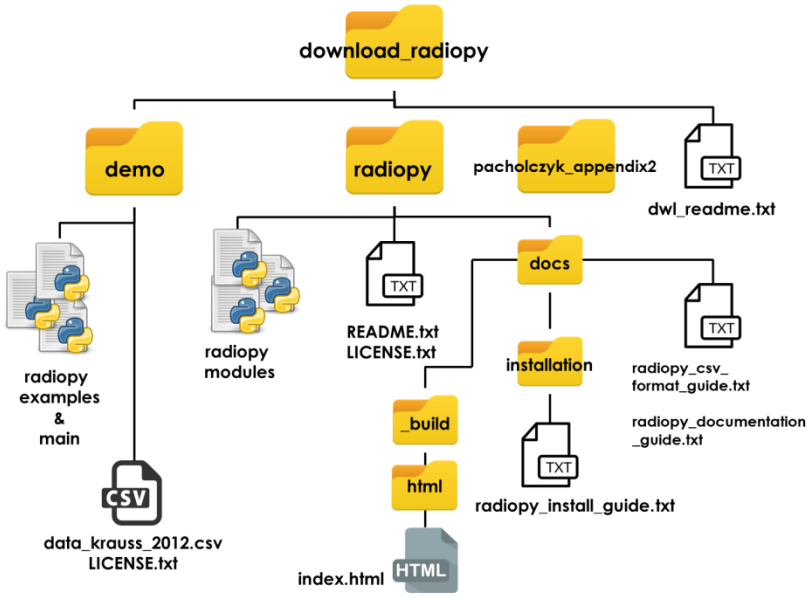

Figure 10. *This diagram shows what is inside the downloadable folder for RadioPy package "download_radiopy". You may start with "README.txt" and "radiopy_install:guide.txt". For viewing RadioPy's source code documentation you must open "index.html".*

4.8 Code and documentation structure

Finally, and to put a closure on Section 4, we explain a brief guide on the structure of the code and documentation released with RadioPy.

The package is downloaded in a compressed folder called "download_radiopy". Figure 10 schematically shows the insides of the folder. The figure is mostly self-explanatory. The most important documents are a README.txt file, an installation guide (radiopy_install_guide.txt), and an html document containing the package's source code documentation at: download_radiopy/radiopy/docs/_build/html/index.html

5 Case study

5.1 Fit with synthetic values

The first case studio is a fitting to a theoretical function of "fake" values. Figure 11 shows the theoretical curve (the same as in Figure 1) and its fitting. The model example (Figure 1 and left in Figure 11) is a theoretical curve generated by applying Equation (13) to an array of frequencies (25 points between 1 GHz and 50 GHz) with parameters set to be:

- Exponent p = 3.0
- Flux density st = 7.0 mJy
- Frequency vt = 4.0 GHz

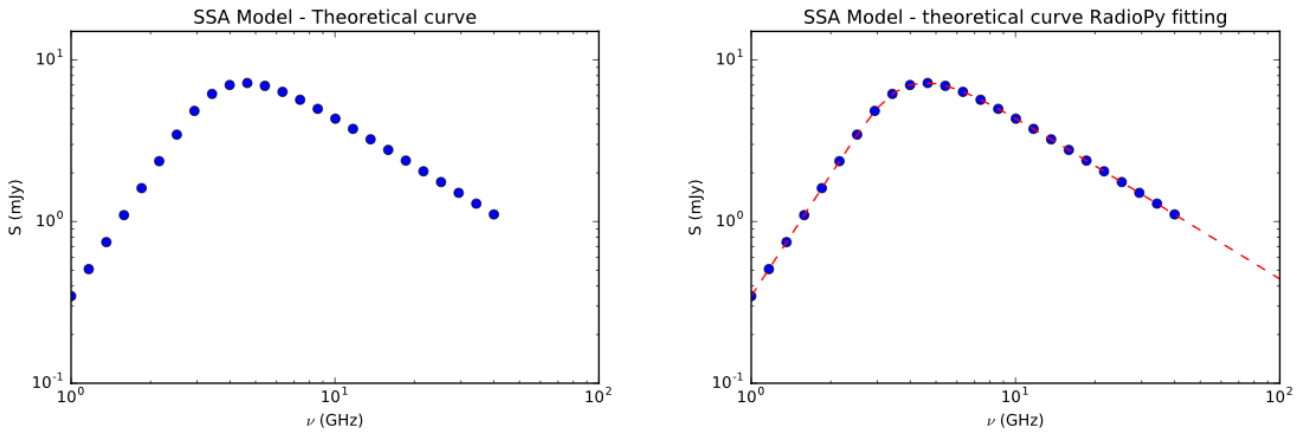

Figure 11. *Both graphics show theoretical flux density (in mJy) vs frequency (in GHz) represented at logarithmic scale. The left one presents points generated by applying Equation (13) to an array of frequencies for parameters: p = 3; st = 7; vt = 4. Right: the red dashed line represents the fitting for the theoretical curve represented at the left part of the Figure. Fit parameters are discussed in the text below.*

Global data:

Time (days): 30
Distance (Mpc): 10 ± 0.5

Fit parameters:

Exponent p: 3.003 ± 0.002
Flux density st (mJy): 7.010 ± 0.004
Frequency vt (GHz): 4.003 ± 0.002

Statistics:

Reduced chi-square per degree of freedom: $1.845 \cdot 10^{31}$
Quadratic error: 0.0004
Error bar correction factor: 1

The data shown above were given by the RadioPy method "Report.infoEpoch()" and the graphs in Figure 11 were generated by RadioPy's method "SNGraph.plotEpoch()". At first glance we can see that the program adjusted the parameters satisfactorily, making a good fit (also seen at Figure 11-right). The χ^2 statistic is enormous but that is okay since the model receives error values (for the flux densities) very close to zero and the calculus of the chi-square statistic weights the uncertainty by dividing by the errors (Equation (17)). The relevant statistic for this demonstration is the quadratic error summation $Q = \sum_N (M_i - T_i)^2$. When this statistics is close to zero (as it is the case) this indicates an excellent proximity of the "measured data" (M_i) and theoretical values (T_i) created by the fit, meaning that we have an almost perfect fit.

This test with "fake" values demonstrates that the fitting algorithms of package RadioPy are precise and they give appropriate results.

5.2 Comparison with Krauss et al. (2012)

Once a theoretical test has been run it is time for the algorithm to demonstrate its capacities with a real case. We took the data for SN 2011dh from Krauss et al. [9] and performed a fitting with RadioPy. We perform our own fitting with RadioPy's method "SNFit.fitEpoch()" and the graphics are shown at Figure 12 whereas the data are tabulated at Table 2.

The first considerations we have to make for comparing Figure 12 and the data from Krauss et al. is that show different error bars. They considered that the uncertainties recorded by their measuring equipment were small compared to the differences with the theoretical curve and thus they multiplied the error bars by a factor between 3 and 7 (depending on the epoch). The purpose of this error correction was to make the reduced χ^2

statistic be close to one, $\chi^2 = 1$. Nevertheless, they did not plot the augmented error bars but the original ones recorded by the equipment. RadioPy has methods to correct the error as Krauss et al. did, seeking for the χ^2 statistic to be one. The uncertainties of the synchrotron spectra at Figure 12 were corrected by a factor between 5 and 10 but those corrections were plotted. That is the origin of the discrepancies on the error bars between the two figures.

Taking that in mind we can see that the fitting is good and the algorithm works as expected. Data and errors are also reasonable and good-looking. Both Figure 12 and the graphics of the author's article look alike and point to similar results. Both fittings were performed fixing the value of parameter "p". Despite the fact that for each epoch the best fit for the measured data have different values for parameter "p", in practice these exponents should not change with time. This behaviour occurs due to external effects that distort the data from the original shape that the theory states (Equation (12)). Krauss et al. [9] found that the "effective" average value for parameter p was p = 2.8. RadioPy has methods that allow researchers to decide whereas they want to perform a fitting fixing the value of parameter p or if they want to treat it as a third free parameter. A first fitting gave different values of p for each epoch but RadioPy also calculates the average of the exponent "p" for the fitting of a sequence of observations, calculating it to be $p_{avg} = 3.0$ for this case. Figure 12 shows the fitting with p fixed to p = 3.0, which we think to be a better fit.

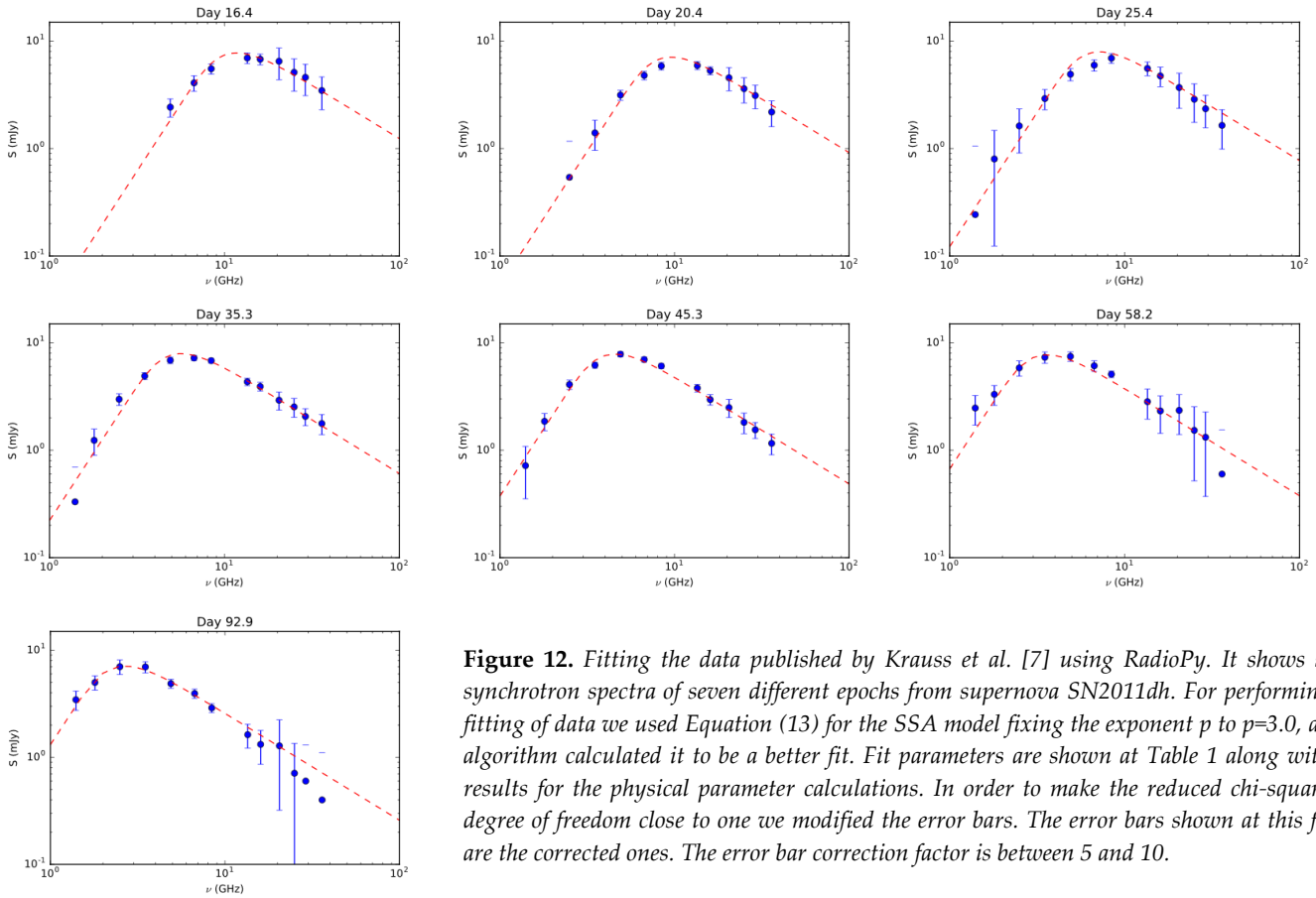

Figure 12. *Fitting the data published by Krauss et al. [7] using RadioPy. It shows seven synchrotron spectra of seven different epochs from supernova SN2011dh. For performing the fitting of data we used Equation (13) for the SSA model fixing the exponent p to p=3.0, as our algorithm calculated it to be a better fit. Fit parameters are shown at Table 1 along with the results for the physical parameter calculations. In order to make the reduced chi-square per degree of freedom close to one we modified the error bars. The error bars shown at this figure are the corrected ones. The error bar correction factor is between 5 and 10.*

Day	Exponent p	Flux S_τ (mJy)	Frequency ν_τ (GHz)	Radious R (1e15 cm)	Magnetic field B (G)	Wind density A (5e-11·g·cm^{-1})
16.400	3.000	7.552 ± 0.468	10.347 ± 0.641	3.310 ± 0.174	1.224 ± 0.075	3.304 ± 0.406
20.400	3.000	6.896 ± 0.418	8.407 ± 0.510	4.005 ± 0.209	0.978 ± 0.059	3.267 ± 0.392
25.400	3.000	7.727 ± 0.671	6.312 ± 0.548	5.732 ± 0.342	0.713 ± 0.061	2.686 ± 0.459
35.300	3.000	7.710 ± 0.350	4.970 ± 0.226	7.286 ± 0.356	0.560 ± 0.026	3.206 ± 0.292
45.300	3.000	7.635 ± 0.306	4.017 ± 0.161	8.865 ± 0.424	0.459 ± 0.019	3.542 ± 0.286
58.200	3.000	7.525 ± 0.472	3.164 ± 0.199	11.311 ± 0.598	0.358 ± 0.022	3.551 ± 0.442
92.900	3.000	6.960 ± 0.273	2.342 ± 0.092	15.212 ± 0.726	0.258 ± 0.010	4.703 ± 0.375

Table 1. *Values of the fit parameters flux density S_{ν_τ} and frequency ν_τ along with the values for the physical magnitudes radius R, magnetic field B and circumstellar density A_* for the seven different epochs measured for supernova SN2011dh and shown at Figure 12. For performing the fitting of data we used Equation (13) for the SSA model, fixing the exponent to p=3.0. Source: RadioPy generated report.*

The required parameters for calculating the physical magnitudes are S_{vop} and ν_{op} -see Equations (14), (15) and (16)-; respectively the flux density at the observed peak and the frequency at which the spectrum's peak occurs. This means that the part which is important for the physical analysis is the region of frequencies between low (optically thick) and high (optically thin), i.e. frequencies near the maximum flux density. Observing the sequence of epochs a shifting of the frequency at the peak (ν_{op}) can be noticed. This indicates an interesting behaviour for the aging of the supernova. RadioPy implements methods for the automatic calculations of this evolution of the physical parameters (Figure 13) that facilitate the researchers the analysis of the interesting physics of supernovae.

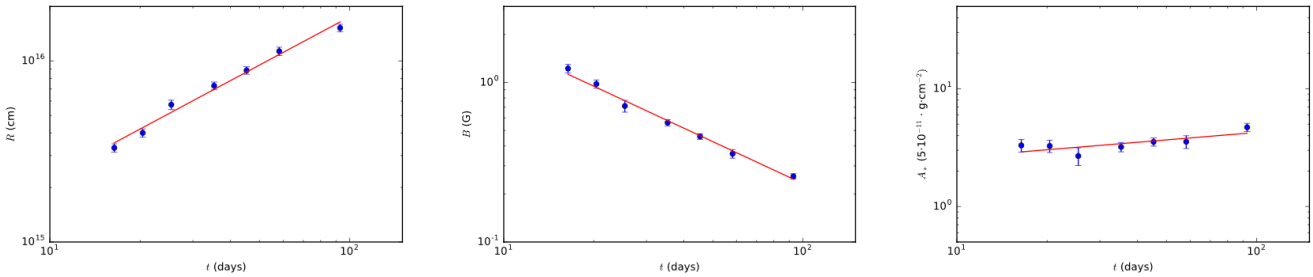

Figure 13. *These three graphics show the evolution of the physical parameters of the SN 2011dh through time as they are calculated for each of the seven measured epochs. Left: the evolution of the shockwave radius (R) fitted to a straight line with a slope $m = 0.887$ –at the logarithmic scale-. Centre: time evolution of the intensity of the magnetic field B; slope of the fitting $m = -0.879$. Right: time evolution of the parametrized circumstellar density A_* fitted to a straight line with slope $m = 0.221$. Graphics generated with RadioPy.*

From Figure 13 we see that the shockwave radius increases with time as $R \propto t^{0.9}$ which is almost a lineal dependency. As the radius increases the energy of the shockwave is dispersed, leading to a similar decreasing of the intensity of the magnetic field B that depends as $B \propto t^{-0.9}$. The circumstellar density has a dependency with $A = t^{0.2}$, i.e. almost no change with time.

Another remarkable characteristic of both the graphics of the author and the ones created by RadioPy (Figure 12) is that the measured data described in Krauss et al. [9] diverge from the pure theoretical model (given by Equation (12)). Such discrepancy is likely caused, as also the authors of Krauss et al. [9] believed, by external effects that distort the shape of the synchrotron spectra. This "distortion" can be easily seen at days 25 and 35 (see Figure 12) in which a curvature appears where there should be straight slopes. This causes the χ^2 to grow

big and thus the error bar correction in these cases must be greater (with subsequent augmentation of the error bars –Figure 12-). In this such distortion could be caused by external absorption of the radiation (process involving thermal emission, free-free radiation for example) that it is not contemplated by the model. These distortions are part of the three main reasons for performing the fitting. $S_{\nu \mathrm{op}}$ and ν_{op} values could be extracted directly from the measured synchrotron spectrum, but instead we calculate a theoretical curve to find the maximum. The three main reasons to perform the fitting are:

- Sparse sampling: generally 15 to 30 measurements are not enough to accurately find a maximum, even with interpolation.
- Noise: usually radio sources are too far and thus data present noise corruption.
- External effects: many effects play their part on the events where such high energies are unleashed. The actual measured curves present distortions from the theoretical curves.

For all those three reasons the methodology on radio astrophysics is to perform a fitting over the data and to take the peak of the theoretical curve as the correct one.

5.2.1 Analysis with RadioPy's main program

When downloading RadioPy package the download folder includes a demonstration with a main program that uses package RadioPy in it. That .py file was created as an easy-to-use application for the analysis a sequence of epochs of the radio evolution of a supernova. That demo uses the data from the SN 2011dh published by Krauss et al. [9]. Here we present a pseudo-code of that main program in order to show how to use RadioPy for performing a complete analysis on a sequence of epochs

```
import radiopy

# In the method 'readCSV', introduce the name of the .csv file that
# the program must read, following the format described at the
# instructions. The program will do the rest.

# Initialize the fitting class to use the desired model
SNFit.init(KraussModel)

# New instance of class ObservationSequence
observSeq = new instance of ObservationSequence()
# Read and record the data from the csv file
filename='data_krauss_2012.csv'
observSeq.readCSV(filename)

# From the observations, calculate a sequence of Results.
resultSeq = SNFit.sequenceFitting(observSeq)

# Save the figures for the fittings at folder 'figures'.
SNGraph.saveSequence(observSeq,resultSeq)

# Calculations of the temporal evolution of the parameters of the
# ResultSequence.
resultSeq = TimeEvol.timeEvol(resultSeq)
# Now save the figures resulting from such calculations.
SNGraph.saveTimeEvol(resultSeq)

# Create an html report in folder 'reports' using svg figures.
Report.createHTMLReport(resultSeq,reportName:str,'svg')

# Create a pdf report in folder 'reports'.
Report.createPDFReport(resultSeq,reportName:str)
```

As a result, the main program creates two folders, "figures" and "report" and then creates and saves all the figures in the corresponding folder. From that folder the method for creating reports takes the figures and fills the folder "report" with both an .html and a .pdf report.

6 Conclusions & future

6.1 Conclusions

In conclusion RadioPy is a Python-implemented open source software tool for radio astrophysics that can automatically calculate physical magnitudes by performing a fitting process of flux vs frequency data measured from the sky. The package provides robust algorithms for fitting the SSA model and is designed with an easily extensible object-oriented architecture that allows RadioPy to adopt new models or modify the current ones making a customizable tool. Its structure also makes it easy to use and sets a standard on how to store and treat data of synchrotron spectrum. RadioPy's capacities were proven with a case studio of the radio evolution of a supernova, in which we made a comparison with the results in Krauss et al. [9].

This tool accomplished all the goals listed at the beginning (section 1.3) and for so we consider the project to be a success.

As an additional conclusion, we found that Pacholczyk's book [5] presents some errata that we discussed and corrected in Appendix B.

RadioPy is publicly available in GitHub:

https://github.com/mapereztorres/RadioPy

6.2 Future

The RadioPy package for radio astrophysics was designed to evolve and adopt new methods and models of radio astrophysics, such as those presented by authors that are mentioned in this document: Kardashev [3] and Murgia et al. [7]. The released version 1 (June 2017) provides methods for fitting the SSA model approximation proposed by Krauss et al. [9]. This model can be used for different radio sources that emit non-thermal radiation (synchrotron spectrum) such as supernovae or radiogalaxies. It is especially adapted for working with sequences of observations for a single event (radio evolution of a supernova) but the tool is expected to evolve and integrate other phenomena and functionalities. Open-source, easy extendible and user-friendly, RadioPy expects that researchers across the planet would join their efforts to build a complete software tool for radio astrophysics.

As it was said at the case study (5.2) some spectrum are distorted. The data of SN2011dh at some epochs cannot be perfectly fitted either at low ($\leq$ 4 GHz) frequencies (suggesting that maybe external absorption by free-free electrons could play a role), or a high ($\geq$ 20 GHz) frequencies, which suggest an evolution of the initial population of relativistic electrons. Therefore, the most obvious next steps in the development of RadioPy, within the scenario of interpreting the radio emission from supernovae, are (a) the inclusion of external free-free absorption and (b) the evolution of the initial population of relativistic electrons.

The inclusion of free-free absorption would then allow to estimate the density of the thermal electrons in the circumstellar region around the supernova. Similarly, taking into account the evolution of the relativistic electrons should naturally lead to the appearance of a break in energy, above which there are no electrons. This in turn leads to the appearance of a break in the slope of the optically thin emission of the supernova.

These and other models and interesting applications and improvements are easy to set with the aid of RadioPy package's structure, described in this document.

Bibliography

[1] Karl G. Jansky. "*Radio waves from outside the solar system*". Nature, 132, 66 (1933). doi:10.1038/132066a0

[2] Vitaly L. Ginzburg, S. I. Syrovatskii. "*Cosmic magnetobremsstrahlung (synchrotron radiation)*". Annual review of astronomy and astrophysics, vol. 3, pp. 297-345 (1965). DOI: 10.1146/annurev.aa.03.090165.001501

[3] Nikolai S. Kardashev. "*Nonstationariness of spectra of young sources of nonthermal radio emission*". Soviet astronomy-AJ, vol. 6, No. 3 (1962).

[4] Vitaly L. Ginzburg, S. I. Syrovatskii. "*Developments in the theory of synchrotron radiation and its reabsorption*". Annual review of astronomy and astrophysics, vol. 7, pp. 375-420 (1969). DOI: 10.1146/annurev.aa.07.090169.002111

[5] Andrzej G. Pacholczyk. "*Radio astrophysics: nonthermal processes in galactic and extragalactic sources*". WH Freeman and Company (1970).

[6] Roger A. Chevalier. "*Synchrotron self-absorption in radio supernovae*". The astrophysical journal, 499: 810-819 (1998).

[7] M. Murgia, C. Fanti, R. Fanti, L. Gregorini, U. Klein, K.H. Mack and M. Vigotti. "*Synhrotron spectra and ages of compact steep spectrum radio sources*". Astronomy and astrophysics 345, 769-777 (1999).

[8] Gabriella di Gennaro. "*A study of extended radio galaxies in the Shapley concentration core* ". Thesis work. Alma mater studiorum, Università di Bologna (2015).

[9] M. I. Krauss, A. M. Soderberg, L. Chomiuk, B. A. Zauderer, A. Brunthaler, M. F. Bietenholz, R. A. Chevalier, C. Fransson and M. Rupen. "*Expanded very large array observations of the radio evolution of SN 2011dh*". The astrophysical journal letters, 750:L40 (2012).

[10] LMFIT package: https://lmfit.github.io/lmfit-py/index.html ; retrieved 6th June, 2017.

[11] Lord Rayleigh. "*XVII. On the resolving-power of telescopes*". The london edinburgh and dublin philosophical magazine and journal of science, 5th ser. v. 10 (1880).

[12] Paul A. Tipler, Gene Mosca. "*Física para la ciencia y la tecnología*". Reverté, 6 edición, vol. 2, pp. 1160 (2010). ISBN: 978-84-291-4428-4.

[13] N. Bartel, M. F. Bietenholz, M. P. Rupen, A. J. Beasley, D. A. Graham, V. I. Altunin, T. Venturi, G. Umana, W. H. Cannon, and J. E. Conway. "*SN 1993J VLBI. II. Related changes of the deceleration, flux density decay, and spectrum*". The astrophysical journal, 581:404–426 (2002). DOI: 10.1086/344198

[14] Donald W. Marquardt. "*An algorithm for least-squares estimation of nonlinear parameters*". Journal of the Society for Industrial and Applied Mathematics, 11, 431–441 (1963). doi:10.1137/0111030

[15] Philip R. Bevington, D. Keith Robinson. "*Data reduction and Error analysis for the physical sciences*". McGraw-Hill, third edition (2003). ISBN: 0-07247227-8.

[16] Autar K. Kaw, Egwu K. Kalu, Duc Nguyen. "*Numerical methods with applications*". University of South Florida, 1st edition (2011). ISBN: 9780578057651.

[17] John R. Taylor. "*An introduction to error analysis: the study of uncertainties in physical measurements*". University Science Books Sausalito CA, 2nd edition (1997). ISBN: 0-935702-42-3.

[18] Richard L. Burden, J. Douglas Faires. "*Numerical Analysis*". Brooks/Cole Cengage Learning, 9th Edition (2011). ISBN-13: 978-0-538-73351-9. ISBN-10: 0-538-73351-9.

[19] Dusty Phillips. "*Python 3 Object-oriented programming*". Packt publishing, second edition (2015). ISBN 978-1-78439-878-1.