CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO
DEPARTAMENTO DE FÍSICA

“Constricciones Cosmológicas de los modelos Materia Oscura con Múltiples Campos Escalares y Energía Oscura con Masa de Compton”

Tesis que presenta

Luis Osvaldo Téllez Tovar

para obtener el Grado de

Doctor en Ciencias

en la Especialidad de

Física

Directores de tesis: Dr. Tonatiuh Matos Chassin
Dr. José Alberto Vázquez González

Ciudad de México Enero, 2023

CENTER FOR RESEARCH AND ADVANCED STUDIES OF THE NATIONAL
POLYTECHNIC INSTITUTE

PHYSICS DEPARTMENT

"Cosmological Constraints on the Multi Scalar Field Dark Matter Model and Compton Mass Dark Energy Model"

by

Luis Osvaldo Téllez Tovar

In order to obtain the

Doctor of Science

degree, speciality in

Physics

Advisors: Ph. D. Tonatiuh Matos Chassin
Ph. D. José Alberto Vázquez González

Mexico City January, 2023

Centro de Investicación y de Estudios Avanzados del Instituto Politécnico Nacional

DEPARTAMENTO DE FISICA

Cosmological constraints on the Multi Scalar Field Dark Matter model and Compton Mass Dark Energy model

This dissertation is submitted for the degree of
Doctor of Philosophy in Physics

A thesis by

Luis Osvaldo Téllez Tovar

Advised by

Dr. Tonatiuh Matos Chassin

Dr. José Alberto Vázquez González

To Evolet,

Khloé and Dante.

Acknowledgements.

To my mother Ernestina, my father Porfirio and my sister Ximena for giving me their love and support. To Evolet, Khloé and Dante for always receiving me with a smile and with love, for allowing me to play and continue learning with you. To my grandmother Nena, because she never stops worrying about me and for her wise advice. To my family for giving me their words of encouragement and for their unconditional support. To my teachers, assistants, friends and classmates, for all the knowledge they shared with me. To Marianita and Malu for helping me with all those administrative questions. To Dr. Tonatiuh Matos and Dr. José Alberto Vázquez for being patient with me, for sharing their knowledge and for all the support they gave me. To CONACYT and Cinvestav for the support and facilities granted to carry out this work.

Difficulties strengthen the mind,

as labor does the body.

Seneca.

Contents

Publications

This thesis is an adaptation of the following articles produced during my studies:

1. L.E. Padilla, **L.O. Tellez**, L.A. Escamilla and J.A. Vazquez, *Cosmological Parameter Inference with Bayesian Statistics*, Universe 7 (2021) 213 [1903.11127]. Used in chapters 2 and 4.

2. **L.O. Téllez-Tovar**, T. Matos and J.A. Vázquez, *Cosmological constraints on the Multi Scalar Field Dark Matter model*, [2112.09337]. *Accepted for publication in PRD.* Used in chapters 3 and 5.

3. T. Matos and **L.O. Téllez-Tovar**, *The cosmic microwave background and mass power spectrum profiles for a novel and efficient model of dark energy*, Revista Mexicana de Física 68 (2022) [2203.07876]. Used in chapter 6.

Resumen

En esta tesis presentamos las constricciones cosmológicas del modelo Multi Scalar Field Dark Matter (MSFDM), en el que suponemos que la materia oscura está formada por distintos campos escalares ultra ligeros. Como primera aproximación consideramos que los campos son reales y que no interactúan entre ellos. Estudiamos las ecuaciones que describen el fondo y las perturbaciones para N-campos. También presentamos la evolución de los parámetros adimensionales de densidad, el espectro de potencias de materia y el espectro de la radiación cósmica de fondo (CMB por sus siglas en inglés). En particular, nos enfocamos en el caso de dos campos escalares con distintas combinaciones de los potenciales $V(\phi) = 1/2m_\phi^2\phi^2$, $V(\phi) = m_\phi^2 f^2 \left[1 + \cos(\phi/f)\right]$ and $V(\phi) = m_\phi^2 f^2 \left[\cosh(\phi/f) - 1\right]$. Sin embargo, el formalismo matemático y los códigos empleados pueden ser extendidos a más campos. Por otro lado, usamos los datos de BAO, Big Bang Nucleosynthesis, bosques de Ly-α y Supernovas para hallar las restricciones de los parámetros para los casos en que tenemos un solo campo y dos campos, también calculamos la evidencia bayesiana. Hallamos que algunas combinaciones de los potenciales son penalizados por esta, mientras que otros muestran un preferencia igual al modelo de materia oscura fría. Por otro lado, estudiamos numéricamente un nuevo modelo para explicar la expansión acelerada del Universo en el que no es necesaria la existencia de la energía oscura. Usando distintas aproximaciones al parámetro de Hubble y considerando un fluido efectivo obtenemos la evolución de los parámetros adimensionales de densidad, el espectro de potencias de materia y el espectro del CMB.

Abstract

In this work we show the cosmological constraints on the Multi Scalar Field Dark Matter model (MSFDM), we assume the dark matter is made up of different ultra-light scalar fields. As a first approximation, we consider they are real and do not interact with each other. We study the equations for both the background and perturbations for N-fields and present the evolution of the density parameters, the mass power spectrum and the CMB spectrum. Here, we focus on two scalar fields with several combinations for the potentials $V(\phi) = 1/2m_\phi^2\phi^2$, $V(\phi) = m_\phi^2 f^2 \left[1 + \cos(\phi/f)\right]$ and $V(\phi) = m_\phi^2 f^2 \left[\cosh(\phi/f) - 1\right]$. However, the mathematical formalism and the codes used can be extended to more fields. We use the data from BAO, Big Bang Nucleosynthesis, Lyman-α forest and Supernovae to find constraints on the sampling parameters for the cases of a single field and double field, along with the Bayesian evidence. We found that some combinations of the potentials get penalized through the evidence, however for others there is a preference as good as for the cold dark matter. On the other hand, we numerically study a new model to explain the accelerated expansion of the Universe in which the existence of dark energy is not necessary. Using different approaches to the Hubble parameter and considering an effective fluid, we obtain the evolution of the dimensionless density parameters, the matter power spectrum and the CMB spectrum.

Outline

The outline of the thesis is as follows. In **Chapter** 1 we start by briefly introducing the observations that led to the need for dark matter and dark energy. In **Chapter** 2, we present the necessary concepts of the theoretical framework to develop this thesis. We provide a brief review of General Relativity, the Friedmann-Lemaître-Robertson-Walker metric (which describes an expanding Universe that is also homogeneous and isotropic), the concepts of redshift, Friedmann and continuity equations. Then we describe the Λ Cold Dark Matter model (ΛCDM), considered as the standard model of Cosmology, its current issues and some possible solutions. In particular we review the Scalar Field Dark Matter. We also describe the cosmological observations that we use throughout this work.

In **Chapter** 3, we describe the motivations for having more than one scalar field as dark matter. We introduce the Multi Scalar Field Dark Matter model (MSFDM) presenting the mathematical formalism for the background and the linear perturbations for n fields that, as a first approximation, are real, spatially homogeneous and non-interacting with each other, each with a potential. Using an amended version of the CLASS code we obtained the evolution of the background, Matter Power Spectrum and Cosmic Microwave Background power spectrum for the particular case of two scalar fields with the quadratic, trigonometric cosine and hyperbolic cosine potentials.

In **Chapter** 4, we describe the process of parameter inference with Bayesian statistics. We review the difference between frequentist and Bayesian statistics, as well as Bayes' theorem and some other concepts such as prior, likelihood and Chi-square to mention a few, and describe the numerical techniques of the MCMC and the convergence criterion.

In **Chapter** 5, we present the constraints of the SFDM and MSFDM parameters using an amended version of the Monte Python code for different combinations of potentials and observations. On the other hand, we make a comparison with the axionCAMB code that considers a single field as dark matter with the quadratic potential. We check extensions to our model by varying the neutrino masses and with non-zero curvature. We also study, in an introductory way, whether our model can be described by the α β γ-parametrisation used for non-CDM models.

In **Chapter** 6, we present the motivations and mathematical formalism of the Compton Mass Dark Energy model (CMaDE) as well as the evolution of the background, Matter Power Spectrum and Cosmic Microwave Background power spectrum using another amended version of the CLASS code. In **Chapter** 7, we present the conclusions and future work.

In the appendices we show how to install, modify, and use the CLASS and Monte Python codes, the Planck 18 likelihood and the Ly-α notebook (from which we got the data used in chapters 3 and 5) used in this work.

Chapter 1

Introduction

Although fundamental problems in physics have been solved in recent years, such as the existence of the Higgs boson, a crucial part of the Standard Model of particles; or gravitational waves and black holes, predicted by Albert Einstein from the Theory of General Relativity; phenomena remain to be explained. In particular, what are dark matter and dark energy.

The idea of dark matter has its origin in the 1930's when Fritz Zwicky concluded that there was very little luminous matter in the Coma cluster to explain the movement of the approximately 1000 galaxies in it, that is, it takes more matter than observed to explain Coma dynamics [1]. Years later, in the 70's, the astronomer Vera Rubin studied the speed of the nearby galaxies she observed [2, 3]. It was believed that the speed with which matter in a spiral galaxy should move had to be a function of its radius from the center of the galaxy[4]

$$v = \sqrt{\frac{GM(R)}{R}}. \tag{1.1}$$

Where R is the radius measured from the center of the galaxy and $M(R)$ is the mass depending on the radius. This means that farther away matter is, the slower its speed will be. However, it was found to remain almost constant for large radius which did not agree with the observed matter. This led to reconsidering the idea that there must be a type of matter that is not visible and that it must only have gravitational interaction in order to explain the observed velocities. Another evidence of the dark matter existence is the relative motion between galaxies. The gravitational force a single galaxy feels produced by those around it is studied. Dark matter is also necessary to explain structure formation in the Universe (galaxies, galaxy clusters, cosmic background radiation). It is believed that this structure started to formed in the early Universe because of small irregularities that grew due to gravitational interaction. However, it is accepted that baryonic matter (short name used to refer to all the know particles of the Standard Model of Particles) is not enough to cause the gravitational force that helped the growth of irregularities in the early Universe.

On the other hand, when the relative motion of galaxies was studied, it was assumed that they all must fulfill the relation

$$v = H_0 r + V, \tag{1.2}$$

where v is the speed of separation, r the distances between galaxies and v the peculiar speed. So if we knew H_0, we could know the evolution of the Universe. This means that we need to observe as many galaxies as possible to obtain information on H_0. However, measuring them has problems because we do not know any law to determine the peculiar speed. On the other hand, we also need to accurately measure the distances between them where the usual technique is parallax but this fails for very distant galaxies. So, other objects that behave more or less the same between them, that there are enough of them, and that they shine a lot; are needed. One of the objects that meets these conditions is a type of star called Cepheids because its luminosity varies periodically practically without irregularities, if we know the brightness or its luminosity (knowing its period) we can know the distance, its light decreases proportionally to the square of the distance to the observer and its brightness is dim at far distances [4]. When Hubble studied the Cepheids [5], he found that there was a proportional relation between the distance of a galaxy and the speed with which it was receding

$$z = \frac{H_0}{c} D, \tag{1.3}$$

where z is the redshift, c the speed of light, D the distance and H_0 the so called Hubble parameter. The meaning of this equation is that galaxies with greater distance are moving faster from us at this instant. This relation only holds if the Universe is expanding, although it was not yet known whether this expansion was accelerated, constant, or decelerated.

However, using Cepheids also has problems because, at large distances, they are not bright enough. So, the next object to be used were Type Ia supernovae stars because they are very bright and the luminosity is similar between them, the luminosity-shape of the light curve relation is empirical and because the luminous distance, which depends on H, is measured. Using these stars, Perlmutter, Schmidt, and Riess [6, 7] found that to explain their observations the Universe must have an accelerating expansion, and it has been proposed that dark energy is causing this expansion. However, almost nothing is known about dark energy. In addition to the observations that we have mentioned, there is currently other evidence [8, 9, 10] of the existence of dark matter and dark energy such as the Cosmic Microwave Background (CMB) or the Baryonic Acoustic Oscillation (BAO).

As we will see in later chapters, there is a standard model that describes these phenomena known as ΛCDM in which dark matter is cold (CDM) and dark energy is a cosmological constant (Λ). However, this one still has problems to solve. In this work we study alternatives models that could provide a solution to such problems.

[11, 12, 13]

Chapter 2

Cosmology

The beginning of the cosmology as it is known today emerged after 1920, when the Shapley-Curtis debate was carried out [14]. This debate was held between the astronomers Harlow Shapley and Heber Curtis, resulting in a revolution for astronomy at that time by reaching an important conclusion: “The Universe had a larger scale than the Milky Way”. Several observations at that epoch established that the size and dynamics of the cosmos could be explained by Einstein’s General Theory of Relativity. At the beginning, cosmology was a speculative science based only on a few datasets, and it was characterized by a dispute between two cosmological models: the steady state model and the Big Bang (BB) theory. It was not until 1990, when the amount of data increased enough to discriminate and rule out compelling theories, that the BB model awarded was the most accepted.

Since Cosmology is based on the General Theory of Relativity, in this chapter we will summarize the basic concepts used in this work.

2.1 Dynamics and geometry

In order to specify the geometry of the Universe, an essential assumption is the Cosmological Principle: for a particular time and on sufficiently large scales, the observable Universe can be considered homogeneous and isotropic, with great precision. For example, at scales

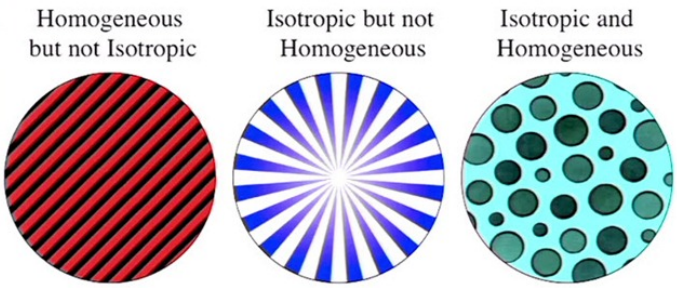

Homogeneous but not Isotropic
Isotropic but not Homogeneous
Isotropic and Homogeneous

Figure 2.1: In the left panel we see an example of homogeneity because the image looks the same no matter where we stand. However, it is not isotropic because the lines follow a particular direction. The central panel shows isotropy because the lines go out in all directions but it is not homogeneous because it has a center, therefore the image does depend on where we stand. In the panel on the right we see an example of isometry and homogeneity because the circles do not follow any direction and there is no center. [15]

greater than 100 Mega-parsecs, the distribution of galaxies observed on the celestial sphere justifies the assumption of isotropy. The uniformity observed in the temperature distribution (one part in 10^5) measured through the Cosmic Microwave Background (CMB) is the best observational evidence we have in favor of a universal isotropy. Therefore, if isotropy is taken for granted and by taking into account that our position in the Universe has no preference (Copernican Principle), the homogeneity follows when considering isotropy in each point. Here isotropy means that the Universe is observed equally in all directions and homogeneity means that the universe is observed equally in each point of space, see Figure 2.1 for an example.

2.1.1 The metric

In General Relativity, the interaction between the geometry of spacetime and the amount of matter contained is studied. In general, spacetime is not necessarily flat but can be curved. The curvature of spacetime produces physical effects on the contained matter, these effects are usually associated with a gravitational field. On the other hand, the curvature of spacetime is related to the contained matter by means of an energy-momentum tensor [16]. The above expressions can be summarized by paraphrasing Wheeler: *"matter tells spacetime how to curve and, in turn, the geometry of this curvature tells matter how to move"*. We can write this sentence down by the Einstein equations

$$G_{\mu\nu} = 8\pi G T_{\mu\nu}, \tag{2.1}$$

where $G_{\mu\nu}$ is the Einstein tensor (geometry of the spacetime), $T_{\mu\nu}$ is the energy-momentum tensor (matter contained in the Universe) and G is the gravitational Newton constant [17, 16, 18, 19].

The distance between two points in a curved spacetime can be measured as

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu, \tag{2.2}$$

where $g_{\mu\nu}$ is the metric tensor that contains all the information about the geometry of the spacetime. From now on, and unless stated otherwise, greek letters μ, ν, ... denote spacetime indices ranging from 0 to 3, while latin letters i, j, ... denote spatial coordinates ranging from 1 to 3.

2.1.2 Symmetric Spaces

The fact that the Universe is homogeneous and isotropic allows us to divide space into three-dimensional slices, each at a fixed time, these being also homogeneous and isotropic. Let's classify these slices. Note that these three dimensional spaces have constant curvature. They can have zero, positive or negative curvature. For the flat case, zero curvature, we have a three-dimensional Euclidean space E^3 whose line element is given by

$$dl^2 = d\vec{x}^2 = \delta_{ij} dx^i dx^j. \tag{2.3}$$

For positive curvature, we have a three-dimensional space represented by a sphere inside a four-dimensional Euclidean space E^4. The line element is

$$dl^2 = d\vec{x}^2 + du^2, \tag{2.4}$$

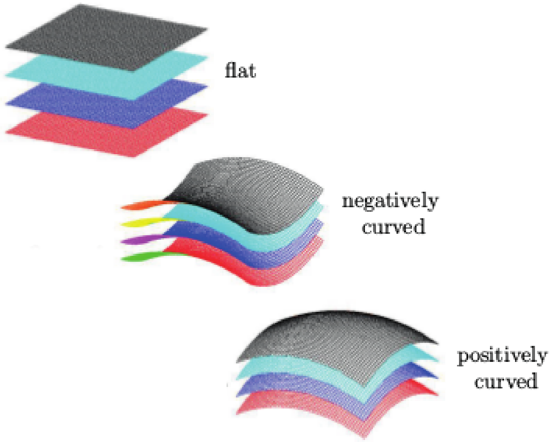


Figure 2.2: The spacetime can be foliated into spatial hypersurfaces with plane, positive and negative curvature [17].

with

$$\vec{x}^2 + u^2 = a^2. \tag{2.5}$$

Where a is the radius of the sphere. The surface of the sphere is also homogeneous and isotropic [17].

In the case of negative curvature, three-dimensional space can be represented as a hyperboloid. The line element is given by

$$dl^2 = d\vec{x}^2 - du^2, \tag{2.6}$$

with

$$\vec{x}^2 - u^2 = -a^2. \tag{2.7}$$

Where a is an arbitrary constant. The hyperboloid is also homogeneous and isotropic [17].

To study the cases of nonzero curvature, we do, for convenience, a rescaling such that we work with dimensionless quantities. In particular, we consider the change

$$\vec{x} \to a\vec{x}, \tag{2.8}$$

$$u \to au. \tag{2.9}$$

Where a has units of length. We can rewrite the line element for both cases as

$$dl^2 = a^2 \left[d\vec{x}^2 \pm du^2\right], \tag{2.10}$$

with

$$\vec{x}^2 \pm u^2 = \pm 1. \tag{2.11}$$

From the above condition we have

$$udu = \mp \vec{x} \cdot d\vec{x}. \tag{2.12}$$

Using the above expression we can rewrite the line element as

$$dl^2 = a^2 \left[d\vec{x}^2 \pm \frac{(\vec{x} \cdot d\vec{x})^2}{1 \mp \vec{x}^2}\right]. \tag{2.13}$$

In order to include the case of zero curvature (flat space) the line element must be rewritten as

$$dl^2 = a^2 \left[d\vec{x}^2 + k\frac{(\vec{x} \cdot d\vec{x})^2}{1 - k\vec{x}^2}\right] \equiv a^2 \gamma_{ij} dx^i dx^j, \tag{2.14}$$

where

$$\gamma_{ij} \equiv \delta_{ij} + \kappa \frac{x_i x_j}{1 - \kappa \left(x_k x^k\right)}. \tag{2.15}$$

While $\kappa = 0$ corresponds to the flat case, $\kappa = +1$ represents the case with positive curvature (sphere) and $\kappa = -1$ corresponds to the negative curvature (hyperboloid). For $dl^2 > 0$ we must take $a^2 > 0$. And the form of γ_{ij} depends on the chosen coordinates [17].

For convenience, we will use spherical polar coordinates. This will transform Equation (2.14) into a diagonal metric. For this, we will use [17]

$$\begin{aligned} d\vec{x}^2 &= dr^2 + r^2 d\Omega^2, & (2.16) \\ \vec{x} \cdot d\vec{x} &= rdr. & (2.17) \end{aligned}$$

Where $d\Omega^2 \equiv d\theta^2 + sin^2\theta d\phi^2$. The line element is given by

$$dl^2 = a^2 \left[\frac{dr^2}{1 - kr^2} + r^2 dr^2\right]. \tag{2.18}$$

We can further simplify the above expression

$$dl^2 = a^2 \left[d\chi^2 + S_k^2\left(\chi\right) d\Omega^2 \right]. \tag{2.19}$$

Where we have redefined the radial coordinate such that

$$d\chi = \frac{dr}{\sqrt{1-kr^2}}. \tag{2.20}$$

The form of S_k depends on the value of k.

$$S_k(\chi) \equiv \begin{cases} \sinh\chi & k=-1, \\ \chi & k=0, \\ \sin\chi & k=+1. \end{cases} \tag{2.21}$$

2.2 FLRW metric

To study the Universe, we use the FLRW or Friedmann-Lemaître-Robertson-Walker metric. This metric describes an expanding Universe that is also homogeneous and isotropic. It is obtained by adding the time coordinate to the metric discussed in the previous section. That is

$$ds^2 = dt^2 - a^2(t)\gamma_{ij}dx^i dx^j. \tag{2.22}$$

Where $a(t)$, called the scale factor, is a time-dependent parameter. Note that (2.22) depends only on the scale factor and the curvature parameter k. The coordinates $x^i \equiv \{x^1, x^2, x^3\}$ are called comoving coordinates and are related to the physical coordinates as $x^i_{phys} = a(t)x^i$. The comoving coordinate system follows the expansion of the Universe in such a way that the spatial coordinates of non-moving objects do not change. The distance between two objects measured in comoving coordinates remains the same. However, the physical or proper distance of objects grows over time as the Universe expands. The physical velocity of an object is given by

$$v^i_{fis} \equiv \frac{dx^i_{fis}}{dt} = a(t)\frac{dx^i}{dt} + \frac{da}{dt}x^i \equiv v^i_{pec} + Hx^i_{fis}, \tag{2.23}$$

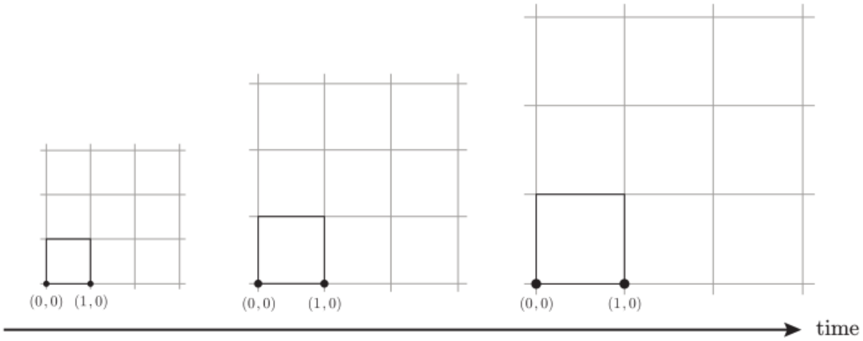


Figure 2.3: The distance between two points measured from the comoving coordinate system remains constant as the Universe expands. The physical distance is proportional to the comoving by the scale factor $a(t)$ [17].

where we define the Hubble parameter as

$$H \equiv \frac{\dot{a}}{a}. \tag{2.24}$$

In (2.23) there are two contributions: the term v^i_{pec} is called the peculiar velocity while Hx^i_{fis} is called the Hubble flux. The peculiar velocity is the velocity of the object measured by a comoving observer, that is, one moving with the Hubble flux [17].

In spherical coordinates the FLRW metric is given by

$$ds^2 = dt^2 - a^2(t)\left[\frac{dr^2}{1-kr^2} + r^2 d\Omega^2\right]. \tag{2.25}$$

The expression (2.25) can be rescaled and still represent the same thing using the following changes of variables

$$a \rightarrow \lambda a, \tag{2.26}$$

$$r \rightarrow \frac{r}{\lambda}, \tag{2.27}$$

$$k \rightarrow \lambda^2 k. \tag{2.28}$$

This freedom allows us to set the value of the scale factor to one at present. That is

$$a(t_0) \equiv 1 = a_0. \tag{2.29}$$

Where t_0 represents the current time. And all parameters that evaluate to t_0 will be denoted with the subscript 0.

Using (2.20) and (2.21) we can write the FLRW metric as

$$ds^2 = dt^2 - a^2(t)\left[d\chi^2 + S_k^2\left(\chi\right)d\Omega^2\right]. \tag{2.30}$$

The metric written as (2.30) is useful when we want to study the propagation of light [17]. For the same purpose, conformal time is defined as

$$d\tau = \frac{dt}{a(t)}. \tag{2.31}$$

Substituting (2.31) into (2.30) we can rewrite the FLRW metric as

$$ds^2 = a^2(t)\left[d\tau^2 - \left(d\chi^2 + S_k^2\left(\chi\right)d\Omega^2\right)\right]. \tag{2.32}$$

Note that (2.32)) has a form similar to the Minkowski metric multiplied by the scale factor whose argument is now the conformal time.

2.3 Redshift

Part of the information we have about the Universe has been obtained through the light that reaches us from the astrophysical bodies that are in it. However, to obtain a good interpretation of this information, it must be taken into account that, throughout its journey, the light undergoes changes. When an object approaches us, the light we receive from it is compacted making it bluer. If the object is moving away then the wave lengthens and is seen redder. It follows that the light emitted at time t_1 with wavelength λ_1 will be observed at t_0 with a wavelength given by

$$\lambda_0 = \frac{a(t_0)}{a(t_1)}\lambda_1. \tag{2.33}$$

Where a is the scale factor at different times. We define the redshift parameter as

$$z \equiv \frac{\lambda_0 - \lambda_1}{\lambda_1}, \tag{2.34}$$

or, using Equation (2.33),

$$1 + z = \frac{a(t_0)}{a(t_1)}. \tag{2.35}$$

If we use the convention $a(t_0) \equiv 1$, we have

$$1 + z = \frac{1}{a(t_1)}. \tag{2.36}$$

2.4 Friedmann and continuity equations

The content of the Universe needs to satisfy homogeneity and isotropy, as well; hence, here it is described by the energy-momentum tensor of a perfect fluid

$$T_{\mu\nu} = (\rho + P)U_\mu U_\nu - P g_{\mu\nu}, \tag{2.37}$$

where ρ is the energy density, P is the fluid pressure, and U_μ is the 4-velocity relative to the observer. If we take the velocity as $U^\mu = (1, 0, 0, 0)$ (comoving observer), the energy-momentum tensor reduces to

$$T^\mu_\nu = g^{\mu\lambda} T_{\lambda\nu} = \begin{pmatrix} \rho & 0 & 0 & 0 \\ 0 & -P & 0 & 0 \\ 0 & 0 & -P & 0 \\ 0 & 0 & 0 & -P \end{pmatrix}. \tag{2.38}$$

Using Equations (2.1) and (2.38), with the FLRW metric, we can obtain the Friedmann equations

$$H^2 \equiv \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3} \sum_i \rho_i - \frac{\kappa}{a^2}, \tag{2.39}$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} \sum_i (\rho_i + 3P_i)\,. \tag{2.40}$$

In these expressions, H accounts for the rate of expansion (or contraction) of the Universe, named as the Hubble parameter. Subindex i labels all the components that we believe the Universe is made of. These equations describe the evolution of the Universe. By combining Equations (2.39) and (2.40), we can obtain the continuity equation given by

$$\dot{\rho} + 3\frac{\dot{a}}{a}\left(\rho + P\right) = 0. \tag{2.41}$$

The meaning of (2.41) is the conservation of the energy-momentum tensor ($\nabla_\mu T^{\mu\nu} = 0$). In order to close the system, we need to include an equation of state that relates pressure and energy density for a given fluid. In particular, we are interested on barotropic fluids, which generally have the form of $P = \omega\rho$.

2.5 Content of the Universe and the ΛCDM Model

Once the equations that define the dynamics of the Universe are known, it is necessary to specify its content. The standard cosmological model, also known as Λ Cold Dark Matter (ΛCDM), is one of the most accepted models to describe the Universe, with its content being:

- **Dust:** It has no pressure ($P = 0$), and its energy density takes the form of $\rho \propto a^{-3}$. Dust is conformed by baryons (ordinary matter).

- **Dark matter:** It is proposed to explain several astrophysical observations, like the dynamics of galaxies in the Coma cluster or the rotation curves of galaxies [4, 20]. The ΛCDM model assumes the dark matter only interacts gravitationally (and possibly by weak interaction) with the rest of the Universe, hence its name, Cold Dark Matter (CDM). Since it is proposed as interacting only via gravitational force, there can be several candidates to fulfill this requirement: it could be conformed by weakly interacting massive particles (WIMPs), by gravitationally-interacting massive particles (GIMPs); by axions (hypothetical elementary particles); or by sterile neutrinos, just to mention a few. For a short review about Dark Matter and possible candidates, see Reference [21].

- **Radiation:** This corresponds to relativistic particles that follow the relation $P = \frac{1}{3}\rho$, which implies a density with a behavior $\rho \propto a^{-4}$. We consider photons ρ_γ and massless neutrinos ρ_ν as radiation, so the total radiation energy density in the Universe is given by

$$\rho_r = \rho_\gamma + \rho_\nu. \tag{2.42}$$

The relation between these quantities is

$$\rho_\nu = N_{\text{eff}} \times \frac{7}{8} \times \left(\frac{4}{11}\right)^{4/3} \rho_\gamma, \tag{2.43}$$

where N_{eff} is the effective number of relativistic degrees of freedom, with standard value $N_{\text{eff}} = 3.046$ [22].

- **Dark Energy:** It is introduced to explain the current accelerated expansion of the Universe. In the ΛCDM model, dark energy is given by the cosmological constant Λ or equivalently by an equation of state $\omega = -1$.

Component	ω
Dust	0
Radiation	1/3
Cosmological Constant	-1

Table 2.1: Equation of state associated to each component of the Universe.

Each of these components can be described by its equation of state shown in Table 2.1, and defining the density parameter

$$\Omega_i \equiv \frac{\rho_i}{\rho_{\text{crit}}}, \quad \text{with} \quad \rho_{\text{crit}} = \frac{3H^2}{8\pi G}, \tag{2.44}$$

with ρ_{crit} being the condition to have a flat Universe or equivalently zero curvature, we can rewrite (2.39) as

$$\frac{H^2}{H_0^2} = \Omega_{r,0}a^{-4} + \Omega_{m,0}a^{-3} + \Omega_{k,0}a^{-2} + \Omega_{\Lambda,0}, \tag{2.45}$$

where $\Omega_{r,0}$ is the radiation density parameter, $\Omega_{m,0} \equiv \Omega_{b,0} + \Omega_{DM,0}$ corresponds to the total matter, $\Omega_{b,0}$ to baryons, $\Omega_{DM,0}$ to dark matter, $\Omega_k \equiv -\kappa/(aH)^2$ the curvature density parameter, and $\Omega_\Lambda \equiv \Lambda/3H^2$ associated with the Cosmological Constant, and the subscript zero indicates they are evaluated by today's values.

2.6 Cosmological parameters

2.6.1 Base parameters

These parameters, also known as standard parameters, are the main quantities used in the description of the Universe. They are not predicted by a fundamental theory, but their values must be fitted to provide the best description of the current astrophysical and cosmological observables.

To explain the homogeneous and isotropic Universe, we can use the density parameter of each component $\Omega_{i,0}$ and the Hubble parameter H_0 related by (2.45).

In particular, the radiation contribution is measured with great precision, so that Ω_γ is pinned down very accurately, and, hence, there is no need to fit this parameter. Similarly, neutrinos, as long as they maintain a relativistic behavior, can be related to the density of the photons through (2.43).

On the other hand, the existence of strong degeneracies from different combinations of parameters is also notorious. In particular, the geometric degeneracy involving Ω_m, Ω_Λ and the curvature parameter $\Omega_k = 1 - \Omega_m - \Omega_\Lambda$. To reduce these degeneracies, it is common to introduce a combination of cosmological parameters such that they have orthogonal effects in the measurements.

2.6.2 Derived parameters

The above standard set of parameters provides an adequate description of the cosmological models. However, this parameterization is not unique, and some others can be as good as this one. Various parameterizations make use of the knowledge of the physics or the sensitivity of the detectors and can, therefore, be interpreted more naturally. In general, other parameters could have been used to describe the Universe, for example: the age of the Universe, the current temperature of the neutrino background, the epoch of equality of matter-radiation, or the epoch of reionization. In the ΛCDM model, in order to decrease degeneracies, the physical energy densities $\Omega_{DM,0}h^2$ and $\Omega_{b,0}h^2$ are used as base parameters [22].

2.7 ΛCDM and beyond

The success of ΛCDM model relies mainly on the accurate agreement with several cosmological observations, for example measurements of the current accelerated expansion of the Universe and the Cosmic Microwave Background Radiation (CMB) [10].

The best description for the dark matter assumes to be made up of pressureless, non-relativistic, neutral and non-baryonic particles whose interaction is primarily through gravity. However, the assumption of a particle with these properties brings up many unexplained features, mainly at galactic scales, i.e. the central density behaviour in galactic halos or the overpopulation of substructures at small scales; for an extended review about the problems and possible solutions see Refs. [20, 23, 24].[1] On the other hand, so far it was not possible to detect WIMPs either directly in the laboratory, or indirectly by astronomical signals of distant objects have failed. This leads us to explore alternatives to the ΛCDM model.

With this in mind, several alternatives have been suggested. For instance, the Scalar

[1] Recent studies suggest there is no longer a Missing Stellites Problem (MSP), however there could be the problem so many satellites, see [25, 26].

Field Dark Matter (SFDM) model proposes the dark matter as a spin 0 boson particle [27, 28, 29, 30, 31, 32]; or the Self Interacting Dark Matter, as its name states, relies on the cold dark matter to be made of self interacting particles [33, 34, 35]. On the other hand, in order to explain the accelerated expansion of the Universe, there exist different modifications to the theory of General Relativity, i.e., $f(R)$ theories [36], braneworld models [37, 38]. There are also some alternatives to the cosmological constant as Dark Energy, i.e., scalar fields (quintessence, K-essence, phantom, quintom, non-minimally coupled scalar fields [39, 40, 41, 42, 43]); or many more alternatives, i.e., anisotropic Universes [44, 45, 46]. Finally, if the dark energy is assumed to be a perfect fluid, one of the most popular time-evolving parameterizations for its equation of state consists of expanding ω in a Taylor series, for example, the Chevallier-Polarski-Linder (CPL) $\omega = \omega_0 + \omega_a (1-a)$, with two free parameters ω_0, ω_a [47, 48]. It may also be expanded into Fourier series [49], or many more approaches, as have been suggested to account for a dynamical dark energy [50, 51, 52].

2.8 Scalar Field Dark Matter

An alternative that may alleviate the problems of ΛCDM, mentioned in section 2.7, is to consider a dark matter but now described by a single scalar field ϕ with an associated potential $V(\phi)$, whose evolution is carried out by the Klein-Gordon equation. The idea of assuming a scalar field as the Dark Matter (DM) of the Universe was introduced about two decades ago, where the simplest possibility is to be real, or complex, minimally coupled to gravity and interacting with ordinary matter only gravitationally [53, 54, 55, 56]. Throughout the years, this model has been rediscovered and received many names, for example: Scalar Field DM (SFDM) [55], fuzzy DM [57], Bose-Einstein condensate DM [58] and, more recently, ultra-light axion DM [59], but a systematic study was carried out since then [55]; here we will refer to it as SFDM, as it was named in [60]. Based on this idea, the particle associated to the field is an ultralight boson whose mass oscillates around

$$m_\phi \sim 10^{-22} eV, \tag{2.46}$$

and hence is able to form Bose-Einstein condensates that conform the galactic structures [59, 61, 62, 63, 64]. In this work we will consider that the scalar field is already formed and we will not delve into their origin; see Refs. [59, 35] for details.

For an expanding universe the scalar field cools down along with the expansion, and after a while, this causes that the field, the boson gas, freezes and then condensates. For an ideal boson gas the condensation temperature goes like $T_c \sim m^{-5/3}$, implying that for a mass big enough the condensation temperature becomes small, but the opposite happens if the mass turns out to be light, or ultra-light, the condensation temperature could be very high.

Assuming a Friedmann-Lemaitre-Robertson-Walker metric, a spatially homogeneous scalar field and that all other components are modeled as perfect fluids, we describe the background dynamics with the following equations

$$
\begin{aligned}
H^2 &= \frac{\kappa^2}{2}\left(\sum_I \rho_I + \rho_\phi\right) , && \text{(2.47a)}\\
\dot{\rho_I} &= -3\frac{\dot{a}}{a}\left(\rho_I + p_I\right) , && \text{(2.47b)}\\
\ddot{\phi} &= -3H\dot{\phi} - \partial_\phi V(\phi) . && \text{(2.47c)}
\end{aligned}
$$

In these equations, dots represent derivatives with respect to the cosmic time, H is the Hubble parameter, ρ_I and p_I are the energy and the pressure of the I-th fluid component, and $\kappa^2 = 8\pi G$. Equation (2.47b) is the continuity equation. The scalar field density and pressure are given by

$$
\rho_\phi = (1/2)\dot{\phi}^2 + V(\phi), \qquad p_\phi = (1/2)\dot{\phi}^2 - V(\phi). \tag{2.48}
$$

In order to solve the system of equations, we can use the following change of variables

$$
\begin{aligned}
x &= \frac{\kappa}{\sqrt{6}}\phi' , && \text{(2.49a)}\\
y &= \frac{\kappa}{\sqrt{6}}\frac{\sqrt{2V(\phi)}}{H} , && \text{(2.49b)}\\
z_I &= \frac{\kappa}{\sqrt{6}}\frac{\sqrt{2\rho_I}}{H} . && \text{(2.49c)}
\end{aligned}
$$

Where x and y represents the kinetic and potential parts of the scalar field, respectively; z_I is the new variable for the I-th component and $\dot{q} = Hq'$ with $q' = \frac{dq}{dlna}$. These variables are constrained by the following equation

$$x^2 + y^2 + \sum_I z_I^2 = 1. \tag{2.50}$$

As an example, suppose that the scalar field has the potential $V(\phi) = (1/2)m_\phi^2\phi^2$ and that in the Universe there is radiation (z_γ), baryons (z_b) and cosmological constant (z_Λ). The equations to solve are

$$\begin{aligned}
x' &= -3x + \frac{3}{2}\Pi x - sy\,, && (2.51a)\\
y' &= \frac{3}{2}\Pi y + sx\,, && (2.51b)\\
z'_\gamma &= \frac{3}{2}\left(\Pi - \frac{4}{3}\right) z_\gamma\,, && (2.51c)\\
z'_b &= \frac{3}{2}\left(\Pi - 1\right) z_b\,, && (2.51d)\\
z'_\Lambda &= \frac{3}{2}\Pi z_\Lambda\,, && (2.51e)\\
s' &= \frac{3}{2}\Pi s\,. && (2.51f)
\end{aligned}$$

Where $\Pi = 2x^2 + 4/3z_\gamma^2 + z_b^2$ and $s \equiv \frac{m}{H}$ is used to close the system, see [62] for details. On the other hand, $\Omega_\phi = x^2 + y^2$ and $\Omega_I = z_I^2$. In the left panel of Figure 2.4 we show the solution of these equations, this is, the evolution of the density parameters and we found that, at background level, SFDM can reproduce ΛCDM. However, this change of variable has a disadvantage because it is computationally more expensive to obtain the evolution of the scalar field due to its oscillatory nature, as can be seen in the right panel of the same Figure where we have lowered the resolution of the code to be able to see the oscillations.

To solve this problem, in the reference [65], the authors propose a change of variables in such a way that we can avoid these oscillations:

$$\frac{\kappa\dot{\phi}}{\sqrt{6}H} \equiv \Omega_\phi^{1/2}\sin(\theta/2)\,, \quad \frac{\kappa V^{1/2}}{\sqrt{3}H} \equiv \Omega_\phi^{1/2}\cos(\theta/2)\,. \tag{2.52a}$$

We will study more of this change of variables in the next chapter.

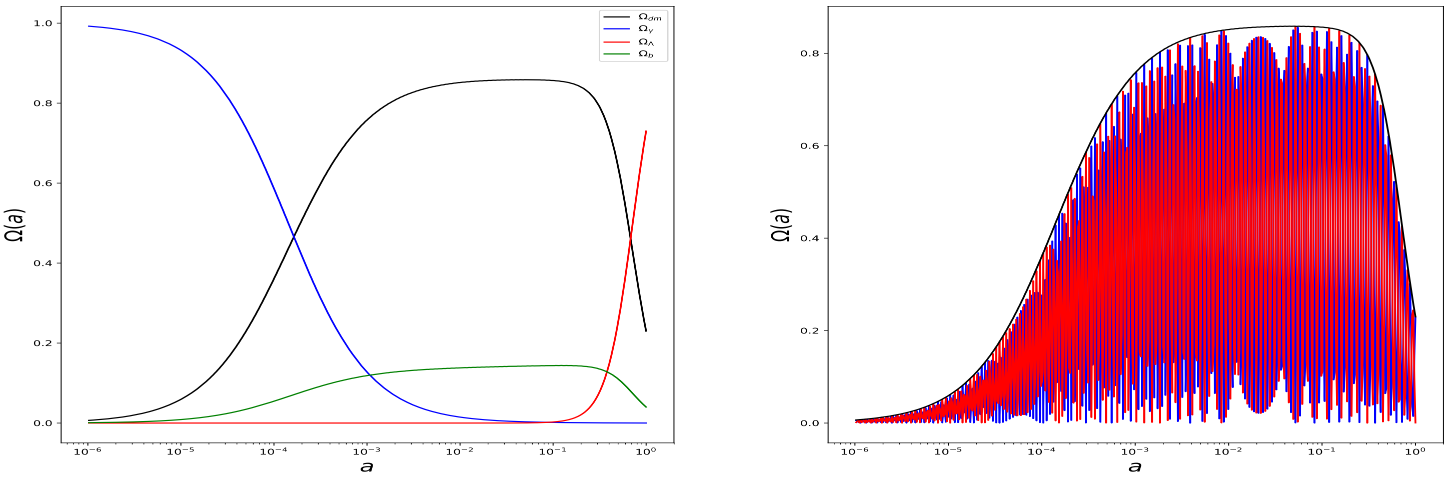


Figure 2.4: Evolution of the density parameters with Scalar Field Dark Matter (left panel) and the evolution of the kinetic (red lines) and potential (blue lines) part of the scalar field while the black solid line is the sum of both part.

2.9 Cosmological observations

In this section, we review some observables, which give information about dark matter and dark energy, and which we will use later to constrain the parameters of the models presented here. [11].

2.9.1 Baryon Acoustic Oscillations (BAO)

The BAO is a statistical property, a feature in the correlation function of galaxies or in the power spectrum. The best description of the early Universe considers that it was made of plasma of coupled photons and matter (baryons and dark matter). The interaction between the gravitational force due to matter and the radiation pressure formed spherical waves in the plasma. When the Universe cooled down enough, the protons and electrons were able to join together, forming hydrogen atoms; therefore, this process allowed photons to decouple from the rest of the baryons. The photons began to travel uninterrupted, while the gravitational field attracted matter towards the center of the spherical wave. The final configuration is an overdensity of matter in the center and a shell of baryons of fixed radius called sound horizon. This radius, used as a standard ruler, is the maximum distance that sound waves could have traveled through the primordial plasma before recombination. The sound horizon r_d is given by

$$r_d = \int_{z_d}^{\infty} \frac{c_s(z)}{H(z)} dz, \tag{2.53}$$

where the sound speed (in terms of redshift z) in the photon-baryon fluid is $c_s(z) = 3^{-1/2} c \left[1 + \frac{3}{4}\rho_b(z)/\rho_\gamma(z)\right]^{-1/2}$, and z_d is the redshift when photons and baryons decouple.

The BAO scale is determined by adopting a fiducial model to be able to translate the angular and redshift separations at comoving distances. The information of the measurement is found in the ratio (α) of the measured BAO scale and that predicted by the fiducial model (fid). In an anisotropic fit, two ratios are used, one perpendicular $\alpha_\perp$ and one parallel $\alpha_\parallel$ to the line of sight. A measurement of $\alpha_\perp$ constrains the ratio of the comoving angular diameter distance to the sound horizon [66]:

$$\frac{D_M(z)}{r_d} = \alpha_\perp \frac{D_{M,fid}(z)}{r_{d,fid}}, \tag{2.54}$$

where the comoving angular diameter distance is given by

$$D_M(z) = \frac{c}{H_0} S_k\left(\frac{D_c(z)}{c/H_0}\right). \tag{2.55}$$

The line-of-sight comoving distance is defined as

$$D_c(z) = \frac{c}{H_0} \int_0^z dz' \frac{H_0}{H(z')}, \tag{2.56}$$

and $S_k(z)$

$$S_k(x) = \begin{cases} \sinh\left(\sqrt{\Omega_k} x\right)/\sqrt{\Omega_k} & \Omega_k > 0, \\ x & \Omega_k = 0, \\ \sin\left(\sqrt{-\Omega_k} x\right)/\sqrt{-\Omega_k} & \Omega_k < 0. \end{cases} \tag{2.57}$$

The Hubble parameter can be constrained by measuring $\alpha_\parallel$ using an analogous quantity

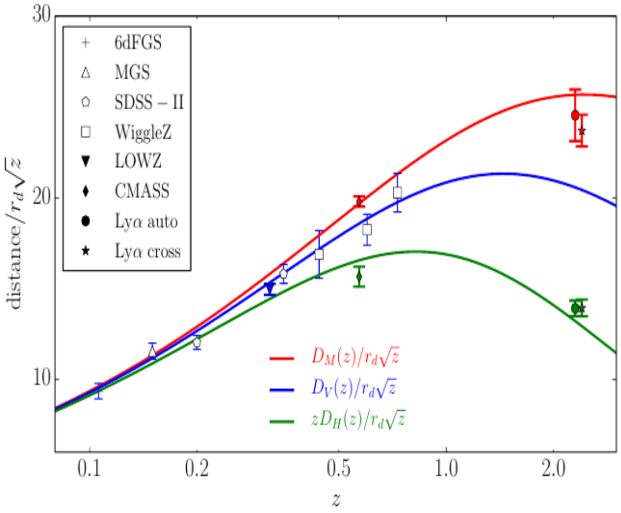


Figure 2.5: BAO Hubble diagram. BAO measurements of D_V/r_d, D_M/r_d and zD_H/r_d from the sources indicated in the legend. The scaling factor $\sqrt{z}$ is included for a better display of the error bars. Solid lines are plotted by using the best-fit values obtained by the Planck satellite [67]. The Lyα cross-correlation points have been shifted in redshift; auto-correlation points are plotted at the correct effective redshift.

$$\frac{D_H(z)}{r_d} = \alpha_{\|}\frac{D_{H,fid}(z)}{r_{d,fid}}, \tag{2.58}$$

with $D_H(z) = c/H(z)$.

If redshift-space distortions are weak (which is valid for luminous galaxy surveys but not for the Ly-α), an isotropic analysis measures an effective combination of (2.54) and (2.58), and the volume averaged distance $D_V(z)$ [66]

$$\frac{D_V(z)}{r_d} = \alpha\frac{D_{V,fid}(z)}{r_{d,fid}}, \tag{2.59}$$

with $D_V(z) = [zD_H(z)D_M^2(z)]^{1/3}$.

The BAO measurements constrain the cosmological parameters through the radius of the sound horizon r_d, Hubble distance $D_H(z)$ and the comoving angular diameter distance $D_M(z)$; see Figure 2.5.

2.9.2 Supernovae type Ia (SNIa)

This type of supernova occurs in binary star systems, one of which is a white dwarf that accretes matter from the star that accompanies it. When the white dwarf accumulates sufficient mass (≈1.4 solar masses), its core will start the ignition temperature for carbon fusion, and, within a few seconds, it releases enough energy to produce the supernova [68]. Since type Ia supernovae (SNIa) are hypothesized to occur near the same mass limit of 1.4 solar masses, commonly referred to as Chandrasekhar mass, their luminosity peaks are fairly consistent and can be standardized and, thus, be used as standard candles [22]. From several analyses of SNIa, the Supernova Cosmology Project and High-z Supernova Search Team both found evidence that the Universe is currently expanding at an accelerated rate [69, 70, 71].

These stars allow us to measure relative distances using the luminosity distance given by

$$D_L \equiv \sqrt{\frac{L}{4\pi S}}, \tag{2.60}$$

where L is the luminosity defined as the energy emitted per unit solid angle per second, and S is the radiation flux density defined as the energy received per unit area per second [4]. The observable quantity is the radiation flux density received, and it cannot be translated into the luminosity density unless the absolute luminosity of the object is known. Even if the luminosity is unknown, it will appear as a scaling factor [4]. The relation between D_L and the cosmological parameters is given by

$$D_L = D_M(1+z), \tag{2.61}$$

where D_M is provided by Equation (2.55). Another important quantity in the observation of supernovae is the standardized distance modulus

$$\mu = m_B^* - M_B + \alpha X_1 - \beta C, \tag{2.62}$$

where m_B^* is the observed peak magnitude in the rest frame of blue band (B), α, β, and M_B are parameters that depend on host galaxy properties [72]. X_1 is the time stretching of the light curve, and C is the supernova color at maximum brightness. The relation between the standardized distance modulus and the luminosity distance is

$$\mu = 5log_{10}\left(\frac{D_L}{10pc}\right). \tag{2.63}$$

Although SNIa have been widely used to restrict cosmological models, there are still discussions about the way it is done. That is, in order to extract information from them, an a priori cosmological model has to be assumed, which may biased the outcomes [73].

2.9.3 Cosmic Microwave Background (CMB):

Corresponds to the radiation that permeates all the Universe, discovered in 1965. Before recombination, baryons and photons were tightly coupled, and once photons decouple from the rest of the matter, they traveled uninterrupted until reach us. The temperature radiation measured at different parts of the sky contains information of the last scattering epoch, gravitational lensing, among others. Here, the CMB displays the primordial anisotropies studied in the angular power spectrum. One of the most important recent collaborations that studies the CMB corresponds to the Planck satellite, and previous probes include COBE [74] and WMAP [75]. It is a European Space Agency mission, in which the main objective is to measure the temperature, polarization, and anisotropies of the CMB over the entire sky. These results would allow to determine the properties of the Universe at large scales and the nature of dark matter and dark energy, as well as to test inflationary theories, determine whether the Universe is homogeneous or not, and obtain maps of galaxies in the microwave [10, 76, 77, 78, 79, 80, 81, 82].

2.9.4 Lyman-α forest (Ly-α)

Quasars (quasi-stellar objects) are highly active nuclei of very luminous galaxies that emit ultraviolet light. When such light travels through space it passes through clouds of neutral hydrogen, which is one of the most common materials in the Universe. As they pass, the photons excite the electrons in the hydrogen atoms causing an absorption or emission in the alpha line of the Lyman series that is reflected in the spectrum that is made here on Earth once the light has arrived. We will see a dip in the corresponding wavelength. When light passes through a large amount of hydrogen we have a dense array of absorption lines, for this reason it is called Lyman alpha forest (Ly-α) [83, 84].

Because most of the hydrogen clouds are less massive than dwarf galaxies, studying the Ly-α forests gives us information on density fluctuations on small scales through the MPS. In inflationary theories, the MPS is related to the Primordial Power Spectrum, so we can also obtain information about the initial conditions of density inhomogeneities (necessary for structure formation) at early ages of the Universe. The Ly-α forests also give us information about the expansion of the Universe because there are redshifts in the light spectrum caused by this expansion or mesauring Hubble parameter [85]. On the other hand, we can measure the position of the BAO peak and obtain constraints on dark energy and dark matter models [86].

In this work we indirectly used the information from the Ly-α forest because we used the MPS points obtained from reference [86] to constrain the parameters of our model. It is important to mention that the constraints obtained should be seen as approximations because the process to obtain the MPS of the Ly-α data depends on a fiducial model, which in the cited reference corresponds to ΛCDM. The proper way would be to do hydrodynamic simulations with our model as can be seen in [87, 88]. However, this will be left as future work.

Chapter 3

Multi Scalar Field Dark Matter

3.1 Multi Scalar Field Dark Matter

One motivation for studying multiple scalar fields as dark matter is that the galaxies start forming and the re-collapse raises the temperature of the bosons again. Therefore, depending on the initial conditions of the galaxy formation, the boson particles can produce excited states, although most of the boson particles remain in the condensate state, or in the ground state [62, 61, 89, 63, 90]. These excited particles can be interpreted as other scalar fields. Thus, once the galaxy is already formed, if it still contains boson particles in several quantum states, then it can be interpreted as a galaxy with different scalar fields. On the other hand, for heavy particles these vibrations could be neglected, however for ultra-light particles an excitation could be comparable with its original mass. In an effective way the scalar field contains the mass plus the effective mass of the excitation's energy. Thus, this boson gas of particles in excited states could be seen effectively as many scalar fields with different potentials, hence the introduction of the Multi Scalar Field Dark Matter model.

Another motivation to introduce several scalar fields with different potentials could be encouraged because if the 4% of just the baryonic matter in the universe is so diverse, then we can suspect that the 26% of the matter, the dark matter, could be made of several species with different properties too. This diversity of particles could be observed and be tested at various scales, for instance at the galactic level by observing the rotation curves as well as at the largest scales of the Universe. If the dark matter is formed of scalar fields, then both results should match flawlessly. Here we will focus on the cosmological implications.

On the other hand, it has been shown that a scalar field with a convex potential behaves, in average, like dust during late times and hence mimics the behaviour of the cold dark matter. However, depending on the specific form of the potential and even whether the field is real or complex, it may have different behaviors before acting like a pressure-less fluid. So, in order to have a dark matter evolution, it is necessary that the dependence of the potential with respect to the field is such that it presents a minimum value at some critical point around which the field oscillates [64, 59, 91].

Some examples of such kind of potentials are the parabolic function $V(\phi) = 1/2m_\phi^2\phi^2$ [62, 63, 65] and the self-interacting potential with a quartic term contribution $V(\phi) = 1/2m_\phi^2\phi^2 + \lambda_\phi\phi^4$ [35, 92, 93, 94], or the axion like potential $V(\phi) = m_\phi^2 f^2\left[1+\cos(\phi/f)\right]$ [95, 96, 97] and its analog $V(\phi) = m_\phi^2 f^2\left[\cosh(\phi/f)-1\right]$ [98, 99]. Here m_ϕ is interpreted as the mass of the field, λ_ϕ is the self-interacting constant and f represents a decaying constant.

For a single field, several constrictions on its mass have been imposed by using CMB and matter power spectrum [100], galactic dynamics [101], dwarf galaxies [102, 103, 104], N-body simulations with reionization process [105] and Lyman-α flux spectra [106, 87]. However, the presence of small inconsistencies among datasets are also found, and can be seen in Figure 1 of reference [35], and in reference [107].

This single field model provides a very good description of the evolution of the cosmological densities and the peaks of the CMB as well as the number of substructures in galaxy arrays, among others [61, 62, 63, 64, 59]. Nonetheless it still presents some open issues [59]. For example, numerical simulations have shown that the mass of the field could vary for different scales of the simulation in order to fit the observations, for instance on the formation of galaxies [108], the whirling plane of satellite galaxies around the Milky Way, Andromeda and Centaurus A galaxies [109], the same mass scale in satellite galaxies of the Milky Way [110], or the σ_8 and H_0 tension [111, 112], just to mention a few.

In order to alleviate these discrepancies, we open up the possibility that the dark matter may be composed of several types of scalar fields. We present a model where the dark matter may be made up by several scalar fields, with different potentials, and to show its constraints imposed by current cosmological observations. This model may help us to alleviate the inconsistencies among the constrictions of the mass values obtained by different observations, arguing that they could be different dark matter particles [113]. Also, if we consider two scalar fields with different masses, the same mass scale in the satellite galaxies of the Milky Way could be explained, i.e. one type of particle could form the host galaxy and the other the satellites [114]. We will refer to this model as the Multi Scalar Field Dark Matter (MSFDM). Other areas have included similar ideas where two or more fields are used, for instance a combination of the inflaton and the SFDM [35], two scalar fields as dark energy [115, 39], the inflaton and the curvaton [116], two scalar fields for inflation [117, 118], interactions between dark energy and dark matter [119] or the axiverse model [113, 120, 121] (see also [122]).

Given the motivations above, in the following sections we study the background dynamics and the linear perturbations of the model. As a first approximation we consider the scalar fields are spatially homogeneous, real and with no interaction among each other.

3.2 Background dynamics

We base our analysis on a flat Universe filled up with the standard components: baryons, dark energy in the form of a cosmological constant (Λ), photons and neutrinos as relativistic species and dark matter (DM). For the neutrinos, we consider the base model used in [10], in which they assumed a normal mass hierarchy [123, 124, 125, 126], two massless neutrinos and a massive one with the minimal mass $\sum m_\nu = 0.06$ eV. In particular, we assume the DM is described by multiple real scalar fields ϕ_i endowed with their corresponding potentials $V_i(\phi_i)$, whereas the rest of the matter components are modeled as perfect fluids. Assuming a Friedmann-Lemaitre-Robertson-Walker metric, the equations of motion for the background dynamics are

$$\begin{aligned} H^2 &= \frac{\kappa^2}{2}\left(\sum_I \rho_I + \sum_i \rho_{\phi i}\right) , && (3.1a) \\ \dot{\rho}_I &= -3\frac{\dot{a}}{a}\left(\rho_I + p_I\right) , && (3.1b) \\ \ddot{\phi}_i &= -3H\dot{\phi}_i - \partial_{\phi i} V_i(\phi_i) \, . && (3.1c) \end{aligned}$$

Here, dots represent derivatives with respect to the cosmic time t, H is the Hubble parameter, $\kappa^2 = 8\pi G$, and ρ_I and p_I are the energy density and pressure of the I-th fluid species respectively, whereas for the scalar fields we have the associated density and pressure given by the standard expressions

$$\rho_{\phi i} = (1/2)\dot{\phi}_i^2 + V_i(\phi_i), \qquad p_{\phi i} = (1/2)\dot{\phi}_i^2 - V_i(\phi_i). \tag{3.2}$$

Notice that we are assuming different species of scalar fields, represented each one by the subindex i in the above equations. The Klein-Gordon Equations (3.1c), for each of the fields, can be written in a more manageable form by using the following polar transformation [127][1]

$$\frac{\kappa\dot{\phi}_i}{\sqrt{6}H} \equiv \Omega_{\phi i}^{1/2}\sin(\theta_i/2) \, , \quad \frac{\kappa V_i^{1/2}}{\sqrt{3}H} \equiv \Omega_{\phi i}^{1/2}\cos(\theta_i/2) \, , \tag{3.3a}$$

where $\Omega_{\phi i} \equiv \kappa^2\rho_{\phi i}/3H^2$ represents the dimensionless density parameter, and similarly θ_i is an angular degree of freedom directly related to the equation of state (EoS) for each one of the fields, $w_{\phi i} \equiv p_{\phi i}/\rho_{\phi i} = -\cos\theta_i$. Additionally, in order to close the system, we define the potential variables y_{ji} as

$$y_{ji} \equiv -\frac{2\kappa 6^{j/2}}{\sqrt{3}\kappa^j}\frac{\partial^j_{\phi_i} V_i^{1/2}}{H}. \tag{3.3b}$$

whose form depend on the potential for a particular field. For the potentials that we will study here, it is enough to take $j = 1, 2$. However, j can take larger values. These are y_{1i}

[1] The equivalence between the polar transformation and the fluid equations can be seen in [128, 129]

and y_{2i} given by

$$y_{1i} \equiv -2\sqrt{2}\frac{\partial_{\phi i} V_i^{1/2}}{H}\,, \quad y_{2i} \equiv -4\sqrt{3}\frac{\partial^2_{\phi i} V_i^{1/2}}{\kappa H}\,. \tag{3.3c}$$

As a proof of the concept, we focus our study on the following potentials

$$V_i(\phi_i) = \begin{cases} m^2_{\phi i} f_i^2 \left[1 + \cos(\phi_i/f_i)\right] & \text{cos} \\ (1/2) m^2_{\phi i}\phi_i^2 & \text{quadratic} \\ m^2_{\phi i} f_i^2 \left[\cosh(\phi_i/f_i) - 1\right] & \text{cosh} \end{cases} . \tag{3.4}$$

and their possible combinations; with f_i being a characteristic energy scale for the scalar fields ϕ_i, and $m_{\phi i}$ its corresponding mass scale. It can be seen that the variables (3.3c) for the aforementioned potentials can be written as

$$\begin{aligned} y^2_{1i} &= 4\frac{m^2_{\phi i}}{H^2} - 2\lambda_{\phi i}\Omega_{\phi i}\,, && (3.5a) \\ y_{2i} &= \lambda_{\phi i} y_{1i}\,. && (3.5b) \end{aligned}$$

Notice that the three functional forms in (3.4) can be compressed into a dimensionless parameter, $\lambda_{\phi i} = 3/\kappa^2 f_i^2$, which facilitates the numerical calculations. Positive values of $\lambda_{\phi i} > 0$ describe the cosine potential and negative ones $\lambda_{\phi i} < 0$ the cosh potential, whereas the quadratic case corresponds to $\lambda_{\phi i} = 0$ (for more details see [65, 95, 99]). Then, for each field, the associated Klein-Gordon Equation (3.1c) is represented by the following set of coupled equations [2]

$$\begin{aligned} \theta_i' &= -3\sin\theta_i + y_{1i}\,, && (3.6a) \\ \Omega'_{\phi i} &= 3\left(w_{tot} + \cos\theta_i\right)\Omega_{\phi i}\,, && (3.6b) \\ y'_{1i} &= \frac{3}{2}\left(1 + w_{tot}\right) y_{1i} + \frac{1}{2}\lambda_{\phi i}\Omega^{1/2}_{\phi i}\sin\theta_i\,, && (3.6c) \end{aligned}$$

[2] For a generic potential the differential equations of (3.3b) are given by

$$y'_{ji} = -\frac{\dot{H}}{H^2} y_{i,j} + \Omega^{1/2}_{\phi_i}\sin\left(\frac{\theta_i}{2}\right) y_{i,j+1}.$$

For the quadratic, cos and cosh potentials we only need to calculate up to y_{2i} to close the system of equations.

with $w_{tot} = \sum_I \Omega_I w_I + \sum_i \Omega_i w_i$, where $\Omega_I \equiv \kappa^2 \rho_I / 3H^2$ and $w_I = p_I/\rho_I$. The prime denotes derivative with respect to the number of e-folds $N = \ln a$, and for any given variable q we have the relationship $\dot{q} = Hq'$.

The initial conditions necessary to solve these equations are

$$\frac{m_{\phi i}}{H_0} = \frac{5}{2}\Omega_{r0}^{1/2} a_{ini}^{-2} \theta_{i,ini}, \tag{3.7a}$$

for the quadratic case [65]; for the cosine potential we have [95]

$$\begin{aligned} \Omega_{\phi i,ini} &= a_{osc}^{-3} a_{ini}, & (3.7b)\\ y_{1i,ini} &= 5\theta_{i,ini}, & (3.7c)\\ \frac{m_{\phi i}^2}{H_{ini}^2} &= \frac{y_{1i,ini}^2}{4} + \lambda_{\phi i}. & (3.7d) \end{aligned}$$

and for the hyperbolic cosine [99]

$$\begin{aligned} \Omega_{\phi i,ini} &= -\frac{12}{\lambda_{\phi i}}, & (3.7e)\\ \cos(\theta_{\phi i,ini}) &= -\frac{1}{3}. & (3.7f) \end{aligned}$$

The main purpose in all cases is to match a given value of the density parameter $\Omega_{\phi i,0}$ at the present time with the initial values of the dynamical quantities $(\theta_i, y_{1i}, \Omega_{\phi i})_{\rm ini}$ at early enough times (typically for a scale factor of the order of $a_{\rm ini} \simeq 10^{-14}$).

In general terms, the mass parameter $m_{\phi i}$ determines the start of the rapid oscillations of the field ϕ_i around the minimum of the potential V_i, which happens at around $H \simeq 3m_{\phi i}$. For $\lambda_{\phi i} \neq 0$, Eq. (3.5a) becomes a constraint equation that must be satisfied by the field variables at all times, whereas for $\lambda_{\phi i} = 0$ it simply tells us that $y_{1i} = 2m_{\phi i}/H$. In fact, one expects that at late times $m_{\phi i} \gg H$ so that for the three cases in (3.4) the relation $y_{1i} = 2m_{\phi i}/H$ should be satisfied with high accuracy. Only for the case $\lambda_{\phi i} < 0$ (cosh potential) it is also necessary to impose the scaling solution during radiation domination: $\Omega_{\phi i} = -12/\lambda_{\phi i}$ and $\cos\theta_i = -1/3$, and then the initial value of y_{1i} is calculated from Eq. (3.5a). Furthermore, the field mass $m_{\phi i}$ is not an independent parameter in this case, and it has been shown that the two parameters are related through [99]

$$\frac{m_{\phi i}}{H_{\rm ini}} = 1.5 \left[\left(\frac{\lambda_{\phi i}}{3} - 4 \right) \frac{\Omega_{\phi i,0}}{\Omega_{r,0}} a_{\rm ini} \right]^2 , \tag{3.8}$$

with $\Omega_{r,0}$ the present density parameter of relativistic species and $a_{\rm ini}$ the initial value of the scale factor.

3.3 Linear density perturbations

We consider the linear perturbations for the scalar fields by expanding the field to the leading order, with $\phi_i(\vec{x}, t) = \phi_i(t) + \varphi_i(\vec{x}, t)$, where $\phi_i(t)$ are the background fields described in the above section, whereas φ_i are the field linear perturbations. The perturbed metric, in the synchronous gauge [3] is $ds^2 = -dt^2 + a^2(t)\left(\delta_{lm} + h_{lm}\right) dx^l dx^m$, with h_{lm} being the tensor perturbations of the metric. Working in Fourier space, the perturbed Klein-Gordon equation for each field is given by

$$\ddot{\varphi}_i = -3H\dot{\varphi}_i - \left(\frac{k^2}{a^2} + \partial^2_{\phi i} V_i \right) \varphi_i - \frac{1}{2}\dot{h}\dot{\phi}_i \, . \tag{3.9}$$

In Eq. (3.9), k is the comoving wavenumber, h is the trace of h_{lm} and $\dot{h}$ is known as the metric continuity. Following the idea presented for the background in the previous section, we use the polar variables [65, 95],

$$\sqrt{\frac{2}{3}} \frac{\kappa\dot{\varphi}_i}{H} = -\Omega^{1/2}_{\phi i} e^{\alpha_i} \cos\left(\frac{\vartheta_i}{2}\right) , \qquad \frac{\kappa y_{i,1}\varphi_i}{\sqrt{6}} = -\Omega^{1/2}_{\phi i} e^{\alpha_i} \sin\left(\frac{\vartheta_i}{2}\right) , \tag{3.10}$$

where α_i and ϑ_i are the new perturbation quantities. If we define the new quantities

$$\delta_{0i} = -e^{\alpha_i} \sin\left(\frac{\theta_i - \vartheta_i}{2}\right) , \qquad \delta_{1i} = -e^{\alpha_i} \cos\left(\frac{\theta_i - \vartheta_i}{2}\right) , \tag{3.11}$$

[3] The synchronous gauge is the most used to study cosmological perturbations and allows to solve the system numerically in a simpler way [65, 130].

where the density contrast is $\delta_{\phi i} \equiv \delta\rho_{\phi i}/\rho_{\phi i} = \delta_{0i}$, then the perturbed Klein-Gordon Equation (3.9) can be rewritten as

$$\begin{aligned} \delta_{0i}' &= \left[-3\sin(\theta_i) - \omega_i\left(1-\cos(\theta_i)\right)\right]\delta_{1i} + \omega_i \sin(\theta_i)\,\delta_{0i} - \frac{1}{2}h'\left(1-\cos(\theta_i)\right), \quad (3.12)\\ \delta_{1i}' &= \left[-3\cos(\theta_i) - \omega_i\sin(\theta_i) + \frac{y_{i,2}}{y_{i,1}}\Omega_{\phi_i}^{1/2}\sin\left(\frac{\theta_i}{2}\right)\right]\delta_{1i} \\ &+ \left[\omega_i\left(1+\cos(\theta_i)\right) - \frac{y_{i,2}}{y_{i,1}}\Omega_{\phi_i}^{1/2}\cos\left(\frac{\theta_i}{2}\right)\right]\delta_{0i} - \frac{1}{2}h'\sin(\theta_i)\,. \quad (3.13) \end{aligned}$$

Where ω_i and its respective differential equation (needed to close the system) are given by:

$$\begin{aligned} \omega_i &\equiv \frac{k^2}{a^2H^2y_{i,1}}, \quad (3.14)\\ \omega_i' &= \omega_i\left[\frac{3}{2}\omega_{tot} - \frac{1}{2} - \frac{y_{i,2}}{y_{i,1}}\Omega_{\phi_i}^{1/2}\sin\left(\frac{\theta_i}{2}\right)\right]. \quad (3.15) \end{aligned}$$

For the potentials we study here Equation (3.9) can be rewritten as

$$\begin{aligned} \delta_{0i}' &= -\left[3\sin\theta_i + \frac{k^2}{k_{Ji}^2}(1-\cos\theta_i)\right]\delta_{1i} + \frac{k^2}{k_{Ji}^2}\sin(\theta_i)\,\delta_{0i} - \frac{1}{2}h'\left(1-\cos\theta_i\right), \quad (3.16a)\\ \delta_{1i}' &= -\left[3\cos\theta_i + \left(\frac{k^2}{k_{Ji}^2} - \frac{\lambda_{\phi i}\Omega_{\phi i}}{2y_{1i}}\right)\sin\theta_i\right]\delta_{1i} + \left(\frac{k^2}{k_{Ji}^2} - \frac{\lambda_{\phi i}\Omega_{\phi i}}{2y_{1i}}\right)(1+\cos\theta_i)\,\delta_{0i} \\ &-\frac{1}{2}h'\sin\theta_i\,, \quad (3.16b) \end{aligned}$$

where we have introduced the Jeans wavenumber as $k_{Ji}^2 = H^2a^2y_{1i}$. Other quantities of interest are the perturbations for the energy density $\delta\rho_{\phi i}$, pressure $\delta p_{\phi i}$ and velocity divergence $\Theta_{\phi i}$, which are explicitly read

$$\begin{aligned} \delta\rho_{\phi i} &= \dot{\phi}_i\dot{\varphi}_i + \partial_{\phi i}V\varphi_i\,, \quad (3.17a)\\ \delta p_{\phi i} &= \dot{\phi}_i\dot{\varphi}_i - \partial_{\phi i}V\varphi_i\,, \quad (3.17b)\\ \left(\rho_{\phi i} + p_{\phi i}\right)\Theta_{\phi i} &= \frac{k^2}{a}\dot{\phi}_i\varphi_i\,. \quad (3.17c) \end{aligned}$$

In terms of the new variables δ_{0i} and δ_{1i}, they are written as

$$\begin{aligned} \delta\rho_{\phi i} &= \delta_{0i}\,\rho_{\phi i}\,, \quad (3.18a)\\ \delta p_{\phi i} &= \left(\delta_{1i}\sin\theta_i - \delta_{0i}\cos\theta_i\right)\,\rho_{\phi i}\,, \quad (3.18b)\\ \left(\rho_{\phi i} + p_{\phi i}\right)\Theta_{\phi i} &= \frac{k^2\rho_{\phi i}}{aHy_{i,1}}\left[\left(1-\cos\theta_i\right)\delta_{1i} - \sin\theta_i\delta_{0i}\right]. \quad (3.18c) \end{aligned}$$

Again, depending on the value of $\lambda_{\phi i}$, one recovers the perturbed equations of any of the three different potentials of this work (3.4). The initial conditions for the perturbations simply are $\delta_{0\mathrm{init}} = 0$ and $\delta_{1\mathrm{init}} = 0$. It has been shown that the dynamical variables δ_{0i} and δ_{1i} quickly reach an attractor behavior driven by the non-homogeneous term h' in Equations (3.16) [65, 95, 99].

3.4 Numerical results

In this section we show the background evolution, mass power spectrum (MPS) and CMB power spectrum for different combinations of potentials, obtained with a modified version of the CLASS code that is able to deal with multiple scalar fields [131, 65, 95]. This version of the code is publicly available and can be found in [132].

The MPS is the Fourier transform of the two-point correlation function of the distribution of matter and gives us information on the overdensities of matter and how much they grow. As we will see below, it gives us information about structure formation at different scales. In the same way, the CMB power spectrum give us information about the temperature fluctuations. Both of them are sensitive to dark matter. See [133, 134, 17, 135, 86] for details.

In the following results, we use the ratio $R = \Omega_{\phi 1,0}/\Omega_{\mathrm{DM},0}$ to parameterize the energy density of the scalar fields, where $\Omega_{\mathrm{DM},0} = \Omega_{\phi 1,0} + \Omega_{\phi 2,0} + \Omega_{\mathrm{cdm},0}$ represents the current total dark matter contribution from the scalar fields sector. The combination of the fields is symmetric, then for reference we take ϕ_1 to define R, otherwise if ϕ_j is taken as reference we need to redefine $R = \Omega_{\phi j}/\left(\sum_i \Omega_{\phi i} + \Omega_{\mathrm{cdm}}\right)$. The mass values for the fields were taken from the references mentioned above, in particular, those reported in [114], where the existence of at least two scalar fields with masses of 10^{-22} and 10^{-20} eV is proposed to explain the observations of the galaxy cores.

Throughout this work, we consider several combinations for the fields, and combined them with the CDM as well. This is because, until now, the nature of the dark matter is not yet determined and several possibilities should bear in mind. As the next approximation to the single field, here we assume two types of fields with different combinations of potentials as dark matter, however, once we have enough accurate data, the analysis (and the code) could be easily generalized to N different fields.

In the first combination we assume the field one has $V(\phi_1) = \frac{1}{2}m_{\phi 1}^2\phi_1^2$ while the rest of the dark matter density is conformed by CDM, therefore $\Omega_{\phi 2,0} = 0$. As we can see on the left panel of Figure 3.1, the main difference in the background is the start of the scalar field oscillations due to the mass and R values. If $R > 0.5$ the oscillations approach those of a single scalar field while if $R < 0.5$ the oscillations are less evident until they disappear when $R = 0$ (there is only CDM). In the mass power spectrum, the right panel of the same figure, we notice the expected cut-off in small scales for the lightest mass values ($m_{\phi 1} < 10^{-20}$ eV) but the behaviour is lost for heavier masses ($m_{\phi 1} > 10^{-20}$ eV) because the scalar field behaves like dust over the shown scales. Similar to the background, we found that the MPS of the combinations is bounded by the cases $R = 1$ (only SFDM) and $R = 0$ (only CDM). We can see an example of this in Figure 3.1 for $m_{\phi 1} = 10^{-22}$ eV (black line) and 10^{-20} eV (blue line) with $R = 0.2$ (dotted) and $R = 0.8$ (dashed). For $R = 0.8$ the oscillations approach the case of a single SFDM but disappear for $R = 0.2$, this is evident for $m_{\phi 1} = 10^{-22}$ eV. In the same way, in the right panel, when $R = 0.8$ the mass power spectrum shows a cut-off similar to the single case for both values $m_{\phi 1} = 10^{-22}$ and 10^{-20} eV, but it behaves like CDM (red solid line) for $R = 0.2$.

For the second combination we assume two scalar fields with no CDM, both of them with a quadratic potential $V(\phi_{1,2}) = \frac{1}{2}m_{\phi 1,2}^2\phi_{1,2}^2$ (see Figure 3.2). We found that the total contribution to the background evolution, in particular the oscillations of the fields, depends on the contribution of each one through R and are bounded by the oscillations of the lightest and the heaviest field respectively. For the MPS, the cut-off is more evident when the lightest field accounts for the principal contribution to $\Omega_{\rm DM,0}$ ($m_{\phi 2} > m_{\phi 1}$ and $R > 0.5$ or $m_{\phi 2} < m_{\phi 1}$ and $R < 0.5$), the behavior is closer to single field models ($R = 0$ or $R = 1$) depending of the dominant field, as expected. In the left panel of Figure 3.2

we show the total contribution of the fields to the background using $m_{\phi 1} = 10^{-22}$ eV with $m_{\phi 2} = 10^{-24}$ eV (green). The oscillations are more evident for $R = 0.8$ (dashed) than for $R = 0.2$ (dotted) and both of them are bounded by the single cases (solid). This is seen as well, but it is less noticeable due to the masses values, for the combination $m_{\phi 1} = 10^{-22}$ eV with $m_{\phi 2} = 10^{-20}$ eV (blue). For the MPS, we use the same values and we find a behavior similar to the previous case.

We also study a third combination of double field but now with $V(\phi_1) = \frac{1}{2}m_{\phi 1}^2\phi_1^2$ and $V(\phi_2) = m_{\phi 2}^2 f^2\left[1+\cos(\phi_2/f)\right]$ respectively. For the background we observe a similar behavior than in the previous combinations, that is, once we kept the masses fixed and solely modify the $\lambda_{\phi 2}$ values, we see no difference among them. The oscillations of the fields depend on the masses and R, similarly to the quadratic potentials case. See, for example, Figure 3.3 where we take $m_{\phi 1} = 10^{-22}$ eV along with $m_{\phi 2} = 10^{-24}$ eV with $\lambda_{\phi 2} = 10^4$ (green), $m_{\phi 2} = 10^{-22}$ eV with $\lambda_{\phi 2} = 10^5$ (black) and $m_{\phi 2} = 10^{-20}$ eV with $\lambda_{\phi 2} = 10^5$ (blue) for $R = 0.2$ (dotted) and $R = 0.8$ (dashed). For the MPS we see differences depending on $m_{\phi 1,2}$ values or R and also by varying $\lambda_{\phi 2}$ values. In order to see a bump in the MPS we need that the second field dominates ($R < 0.5$) with lighter mass and higher $\lambda_{\phi 2}$. In the right side of Figure 3.3 we can see the bump before the cut-off ($k \sim 0.9$) for the green and black dotted lines, because in both cases the second field dominates and $\lambda_{\phi 2}$ has high values with lighter masses. On the other hand, in the blue dotted line we do not observe the bump yet because the field becomes too heavy ($m_{\phi 2} \geq 10^{-20}$ eV).

Finally, we study the combination of the field potential $V(\phi_1) = \frac{1}{2}m_{\phi 1}^2\phi_1^2$ with $V(\phi_2) = m_{\phi 2}^2 f^2\left[\cosh(\phi_2/f) - 1\right]$. In the background and MPS we have a similar behavior as before. This can be seen in Figure 3.4 where we take the combination $m_{\phi 1} = 10^{-22}$ eV, $m_{\phi 2} = 1.54\times 10^{-22}$ eV, $\lambda_{\phi 2} = -8\times 10^3$ (black), $m_{\phi 1} = 10^{-22}$ eV, $m_{\phi 2} = 0.6\times 10^{-20}$ eV, $\lambda_{\phi 2} = -5\times 10^4$ (blue) and $m_{\phi 1} = 10^{-22}$ eV, $m_{\phi 2} = 0.3\times 10^{-18}$ eV, $\lambda_{\phi 2} = -4\times 10^4$ (gray). It is important to note that both fields have the same contribution ($R = 0.5$), other combinations require further analysis. In the mass power spectrum, at small scales, we see that $\lambda_{\phi 2}$ has a greater contribution compared to the previous case because in the cosh-like potential the scalar field mass depends on the $\lambda_{\phi 2}$ value, contrary to the cos-like potential in which these parameters are independent. When $m_{\phi 2} \sim m_{\phi 1}$ we can appreciate a different behavior depending on the value of λ_ϕ, for example, the black dotted line. However, if $m_{\phi 2} \geq m_{\phi 1}$, the change in $\lambda_{\phi 2}$ does not display noticeable changes (blue dotted and gray dotted lines).

In general for the background, we found slight differences at early times with respect to ΛCDM, where the oscillations presented could give us information about how light the scalar fields masses can be. We found too a cut-off at small scales in the mass power spectrum that differentiates our model from the CDM, and the shape below the cut-off depends on the multi-field dynamics. The cut-off tell us that not all structures can be formed at small scales.

Regarding the CMB spectrum, when the mass of at least one of the fields, for any combination, is less than $m_{\phi i} < 10^{-26}$ eV, the spectrum of the fields differs from the ΛCDM spectrum as seen in [100]. However, for masses greater than $m_{\phi i} > 10^{-26}$ eV the same CMB power spectrum is obtained as for ΛCDM regardless of the value of λ_ϕ or R. See, for example Figure 3.5 where we show the CMB power spectra for $V(\phi_{1,2}) = \frac{1}{2} m_{\phi 1,2}^2 \phi_{1,2}^2$ with $m_{\phi 1} = 10^{-22}$ eV, $m_{\phi 2} = 10^{-24}$ eV and $R = 0.8$ (green dashed line); $V(\phi_1) = \frac{1}{2} m_{\phi 1}^2 \phi_1^2$ with $V(\phi_2) = m_{\phi 2}^2 f^2 \left[1 + \cos(\phi_2/f)\right]$ for $m_{\phi 1} = 10^{-22}$ eV, $m_{\phi 2} = 10^{-22}$ eV, $\lambda_{\phi 2} = 10^5$ and $R = 0.2$ (black dotted line) and $V(\phi_1) = \frac{1}{2} m_{\phi 1}^2 \phi_1^2$ with $V(\phi_2) = m_{\phi 2}^2 f^2 \left[\cosh(\phi_2/f) - 1\right]$ for $m_{\phi 1} = 10^{-22}$ eV, $m_{\phi 2} \propto 10^{-20}$ eV, $\lambda_{\phi 2} = -5 \times 10^4$ and $R = 0.5$ (blue dotted line). For the three cases we have obtained nearly the same CMB power spectra as for the ΛCDM model (solid red line). We also plotted the Planck-18 dataset as reference [136].

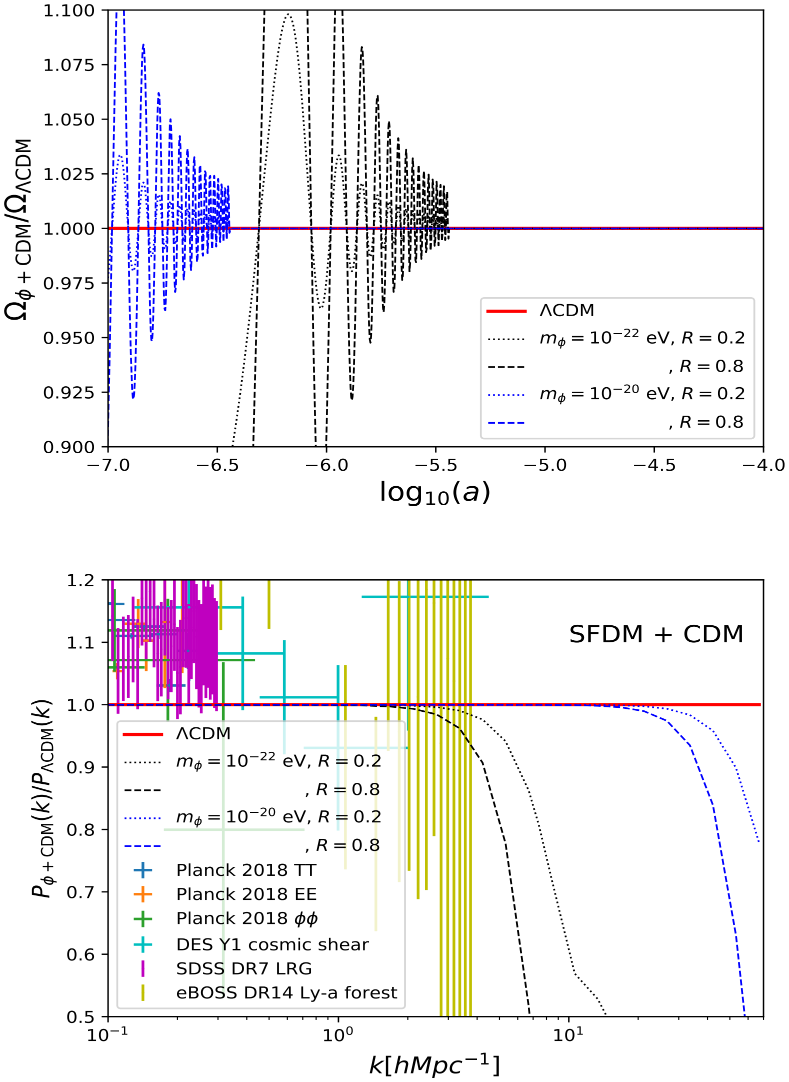


Figure 3.1: Evolution, zoomed-in on early times, of the ratio of the density parameters $\Omega_{\phi+\mathrm{CDM}}/\Omega_{\Lambda\mathrm{CDM}}$ (left) and the ratio of the linear matter power spectrum (right) at $z = 0$, for a SFDM + CDM model using ΛCDM as reference (solid red lines). The field potential is the quadratic one $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$, and R represents the ratio of the field contribution to the total DM. Black lines represent a mass value of $m_\phi = 10^{-22}$ eV while blue lines refer to $m_\phi = 10^{-20}$ eV. Dashed and dotted lines represent $R = 0.8$ and $R = 0.2$ values respectively. The data points for the MPS were obtained from [86].

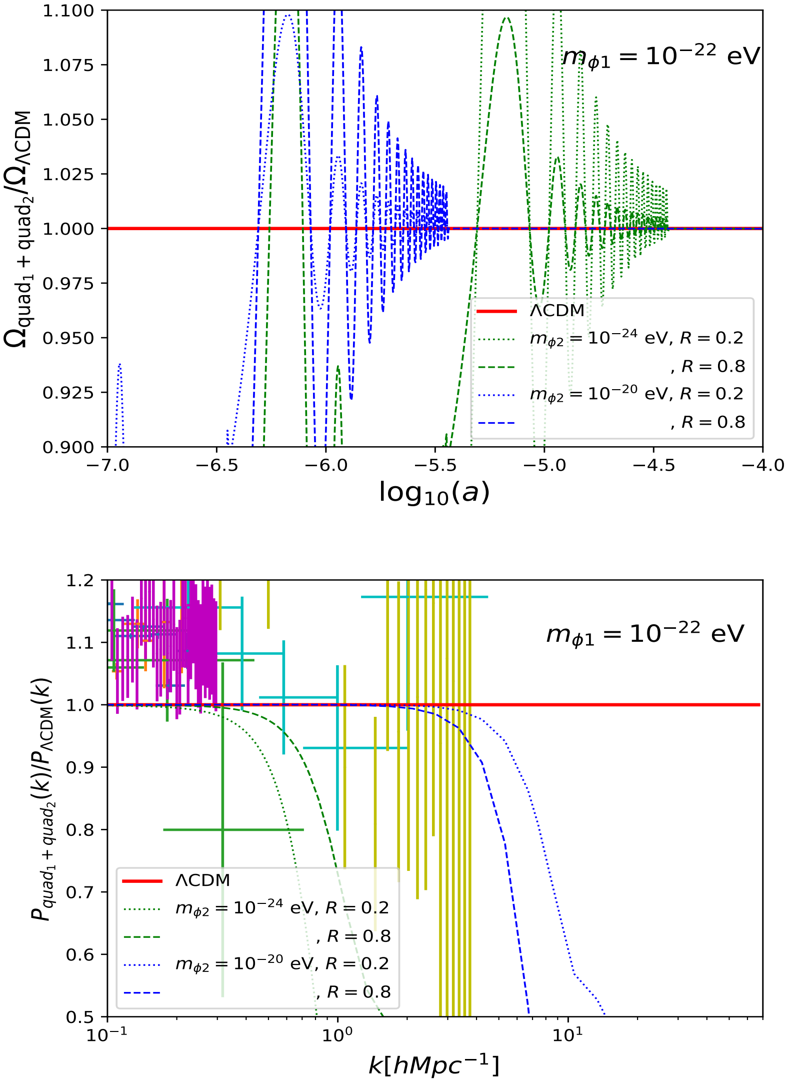


Figure 3.2: Evolution, zoomed-in on early times, of the ratio of the density parameters $\Omega_{quad_1+quad_2}/\Omega_{\Lambda\mathrm{CDM}}$ (left) and the ratio of the linear matter power spectrum (right) at $z = 0$, for a double field model using ΛCDM as reference (solid red lines). The potential for both fields is the quadratic one $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$, and R represents the ratio of the fields contribution to the total DM. Green lines represent a mass value of $m_{\phi 2} = 10^{-24}$ eV and blue lines refer to $m_{\phi 2} = 10^{-20}$ eV while $m_{\phi 1}$ is fixed to 10^{-22} eV. Dashed and dotted lines represent $R = 0.8$ and $R = 0.2$ values respectively. The data points for the MPS were obtained from [86]. MPS data labels are the same as in Figure 3.1.

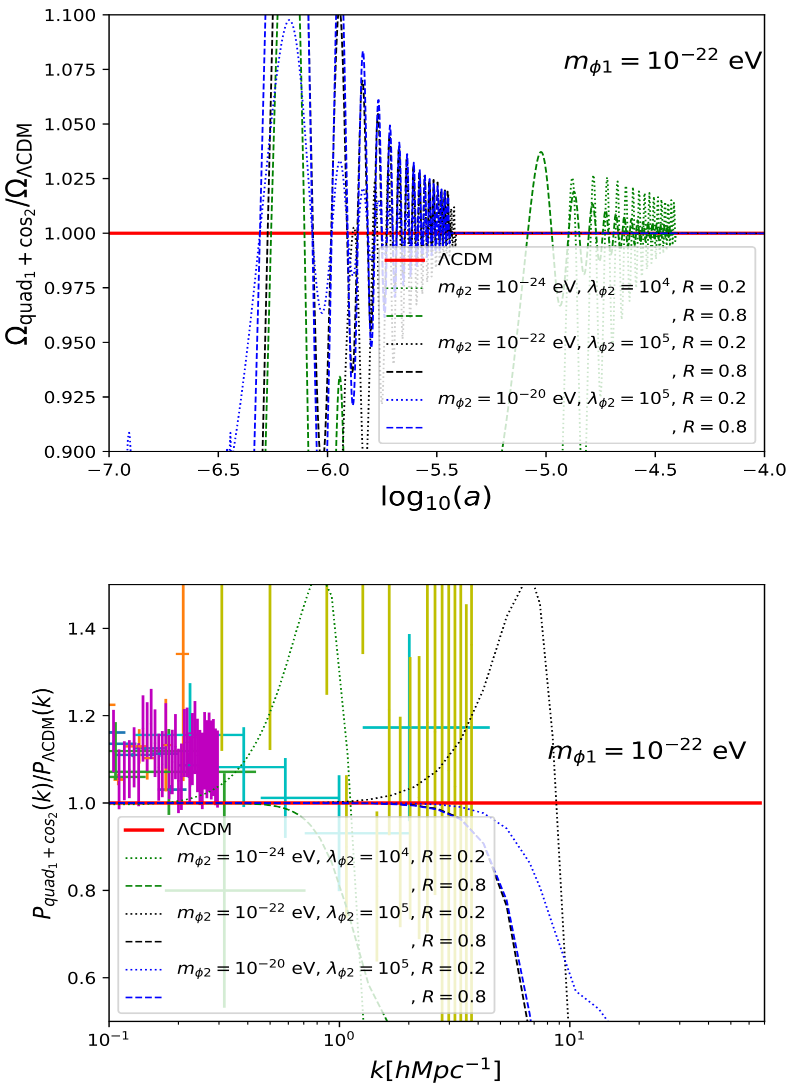


Figure 3.3: Evolution, zoomed-in on early times, of the ratio of the density parameters $\Omega_{quad_1+cos_2}/\Omega_{\Lambda\mathrm{CDM}}$ (left) and the ratio of the linear matter power spectrum (right) at $z = 0$, for a model with two fields using ΛCDM as reference (solid red lines). Here, the field one has the potential $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ with mass value $m_\phi = 10^{-22}$ eV as reference. While the second field has $V(\phi_2) = m_{\phi 2}^2 f^2 \left[1 + \cos(\phi_2/f)\right]$. Green lines represent a mass value of $m_{\phi 2} = 10^{-24}$ eV with $\lambda_{\phi 2} = 10^4$, black lines indicate $m_{\phi 2} = 10^{-22}$ eV with $\lambda_{\phi 2} = 10^5$ and blue lines refer to $m_{\phi 2} = 10^{-20}$ eV with $\lambda_{\phi 2} = 10^5$. Dashed and dotted lines represent $R = 0.8$ and $R = 0.2$ values respectively, where R represents the ratio of the fields contribution to the total DM. The data points for the MPS were obtained from [86]. MPS data labels are the same as in Figure 3.1.

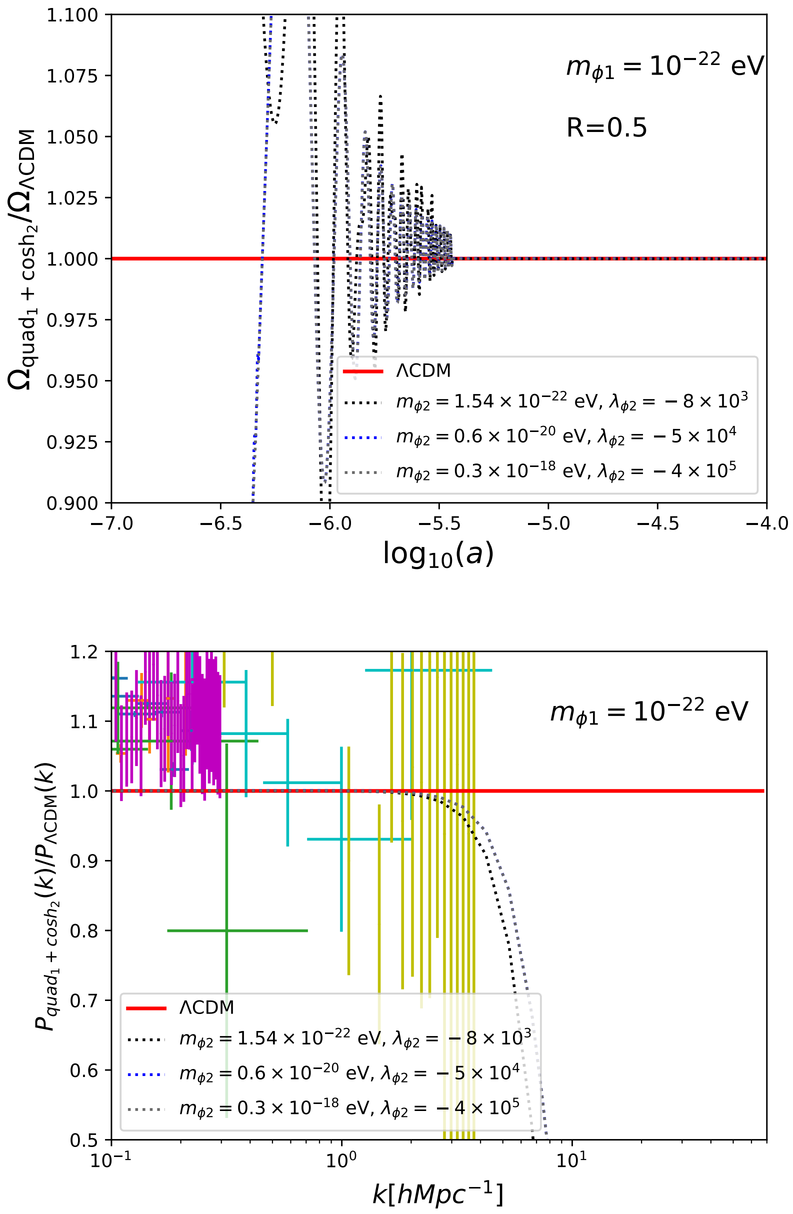


Figure 3.4: Evolution, zoomed-in on early times, of the ratio of the density parameters $\Omega_{quad_1+cosh_2}/\Omega_{\Lambda\mathrm{CDM}}$ (left) and the ratio of the linear matter power spectrum (right) at $z = 0$, for a model with two fields using ΛCDM as reference (solid red lines). The field one has the potential $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ with mass value $m_\phi = 10^{-22}$ eV as reference, and the second field has $V(\phi_2) = m_{\phi 2}^2 f^2 \left[\cosh(\phi_2/f) - 1\right]$. Black lines represent a mass value of $m_{\phi 2} = 1.54 \times 10^{-22}$ eV with $\lambda_{\phi 2} = -8 \times 10^3$, blue lines indicate $m_{\phi 2} = 0.6 \times 10^{-20}$ eV with $\lambda_{\phi 2} = -5 \times 10^4$ and green lines refer to $m_{\phi 2} = 0.3 \times 10^{-18}$ eV with $\lambda_{\phi 2} = -4 \times 10^5$. Dotted lines represent $R = 0.5$ where R represents the ratio of the fields contribution to the total DM. The data points for the MPS were obtained from [86]. MPS data labels are the same as in Figure 3.1.

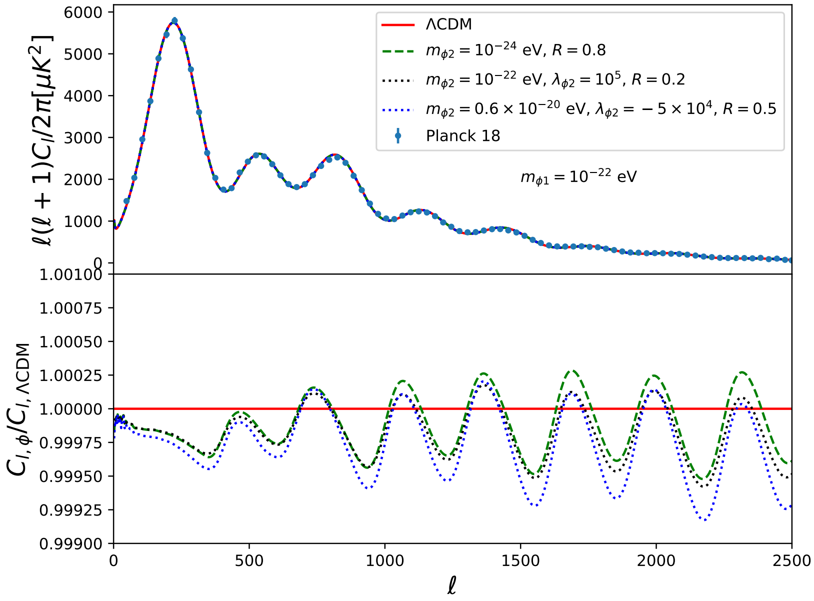


Figure 3.5: The CMB power spectrum (upper panel) and the ratio (lower panel) using ΛCDM as reference (solid red lines), for $V(\phi_{1,2}) = \frac{1}{2}m^2_{\phi 1,2}\phi^2_{1,2}$ (green dashed line), $V(\phi_1) = \frac{1}{2}m^2_{\phi 1}\phi^2_1$ with $V(\phi_2) = m^2_{\phi 2}f^2\left[1 + \cos(\phi_2/f)\right]$ (black dotted line) and $V(\phi_1) = \frac{1}{2}m^2_{\phi 1}\phi^2_1$ with $V(\phi_2) = m^2_{\phi 2}f^2\left[\cosh(\phi_2/f) - 1\right]$ (blue dotted line).

Chapter 4

Parameter inference with Bayesian statistics

In the previous chapter we numerically studied the evolution of the background and the linear perturbations of our model using different combinations and values of the parameters that we put in by hand. However, to get a complete picture we want to know, given cosmological observations, what is the most likely value that the parameters of our model can take. However, since we have a unique Universe, we cannot rely on a frequentist interpretation of statistics (we are not able to create multiple Universes and make a frequentist inference). An alternative approach that will help in our task is the Bayesian statistics. In Bayesian statistics, the probability is interpreted as a "degree of belief", and it may be useful when repetitive processes are complicated to reproduce.

4.1 Frequentist and Bayesian statistics

The difference between both descriptions is on the definition of probability. From the frequentist statistic, probability has meaning in limiting cases of repeated measurements

$$P = \frac{n}{N}, \tag{4.1}$$

where n denotes the number of successes, and N the total number of trials. This is, we define the probability P as the limit of the number of independent trials going to infinity. Then, for frequentist statistics, probabilities are fundamentally related to frequencies of events. While, in Bayesian statistics, the concept of probability is extended to cover degrees of certainty about a statement. For Bayesian statistics, probabilities are fundamentally related to our knowledge concerning an event.

To understand the consequences of these differences we introduce some key concepts. However, see references [137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 22] and references therein for details.

Let x be a random variable related to a particular event and $P(x)$ its corresponding probability distribution, for both cases, the same rules of probabilities apply [1]:

$$P(x) \geq 0, \tag{4.2}$$

$$\int_{-\infty}^{\infty} dx P(x) = 1. \tag{4.3}$$

For mutually exclusive events, we have

$$P(x_1 \cup x_2) = P(x_1) + P(x_2), \tag{4.4}$$

but, in general

$$P(x_1 \cup x_2) = P(x_1) + P(x_2) - P(x_1 \cap x_2). \tag{4.5}$$

The meaning of these rules is that Equation (4.2) is necessary due to the probability of having an event is always positive; Equation (4.3) is a normalized relation, which tells us that we are certain to obtain one of the possible outcomes; in Equation (4.4) we have that the probability of obtaining an observation, from a set of mutually exclusive events, is given by the individual probabilities of each event; finally, and in general, if one event occurs given the occurrence of another then the probability that both x_1 and x_2 happen is equal to the probability of x_1 times the probability of x_2 given that x_1 has already happened

$$P(x_1 \cap x_2) = P(x_1)P(x_2|x_1). \tag{4.6}$$

[1] The corresponding discrete definition can be given by replacing $\int dx \to \sum$.

If two events x_1 and x_2 are mutually exclusive, then

$$P(x_1 \cap x_2) = 0 = P(x_2 \cap x_1). \tag{4.7}$$

The rules of probability distributions must be fulfilled by both frequentist and Bayesian statistics. However, there are some consequences derived by the fact that these two scenarios have a different definition of probability.

4.1.1 Frequentist statistics

In the frequentist inferential procedure we need these basic ingredients: the data, the model and an estimation procedure. The main assumption in frequentist statistics is that the data has a definite, but unknown, underlying distribution to which all inference pertains.

The data is a measurement or observation, denoted by X, that can take any value from a corresponding sample space. A sample space of an observation X can be defined as a measurable space $(x, \hat{B})$ that contains all values that X can take upon measurement. In this point of view, it is considered that there is a probability function $P_0 : \hat{B} \to [0, 1]$ in the sample space $(x, \hat{B})$ representing the "true distribution of the data"

$$X \sim P_0.$$

On the other hand, the model Q is a collection of probability measurements $P_\theta : \hat{B} \to [0, 1]$ in the sample space $(x, \hat{B})$. The distributions P_θ are called *model distributions*, with θ being the model parameters; in frequentist statistics θ is unchanged. A model Q is said to be well-specified if it contains P_0, i.e.,

$$P_0 \in Q.$$

Finally, we need a point-estimator (or estimator) for P_0. This is, we need a map $\hat{P} : x \to Q$, representing our best guess $\hat{P} \in Q$ for P_0 based on the data X. Therefore, the frequentist statistics is based on trying to answer the following questions: "what the data is trying to tell about P_0?" or "considering the data, what can we say about the mean value of P_0?"

As we saw above, in this formalism, the probability depends on the frequency. So, for this formalism to be valid, we must assume that we have a hypothetical infinite sequence of events and hence we know the probability by assuming how many times we get the desired result from this infinite sequence. For example, if we have a die and we want to know the probability that any of the sides will come up, we must assume that the die has been thrown infinitely many times and that, on average, each side has come up one sixth of the times. This gives us that the probability is 1/6 for each side.

We must be careful not to confuse frequentist probability with classical probability. In the classical view, we assume that all options are equally likely to occur. For example, in the case of dice, we can assume that all sides have the same probability of occurring which gives us 1/6 for each. Although the probability is the same, the assumptions made to arrive at it are different.

However, frequentist statistics start to break down when we ask questions like, for example, what is the probability that it will rain tomorrow. We need to imagine an infinite succession of mornings and see what fraction of them have rain, which is not very intuitive.

4.1.2 Bayesian statistics

On the other hand, in the Bayesian point of view, data and model are elements of the same space [137], there is no formal distinction between measured quantities X and parameters θ. One may envisage the process of generating a measurement's outcome $Y = y$ as two draws, one draw for Θ, where Θ is a model with associated probabilities to the parameter θ, to select a value of θ and a subsequent draw for P_θ to arrive at $X = x$. This perspective may seem rather absurd when thinking in a Frequentist way, but, in Bayesian statistics, where probabilities are related to our own knowledge, it results natural to associate probability distributions to our parameters. In this way, an element P_θ of the model is interpreted simply as the distribution of X given the parameter value θ, i.e., as the conditional distribution $X|\theta$.

Bayesian probability is based on the synthesis of evidence and personal judgment in looking at that evidence. It is based on personal knowledge that we have and can change from one person to another.

As an example, let us consider Chile whose land area is 756 950 km^2 and suppose we divide it into four regions:

$$\begin{aligned} A_1 &\rightarrow < 10\ 000 \text{ km}^2, \\ A_2 &\rightarrow [10\ 000, 50\ 000] \text{ km}^2, \\ A_3 &\rightarrow [50\ 000, 100\ 000] \text{ km}^2, \\ A_4 &\rightarrow > 100\ 000 \text{ km}^2. \end{aligned} \tag{4.8}$$

We want to know how big the Atacama desert is. Let's start by putting probabilities depending on what we know about this desert. In my case, I know that the Atacama desert is big but I don't know how big it is, so I choose

$$\begin{aligned} P(A_1) &= 0.1, \\ P(A_2) &= 0.4, \\ P(A_3) &= 0.4, \\ P(A_4) &= 0.1. \end{aligned} \tag{4.9}$$

Now, the desert is known to be the fourth largest region in Chile so I'll modify the probabilities by

$$\begin{aligned} P(A_1) &= 0.1, \\ P(A_2) &= 0.1, \\ P(A_3) &= 0.4, \\ P(A_4) &= 0.4. \end{aligned} \tag{4.10}$$

Frequentist	**Bayesian**
Data are a repeatable random sample. There is a frequency.	Data are observed from the realized sample.
Underlying parameters remain constant during this repeatable process.	Parameters are unknown and described probabilistically.
Parameters are fixed.	Data are fixed.

Table 4.1: Main differences between the Bayesian and Frequentist interpretations.

On the other hand, Santiago is the smallest region in the country with 15 403 km^2. This leads me to modify the probabilities again by

$$\begin{aligned} P(A_1) &= 0, \\ P(A_2) &= 0.2, \\ P(A_3) &= 0.4, \\ P(A_4) &= 0.4. \end{aligned} \tag{4.11}$$

While Aysén is the third largest region with an area of 108 494 km^2. Finally, I'll change again

$$\begin{aligned} P(A_1) &= 0, \\ P(A_2) &= 0.2, \\ P(A_3) &= 0.7, \\ P(A_4) &= 0.1. \end{aligned} \tag{4.12}$$

This example shows us that the Bayesian probability changes depending on the knowledge we have.

Finally, table 4.1 provides a short summary of the most important differences between the two statistics.

4.2 Bayes' theorem, priors, posteriors and other concepts

The Bayes' theorem is a direct consequence of the axioms of probability shown in Equations (4.2) - (4.6). From Equation (4.6), without loss of generality, it must be fulfilled that $P(x_1 \cap x_2) = P(x_2 \cap x_1)$. In such a case, the following relation applies:

$$P(x_2|x_1) = \frac{P(x_1|x_2)P(x_2)}{P(x_1)}. \tag{4.13}$$

As mentioned above, in the Bayesian framework, data and model are part of the same space. Given a model or hypothesis H, with $x_1 \to D$ as a set of data, and $x_2 \to \theta$ as the parameter vector of the model, we can rewrite the above equation as

$$P(\theta|D,H) = \frac{P(D|\theta,H)P(\theta|H)}{P(D|H)}. \tag{4.14}$$

This last equation is known as the Bayes' theorem and is the most important tool in a Bayesian inference procedure. In this result, $P(\theta|D,H)$ is called the posterior probability of the model. $L(D|\theta,H) \equiv P(D|\theta,H)$ is the likelihood, $\pi(\theta) \equiv P(\theta|H)$ is called the prior and expresses the knowledge about the model before acquiring the data (this prior can be fixed depending on either previous experiment results or the theory behind), $\mathcal{Z} \equiv P(D|H)$ is the evidence of the model, usually referred to as the Bayesian Evidence.

The prior refers to the information we have a priori of the model. It can be defined in various ways; however, a common one is the uniform prior or flat prior:

$$\pi(\theta) \propto c, \tag{4.15}$$

with c being a constant. This type of prior is telling us that every parameter value is equally probable a priori. Using this prior also means that the posterior probability will be proportional to the likelihood (since the Bayesian Evidence is a constant). We can define different types of priors, however, in this work we will consider only the flat type. It is important to mention that the choice of a prior will influence the posterior distribution, although its effect is reduced as more data are collected.

On the other hand, regarding the Bayesian Evidence, we notice that it acts as a normalizing factor, and is given by the average of the likelihood over the prior

$$P(D|H) = \int d^N\theta P(D|\theta, H)P(\theta|H), \tag{4.16}$$

where N is the dimensionality of the parameter space. The Bayesian evidence plays an important role for selecting the model that best describes the data, this process being known as model selection. For convenience, the ratio of two evidences

$$K \equiv \frac{P(D|H_0)}{P(D|H_1)} = \frac{\int d^{N_0}\theta_0 \; P(D|\theta_0, H_0)P(\theta_0|H_0)}{\int d^{N_1}\theta_1 \; P(D|\theta_1, H_1)P(\theta_1|H_1)} = \frac{\mathcal{Z}_0}{\mathcal{Z}_1}, \tag{4.17}$$

or equivalently the difference in log evidence $\ln \mathcal{Z}_0 - \ln \mathcal{Z}_1$ is often termed as the Bayes factor $\mathcal{B}_{0,1}$:

$$\mathcal{B}_{0,1} = \ln \frac{\mathcal{Z}_0}{\mathcal{Z}_1}, \tag{4.18}$$

where θ_i is a parameter vector (with dimensionality N_i) for the hypothesis H_i and $i = 0, 1$. In Equation (4.18), the quantity $\mathcal{B}_{0,1} = \ln K$ provides an idea on how well model 0 may fit the data when compared to model 1. Jeffreys provided a suitable guideline scale on which we are able to make qualitative conclusions, see Table 4.2 [147].

Bayes' theorem has an enormous implication with respect to a statistical inferential point of view. Usually, we collect some data and then interpret it with a given model; however, we sometimes do the opposite. That is, as we will see in the next chapter, first we have a set of data, and then we confront a model considering the probability that our model fits the data. Bayes' theorem provides a tool to relate both scenarios. Then, with this theorem, we are able to select the model that best fits the data.

$\vert\mathcal{B}_{0,1}\vert$	Odds	Probability	Strength
< 1.0	$< 3:1$	< 0.750	Inconclusive
1.0-2.5	$\sim 12:1$	0.923	Significant
2.5-5.0	$\sim 150:1$	0.993	Strong
> 5.0	$> 150:1$	> 0.993	Decisive

Table 4.2: Jeffreys guideline scale for evaluating the strength of evidence when two models are compared.

4.2.1 The Likelihood

When testing a single model the Bayesian evidence is usually set apart. Then, without loss of generality, we can fix it to $P(D|H) = 1$ and if we ignore the prior (when we have a large amount of data), we can identify the posterior with the likelihood $P(\theta|D, H) \propto L(D|\theta, H)$; thus, by maximizing it, we can find the most probable set of parameters for a model given the data. However, having ignored the Bayesian evidence and the prior, we are not able to provide an absolute probability for a given model, but only relative probabilities. On the other hand, it is possible to report results independently of the prior by using the Likelihood ratio. The likelihood at a particular point in the parameter space can be compared with the best-fit value, or the maximum likelihood $L_{\rm max}$. Then, we can say that some parameters are acceptable if the likelihood ratio

$$\Lambda = -2 \ln \left[\frac{L(D|\theta, H)}{L_{\rm max}} \right], \tag{4.19}$$

is bigger than a given value.

Suppose we have a single-peaked distribution whose mean is given by

$$\hat{\theta} = \int d\theta \theta P(\theta|D, H). \tag{4.20}$$

If the model is well-specified and the expectation value of $\hat{\theta}$ corresponds to the real or most probable value θ_0, we have

$$\langle\hat{\theta}\rangle = \theta_0, \tag{4.21}$$

then we say that $\hat{\theta}$ is unbiased. If we do the Taylor expansion of the log likelihood around its maximum, we have

$$\begin{aligned}\ln L(D|\theta) &= \ln L(D|\theta_0) \\ &+ \frac{1}{2}(\theta_i - \theta_{0i})\frac{\partial^2 \ln L}{\partial\theta_i\partial\theta_j}(\theta_j - \theta_{0j}) \\ &+ ...,\end{aligned} \tag{4.22}$$

where θ_0 is the parameter vector of the real model. In this way, we can express the likelihood as a multi-variable likelihood given by

$$L(D|\theta) = L(D|\theta_0)\exp\left[-\frac{1}{2}(\theta_i - \theta_{0i})H_{ij}(\theta_j - \theta_{0j})\right], \tag{4.23}$$

where

$$H_{ij} = -\frac{\partial^2 \ln L}{\partial\theta_i\partial\theta_j}, \tag{4.24}$$

is called the Hessian matrix. It controls whether the estimates of θ_i and θ_j are correlated, and if it is diagonal, they are are uncorrelated.

The above expression for the likelihood is a good approximation as long as our posterior distribution possesses a single-peak. On the other hand, if the data errors are normally distributed, then the likelihood for the data will be a Gaussian function as well. In fact, this is true if the model is linearly dependent on the parameters. However, if the data is not normally distributed, we can use the central limit theorem to justify that the resulting distribution will be best approximated by a multi-variate Gaussian distribution [140].

4.2.2 The Chi-square and goodness of fit

The main aim of parameter estimation is to maximize the likelihood in order to obtain the most probable set of model parameters given the data. Considering the Gaussian approximation in Equation (4.23), we can see the likelihood will be maximum if the quantity

$$\chi^2 \equiv (\theta_i - \theta_{0i})H_{ij}(\theta_j - \theta_{0j}), \tag{4.25}$$

is minimum. The quantity χ^2 is called chi-square and is related to the Gaussian likelihood via $L = L_0 e^{-\chi^2/2}$. We can say that maximizing the Gaussian likelihood is equivalent to minimizing the chi-square. However, there are some circumstances where the likelihood cannot be described by a Gaussian distribution, in these cases the chi-square and the likelihood are no longer equivalent.

The probability distribution for different values of χ^2 around its minimum is given by the χ^2 distribution for $v = n - M$ degrees of freedom, where n is the number of independent data points and M the number of parameters. We can calculate the probability that an observed χ^2 exceeds a value $\hat{\chi}$ for the correct model. This probability is given by $Q(v, \hat{\chi}) = 1 - \Gamma(v/2, \hat{\chi}/2)$ [148], where Γ is the incomplete Gamma function. Then, the probability that the observed χ^2 (even the correct model) is less than a given value $\hat{\chi}^2$ is $1 - Q$. This statement is strictly true for Gaussian likelihoods.

Given Q evaluated with the best-fit values (minimum chi-square), we can have a measure of the goodness of fit. If Q is small (small probability), we have:

- The model is wrong and can be rejected.
- The errors are underestimated.
- The error measurements are not normally distributed.

While, if Q is too large we can interpret it as:

- Errors have been overestimated.
- Data are correlated or non-independent.
- The distribution is non-Gaussian.

4.2.3 Contour plots and confidence regions

Once the best fit parameters are obtained, we would like to know the confidence regions where values could be considered good candidates. We take values inside a compact region

			$\Delta\chi^2$	
σ	p	$M=1$	$M=2$	$M=3$
1	68.3%	1.00	2.30	3.53
2	95.4%	4.00	6.17	8.02
3	99.73%	9.00	11.8	14.20

Table 4.3: $\Delta\chi^2$ for the conventional 68.3%, 95.4% and 99.73% as a function of the number of parameters (M) for the joint confidence level.

around the best fit value with constant χ^2 boundaries. When the χ^2 has more than one minimum, it is said that we have non-connected confidence regions, and for multi-variate Gaussian distributions (as the likelihood approximation in Equation (4.23)) these are ellipsoidal regions. Following Reference [140], we show an example of how to calculate the confidence regions: we a small perturbation from the best fit of chi-square $\Delta\chi^2 = \chi^2 - \chi^2_{\text{best}}$ and, using the properties of χ^2 distribution we define confidence regions for variations on χ^2 to its minimum. In Table 4.3, we see the 68.3%, 95.4%, and 99.73% confidence levels as a function of number of parameters M for the joint confidence level. For Gaussian distributions, these are the conventional 1, 2, and 3 σ confidence levels.

The steps to compute constant χ^2 confidence regions is as follows: after finding the best fit by minimizing χ^2 (or maximizing the likelihood) and checking that Q is acceptable for the best parameters, then:

1. Be M the number of parameters, n the number of data and p the confidence limit desired.

2. Solve the equation:

$$Q(n-M, min(\chi^2) + \Delta\chi^2) = p. \tag{4.26}$$

3. Find the parameter region where $\chi^2 \leq min(\chi^2) + \Delta\chi^2$. This defines the confidence

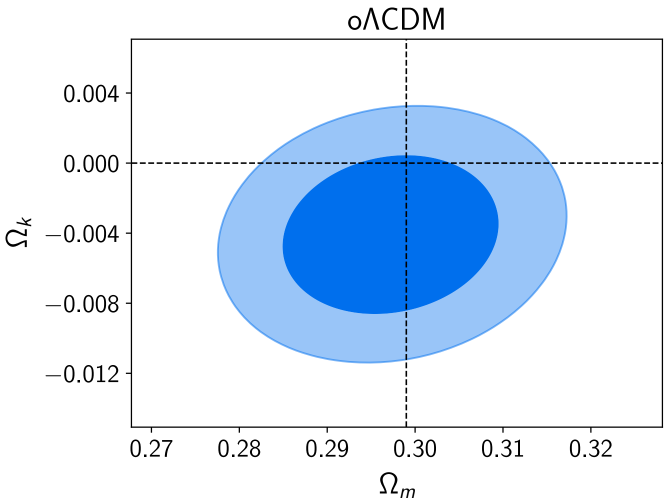


Figure 4.1: Example of a confidence region. We see the 1 and 2 σ confidence level for the parameters Ω_m and Ω_k for the model oΛCDM where curvature is an extra parameter. Dotted lines correspond to the value for ΛCDM. Taken from [11].

region.

Leaving aside the model, in Figure 4.1 we see an example of confidence regions at 1 and 2 σ where the most probable values of the parameters shown can be found.

4.2.4 Marginalization

In general, a model depends on more than one parameter. Sometimes, some of these parameters θ_i may be of less interest. For example, they may correspond to nuisance parameters, like calibration factors, or it may be the case that we are interested in only one of the parameter constraints rather than the joint of two or more of them simultaneously. Then, we marginalize over the uninteresting parameters by

$$P(\theta_1, ..., \theta_j, H|D) = \int d\theta_{j+1}...d\theta_m P(\theta, H|D), \tag{4.27}$$

where m is the total number of parameters in the model, and θ_1,...,θ_j denote the parameters we are interested in.

4.3 Numerical tools

Most of the time, it is very difficult to compute the posterior distribution analytically. In particular in Cosmology where, even the simplest model, they have six free parameters. For these cases, and as we saw in the section 3.4, the numerical tools play an important role. There are several options to carry out this work; nevertheless, in this section, we focus on the Markov Chain Monte Carlo (MCMC) with the Metropolis Hastings algorithm (MHA).

4.3.1 MCMC techniques

The purpose of the MCMC algorithm is to build up a sequence of points, the chain, in a parameter space in order to evaluate the posterior of Equation (4.14). Here, we review the basic steps for this procedure; however, it is recommendable to check [149, 150, 151, 152, 153] for a more formal version of the MCMC theory.

Monte Carlo simulations are algorithms that use random number generators to approximate a specific quantity. On the other hand, a sequence $X_1, X_2, \ldots$ of elements of some set is a Markov Chain if the conditional distribution of X_{n+1} given $X_1, \ldots, X_n$ depends only on X_n. In other words, a Markov Chain is a process where we can compute subsequent steps based only in the information given at the present. An important property of a Markov Chain is that it converges to a stationary state where successive elements of the chain are samples from the target distribution; in our case, it converges to the posterior $P(\theta|D,H)$. Hence, we can estimate all the usual quantities of interest out of the posterior (mean, variance, etc.). The combination of both procedures is called an MCMC. The number of points required to get good estimates in MCMCs scales linearly with the number of parameters, so this method becomes much faster than grids as the dimensionality increases.

The target density is approximated by a set of delta functions:

$$p(\theta|D,H) \simeq \frac{1}{N}\sum_{i=1}^{N}\delta(\theta-\theta_i), \tag{4.28}$$

where N is the number of points in the chain. Then, the posterior mean is computed as

$$\langle \theta \rangle = \int d\theta \theta P(\theta, H|D) \simeq \frac{1}{N}\sum_{i=1}^{N} \theta_i, \tag{4.29}$$

where $\simeq$ follows because the samples θ_i are generated out of the posterior by construction. So, we can estimate any integrals (such as the mean, variance, etc.) as

$$\langle f(\theta) \rangle \simeq \frac{1}{N}\sum_{i=1}^{N} f(\theta_i). \tag{4.30}$$

In a Markov Chain, it is necessary to generate a new point θ_{i+1} from the present point θ_i. However, we need a criterion for accepting or rejecting this new point depending on whether it turns out to be better for our model or not. If the new step is worse than the previous one, we may still accept it since, if we only accept steps with better probability, we could be converging into a local maximum in our parameter space and, therefore, just exploring a small region of the entire space. The simplest algorithm that contains all this information in its methodology is known as the Metropolis-Hastings algorithm.

4.3.2 Metropolis-Hastings algorithm

In the Metropolis-Hastings algorithm [154, 155], we need to start from a random initial point θ_i, with an associated posterior probability $p_i = p(\theta_i|D, H)$. It is necessary to propose a candidate θ_c by drawing from a proposal distribution $q(\theta_i, \theta_c)$ used as a generator of new random steps. Then, the probability of acceptance of the new point is given by

$$p(acceptance) = \min\left[1, \frac{p_c q(\theta_c, \theta_i)}{p_i q(\theta_i, \theta_c)}\right]. \tag{4.31}$$

If the proposal distribution is symmetric, the algorithm is reduced to the Metropolis algorithm

$$p(acceptance) = \min\left[1, \frac{p_c}{p_i}\right]. \tag{4.32}$$

We generate a random number between 0 and 1. If this number is less than p, the proposal θ_c is accepted. Otherwise it is rejected and the method is repeated.

The complete algorithm can be expressed by the following steps:

1. Choose a random initial condition θ_i in the parameter space and compute the posterior distribution.

2. Generate a new candidate from a proposal distribution in the parameter space and compute the corresponding posterior distribution.

3. Accept or reject the new point with the help of the Metropolis-Hastings algorithm.

4. If the point is rejected, repeat the previous point in the chain.

5. Repeat steps 2–4 until you have a large enough chain.

4.3.3 Convergence test

Just as we need a criterion to accept or reject the points of the chains, we also need a criterion to know if our chains have converged. We need to verify that the points in the chain are not converging to a false convergent point or to a local maximum. In this sense, we need that our algorithm takes into account this possible difficulty.

The usual test is the Gelman-Rubin convergence criterion [156, 157]. That is, by starting with M chains with very different initial points and N points per chain, if θ_i^j is a point in the parameter space of position i and belonging to the chain j, we need to compute the mean of each chain:

$$\langle \theta^j \rangle = \frac{1}{N} \sum_{i=1}^{N} \theta_i^j, \tag{4.33}$$

and the mean of all the chains

$$\langle \theta \rangle = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \theta_i^j. \tag{4.34}$$

Then, the chain-to-chain variance B is given by

$$B = \frac{1}{M-1} \sum_{j=1}^{M} (\langle \theta^j \rangle - \langle \theta \rangle)^2, \tag{4.35}$$

and the average variance of each chain is

$$W = \frac{1}{M(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{M} (\theta_i^j - \langle \theta^j \rangle)^2. \tag{4.36}$$

If our chains converge, W and B/N must agree. In fact, we say that the chains converge when the quantity

$$\hat{R} = \frac{\frac{N-1}{N} W + B(1 + \frac{1}{M})}{W}, \tag{4.37}$$

which is the ratio of the two estimates, approaches unity. A typical convergence criterion is when $0.97 < \hat{R} < 1.03$.

Chapter 5

Multi Scalar Field Dark Matter: Cosmological Constraints

In this chapter we present the Bayesian inference procedure in order to constrain the MSFDM models by using different data sets. In order to perform the cosmological analysis, we generate Monte Carlo Markov Chains with the Metropolis-Hastings algorithm using a modified version of the Monte Python code [158, 159] with the modified version of CLASS used in subsection 3.4. We verified that our chains converged using the Gelman-Rubin criterion $R - 1 < 0.03$ implemented in Monte Python. Then we discuss the merits of the models with respect to CDM within the framework of the Bayesian model selection.

5.1 Cosmological constraints

In the previous section we saw that the main difference throughout the models rests on the mass power spectrum at small scales, hence we use the 3D matter power spectrum inferred from Lyman-α data from BOSS and eBOSS collaboration [86][1]. We also use the Ly-α BAO from eBOSS DR14 [160], the Galaxy BAO from DR12 [161], 6dFGS [162] and

[1] As mentioned in [86], the process of inferring the MPS is model-dependent, therefore we will consider the constraints obtained here as an approximation.

SDSS DR7 [163], and the SNe Ia survey Pantheon [164] to improve the constraining power. On the other hand, we found that the parameter constraints describing the scalar-field remain the same when incorporating the Planck 18 dataset. For this reason and to reduce computational time we have only used the data already described. See appendix 5.4 for further details.

The sampling parameters in the analysis are the physical baryon density parameter $\omega_{b,0}$, the logarithmic power spectrum scalar amplitude $\log(10^{10}A_s)$, the scalar spectral index n_s, the Thomson scattering optical depth due to reionization τ_{reio}, the scalar field masses $m_{\phi i}$, the decay parameters $\lambda_{\phi i}$ and, instead of cold dark matter density parameter, we use the scalar field density parameters $\Omega_{\phi i,0}$. However we found, from previous analysis, that the posteriors of $\log(10^{10}A_s)$, n_s and τ_{reio} do not present a change respect to ΛCDM, therefore we will kept these parameters fixed. See appendix 5.4. The flat priors used for the remaining sampling parameters are as follow: $H_0 = [10, 100]$ for the Hubble constant in km s^{-1}Mpc^{-1}, $\omega_{b,0} = [0.005, 0.1]$ for the physical baryon density and $\Omega_{\phi 1,0} = \Omega_{\phi 2,0} = [0, 1]$ for the scalar field density parameters today. For the scalar field parameters, we choose the priors to be consistent with the numerical results we found with CLASS, and since these parameters can take values of powers of ten, we chose a logarithmic base to efficiently cover the entire parameter space and reduce the computational cost. This means that we have $\log_{10}(m_{\phi i}/\mathrm{eV}) = [-24, -17]$ for the scalar field masses, $\log_{10}(\lambda_{\phi i}) = [1, 6]$ for the parameter on the trigonometric cosine potential and $\lambda_{\phi i,\mathrm{aux}} = [-6, -1]$ for the hyperbolic cosine potential. For this case, we cannot use directly $\log_{10}(\lambda_{\phi i})$ because $\lambda_{\phi i} < 0$, so in order to cover the parameter space we use the auxiliary variable $\lambda_{\phi i,\mathrm{aux}}$ whose relation with $\lambda_{\phi i}$ is given by $\lambda_{\phi i} = -10^{-\lambda_{\phi i,\mathrm{aux}}}$. In what follows we will use quad$_i$ to refer to the potential $V_i(\phi_i) = (1/2)m^2_{\phi i}\phi^2_i$, cos$_i$ for $V_i(\phi_i) = m^2_{\phi i}f^2_i\left[1 + \cos(\phi_i/f_i)\right]$ and cosh$_i$ for $V_i(\phi_i) = m^2_{\phi i}f^2_i\left[\cosh(\phi_i/f_i) - 1\right]$. If there is no subscript, it means that it is the single field case.

First, in Figure 5.1, we show the constraints for the single field cases, that is, the 1D marginalized posterior distribution of the free parameters for each model corresponding to

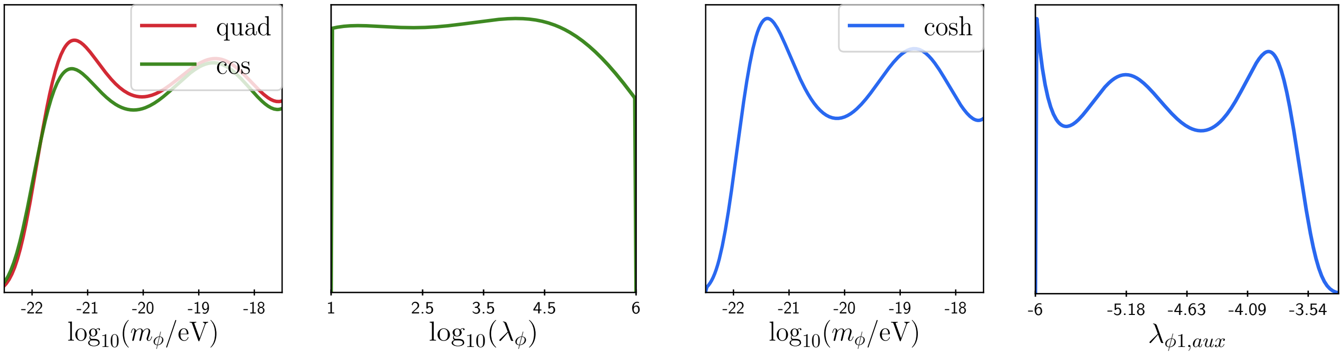


Figure 5.1: 1D posterior distribution for the single SFDM with potentials $V(\phi) = 1/2m_\phi^2\phi^2$ (red), $V(\phi) = m_\phi^2 f^2\,[1 + \cos(\phi/f)]$ (green) and $V(\phi) = m_\phi^2 f^2\,[\cosh(\phi/f) - 1]$ (blue). The constraints on the field mass are very similar for the three potentials, whereas the self-interaction parameter λ_ϕ of the trigonometric potential remains unconstrained by the data.

the potentials $V(\phi) = 1/2m_\phi^2\phi^2$, $V(\phi) = m_\phi^2 f^2\,[1 + \cos(\phi/f)]$, and $V(\phi) = m_\phi^2 f^2\,[\cosh(\phi/f) - 1]$. It can be noticed that the posteriors of the field masses are very similar for the three potentials, and that the observations considered can only put a lower bound on it, which is $\log(m_\phi/\mathrm{eV}) \gtrsim -21.9$ at 95% CL for the quadratic and hyperbolic cosine potentials and $\log(m_\phi/\mathrm{eV}) \gtrsim -21.8$ at 95% CL for the trigonometric cosine potential. In the case of the trigonometric potential, we see that its extra parameter λ_ϕ appears unconstrained and the posterior looks practically the same with the prior we considered above for this parameter, which is consistent with previous studies [96, 106, 87]. As for the hyperbolic cosine, we have already mentioned the close relationship between the field mass m_ϕ and the self-interaction parameter λ_ϕ, and then the constraints on the former are translated to $\log(-\lambda_\phi) \gtrsim 3.6$ at 95% CL. The excluded values of λ_ϕ correspond to the cases in which SFDM has a significant contribution as an early radiation component. An example of this can be seen in Figure 5.2.

For completeness, we also present in Figure 5.3 the 1D posterior distributions for the cosmological common parameters of the models: the density parameters (SFDM and Λ), and the Hubble constant H_0. The posteriors are very similar for the three models, which

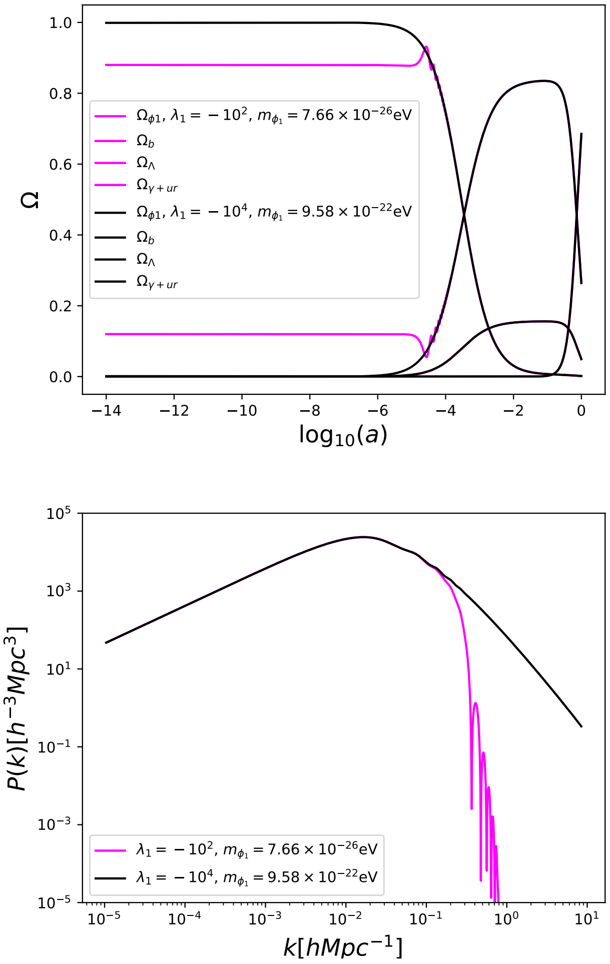


Figure 5.2: Background and MPS evolution for the SFDM with potential $V(\phi) = m_\phi^2 f^2 \left[\cosh(\phi/f) - 1\right]$. Magenta lines correspond to $\lambda_\phi = -10^2$ with $m_\phi = 7.66 \times 10^{-26}$ eV, which are ruled out from the constraints obtained. We can see the scalar field contribution to radiation. On the other hand, black lines, that correspond to $\lambda_\phi = -10^4$ with $m_\phi = 9.58 \times 10^{-22}$ eV, represent values within the confidence regions.

show that one recovers the results of the standard ΛCDM model.

In Figure 5.4 we show the posteriors [2] for the combinations $V(\phi_1) = 1/2 m_{\phi 1}^2 \phi_1^2$ with $V(\phi_2) = 1/2 m_{\phi 2}^2 \phi_2^2$, $V(\phi_1) = 1/2 m_{\phi 1}^2 \phi_1^2$ with $V(\phi_2) = m_{\phi 2}^2 f^2 \left[1 + \cos(\phi_2/f)\right]$ and $V(\phi_{1,2}) = m_{\phi 1,2}^2 f^2 \left[1 + \cos(\phi_{1,2}/f)\right]$. For the quadratic-quadratic and quadratic-cos combinations we

[2] The triangle plots of the MSFDM paremeters in Figure 5.1 and Figure 5.4 can be seen in appendix 5.4

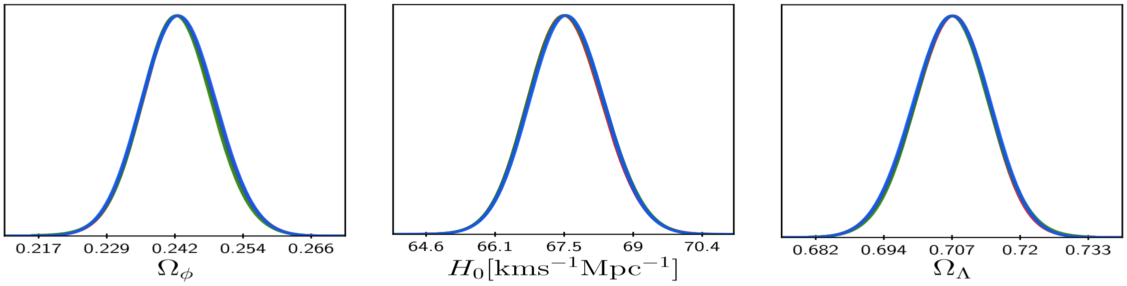


Figure 5.3: 1D posterior distribution for the cosmological common parameters of all the models studied in this work.

found a lower bound for the mass of both fields given by $\log_{10}(m_{\phi 1,2}/\mathrm{eV}) = -22.1$ at 95%. While, for the cos-cos combination, the lower bound for the mass of field one is $\log_{10}(m_{\phi 1}/\mathrm{eV}) = -22.1$ at 95% and for the second field $\log_{10}(m_{\phi 2}/\mathrm{eV}) = -22.2$ at 95%. We also found that the density parameter $\Omega_{\phi i}$ of one field depends on the $\Omega_{\phi j}$ of the other field. That is, if $\Omega_{\phi 1}$ dominates then $\Omega_{\phi 2}$ is small, and on the other way around because the total contribution remains constant. In [114] the authors propose two scalar fields (with the possibility of a third one) with masses $m_{\phi 1} \approx 10^{-22}$ eV and $m_{\phi 2} \approx 10^{-20}$ eV, and these values are within our confidence regions. The fact that the lighter field dominates over the massive field is also in agreement with our results. The triangle plots of the MSFDM parameters in Figure 5.1 and Figure 5.4 can be seen in appendix 5.4.

	quad	cosh	$\mathrm{quad}_1 + \mathrm{quad}_2$
$\ln \mathcal{B}_{i,\Lambda\mathrm{CDM}}$	1.59	0.85	2.13

Table 5.1: Bayes factor for SFDM and double-SFDM with different potentials using ΛCDM as reference.

As we have mentioned, the main objective is to present the model of Multiple Scalar Fields as Dark Matter. Although, for completeness, we compute the Bayes factor, $\ln \mathcal{B}_{i,j}$, by using the numerical package MCEvidence [165], and the ratio is done respect to CDM. The results are shown in Table 5.1. Following the Jeffreys guideline [11], if $\ln \mathcal{B}_{i,j} > 5$ we

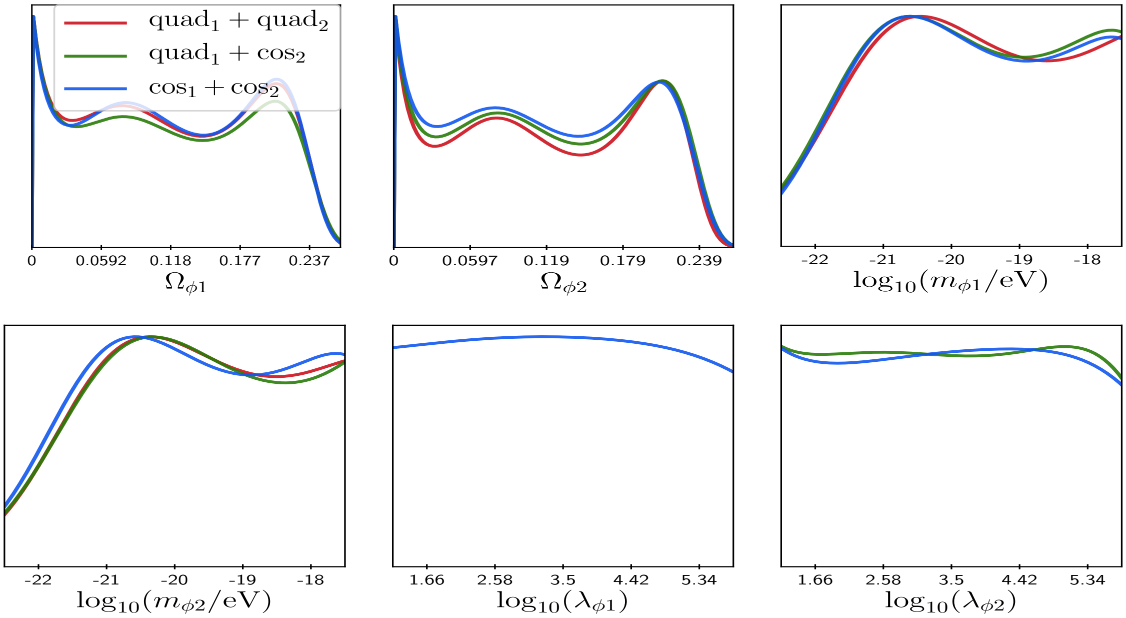


Figure 5.4: 1D posterior distribution for the two SFDM with the potentials $V(\phi_{1,2}) = 1/2m^2_{\phi1,2}\phi^2_{1,2}$ (red), $V(\phi_1) = 1/2m^2_{\phi1}\phi^2_1$ with $V(\phi_1) = m^2_{\phi1}f^2\left[1+\cos(\phi_1/f)\right]$ (green) and $V(\phi_{1,2}) = m^2_{\phi1,2}f^2\left[1+\cos(\phi_{1,2}/f)\right]$ (blue).

have a decisive strength against model i; if $5 > \ln\mathcal{B}_{i,j} > 2.5$ it means a strong strength; if $2.5 > \ln\mathcal{B}_{i,j} > 1$ we have a significant strength and if $\ln\mathcal{B}_{i,j} < 1$ the data prefers model i. In general, we should be careful taking the values of the Bayes factor since some of the parameters were unconstrained by the data [166]. On the other hand, the fact that the posteriors of the parameters of the scalar fields are not Gaussian prevents us from using other information criteria such as the AIC or the BIC; or else, the Bayesian complexity, since they need (or assume) the Gaussianity of the posterior. See, for example, [167] or [168]. On the other hand, we do not present the Bayes factor for the trigonometric potential cosine nor for the models with more than one field because the posteriors are not mono-modal and some parameters are not constrained which can cause the method used in MCEvidence to fail [165].

We found that the maximum of the likelihoods correspond to the same value $-\ln\mathcal{L}_{\mathrm{max}} = 523.03$. That is, we can find a suitable combination of parameters such that the model in

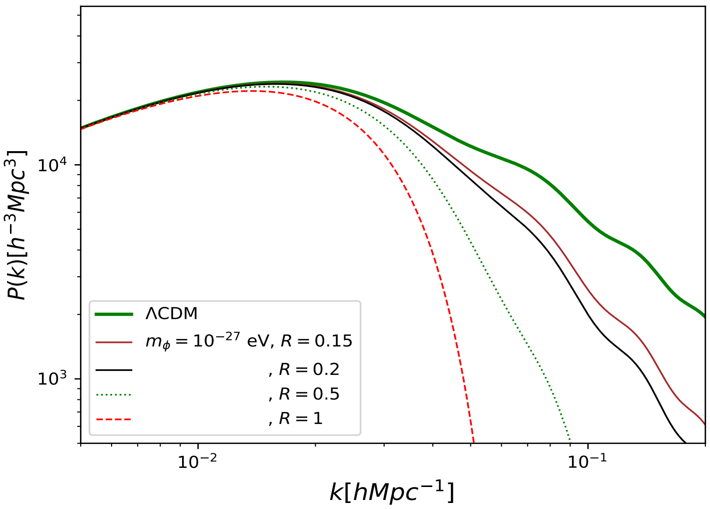


Figure 5.5: Evolution of the linear matter power spectrum at $z = 0$ for a CDM+SFDM model with $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ with mass value $m_\phi = 10^{-27}$ eV and $R = 0.15$, 0.2, 0.5 and 1. Colors and line types were chosen to match Figure 2 of [100] for an easy comparison.

place resembles the behaviour of a cold dark matter.

5.2 Comparison with axionCAMB

In addition to CLASS, there is a code that solves the cosmology for an ultra-light axion, this is a modification to the CAMB code [169, 170] called axionCAMB [100, 171]. For the case where only the scalar field is present, the formalism presented in this work and the results of axionCAMB [100] are in agreement as mentioned in [65]. Similarly, in [100], the authors also studied the combination of an ultra-light axion with CDM. In Figure 5.5, we have reproduced the right side of figure 2 of the same reference for $R > 0.1$ and $m_\phi = 10^{-27}$ eV. We found that the code presented here is in good agreement with the results of [100] except for small values of R, which translates to very small contributions from the scalar field ($\Omega_\phi \sim< 0.02$).

5.3 Extensions to MSFDM model: neutrino masses and curvature.

It is well known that neutrinos have effects on the evolution of the Universe [172, 173], in particular on the MPS [174] where the MSFDM model has differences respect to ΛCDM. Therefore, in this section we vary the neutrino masses to see the effects on our model. We assume the neutrino model already presented in section 3.2. In Figure 5.6 we show the evolution of the linear matter power spectrum for the SFDM model (one field) with the potential $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ with mass values $m_\phi = 10^{-24}$, 10^{-22} and 10^{-20} eV each combined with neutrino masses $m_\nu = 0.2$, 0.4, 0.6, 0.8 and 1 eV; bigger values are already discarded [175, 176, 177, 178]. We found no significant differences respect to the neutrino base model. However, there could be degeneration with the MSFDM. For example, in Figure 5.7 we compare the evolution of the MPS for the double field model, with potentials $V(\phi_{1,2}) = \frac{1}{2}m_{\phi 1,2}^2\phi_{1,2}^2$, and the single field case with different neutrino masses showed in Figure 5.6. For the double field model, we fix the mass value of field one to $m_{\phi 1} = 10^{-22}$ eV while the second mass takes $m_{\phi 2} = 10^{-24}$ eV and 10^{-20} eV, with $R = 0.2$ and $R = 0.8$. We can see that the MPS of both models are very similar.

In the same way, we study the case where we add curvature to the scalar field model and find the same results as for ΛCDM. This can be seen in the left panel of Figure 5.8 where we show the 1D posterior for Ω_k.

5.4 MSFDM: Planck 18 results.

In this appendix we show the triangle plots for the different potential combinations analyzed in section 3.4. In Figure 5.9 we show the posteriors of the physical baryon density parameter $\omega_{b,0}$, the logarithmic power spectrum scalar amplitude $\log(10^{10}A_s)$, the scalar spectral index n_s, the Thomson scattering optical depth due to reionization τ_{reio}, the Hubble constant H_0 in km s^{-1}Mpc^{-1}, the scalar field mass $\log_{10}(m_{\phi i}/\text{eV})$, the Dark Energy density parameter

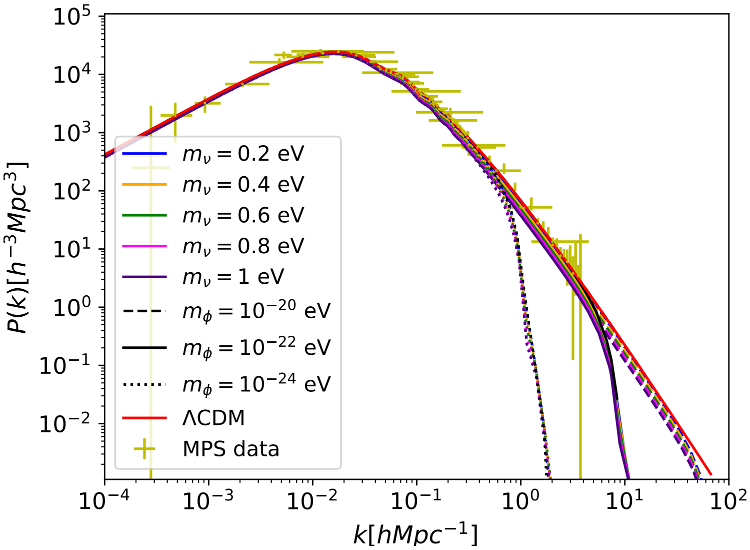


Figure 5.6: Evolution of the linear matter power spectrum at $z = 0$ for $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ with neutrino masses $m_\nu = 0.2$, 0.4, 0.6, 0.8 and 1 eV. Coloured dotted lines represent the combinations with $m_\phi = 10^{-24}$ eV, coloured solid lines with $m_\phi = 10^{-22}$ eV and coloured dashed lines with $m_\phi = 10^{-20}$ eV. For example, the blue dashed line represents a model with $m_\nu = 0.2$ eV and $m_\phi = 10^{-20}$ eV. The black lines represent the case for the SFDM with masses $m_\phi = 10^{-24}$ eV (dotted), $m_\phi = 10^{-22}$ eV (solid) and $m_\phi = 10^{-20}$ eV (dashed) using the base model for neutrinos used in [10]. Solid red line represents the MPS for ΛCDM. The data points for the MPS are the same used in section 3.4 and were obtained from [86].

Ω_Λ and the Dark Matter density parameter Ω_{DM}. The last one refers to Cold Dark Matter for ΛCDM and to the scalar field for SFDM using Planck 18 data [10, 179], the 3D matter power spectrum inferred from Lyman-α data from BOSS and eBOSS collaboration [86], the Ly-α BAO from eBOSS DR14 [160], the Galaxy BAO from DR12 [161], 6dFGS [162] and SDSS DR7 [163], and the SNe Ia survey Pantheon [164]. As we can see, there are no differences between the constrictions of the ΛCDM and SFDM parameters and they agree with the values reported in [10]. Thus, in section 5.1 we decided to vary only the parameters corresponding to the scalar fields, leaving the rest of the parameters fixed. Using the fact that the SFDM model provides similar constraints for the basic parameters using the Planck data and given that we found similar restrictions for the scalar field mass

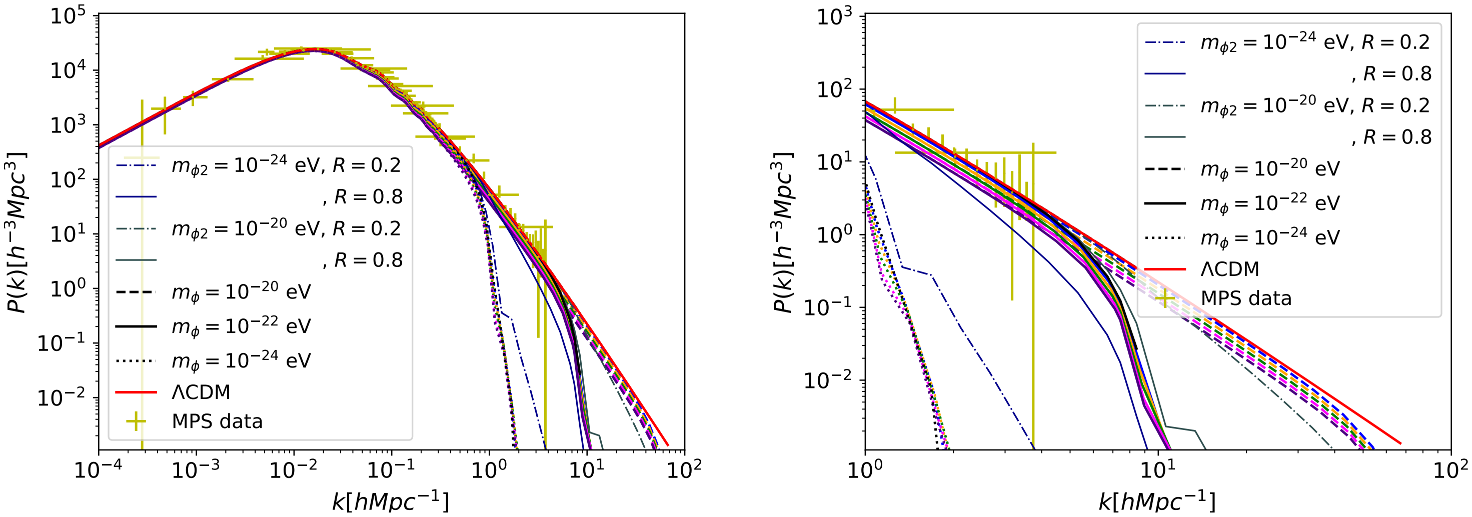


Figure 5.7: Evolution of the linear matter power spectrum at $z = 0$ for a single field with potential $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ with neutrino masses $m_\nu = 0.2$ (blue), 0.4 (orange), 0.6 (green), 0.8 (magenta) and 1 eV (indigo) compared with the double field model with $V(\phi_{1,2}) = \frac{1}{2}m_{\phi 1,2}^2\phi_{1,2}^2$ using $m_{\phi 1} = 10^{-22}$ eV and different values of $m_{\phi 2}$ and R. For the single case varying neutrino mass, coloured dotted lines represent the combinations with $m_\phi = 10^{-24}$ eV, coloured solid lines with $m_\phi = 10^{-22}$ eV and coloured dashed lines with $m_\phi = 10^{-20}$ eV. The dark blue and gray lines represent the MSFDM evolution and solid red line represents the MPS for ΛCDM. The data points for the MPS are the same used in section 3.4 and were obtained from [86]. On the right side, a small-scale zoom is shown where you can see the consequence of changing the values of the neutrino masses and the scalar fields.

with and without Planck 18 data, we decided to use only the MPS from Ly-α, BAO and SNe Ia data. This change in the data set used also gives us a reduction in computational time.

On the other hand, in Figures 5.10, 5.11, 5.12, 5.13, 5.14 and 5.15 we also show the 1D and 2D posteriors of the combinations studied in section 5.1 when we consider each data set separately. In all cases we find that we need to combine the data sets to improve the constraints on the parameters of the scalar fields. Note that by using only the BAO data, the restriction for H_0 gives high values compared to Planck 18, so it is necessary to use more data sets in order to constrain this parameter. See [160].

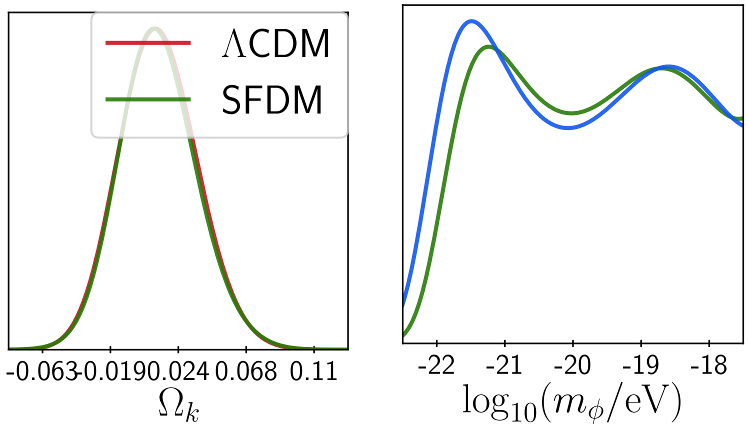


Figure 5.8: In the left panel we show the constraints on the curvature parameter for ΛCDM and for the SFDM model. In the right panel, we show the mass value constriction for the SFDM model with potential $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ using the data from Planck 18, MPS from Ly-α BAO and SNe Ia (blue) and, MPS from Ly-α BAO and SNe Ia (green).

Finally, in Figures 5.16, 5.17, 5.17, 5.19, 5.20 and 5.21 we compare the constraints obtained using only the Planck 18 data with the Planck 18 plus MPS from Ly-α BAO and SNe Ia data. In general we see that using more data constrains the cosmological parameters better. And that, in particular, using only Planck does not give us information about the parameters of the scalar fields. This agrees with the results obtained in section 3.4 where the CMB spectrum is the same for the different models.

5.5 A general parametrisation

In [180, 181] the authors proposed a general parametrisation, called α, β, γ-parametrisation, to describe the transfer function of non-CDM (nCDM) models, in particular, the fuzzy dark matter model which in our case corresponds to a single SFDM with the quadratic potential. In their studies they found that models allowed correspond those that meet $m_\phi \gtrsim 10^{-22}$ eV. In a first test, we found this parametrisation could also describe the combination of two fields with different combinations of potentials that with the possible exception of the cases in which the characteristic bump of the trigonometric cosine-type potential occurs. In Figure 5.22, we show some examples of the transfer function for the combinations SFDM+CDM, $quad_1 + quad_2$, $quad_1 + cos_2$ and $quad_1 + cosh_2$ (solid lines) and the transfer function with the α, β, γ-parametrisation (dashed lines). Although the values of the parameters α, β and γ were set by hand, we can see they resemble the numerical results we obtained with the exception of the cosine potential, where the characteristic bump cannot be described by this parametrisation. However, further studies similar to the one done in [182] are necessary to determine the relationship between the properties of the scalar fields and the α, β and γ parameters and to obtain the respective constraints.

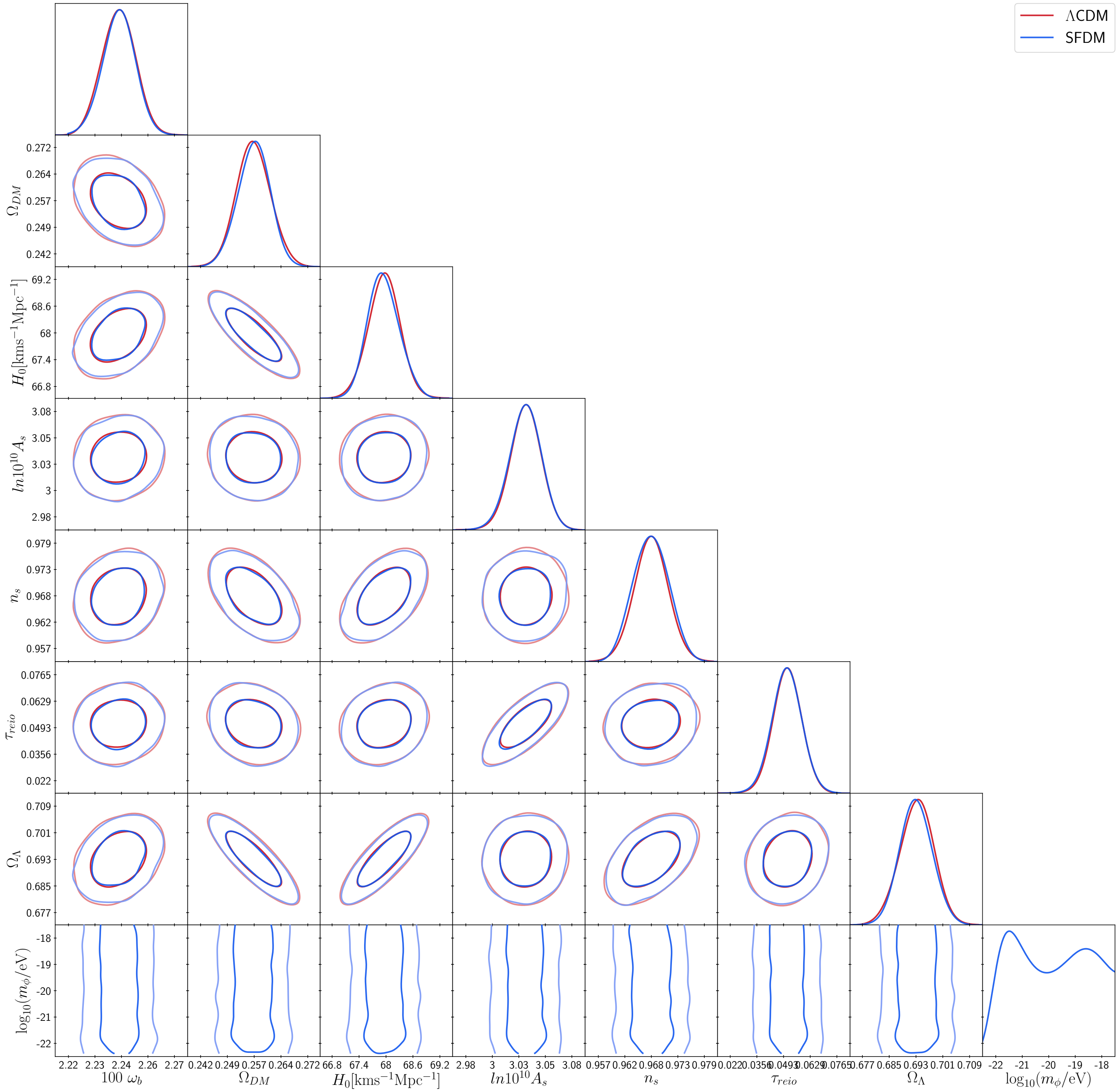


Figure 5.9: Triangle plots for the SFDM with potential $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ compared with ΛCDM as reference using the data from Planck 18, MPS from Ly-α BAO and SNe Ia. See text for details.

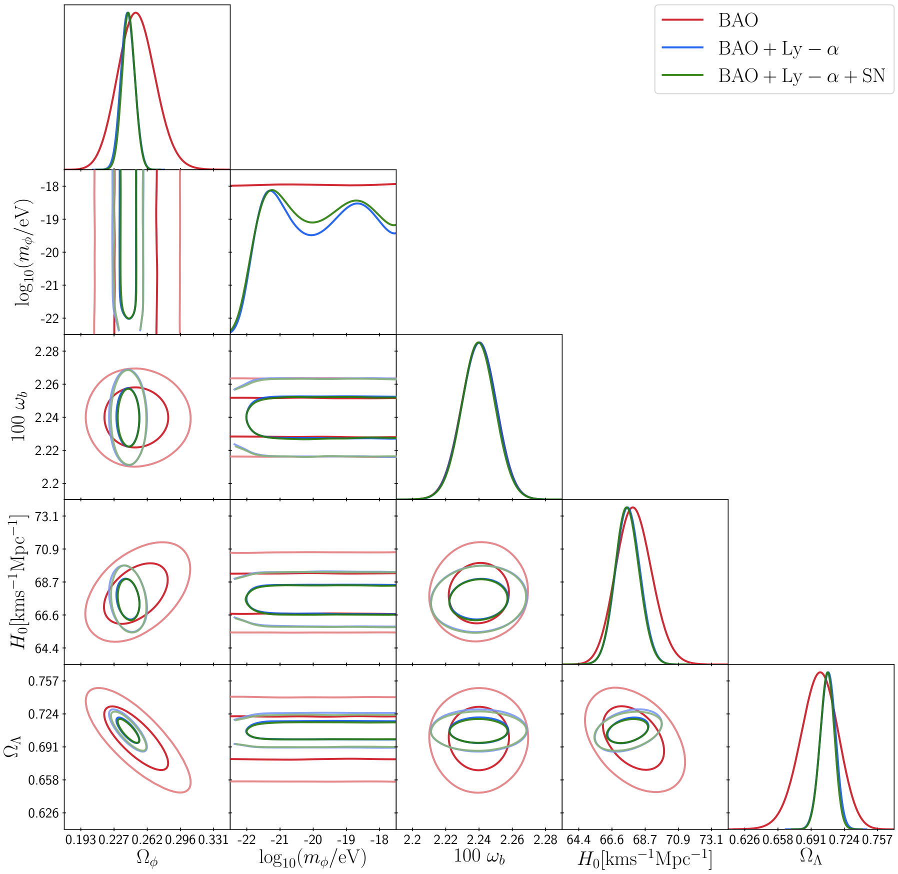


Figure 5.10: 1D and 2D posterior distribution for the potential $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$. We found constrictions for m_ϕ and Ω_ϕ. We see that it is necessary to use different data sets to constrain the parameters of the scalar field. Here and in the following figures the Ly-α label refers to the MPS inferred from these data. See the text for details.

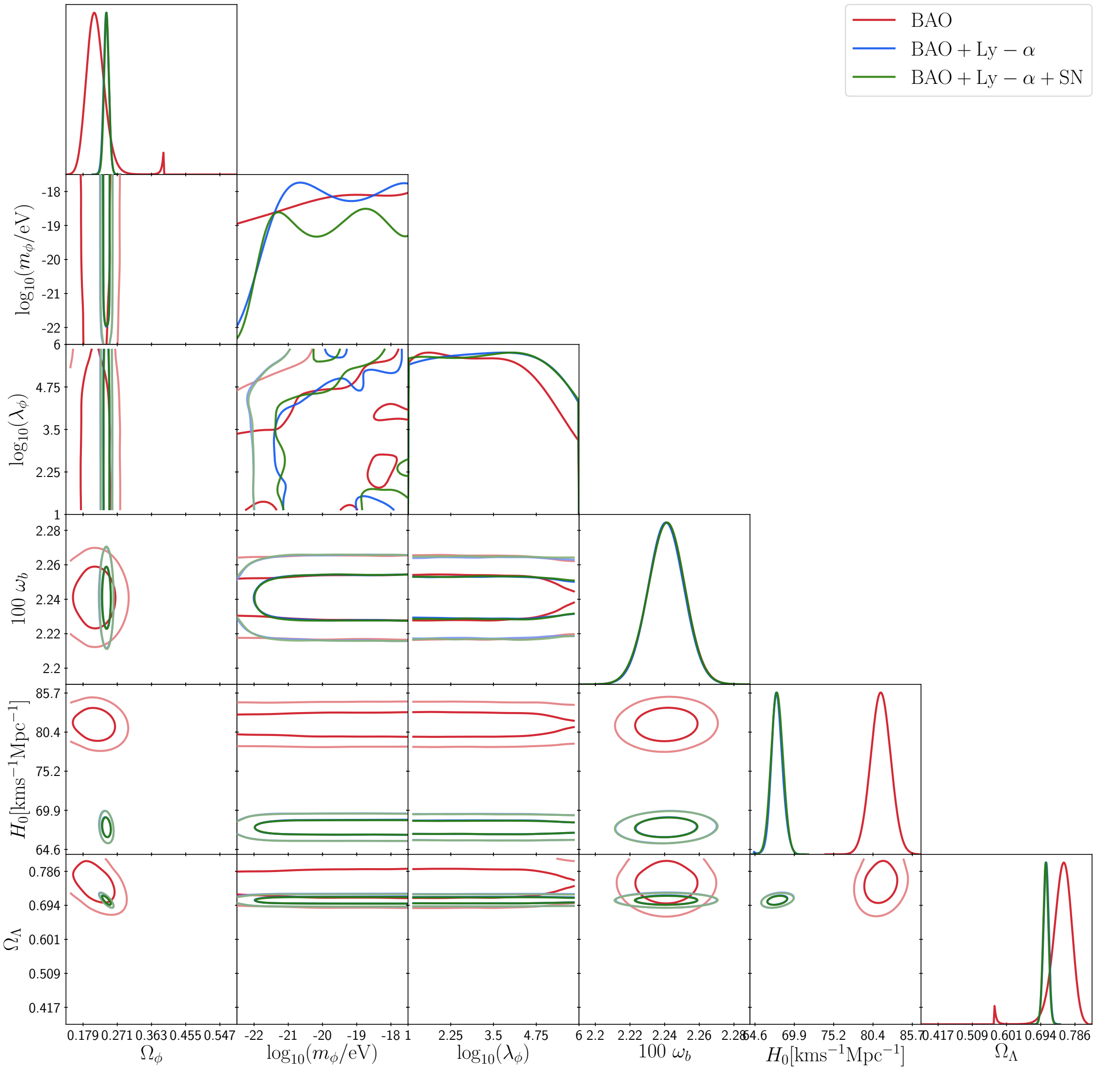


Figure 5.11: 1D and 2D posterior distribution for the potential $V(\phi) = m_\phi^2 f^2 \left[1 + \cos(\phi/f)\right]$. We found constrictions only for m_ϕ and Ω_ϕ. It is necessary to use different data sets to constrain the parameters of the scalar field.

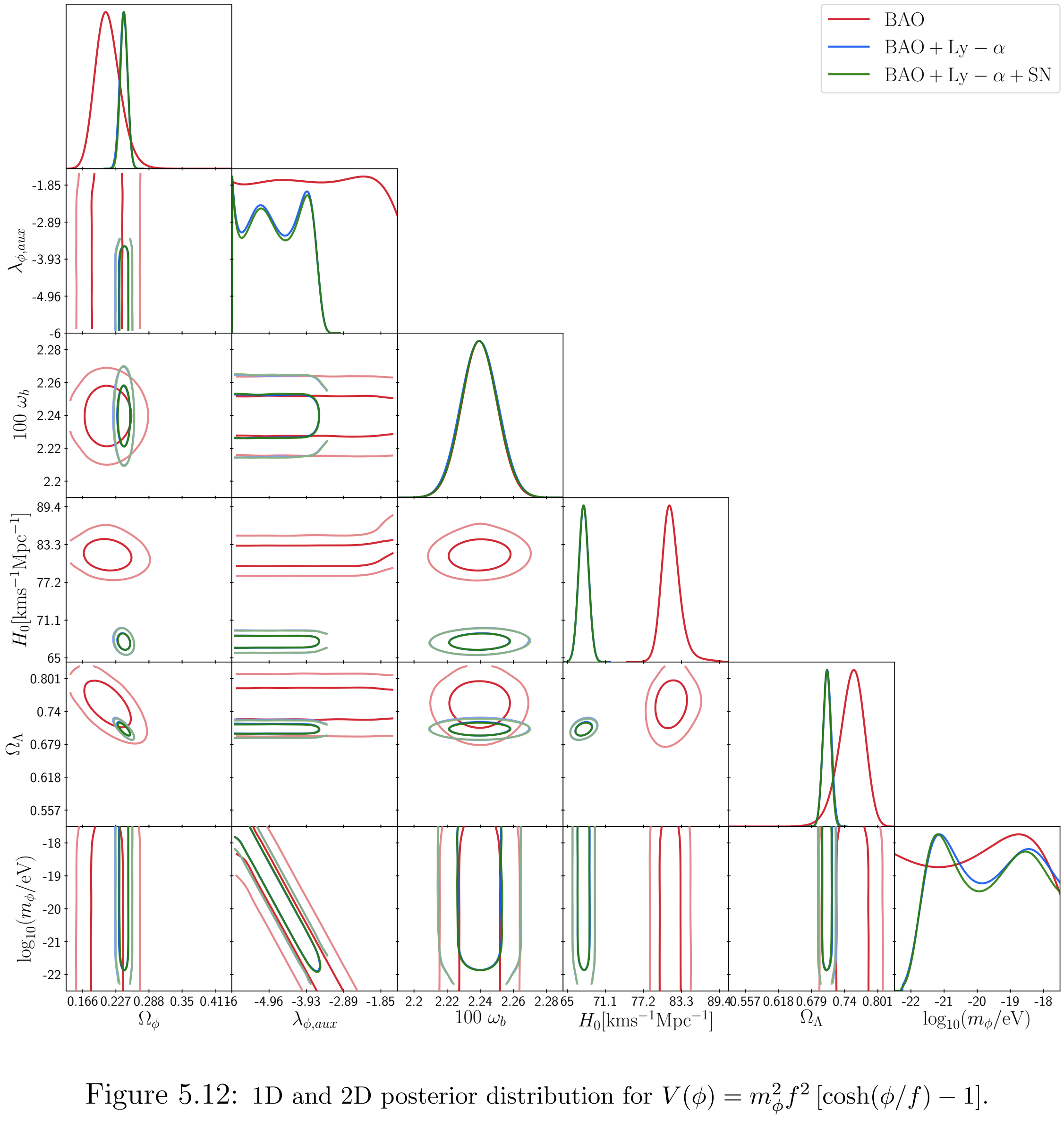


Figure 5.12: 1D and 2D posterior distribution for $V(\phi) = m_\phi^2 f^2\,[\cosh(\phi/f) - 1]$.

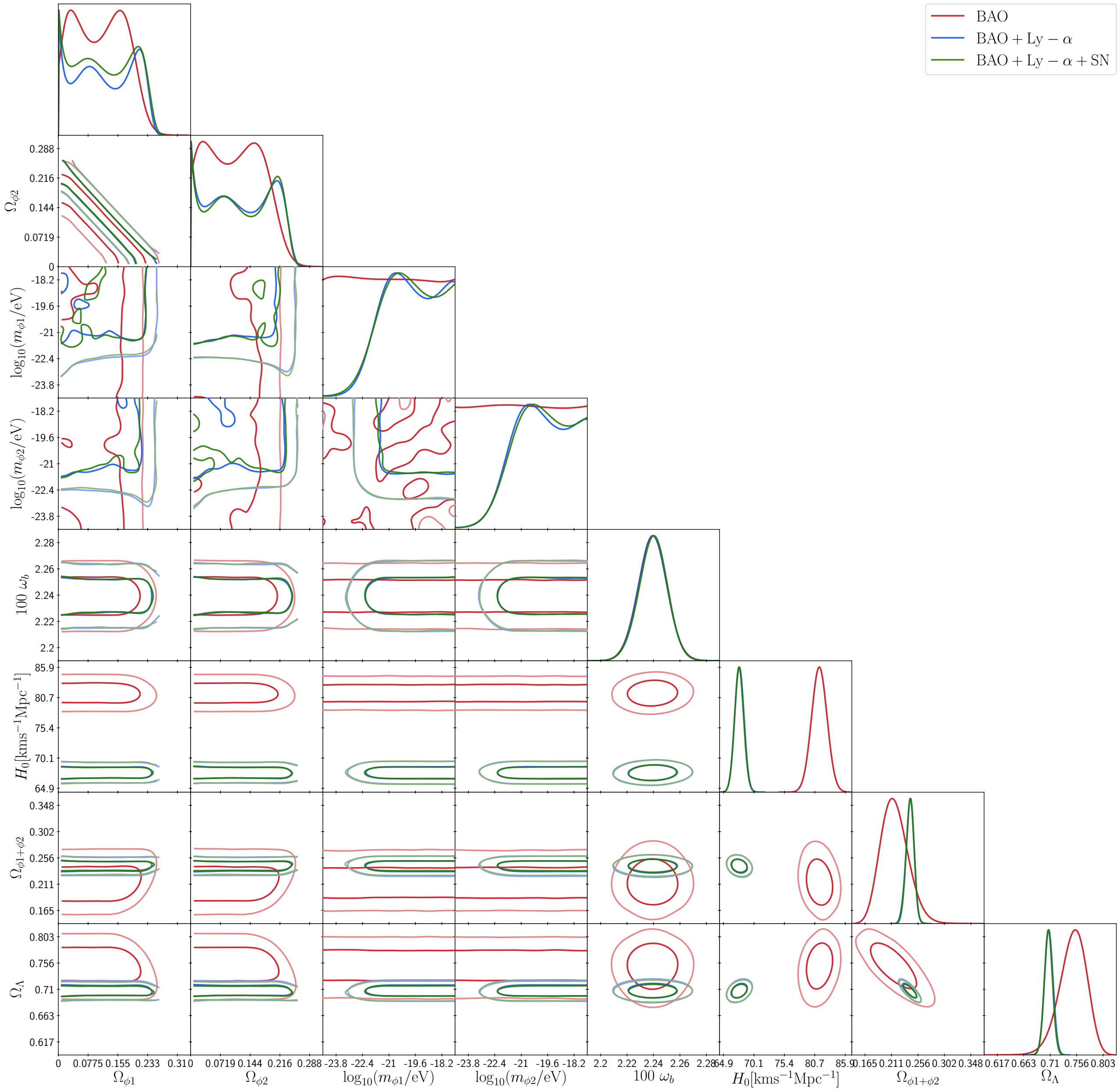


Figure 5.13: 1D and 2D posterior distribution of the fields considering potential $V(\phi_1) = 1/2m_{\phi 1}^2\phi_1^2$ for the first field with $V(\phi_2) = 1/2m_{\phi 2}^2\phi_2^2$.

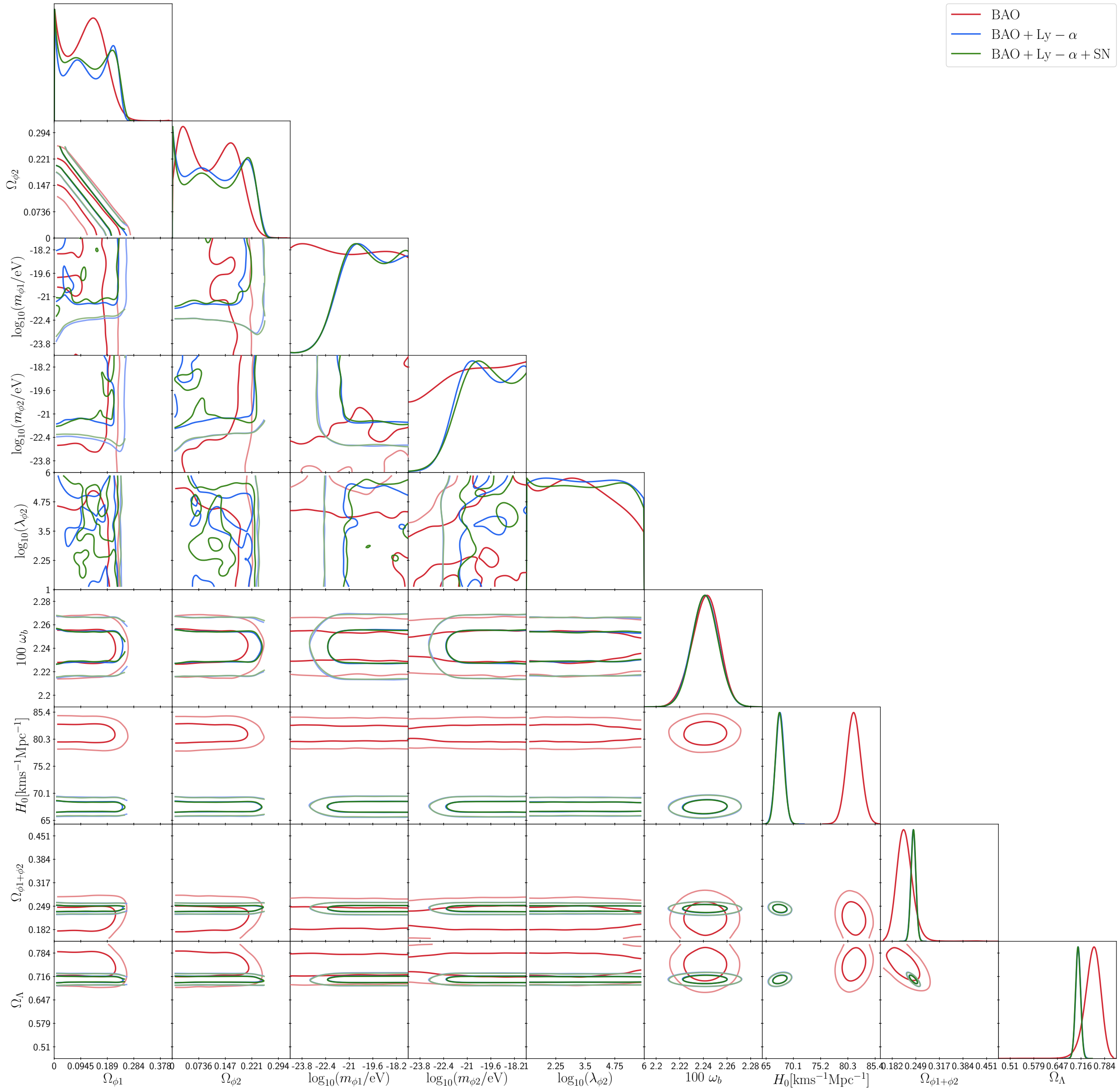


Figure 5.14: 1D and 2D posterior distribution of the fields considering potential $V(\phi_1) = 1/2m_{\phi 1}^2\phi_1^2$ for the first field with $V(\phi_2) = m_{\phi 2}^2 f^2\,[1+\cos(\phi_2/f)]$.

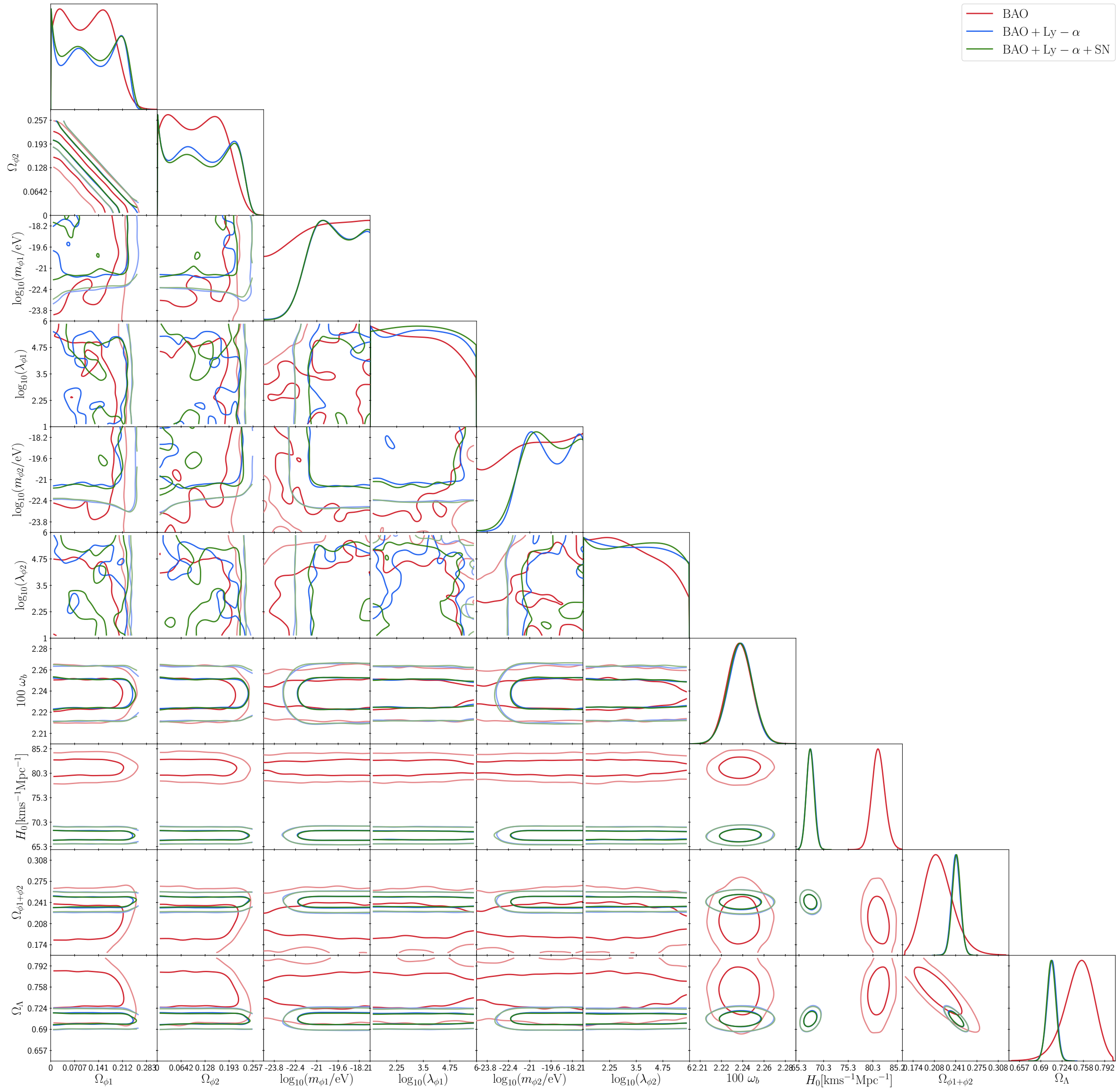


Figure 5.15: 1D and 2D posterior distribution of the fields when both have the potential $V(\phi) = m_\phi^2 f^2\,[1 + \cos(\phi/f)]$.

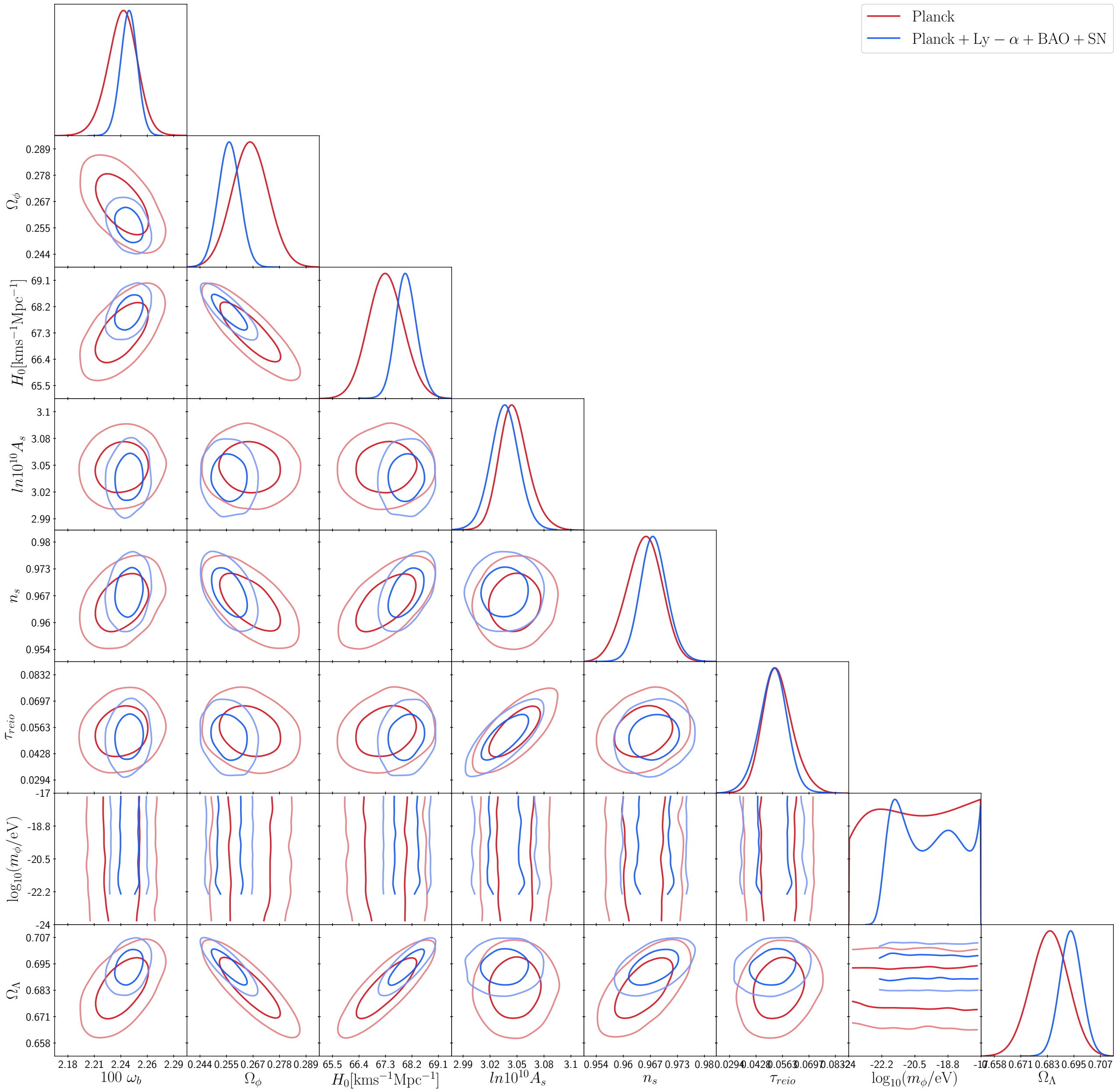


Figure 5.16: 1D and 2D posterior distribution of cosmological and scalar field parameters for a single field with $V(\phi) = \frac{1}{2}m_\phi^2\phi^2$ using Planck 18 data and Planck 18 plus MPS from Ly-α BAO and SNe Ia data.

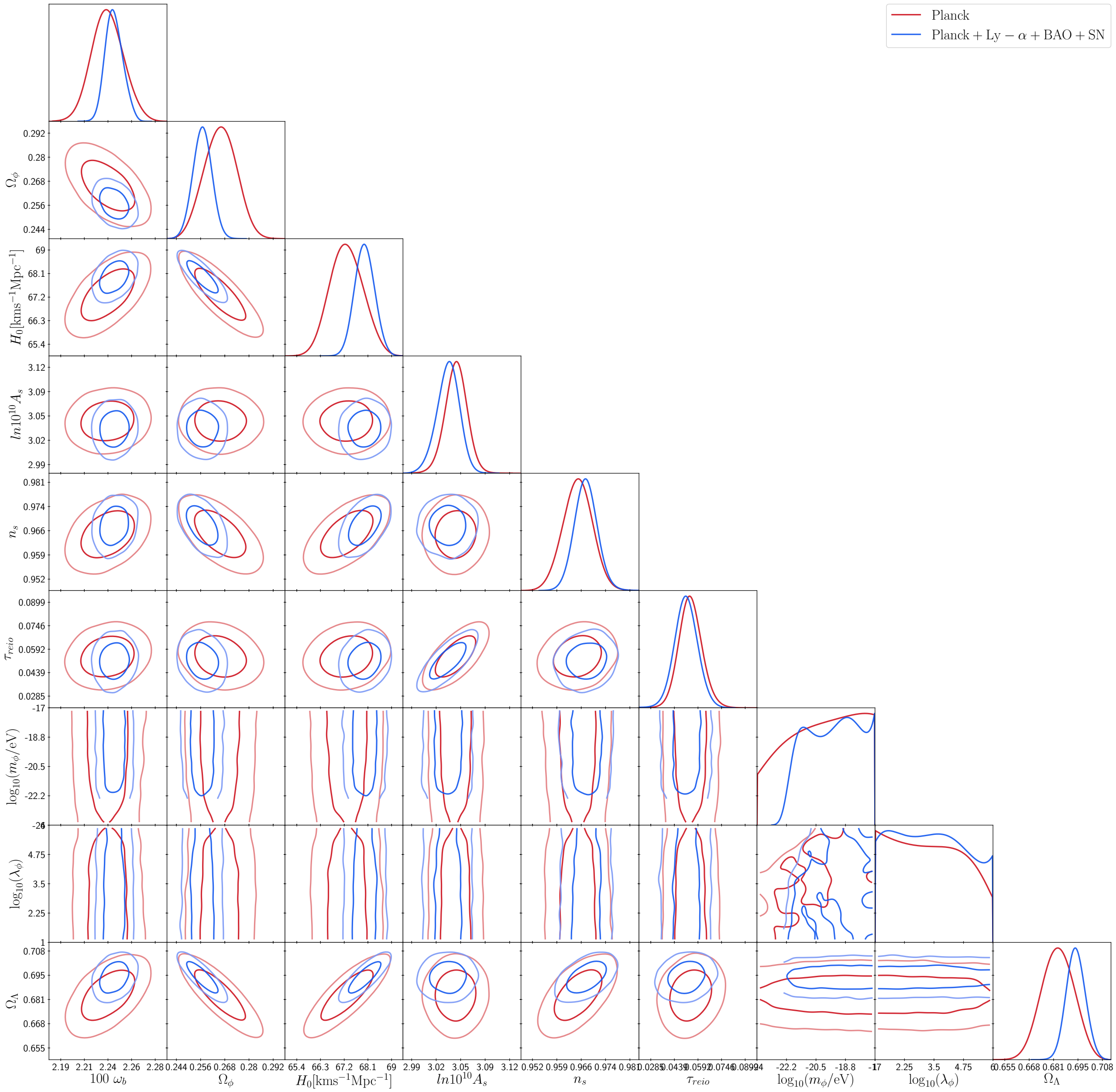


Figure 5.17: 1D and 2D posterior distribution of cosmological and scalar field parameters for a single field with $V(\phi) = m_\phi^2 f^2 \left[1 + \cos(\phi/f)\right]$ using Planck 18 data and Planck 18 plus MPS from Ly-α BAO and SNe Ia data.

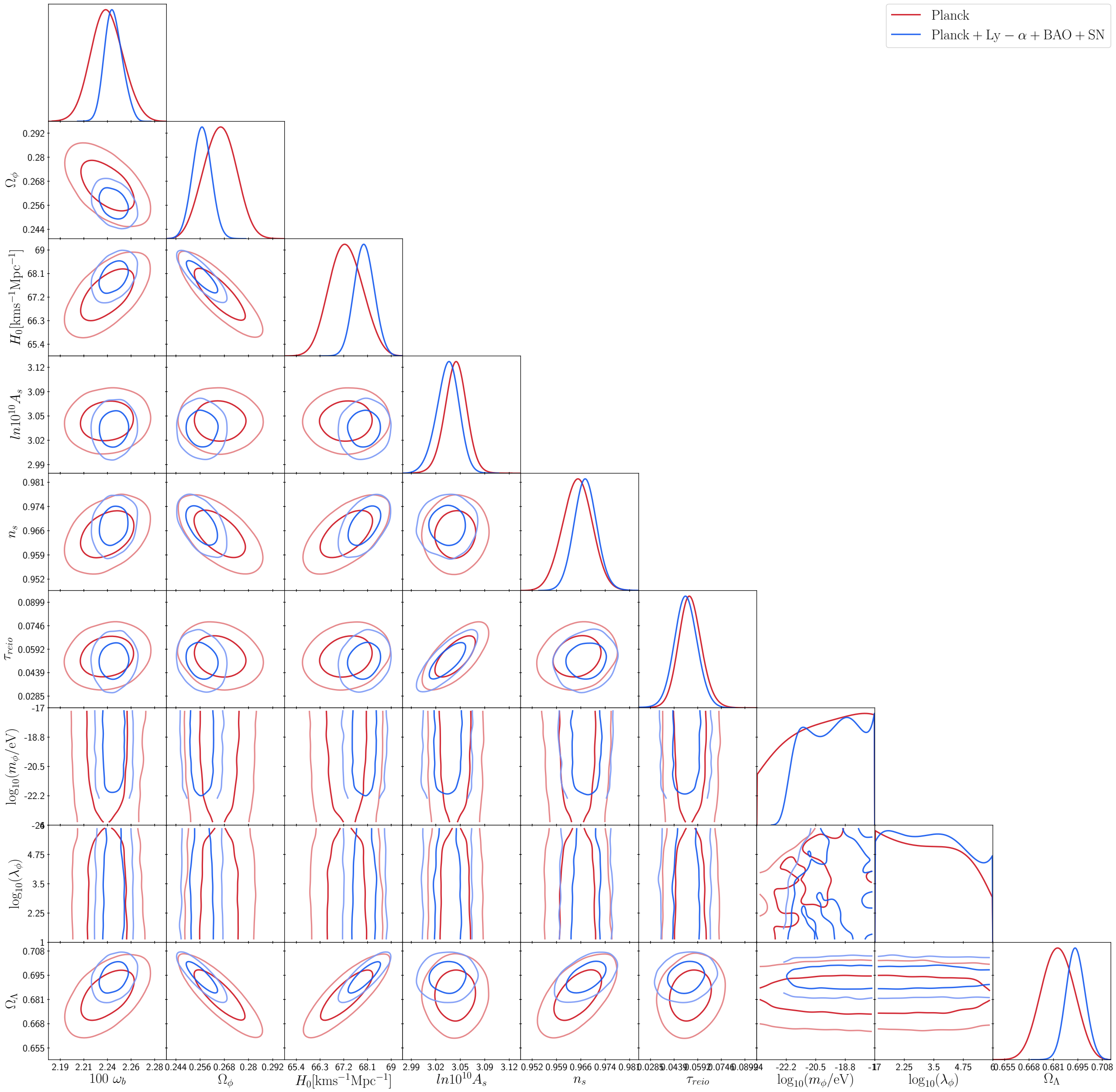


Figure 5.18: 1D and 2D posterior distribution of cosmological and scalar field parameters for a single field with $V(\phi) = m_\phi^2 f^2\left[\cosh(\phi/f) - 1\right]$ using Planck 18 data and Planck 18 plus MPS from Ly-α BAO and SNe Ia data.

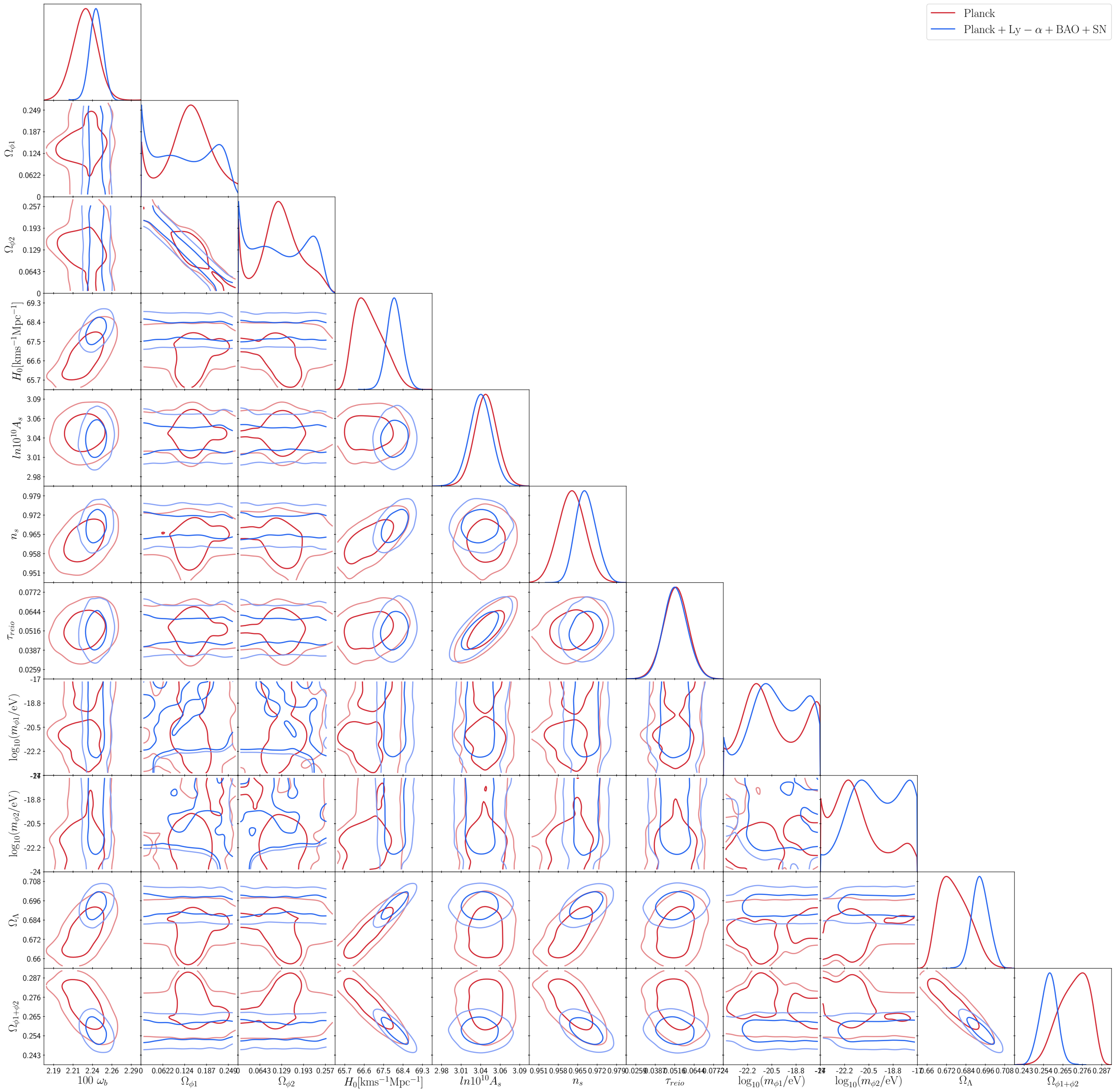


Figure 5.19: 1D and 2D posterior distribution of cosmological and scalar field parameters for double field with considering potential $V(\phi_1) = 1/2m_{\phi 1}^2\phi_1^2$ for the first field with $V(\phi_2) = 1/2m_{\phi 2}^2\phi_2^2$ using Planck 18 data and Planck 18 plus MPS from Ly-α BAO and SNe Ia data.

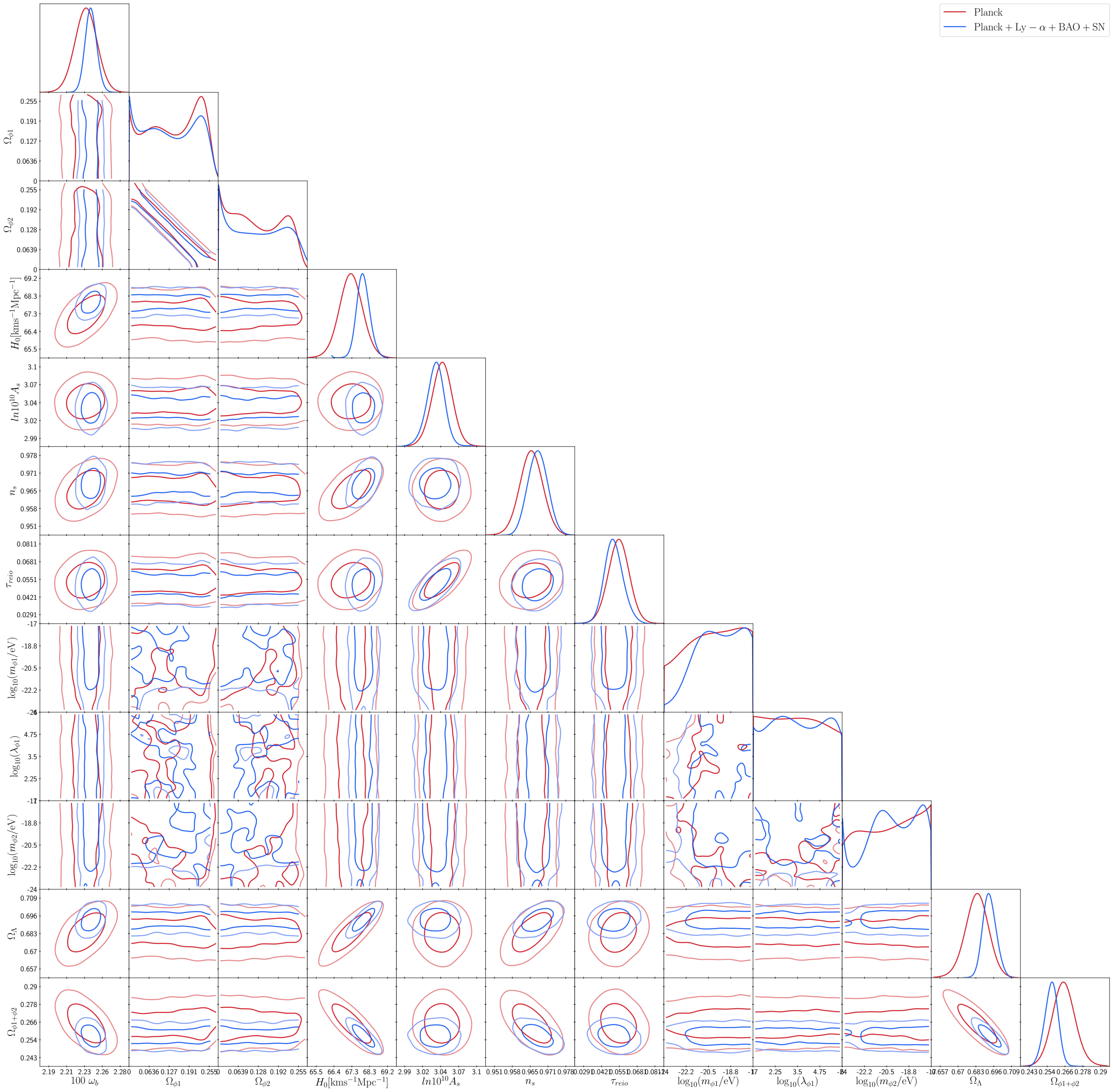


Figure 5.20: 1D and 2D posterior distribution of cosmological and scalar field parameters for double field with potential $V(\phi_1) = 1/2m_{\phi 1}^2\phi_1^2$ for the first field and $V(\phi_2) = m_{\phi 2}^2 f^2\left[1 + \cos(\phi_2/f)\right]$ using Planck 18 data and Planck 18 plus MPS from Ly-α BAO and SNe Ia data.

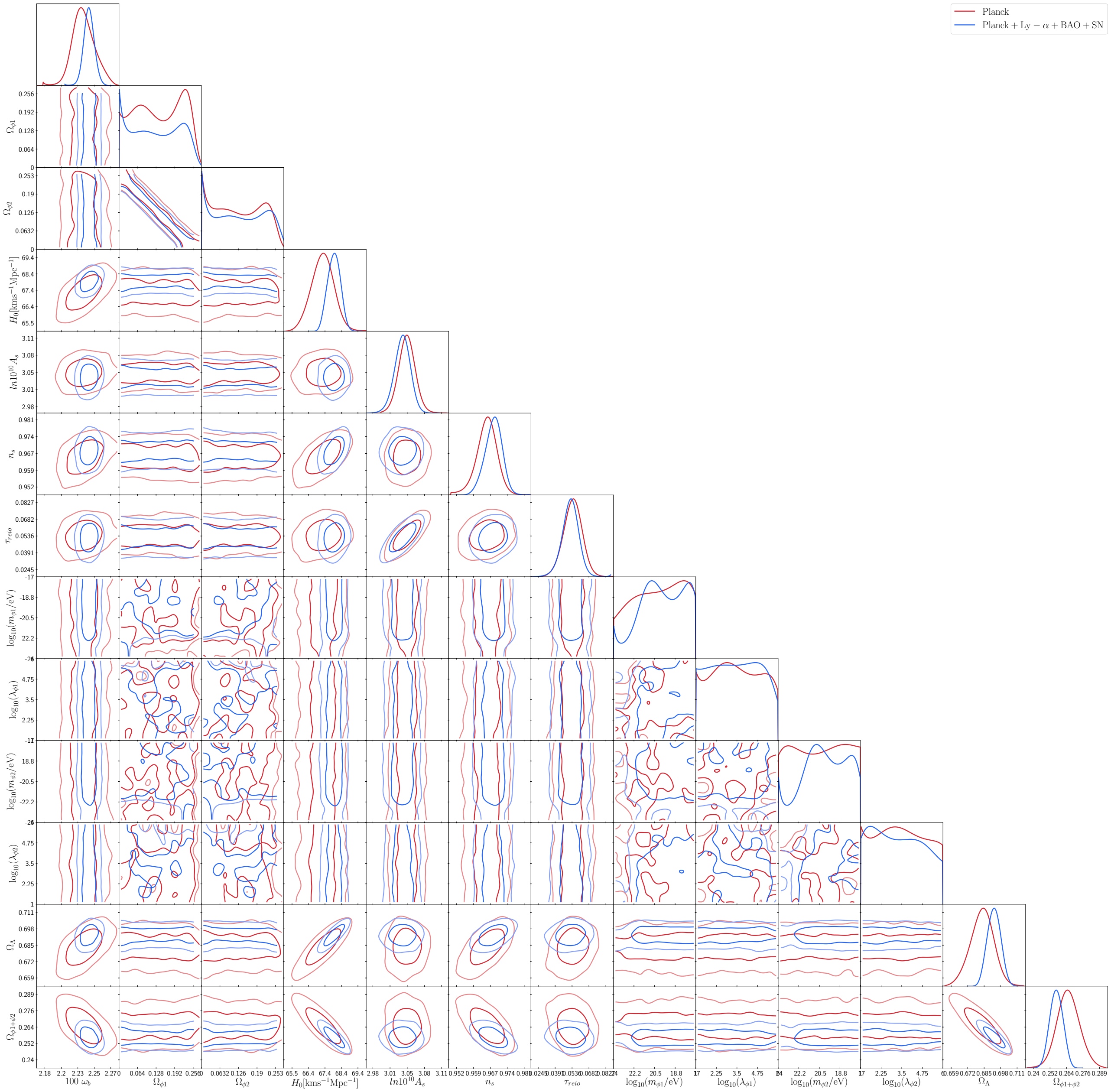


Figure 5.21: 1D and 2D posterior distribution of cosmological and scalar field parameters for double field when both have the potential $V(\phi) = m_\phi^2 f^2 \left[1 + \cos(\phi/f)\right]$ using Planck 18 data and Planck 18 plus MPS from Ly-α BAO and SNe Ia data.

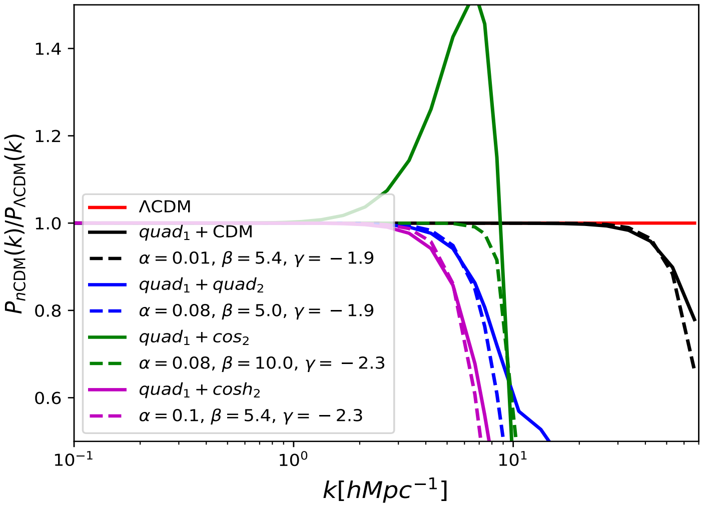


Figure 5.22: Transfer functions for the combinations SFDM+CDM with $m_\phi = 10^{-20}$ eV and $R = 0.2$ (black solid line); $quad_1 + quad_2$ with $m_{\phi_1} = 10^{-22}$ eV, $m_{\phi 2} = 10^{-20}$ eV and $R = 0.2$ (blue solid line); $quad_1 + cos_2$ with $m_{\phi_1} = 10^{-22}$ eV, $m_{\phi 2} = 10^{-22}$ eV, $\lambda_{\phi 2} = 10^5$ and $R = 0.2$ (green solid line); and $quad_1 + cosh_2$ with $m_{\phi_1} = 10^{-22}$ eV, $m_{\phi 2} = 0.3 \times 10^{-18}$ eV, $\lambda_{\phi 2} = -4 \times 10^5$ and $R = 0.5$. The dashed lines correspond to the transfer function using the α, β, γ-parametrisation for different combinations of α, β and γ.

Chapter 6

Compton Mass Dark Energy model (CMaDE)

In this chapter we present a work in progress in which we apply both the numerical and mathematical tools learned in the previous chapters to an alternative model of dark energy to explain the accelerated expansion of the Universe.

One of the beliefs is that the explanation for the acceleration of the universe could come from quantum mechanics, that is, from a theory of quantum gravity. In the reference [183] the authors propose an effective way to introduce the quantum character of the graviton, using analogies with other fields and interactions. They show that with this proposal the system behaves very similar to the ΛCDM case. In this chapter we will show that the predictions of the ΛCDM and CMaDE models are indistinguishable, at least at cosmological scales, since the CMB and MPS profiles are very similar, but the CMaDE model using an explanation of quantum mechanics without dark energy. In what follows we remain the main ideas of [183], but then we use the functional form of the dark energy in effective way.

The main arguments is that in the case of a massless particle, such as the gravitational interaction mediator or graviton, the energy due to its momentum $E = pc$, is not contained in the Einstein equations. In the Einstein's equations it is implicit that the mass of the mediator of the gravitational interaction is zero. On the other side, the energy of the graviton due to its moment comes from the quantum mechanical character of the graviton. But everything in nature gravitates. The claim of [183] is that this energy also gravitates and must be counted as extra energy.

The hypotheses are: if Gravitation is a quantum mechanical interaction its mediator has a Compton effective mass and its corresponding wavelength λ is limited by the size of the observable universe. Using these arguments, they found that the cosmological constant is given by [1]

$$\Lambda = \frac{2\pi^2}{\lambda^2}. \tag{6.1}$$

We will use Λ as indicated in (6.1) as an effective result, where Λ varies very slowly.

For an observer today, the gravitational interaction travels a distance R_H during its life, the wavelength will be $\lambda = (c/H_0)R_H$ long, where R_H is the unitless length given by

$$R_H = H_0 \int_0^t \frac{dt'}{a} = \int_{-\infty}^{N} \frac{H_0}{H} e^{-N'} dN'. \tag{6.2}$$

In order to obtain the Friedmann equation for our model, we consider that

$$H^2 + \frac{k}{a^2} = \frac{\kappa^2}{3}\left(\rho_m + \rho_r + \rho_\Lambda\right). \tag{6.3}$$

Using equations (6.1) and (6.2) in the derivative of (6.3), we get that [183]

$$\begin{aligned} \frac{HH'}{H_0^2} + \Omega_{0k}e^{-2N} + \frac{3}{2}\Omega_{0m}e^{-3N} + 2\Omega_{0r}e^{-4N} & - \\ \sqrt{\frac{3}{2}}\frac{H_0 e^{-N}}{H\pi}\left(\frac{H^2}{H_0^2} - \Omega_{0k}e^{-2N} - \Omega_{0m}e^{-3N} - \Omega_{0r}e^{-4N}\right)^{\frac{3}{2}} & = \; 0. \end{aligned} \tag{6.4}$$

From the above we have that the CMaDE Friedmann equation is given by (6.4) and

$$\Omega_\Lambda = \frac{2}{3}\frac{\pi^2}{R_H^2}. \tag{6.5}$$

[1] Be careful not to confuse the λ presented here, as a wavelength, with the λ of the scalar field models where it is related to the potentials decay constants.

Here it is important to note that given (6.1) with (6.2) for the function Λ implies that the CMaDE model only has curvature as a free constant to fit all observations. If we integrate (6.4) and (6.2) we find that $R_H \sim 3$. Also note that, because Λ is not a constant, the Bianchi identities have an extra term

$$\dot{\Lambda} = H\frac{d\Lambda}{dN} = -4\pi^2\left(\frac{H_0}{c}\right)^2 H_0\frac{e^{-N}}{R_H^3}. \tag{6.6}$$

We obtain that $\dot{\Lambda} = -4.48 \times 10^{-16}h_0^3/R_H^3/\text{Mpc}^2/\text{yr}$; its value today is $\dot{\Lambda} \sim -5 \times 10^{-17}h_0^3/\text{Mpc}^2/\text{yr}$, which is really very small, being $H_0 = 100h_0\text{Km/sec/Mpc}$. Note that just after inflation we can put that $\dot{\Lambda} = -1.5\times10^{-72}h_0^3e^{-N}/R_H^3/\text{cm}^2/\text{sec}$, which depends on the value of R_H. However, the redshift for inflation is $z \sim 10^{26}$, this means that $N \sim -60$. So, before inflation the wavelength is small, the exponential factor is big and the Bianchi identities have an extra term given by (6.6).

When inflation ends, the wavelength grows up about e^{60} times thus $R_H \sim \lambda_0 e^{60}$ grows very fast and (6.6) becomes very small. This means that the Bianchi identities hold up very well, because $\dot{\Lambda} \sim 0$, i.e., after inflation Λ is almost constant. So the equation (6.1) can be viewed as a very slowly varying cosmological constant.

Here, we will see equation (6.1) with the integral (6.2) as an effective proposal and check if them can explain the observable universe, leaving for future work the possible quantum gravity explanation of the equations (6.1) and (6.2). Note that this λ is similar to the proposal of holographic dark energy where we know that this model is not able to explain the dark energy behavior of the universe [184]. The difference of (6.2) with the holographic model is that the holographic wavelength is the distance to the horizon of the universe, this integral has an extra scale factor outside the corresponding integral (6.2). The other main difference is that the holographic model has a free constant in the cosmological function Λ, while the equation (6.1) has no free parameters. So, let us think of the equation (6.1) as an effective proposal and its justification are the results that we find in this work.

In Figure 6.1 we compare the numerical solution of (6.4) with the evolution of the Hubble parameter in ΛCDM, $H_{\Lambda CDM} = H_0\sqrt{\Omega_{0m}e^{-3N} + \Omega_{0r}e^{-4N} + \Omega_\Lambda}$. Note that H has

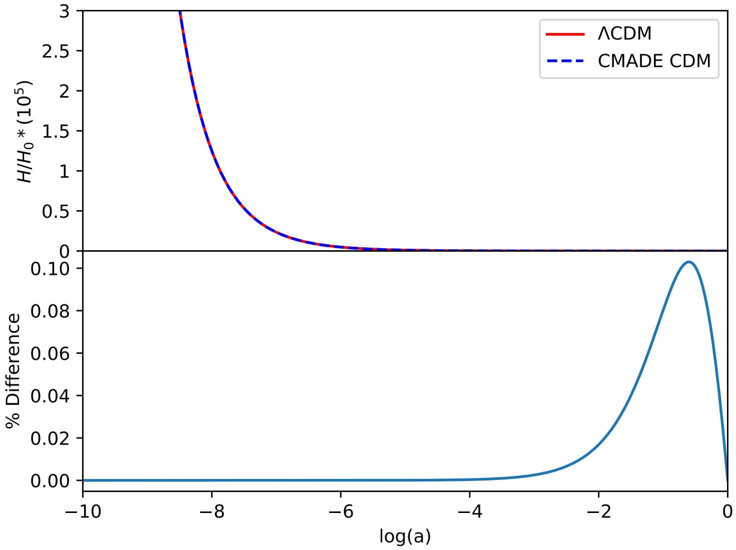


Figure 6.1: In the upper panel we show the evolution of the Hubble parameter using the CMaDE (equation (6.4), point line) and the ΛCDM model (solid line). We used the Planck values $\Omega_{0m} = 0.315$, $\Omega_{0r} = 10^{-4}$, $H_0 = 67.3$ km/s/Mpc in both plots and the value $\Omega_{0\Lambda} = 0.684$ for the ΛCDM curve and (6.7) for the CMaDE model. In the lower panel we show the proportional difference between both curves $(H_{CMaDE} - H_{\Lambda CDM})/H_{\Lambda CDM}$.

the same evolution for both models implying same predictions. Note too that the CMaDE density remains subdominant for large redshifts and is a bit different than the evolution of LCDM for small redshifts.

Solving numerically (6.4) for a flat space-time we carry out the integral (6.2) and we find that $R_H = 3.083$ in (6.1). With these results we obtain that

$$\Lambda = 2\left(\frac{\pi}{3.087}\right)^2 \frac{H_0^2}{c^2} = \frac{3H_0^2}{c^2}\Omega_{0\Lambda}, \tag{6.7}$$

We can see that the value of Ω_Λ strongly depends on the size of the wavelength. (6.2).

We can use the size of the universe horizon to determine the value of the wavelength λ. Thus we can determine the value of the CMaDE now and give an explanation of the cosmological and coincidence problems.

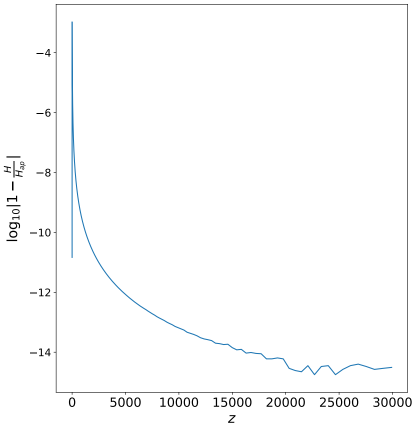


Figure 6.2: Evolution of the $1 - H/H_{app}$ using the numerical integration of (6.4) (H) and equation (6.10) (H_{ap}), with $q = 0.695$. We plot $\log(|1 - H/H_{app}|)$, observe that this ratio is always less than 10^{-3}.

In what follows we want to study the possibility that the CMaDE model is capable of reproducing all the observations of the universe that we have so far. Strictly speaking we have to solve equation (6.4) and solve the whole cosmology using it. However, in this chapter we first solve the entire cosmology using an approximation. Here we will focus on the temperature fluctuations of the cosmic microwave background (CMB) and the mass power spectrum (MPS) only, leaving a more in-depth analysis of the rest for future work. In order to find a suitable approximation to the equation (6.4), we proceed as follows. We know that during the epoch dominated by matter $H = 2/(3t) = H_0/a^{3/2}$ [185], so we found that the evolution of R_H is given by $R_H = 2\sqrt{a}$. Thus, during this time we have that

$$\Lambda^{MD} \sim \frac{\pi^2}{6}\frac{3H_0^2}{c^2}\frac{1}{4a}. \tag{6.8}$$

So, we find that the field equation for Λ^{MD} is $\dot{\Lambda}^{MD} + H\Lambda^{MD} = 0$. We use this approximation to get the Hubble parameter evolution, given as follows

$$H = H_0\sqrt{\Omega_{0m}e^{-3N} + \Omega_{0r}e^{-4N} + \Omega_{0k}e^{-2N} + \Omega_{0\Lambda}e^{-N}}. \tag{6.9}$$

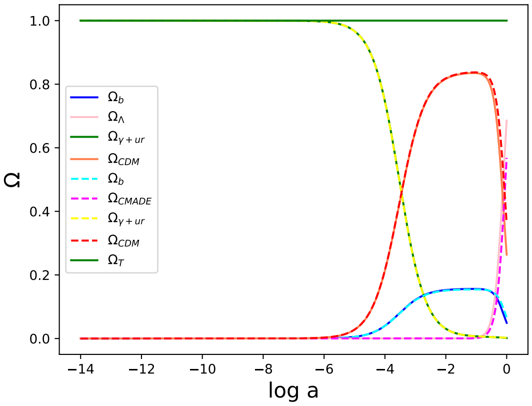


Figure 6.3: Evolution of the Ω's using equations (6.9) and (6.10) and the corresponding ones using the ΛCDM model.

However, this approximation is not good enough for the numerical solution of (6.4). Instead of that we will approximate it with the function

$$H_{app} = H_0\sqrt{\Omega_{0m}e^{-3N} + \Omega_{0r}e^{-4N} + \Omega_{0k}e^{-2N} + \Omega_{0\Lambda}e^{qN}}. \tag{6.10}$$

where q and $\Omega_{0\Lambda}$ are constants that fit the numerical solution. The similarity between the function (6.10) with the numerical integration of (6.4) is very good everywhere, see figure 6.2.

The radiation content of the universe, CMB photons plus neutrinos, is given by $\rho_r = 2(1 + 3 \times 7/8(4/11)^{4/3})T^4$. The CMB observations indicate that $T = 2.7255$ K, thus $\Omega_{0r} = 9.54 \times 10^{-5}$. We set $\Omega_{0\Lambda}$ such that $H_{app} = H_0$ at $N = 0$. These values, again, are very close to that obtained in ΛCDM.

In Figure 6.3 we see the evolution of the Ω's for the CMaDE model, using the function (6.10) and the ΛCDM model, we see the similarity of the evolution.

Thus, the next step is to see whether this approximation gives us the correct behavior of the CMB and MPS profiles. In Figure 6.4 we see the comparison between the profiles of the CMaDE and ΛCDM models using an amended version of CLASS code [131], again the similarity between both models is very close. The only difference we find for the flat universe is an excess of temperature predicted by the CMaDE model in the first maximum, but in the rest, of the two profiles, the coincidence with the observations using the Planck data is very good. It is very remarkable that the value of Ω_Λ in the CMaDE model is completely theoretical, so it is quite relevant that this match with the observations is so good. We believe that the small differences could be due to the fact that we are using an approximation for the CMaDE model and not the solution of the equation (6.4) or by some extra phenomenon.

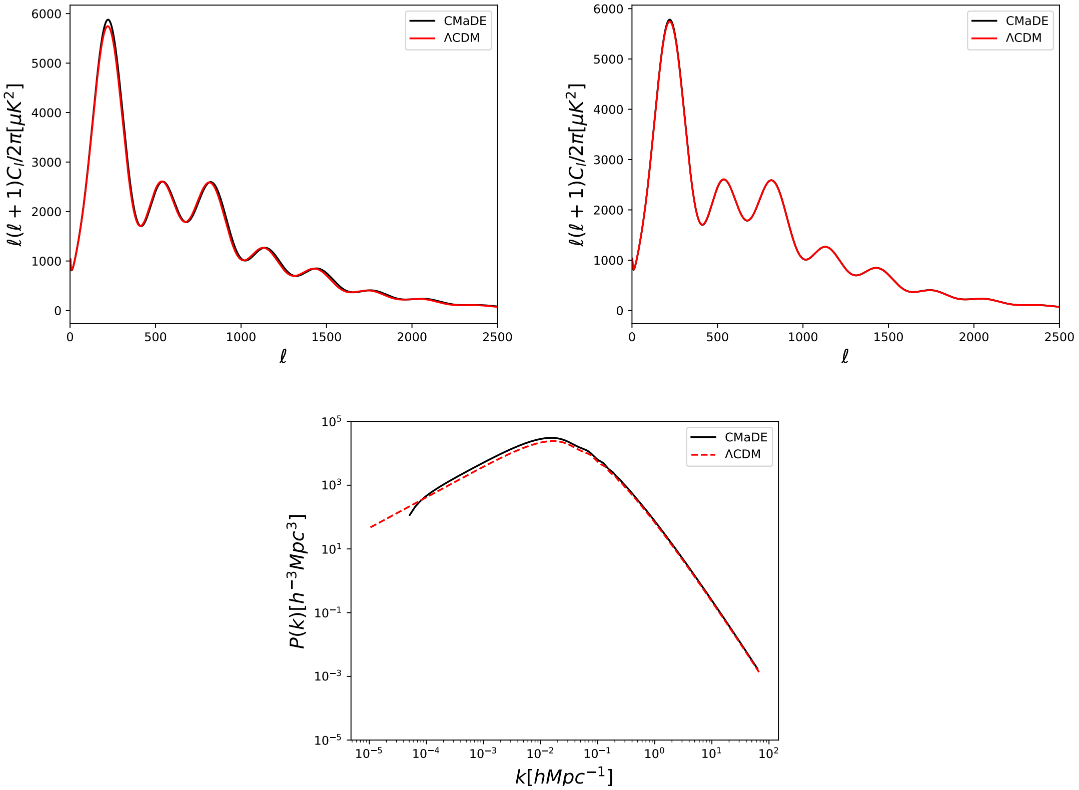


Figure 6.4: Profiles of the CMB for a flat universe (upper panel) and for a closed universe with $\Omega_{0k} = -0.003$ (middle panel) and MPS (lower panel) observations using an amended version of CLASS code [131]. We compare them with the best fit of the ΛCDM model, using data from the Planck satellite. Note that the CMB temperature fluctuations for the flat universe are the same as the ΛCDM, the only difference is in the first maximum. For the MPS there are very small discrepancies for the small structure. The CMaDE model settings are $\Omega_{0r} = 5.67 \times 10^{-5}$, $q = 0.694$, $H_0 = 72.6$ km/s/Mpc and $\Omega_{0b} = 0.044$ for the flat universe and $q = 0.695$, $\Omega_{0k} = -0.003$, $H_0 = 72.6$ km/s/Mpc and $\Omega_{0b} = 0.043$ for the closed universe. Observe that the value of H_0 is very close to the observed one from the local distance ladder [186]

Chapter 7

Conclusions and future work

In this work we presented the cosmological constraints of the Multi Scalar Field Dark Matter (MSFDM) model, in which we assume the dark matter is composed of different ultralight scalar fields. The idea was introduced to alleviate some of the cosmological and astrophysical discrepancies, for instance, the distinct values of the single scalar field mass obtained when considering different observations, where more than one field is needed in order to explain them. As a first approximation, we took for granted that the scalar fields are real, spatially homogeneous and they do not interact with each other. We presented the equations that describe the evolution of the background and perturbations for N scalar fields, and by using the polar change of variables we avoided the scalar field characteristic oscillations. Thus, we obtained a general expression for the fields evolution that depend on the potential and its derivatives, in particular, the equations for the potentials $V(\phi) = 1/2m_\phi^2\phi^2$, $V(\phi) = m_\phi^2 f^2 \left[1 + \cos(\phi/f)\right]$ and $V(\phi) = m_\phi^2 f^2 \left[\cosh(\phi/f) - 1\right]$. Under this change of variables, these three configurations are described by a single system of equations where the parameter λ_ϕ is able to decide the type of potential to use.

We showed the evolution for the background, mass power spectrum (MPS) and the CMB power spectrum using a modified version of the CLASS code [131, 65, 95] with two dark matter components. We considered the following combinations: a) cold dark matter and a scalar field with potential $V(\phi) = 1/2m_\phi^2\phi^2$, and two scalar field models, with b) both potentials $V(\phi) = 1/2m_\phi^2\phi^2$, c) $V(\phi) = 1/2m_\phi^2\phi^2$ and $V(\phi) = m_\phi^2 f^2\left[1+\cos(\phi/f)\right]$, and d) $V(\phi) = 1/2m_\phi^2\phi^2$ and $V(\phi) = m_\phi^2 f^2\left[\cosh(\phi/f)-1\right]$. Since the fields are independent from each other, we introduced the parameter R to have the field one as reference.

We showed that the evolution of the background is mainly affected at the beginning of the scalar fields oscillations since its amplitude depends on the masses and the contribution that each one has to the total dark matter, being bounded by the heaviest and lightest masses; however these do not depend on λ_ϕ. After the oscillations have started, the fields evolution only depend on $\Omega_{\phi i,0}$. On the other hand we found that, in all combinations, the MPS presents the characteristic cut-off of the single field at small scales. However, it does not present the oscillations of the single case due to the superposition of the different fields. This cut-off is far form the ΛCDM behavior when the lightest field dominates, and if the heaviest one dominates the MPS behavior approaches the ΛCDM model. For the case that one of the fields has the axion potential, the characteristic bump appeared when it had a light mass with large λ_ϕ values. On the contrary, if the mass is heavier, the bump disappeared regardless of the values of λ_ϕ and R. On the other hand, in the CMB power spectrum, we have not found significant changes unless one of the fields has $m_{\phi i} \leq 10^{-26}$ eV. That is, regardless of the number of fields we can always find a combination of the fields that matches the CMB observations where the total contribution of dark matter is $\Omega_{\rm DM} = 0.264$, given by Planck 18 data [10].

We performed the parameter inference analysis with the Monte Python code [158, 159] using BAO, Big Bang Nucleosynthesis, Ly-α forest and Supernovae for a single scalar field with the three potentials mentioned above and for double scalar fields. In the latter case, for simplicity we used the following combination of potentials: $V(\phi_1) = 1/2m_{\phi 1}^2\phi_1^2$ with $V(\phi_2) = 1/2m_{\phi 2}^2\phi_2^2$, $V(\phi_1) = 1/2m_{\phi 1}^2\phi_1^2$ with $V(\phi_2) = m_{\phi 2}^2 f^2\left[1+\cos(\phi_2/f)\right]$ and $V(\phi_1) = m_{\phi 1}^2 f^2\left[1+\cos(\phi_1/f)\right]$ with $V(\phi_2) = m_{\phi 2}^2 f^2\left[1+\cos(\phi_2/f)\right]$. For the single case we found a bound for the mass values that corresponds to the one reported by the Ly-α data. We also presented the constrictions for the potential $V(\phi) = m_\phi^2 f^2\left[\cosh(\phi/f)-1\right]$ where we have found a bound for λ_ϕ and, therefore, for the scalar field mass where, the latter, is also in agreement with the reported by the Ly-α. In this case, the excluded values correspond to the cases in which the scalar field contributes to radiation as can be seen in Figure 5.2. For the double field models we found bounds for the masses and that the contribution of the fields depends on each other. Within our confidence regions are the masses reported in [114], in which more than one scalar fields is needed to explain the observed galaxies halos.

Finally, we have found that adding more scalar fields in order to explain astrophysical phenomena does not affect the known cosmology. So the MSFDM can be an alternative candidate to dark matter that is able to explain the observations at the cosmological and astrophysical levels. Its difference with other models is seen in the MPS at small scales. The results presented here can be generalized to larger number of fields with different potentials. We expect that forthcoming observations of collaborations such as DESI and LSST will allow us to better constrain the parameters of our model.

The future work, for this model, is to study the case in which the fields are complex and have different behaviors throughout their evolution, as well as interactions between complex or real fields. Cases with different potentials can also be studied. For example, a pending job is to study the potential that includes the interaction of a field with itself, which is paused because by using the polar change of variables the differential equations become complicated to solve even numerically because we have not been able to find the initial conditions for the system. Another topic that is interesting to study is the interaction of one scalar field with another, and even the interaction between scalar field dark matter with scalar field dark energy. We must also investigate the impact of the hyperbolic cosine type potential on the initial conditions of the other fields, since we must remember that, as we saw in the section 3.4, this potential only allows combinations in which both fields have the same contribution to the total dark matter.

On the other hand, as mentioned in the section 5.1, we use the MPS data obtained from Ly-α where ΛCDM has been assumed as the fiducial model. To do it properly, we must do the hydrodynamic simulations that include the information from our model to extract the Lyα information now with the correct model and make the Bayesian inference with these results.

Regarding the CMaDE model, we saw that using the approach presented in the chapter 6 we can numerically reproduce the same results as the LCDM model. However, we want to solve the entire differential equation without the approximation. Future work is to introduce it in CLASS. However, it remains to be determined if the normal version of CLASS can solve it or if it is necessary to look for another one, for example, Hi-CLASS. Once this is done, the next step is to find the constraints of the model through Bayesian inference as was done with the MSFDM model. The idea is to separately use the Planck 18 CMB data and the Pantheon SN data to determine the value of H0 and see if CMaDE can resolve the discrepancy between these data sets.

Appendix A

CLASS code.

The Cosmic Linear Anisotropy Solving System (CLASS) is a code designed to simulate the evolution of linear perturbations in the universe and to compute CMB and large scale structure observables using the ΛCDM as fiducial model [131]. It is written in C to take advantage of the computational speed that this programming language has compared to others. In addition to the use of the "classes"[1] that allowed to write it in such a way that they can be calculated separately, for example, the background and the perturbations. In this work we are only interested in the part of the dark matter, so the rest of the equations to be solved are the same as in ΛCDM, which can be seen in [131, 130].

A.1 Installing CLASS

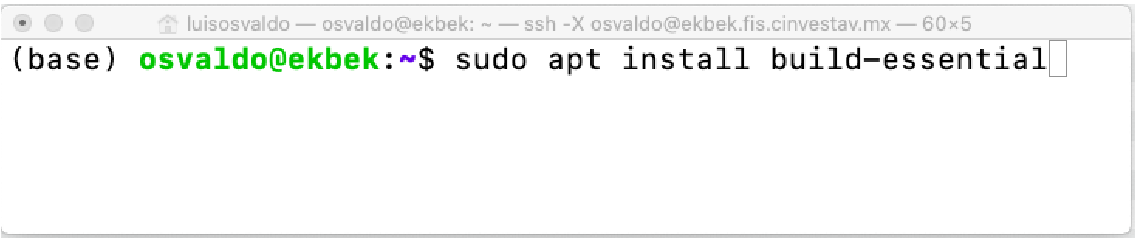
Throughout this work we use two operating systems: macOS and Ubuntu. Although they are both based on linux, the way to install CLASS is a bit different.

[1] For readers unfamiliar with programming, we can say that a class is a template that defines variables, functions, and methods. This will be useful because we can define a class in the background only once and we can call it throughout the code without having to define it every time we use it.

A.1.1 Ubuntu

This operating system is installed on the ekbek server belonging to the Astrophysical Super Computing Laboratory (LaSumA for its acronym in Spanish). It currently has the version 20.04.3 LTS with 8 processors Intel(R) Core(TM) i7-7700 CPU @3.60 GHz with 4 cores each and 8 GB of RAM memory.

Since CLASS is written in C, we need a compiler. In this case we will use gcc. We also need to install make. Usually they are already installed with Ubuntu but if not, it can be done with



luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 60×5

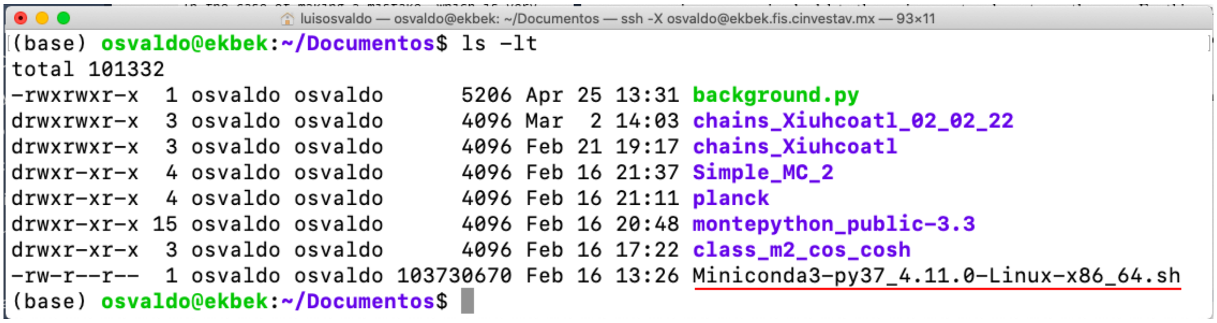
```
(base) osvaldo@ekbek:~$ sudo apt install build-essential
```

If we only wanted to use CLASS, these requirements would suffice. However, as we will see later, we need python to be able to do the link with Monte Python.

Although most versions of Ubuntu already include some version of python, we recommend using a virtual environment as a good programming practice. A python environment allows us to install, in isolation, specific versions of libraries that we are going to use. That is, we can have as many python environments as we want and what we do to one doesn't affect the others. In the case of making a mistake, which is very common when programming, we can simply delete the environment and create another one. For this purpose, we recommend installing miniconda [187], which is a minimal conda installer used to manage python and its libraries.

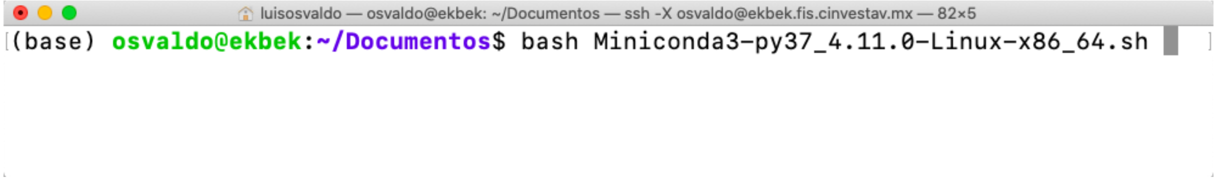
We must download the installer from [188]. In this case, we download the version "Miniconda3-py37_4.11.0-Linux-86_64.sh". See Figure A.1.

To install it we write in the terminal "bash Miniconda3-py37_4.11.0-Linux-86_64.sh" as in Figure A.2


luisosvaldo — osvaldo@ekbek: ~/Documentos — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 93×11

```
(base) osvaldo@ekbek:~/Documentos$ ls -lt
total 101332
-rwxrwxr-x  1 osvaldo osvaldo      5206 Apr 25 13:31 background.py
drwxrwxr-x  3 osvaldo osvaldo      4096 Mar  2 14:03 chains_Xiuhcoatl_02_02_22
drwxrwxr-x  3 osvaldo osvaldo      4096 Feb 21 19:17 chains_Xiuhcoatl
drwxr-xr-x  4 osvaldo osvaldo      4096 Feb 16 21:37 Simple_MC_2
drwxr-xr-x  4 osvaldo osvaldo      4096 Feb 16 21:11 planck
drwxr-xr-x 15 osvaldo osvaldo      4096 Feb 16 20:48 montepython_public-3.3
drwxr-xr-x  3 osvaldo osvaldo      4096 Feb 16 17:22 class_m2_cos_cosh
-rw-r--r--  1 osvaldo osvaldo 103730670 Feb 16 13:26 Miniconda3-py37_4.11.0-Linux-x86_64.sh
(base) osvaldo@ekbek:~/Documentos$ 
```

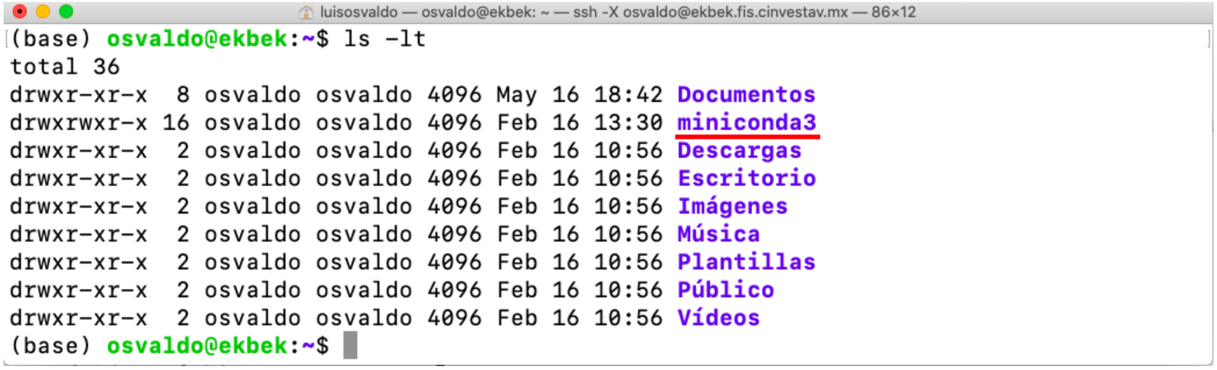
Figure A.1: We download the file in the “Documentos” directory.


luisosvaldo — osvaldo@ekbek: ~/Documentos — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 82×5

```
(base) osvaldo@ekbek:~/Documentos$ bash Miniconda3-py37_4.11.0-Linux-x86_64.sh 
```

Figure A.2: Installing miniconda.

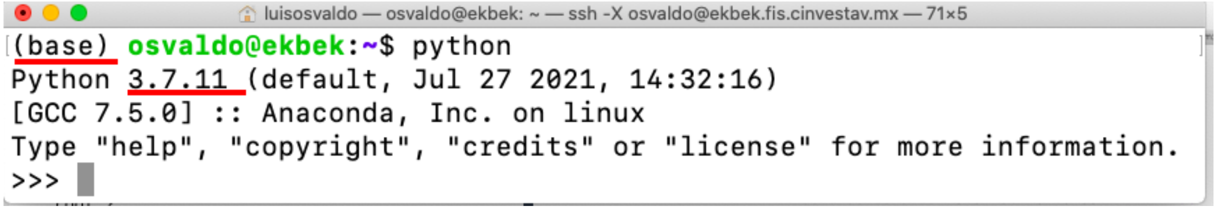
We just have to accept everything. This will create a new directory called ”miniconda3” (because we have downloaded the version that has python 3.7) in the root of our user as can be seen in Figure A.3. In case we have some version of python previously installed, miniconda will ask us to modify the PATH (.bashrc file) so that this python is used by default. We must tell it “yes”, please.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 86×12

```
(base) osvaldo@ekbek:~$ ls -lt
total 36
drwxr-xr-x  8 osvaldo osvaldo 4096 May 16 18:42 Documentos
drwxrwxr-x 16 osvaldo osvaldo 4096 Feb 16 13:30 miniconda3
drwxr-xr-x  2 osvaldo osvaldo 4096 Feb 16 10:56 Descargas
drwxr-xr-x  2 osvaldo osvaldo 4096 Feb 16 10:56 Escritorio
drwxr-xr-x  2 osvaldo osvaldo 4096 Feb 16 10:56 Imágenes
drwxr-xr-x  2 osvaldo osvaldo 4096 Feb 16 10:56 Música
drwxr-xr-x  2 osvaldo osvaldo 4096 Feb 16 10:56 Plantillas
drwxr-xr-x  2 osvaldo osvaldo 4096 Feb 16 10:56 Público
drwxr-xr-x  2 osvaldo osvaldo 4096 Feb 16 10:56 Vídeos
(base) osvaldo@ekbek:~$ 
```

Figure A.3: “miniconda3” directory in our user root.

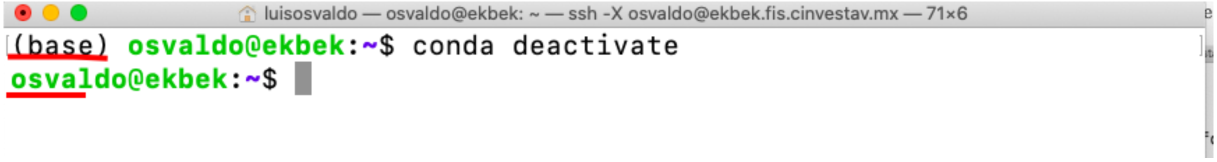
Now we close and reopen the terminal. If everything has gone well, we will see next to our user the word “(base)” as in Figure A.4. This means that we are working with the default python environment created by miniconda, in this case it is python 3.7.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×5

```
(base) osvaldo@ekbek:~$ python
Python 3.7.11 (default, Jul 27 2021, 14:32:16)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Figure A.4: Python base environment.

If we want to deactivate the base environment of miniconda and use the python that Ubuntu has by default, we write “conda deactivate”. See Figure A.5.

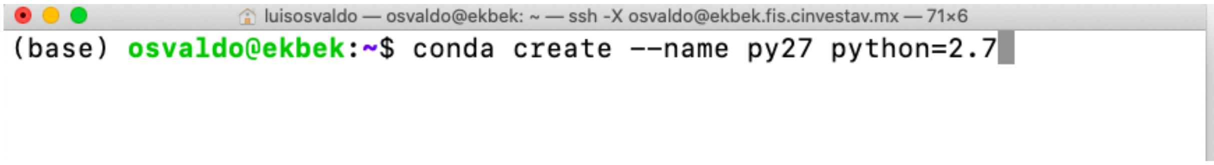

luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6

```
(base) osvaldo@ekbek:~$ conda deactivate
osvaldo@ekbek:~$ 
```

Figure A.5: Deactivating Python base environment.

If we want to activate it again, we just have to close and open the terminal.

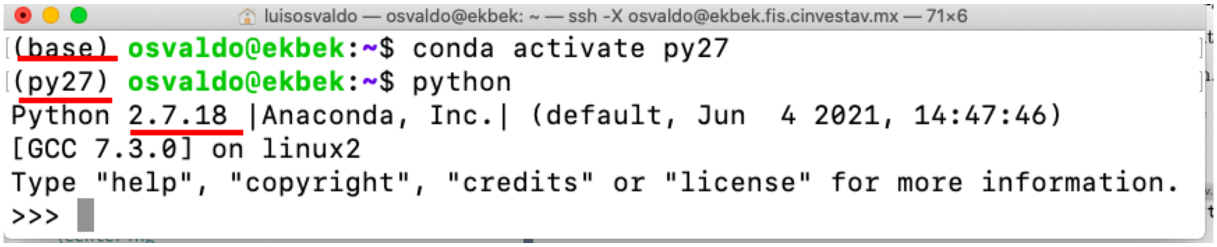
Now we must create the environment in which we are going to work. Although the latest versions of Monte Python and CLASS have compatibility with python 3, we use python 2.7 because Monte Python was written in this version. We must write “conda create –name py27 python=2.7” in the terminal, this will create a python 2.7 environment called py27. See Figure A.6. We can give the environments the name we want with the version of python we need.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6

```
(base) osvaldo@ekbek:~$ conda create --name py27 python=2.7
```

Figure A.6: Creating Python py27 environment.

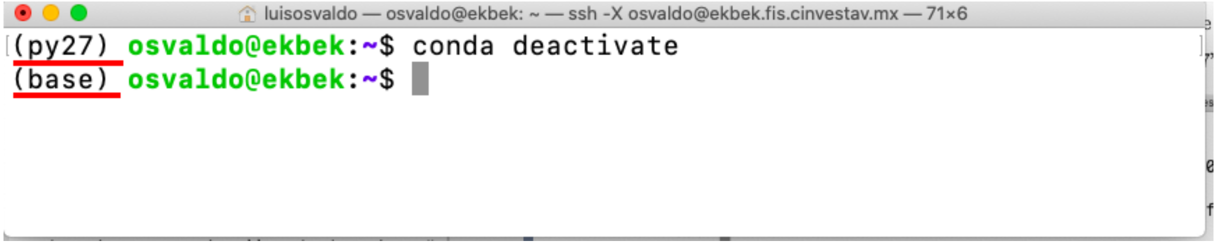
To activate it we write “conda activate py27”. In Figure A.7 we can see the change between environments, that is, we went from “base” to “py27” with python 2.7.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6

```
(base) osvaldo@ekbek:~$ conda activate py27
(py27) osvaldo@ekbek:~$ python
Python 2.7.18 |Anaconda, Inc.| (default, Jun  4 2021, 14:47:46)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Figure A.7: Activating Python py27 environment.

To deactivate we write “conda deactivate” as in Figure A.8. We see the change from “py27” to “base”.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6

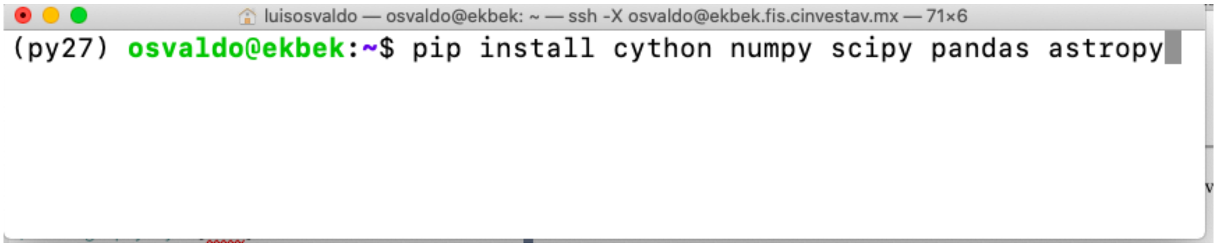
```
(py27) osvaldo@ekbek:~$ conda deactivate
(base) osvaldo@ekbek:~$ 
```

Figure A.8: Deactivating Python py27 environment.

The next step is to install the necessary python libraries. For this we will use the pip and conda package managers. We use conda when the library dependencies are outside of python and thus cannot be handled by pip. Activate the py27 environment and write in the terminal “pip install cython numpy scipy pandas astropy” as in Figure A.9. This will install the libraries cython, numpy, scipy, pandas and astropy at the same time. We can also install them one by one. For example, “pip install cython”, press Enter; then “pip install numpy” and so on.

For the numexpr and mpi4py [2] we will use conda. This is “conda install numexpr”, we wait for this library to finish installing. Then we write “conda install mpi4py” as in Figure

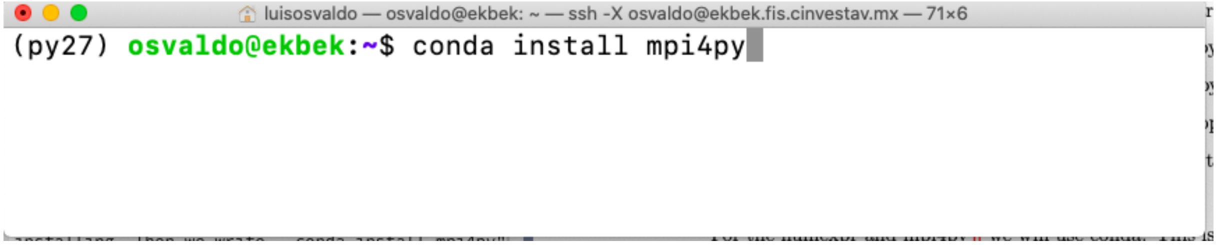
[2] This library will be essential when running Monte Python because it will allow us to parallelize the code, that is, run several processes at the same time.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6

```
(py27) osvaldo@ekbek:~$ pip install cython numpy scipy pandas astropy
```

Figure A.9: Installing libraries with pip.

A.10.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6

```
(py27) osvaldo@ekbek:~$ conda install mpi4py
```

Figure A.10: Installing libraries with conda.

In addition to the C compiler, python and their libraries; we also need to install LaTeX to do the plotting when we study the Monte Python results. For this we must write in the terminal "sudo apt install texlive-latex-recommended" and then "sudo apt-get install dvipng texlive-latex-extra texlive-fonts-recommended" as seen in Figure A.11.

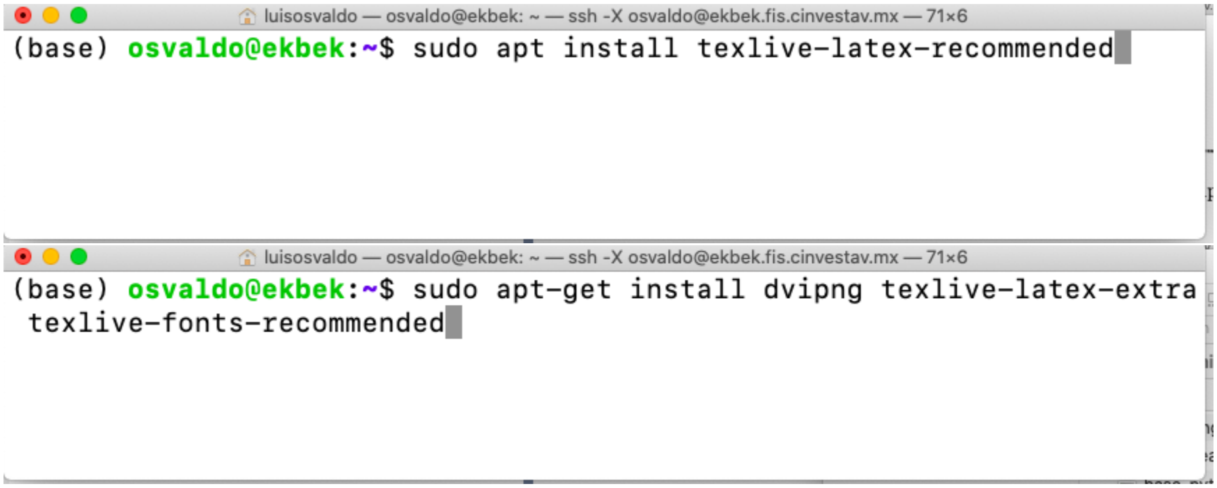
We already have everything we need to install CLASS. First we need to download it from [189] for the latest version (without MSFDM) or from [132] for the amended version used in this work. In any case, we will download it as ZIP [3] as seen in Figure A.12.

Once downloaded and decompressed, we will enter the "class_public" directory. See Figure A.13.

Then, activate the py27 environment, write "make" and wait for the process to finish as seen in Figure A.14.

The first test to know that CLASS was installed correctly is to enter python and type

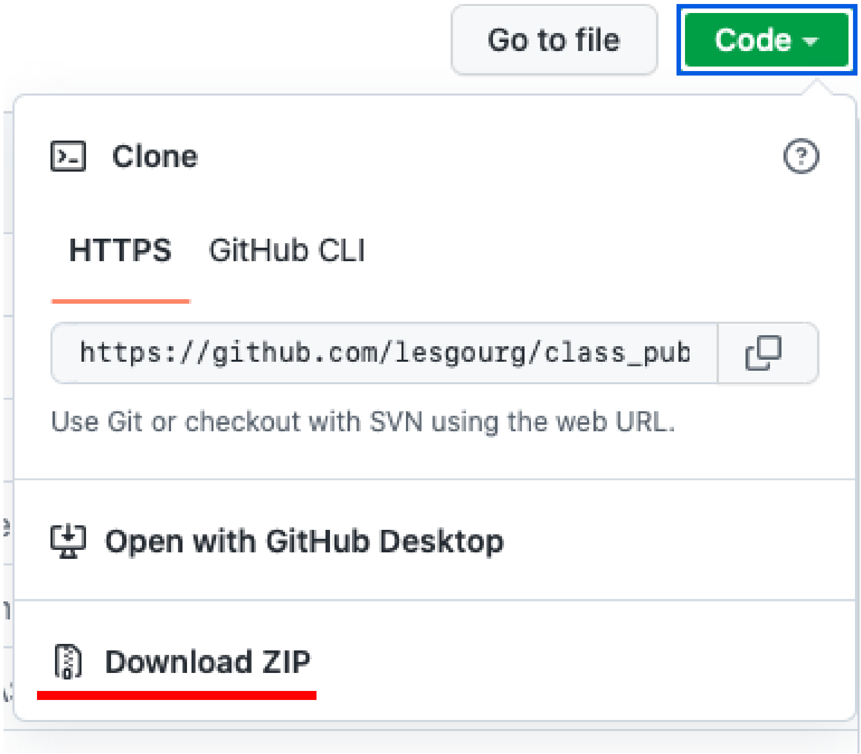
[3] For more advanced users, it can be downloaded using git which allows us to have updated versions. See [190] for a tutorial.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6
luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 71×6

```
(base) osvaldo@ekbek:~$ sudo apt install texlive-latex-recommended
```

```
(base) osvaldo@ekbek:~$ sudo apt-get install dvipng texlive-latex-extra
 texlive-fonts-recommended
```

Figure A.11: Installing LaTeX.

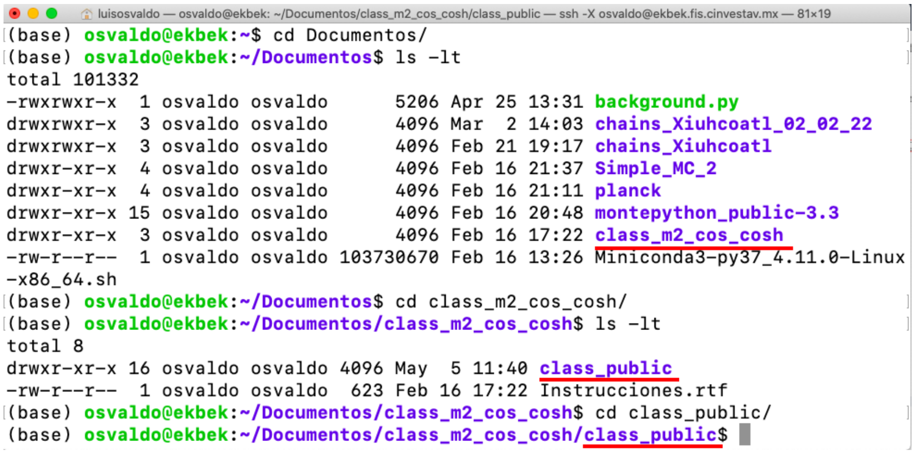


```
https://github.com/lesgourg/class_pub
```

Use Git or checkout with SVN using the web URL.

Figure A.12: Downloading CLASS.

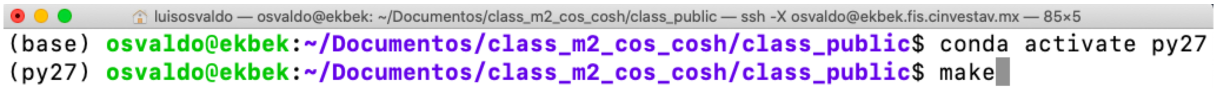
“import classy” in the terminal. It should not display any legend, it should display the following line of code. See Figure A.15. The classy library is a wrapper written in python



luisosvaldo — osvaldo@ekbek: ~/Documentos/class_m2_cos_cosh/class_public — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 81×19

```
(base) osvaldo@ekbek:~$ cd Documentos/
(base) osvaldo@ekbek:~/Documentos$ ls -lt
total 101332
-rwxrwxr-x  1 osvaldo osvaldo      5206 Apr 25 13:31 background.py
drwxrwxr-x  3 osvaldo osvaldo      4096 Mar  2 14:03 chains_Xiuhcoatl_02_02_22
drwxrwxr-x  3 osvaldo osvaldo      4096 Feb 21 19:17 chains_Xiuhcoatl
drwxr-xr-x  4 osvaldo osvaldo      4096 Feb 16 21:37 Simple_MC_2
drwxr-xr-x  4 osvaldo osvaldo      4096 Feb 16 21:11 planck
drwxr-xr-x 15 osvaldo osvaldo      4096 Feb 16 20:48 montepython_public-3.3
drwxr-xr-x  3 osvaldo osvaldo      4096 Feb 16 17:22 class_m2_cos_cosh
-rw-r--r--  1 osvaldo osvaldo 103730670 Feb 16 13:26 Miniconda3-py37_4.11.0-Linux
-x86_64.sh
(base) osvaldo@ekbek:~/Documentos$ cd class_m2_cos_cosh/
(base) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh$ ls -lt
total 8
drwxr-xr-x 16 osvaldo osvaldo 4096 May  5 11:40 class_public
-rw-r--r--  1 osvaldo osvaldo  623 Feb 16 17:22 Instrucciones.rtf
(base) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh$ cd class_public/
(base) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh/class_public$ 
```

Figure A.13: Enter the class_public directory.

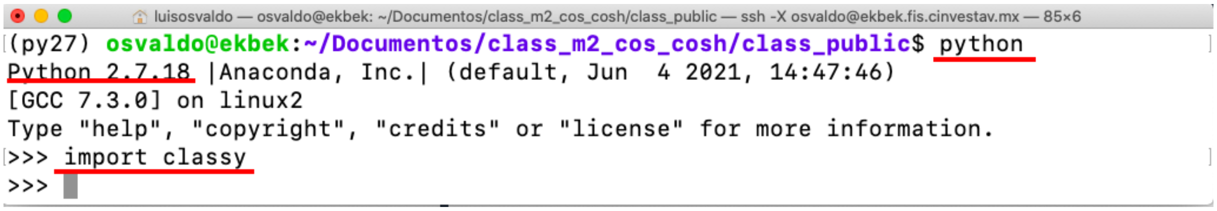


luisosvaldo — osvaldo@ekbek: ~/Documentos/class_m2_cos_cosh/class_public — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 85×5

```
(base) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh/class_public$ conda activate py27
(py27) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh/class_public$ make
```

Figure A.14: Using make to compile CLASS.

generated by compiling CLASS. This will be the link to Monte Python. On the other hand, we can also use it as any python library and run our code in a jupyter notebook, for example [4]. However, for the purposes of this work, we will work with CLASS from C.



luisosvaldo — osvaldo@ekbek: ~/Documentos/class_m2_cos_cosh/class_public — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 85×6

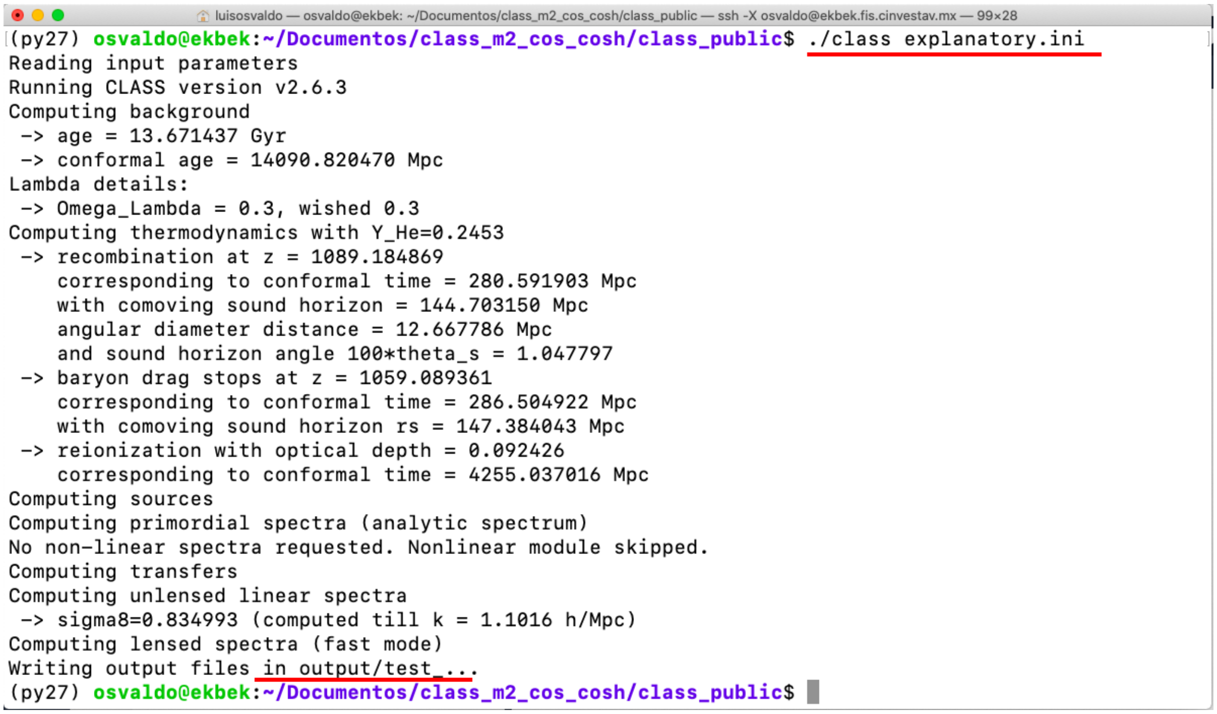
```
(py27) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh/class_public$ python
Python 2.7.18 |Anaconda, Inc.| (default, Jun  4 2021, 14:47:46)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import classy
>>> 
```

Figure A.15: First test importing classy.

The next test is to run CLASS with an input file by typing "./class explanatory.ini". In Figure A.16 we can see an example of a correct execution as well as the direction and name of the output files. On the other hand, "explanatory.ini" is the default input file and it is recommended not to modify it as it contains detailed instructions for all the parameters

[4] See the notebooks directory in class_public

and models included in CLASS, as an example see Figure A.17. If you want to use another model or change the parameters, we recommended to copy it or create a new file. The input files will be explained later.



luisosvaldo — osvaldo@ekbek: ~/Documentos/class_m2_cos_cosh/class_public — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 99×28

```
(py27) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh/class_public$ ./class explanatory.ini
Reading input parameters
Running CLASS version v2.6.3
Computing background
 -> age = 13.671437 Gyr
 -> conformal age = 14090.820470 Mpc
Lambda details:
 -> Omega_Lambda = 0.3, wished 0.3
Computing thermodynamics with Y_He=0.2453
 -> recombination at z = 1089.184869
    corresponding to conformal time = 280.591903 Mpc
    with comoving sound horizon = 144.703150 Mpc
    angular diameter distance = 12.667786 Mpc
    and sound horizon angle 100*theta_s = 1.047797
 -> baryon drag stops at z = 1059.089361
    corresponding to conformal time = 286.504922 Mpc
    with comoving sound horizon rs = 147.384043 Mpc
 -> reionization with optical depth = 0.092426
    corresponding to conformal time = 4255.037016 Mpc
Computing sources
Computing primordial spectra (analytic spectrum)
No non-linear spectra requested. Nonlinear module skipped.
Computing transfers
Computing unlensed linear spectra
 -> sigma8=0.834993 (computed till k = 1.1016 h/Mpc)
Computing lensed spectra (fast mode)
Writing output files in output/test_...
(py27) osvaldo@ekbek:~/Documentos/class_m2_cos_cosh/class_public$
```

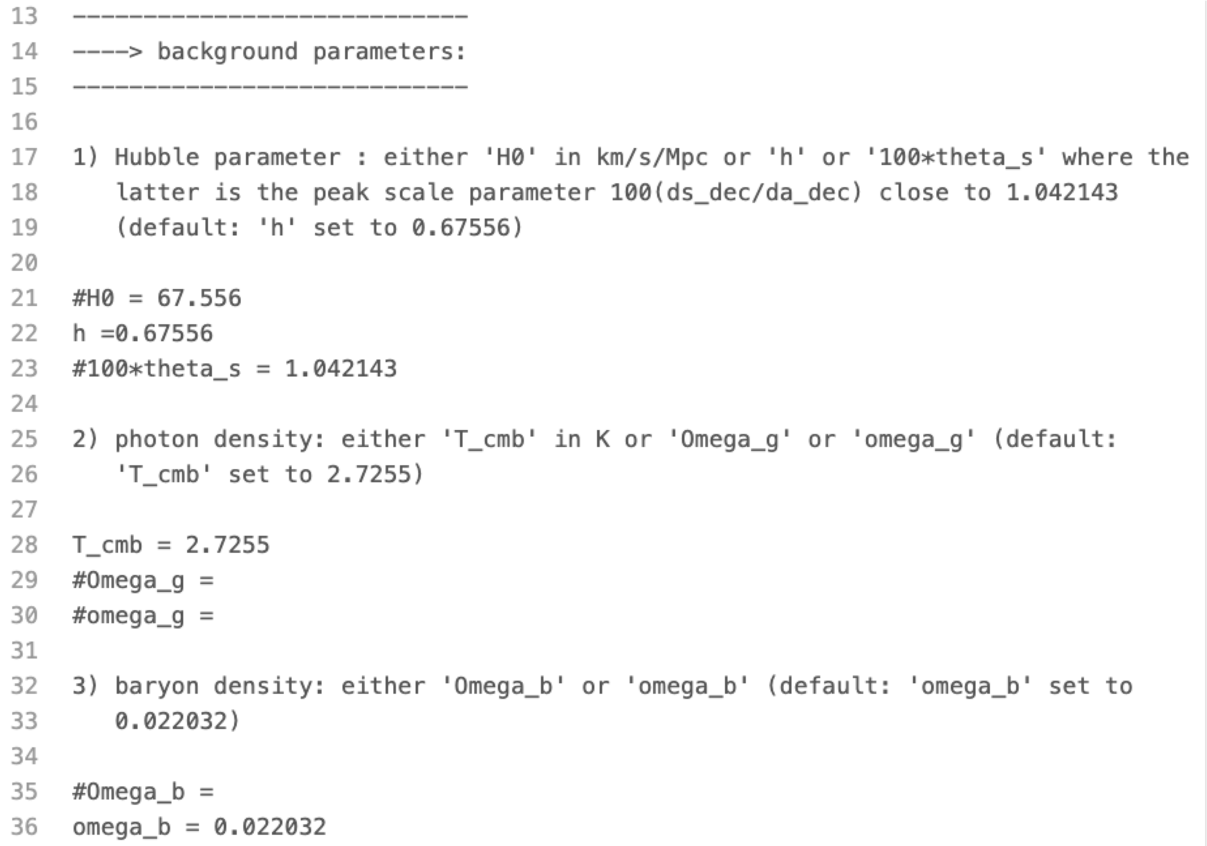
Figure A.16: Second test using an input file.

A.1.2 macOS

We must follow the same steps and in the same order as in Ubuntu. Here we will mention only the differences. The first difference is that on macOS we don't have apt-get to install libraries from the terminal, instead we have homebrew. Instructions for installing it can be found at [191], see Figure A.18.

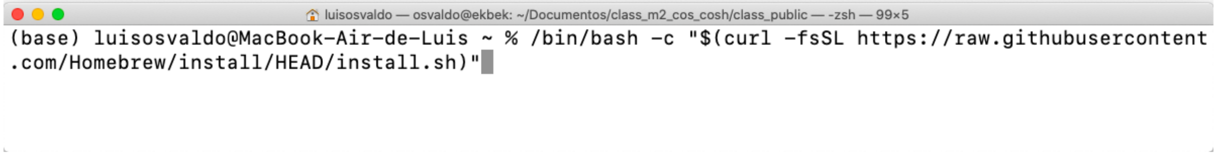
After it has been installed, we type “brew install gcc” and then “brew install cmake” as in Figure A.19. cmake is the analogue of make in Ubuntu.

The other difference is before CLASS is compiled, that is, before typing make in the terminal inside the class_public directory. We must modify the Makefile file that is in the



```
----------------------------
----> background parameters:
----------------------------

1) Hubble parameter : either 'H0' in km/s/Mpc or 'h' or '100*theta_s' where the
   latter is the peak scale parameter 100(ds_dec/da_dec) close to 1.042143
   (default: 'h' set to 0.67556)

#H0 = 67.556
h =0.67556
#100*theta_s = 1.042143

2) photon density: either 'T_cmb' in K or 'Omega_g' or 'omega_g' (default:
   'T_cmb' set to 2.7255)

T_cmb = 2.7255
#Omega_g =
#omega_g =

3) baryon density: either 'Omega_b' or 'omega_b' (default: 'omega_b' set to
   0.022032)

#Omega_b =
omega_b = 0.022032
```

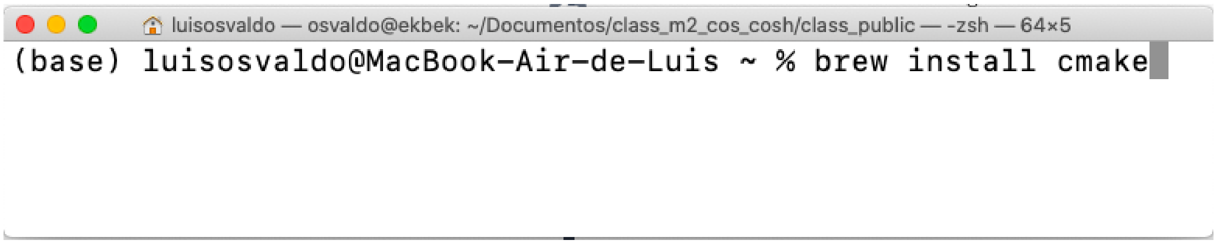
Figure A.17: explanatory.ini file snippet.



luisosvaldo — osvaldo@ekbek: ~/Documentos/class_m2_cos_cosh/class_public — -zsh — 99×5

```
(base) luisosvaldo@MacBook-Air-de-Luis ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent
.com/Homebrew/install/HEAD/install.sh)"
```

Figure A.18: Installing homebrew.

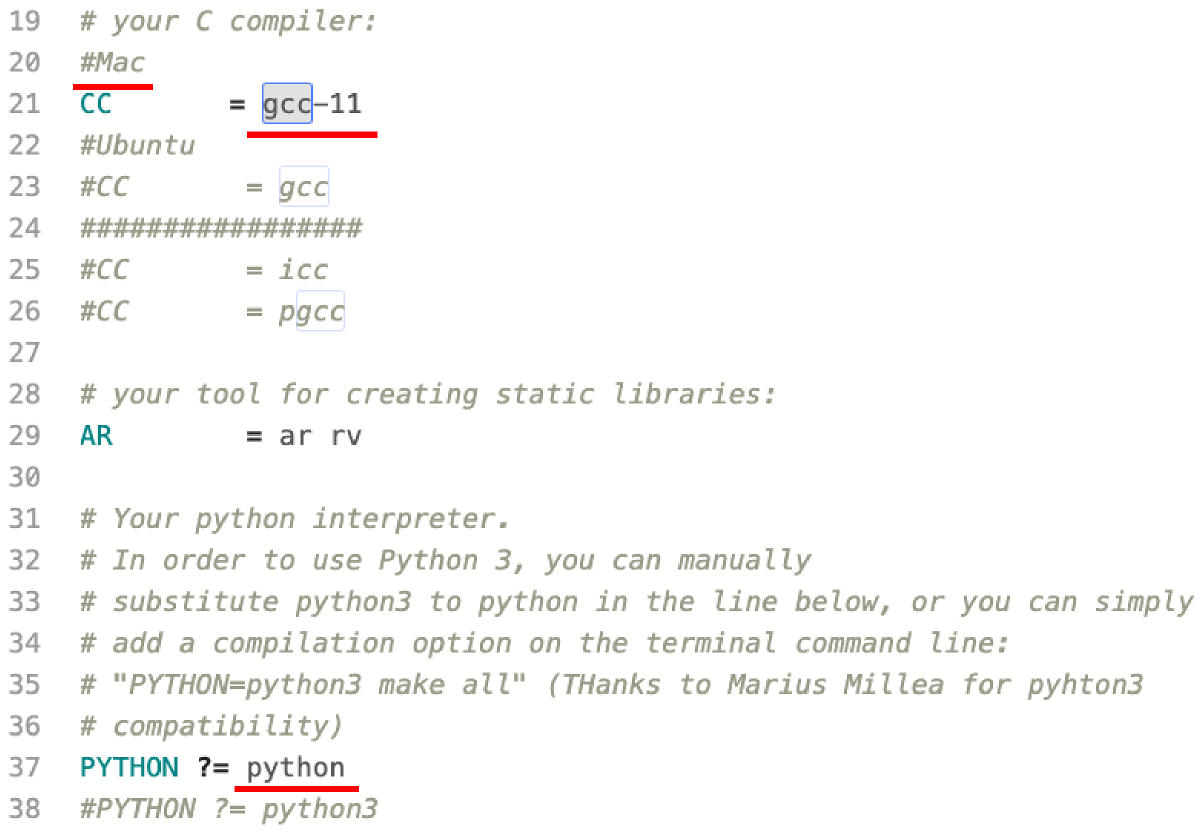


luisosvaldo — osvaldo@ekbek: ~/Documentos/class_m2_cos_cosh/class_public — -zsh — 64×5

```
(base) luisosvaldo@MacBook-Air-de-Luis ~ % brew install cmake
```

Figure A.19: Using homebrew to install libraries.

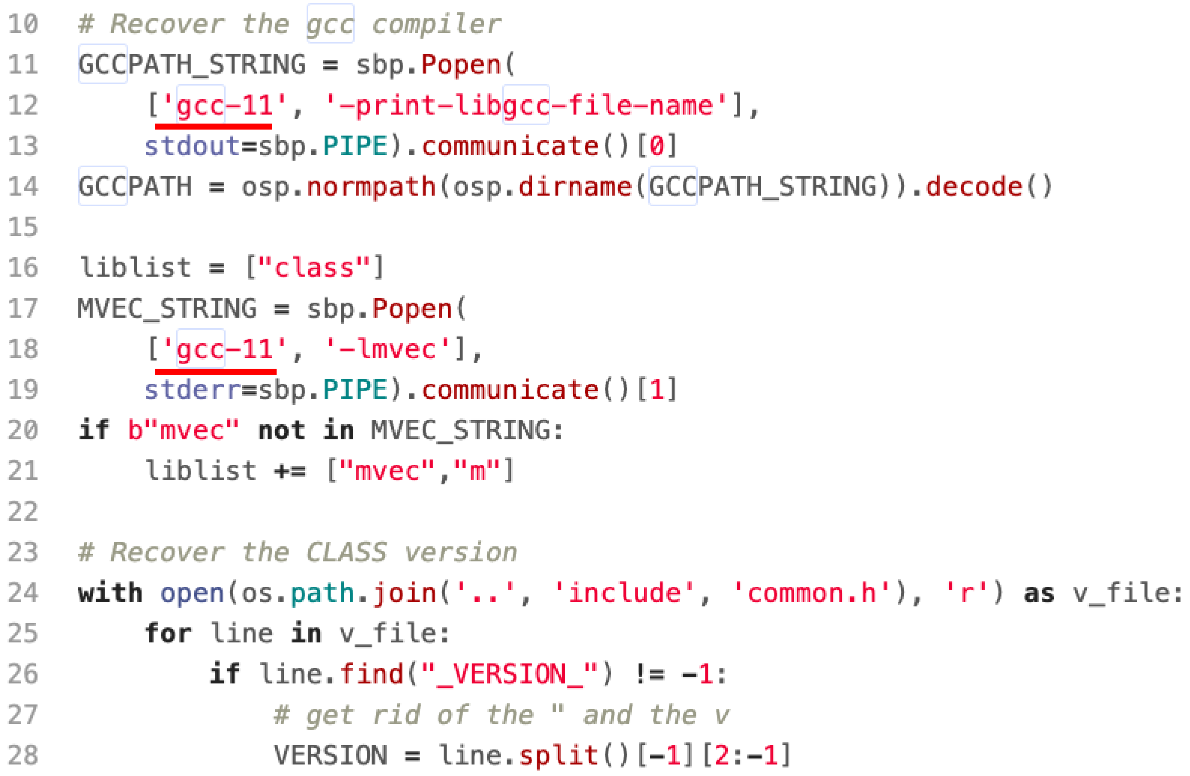
aforementioned directory. Here we need to add the gcc version number that homebrew puts when installing the compiler. As can be seen in the Figure A.20, in this case we have gcc-11. In principle, we should do the same with python but, since we have installed miniconda, it still uses the word python to refer to all versions so it remains the same.



```
# your C compiler:
#Mac
CC         = gcc-11
#Ubuntu
#CC         = gcc
#################
#CC         = icc
#CC         = pgcc

# your tool for creating static libraries:
AR          = ar rv

# Your python interpreter.
# In order to use Python 3, you can manually
# substitute python3 to python in the line below, or you can simply
# add a compilation option on the terminal command line:
# "PYTHON=python3 make all" (THanks to Marius Millea for pyhton3
# compatibility)
PYTHON ?= python
#PYTHON ?= python3
```

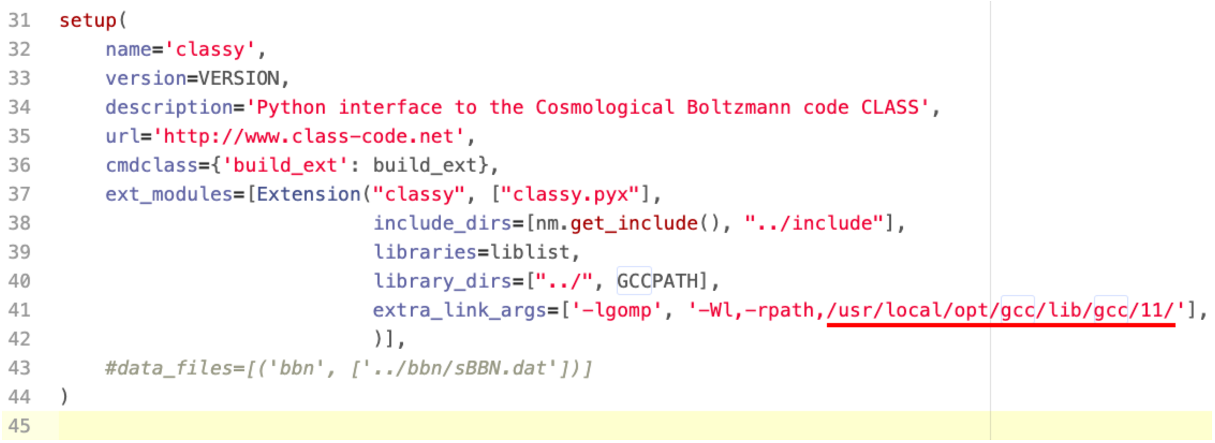
Figure A.20: Modifying the Makefile file to make it compatible with mac. We changed the C compiler name.

We also need to modify the setup.py file inside the class_public/python directory and add the version of gcc installed with brew and add the direction where the compiler was installed as seen in Figures A.21 and A.22.



```
# Recover the gcc compiler
GCCPATH_STRING = sbp.Popen(
    ['gcc-11', '-print-libgcc-file-name'],
    stdout=sbp.PIPE).communicate()[0]
GCCPATH = osp.normpath(osp.dirname(GCCPATH_STRING)).decode()

liblist = ["class"]
MVEC_STRING = sbp.Popen(
    ['gcc-11', '-lmvec'],
    stderr=sbp.PIPE).communicate()[1]
if b"mvec" not in MVEC_STRING:
    liblist += ["mvec","m"]

# Recover the CLASS version
with open(os.path.join('..', 'include', 'common.h'), 'r') as v_file:
    for line in v_file:
        if line.find("_VERSION_") != -1:
            # get rid of the " and the v
            VERSION = line.split()[-1][2:-1]
```

Figure A.21: We changed the C compiler name in setup.py file.



```
setup(
    name='classy',
    version=VERSION,
    description='Python interface to the Cosmological Boltzmann code CLASS',
    url='http://www.class-code.net',
    cmdclass={'build_ext': build_ext},
    ext_modules=[Extension("classy", ["classy.pyx"],
                           include_dirs=[nm.get_include(), "../include"],
                           libraries=liblist,
                           library_dirs=["../", GCCPATH],
                           extra_link_args=['-lgomp', '-Wl,-rpath,/usr/local/opt/gcc/lib/gcc/11/'],
                           )],
    #data_files=[('bbn', ['../bbn/sBBN.dat'])]
)

```

Figure A.22: Adding the C compiler direction in setup.py file.

Appendix B

Modifying CLASS

In this section we will show how we added scalar fields to CLASS, we take these references [65, 96, 99] and the implementation of the scalar field as dark energy that is included by default in CLASS as a guide. The difference is that for dark energy the scalar field is an option to choose between cosmological constant, fluid or the field. Whereas we include the scalar fields as extra components of the Universe separated from dark energy. This allows us to have both scalar field dark matter and scalar field dark energy.

The main CLASS files are in the class_public/source directory. This is where all the equations to be solved are. These are

- background.c
- input.c
- lensing.c
- nonlinear.c
- output.c
- perturbations.c
- primordial.c

- spectra.c
- thermodynamics.c
- transfer.c

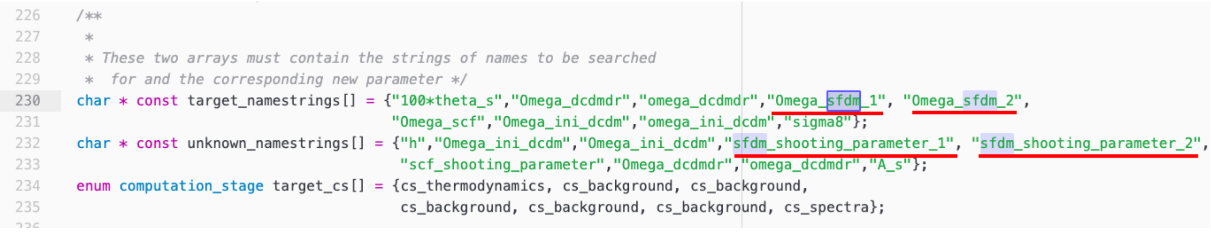
Each of these corresponds to a C library with a .h extension found in the class_public/include directory. These libraries are called in the header of their respective .c file.

Appendix C

input.c

Let's start with the input.c file with which CLASS extracts the parameter values from the input file and calculates the initial conditions to solve the differential equations. Throughout this and all files, the corresponding variables will be labeled with sfdm_1 or sfdm_2 depending on each case.

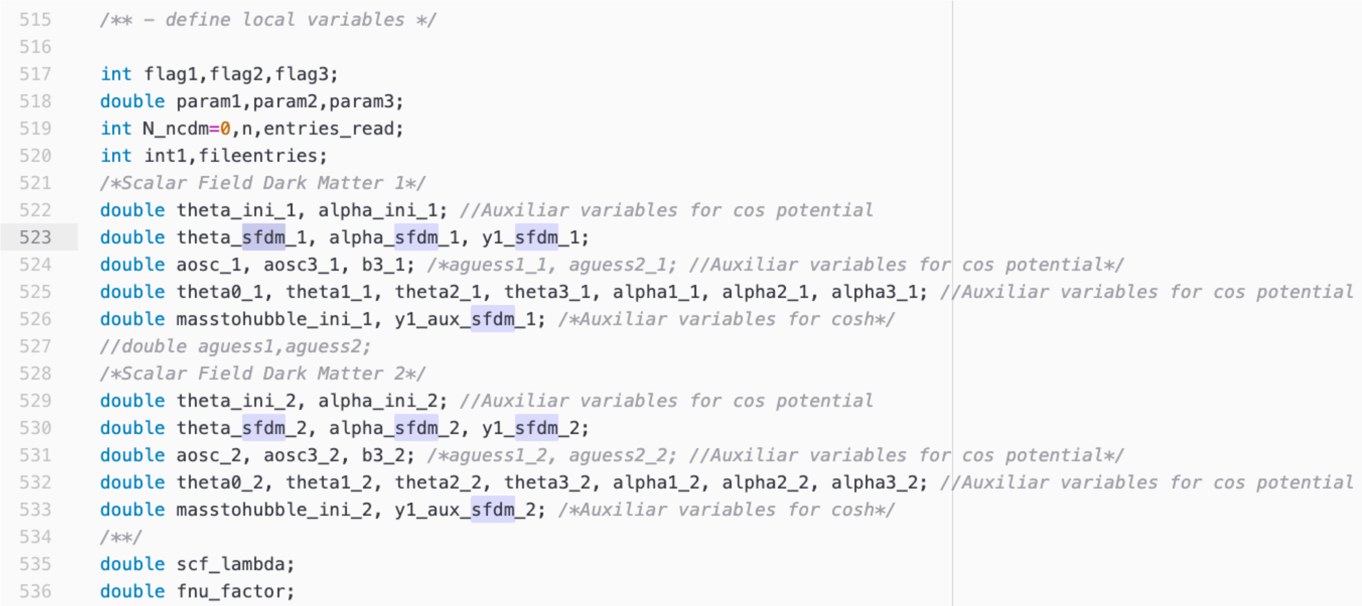
To find the initial conditions we copy the default CLASS method for handling scalar fields, this is called the shooting method. We need a target variable, that is, the value to be reached, and a method's own variable. In our case we will use the current contribution of the scalar fields to the content of the Universe, that is, the Omegas. See Figure C.1.



```
/**
 *
 * These two arrays must contain the strings of names to be searched
 *  for and the corresponding new parameter */
char * const target_namestrings[] = {"100*theta_s","Omega_dcdmdr","omega_dcdmdr","Omega_sfdm_1", "Omega_sfdm_2",
                                     "Omega_scf","Omega_ini_dcdm","omega_ini_dcdm","sigma8"};
char * const unknown_namestrings[] = {"h","Omega_ini_dcdm","Omega_ini_dcdm","sfdm_shooting_parameter_1", "sfdm_shooting_parameter_2",
                                      "scf_shooting_parameter","Omega_dcdmdr","omega_dcdmdr","A_s"};
enum computation_stage target_cs[] = {cs_thermodynamics, cs_background, cs_background,
                                      cs_background, cs_background, cs_background, cs_spectra};
```

Figure C.1: We haven't defined any variables yet. We're just giving the code the name of the parameters to use for the shooting method. In this case the parameters to reach are the Omegas and the parameters to be used for adjustments are the shooting parameters.

Next we define the local variables, where alpha_sfdm is related to Ω_ϕ as we will see in background.c See Figure C.2.



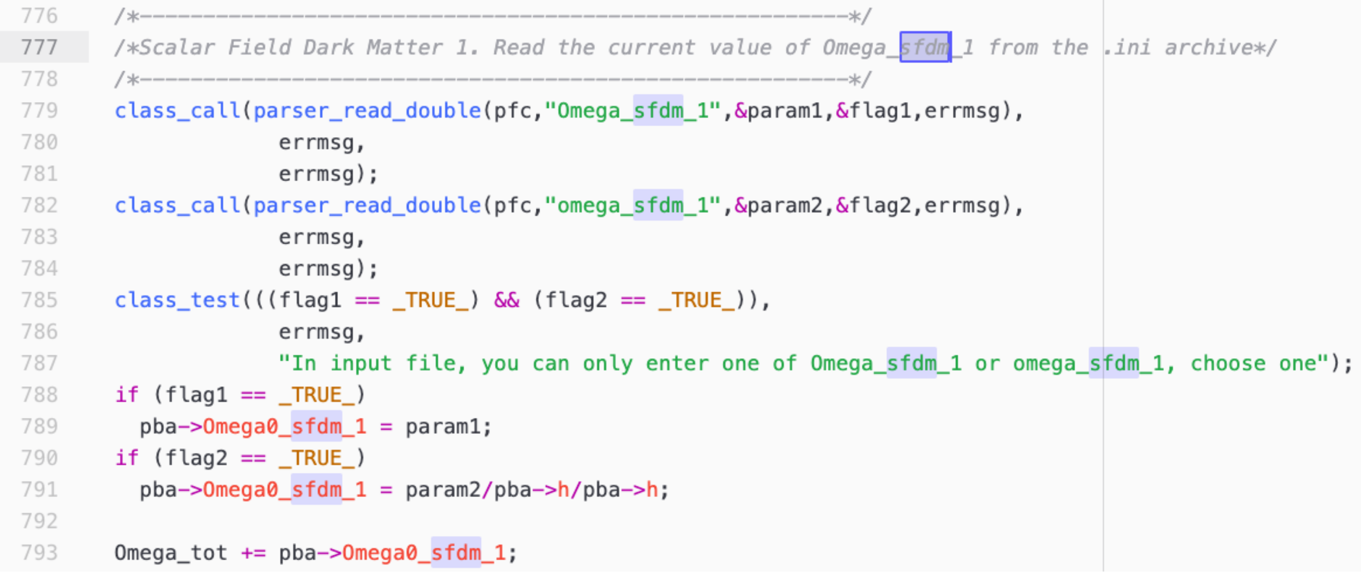
```c
/** - define local variables */

int flag1,flag2,flag3;
double param1,param2,param3;
int N_ncdm=0,n,entries_read;
int int1,fileentries;
/*Scalar Field Dark Matter 1*/
double theta_ini_1, alpha_ini_1; //Auxiliar variables for cos potential
double theta_sfdm_1, alpha_sfdm_1, y1_sfdm_1;
double aosc_1, aosc3_1, b3_1; /*aguess1_1, aguess2_1; //Auxiliar variables for cos potential*/
double theta0_1, theta1_1, theta2_1, theta3_1, alpha1_1, alpha2_1, alpha3_1; //Auxiliar variables for cos potential
double masstohubble_ini_1, y1_aux_sfdm_1; /*Auxiliar variables for cosh*/
//double aguess1,aguess2;
/*Scalar Field Dark Matter 2*/
double theta_ini_2, alpha_ini_2; //Auxiliar variables for cos potential
double theta_sfdm_2, alpha_sfdm_2, y1_sfdm_2;
double aosc_2, aosc3_2, b3_2; /*aguess1_2, aguess2_2; //Auxiliar variables for cos potential*/
double theta0_2, theta1_2, theta2_2, theta3_2, alpha1_2, alpha2_2, alpha3_2; //Auxiliar variables for cos potential
double masstohubble_ini_2, y1_aux_sfdm_2; /*Auxiliar variables for cosh*/
/**/
double scf_lambda;
double fnu_factor;
```

Figure C.2: Defining local variables.

We then tell CLASS how to extract the data from the input files. It can be read in different ways, in this case we will only use two: Ω_ϕ and ω_{phi}. In the class_call function we see in green and in quotes how CLASS will look for the parameter in the input file. Then, we see in which variable the value will be stored, for example &param1 or &param2. The entry that follows, &flag1 or &flag2, are boolean variables used to test and determine how the extracted value is going to be transformed. In class_test the chosen way to extract the parameter is determined depending on which flag is set on. If both are, it will return an error message. Then the value is added to the total of the components of the Universe. See Figure C.3.

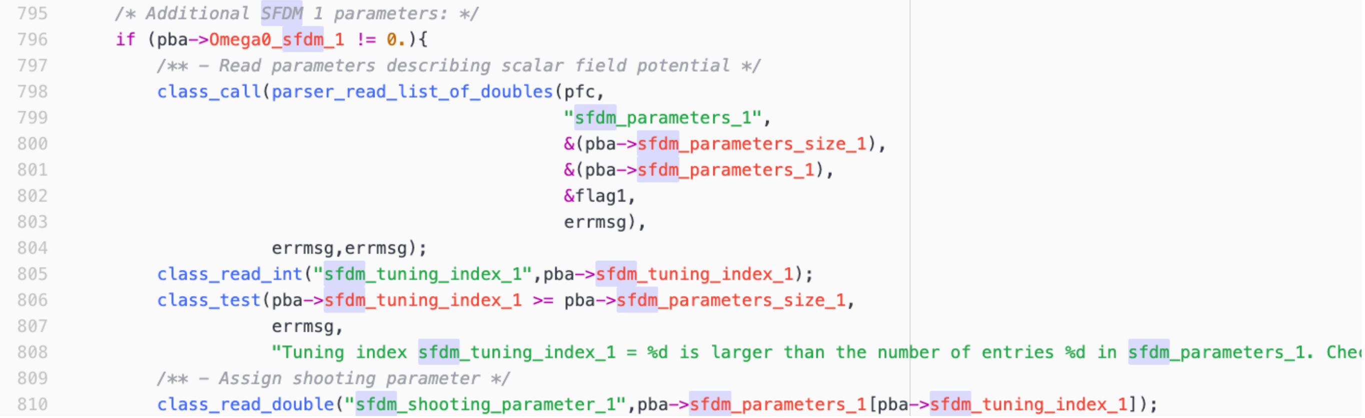
If the contribution of the scalar field is different from zero, CLASS will read the other parameters of the scalar field: the mass, the decay constant and the tuning_index that will be used for the shooting method. See Figure C.4

The initial conditions for the differential equations of the scalar field are now obtained. We treat the three potentials by means of a nested "if" using the value of the lambda



```
/*----------------------------------------------------------*/
/*Scalar Field Dark Matter 1. Read the current value of Omega_sfdm_1 from the .ini archive*/
/*----------------------------------------------------------*/
class_call(parser_read_double(pfc,"Omega_sfdm_1",&param1,&flag1,errmsg),
             errmsg,
             errmsg);
class_call(parser_read_double(pfc,"omega_sfdm_1",&param2,&flag2,errmsg),
             errmsg,
             errmsg);
class_test(((flag1 == _TRUE_) && (flag2 == _TRUE_)),
             errmsg,
             "In input file, you can only enter one of Omega_sfdm_1 or omega_sfdm_1, choose one");
if (flag1 == _TRUE_)
  pba->Omega0_sfdm_1 = param1;
if (flag2 == _TRUE_)
  pba->Omega0_sfdm_1 = param2/pba->h/pba->h;

Omega_tot += pba->Omega0_sfdm_1;
```

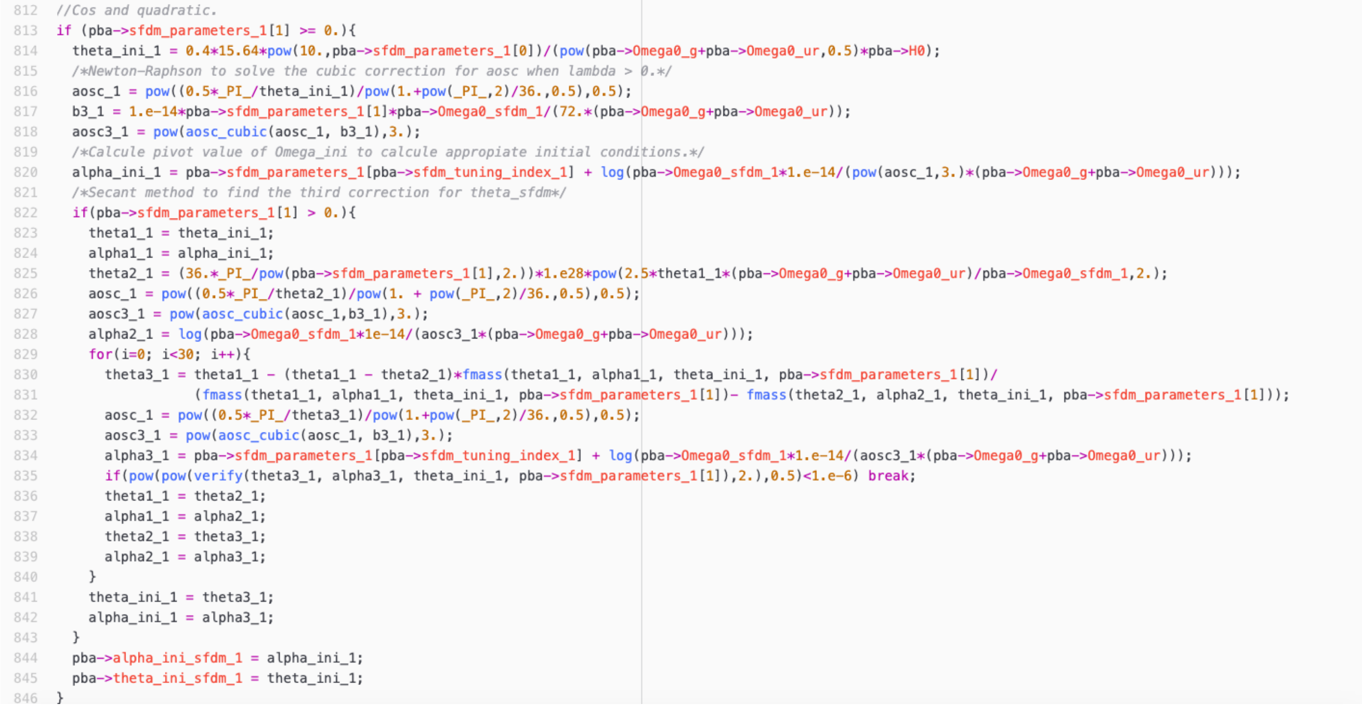
Figure C.3: Extracting the values parameters from input files.



```
/* Additional SFDM 1 parameters: */
if (pba->Omega0_sfdm_1 != 0.){
    /** - Read parameters describing scalar field potential */
    class_call(parser_read_list_of_doubles(pfc,
                                           "sfdm_parameters_1",
                                           &(pba->sfdm_parameters_size_1),
                                           &(pba->sfdm_parameters_1),
                                           &flag1,
                                           errmsg),
               errmsg,errmsg);
    class_read_int("sfdm_tuning_index_1",pba->sfdm_tuning_index_1);
    class_test(pba->sfdm_tuning_index_1 >= pba->sfdm_parameters_size_1,
               errmsg,
               "Tuning index sfdm_tuning_index_1 = %d is larger than the number of entries %d in sfdm_parameters_1. Che
    /** - Assign shooting parameter */
    class_read_double("sfdm_shooting_parameter_1",pba->sfdm_parameters_1[pba->sfdm_tuning_index_1]);
```

Figure C.4: Reading the scalar field parameters from the input file.

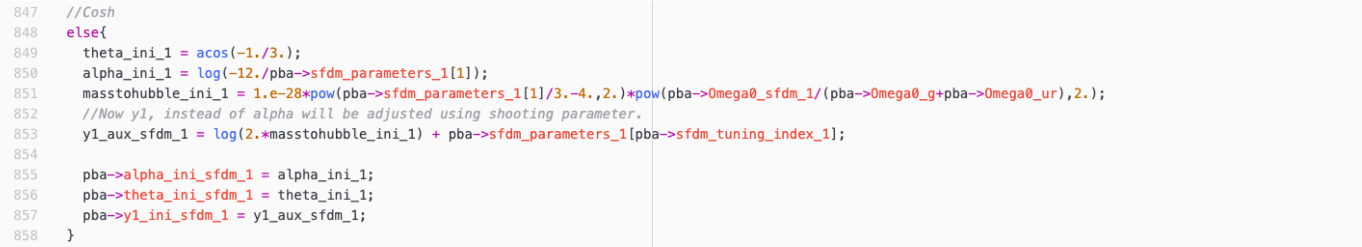
parameter as a flag. We begin by considering the case greater than or equal to zero corresponding to cos and quadratic, respectively. First, the initial conditions of the quadratic case are obtained. Inside this "if" we have another "if" to deal with the potential type cos whose initial conditions need those of the quadratic case. For this we implement the Newton-Raphson method as it is done in the reference [96]. See Figure C.5. Notice that in the code (using the atom editor), the variables are colored red. Next to them there is an arrow and the word pba which means that it is a variable that belongs to the background. This is one of the advantages of classes in C, these variables are defined in another file, namely background.c, and we can use them in other parts of the code without defining them.



```
//Cos and quadratic.
if (pba->sfdm_parameters_1[1] >= 0.){
  theta_ini_1 = 0.4*15.64*pow(10.,pba->sfdm_parameters_1[0])/(pow(pba->Omega0_g+pba->Omega0_ur,0.5)*pba->H0);
  /*Newton-Raphson to solve the cubic correction for aosc when lambda > 0.*/
  aosc_1 = pow((0.5*_PI_/theta_ini_1)/pow(1.+pow(_PI_,2)/36.,0.5),0.5);
  b3_1 = 1.e-14*pba->sfdm_parameters_1[1]*pba->Omega0_sfdm_1/(72.*(pba->Omega0_g+pba->Omega0_ur));
  aosc3_1 = pow(aosc_cubic(aosc_1, b3_1),3.);
  /*Calcule pivot value of Omega_ini to calcule appropiate initial conditions.*/
  alpha_ini_1 = pba->sfdm_parameters_1[pba->sfdm_tuning_index_1] + log(pba->Omega0_sfdm_1*1.e-14/(pow(aosc_1,3.)*(pba->Omega0_g+pba->Omega0_ur)));
  /*Secant method to find the third correction for theta_sfdm*/
  if(pba->sfdm_parameters_1[1] > 0.){
    theta1_1 = theta_ini_1;
    alpha1_1 = alpha_ini_1;
    theta2_1 = (36.*_PI_/pow(pba->sfdm_parameters_1[1],2.))*1.e28*pow(2.5*theta1_1*(pba->Omega0_g+pba->Omega0_ur)/pba->Omega0_sfdm_1,2.);
    aosc_1 = pow((0.5*_PI_/theta2_1)/pow(1. + pow(_PI_,2)/36.,0.5),0.5);
    aosc3_1 = pow(aosc_cubic(aosc_1,b3_1),3.);
    alpha2_1 = log(pba->Omega0_sfdm_1*1e-14/(aosc3_1*(pba->Omega0_g+pba->Omega0_ur)));
    for(i=0; i<30; i++){
      theta3_1 = theta1_1 - (theta1_1 - theta2_1)*fmass(theta1_1, alpha1_1, theta_ini_1, pba->sfdm_parameters_1[1])/
                 (fmass(theta1_1, alpha1_1, theta_ini_1, pba->sfdm_parameters_1[1])- fmass(theta2_1, alpha2_1, theta_ini_1, pba->sfdm_parameters_1[1]));
      aosc_1 = pow((0.5*_PI_/theta3_1)/pow(1.+pow(_PI_,2)/36.,0.5),0.5);
      aosc3_1 = pow(aosc_cubic(aosc_1, b3_1),3.);
      alpha3_1 = pba->sfdm_parameters_1[pba->sfdm_tuning_index_1] + log(pba->Omega0_sfdm_1*1.e-14/(aosc3_1*(pba->Omega0_g+pba->Omega0_ur)));
      if(pow(pow(verify(theta3_1, alpha3_1, theta_ini_1, pba->sfdm_parameters_1[1]),2.),0.5)<1.e-6) break;
      theta1_1 = theta2_1;
      alpha1_1 = alpha2_1;
      theta2_1 = theta3_1;
      alpha2_1 = alpha3_1;
    }
    theta_ini_1 = theta3_1;
    alpha_ini_1 = alpha3_1;
  }
  pba->alpha_ini_sfdm_1 = alpha_ini_1;
  pba->theta_ini_sfdm_1 = theta_ini_1;
}
```

Figure C.5: Obtaining the initial conditions for the quadratic and cos potentials.

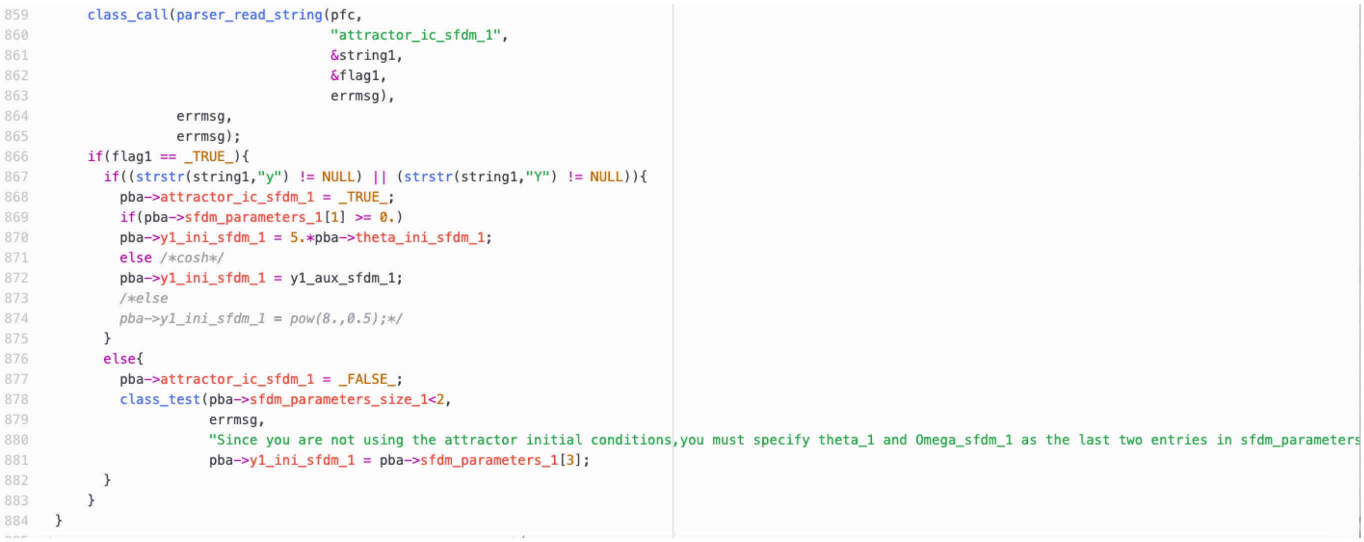
In the case of the cosh, outside of the if, we put the initial conditions in an else. See Figure C.6.



```
 //Cosh
 else{
   theta_ini_1 = acos(-1./3.);
   alpha_ini_1 = log(-12./pba->sfdm_parameters_1[1]);
   masstohubble_ini_1 = 1.e-28*pow(pba->sfdm_parameters_1[1]/3.-4.,2.)*pow(pba->Omega0_sfdm_1/(pba->Omega0_g+pba->Omega0_ur),2.);
   //Now y1, instead of alpha will be adjusted using shooting parameter.
   y1_aux_sfdm_1 = log(2.*masstohubble_ini_1) + pba->sfdm_parameters_1[pba->sfdm_tuning_index_1];

   pba->alpha_ini_sfdm_1 = alpha_ini_1;
   pba->theta_ini_sfdm_1 = theta_ini_1;
   pba->y1_ini_sfdm_1 = y1_aux_sfdm_1;
 }
```

Figure C.6: Obtaining the initial conditions for the cosh potentials.

The next piece of information that is extracted from the input file is whether or not we are going to use the scalar field's attractor condition. This is the relationship of the initial values that the parameters have between them and that were presented in section 3.2. The first "if" checks if they are going to be used and inside it there is an "if" and an else for the cases of cos and quadratic, and cosh, respectively. Outside of the main "if" is an else, here are the actions that the code will do when the attractor is not used. It takes as initial condition the last entry of the array of parameters of the scalar field. See Figure C.7 We recommend always using the attractor condition. Otherwise, finding the initial conditions would have to be done manually and could become very complicated.



```
    class_call(parser_read_string(pfc,
                                  "attractor_ic_sfdm_1",
                                  &string1,
                                  &flag1,
                                  errmsg),
               errmsg,
               errmsg);
    if(flag1 == _TRUE_){
      if((strstr(string1,"y") != NULL) || (strstr(string1,"Y") != NULL)){
        pba->attractor_ic_sfdm_1 = _TRUE_;
        if(pba->sfdm_parameters_1[1] >= 0.)
        pba->y1_ini_sfdm_1 = 5.*pba->theta_ini_sfdm_1;
        else /*cosh*/
        pba->y1_ini_sfdm_1 = y1_aux_sfdm_1;
        /*else
        pba->y1_ini_sfdm_1 = pow(8.,0.5);*/
      }
      else{
        pba->attractor_ic_sfdm_1 = _FALSE_;
        class_test(pba->sfdm_parameters_size_1<2,
                   errmsg,
                   "Since you are not using the attractor initial conditions,you must specify theta_1 and Omega_sfdm_1 as the last two entries in sfdm_parameters
                   pba->y1_ini_sfdm_1 = pba->sfdm_parameters_1[3];
      }
    }
}
```

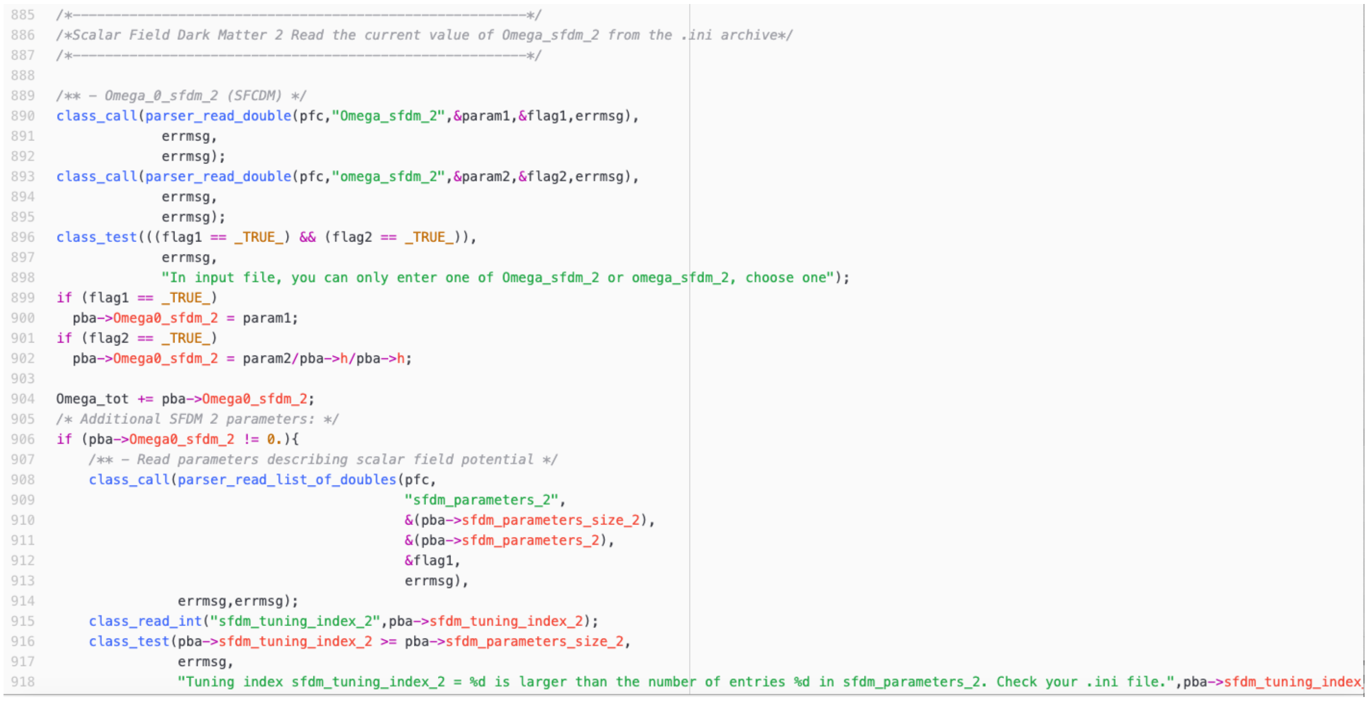
Figure C.7: Determining if the scalar field attractor condition will be used or not.

We do the same for the second field taking care to change the number 1 to 2. See Figures C.8, C.9 and C.10.

Now we are going to set the default values that the scalar fields variables will take. These values are what CLASS will assign if it doesn't find scalar fields information in the input file, the .ini file. See Figure C.11.

The lines of Figures C.12, C.13 and C.14 are adapted from those of the scalar field as dark energy.

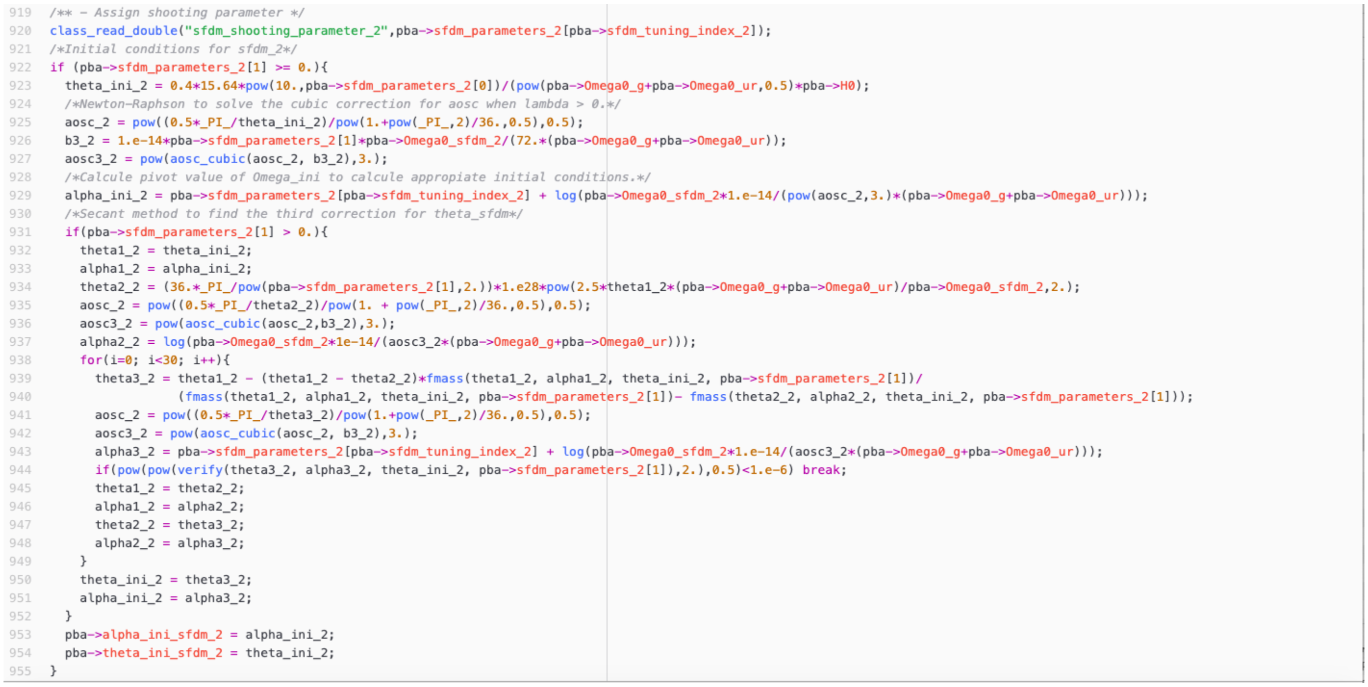
At the end, we define the auxiliary functions for the Newton-Raphson method. See



```c
/*---------------------------------------------------------*/
/*Scalar Field Dark Matter 2 Read the current value of Omega_sfdm_2 from the .ini archive*/
/*---------------------------------------------------------*/

/** - Omega_0_sfdm_2 (SFCDM) */
class_call(parser_read_double(pfc,"Omega_sfdm_2",&param1,&flag1,errmsg),
             errmsg,
             errmsg);
class_call(parser_read_double(pfc,"omega_sfdm_2",&param2,&flag2,errmsg),
             errmsg,
             errmsg);
class_test(((flag1 == _TRUE_) && (flag2 == _TRUE_)),
             errmsg,
             "In input file, you can only enter one of Omega_sfdm_2 or omega_sfdm_2, choose one");
if (flag1 == _TRUE_)
  pba->Omega0_sfdm_2 = param1;
if (flag2 == _TRUE_)
  pba->Omega0_sfdm_2 = param2/pba->h/pba->h;

Omega_tot += pba->Omega0_sfdm_2;
/* Additional SFDM 2 parameters: */
if (pba->Omega0_sfdm_2 != 0.){
    /** - Read parameters describing scalar field potential */
    class_call(parser_read_list_of_doubles(pfc,
                                            "sfdm_parameters_2",
                                            &(pba->sfdm_parameters_size_2),
                                            &(pba->sfdm_parameters_2),
                                            &flag1,
                                            errmsg),
               errmsg,errmsg);
    class_read_int("sfdm_tuning_index_2",pba->sfdm_tuning_index_2);
    class_test(pba->sfdm_tuning_index_2 >= pba->sfdm_parameters_size_2,
               errmsg,
               "Tuning index sfdm_tuning_index_2 = %d is larger than the number of entries %d in sfdm_parameters_2. Check your .ini file.",pba->sfdm_tuning_index
```

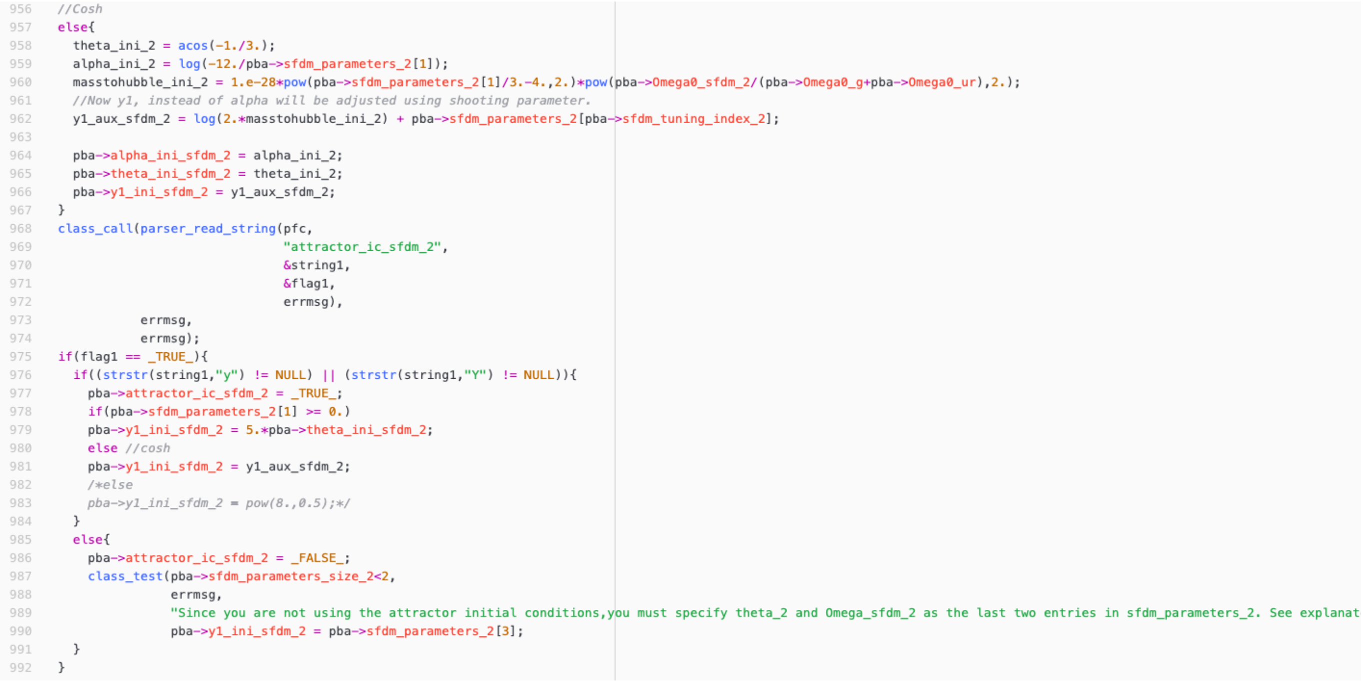
Figure C.8: Implementing the second field.

Figure C.15.



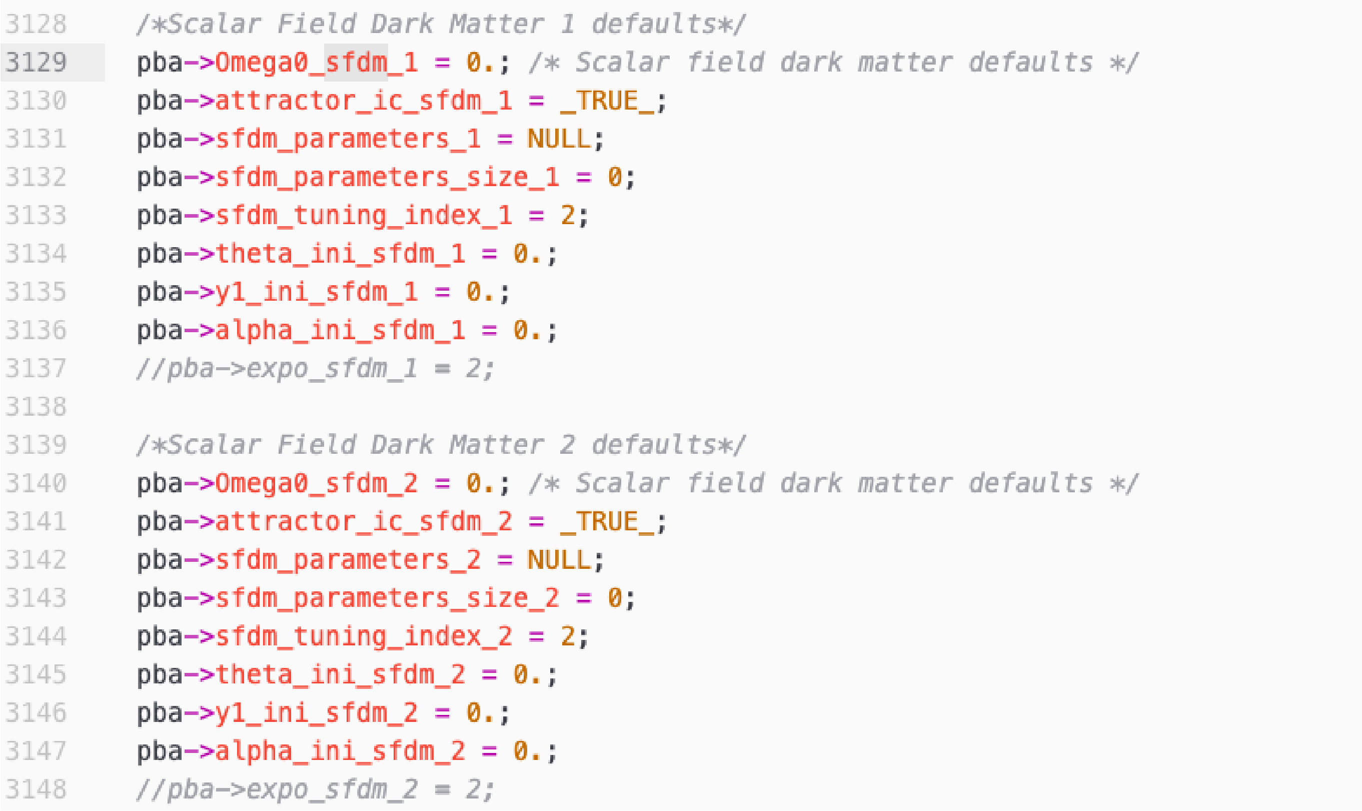
```c
/** - Assign shooting parameter */
class_read_double("sfdm_shooting_parameter_2",pba->sfdm_parameters_2[pba->sfdm_tuning_index_2]);
/*Initial conditions for sfdm_2*/
if (pba->sfdm_parameters_2[1] >= 0.){
  theta_ini_2 = 0.4*15.64*pow(10.,pba->sfdm_parameters_2[0])/(pow(pba->Omega0_g+pba->Omega0_ur,0.5)*pba->H0);
  /*Newton-Raphson to solve the cubic correction for aosc when lambda > 0.*/
  aosc_2 = pow((0.5*_PI_/theta_ini_2)/pow(1.+pow(_PI_,2)/36.,0.5),0.5);
  b3_2 = 1.e-14*pba->sfdm_parameters_2[1]*pba->Omega0_sfdm_2/(72.*(pba->Omega0_g+pba->Omega0_ur));
  aosc3_2 = pow(aosc_cubic(aosc_2, b3_2),3.);
  /*Calcule pivot value of Omega_ini to calcule appropiate initial conditions.*/
  alpha_ini_2 = pba->sfdm_parameters_2[pba->sfdm_tuning_index_2] + log(pba->Omega0_sfdm_2*1.e-14/(pow(aosc_2,3.)*(pba->Omega0_g+pba->Omega0_ur)));
  /*Secant method to find the third correction for theta_sfdm*/
  if(pba->sfdm_parameters_2[1] > 0.){
    theta1_2 = theta_ini_2;
    alpha1_2 = alpha_ini_2;
    theta2_2 = (36.*_PI_/pow(pba->sfdm_parameters_2[1],2.))*1.e28*pow(2.5*theta1_2*(pba->Omega0_g+pba->Omega0_ur)/pba->Omega0_sfdm_2,2.);
    aosc_2 = pow((0.5*_PI_/theta2_2)/pow(1. + pow(_PI_,2)/36.,0.5),0.5);
    aosc3_2 = pow(aosc_cubic(aosc_2,b3_2),3.);
    alpha2_2 = log(pba->Omega0_sfdm_2*1e-14/(aosc3_2*(pba->Omega0_g+pba->Omega0_ur)));
    for(i=0; i<30; i++){
      theta3_2 = theta1_2 - (theta1_2 - theta2_2)*fmass(theta1_2, alpha1_2, theta_ini_2, pba->sfdm_parameters_2[1])/
                 (fmass(theta1_2, alpha1_2, theta_ini_2, pba->sfdm_parameters_2[1])- fmass(theta2_2, alpha2_2, theta_ini_2, pba->sfdm_parameters_2[1]));
      aosc_2 = pow((0.5*_PI_/theta3_2)/pow(1.+pow(_PI_,2)/36.,0.5),0.5);
      aosc3_2 = pow(aosc_cubic(aosc_2, b3_2),3.);
      alpha3_2 = pba->sfdm_parameters_2[pba->sfdm_tuning_index_2] + log(pba->Omega0_sfdm_2*1.e-14/(aosc3_2*(pba->Omega0_g+pba->Omega0_ur)));
      if(pow(pow(verify(theta3_2, alpha3_2, theta_ini_2, pba->sfdm_parameters_2[1]),2.),0.5)<1.e-6) break;
      theta1_2 = theta2_2;
      alpha1_2 = alpha2_2;
      theta2_2 = theta3_2;
      alpha2_2 = alpha3_2;
    }
    theta_ini_2 = theta3_2;
    alpha_ini_2 = alpha3_2;
  }
  pba->alpha_ini_sfdm_2 = alpha_ini_2;
  pba->theta_ini_sfdm_2 = theta_ini_2;
}
```

Figure C.9: Implementing the second field.



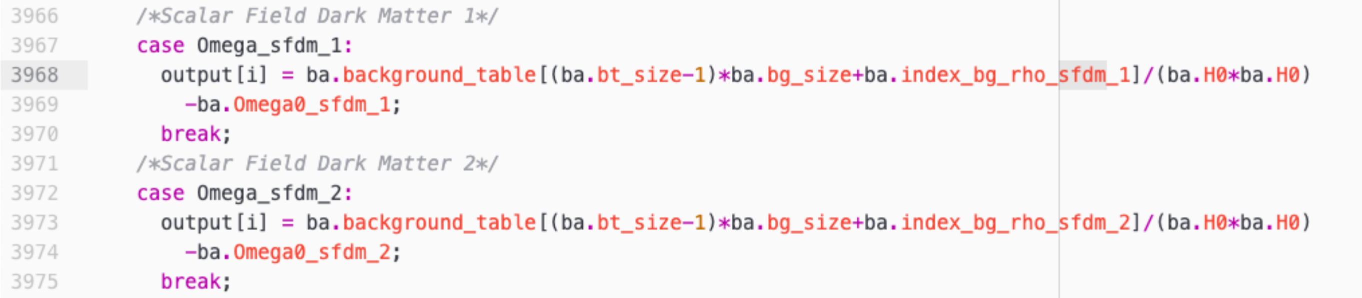
```c
  //Cosh
  else{
    theta_ini_2 = acos(-1./3.);
    alpha_ini_2 = log(-12./pba->sfdm_parameters_2[1]);
    masstohubble_ini_2 = 1.e-28*pow(pba->sfdm_parameters_2[1]/3.-4.,2.)*pow(pba->Omega0_sfdm_2/(pba->Omega0_g+pba->Omega0_ur),2.);
    //Now y1, instead of alpha will be adjusted using shooting parameter.
    y1_aux_sfdm_2 = log(2.*masstohubble_ini_2) + pba->sfdm_parameters_2[pba->sfdm_tuning_index_2];

    pba->alpha_ini_sfdm_2 = alpha_ini_2;
    pba->theta_ini_sfdm_2 = theta_ini_2;
    pba->y1_ini_sfdm_2 = y1_aux_sfdm_2;
  }
  class_call(parser_read_string(pfc,
                                "attractor_ic_sfdm_2",
                                &string1,
                                &flag1,
                                errmsg),
             errmsg,
             errmsg);
  if(flag1 == _TRUE_){
    if((strstr(string1,"y") != NULL) || (strstr(string1,"Y") != NULL)){
      pba->attractor_ic_sfdm_2 = _TRUE_;
      if(pba->sfdm_parameters_2[1] >= 0.)
      pba->y1_ini_sfdm_2 = 5.*pba->theta_ini_sfdm_2;
      else //cosh
      pba->y1_ini_sfdm_2 = y1_aux_sfdm_2;
      /*else
      pba->y1_ini_sfdm_2 = pow(8.,0.5);*/
    }
    else{
      pba->attractor_ic_sfdm_2 = _FALSE_;
      class_test(pba->sfdm_parameters_size_2<2,
                 errmsg,
                 "Since you are not using the attractor initial conditions,you must specify theta_2 and Omega_sfdm_2 as the last two entries in sfdm_parameters_2. See explanat
                 pba->y1_ini_sfdm_2 = pba->sfdm_parameters_2[3];
    }
  }
```

Figure C.10: Implementing the second field.



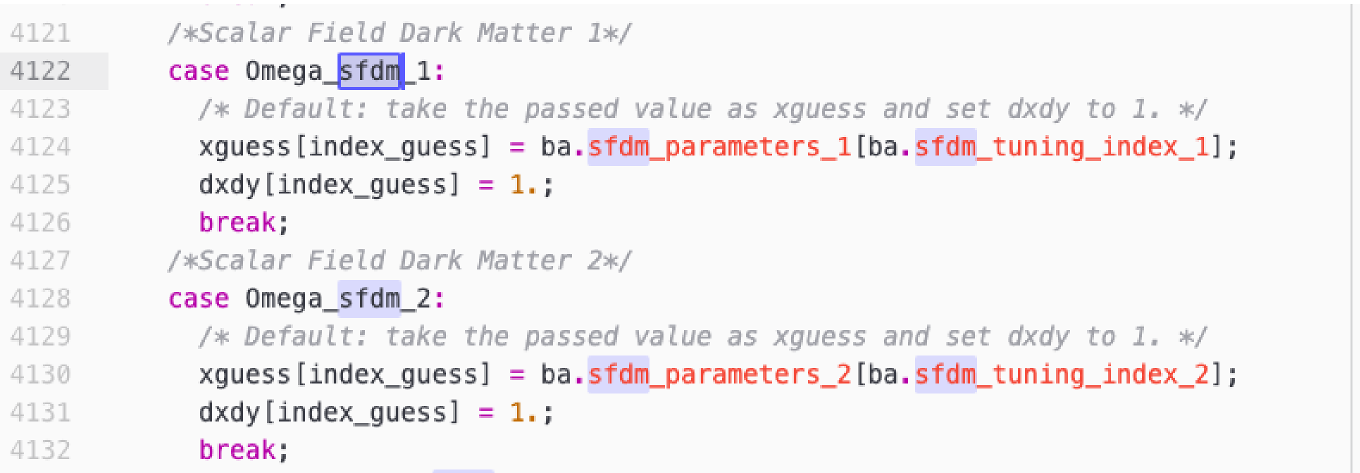
```
/*Scalar Field Dark Matter 1 defaults*/
pba->Omega0_sfdm_1 = 0.; /* Scalar field dark matter defaults */
pba->attractor_ic_sfdm_1 = _TRUE_;
pba->sfdm_parameters_1 = NULL;
pba->sfdm_parameters_size_1 = 0;
pba->sfdm_tuning_index_1 = 2;
pba->theta_ini_sfdm_1 = 0.;
pba->y1_ini_sfdm_1 = 0.;
pba->alpha_ini_sfdm_1 = 0.;
//pba->expo_sfdm_1 = 2;

/*Scalar Field Dark Matter 2 defaults*/
pba->Omega0_sfdm_2 = 0.; /* Scalar field dark matter defaults */
pba->attractor_ic_sfdm_2 = _TRUE_;
pba->sfdm_parameters_2 = NULL;
pba->sfdm_parameters_size_2 = 0;
pba->sfdm_tuning_index_2 = 2;
pba->theta_ini_sfdm_2 = 0.;
pba->y1_ini_sfdm_2 = 0.;
pba->alpha_ini_sfdm_2 = 0.;
//pba->expo_sfdm_2 = 2;
```

Figure C.11: Assigning default values for scalar field variables.



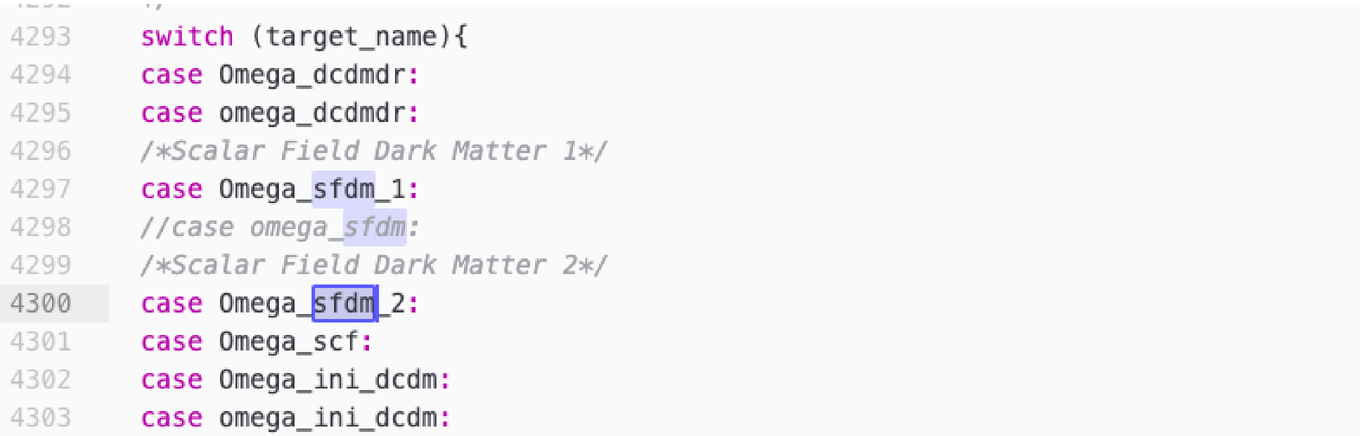
```
/*Scalar Field Dark Matter 1*/
case Omega_sfdm_1:
  output[i] = ba.background_table[(ba.bt_size-1)*ba.bg_size+ba.index_bg_rho_sfdm_1]/(ba.H0*ba.H0)
    -ba.Omega0_sfdm_1;
  break;
/*Scalar Field Dark Matter 2*/
case Omega_sfdm_2:
  output[i] = ba.background_table[(ba.bt_size-1)*ba.bg_size+ba.index_bg_rho_sfdm_2]/(ba.H0*ba.H0)
    -ba.Omega0_sfdm_2;
  break;
```

Figure C.12: Adapting the scalar field lines as dark energy to dark matter.



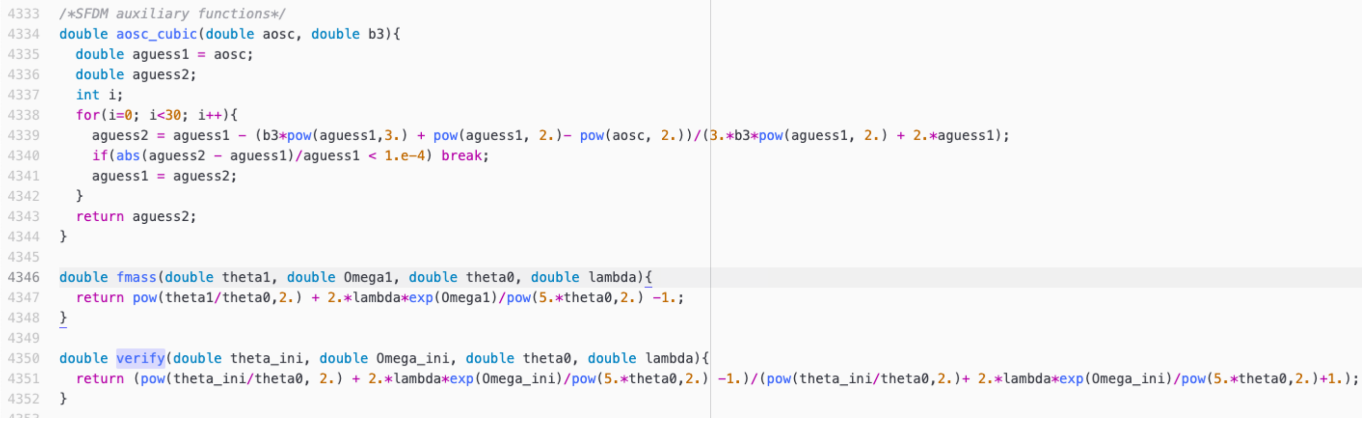
```
/*Scalar Field Dark Matter 1*/
case Omega_sfdm_1:
  /* Default: take the passed value as xguess and set dxdy to 1. */
  xguess[index_guess] = ba.sfdm_parameters_1[ba.sfdm_tuning_index_1];
  dxdy[index_guess] = 1.;
  break;
/*Scalar Field Dark Matter 2*/
case Omega_sfdm_2:
  /* Default: take the passed value as xguess and set dxdy to 1. */
  xguess[index_guess] = ba.sfdm_parameters_2[ba.sfdm_tuning_index_2];
  dxdy[index_guess] = 1.;
  break;
```

Figure C.13: Adapting the scalar field lines as dark energy to dark matter.



```
switch (target_name){
case Omega_dcdmdr:
case omega_dcdmdr:
/*Scalar Field Dark Matter 1*/
case Omega_sfdm_1:
//case omega_sfdm:
/*Scalar Field Dark Matter 2*/
case Omega_sfdm_2:
case Omega_scf:
case Omega_ini_dcdm:
case omega_ini_dcdm:
```

Figure C.14: Adapting the scalar field lines as dark energy to dark matter.



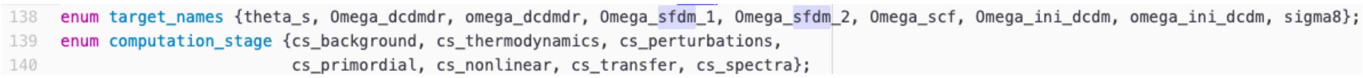
```
/*SFDM auxiliary functions*/
double aosc_cubic(double aosc, double b3){
  double aguess1 = aosc;
  double aguess2;
  int i;
  for(i=0; i<30; i++){
    aguess2 = aguess1 - (b3*pow(aguess1,3.) + pow(aguess1, 2.)- pow(aosc, 2.))/(3.*b3*pow(aguess1, 2.) + 2.*aguess1);
    if(abs(aguess2 - aguess1)/aguess1 < 1.e-4) break;
    aguess1 = aguess2;
  }
  return aguess2;
}

double fmass(double theta1, double Omega1, double theta0, double lambda){
  return pow(theta1/theta0,2.) + 2.*lambda*exp(Omega1)/pow(5.*theta0,2.) -1.;
}

double verify(double theta_ini, double Omega_ini, double theta0, double lambda){
  return (pow(theta_ini/theta0, 2.) + 2.*lambda*exp(Omega_ini)/pow(5.*theta0,2.) -1.)/(pow(theta_ini/theta0,2.)+ 2.*lambda*exp(Omega_ini)/pow(5.*theta0,2.)+1.);
}
```

Figure C.15: Auxiliary functions.

Appendix D

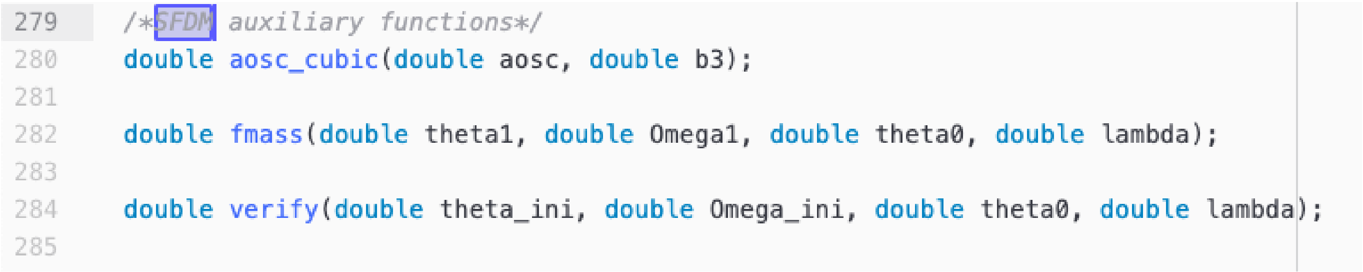
input.h

In this file we are only going to put the name of the parameters that will be the target of the shooting method and the declarations of the auxiliary functions. See Figures D.1 and D.2.



```
enum target_names {theta_s, Omega_dcdmdr, omega_dcdmdr, Omega_sfdm_1, Omega_sfdm_2, Omega_scf, Omega_ini_dcdm, omega_ini_dcdm, sigma8};
enum computation_stage {cs_background, cs_thermodynamics, cs_perturbations,
                        cs_primordial, cs_nonlinear, cs_transfer, cs_spectra};
```

Figure D.1: input.h file modifications.



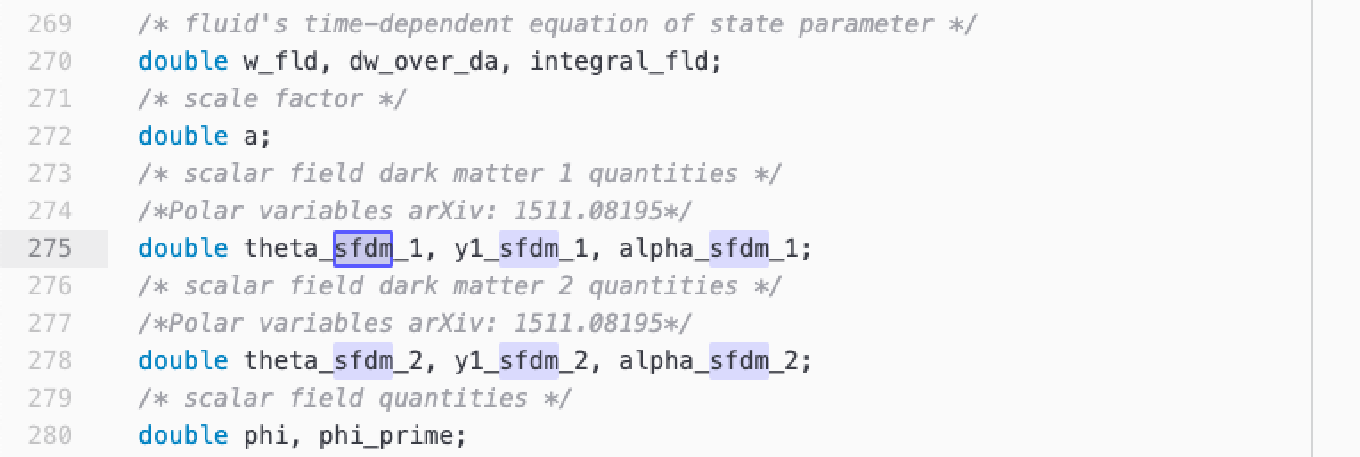
```
/*SFDM auxiliary functions*/
double aosc_cubic(double aosc, double b3);

double fmass(double theta1, double Omega1, double theta0, double lambda);

double verify(double theta_ini, double Omega_ini, double theta0, double lambda);

```

Figure D.2: input.h file modifications.

Appendix E

background.c

In this file we will put the differential equations that we must solve to know the evolution of the scalar fields. We start by defining the local variables for the fields, that is, they will only be used in this file, unlike those accompanied by pba-> or ppt-> that can be used anywhere in the code. See Figure E.1. The term pba means that the variable belongs to the background and ppt to the perturbations. Throughout the appendices we will specify each of these terms when it appears.



```
/* fluid's time-dependent equation of state parameter */
double w_fld, dw_over_da, integral_fld;
/* scale factor */
double a;
/* scalar field dark matter 1 quantities */
/*Polar variables arXiv: 1511.08195*/
double theta_sfdm_1, y1_sfdm_1, alpha_sfdm_1;
/* scalar field dark matter 2 quantities */
/*Polar variables arXiv: 1511.08195*/
double theta_sfdm_2, y1_sfdm_2, alpha_sfdm_2;
/* scalar field quantities */
double phi, phi_prime;
```

Figure E.1: Defining scalar fields local variables.

We will treat the two-field case and the single-field case separately, using nested ifs, because the contribution to the density is different. Having these nested ifs allows the use of the fields to be symmetric.

The if of the two fields is activated when both flags are true, see line 428 in Figure E.2. This happens when a non-zero value is given to the $\Omega_{\phi i}$ in the input file. We are going to store the solutions of the differential equations in the local variables defined before. We will use these local variables to calculate the parameters of the scalar fields. As can be seen on lines 430, 431 and 432 of the Figure E.2, some variables start with index_bi in their name. This means that they are going to be integrated, that is, those that have a differential equation. Then others appear with index_bg , these quantities depend on the solutions of the differential equations, that is, on the index_bi as can be seen in lines 438-442. Then the contributions of the matter and radiation densities of the scalar fields are added to the total densities respectively.

We do the same for the individual fields, paying special attention to the continuity of the number of lines of code. Also note the difference in how the field density is calculated in line 473 of Figure E.3 and line 441 of Figure E.2.

Then, as a precaution, the fields parameters arrays are emptied, see Figure E.5.

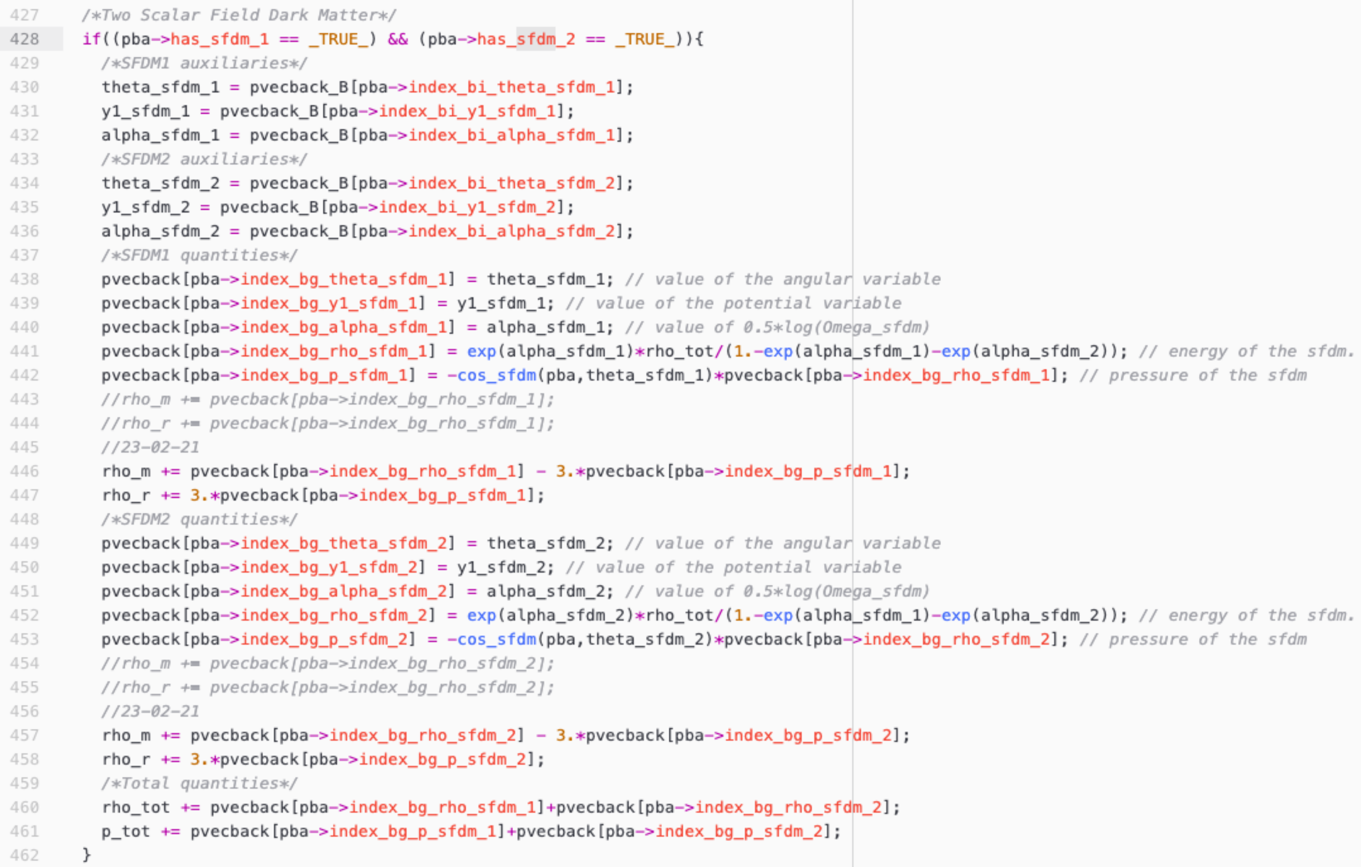
Also as a precaution, the flags that indicate whether or not there are scalar fields are set to false, see Figure E.6. The value will change depending on the information in the input file, see Figure E.7.

Then the index_bg (Figure E.8) and index_bi (Figure E.9) indices used in Figures E.2, E.3 and E.4 are defined.

Next we put the information of the scalar fields that CLASS will show in the terminal when it is executed. These messages will be displayed as long as the field flags are true, see Figures E.10 and E.11.

Now the initial conditions calculated in the input.c file are fixed, see Figure E.12.

Then we put the title of the columns of the fields parameters that will be in the output file, see Figure E.13. On the lines shown in the Figure E.14, we tell CLASS which parameters we want it to save to the output file. We must be careful to put them in the

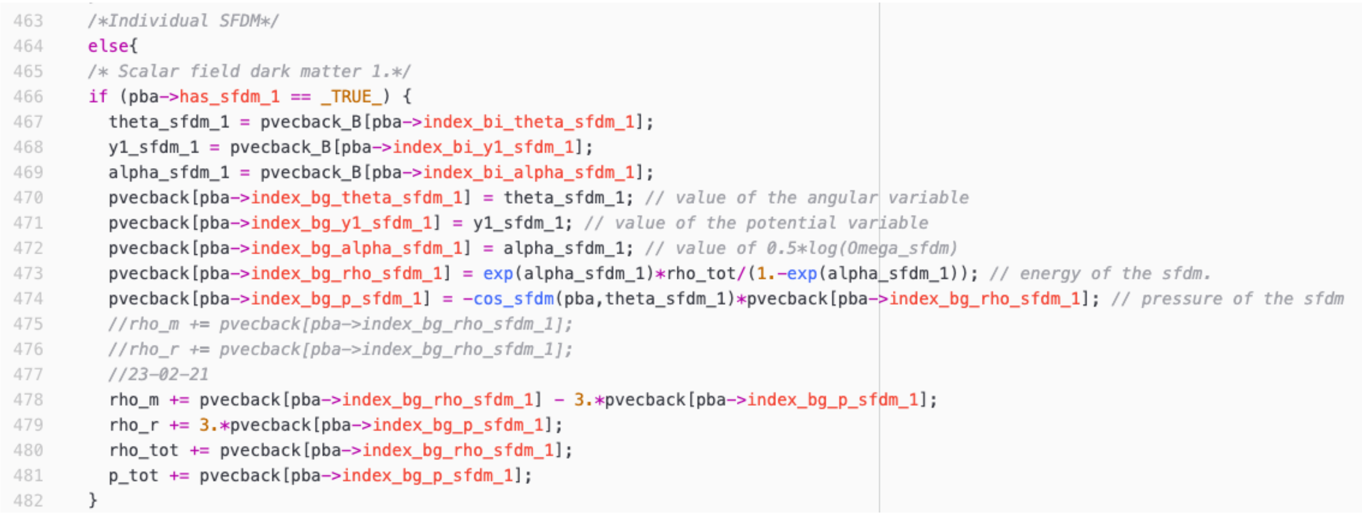


```
/*Two Scalar Field Dark Matter*/
if((pba->has_sfdm_1 == _TRUE_) && (pba->has_sfdm_2 == _TRUE_)){
  /*SFDM1 auxiliaries*/
  theta_sfdm_1 = pvecback_B[pba->index_bi_theta_sfdm_1];
  y1_sfdm_1 = pvecback_B[pba->index_bi_y1_sfdm_1];
  alpha_sfdm_1 = pvecback_B[pba->index_bi_alpha_sfdm_1];
  /*SFDM2 auxiliaries*/
  theta_sfdm_2 = pvecback_B[pba->index_bi_theta_sfdm_2];
  y1_sfdm_2 = pvecback_B[pba->index_bi_y1_sfdm_2];
  alpha_sfdm_2 = pvecback_B[pba->index_bi_alpha_sfdm_2];
  /*SFDM1 quantities*/
  pvecback[pba->index_bg_theta_sfdm_1] = theta_sfdm_1; // value of the angular variable
  pvecback[pba->index_bg_y1_sfdm_1] = y1_sfdm_1; // value of the potential variable
  pvecback[pba->index_bg_alpha_sfdm_1] = alpha_sfdm_1; // value of 0.5*log(Omega_sfdm)
  pvecback[pba->index_bg_rho_sfdm_1] = exp(alpha_sfdm_1)*rho_tot/(1.-exp(alpha_sfdm_1)-exp(alpha_sfdm_2)); // energy of the sfdm.
  pvecback[pba->index_bg_p_sfdm_1] = -cos_sfdm(pba,theta_sfdm_1)*pvecback[pba->index_bg_rho_sfdm_1]; // pressure of the sfdm
  //rho_m += pvecback[pba->index_bg_rho_sfdm_1];
  //rho_r += pvecback[pba->index_bg_rho_sfdm_1];
  //23-02-21
  rho_m += pvecback[pba->index_bg_rho_sfdm_1] - 3.*pvecback[pba->index_bg_p_sfdm_1];
  rho_r += 3.*pvecback[pba->index_bg_p_sfdm_1];
  /*SFDM2 quantities*/
  pvecback[pba->index_bg_theta_sfdm_2] = theta_sfdm_2; // value of the angular variable
  pvecback[pba->index_bg_y1_sfdm_2] = y1_sfdm_2; // value of the potential variable
  pvecback[pba->index_bg_alpha_sfdm_2] = alpha_sfdm_2; // value of 0.5*log(Omega_sfdm)
  pvecback[pba->index_bg_rho_sfdm_2] = exp(alpha_sfdm_2)*rho_tot/(1.-exp(alpha_sfdm_1)-exp(alpha_sfdm_2)); // energy of the sfdm.
  pvecback[pba->index_bg_p_sfdm_2] = -cos_sfdm(pba,theta_sfdm_2)*pvecback[pba->index_bg_rho_sfdm_2]; // pressure of the sfdm
  //rho_m += pvecback[pba->index_bg_rho_sfdm_2];
  //rho_r += pvecback[pba->index_bg_rho_sfdm_2];
  //23-02-21
  rho_m += pvecback[pba->index_bg_rho_sfdm_2] - 3.*pvecback[pba->index_bg_p_sfdm_2];
  rho_r += 3.*pvecback[pba->index_bg_p_sfdm_2];
  /*Total quantities*/
  rho_tot += pvecback[pba->index_bg_rho_sfdm_1]+pvecback[pba->index_bg_rho_sfdm_2];
  p_tot += pvecback[pba->index_bg_p_sfdm_1]+pvecback[pba->index_bg_p_sfdm_2];
}
```

Figure E.2: Assigning scalar fields variables.

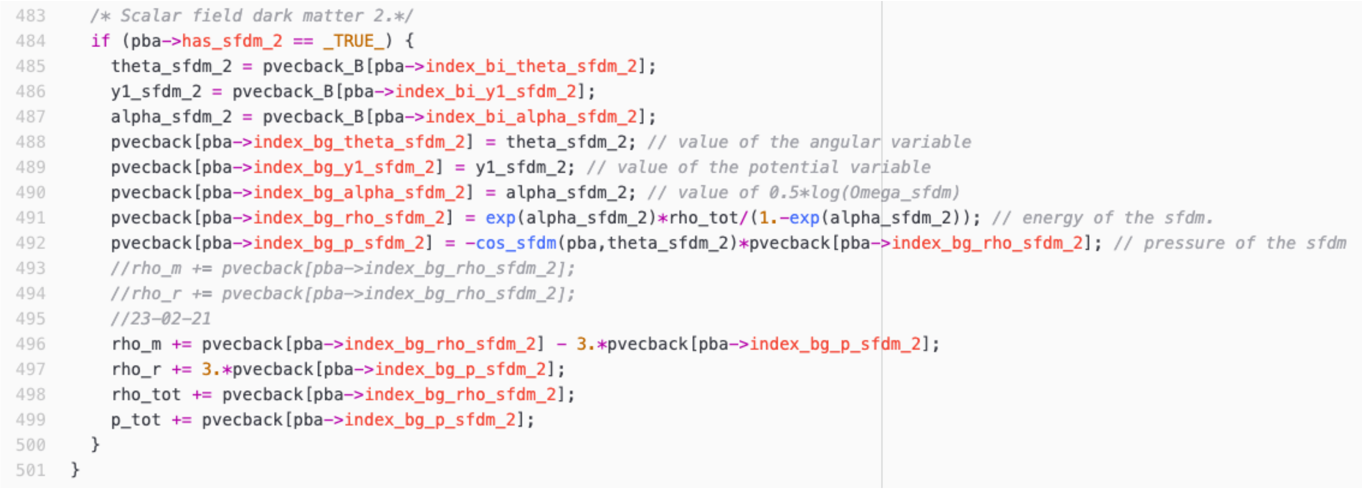
same order as the column titles.

Then we put the differential equations, see Figures E.15 and E.16. Finally, we define the helper functions that we have used in writing the differential equations, see Figure E.17.



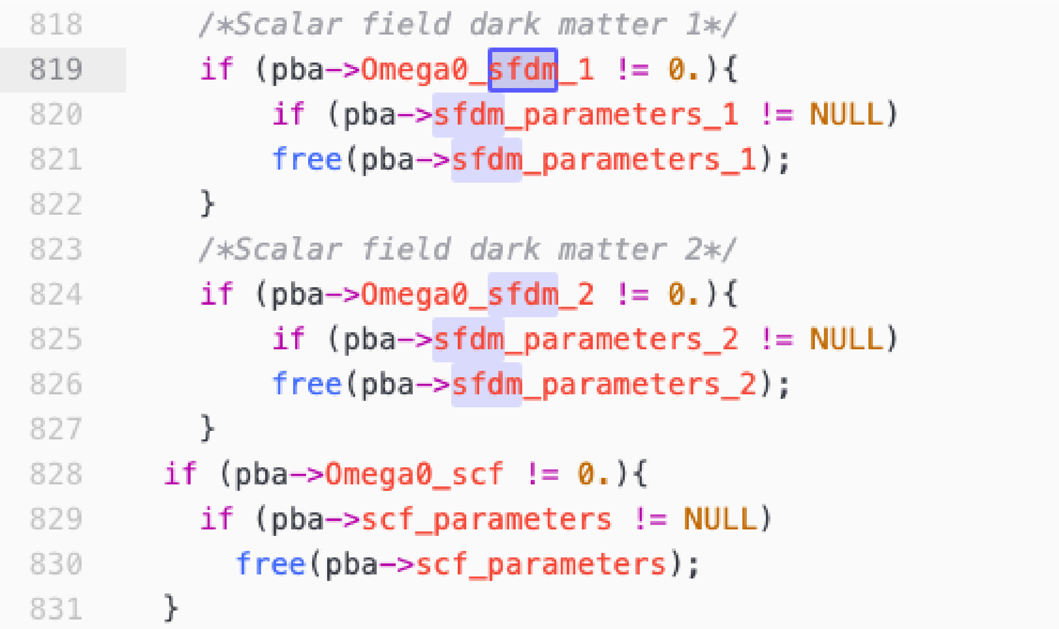
```c
    /*Individual SFDM*/
    else{
    /* Scalar field dark matter 1.*/
    if (pba->has_sfdm_1 == _TRUE_) {
      theta_sfdm_1 = pvecback_B[pba->index_bi_theta_sfdm_1];
      y1_sfdm_1 = pvecback_B[pba->index_bi_y1_sfdm_1];
      alpha_sfdm_1 = pvecback_B[pba->index_bi_alpha_sfdm_1];
      pvecback[pba->index_bg_theta_sfdm_1] = theta_sfdm_1; // value of the angular variable
      pvecback[pba->index_bg_y1_sfdm_1] = y1_sfdm_1; // value of the potential variable
      pvecback[pba->index_bg_alpha_sfdm_1] = alpha_sfdm_1; // value of 0.5*log(Omega_sfdm)
      pvecback[pba->index_bg_rho_sfdm_1] = exp(alpha_sfdm_1)*rho_tot/(1.-exp(alpha_sfdm_1)); // energy of the sfdm.
      pvecback[pba->index_bg_p_sfdm_1] = -cos_sfdm(pba,theta_sfdm_1)*pvecback[pba->index_bg_rho_sfdm_1]; // pressure of the sfdm
      //rho_m += pvecback[pba->index_bg_rho_sfdm_1];
      //rho_r += pvecback[pba->index_bg_rho_sfdm_1];
      //23-02-21
      rho_m += pvecback[pba->index_bg_rho_sfdm_1] - 3.*pvecback[pba->index_bg_p_sfdm_1];
      rho_r += 3.*pvecback[pba->index_bg_p_sfdm_1];
      rho_tot += pvecback[pba->index_bg_rho_sfdm_1];
      p_tot += pvecback[pba->index_bg_p_sfdm_1];
    }
```

Figure E.3: Assigning scalar field 1 variables.



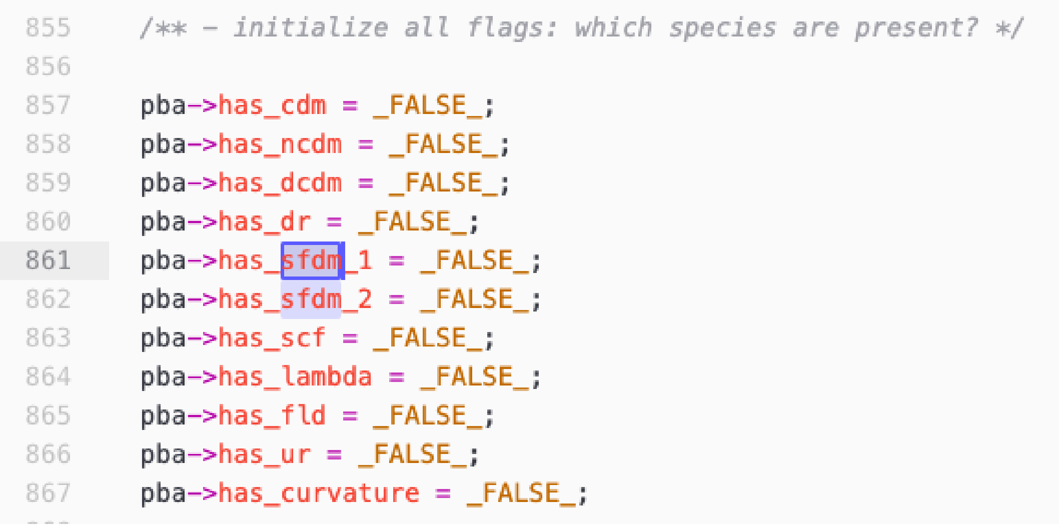
```c
    /* Scalar field dark matter 2.*/
    if (pba->has_sfdm_2 == _TRUE_) {
      theta_sfdm_2 = pvecback_B[pba->index_bi_theta_sfdm_2];
      y1_sfdm_2 = pvecback_B[pba->index_bi_y1_sfdm_2];
      alpha_sfdm_2 = pvecback_B[pba->index_bi_alpha_sfdm_2];
      pvecback[pba->index_bg_theta_sfdm_2] = theta_sfdm_2; // value of the angular variable
      pvecback[pba->index_bg_y1_sfdm_2] = y1_sfdm_2; // value of the potential variable
      pvecback[pba->index_bg_alpha_sfdm_2] = alpha_sfdm_2; // value of 0.5*log(Omega_sfdm)
      pvecback[pba->index_bg_rho_sfdm_2] = exp(alpha_sfdm_2)*rho_tot/(1.-exp(alpha_sfdm_2)); // energy of the sfdm.
      pvecback[pba->index_bg_p_sfdm_2] = -cos_sfdm(pba,theta_sfdm_2)*pvecback[pba->index_bg_rho_sfdm_2]; // pressure of the sfdm
      //rho_m += pvecback[pba->index_bg_rho_sfdm_2];
      //rho_r += pvecback[pba->index_bg_rho_sfdm_2];
      //23-02-21
      rho_m += pvecback[pba->index_bg_rho_sfdm_2] - 3.*pvecback[pba->index_bg_p_sfdm_2];
      rho_r += 3.*pvecback[pba->index_bg_p_sfdm_2];
      rho_tot += pvecback[pba->index_bg_rho_sfdm_2];
      p_tot += pvecback[pba->index_bg_p_sfdm_2];
    }
  }
```

Figure E.4: Assigning scalar field 2 variables.



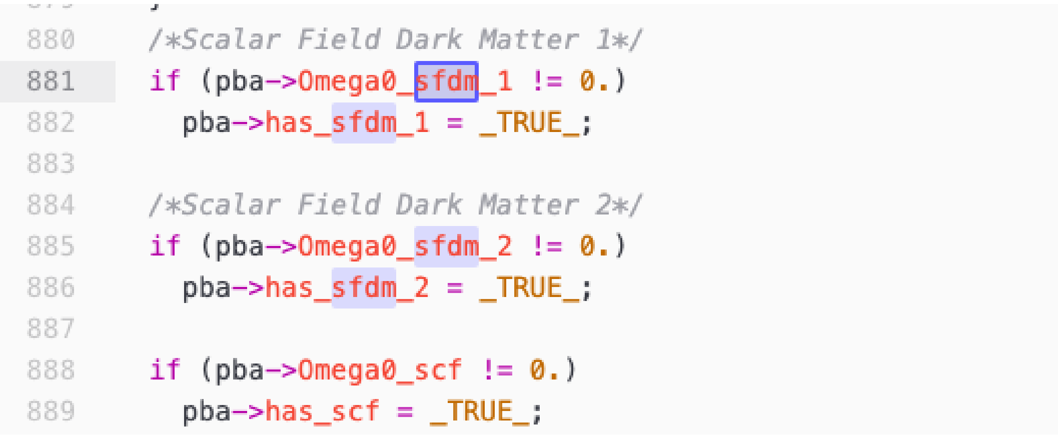
```
  /*Scalar field dark matter 1*/
  if (pba->Omega0_sfdm_1 != 0.){
      if (pba->sfdm_parameters_1 != NULL)
      free(pba->sfdm_parameters_1);
  }
  /*Scalar field dark matter 2*/
  if (pba->Omega0_sfdm_2 != 0.){
      if (pba->sfdm_parameters_2 != NULL)
      free(pba->sfdm_parameters_2);
  }
if (pba->Omega0_scf != 0.){
  if (pba->scf_parameters != NULL)
    free(pba->scf_parameters);
}
```

Figure E.5: The fields parameters arrays are emptied.



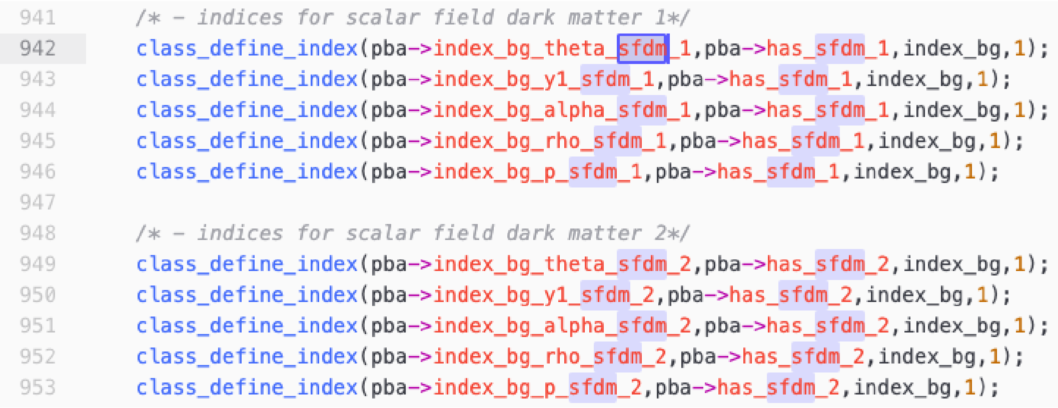
```
/** - initialize all flags: which species are present? */

pba->has_cdm = _FALSE_;
pba->has_ncdm = _FALSE_;
pba->has_dcdm = _FALSE_;
pba->has_dr = _FALSE_;
pba->has_sfdm_1 = _FALSE_;
pba->has_sfdm_2 = _FALSE_;
pba->has_scf = _FALSE_;
pba->has_lambda = _FALSE_;
pba->has_fld = _FALSE_;
pba->has_ur = _FALSE_;
pba->has_curvature = _FALSE_;
```

Figure E.6: Initializing flags.



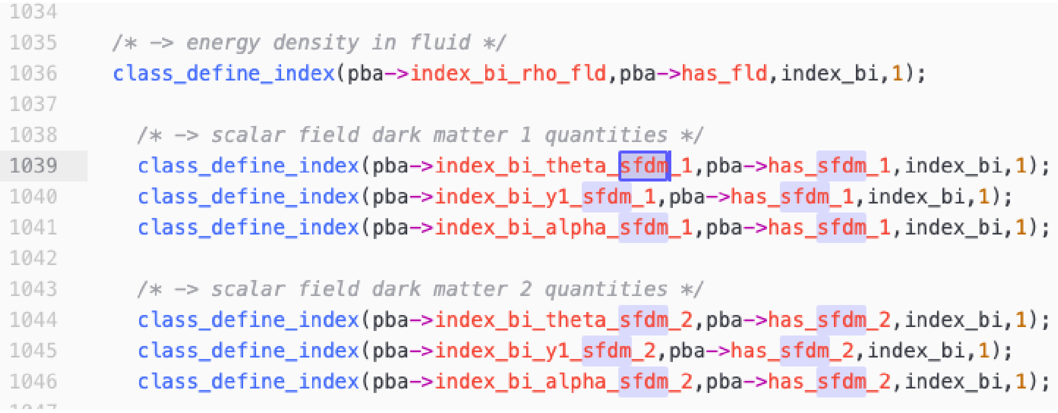
```
}
/*Scalar Field Dark Matter 1*/
if (pba->Omega0_sfdm_1 != 0.)
  pba->has_sfdm_1 = _TRUE_;

/*Scalar Field Dark Matter 2*/
if (pba->Omega0_sfdm_2 != 0.)
  pba->has_sfdm_2 = _TRUE_;

if (pba->Omega0_scf != 0.)
  pba->has_scf = _TRUE_;
```

Figure E.7: Initializing flags.



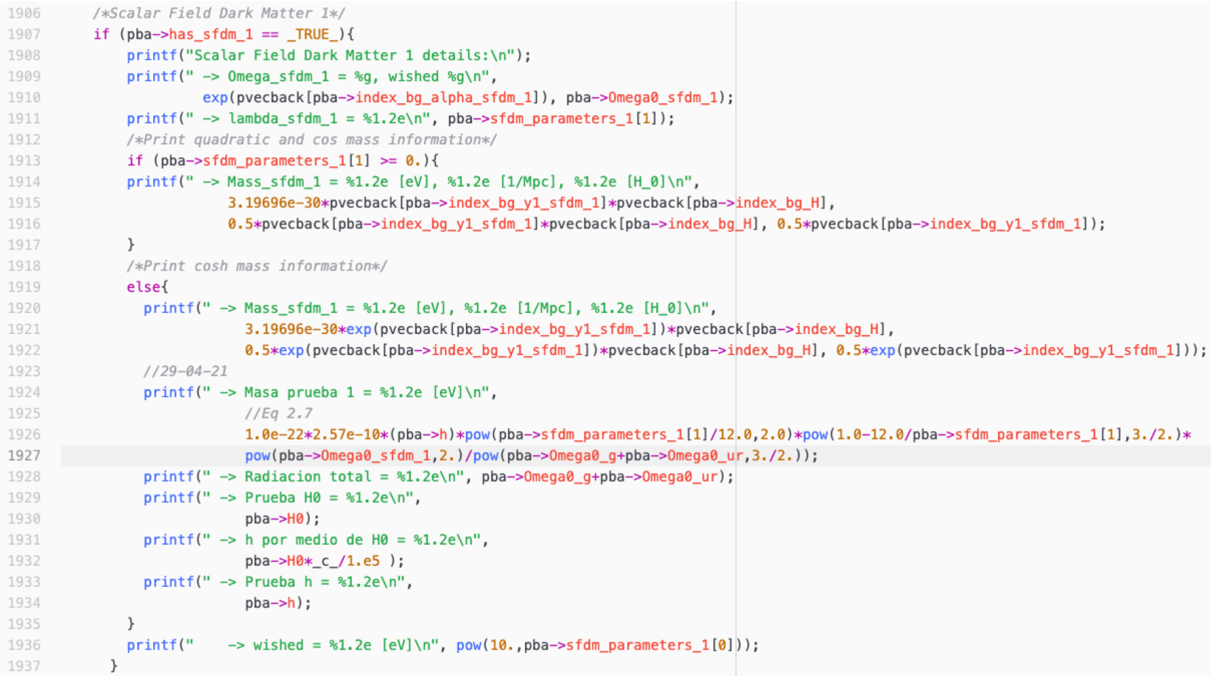
```
/* - indices for scalar field dark matter 1*/
class_define_index(pba->index_bg_theta_sfdm_1,pba->has_sfdm_1,index_bg,1);
class_define_index(pba->index_bg_y1_sfdm_1,pba->has_sfdm_1,index_bg,1);
class_define_index(pba->index_bg_alpha_sfdm_1,pba->has_sfdm_1,index_bg,1);
class_define_index(pba->index_bg_rho_sfdm_1,pba->has_sfdm_1,index_bg,1);
class_define_index(pba->index_bg_p_sfdm_1,pba->has_sfdm_1,index_bg,1);

/* - indices for scalar field dark matter 2*/
class_define_index(pba->index_bg_theta_sfdm_2,pba->has_sfdm_2,index_bg,1);
class_define_index(pba->index_bg_y1_sfdm_2,pba->has_sfdm_2,index_bg,1);
class_define_index(pba->index_bg_alpha_sfdm_2,pba->has_sfdm_2,index_bg,1);
class_define_index(pba->index_bg_rho_sfdm_2,pba->has_sfdm_2,index_bg,1);
class_define_index(pba->index_bg_p_sfdm_2,pba->has_sfdm_2,index_bg,1);
```

Figure E.8: Defining the index_bg indices.



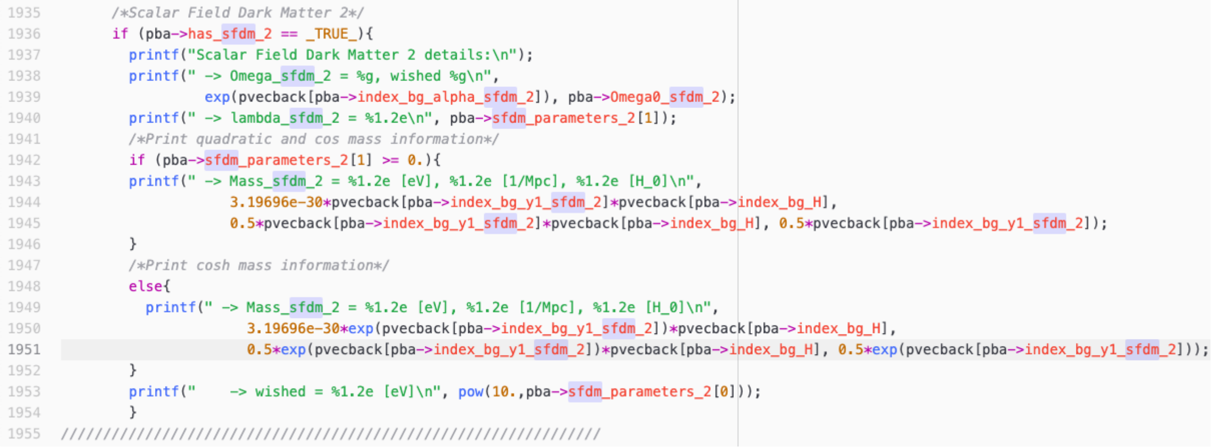
```

/* -> energy density in fluid */
class_define_index(pba->index_bi_rho_fld,pba->has_fld,index_bi,1);

  /* -> scalar field dark matter 1 quantities */
  class_define_index(pba->index_bi_theta_sfdm_1,pba->has_sfdm_1,index_bi,1);
  class_define_index(pba->index_bi_y1_sfdm_1,pba->has_sfdm_1,index_bi,1);
  class_define_index(pba->index_bi_alpha_sfdm_1,pba->has_sfdm_1,index_bi,1);

  /* -> scalar field dark matter 2 quantities */
  class_define_index(pba->index_bi_theta_sfdm_2,pba->has_sfdm_2,index_bi,1);
  class_define_index(pba->index_bi_y1_sfdm_2,pba->has_sfdm_2,index_bi,1);
  class_define_index(pba->index_bi_alpha_sfdm_2,pba->has_sfdm_2,index_bi,1);
```

Figure E.9: Defining the index_bi indices.



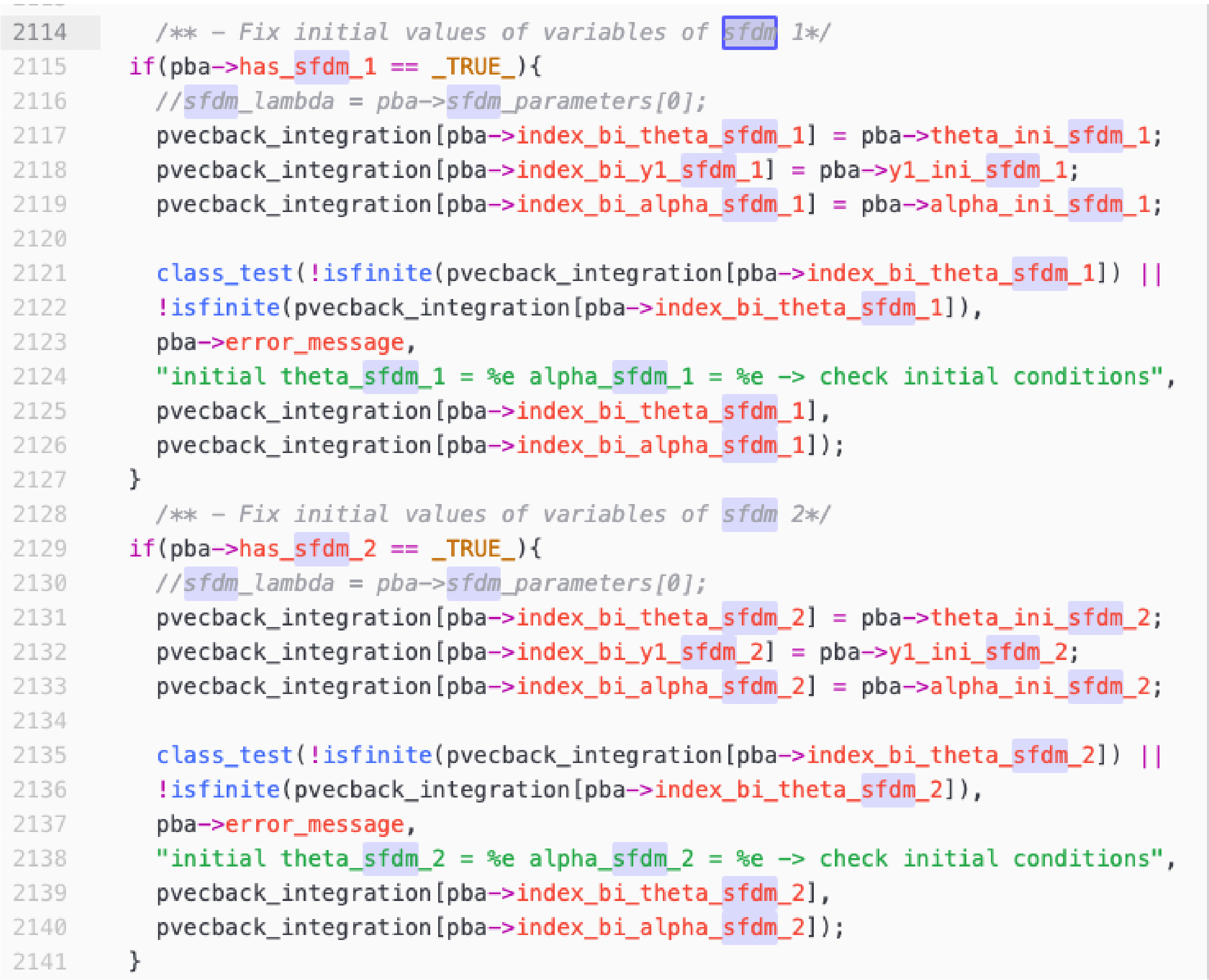
```
    /*Scalar Field Dark Matter 1*/
    if (pba->has_sfdm_1 == _TRUE_){
        printf("Scalar Field Dark Matter 1 details:\n");
        printf(" -> Omega_sfdm_1 = %g, wished %g\n",
                 exp(pvecback[pba->index_bg_alpha_sfdm_1]), pba->Omega0_sfdm_1);
        printf(" -> lambda_sfdm_1 = %1.2e\n", pba->sfdm_parameters_1[1]);
        /*Print quadratic and cos mass information*/
        if (pba->sfdm_parameters_1[1] >= 0.){
        printf(" -> Mass_sfdm_1 = %1.2e [eV], %1.2e [1/Mpc], %1.2e [H_0]\n",
                    3.19696e-30*pvecback[pba->index_bg_y1_sfdm_1]*pvecback[pba->index_bg_H],
                    0.5*pvecback[pba->index_bg_y1_sfdm_1]*pvecback[pba->index_bg_H], 0.5*pvecback[pba->index_bg_y1_sfdm_1]);
        }
        /*Print cosh mass information*/
        else{
          printf(" -> Mass_sfdm_1 = %1.2e [eV], %1.2e [1/Mpc], %1.2e [H_0]\n",
                      3.19696e-30*exp(pvecback[pba->index_bg_y1_sfdm_1])*pvecback[pba->index_bg_H],
                      0.5*exp(pvecback[pba->index_bg_y1_sfdm_1])*pvecback[pba->index_bg_H], 0.5*exp(pvecback[pba->index_bg_y1_sfdm_1]));
          //29-04-21
          printf(" -> Masa prueba 1 = %1.2e [eV]\n",
                      //Eq 2.7
                      1.0e-22*2.57e-10*(pba->h)*pow(pba->sfdm_parameters_1[1]/12.0,2.0)*pow(1.0-12.0/pba->sfdm_parameters_1[1],3./2.)*
                      pow(pba->Omega0_sfdm_1,2.)/pow(pba->Omega0_g+pba->Omega0_ur,3./2.));
          printf(" -> Radiacion total = %1.2e\n", pba->Omega0_g+pba->Omega0_ur);
          printf(" -> Prueba H0 = %1.2e\n",
                      pba->H0);
          printf(" -> h por medio de H0 = %1.2e\n",
                      pba->H0*_c_/1.e5 );
          printf(" -> Prueba h = %1.2e\n",
                      pba->h);
        }
        printf("    -> wished = %1.2e [eV]\n", pow(10.,pba->sfdm_parameters_1[0]));
      }
```

Figure E.10: Printing scalar field 1 information.



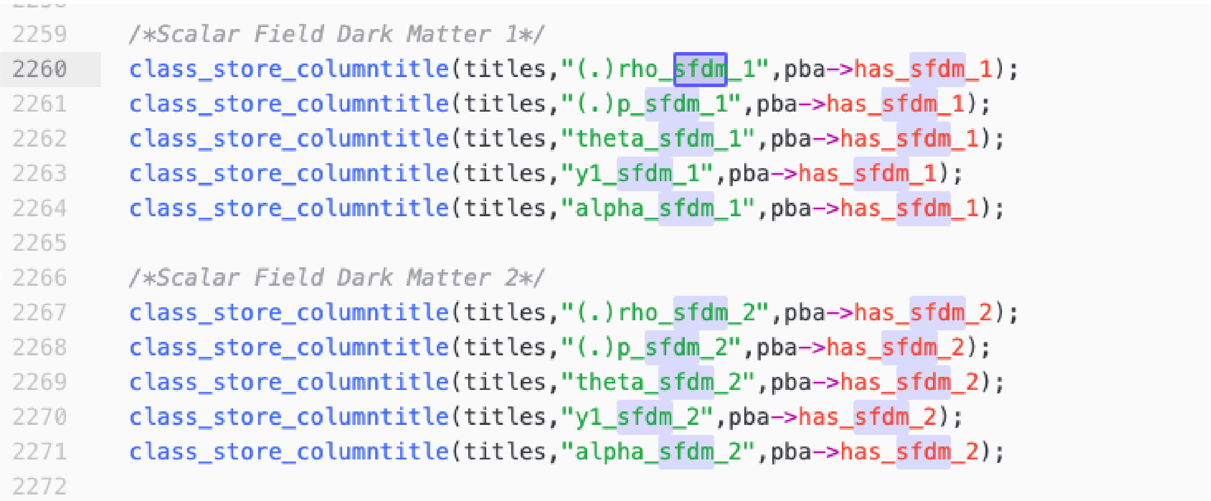
```
      /*Scalar Field Dark Matter 2*/
      if (pba->has_sfdm_2 == _TRUE_){
        printf("Scalar Field Dark Matter 2 details:\n");
        printf(" -> Omega_sfdm_2 = %g, wished %g\n",
                 exp(pvecback[pba->index_bg_alpha_sfdm_2]), pba->Omega0_sfdm_2);
        printf(" -> lambda_sfdm_2 = %1.2e\n", pba->sfdm_parameters_2[1]);
        /*Print quadratic and cos mass information*/
        if (pba->sfdm_parameters_2[1] >= 0.){
        printf(" -> Mass_sfdm_2 = %1.2e [eV], %1.2e [1/Mpc], %1.2e [H_0]\n",
                    3.19696e-30*pvecback[pba->index_bg_y1_sfdm_2]*pvecback[pba->index_bg_H],
                    0.5*pvecback[pba->index_bg_y1_sfdm_2]*pvecback[pba->index_bg_H], 0.5*pvecback[pba->index_bg_y1_sfdm_2]);
        }
        /*Print cosh mass information*/
        else{
          printf(" -> Mass_sfdm_2 = %1.2e [eV], %1.2e [1/Mpc], %1.2e [H_0]\n",
                      3.19696e-30*exp(pvecback[pba->index_bg_y1_sfdm_2])*pvecback[pba->index_bg_H],
                      0.5*exp(pvecback[pba->index_bg_y1_sfdm_2])*pvecback[pba->index_bg_H], 0.5*exp(pvecback[pba->index_bg_y1_sfdm_2]));
        }
        printf("    -> wished = %1.2e [eV]\n", pow(10.,pba->sfdm_parameters_2[0]));
        }
////////////////////////////////////////////////////////////////
```

Figure E.11: Printing scalar field 2 information.



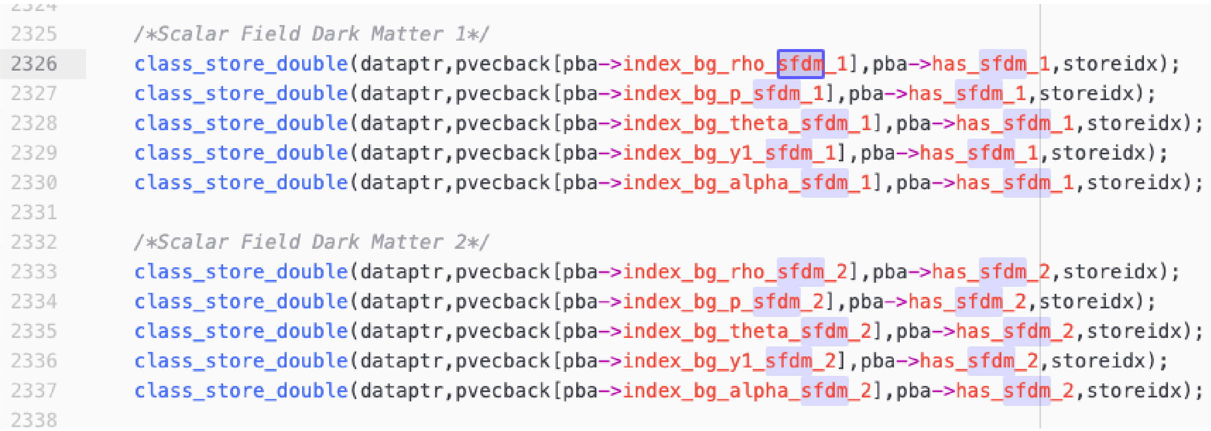
```
    /** - Fix initial values of variables of sfdm 1*/
  if(pba->has_sfdm_1 == _TRUE_){
    //sfdm_lambda = pba->sfdm_parameters[0];
    pvecback_integration[pba->index_bi_theta_sfdm_1] = pba->theta_ini_sfdm_1;
    pvecback_integration[pba->index_bi_y1_sfdm_1] = pba->y1_ini_sfdm_1;
    pvecback_integration[pba->index_bi_alpha_sfdm_1] = pba->alpha_ini_sfdm_1;

    class_test(!isfinite(pvecback_integration[pba->index_bi_theta_sfdm_1]) ||
    !isfinite(pvecback_integration[pba->index_bi_theta_sfdm_1]),
    pba->error_message,
    "initial theta_sfdm_1 = %e alpha_sfdm_1 = %e -> check initial conditions",
    pvecback_integration[pba->index_bi_theta_sfdm_1],
    pvecback_integration[pba->index_bi_alpha_sfdm_1]);
  }
    /** - Fix initial values of variables of sfdm 2*/
  if(pba->has_sfdm_2 == _TRUE_){
    //sfdm_lambda = pba->sfdm_parameters[0];
    pvecback_integration[pba->index_bi_theta_sfdm_2] = pba->theta_ini_sfdm_2;
    pvecback_integration[pba->index_bi_y1_sfdm_2] = pba->y1_ini_sfdm_2;
    pvecback_integration[pba->index_bi_alpha_sfdm_2] = pba->alpha_ini_sfdm_2;

    class_test(!isfinite(pvecback_integration[pba->index_bi_theta_sfdm_2]) ||
    !isfinite(pvecback_integration[pba->index_bi_theta_sfdm_2]),
    pba->error_message,
    "initial theta_sfdm_2 = %e alpha_sfdm_2 = %e -> check initial conditions",
    pvecback_integration[pba->index_bi_theta_sfdm_2],
    pvecback_integration[pba->index_bi_alpha_sfdm_2]);
  }
```

Figure E.12: Fixing initial conditions.



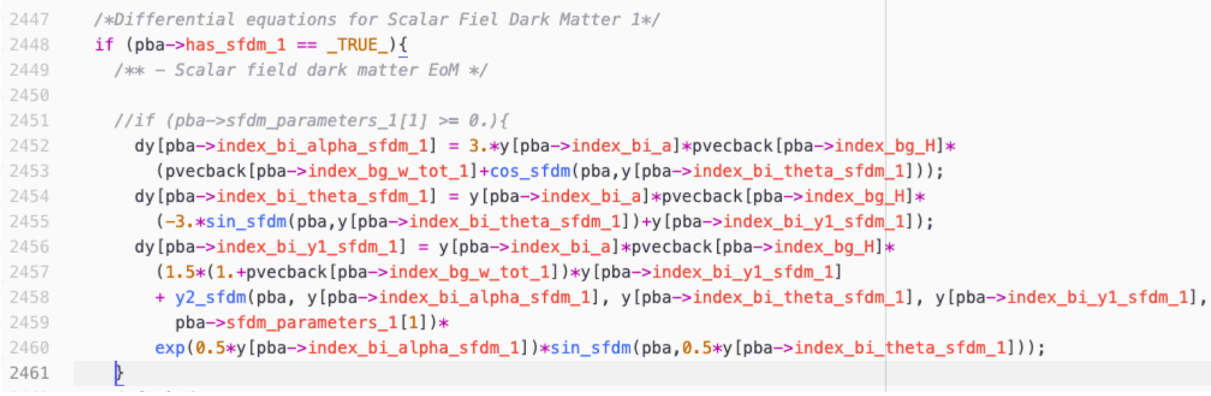
```
/*Scalar Field Dark Matter 1*/
class_store_columntitle(titles,"(.)rho_sfdm_1",pba->has_sfdm_1);
class_store_columntitle(titles,"(.)p_sfdm_1",pba->has_sfdm_1);
class_store_columntitle(titles,"theta_sfdm_1",pba->has_sfdm_1);
class_store_columntitle(titles,"y1_sfdm_1",pba->has_sfdm_1);
class_store_columntitle(titles,"alpha_sfdm_1",pba->has_sfdm_1);

/*Scalar Field Dark Matter 2*/
class_store_columntitle(titles,"(.)rho_sfdm_2",pba->has_sfdm_2);
class_store_columntitle(titles,"(.)p_sfdm_2",pba->has_sfdm_2);
class_store_columntitle(titles,"theta_sfdm_2",pba->has_sfdm_2);
class_store_columntitle(titles,"y1_sfdm_2",pba->has_sfdm_2);
class_store_columntitle(titles,"alpha_sfdm_2",pba->has_sfdm_2);
```

Figure E.13: Naming the columns in the output file.



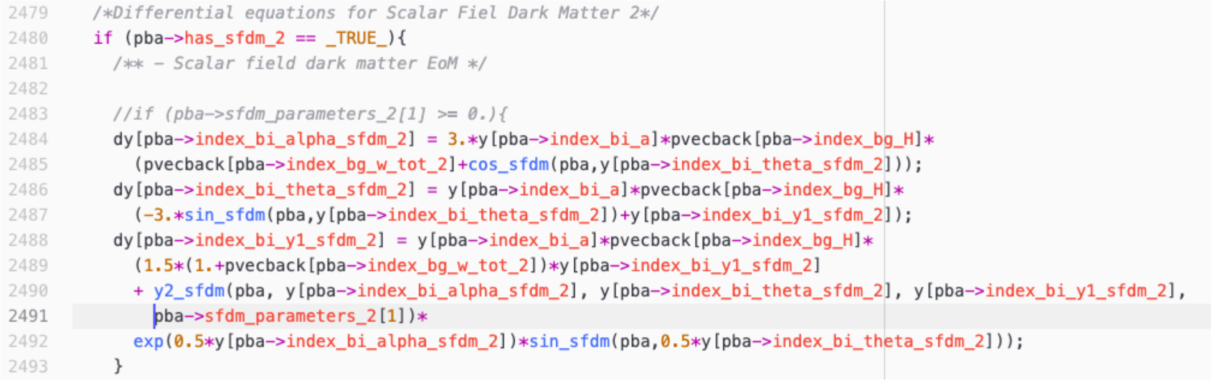
```
/*Scalar Field Dark Matter 1*/
class_store_double(dataptr,pvecback[pba->index_bg_rho_sfdm_1],pba->has_sfdm_1,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_p_sfdm_1],pba->has_sfdm_1,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_theta_sfdm_1],pba->has_sfdm_1,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_y1_sfdm_1],pba->has_sfdm_1,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_alpha_sfdm_1],pba->has_sfdm_1,storeidx);

/*Scalar Field Dark Matter 2*/
class_store_double(dataptr,pvecback[pba->index_bg_rho_sfdm_2],pba->has_sfdm_2,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_p_sfdm_2],pba->has_sfdm_2,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_theta_sfdm_2],pba->has_sfdm_2,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_y1_sfdm_2],pba->has_sfdm_2,storeidx);
class_store_double(dataptr,pvecback[pba->index_bg_alpha_sfdm_2],pba->has_sfdm_2,storeidx);
```

Figure E.14: Assigning the parameters that will be in the output file.



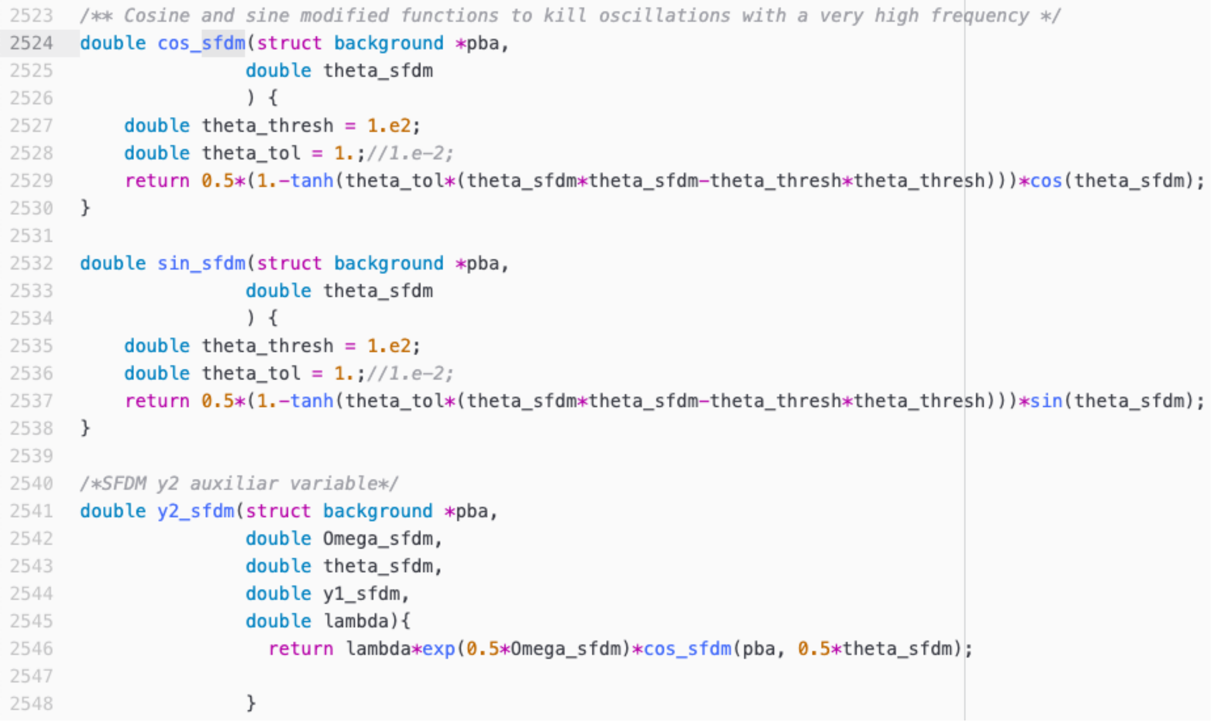
```
/*Differential equations for Scalar Fiel Dark Matter 1*/
if (pba->has_sfdm_1 == _TRUE_){
  /** - Scalar field dark matter EoM */

  //if (pba->sfdm_parameters_1[1] >= 0.){
    dy[pba->index_bi_alpha_sfdm_1] = 3.*y[pba->index_bi_a]*pvecback[pba->index_bg_H]*
      (pvecback[pba->index_bg_w_tot_1]+cos_sfdm(pba,y[pba->index_bi_theta_sfdm_1]));
    dy[pba->index_bi_theta_sfdm_1] = y[pba->index_bi_a]*pvecback[pba->index_bg_H]*
      (-3.*sin_sfdm(pba,y[pba->index_bi_theta_sfdm_1])+y[pba->index_bi_y1_sfdm_1]);
    dy[pba->index_bi_y1_sfdm_1] = y[pba->index_bi_a]*pvecback[pba->index_bg_H]*
      (1.5*(1.+pvecback[pba->index_bg_w_tot_1])*y[pba->index_bi_y1_sfdm_1]
      + y2_sfdm(pba, y[pba->index_bi_alpha_sfdm_1], y[pba->index_bi_theta_sfdm_1], y[pba->index_bi_y1_sfdm_1],
        pba->sfdm_parameters_1[1])*
      exp(0.5*y[pba->index_bi_alpha_sfdm_1])*sin_sfdm(pba,0.5*y[pba->index_bi_theta_sfdm_1]));
  }
```

Figure E.15: Scalar field 1 differential equations.



```
/*Differential equations for Scalar Fiel Dark Matter 2*/
if (pba->has_sfdm_2 == _TRUE_){
  /** - Scalar field dark matter EoM */

  //if (pba->sfdm_parameters_2[1] >= 0.){
  dy[pba->index_bi_alpha_sfdm_2] = 3.*y[pba->index_bi_a]*pvecback[pba->index_bg_H]*
    (pvecback[pba->index_bg_w_tot_2]+cos_sfdm(pba,y[pba->index_bi_theta_sfdm_2]));
  dy[pba->index_bi_theta_sfdm_2] = y[pba->index_bi_a]*pvecback[pba->index_bg_H]*
    (-3.*sin_sfdm(pba,y[pba->index_bi_theta_sfdm_2])+y[pba->index_bi_y1_sfdm_2]);
  dy[pba->index_bi_y1_sfdm_2] = y[pba->index_bi_a]*pvecback[pba->index_bg_H]*
    (1.5*(1.+pvecback[pba->index_bg_w_tot_2])*y[pba->index_bi_y1_sfdm_2]
    + y2_sfdm(pba, y[pba->index_bi_alpha_sfdm_2], y[pba->index_bi_theta_sfdm_2], y[pba->index_bi_y1_sfdm_2],
      pba->sfdm_parameters_2[1])*
    exp(0.5*y[pba->index_bi_alpha_sfdm_2])*sin_sfdm(pba,0.5*y[pba->index_bi_theta_sfdm_2]));
  }
```

Figure E.16: Scalar field 2 differential equations.



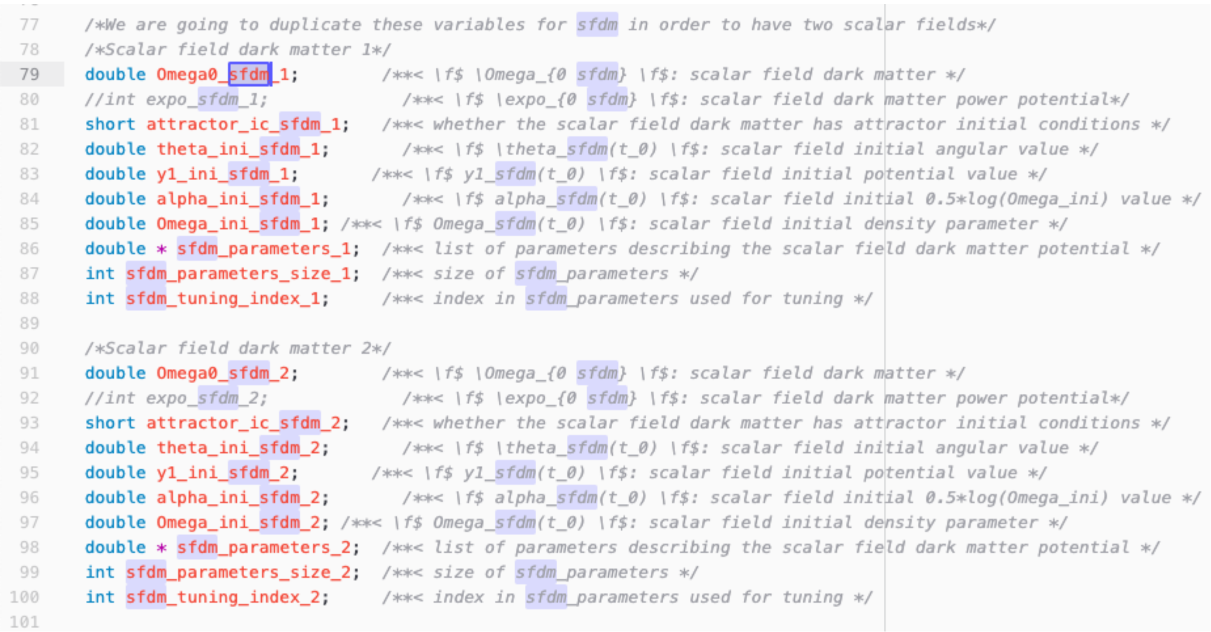
```c
/** Cosine and sine modified functions to kill oscillations with a very high frequency */
double cos_sfdm(struct background *pba,
               double theta_sfdm
               ) {
    double theta_thresh = 1.e2;
    double theta_tol = 1.;//1.e-2;
    return 0.5*(1.-tanh(theta_tol*(theta_sfdm*theta_sfdm-theta_thresh*theta_thresh)))*cos(theta_sfdm);
}

double sin_sfdm(struct background *pba,
               double theta_sfdm
               ) {
    double theta_thresh = 1.e2;
    double theta_tol = 1.;//1.e-2;
    return 0.5*(1.-tanh(theta_tol*(theta_sfdm*theta_sfdm-theta_thresh*theta_thresh)))*sin(theta_sfdm);
}

/*SFDM y2 auxiliar variable*/
double y2_sfdm(struct background *pba,
               double Omega_sfdm,
               double theta_sfdm,
               double y1_sfdm,
               double lambda){
                 return lambda*exp(0.5*Omega_sfdm)*cos_sfdm(pba, 0.5*theta_sfdm);

               }
```

Figure E.17: Defining auxiliary functions for the differential equations.

Appendix F

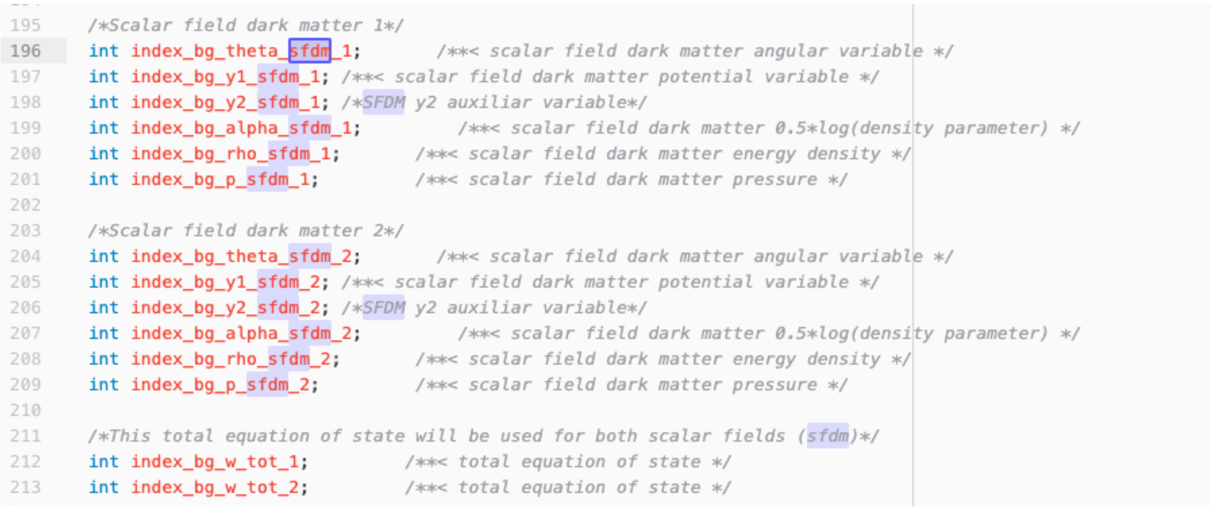
background.h

In this file we declare the variables that we use in the background.c file, that is, we tell CLASS what type each one is, see Figures F.1, F.2, F.3, F.4 and F.5.



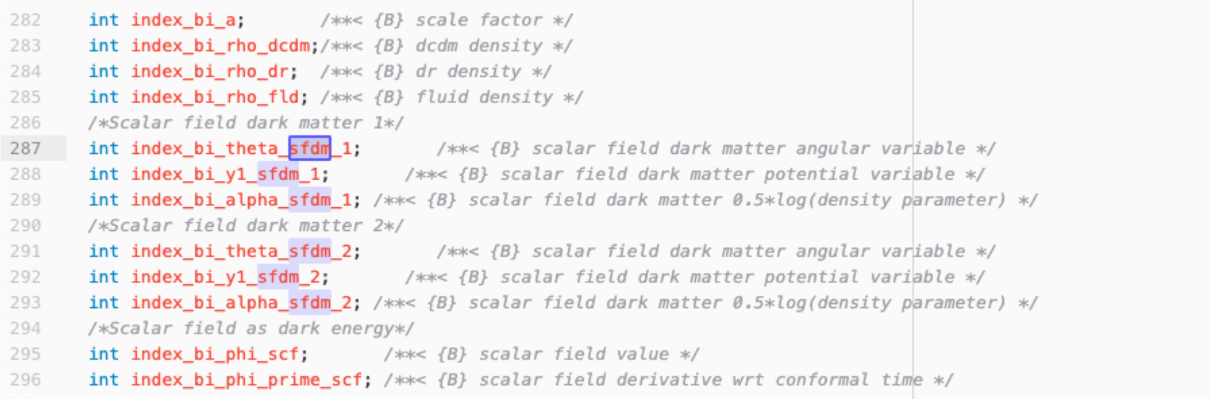
```
/*We are going to duplicate these variables for sfdm in order to have two scalar fields*/
/*Scalar field dark matter 1*/
double Omega0_sfdm_1;        /**< \f$ \Omega_{0 sfdm} \f$: scalar field dark matter */
//int expo_sfdm_1;             /**< \f$ \expo_{0 sfdm} \f$: scalar field dark matter power potential*/
short attractor_ic_sfdm_1;   /**< whether the scalar field dark matter has attractor initial conditions */
double theta_ini_sfdm_1;       /**< \f$ \theta_sfdm(t_0) \f$: scalar field initial angular value */
double y1_ini_sfdm_1;       /**< \f$ y1_sfdm(t_0) \f$: scalar field initial potential value */
double alpha_ini_sfdm_1;       /**< \f$ alpha_sfdm(t_0) \f$: scalar field initial 0.5*log(Omega_ini) value */
double Omega_ini_sfdm_1; /**< \f$ Omega_sfdm(t_0) \f$: scalar field initial density parameter */
double * sfdm_parameters_1;  /**< list of parameters describing the scalar field dark matter potential */
int sfdm_parameters_size_1;  /**< size of sfdm_parameters */
int sfdm_tuning_index_1;     /**< index in sfdm_parameters used for tuning */

/*Scalar field dark matter 2*/
double Omega0_sfdm_2;        /**< \f$ \Omega_{0 sfdm} \f$: scalar field dark matter */
//int expo_sfdm_2;             /**< \f$ \expo_{0 sfdm} \f$: scalar field dark matter power potential*/
short attractor_ic_sfdm_2;   /**< whether the scalar field dark matter has attractor initial conditions */
double theta_ini_sfdm_2;       /**< \f$ \theta_sfdm(t_0) \f$: scalar field initial angular value */
double y1_ini_sfdm_2;       /**< \f$ y1_sfdm(t_0) \f$: scalar field initial potential value */
double alpha_ini_sfdm_2;       /**< \f$ alpha_sfdm(t_0) \f$: scalar field initial 0.5*log(Omega_ini) value */
double Omega_ini_sfdm_2; /**< \f$ Omega_sfdm(t_0) \f$: scalar field initial density parameter */
double * sfdm_parameters_2;  /**< list of parameters describing the scalar field dark matter potential */
int sfdm_parameters_size_2;  /**< size of sfdm_parameters */
int sfdm_tuning_index_2;     /**< index in sfdm_parameters used for tuning */
```

Figure F.1: Defining scalar fields quantities.



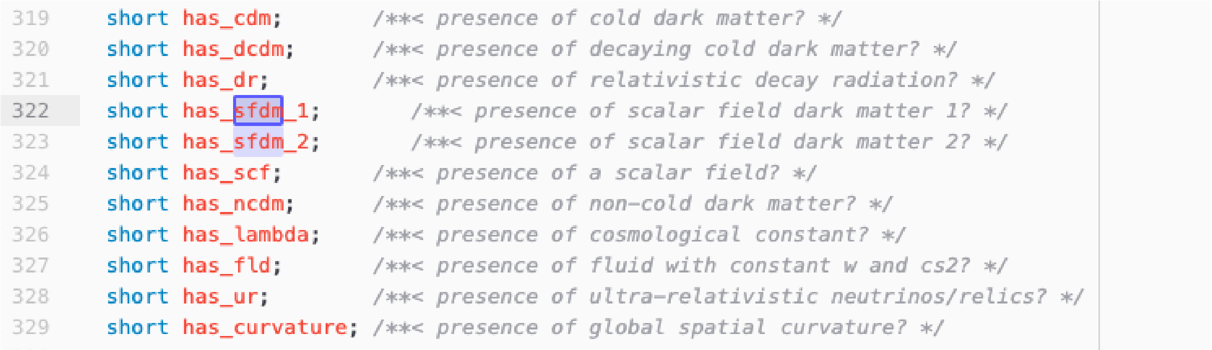
```
/*Scalar field dark matter 1*/
int index_bg_theta_sfdm_1;       /**< scalar field dark matter angular variable */
int index_bg_y1_sfdm_1; /**< scalar field dark matter potential variable */
int index_bg_y2_sfdm_1; /*SFDM y2 auxiliar variable*/
int index_bg_alpha_sfdm_1;         /**< scalar field dark matter 0.5*log(density parameter) */
int index_bg_rho_sfdm_1;       /**< scalar field dark matter energy density */
int index_bg_p_sfdm_1;         /**< scalar field dark matter pressure */

/*Scalar field dark matter 2*/
int index_bg_theta_sfdm_2;       /**< scalar field dark matter angular variable */
int index_bg_y1_sfdm_2; /**< scalar field dark matter potential variable */
int index_bg_y2_sfdm_2; /*SFDM y2 auxiliar variable*/
int index_bg_alpha_sfdm_2;         /**< scalar field dark matter 0.5*log(density parameter) */
int index_bg_rho_sfdm_2;       /**< scalar field dark matter energy density */
int index_bg_p_sfdm_2;         /**< scalar field dark matter pressure */

/*This total equation of state will be used for both scalar fields (sfdm)*/
int index_bg_w_tot_1;         /**< total equation of state */
int index_bg_w_tot_2;         /**< total equation of state */
```

Figure F.2: Defining scalar fields index_bg.



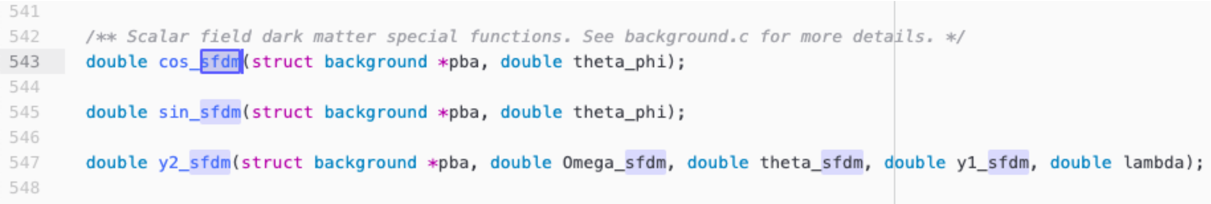
```
int index_bi_a;       /**< {B} scale factor */
int index_bi_rho_dcdm;/**< {B} dcdm density */
int index_bi_rho_dr;  /**< {B} dr density */
int index_bi_rho_fld; /**< {B} fluid density */
/*Scalar field dark matter 1*/
int index_bi_theta_sfdm_1;       /**< {B} scalar field dark matter angular variable */
int index_bi_y1_sfdm_1;       /**< {B} scalar field dark matter potential variable */
int index_bi_alpha_sfdm_1; /**< {B} scalar field dark matter 0.5*log(density parameter) */
/*Scalar field dark matter 2*/
int index_bi_theta_sfdm_2;       /**< {B} scalar field dark matter angular variable */
int index_bi_y1_sfdm_2;       /**< {B} scalar field dark matter potential variable */
int index_bi_alpha_sfdm_2; /**< {B} scalar field dark matter 0.5*log(density parameter) */
/*Scalar field as dark energy*/
int index_bi_phi_scf;       /**< {B} scalar field value */
int index_bi_phi_prime_scf; /**< {B} scalar field derivative wrt conformal time */
```

Figure F.3: Defining scalar fields index_bi.



```
 short has_cdm;       /**< presence of cold dark matter? */
 short has_dcdm;      /**< presence of decaying cold dark matter? */
 short has_dr;        /**< presence of relativistic decay radiation? */
 short has_sfdm_1;       /**< presence of scalar field dark matter 1? */
 short has_sfdm_2;       /**< presence of scalar field dark matter 2? */
 short has_scf;       /**< presence of a scalar field? */
 short has_ncdm;      /**< presence of non-cold dark matter? */
 short has_lambda;    /**< presence of cosmological constant? */
 short has_fld;       /**< presence of fluid with constant w and cs2? */
 short has_ur;        /**< presence of ultra-relativistic neutrinos/relics? */
 short has_curvature; /**< presence of global spatial curvature? */
```

Figure F.4: Defining scalar fields flags.



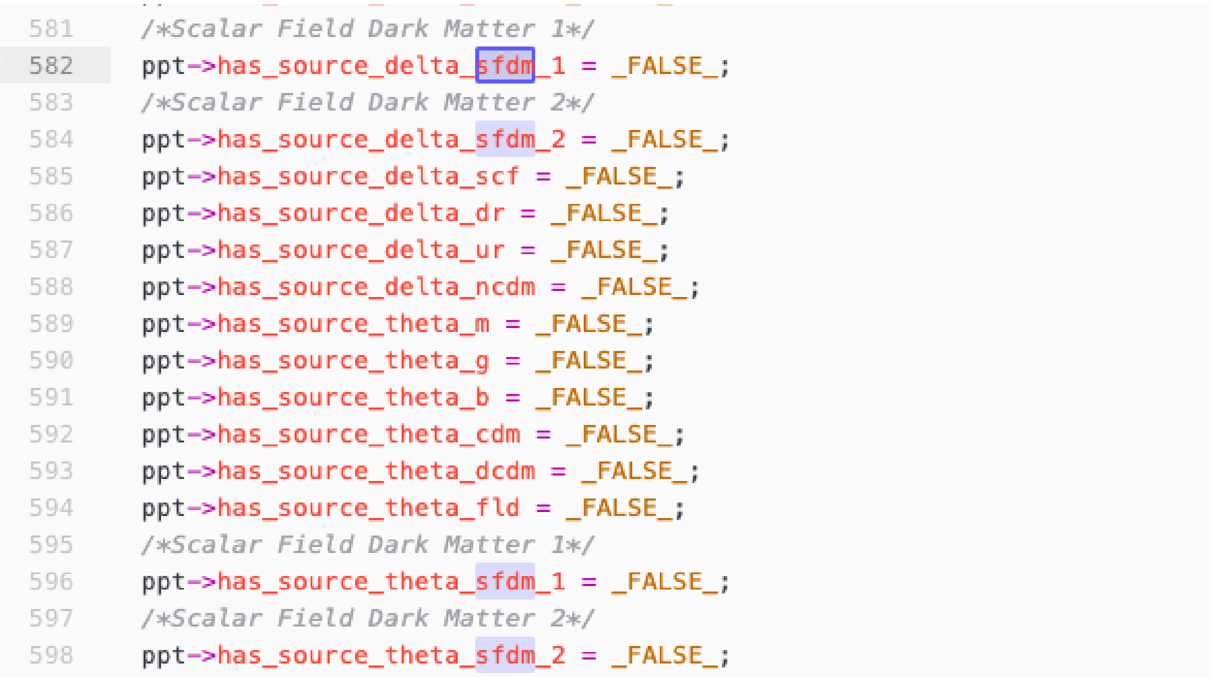
```c

/** Scalar field dark matter special functions. See background.c for more details. */
double cos_sfdm(struct background *pba, double theta_phi);

double sin_sfdm(struct background *pba, double theta_phi);

double y2_sfdm(struct background *pba, double Omega_sfdm, double theta_sfdm, double y1_sfdm, double lambda);

```

Figure F.5: Declaring auxiliary functions for the differential equations.

Appendix G

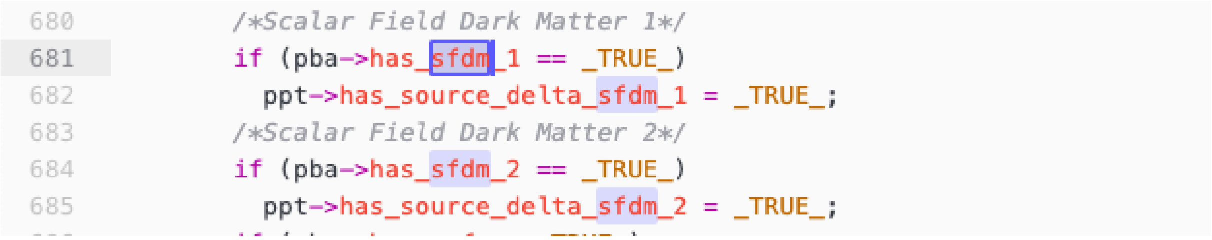
perturbations.c

In this file we are going to put the differential equations for the perturbations. First, as a good programming practice, we initialize the flags associated with the scalar fields to false, see Figure G.1. Then if the input file says that there are scalar fields, the flags are set to true, see Figures G.2 and G.3.



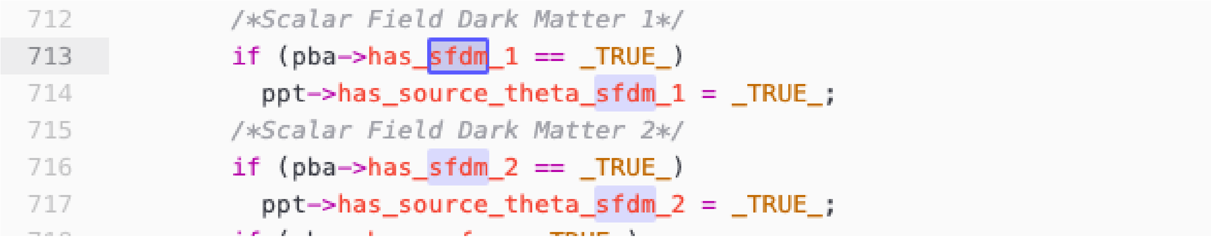
```
/*Scalar Field Dark Matter 1*/
ppt->has_source_delta_sfdm_1 = _FALSE_;
/*Scalar Field Dark Matter 2*/
ppt->has_source_delta_sfdm_2 = _FALSE_;
ppt->has_source_delta_scf = _FALSE_;
ppt->has_source_delta_dr = _FALSE_;
ppt->has_source_delta_ur = _FALSE_;
ppt->has_source_delta_ncdm = _FALSE_;
ppt->has_source_theta_m = _FALSE_;
ppt->has_source_theta_g = _FALSE_;
ppt->has_source_theta_b = _FALSE_;
ppt->has_source_theta_cdm = _FALSE_;
ppt->has_source_theta_dcdm = _FALSE_;
ppt->has_source_theta_fld = _FALSE_;
/*Scalar Field Dark Matter 1*/
ppt->has_source_theta_sfdm_1 = _FALSE_;
/*Scalar Field Dark Matter 2*/
ppt->has_source_theta_sfdm_2 = _FALSE_;
```

Figure G.1: Initializing the scalar fields flags to false.



```
/*Scalar Field Dark Matter 1*/
if (pba->has_sfdm_1 == _TRUE_)
  ppt->has_source_delta_sfdm_1 = _TRUE_;
/*Scalar Field Dark Matter 2*/
if (pba->has_sfdm_2 == _TRUE_)
  ppt->has_source_delta_sfdm_2 = _TRUE_;
```

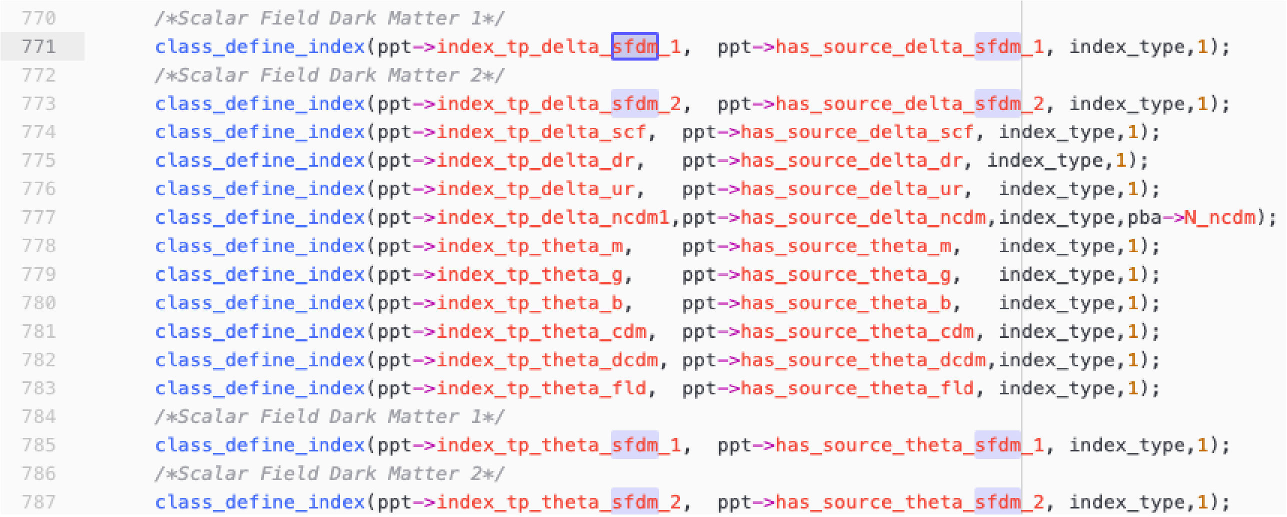
Figure G.2: Setting the scalar fields flags to true.



```
/*Scalar Field Dark Matter 1*/
if (pba->has_sfdm_1 == _TRUE_)
  ppt->has_source_theta_sfdm_1 = _TRUE_;
/*Scalar Field Dark Matter 2*/
if (pba->has_sfdm_2 == _TRUE_)
  ppt->has_source_theta_sfdm_2 = _TRUE_;
```

Figure G.3: Setting the scalar fields flags to true.

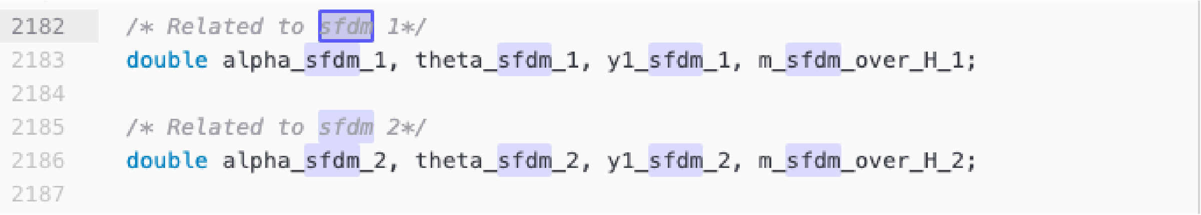
The, we define the indices for the perturbations variables, see Figure G.4. As in background.c, we also need to declare local variables, see Figure G.5.



```
/*Scalar Field Dark Matter 1*/
class_define_index(ppt->index_tp_delta_sfdm_1,  ppt->has_source_delta_sfdm_1, index_type,1);
/*Scalar Field Dark Matter 2*/
class_define_index(ppt->index_tp_delta_sfdm_2,  ppt->has_source_delta_sfdm_2, index_type,1);
class_define_index(ppt->index_tp_delta_scf,   ppt->has_source_delta_scf, index_type,1);
class_define_index(ppt->index_tp_delta_dr,    ppt->has_source_delta_dr, index_type,1);
class_define_index(ppt->index_tp_delta_ur,    ppt->has_source_delta_ur,  index_type,1);
class_define_index(ppt->index_tp_delta_ncdm1,ppt->has_source_delta_ncdm,index_type,pba->N_ncdm);
class_define_index(ppt->index_tp_theta_m,     ppt->has_source_theta_m,   index_type,1);
class_define_index(ppt->index_tp_theta_g,     ppt->has_source_theta_g,   index_type,1);
class_define_index(ppt->index_tp_theta_b,     ppt->has_source_theta_b,   index_type,1);
class_define_index(ppt->index_tp_theta_cdm,   ppt->has_source_theta_cdm, index_type,1);
class_define_index(ppt->index_tp_theta_dcdm,  ppt->has_source_theta_dcdm,index_type,1);
class_define_index(ppt->index_tp_theta_fld,   ppt->has_source_theta_fld, index_type,1);
/*Scalar Field Dark Matter 1*/
class_define_index(ppt->index_tp_theta_sfdm_1,  ppt->has_source_theta_sfdm_1, index_type,1);
/*Scalar Field Dark Matter 2*/
class_define_index(ppt->index_tp_theta_sfdm_2,  ppt->has_source_theta_sfdm_2, index_type,1);
```

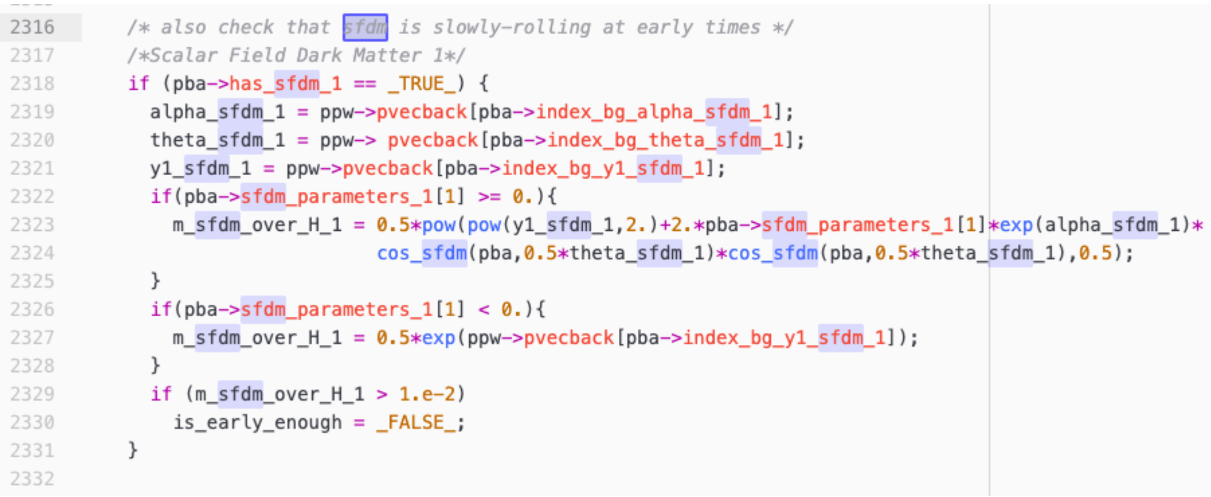
Figure G.4: Defining the perturbations indices.

Then, we check that the slow rolling condition is met for both fields, see Figures G.6 and G.7.



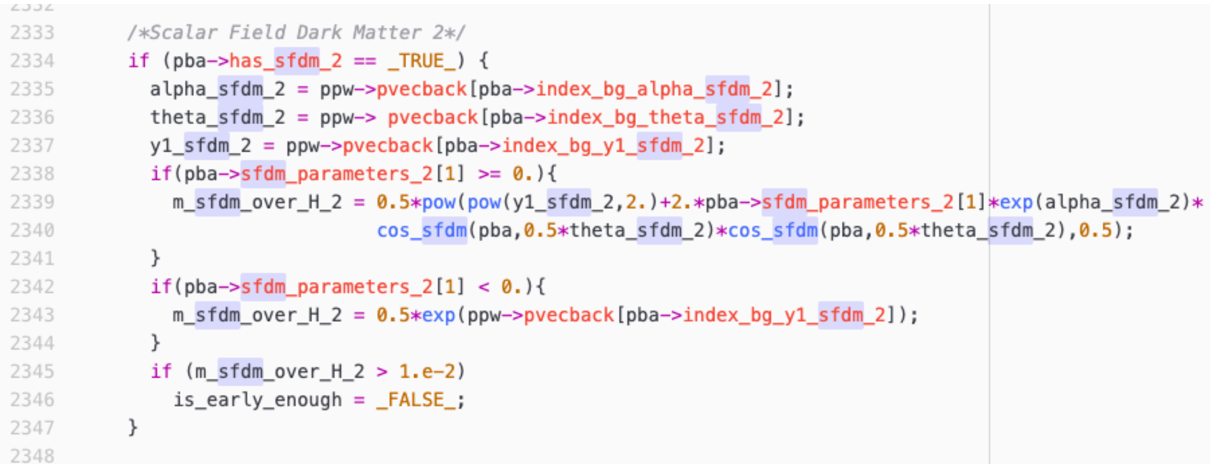
```
/* Related to sfdm 1*/
double alpha_sfdm_1, theta_sfdm_1, y1_sfdm_1, m_sfdm_over_H_1;

/* Related to sfdm 2*/
double alpha_sfdm_2, theta_sfdm_2, y1_sfdm_2, m_sfdm_over_H_2;
```

Figure G.5: Declaring local variables.



```
/* also check that sfdm is slowly-rolling at early times */
/*Scalar Field Dark Matter 1*/
if (pba->has_sfdm_1 == _TRUE_) {
  alpha_sfdm_1 = ppw->pvecback[pba->index_bg_alpha_sfdm_1];
  theta_sfdm_1 = ppw-> pvecback[pba->index_bg_theta_sfdm_1];
  y1_sfdm_1 = ppw->pvecback[pba->index_bg_y1_sfdm_1];
  if(pba->sfdm_parameters_1[1] >= 0.){
    m_sfdm_over_H_1 = 0.5*pow(pow(y1_sfdm_1,2.)+2.*pba->sfdm_parameters_1[1]*exp(alpha_sfdm_1)*
                      cos_sfdm(pba,0.5*theta_sfdm_1)*cos_sfdm(pba,0.5*theta_sfdm_1),0.5);
  }
  if(pba->sfdm_parameters_1[1] < 0.){
    m_sfdm_over_H_1 = 0.5*exp(ppw->pvecback[pba->index_bg_y1_sfdm_1]);
  }
  if (m_sfdm_over_H_1 > 1.e-2)
    is_early_enough = _FALSE_;
}
```

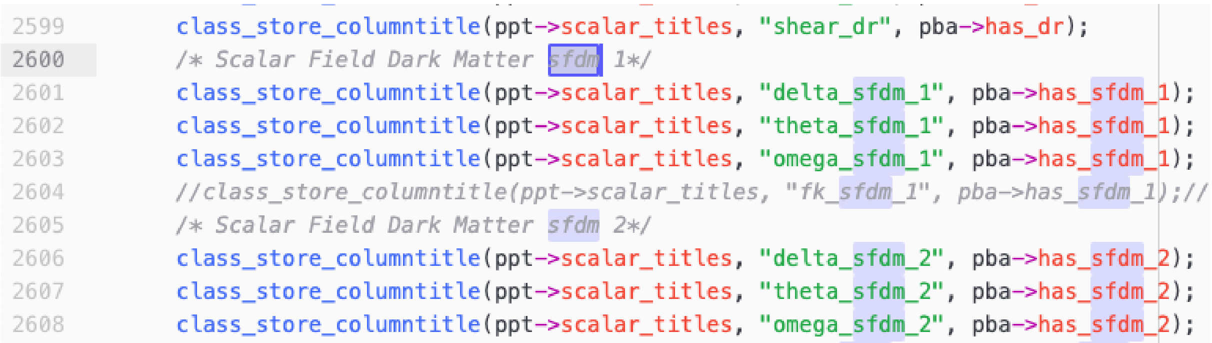
Figure G.6: Checking slow-rolling condition.



```
/*Scalar Field Dark Matter 2*/
if (pba->has_sfdm_2 == _TRUE_) {
  alpha_sfdm_2 = ppw->pvecback[pba->index_bg_alpha_sfdm_2];
  theta_sfdm_2 = ppw-> pvecback[pba->index_bg_theta_sfdm_2];
  y1_sfdm_2 = ppw->pvecback[pba->index_bg_y1_sfdm_2];
  if(pba->sfdm_parameters_2[1] >= 0.){
    m_sfdm_over_H_2 = 0.5*pow(pow(y1_sfdm_2,2.)+2.*pba->sfdm_parameters_2[1]*exp(alpha_sfdm_2)*
                      cos_sfdm(pba,0.5*theta_sfdm_2)*cos_sfdm(pba,0.5*theta_sfdm_2),0.5);
  }
  if(pba->sfdm_parameters_2[1] < 0.){
    m_sfdm_over_H_2 = 0.5*exp(ppw->pvecback[pba->index_bg_y1_sfdm_2]);
  }
  if (m_sfdm_over_H_2 > 1.e-2)
    is_early_enough = _FALSE_;
}
```

Figure G.7: Checking slow-rolling condition.

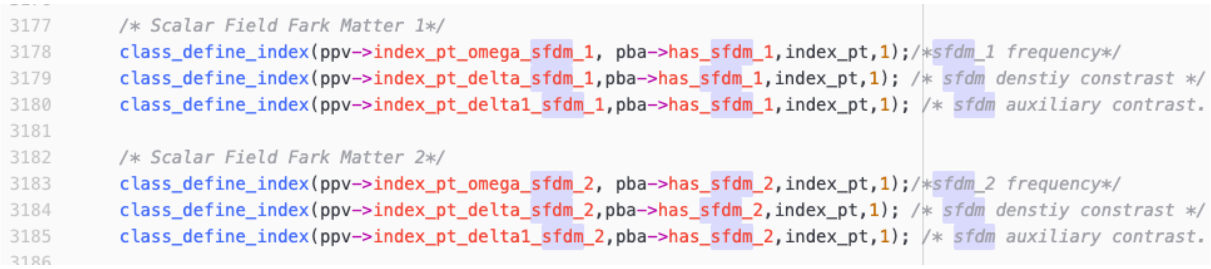
We also give the name to the columns of the parameters of interest that will be saved in the output file, see Figure G.8.



```
    class_store_columntitle(ppt->scalar_titles, "shear_dr", pba->has_dr);
    /* Scalar Field Dark Matter sfdm 1*/
    class_store_columntitle(ppt->scalar_titles, "delta_sfdm_1", pba->has_sfdm_1);
    class_store_columntitle(ppt->scalar_titles, "theta_sfdm_1", pba->has_sfdm_1);
    class_store_columntitle(ppt->scalar_titles, "omega_sfdm_1", pba->has_sfdm_1);
    //class_store_columntitle(ppt->scalar_titles, "fk_sfdm_1", pba->has_sfdm_1);//
    /* Scalar Field Dark Matter sfdm 2*/
    class_store_columntitle(ppt->scalar_titles, "delta_sfdm_2", pba->has_sfdm_2);
    class_store_columntitle(ppt->scalar_titles, "theta_sfdm_2", pba->has_sfdm_2);
    class_store_columntitle(ppt->scalar_titles, "omega_sfdm_2", pba->has_sfdm_2);
```

Figure G.8: Naming the columns in the output file.

We define the indices of the variables with which the differential equations are going to be solved, see Figure G.9.



```
    /* Scalar Field Fark Matter 1*/
    class_define_index(ppv->index_pt_omega_sfdm_1, pba->has_sfdm_1,index_pt,1);/*sfdm_1 frequency*/
    class_define_index(ppv->index_pt_delta_sfdm_1,pba->has_sfdm_1,index_pt,1); /* sfdm denstiy constrast */
    class_define_index(ppv->index_pt_delta1_sfdm_1,pba->has_sfdm_1,index_pt,1); /* sfdm auxiliary contrast.

    /* Scalar Field Fark Matter 2*/
    class_define_index(ppv->index_pt_omega_sfdm_2, pba->has_sfdm_2,index_pt,1);/*sfdm_2 frequency*/
    class_define_index(ppv->index_pt_delta_sfdm_2,pba->has_sfdm_2,index_pt,1); /* sfdm denstiy constrast */
    class_define_index(ppv->index_pt_delta1_sfdm_2,pba->has_sfdm_2,index_pt,1); /* sfdm auxiliary contrast.
```

Figure G.9: Defining scalar fields indices.

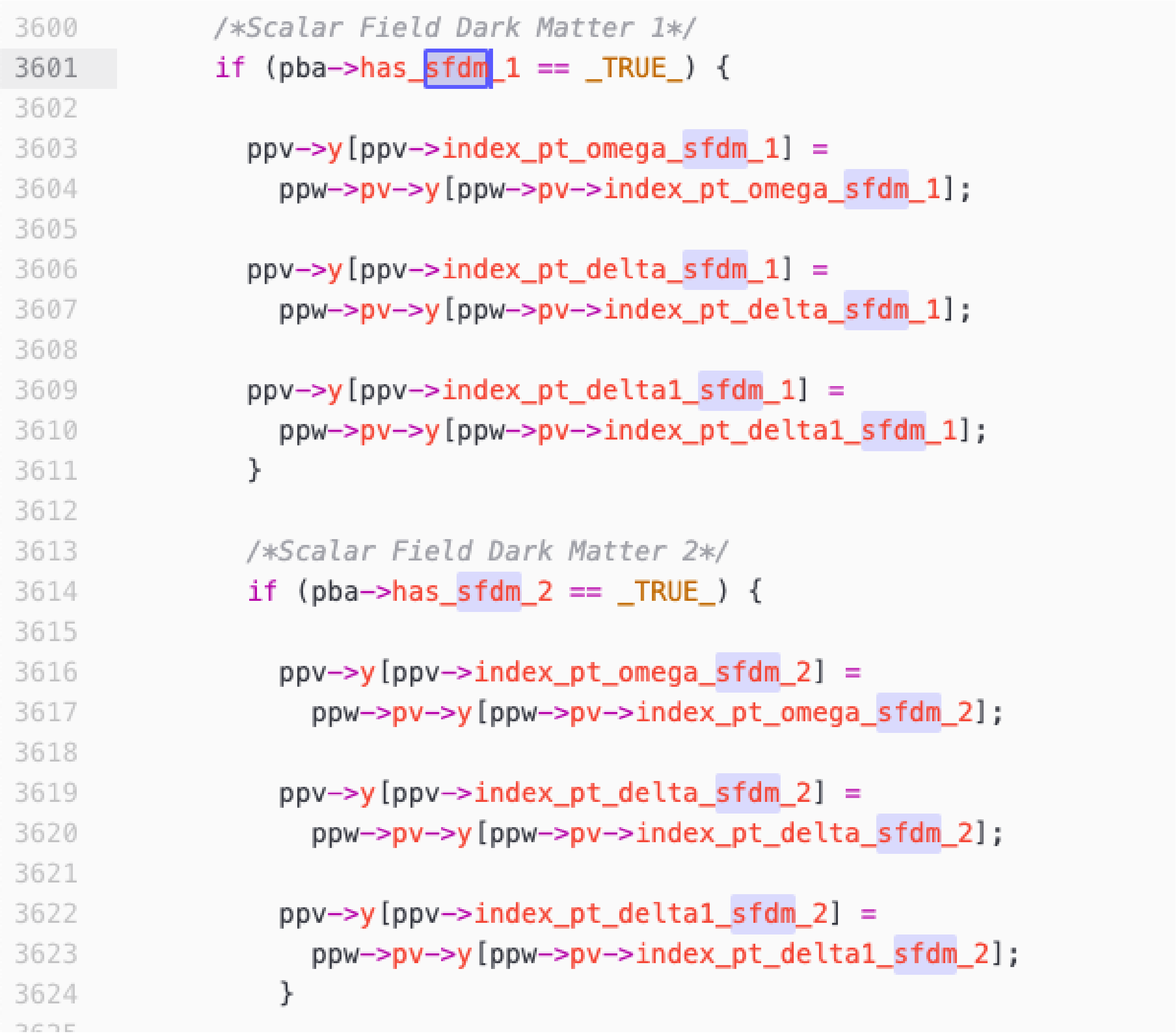
The next step is a technical detail where we have to reassign the indices defined above, see Figure G.10.

Then we declare local variables, see Figure G.11.

We include the scalar fields to the radiation or matter content depending on the value of λ_ϕ.

We define the fraction of each scalar field dark matter as its density divided by the total density, see Figure G.13.

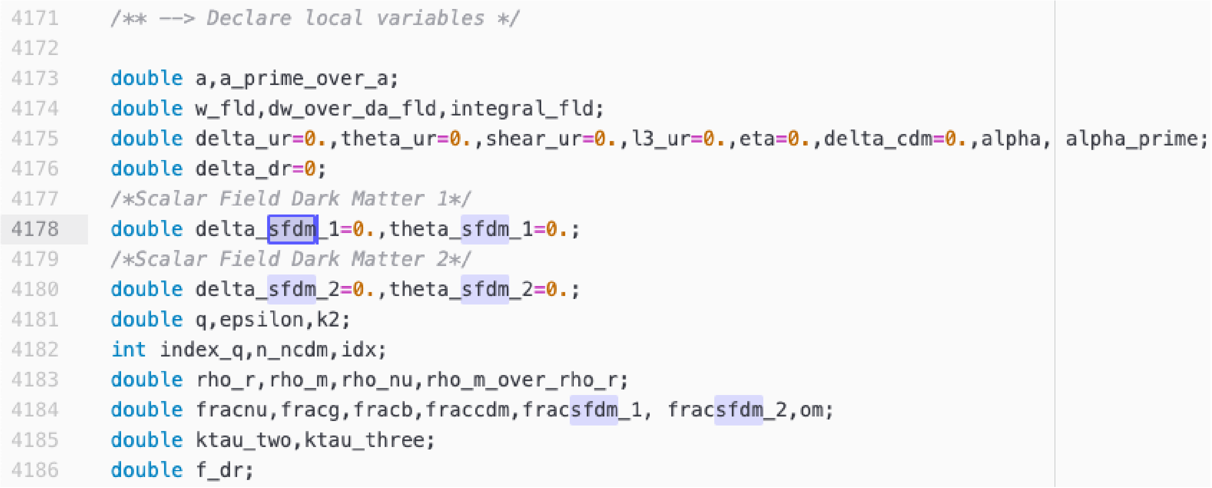
Now we put the initial conditions for the perturbations, see Figures G.14 and G.15. Then we declare other local variables that will appear in the differential equations, see



```
    /*Scalar Field Dark Matter 1*/
    if (pba->has_sfdm_1 == _TRUE_) {

      ppv->y[ppv->index_pt_omega_sfdm_1] =
        ppw->pv->y[ppw->pv->index_pt_omega_sfdm_1];

      ppv->y[ppv->index_pt_delta_sfdm_1] =
        ppw->pv->y[ppw->pv->index_pt_delta_sfdm_1];

      ppv->y[ppv->index_pt_delta1_sfdm_1] =
        ppw->pv->y[ppw->pv->index_pt_delta1_sfdm_1];
      }

      /*Scalar Field Dark Matter 2*/
      if (pba->has_sfdm_2 == _TRUE_) {

        ppv->y[ppv->index_pt_omega_sfdm_2] =
          ppw->pv->y[ppw->pv->index_pt_omega_sfdm_2];

        ppv->y[ppv->index_pt_delta_sfdm_2] =
          ppw->pv->y[ppw->pv->index_pt_delta_sfdm_2];

        ppv->y[ppv->index_pt_delta1_sfdm_2] =
          ppw->pv->y[ppw->pv->index_pt_delta1_sfdm_2];
        }
```

Figure G.10: Reassigning the scalar fields indices.

Figure G.16. Now, we define the quantities (3.18a), see Figures G.17 and G.18. Then, we include the density contrast of the fields to the total density contrast, see Figure G.19. We include the scalar fields to the matter velocity divergence, see Figure G.20. We declare the variables used in Figure G.20, see Figure G.21. If the contribution of the scalar fields is nonzero, we tell the code to include the perturbations of each field, see Figure G.22. Now, we define the quantities (3.18c) for both fields, see Figures G.23 and G.24. Then, we declare as zero the fields local variables used above, see Figure G.25. Now, we store the

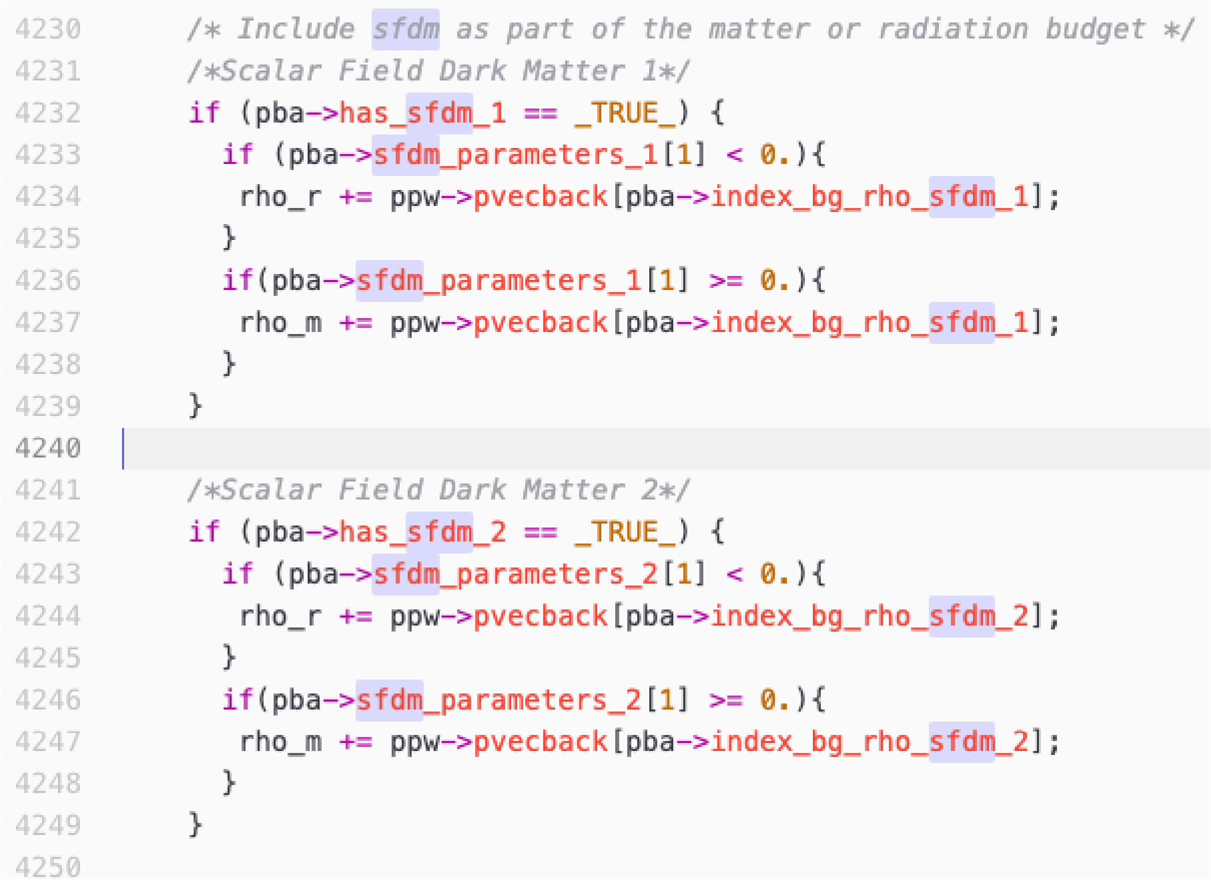


```
/** --> Declare local variables */

double a,a_prime_over_a;
double w_fld,dw_over_da_fld,integral_fld;
double delta_ur=0.,theta_ur=0.,shear_ur=0.,l3_ur=0.,eta=0.,delta_cdm=0.,alpha, alpha_prime;
double delta_dr=0;
/*Scalar Field Dark Matter 1*/
double delta_sfdm_1=0.,theta_sfdm_1=0.;
/*Scalar Field Dark Matter 2*/
double delta_sfdm_2=0.,theta_sfdm_2=0.;
double q,epsilon,k2;
int index_q,n_ncdm,idx;
double rho_r,rho_m,rho_nu,rho_m_over_rho_r;
double fracnu,fracg,fracb,fraccdm,fracsfdm_1, fracsfdm_2,om;
double ktau_two,ktau_three;
double f_dr;
```

Figure G.11: Declaring local variables.

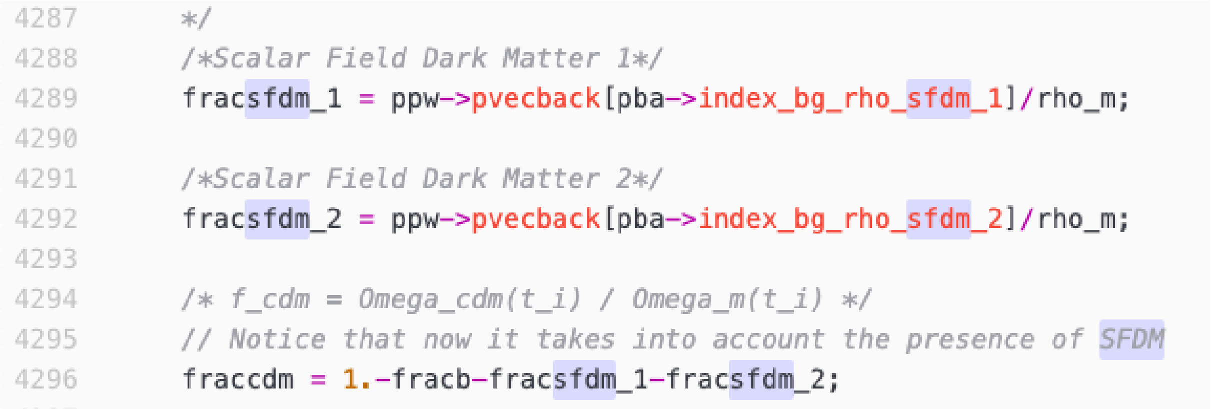
results of the differential equations in local variables for both fields, see Figure G.26. In Figure G.27, we tell CLASS what quantities to put in the output file. These correspond to the local variables we just defined.

Since we are going to enter a new function where the differential equations are going to be defined, we must declare other local variables, see Figure G.28. Finally, we write the differential equations, see Figures G.29 and G.30.



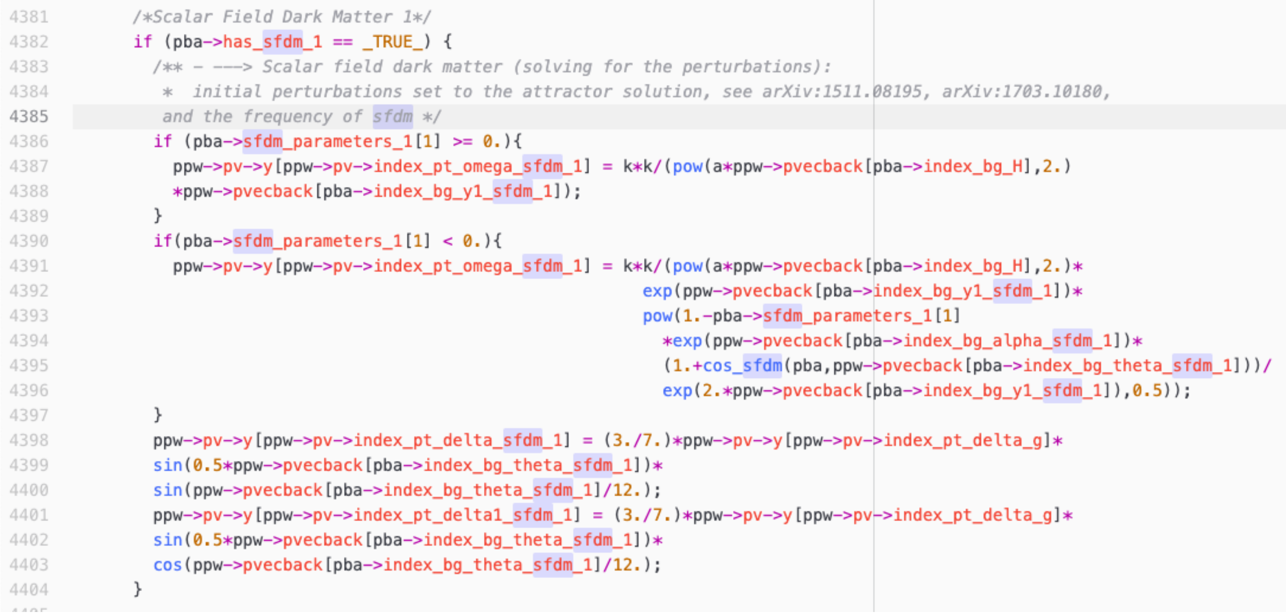
```
    /* Include sfdm as part of the matter or radiation budget */
    /*Scalar Field Dark Matter 1*/
    if (pba->has_sfdm_1 == _TRUE_) {
      if (pba->sfdm_parameters_1[1] < 0.){
       rho_r += ppw->pvecback[pba->index_bg_rho_sfdm_1];
      }
      if(pba->sfdm_parameters_1[1] >= 0.){
       rho_m += ppw->pvecback[pba->index_bg_rho_sfdm_1];
      }
    }

    /*Scalar Field Dark Matter 2*/
    if (pba->has_sfdm_2 == _TRUE_) {
      if (pba->sfdm_parameters_2[1] < 0.){
       rho_r += ppw->pvecback[pba->index_bg_rho_sfdm_2];
      }
      if(pba->sfdm_parameters_2[1] >= 0.){
       rho_m += ppw->pvecback[pba->index_bg_rho_sfdm_2];
      }
    }

```

Figure G.12: Adding the scalar fields to the matter and radiation budget.



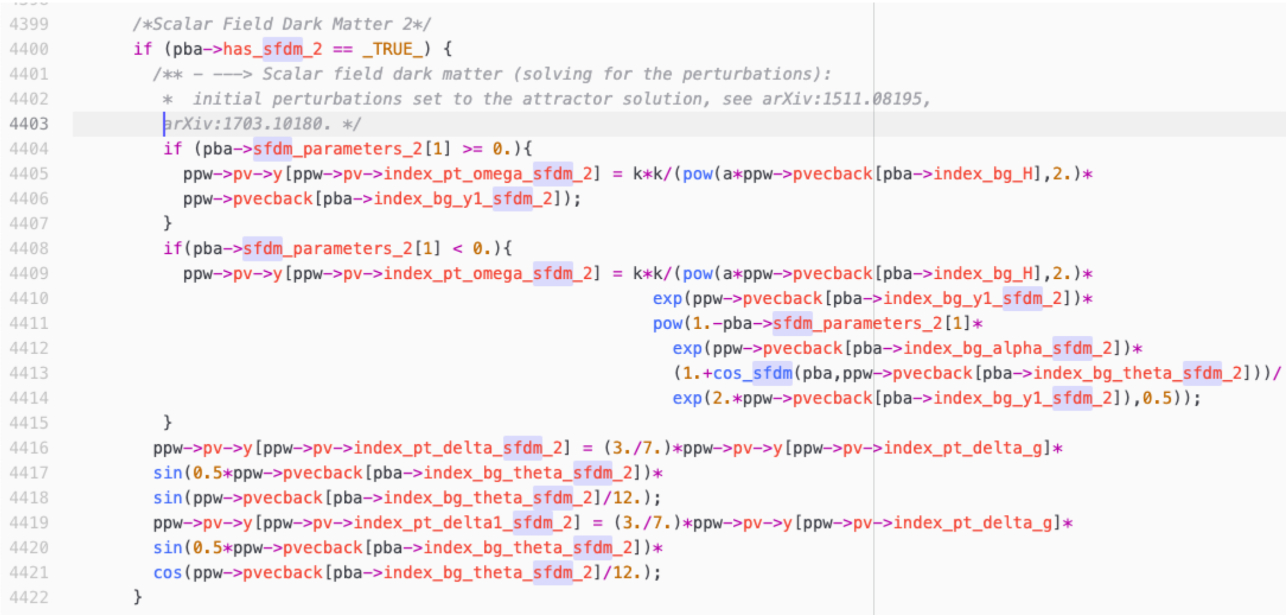
```
    */
    /*Scalar Field Dark Matter 1*/
    fracsfdm_1 = ppw->pvecback[pba->index_bg_rho_sfdm_1]/rho_m;

    /*Scalar Field Dark Matter 2*/
    fracsfdm_2 = ppw->pvecback[pba->index_bg_rho_sfdm_2]/rho_m;

    /* f_cdm = Omega_cdm(t_i) / Omega_m(t_i) */
    // Notice that now it takes into account the presence of SFDM
    fraccdm = 1.-fracb-fracsfdm_1-fracsfdm_2;
```

Figure G.13: Defining the fraction of each scalar field dark matter.



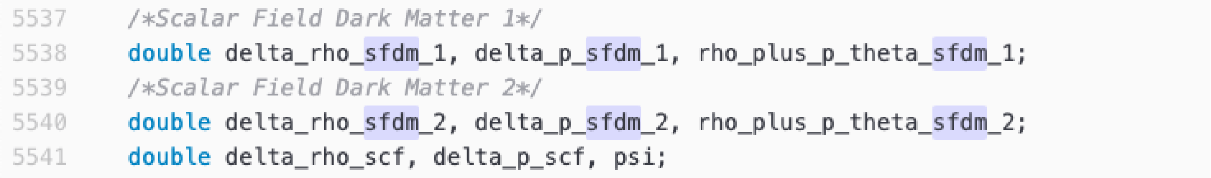
```
      /*Scalar Field Dark Matter 1*/
      if (pba->has_sfdm_1 == _TRUE_) {
        /** - ---> Scalar field dark matter (solving for the perturbations):
         *  initial perturbations set to the attractor solution, see arXiv:1511.08195, arXiv:1703.10180,
         and the frequency of sfdm */
        if (pba->sfdm_parameters_1[1] >= 0.){
          ppw->pv->y[ppw->pv->index_pt_omega_sfdm_1] = k*k/(pow(a*ppw->pvecback[pba->index_bg_H],2.)
          *ppw->pvecback[pba->index_bg_y1_sfdm_1]);
        }
        if(pba->sfdm_parameters_1[1] < 0.){
          ppw->pv->y[ppw->pv->index_pt_omega_sfdm_1] = k*k/(pow(a*ppw->pvecback[pba->index_bg_H],2.)*
                                                         exp(ppw->pvecback[pba->index_bg_y1_sfdm_1])*
                                                         pow(1.-pba->sfdm_parameters_1[1]
                                                           *exp(ppw->pvecback[pba->index_bg_alpha_sfdm_1])*
                                                           (1.+cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1]))/
                                                           exp(2.*ppw->pvecback[pba->index_bg_y1_sfdm_1]),0.5));
        }
        ppw->pv->y[ppw->pv->index_pt_delta_sfdm_1] = (3./7.)*ppw->pv->y[ppw->pv->index_pt_delta_g]*
        sin(0.5*ppw->pvecback[pba->index_bg_theta_sfdm_1])*
        sin(ppw->pvecback[pba->index_bg_theta_sfdm_1]/12.);
        ppw->pv->y[ppw->pv->index_pt_delta1_sfdm_1] = (3./7.)*ppw->pv->y[ppw->pv->index_pt_delta_g]*
        sin(0.5*ppw->pvecback[pba->index_bg_theta_sfdm_1])*
        cos(ppw->pvecback[pba->index_bg_theta_sfdm_1]/12.);
      }
```

Figure G.14: Initial conditions for the scalar field dark matter 1 perturbations.



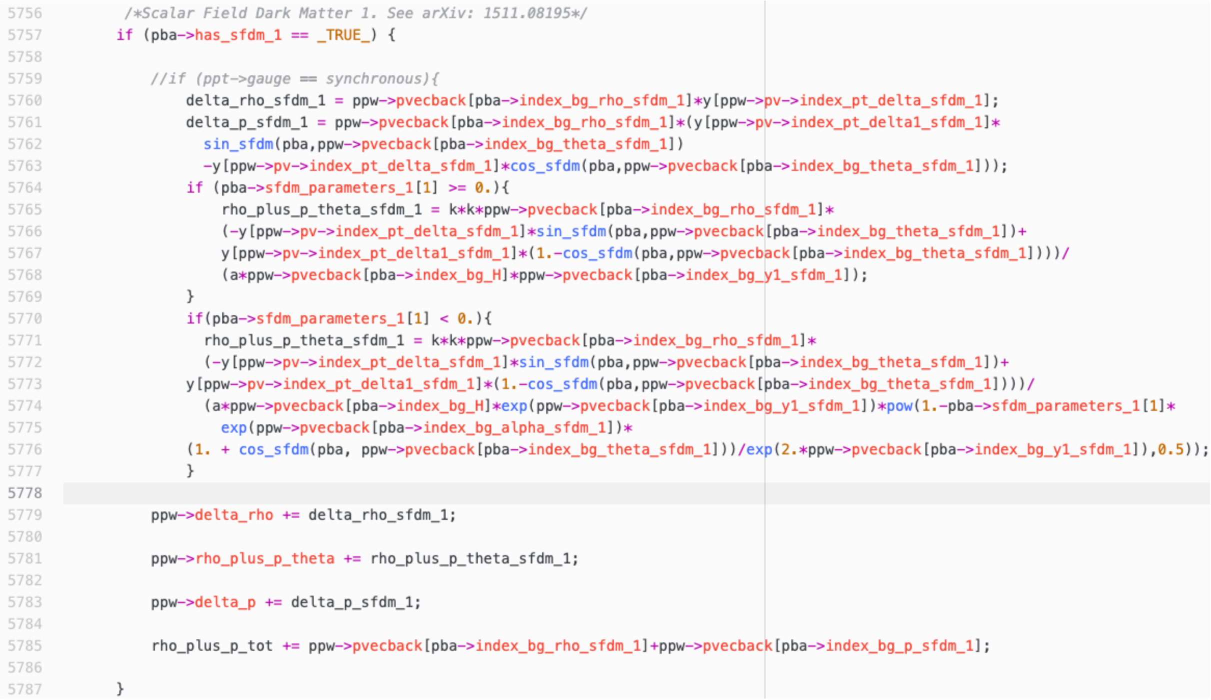
```
      /*Scalar Field Dark Matter 2*/
      if (pba->has_sfdm_2 == _TRUE_) {
        /** - ---> Scalar field dark matter (solving for the perturbations):
         *  initial perturbations set to the attractor solution, see arXiv:1511.08195,
         arXiv:1703.10180. */
         if (pba->sfdm_parameters_2[1] >= 0.){
           ppw->pv->y[ppw->pv->index_pt_omega_sfdm_2] = k*k/(pow(a*ppw->pvecback[pba->index_bg_H],2.)*
           ppw->pvecback[pba->index_bg_y1_sfdm_2]);
         }
         if(pba->sfdm_parameters_2[1] < 0.){
           ppw->pv->y[ppw->pv->index_pt_omega_sfdm_2] = k*k/(pow(a*ppw->pvecback[pba->index_bg_H],2.)*
                                                          exp(ppw->pvecback[pba->index_bg_y1_sfdm_2])*
                                                          pow(1.-pba->sfdm_parameters_2[1]*
                                                            exp(ppw->pvecback[pba->index_bg_alpha_sfdm_2])*
                                                            (1.+cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2]))/
                                                            exp(2.*ppw->pvecback[pba->index_bg_y1_sfdm_2]),0.5));
         }
        ppw->pv->y[ppw->pv->index_pt_delta_sfdm_2] = (3./7.)*ppw->pv->y[ppw->pv->index_pt_delta_g]*
        sin(0.5*ppw->pvecback[pba->index_bg_theta_sfdm_2])*
        sin(ppw->pvecback[pba->index_bg_theta_sfdm_2]/12.);
        ppw->pv->y[ppw->pv->index_pt_delta1_sfdm_2] = (3./7.)*ppw->pv->y[ppw->pv->index_pt_delta_g]*
        sin(0.5*ppw->pvecback[pba->index_bg_theta_sfdm_2])*
        cos(ppw->pvecback[pba->index_bg_theta_sfdm_2]/12.);
      }
```

Figure G.15: Initial conditions for the scalar field dark matter 2 perturbations.



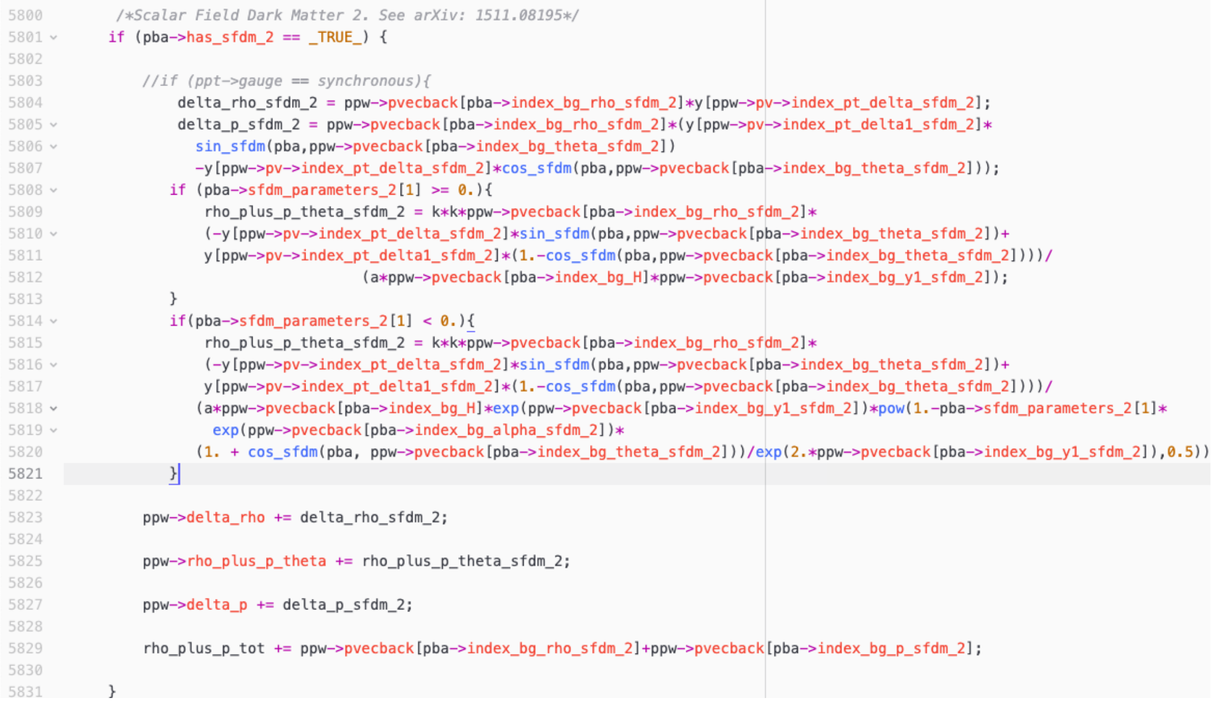
```
/*Scalar Field Dark Matter 1*/
double delta_rho_sfdm_1, delta_p_sfdm_1, rho_plus_p_theta_sfdm_1;
/*Scalar Field Dark Matter 2*/
double delta_rho_sfdm_2, delta_p_sfdm_2, rho_plus_p_theta_sfdm_2;
double delta_rho_scf, delta_p_scf, psi;
```

Figure G.16: Declaring local variables.



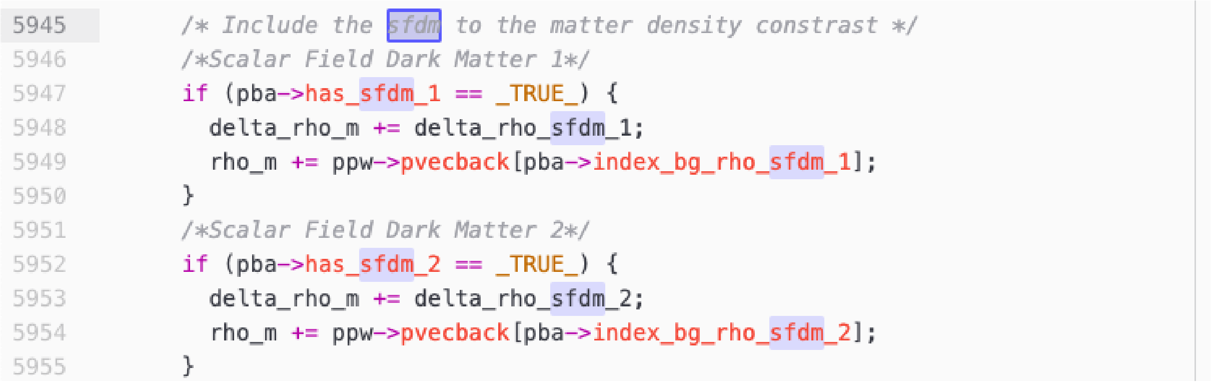
```
 /*Scalar Field Dark Matter 1. See arXiv: 1511.08195*/
if (pba->has_sfdm_1 == _TRUE_) {

    //if (ppt->gauge == synchronous){
        delta_rho_sfdm_1 = ppw->pvecback[pba->index_bg_rho_sfdm_1]*y[ppw->pv->index_pt_delta_sfdm_1];
        delta_p_sfdm_1 = ppw->pvecback[pba->index_bg_rho_sfdm_1]*(y[ppw->pv->index_pt_delta1_sfdm_1]*
          sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])
          -y[ppw->pv->index_pt_delta_sfdm_1]*cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1]));
        if (pba->sfdm_parameters_1[1] >= 0.){
            rho_plus_p_theta_sfdm_1 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_1]*
            (-y[ppw->pv->index_pt_delta_sfdm_1]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])+
            y[ppw->pv->index_pt_delta1_sfdm_1]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])))/
            (a*ppw->pvecback[pba->index_bg_H]*ppw->pvecback[pba->index_bg_y1_sfdm_1]);
        }
        if(pba->sfdm_parameters_1[1] < 0.){
          rho_plus_p_theta_sfdm_1 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_1]*
          (-y[ppw->pv->index_pt_delta_sfdm_1]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])+
        y[ppw->pv->index_pt_delta1_sfdm_1]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])))/
          (a*ppw->pvecback[pba->index_bg_H]*exp(ppw->pvecback[pba->index_bg_y1_sfdm_1])*pow(1.-pba->sfdm_parameters_1[1]*
            exp(ppw->pvecback[pba->index_bg_alpha_sfdm_1])*
        (1. + cos_sfdm(pba, ppw->pvecback[pba->index_bg_theta_sfdm_1]))/exp(2.*ppw->pvecback[pba->index_bg_y1_sfdm_1]),0.5));
        }

    ppw->delta_rho += delta_rho_sfdm_1;

    ppw->rho_plus_p_theta += rho_plus_p_theta_sfdm_1;

    ppw->delta_p += delta_p_sfdm_1;

    rho_plus_p_tot += ppw->pvecback[pba->index_bg_rho_sfdm_1]+ppw->pvecback[pba->index_bg_p_sfdm_1];

}
```

Figure G.17: Defining the quantities 3.18a for scalar field dark matter 1.



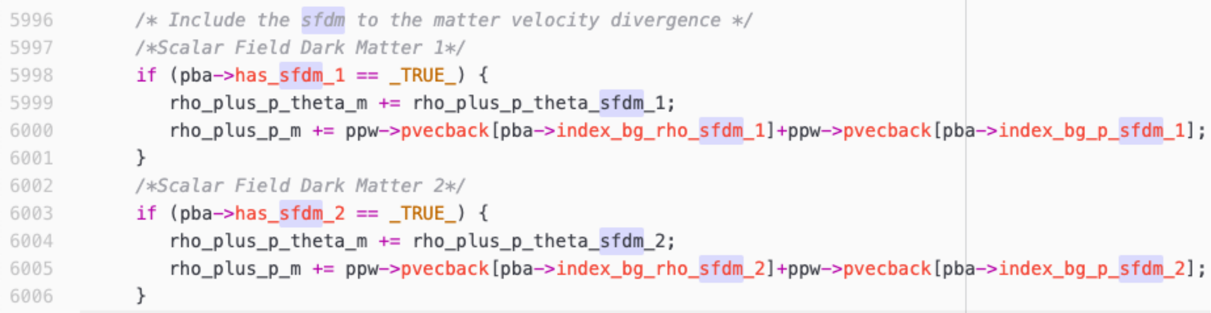
```
 /*Scalar Field Dark Matter 2. See arXiv: 1511.08195*/
if (pba->has_sfdm_2 == _TRUE_) {

    //if (ppt->gauge == synchronous){
        delta_rho_sfdm_2 = ppw->pvecback[pba->index_bg_rho_sfdm_2]*y[ppw->pv->index_pt_delta_sfdm_2];
        delta_p_sfdm_2 = ppw->pvecback[pba->index_bg_rho_sfdm_2]*(y[ppw->pv->index_pt_delta1_sfdm_2]*
          sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])
          -y[ppw->pv->index_pt_delta_sfdm_2]*cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2]));
       if (pba->sfdm_parameters_2[1] >= 0.){
           rho_plus_p_theta_sfdm_2 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_2]*
           (-y[ppw->pv->index_pt_delta_sfdm_2]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])+
           y[ppw->pv->index_pt_delta1_sfdm_2]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])))/
                             (a*ppw->pvecback[pba->index_bg_H]*ppw->pvecback[pba->index_bg_y1_sfdm_2]);
       }
       if(pba->sfdm_parameters_2[1] < 0.){
           rho_plus_p_theta_sfdm_2 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_2]*
           (-y[ppw->pv->index_pt_delta_sfdm_2]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])+
           y[ppw->pv->index_pt_delta1_sfdm_2]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])))/
          (a*ppw->pvecback[pba->index_bg_H]*exp(ppw->pvecback[pba->index_bg_y1_sfdm_2])*pow(1.-pba->sfdm_parameters_2[1]*
            exp(ppw->pvecback[pba->index_bg_alpha_sfdm_2])*
          (1. + cos_sfdm(pba, ppw->pvecback[pba->index_bg_theta_sfdm_2]))/exp(2.*ppw->pvecback[pba->index_bg_y1_sfdm_2]),0.5))
       }

    ppw->delta_rho += delta_rho_sfdm_2;

    ppw->rho_plus_p_theta += rho_plus_p_theta_sfdm_2;

    ppw->delta_p += delta_p_sfdm_2;

    rho_plus_p_tot += ppw->pvecback[pba->index_bg_rho_sfdm_2]+ppw->pvecback[pba->index_bg_p_sfdm_2];

}
```

Figure G.18: Defining the quantities 3.18a for scalar field dark matter 2.



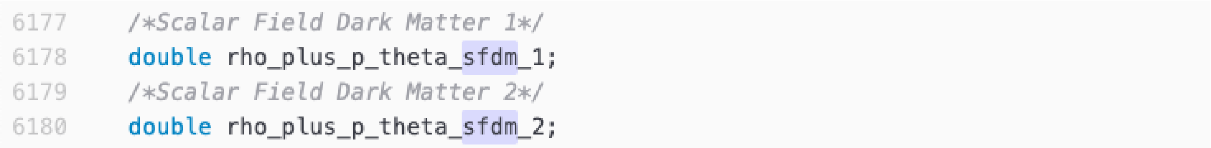
```
/* Include the sfdm to the matter density constrast */
/*Scalar Field Dark Matter 1*/
if (pba->has_sfdm_1 == _TRUE_) {
  delta_rho_m += delta_rho_sfdm_1;
  rho_m += ppw->pvecback[pba->index_bg_rho_sfdm_1];
}
/*Scalar Field Dark Matter 2*/
if (pba->has_sfdm_2 == _TRUE_) {
  delta_rho_m += delta_rho_sfdm_2;
  rho_m += ppw->pvecback[pba->index_bg_rho_sfdm_2];
}
```

Figure G.19: Adding the scalar field density contrast to the total density contrast.



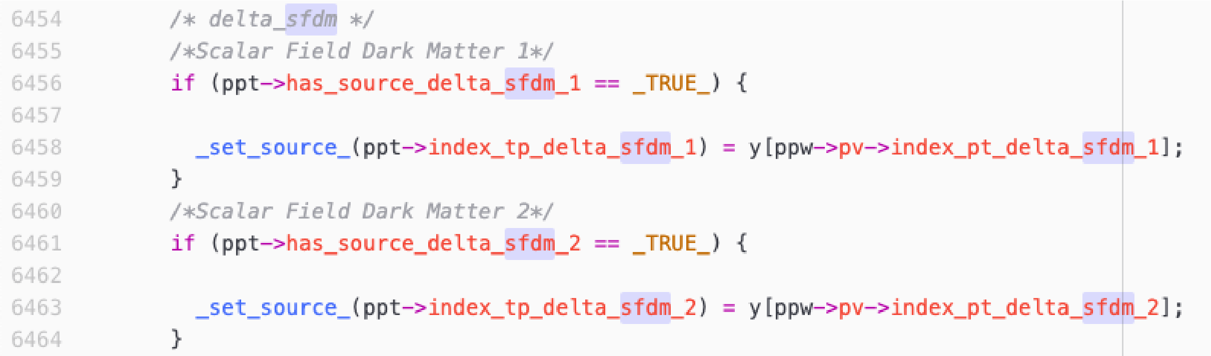
```
    /* Include the sfdm to the matter velocity divergence */
    /*Scalar Field Dark Matter 1*/
    if (pba->has_sfdm_1 == _TRUE_) {
      rho_plus_p_theta_m += rho_plus_p_theta_sfdm_1;
      rho_plus_p_m += ppw->pvecback[pba->index_bg_rho_sfdm_1]+ppw->pvecback[pba->index_bg_p_sfdm_1];
    }
    /*Scalar Field Dark Matter 2*/
    if (pba->has_sfdm_2 == _TRUE_) {
      rho_plus_p_theta_m += rho_plus_p_theta_sfdm_2;
      rho_plus_p_m += ppw->pvecback[pba->index_bg_rho_sfdm_2]+ppw->pvecback[pba->index_bg_p_sfdm_2];
    }
```

Figure G.20: Including the scalar fields to the matter velocity divergence.



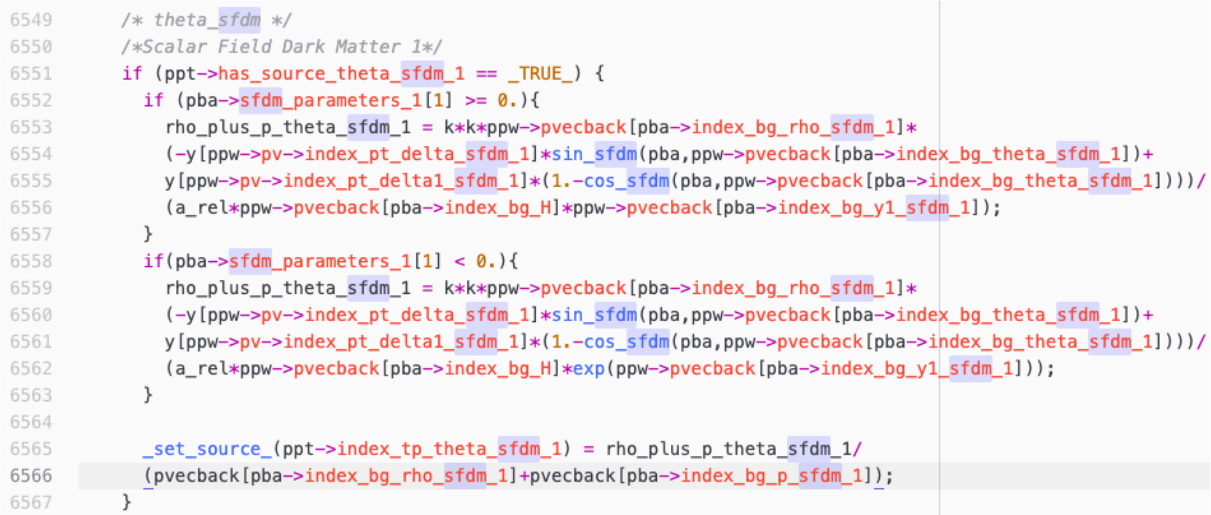
```
  /*Scalar Field Dark Matter 1*/
  double rho_plus_p_theta_sfdm_1;
  /*Scalar Field Dark Matter 2*/
  double rho_plus_p_theta_sfdm_2;
```

Figure G.21: Declaring variables.



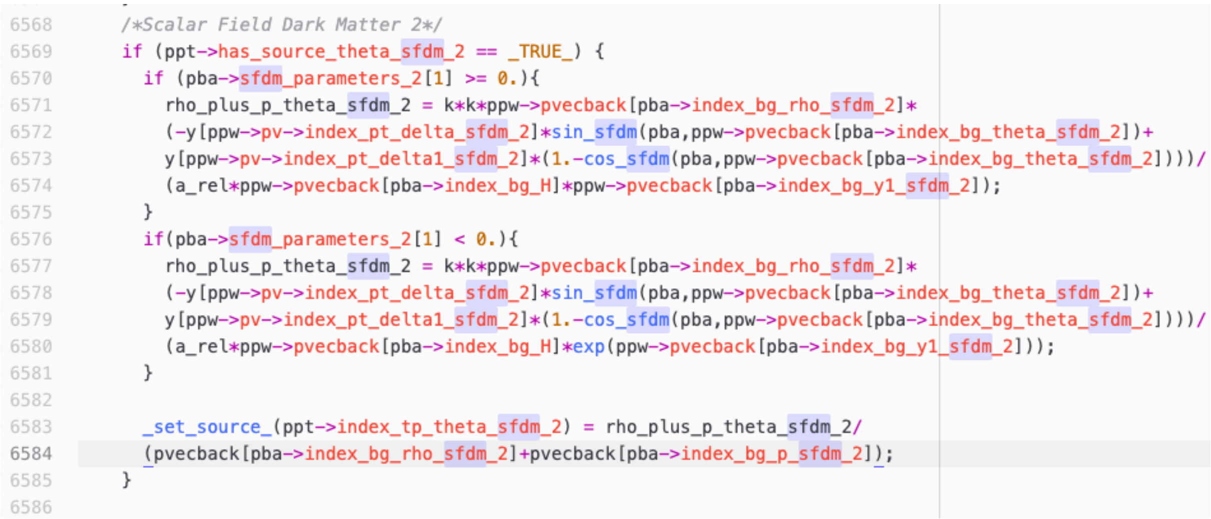
```
      /* delta_sfdm */
      /*Scalar Field Dark Matter 1*/
      if (ppt->has_source_delta_sfdm_1 == _TRUE_) {

        _set_source_(ppt->index_tp_delta_sfdm_1) = y[ppw->pv->index_pt_delta_sfdm_1];
      }
      /*Scalar Field Dark Matter 2*/
      if (ppt->has_source_delta_sfdm_2 == _TRUE_) {

        _set_source_(ppt->index_tp_delta_sfdm_2) = y[ppw->pv->index_pt_delta_sfdm_2];
      }
```

Figure G.22: Including the scalar fields contributions to the perturbations.



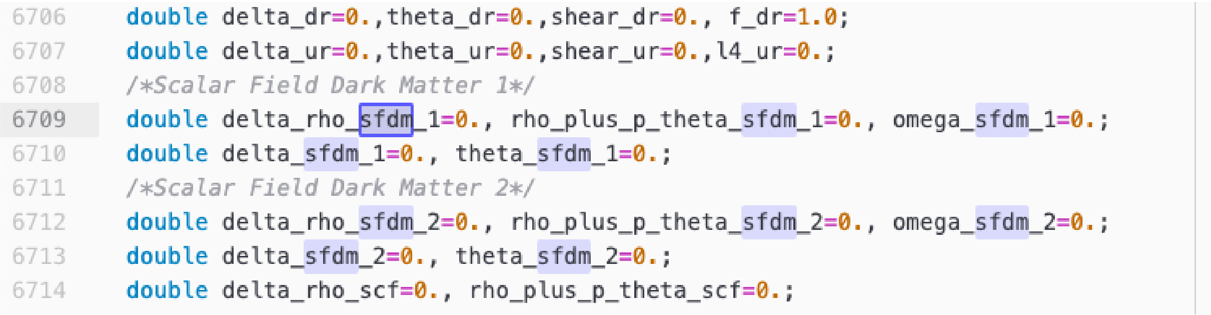
```
/* theta_sfdm */
/*Scalar Field Dark Matter 1*/
if (ppt->has_source_theta_sfdm_1 == _TRUE_) {
  if (pba->sfdm_parameters_1[1] >= 0.){
    rho_plus_p_theta_sfdm_1 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_1]*
    (-y[ppw->pv->index_pt_delta_sfdm_1]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])+
    y[ppw->pv->index_pt_delta1_sfdm_1]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])))/
    (a_rel*ppw->pvecback[pba->index_bg_H]*ppw->pvecback[pba->index_bg_y1_sfdm_1]);
  }
  if(pba->sfdm_parameters_1[1] < 0.){
    rho_plus_p_theta_sfdm_1 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_1]*
    (-y[ppw->pv->index_pt_delta_sfdm_1]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])+
    y[ppw->pv->index_pt_delta1_sfdm_1]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])))/
    (a_rel*ppw->pvecback[pba->index_bg_H]*exp(ppw->pvecback[pba->index_bg_y1_sfdm_1]));
  }

  _set_source_(ppt->index_tp_theta_sfdm_1) = rho_plus_p_theta_sfdm_1/
  (pvecback[pba->index_bg_rho_sfdm_1]+pvecback[pba->index_bg_p_sfdm_1]);
}
```

Figure G.23: Defining the quantity 3.18c for scalar field dark matter 1.



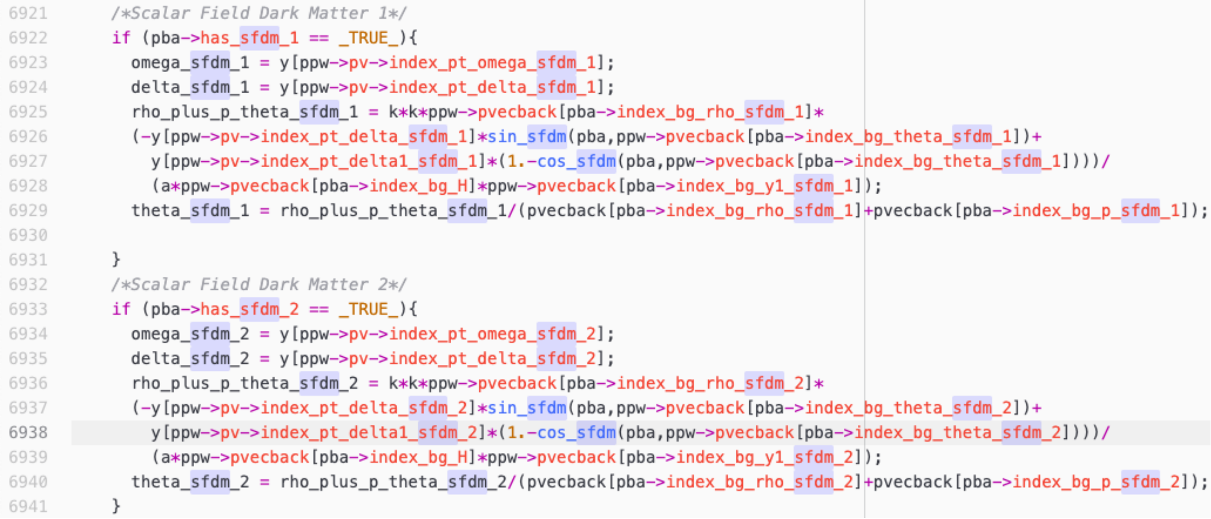
```
/*Scalar Field Dark Matter 2*/
if (ppt->has_source_theta_sfdm_2 == _TRUE_) {
  if (pba->sfdm_parameters_2[1] >= 0.){
    rho_plus_p_theta_sfdm_2 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_2]*
    (-y[ppw->pv->index_pt_delta_sfdm_2]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])+
    y[ppw->pv->index_pt_delta1_sfdm_2]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])))/
    (a_rel*ppw->pvecback[pba->index_bg_H]*ppw->pvecback[pba->index_bg_y1_sfdm_2]);
  }
  if(pba->sfdm_parameters_2[1] < 0.){
    rho_plus_p_theta_sfdm_2 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_2]*
    (-y[ppw->pv->index_pt_delta_sfdm_2]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])+
    y[ppw->pv->index_pt_delta1_sfdm_2]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])))/
    (a_rel*ppw->pvecback[pba->index_bg_H]*exp(ppw->pvecback[pba->index_bg_y1_sfdm_2]));
  }

  _set_source_(ppt->index_tp_theta_sfdm_2) = rho_plus_p_theta_sfdm_2/
  (pvecback[pba->index_bg_rho_sfdm_2]+pvecback[pba->index_bg_p_sfdm_2]);
}
```

Figure G.24: Defining the quantity 3.18c for scalar field dark matter 2.



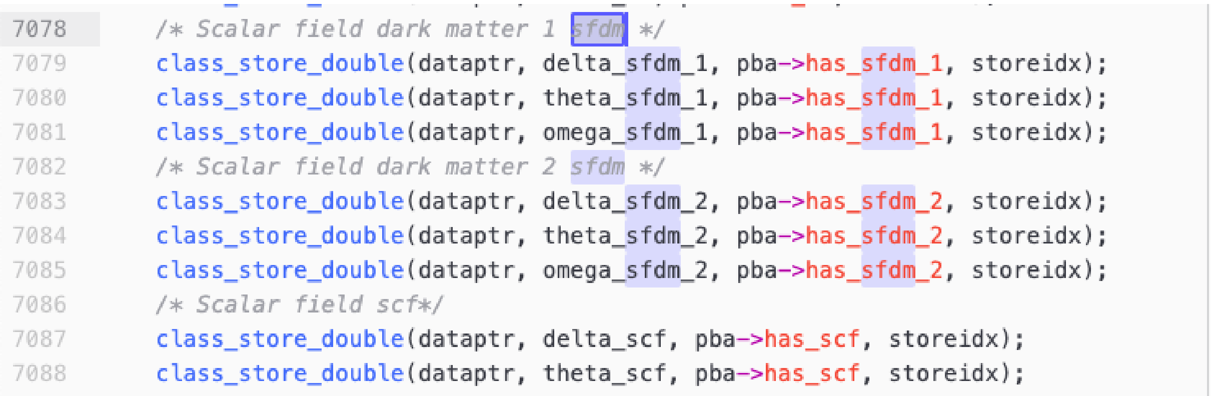
```
double delta_dr=0.,theta_dr=0.,shear_dr=0., f_dr=1.0;
double delta_ur=0.,theta_ur=0.,shear_ur=0.,l4_ur=0.;
/*Scalar Field Dark Matter 1*/
double delta_rho_sfdm_1=0., rho_plus_p_theta_sfdm_1=0., omega_sfdm_1=0.;
double delta_sfdm_1=0., theta_sfdm_1=0.;
/*Scalar Field Dark Matter 2*/
double delta_rho_sfdm_2=0., rho_plus_p_theta_sfdm_2=0., omega_sfdm_2=0.;
double delta_sfdm_2=0., theta_sfdm_2=0.;
double delta_rho_scf=0., rho_plus_p_theta_scf=0.;
```

Figure G.25: Declaring to zero the fields local variables.



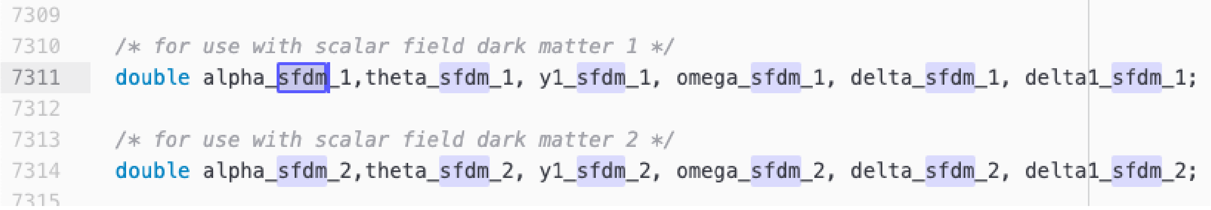
```
/*Scalar Field Dark Matter 1*/
if (pba->has_sfdm_1 == _TRUE_){
  omega_sfdm_1 = y[ppw->pv->index_pt_omega_sfdm_1];
  delta_sfdm_1 = y[ppw->pv->index_pt_delta_sfdm_1];
  rho_plus_p_theta_sfdm_1 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_1]*
  (-y[ppw->pv->index_pt_delta_sfdm_1]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])+
    y[ppw->pv->index_pt_delta1_sfdm_1]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_1])))/
    (a*ppw->pvecback[pba->index_bg_H]*ppw->pvecback[pba->index_bg_y1_sfdm_1]);
  theta_sfdm_1 = rho_plus_p_theta_sfdm_1/(pvecback[pba->index_bg_rho_sfdm_1]+pvecback[pba->index_bg_p_sfdm_1]);

}
/*Scalar Field Dark Matter 2*/
if (pba->has_sfdm_2 == _TRUE_){
  omega_sfdm_2 = y[ppw->pv->index_pt_omega_sfdm_2];
  delta_sfdm_2 = y[ppw->pv->index_pt_delta_sfdm_2];
  rho_plus_p_theta_sfdm_2 = k*k*ppw->pvecback[pba->index_bg_rho_sfdm_2]*
  (-y[ppw->pv->index_pt_delta_sfdm_2]*sin_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])+
    y[ppw->pv->index_pt_delta1_sfdm_2]*(1.-cos_sfdm(pba,ppw->pvecback[pba->index_bg_theta_sfdm_2])))/
    (a*ppw->pvecback[pba->index_bg_H]*ppw->pvecback[pba->index_bg_y1_sfdm_2]);
  theta_sfdm_2 = rho_plus_p_theta_sfdm_2/(pvecback[pba->index_bg_rho_sfdm_2]+pvecback[pba->index_bg_p_sfdm_2]);
}
```

Figure G.26: Storing the results of the differential equations in local variables for both fields.



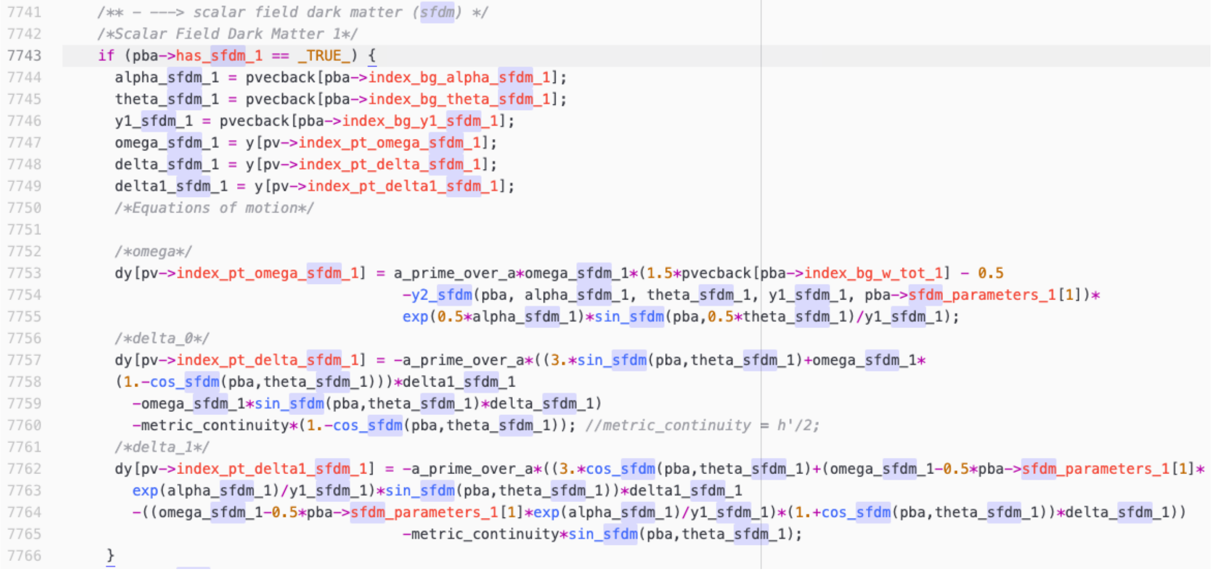
```
/* Scalar field dark matter 1 sfdm */
class_store_double(dataptr, delta_sfdm_1, pba->has_sfdm_1, storeidx);
class_store_double(dataptr, theta_sfdm_1, pba->has_sfdm_1, storeidx);
class_store_double(dataptr, omega_sfdm_1, pba->has_sfdm_1, storeidx);
/* Scalar field dark matter 2 sfdm */
class_store_double(dataptr, delta_sfdm_2, pba->has_sfdm_2, storeidx);
class_store_double(dataptr, theta_sfdm_2, pba->has_sfdm_2, storeidx);
class_store_double(dataptr, omega_sfdm_2, pba->has_sfdm_2, storeidx);
/* Scalar field scf*/
class_store_double(dataptr, delta_scf, pba->has_scf, storeidx);
class_store_double(dataptr, theta_scf, pba->has_scf, storeidx);
```

Figure G.27: Telling CLASS what quantities to put in the output file.



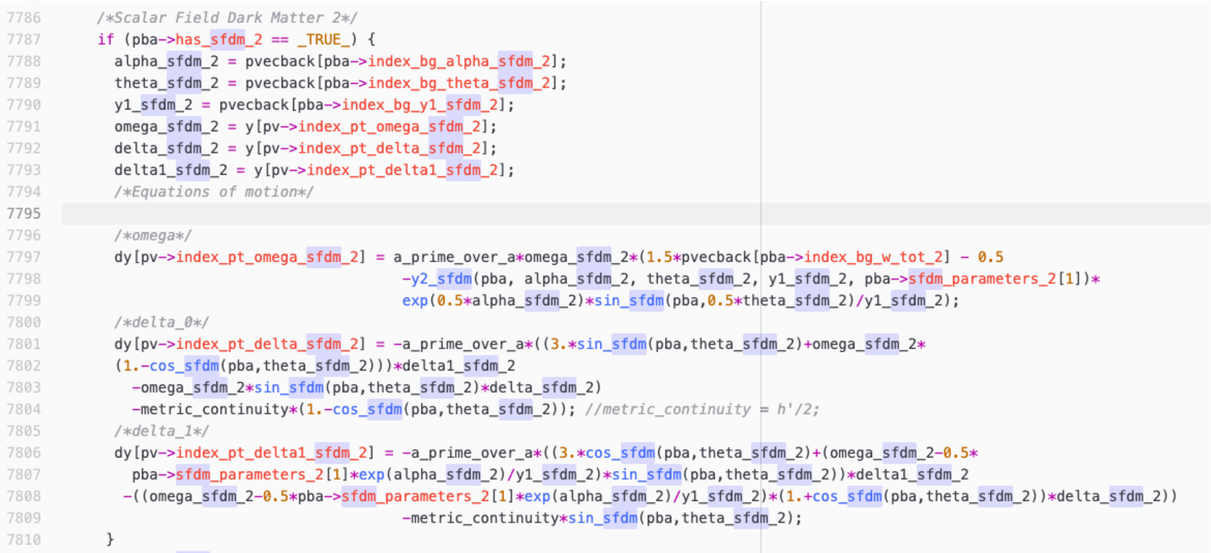
```
/* for use with scalar field dark matter 1 */
double alpha_sfdm_1,theta_sfdm_1, y1_sfdm_1, omega_sfdm_1, delta_sfdm_1, delta1_sfdm_1;

/* for use with scalar field dark matter 2 */
double alpha_sfdm_2,theta_sfdm_2, y1_sfdm_2, omega_sfdm_2, delta_sfdm_2, delta1_sfdm_2;
```

Figure G.28: Declaring local variables.



```
  /** - ---> scalar field dark matter (sfdm) */
  /*Scalar Field Dark Matter 1*/
  if (pba->has_sfdm_1 == _TRUE_) {
    alpha_sfdm_1 = pvecback[pba->index_bg_alpha_sfdm_1];
    theta_sfdm_1 = pvecback[pba->index_bg_theta_sfdm_1];
    y1_sfdm_1 = pvecback[pba->index_bg_y1_sfdm_1];
    omega_sfdm_1 = y[pv->index_pt_omega_sfdm_1];
    delta_sfdm_1 = y[pv->index_pt_delta_sfdm_1];
    delta1_sfdm_1 = y[pv->index_pt_delta1_sfdm_1];
    /*Equations of motion*/

    /*omega*/
    dy[pv->index_pt_omega_sfdm_1] = a_prime_over_a*omega_sfdm_1*(1.5*pvecback[pba->index_bg_w_tot_1] - 0.5
                                    -y2_sfdm(pba, alpha_sfdm_1, theta_sfdm_1, y1_sfdm_1, pba->sfdm_parameters_1[1])*
                                    exp(0.5*alpha_sfdm_1)*sin_sfdm(pba,0.5*theta_sfdm_1)/y1_sfdm_1);
    /*delta_0*/
    dy[pv->index_pt_delta_sfdm_1] = -a_prime_over_a*((3.*sin_sfdm(pba,theta_sfdm_1)+omega_sfdm_1*
    (1.-cos_sfdm(pba,theta_sfdm_1)))*delta1_sfdm_1
      -omega_sfdm_1*sin_sfdm(pba,theta_sfdm_1)*delta_sfdm_1)
      -metric_continuity*(1.-cos_sfdm(pba,theta_sfdm_1)); //metric_continuity = h'/2;
    /*delta_1*/
    dy[pv->index_pt_delta1_sfdm_1] = -a_prime_over_a*((3.*cos_sfdm(pba,theta_sfdm_1)+(omega_sfdm_1-0.5*pba->sfdm_parameters_1[1]*
      exp(alpha_sfdm_1)/y1_sfdm_1)*sin_sfdm(pba,theta_sfdm_1))*delta1_sfdm_1
      -((omega_sfdm_1-0.5*pba->sfdm_parameters_1[1]*exp(alpha_sfdm_1)/y1_sfdm_1)*(1.+cos_sfdm(pba,theta_sfdm_1))*delta_sfdm_1))
                                    -metric_continuity*sin_sfdm(pba,theta_sfdm_1);
 }
```

Figure G.29: Writing the differential equations for scalar field dark matter 1.



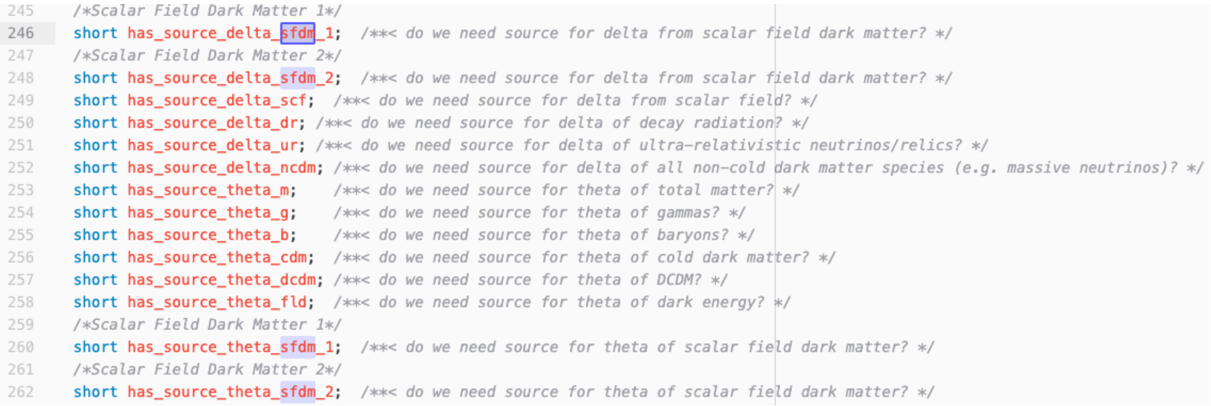
```
    /*Scalar Field Dark Matter 2*/
    if (pba->has_sfdm_2 == _TRUE_) {
      alpha_sfdm_2 = pvecback[pba->index_bg_alpha_sfdm_2];
      theta_sfdm_2 = pvecback[pba->index_bg_theta_sfdm_2];
      y1_sfdm_2 = pvecback[pba->index_bg_y1_sfdm_2];
      omega_sfdm_2 = y[pv->index_pt_omega_sfdm_2];
      delta_sfdm_2 = y[pv->index_pt_delta_sfdm_2];
      delta1_sfdm_2 = y[pv->index_pt_delta1_sfdm_2];
      /*Equations of motion*/

      /*omega*/
      dy[pv->index_pt_omega_sfdm_2] = a_prime_over_a*omega_sfdm_2*(1.5*pvecback[pba->index_bg_w_tot_2] - 0.5
                                       -y2_sfdm(pba, alpha_sfdm_2, theta_sfdm_2, y1_sfdm_2, pba->sfdm_parameters_2[1])*
                                       exp(0.5*alpha_sfdm_2)*sin_sfdm(pba,0.5*theta_sfdm_2)/y1_sfdm_2);
      /*delta_0*/
      dy[pv->index_pt_delta_sfdm_2] = -a_prime_over_a*((3.*sin_sfdm(pba,theta_sfdm_2)+omega_sfdm_2*
      (1.-cos_sfdm(pba,theta_sfdm_2)))*delta1_sfdm_2
        -omega_sfdm_2*sin_sfdm(pba,theta_sfdm_2)*delta_sfdm_2)
        -metric_continuity*(1.-cos_sfdm(pba,theta_sfdm_2)); //metric_continuity = h'/2;
      /*delta_1*/
      dy[pv->index_pt_delta1_sfdm_2] = -a_prime_over_a*((3.*cos_sfdm(pba,theta_sfdm_2)+(omega_sfdm_2-0.5*
        pba->sfdm_parameters_2[1]*exp(alpha_sfdm_2)/y1_sfdm_2)*sin_sfdm(pba,theta_sfdm_2))*delta1_sfdm_2
       -((omega_sfdm_2-0.5*pba->sfdm_parameters_2[1]*exp(alpha_sfdm_2)/y1_sfdm_2)*(1.+cos_sfdm(pba,theta_sfdm_2))*delta_sfdm_2))
                                       -metric_continuity*sin_sfdm(pba,theta_sfdm_2);
   }
```

Figure G.30: Writing the differential equations for scalar field dark matter 2.

Appendix H

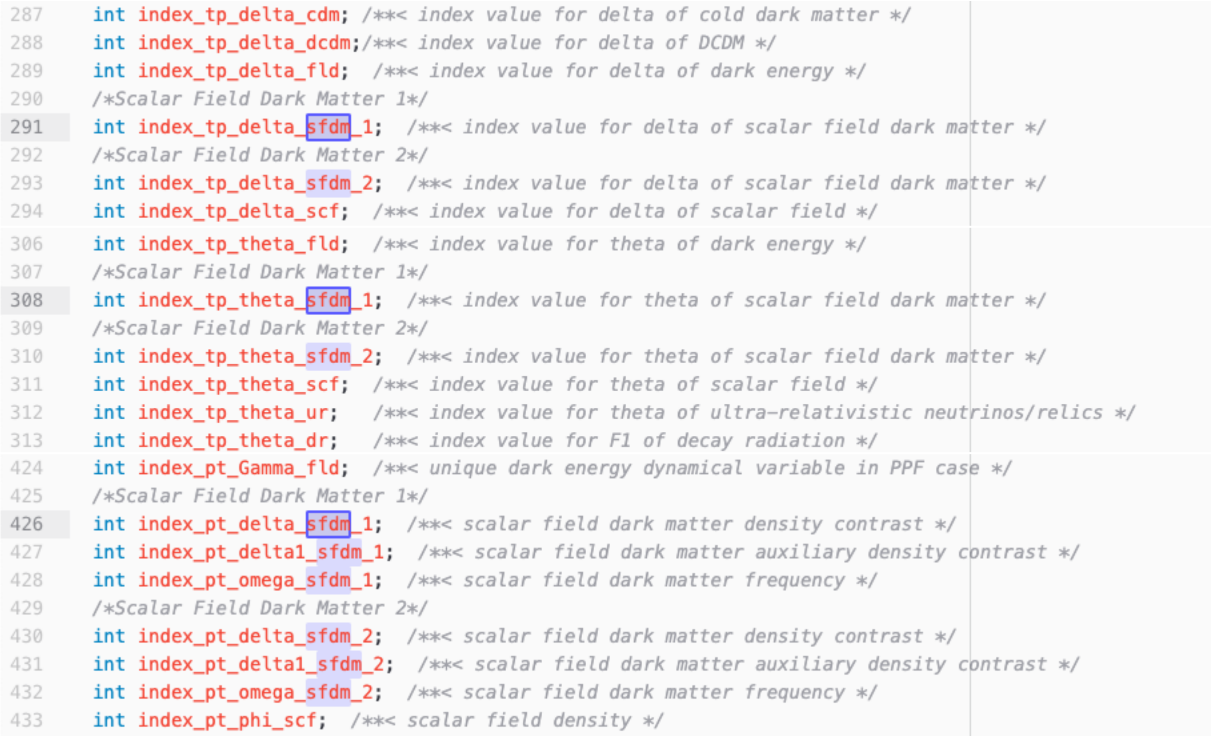
perturbations.h

In this file we declare the variables used in the file perturbations.c. See Figures H.1, H.2



```
/*Scalar Field Dark Matter 1*/
short has_source_delta_sfdm_1;  /**< do we need source for delta from scalar field dark matter? */
/*Scalar Field Dark Matter 2*/
short has_source_delta_sfdm_2;  /**< do we need source for delta from scalar field dark matter? */
short has_source_delta_scf;  /**< do we need source for delta from scalar field? */
short has_source_delta_dr; /**< do we need source for delta of decay radiation? */
short has_source_delta_ur; /**< do we need source for delta of ultra-relativistic neutrinos/relics? */
short has_source_delta_ncdm; /**< do we need source for delta of all non-cold dark matter species (e.g. massive neutrinos)? */
short has_source_theta_m;    /**< do we need source for theta of total matter? */
short has_source_theta_g;    /**< do we need source for theta of gammas? */
short has_source_theta_b;    /**< do we need source for theta of baryons? */
short has_source_theta_cdm;  /**< do we need source for theta of cold dark matter? */
short has_source_theta_dcdm; /**< do we need source for theta of DCDM? */
short has_source_theta_fld;  /**< do we need source for theta of dark energy? */
/*Scalar Field Dark Matter 1*/
short has_source_theta_sfdm_1;  /**< do we need source for theta of scalar field dark matter? */
/*Scalar Field Dark Matter 2*/
short has_source_theta_sfdm_2;  /**< do we need source for theta of scalar field dark matter? */
```

Figure H.1: Declaring flags.



```
287    int index_tp_delta_cdm; /**< index value for delta of cold dark matter */
288    int index_tp_delta_dcdm;/**< index value for delta of DCDM */
289    int index_tp_delta_fld;  /**< index value for delta of dark energy */
290    /*Scalar Field Dark Matter 1*/
291    int index_tp_delta_sfdm_1;  /**< index value for delta of scalar field dark matter */
292    /*Scalar Field Dark Matter 2*/
293    int index_tp_delta_sfdm_2;  /**< index value for delta of scalar field dark matter */
294    int index_tp_delta_scf;  /**< index value for delta of scalar field */
306    int index_tp_theta_fld;  /**< index value for theta of dark energy */
307    /*Scalar Field Dark Matter 1*/
308    int index_tp_theta_sfdm_1;  /**< index value for theta of scalar field dark matter */
309    /*Scalar Field Dark Matter 2*/
310    int index_tp_theta_sfdm_2;  /**< index value for theta of scalar field dark matter */
311    int index_tp_theta_scf;  /**< index value for theta of scalar field */
312    int index_tp_theta_ur;   /**< index value for theta of ultra-relativistic neutrinos/relics */
313    int index_tp_theta_dr;   /**< index value for F1 of decay radiation */
424    int index_pt_Gamma_fld;  /**< unique dark energy dynamical variable in PPF case */
425    /*Scalar Field Dark Matter 1*/
426    int index_pt_delta_sfdm_1;  /**< scalar field dark matter density contrast */
427    int index_pt_delta1_sfdm_1;  /**< scalar field dark matter auxiliary density contrast */
428    int index_pt_omega_sfdm_1;  /**< scalar field dark matter frequency */
429    /*Scalar Field Dark Matter 2*/
430    int index_pt_delta_sfdm_2;  /**< scalar field dark matter density contrast */
431    int index_pt_delta1_sfdm_2;  /**< scalar field dark matter auxiliary density contrast */
432    int index_pt_omega_sfdm_2;  /**< scalar field dark matter frequency */
433    int index_pt_phi_scf;  /**< scalar field density */
```

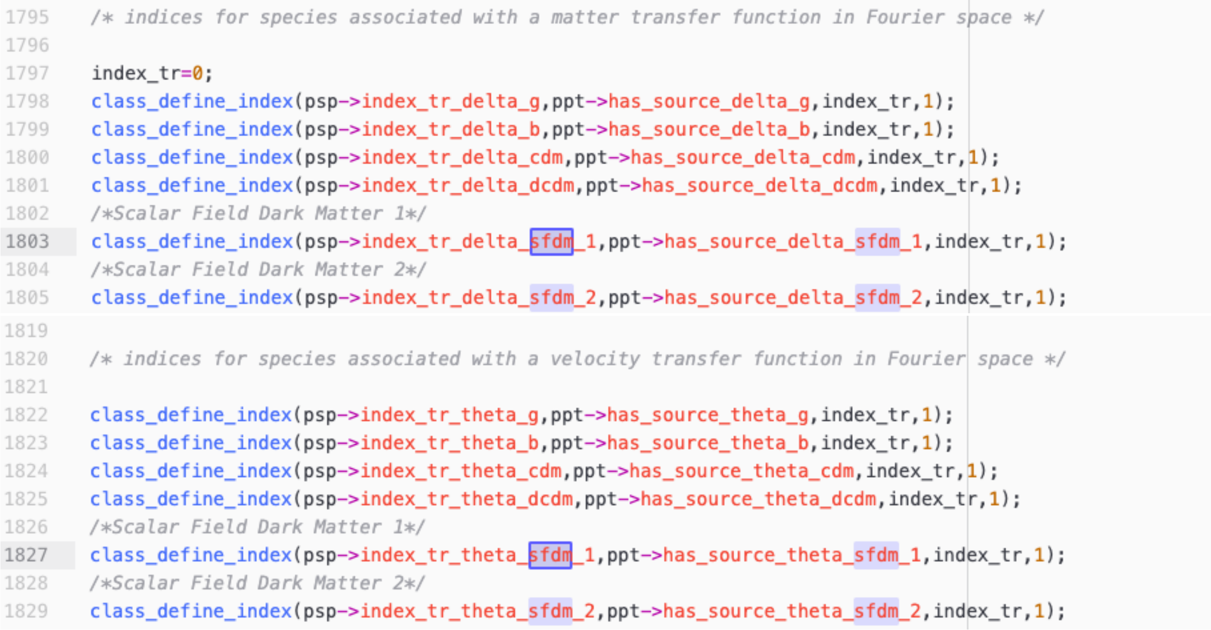
Figure H.2: Declaring indices. Please pay special attention to the code line number.

Appendix I

spectra.c

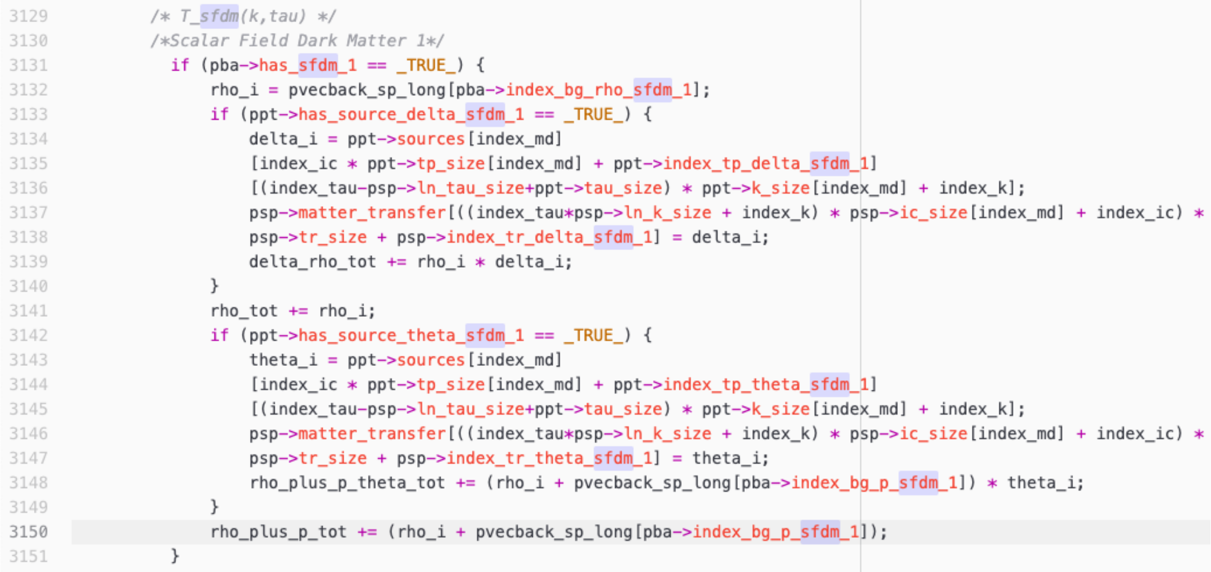
This file computes the anisotropy and Fourier power spectra. Since in our model this is calculated in the same way as in ΛCDM, we just copy and paste the lines of code for the scalar field as dark energy and adapt it to scalar field dark matter. First, we define the indices for the scalar field dark matter, see Figure I.1. If the scalar fields has a non-zero contribution, their transfer function information is added, see Figures I.2 and I.3. Then, we put the name to the columns of the quantities that will be saved in the output file. We can use two different formats, the CLASS format, see Figure I.4, and the CAMB [1] format, see Figure I.5. Using one format or another depends on what we want to do. Then we tell CLASS which variables we want to store, see Figure I.6 for CLASS format and Figure I.7 for CAMB format.

[1] CAMB [170] is CLASS-like code written in fortran and python. The difference between the formats is because the codes use different units.



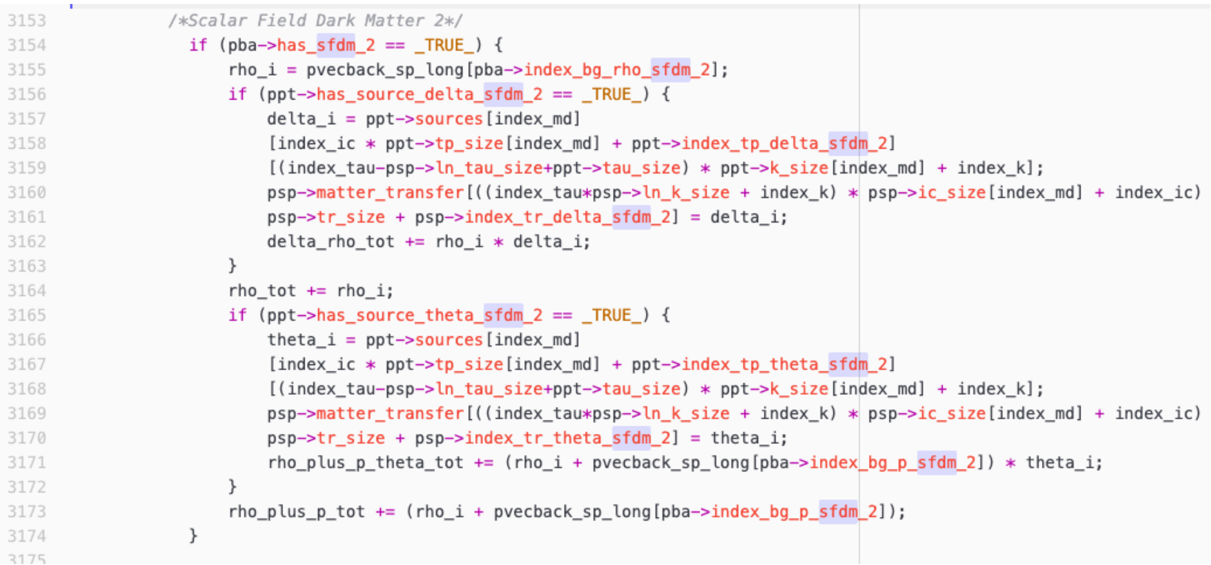
```
/* indices for species associated with a matter transfer function in Fourier space */

index_tr=0;
class_define_index(psp->index_tr_delta_g,ppt->has_source_delta_g,index_tr,1);
class_define_index(psp->index_tr_delta_b,ppt->has_source_delta_b,index_tr,1);
class_define_index(psp->index_tr_delta_cdm,ppt->has_source_delta_cdm,index_tr,1);
class_define_index(psp->index_tr_delta_dcdm,ppt->has_source_delta_dcdm,index_tr,1);
/*Scalar Field Dark Matter 1*/
class_define_index(psp->index_tr_delta_sfdm_1,ppt->has_source_delta_sfdm_1,index_tr,1);
/*Scalar Field Dark Matter 2*/
class_define_index(psp->index_tr_delta_sfdm_2,ppt->has_source_delta_sfdm_2,index_tr,1);

/* indices for species associated with a velocity transfer function in Fourier space */

class_define_index(psp->index_tr_theta_g,ppt->has_source_theta_g,index_tr,1);
class_define_index(psp->index_tr_theta_b,ppt->has_source_theta_b,index_tr,1);
class_define_index(psp->index_tr_theta_cdm,ppt->has_source_theta_cdm,index_tr,1);
class_define_index(psp->index_tr_theta_dcdm,ppt->has_source_theta_dcdm,index_tr,1);
/*Scalar Field Dark Matter 1*/
class_define_index(psp->index_tr_theta_sfdm_1,ppt->has_source_theta_sfdm_1,index_tr,1);
/*Scalar Field Dark Matter 2*/
class_define_index(psp->index_tr_theta_sfdm_2,ppt->has_source_theta_sfdm_2,index_tr,1);
```

Figure I.1: Defining indices.



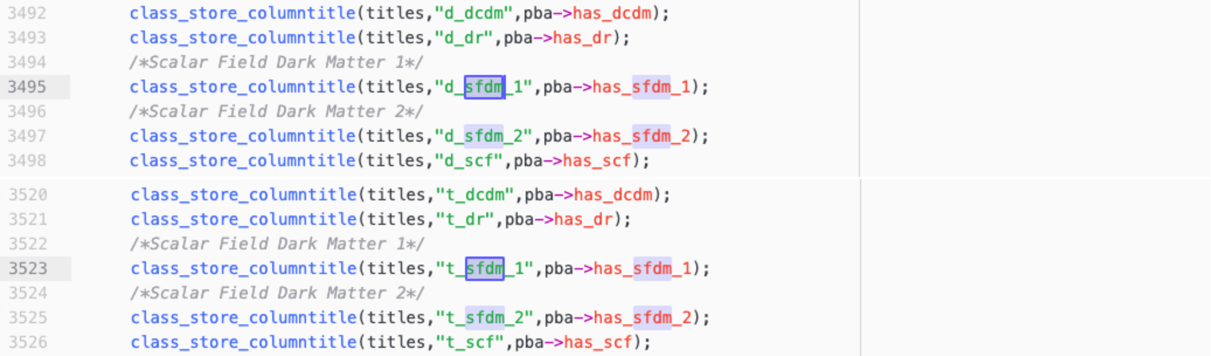
```
    /* T_sfdm(k,tau) */
    /*Scalar Field Dark Matter 1*/
     if (pba->has_sfdm_1 == _TRUE_) {
       rho_i = pvecback_sp_long[pba->index_bg_rho_sfdm_1];
       if (ppt->has_source_delta_sfdm_1 == _TRUE_) {
         delta_i = ppt->sources[index_md]
         [index_ic * ppt->tp_size[index_md] + ppt->index_tp_delta_sfdm_1]
         [(index_tau-psp->ln_tau_size+ppt->tau_size) * ppt->k_size[index_md] + index_k];
         psp->matter_transfer[((index_tau*psp->ln_k_size + index_k) * psp->ic_size[index_md] + index_ic) *
         psp->tr_size + psp->index_tr_delta_sfdm_1] = delta_i;
         delta_rho_tot += rho_i * delta_i;
       }
       rho_tot += rho_i;
       if (ppt->has_source_theta_sfdm_1 == _TRUE_) {
         theta_i = ppt->sources[index_md]
         [index_ic * ppt->tp_size[index_md] + ppt->index_tp_theta_sfdm_1]
         [(index_tau-psp->ln_tau_size+ppt->tau_size) * ppt->k_size[index_md] + index_k];
         psp->matter_transfer[((index_tau*psp->ln_k_size + index_k) * psp->ic_size[index_md] + index_ic) *
         psp->tr_size + psp->index_tr_theta_sfdm_1] = theta_i;
         rho_plus_p_theta_tot += (rho_i + pvecback_sp_long[pba->index_bg_p_sfdm_1]) * theta_i;
       }
       rho_plus_p_tot += (rho_i + pvecback_sp_long[pba->index_bg_p_sfdm_1]);
     }
```

Figure I.2: Adding the scalar field 1 transfer function information.



```
/*Scalar Field Dark Matter 2*/
  if (pba->has_sfdm_2 == _TRUE_) {
      rho_i = pvecback_sp_long[pba->index_bg_rho_sfdm_2];
      if (ppt->has_source_delta_sfdm_2 == _TRUE_) {
          delta_i = ppt->sources[index_md]
          [index_ic * ppt->tp_size[index_md] + ppt->index_tp_delta_sfdm_2]
          [(index_tau-psp->ln_tau_size+ppt->tau_size) * ppt->k_size[index_md] + index_k];
          psp->matter_transfer[((index_tau*psp->ln_k_size + index_k) * psp->ic_size[index_md] + index_ic)
          psp->tr_size + psp->index_tr_delta_sfdm_2] = delta_i;
          delta_rho_tot += rho_i * delta_i;
      }
      rho_tot += rho_i;
      if (ppt->has_source_theta_sfdm_2 == _TRUE_) {
          theta_i = ppt->sources[index_md]
          [index_ic * ppt->tp_size[index_md] + ppt->index_tp_theta_sfdm_2]
          [(index_tau-psp->ln_tau_size+ppt->tau_size) * ppt->k_size[index_md] + index_k];
          psp->matter_transfer[((index_tau*psp->ln_k_size + index_k) * psp->ic_size[index_md] + index_ic)
          psp->tr_size + psp->index_tr_theta_sfdm_2] = theta_i;
          rho_plus_p_theta_tot += (rho_i + pvecback_sp_long[pba->index_bg_p_sfdm_2]) * theta_i;
      }
      rho_plus_p_tot += (rho_i + pvecback_sp_long[pba->index_bg_p_sfdm_2]);
  }
```

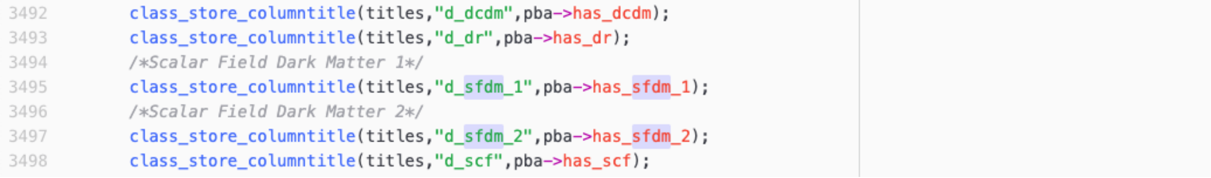
Figure I.3: Adding the scalar field 2 transfer function information.



```
class_store_columntitle(titles,"d_dcdm",pba->has_dcdm);
class_store_columntitle(titles,"d_dr",pba->has_dr);
/*Scalar Field Dark Matter 1*/
class_store_columntitle(titles,"d_sfdm_1",pba->has_sfdm_1);
/*Scalar Field Dark Matter 2*/
class_store_columntitle(titles,"d_sfdm_2",pba->has_sfdm_2);
class_store_columntitle(titles,"d_scf",pba->has_scf);
```

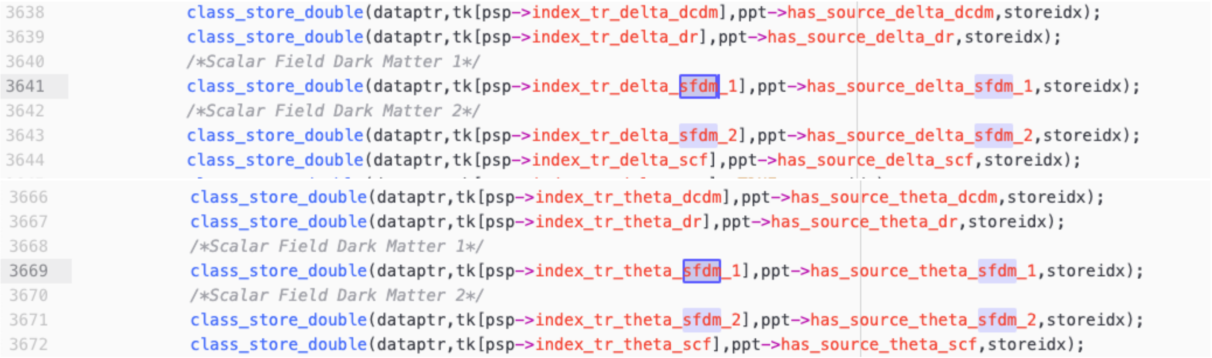
```
class_store_columntitle(titles,"t_dcdm",pba->has_dcdm);
class_store_columntitle(titles,"t_dr",pba->has_dr);
/*Scalar Field Dark Matter 1*/
class_store_columntitle(titles,"t_sfdm_1",pba->has_sfdm_1);
/*Scalar Field Dark Matter 2*/
class_store_columntitle(titles,"t_sfdm_2",pba->has_sfdm_2);
class_store_columntitle(titles,"t_scf",pba->has_scf);
```

Figure I.4: Naming the columns in the output files in CLASS format.



```
class_store_columntitle(titles,"d_dcdm",pba->has_dcdm);
class_store_columntitle(titles,"d_dr",pba->has_dr);
/*Scalar Field Dark Matter 1*/
class_store_columntitle(titles,"d_sfdm_1",pba->has_sfdm_1);
/*Scalar Field Dark Matter 2*/
class_store_columntitle(titles,"d_sfdm_2",pba->has_sfdm_2);
class_store_columntitle(titles,"d_scf",pba->has_scf);
```

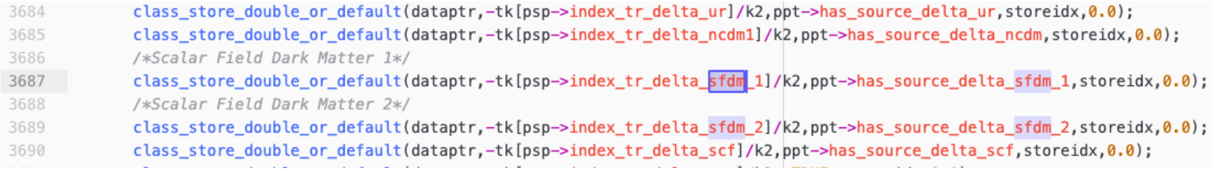
Figure I.5: Naming the columns in the output files in CAMB format.



```
class_store_double(dataptr,tk[psp->index_tr_delta_dcdm],ppt->has_source_delta_dcdm,storeidx);
class_store_double(dataptr,tk[psp->index_tr_delta_dr],ppt->has_source_delta_dr,storeidx);
/*Scalar Field Dark Matter 1*/
class_store_double(dataptr,tk[psp->index_tr_delta_sfdm_1],ppt->has_source_delta_sfdm_1,storeidx);
/*Scalar Field Dark Matter 2*/
class_store_double(dataptr,tk[psp->index_tr_delta_sfdm_2],ppt->has_source_delta_sfdm_2,storeidx);
class_store_double(dataptr,tk[psp->index_tr_delta_scf],ppt->has_source_delta_scf,storeidx);
```

```
class_store_double(dataptr,tk[psp->index_tr_theta_dcdm],ppt->has_source_theta_dcdm,storeidx);
class_store_double(dataptr,tk[psp->index_tr_theta_dr],ppt->has_source_theta_dr,storeidx);
/*Scalar Field Dark Matter 1*/
class_store_double(dataptr,tk[psp->index_tr_theta_sfdm_1],ppt->has_source_theta_sfdm_1,storeidx);
/*Scalar Field Dark Matter 2*/
class_store_double(dataptr,tk[psp->index_tr_theta_sfdm_2],ppt->has_source_theta_sfdm_2,storeidx);
class_store_double(dataptr,tk[psp->index_tr_theta_scf],ppt->has_source_theta_scf,storeidx);
```

Figure I.6: Telling CLASS which variables we want to store in CLASS format.



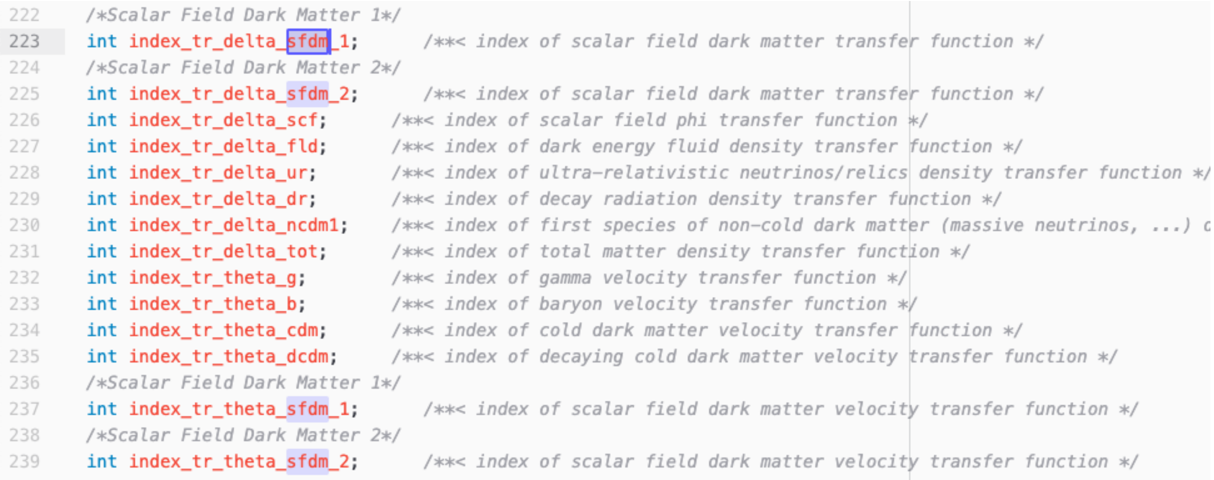
```
class_store_double_or_default(dataptr,-tk[psp->index_tr_delta_ur]/k2,ppt->has_source_delta_ur,storeidx,0.0);
class_store_double_or_default(dataptr,-tk[psp->index_tr_delta_ncdm1]/k2,ppt->has_source_delta_ncdm,storeidx,0.0);
/*Scalar Field Dark Matter 1*/
class_store_double_or_default(dataptr,-tk[psp->index_tr_delta_sfdm_1]/k2,ppt->has_source_delta_sfdm_1,storeidx,0.0);
/*Scalar Field Dark Matter 2*/
class_store_double_or_default(dataptr,-tk[psp->index_tr_delta_sfdm_2]/k2,ppt->has_source_delta_sfdm_2,storeidx,0.0);
class_store_double_or_default(dataptr,-tk[psp->index_tr_delta_scf]/k2,ppt->has_source_delta_scf,storeidx,0.0);
```

Figure I.7: Telling CLASS which variables we want to store in CAMB format.

Appendix J

spectra.h

In this file we declare the indices used in the spectra.c file, see Figure J.1.



```
/*Scalar Field Dark Matter 1*/
int index_tr_delta_sfdm_1;      /**< index of scalar field dark matter transfer function */
/*Scalar Field Dark Matter 2*/
int index_tr_delta_sfdm_2;      /**< index of scalar field dark matter transfer function */
int index_tr_delta_scf;      /**< index of scalar field phi transfer function */
int index_tr_delta_fld;      /**< index of dark energy fluid density transfer function */
int index_tr_delta_ur;       /**< index of ultra-relativistic neutrinos/relics density transfer function */
int index_tr_delta_dr;       /**< index of decay radiation density transfer function */
int index_tr_delta_ncdm1;    /**< index of first species of non-cold dark matter (massive neutrinos, ...) c
int index_tr_delta_tot;      /**< index of total matter density transfer function */
int index_tr_theta_g;        /**< index of gamma velocity transfer function */
int index_tr_theta_b;        /**< index of baryon velocity transfer function */
int index_tr_theta_cdm;      /**< index of cold dark matter velocity transfer function */
int index_tr_theta_dcdm;     /**< index of decaying cold dark matter velocity transfer function */
/*Scalar Field Dark Matter 1*/
int index_tr_theta_sfdm_1;      /**< index of scalar field dark matter velocity transfer function */
/*Scalar Field Dark Matter 2*/
int index_tr_theta_sfdm_2;      /**< index of scalar field dark matter velocity transfer function */
```

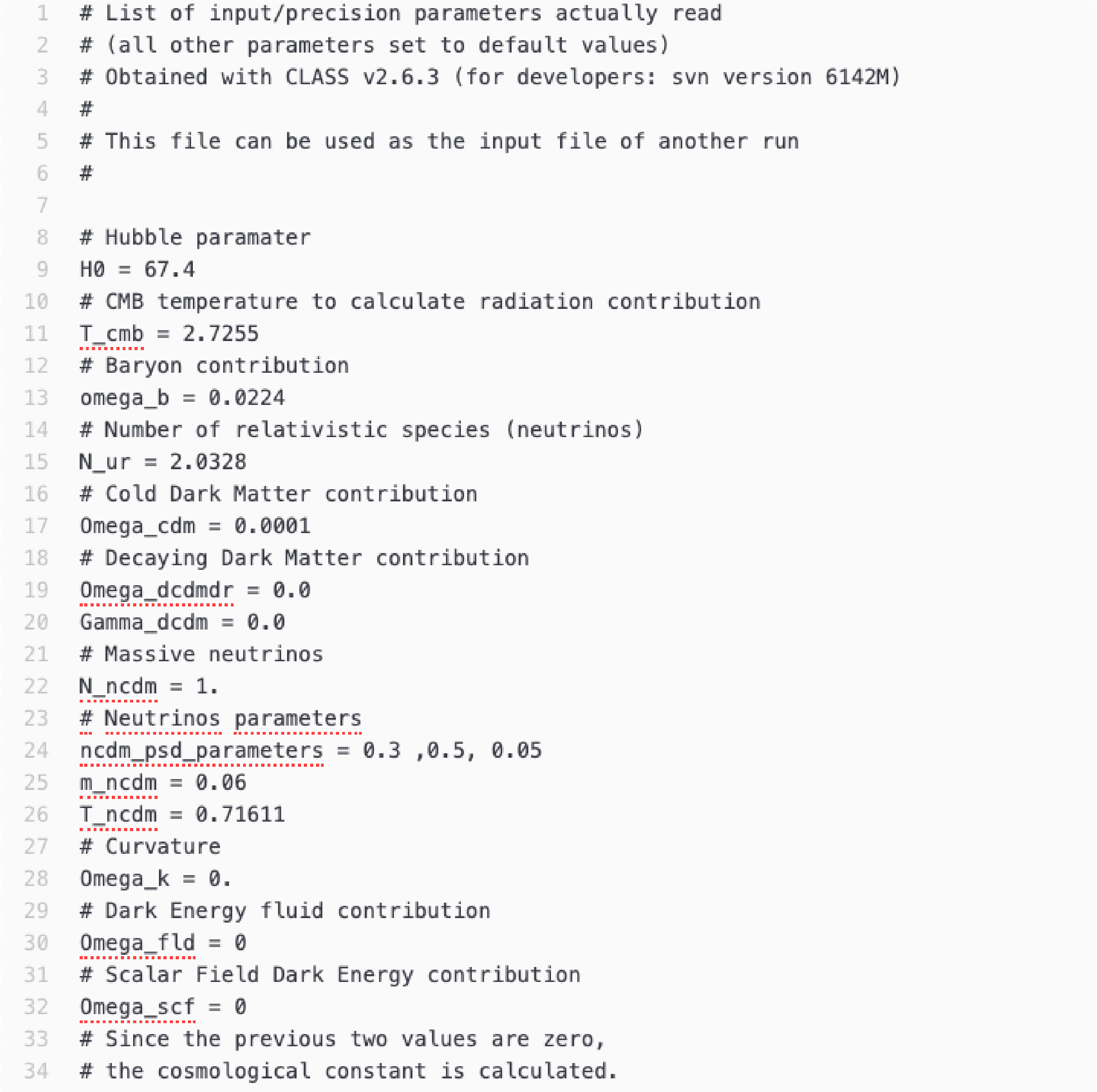
Figure J.1: Defining indices.

Appendix K

Input files

As we have seen in the previous appendices, CLASS needs an input file where it will extract all the values to do the calculations. These files must have the extension .ini. The file that contains all the information about the input parameters is called explanatory.ini. However, here we will review a file that was made specifically for the case of two scalar fields. Something very important to note is that only the lines with the equal sign (=) will be considered as input, the rest will be treated as comments. We can also use # to make comments. Numerically, in order to use the synchronous gauge, we need the cold dark matter contribution to be non-zero, so we assign it a negligible contribution. If any parameter does not appear in the .ini file, CLASS will assign the default value defined in the input.c file. In Figures K.1, K.2 and K.3 we show the .ini file for double field with quadratic and cos potential. For the other combinations used in this work we only need to modify the SFDM sections.

For the quadratic case, we put the mass in the first element of the scalar field parameter array while the second entry must be zero, see Figure K.4. For the cos case, in addition to the mass, we must specify the value of lambda, therefore the second entry of the array must be different from zero, see Figure K.5. And for the cosh case, we must put a negative value in the second entry of the array, that is, the value of lambda, see Figure K.6. In this case, it is not necessary to assign a specific value for the mass, as we remember that it

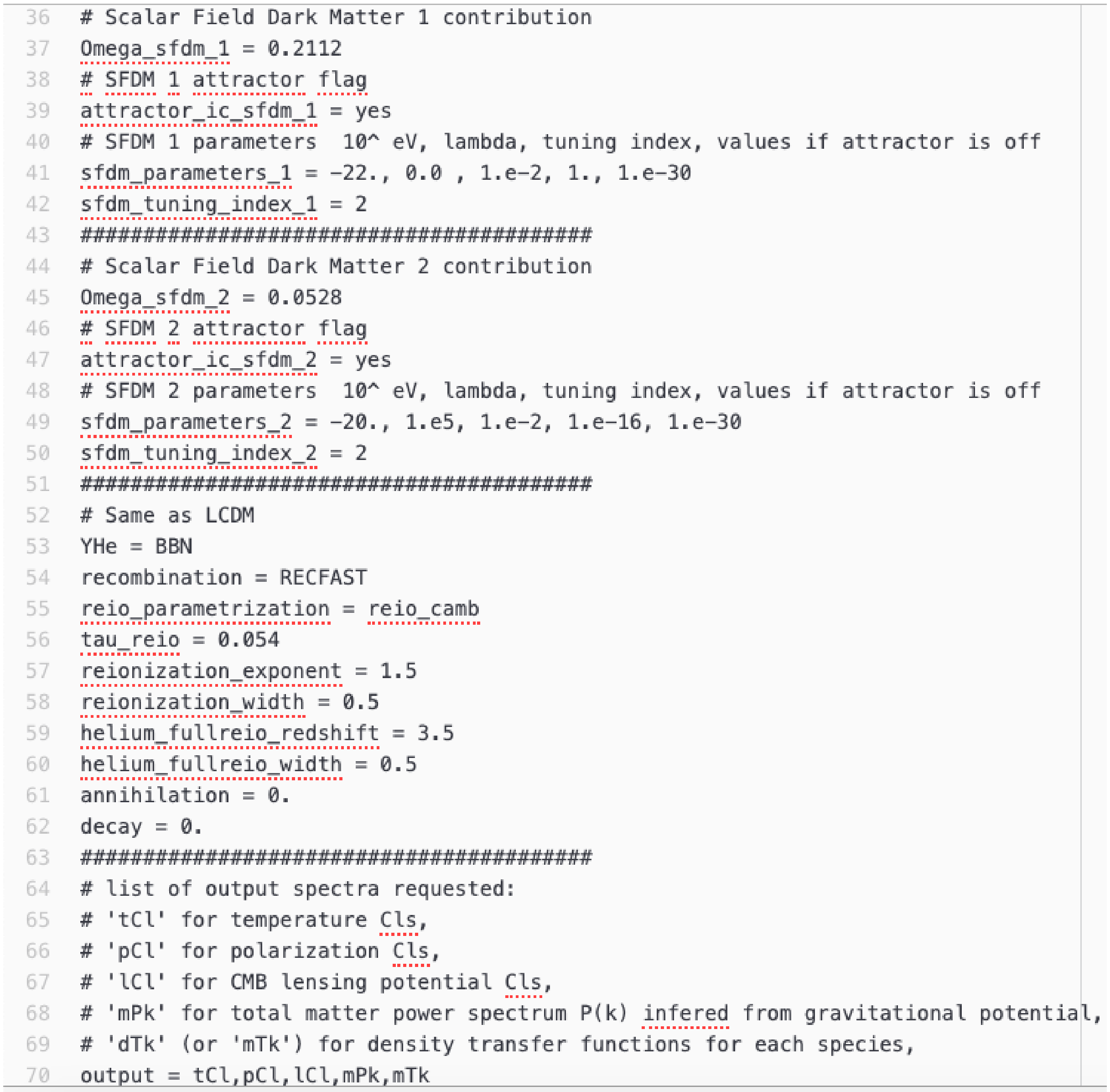


```
# List of input/precision parameters actually read
# (all other parameters set to default values)
# Obtained with CLASS v2.6.3 (for developers: svn version 6142M)
#
# This file can be used as the input file of another run
#

# Hubble paramater
H0 = 67.4
# CMB temperature to calculate radiation contribution
T_cmb = 2.7255
# Baryon contribution
omega_b = 0.0224
# Number of relativistic species (neutrinos)
N_ur = 2.0328
# Cold Dark Matter contribution
Omega_cdm = 0.0001
# Decaying Dark Matter contribution
Omega_dcdmdr = 0.0
Gamma_dcdm = 0.0
# Massive neutrinos
N_ncdm = 1.
# Neutrinos parameters
ncdm_psd_parameters = 0.3 ,0.5, 0.05
m_ncdm = 0.06
T_ncdm = 0.71611
# Curvature
Omega_k = 0.
# Dark Energy fluid contribution
Omega_fld = 0
# Scalar Field Dark Energy contribution
Omega_scf = 0
# Since the previous two values are zero,
# the cosmological constant is calculated.
```

Figure K.1: Double Scalar Field Dark Matter input file.

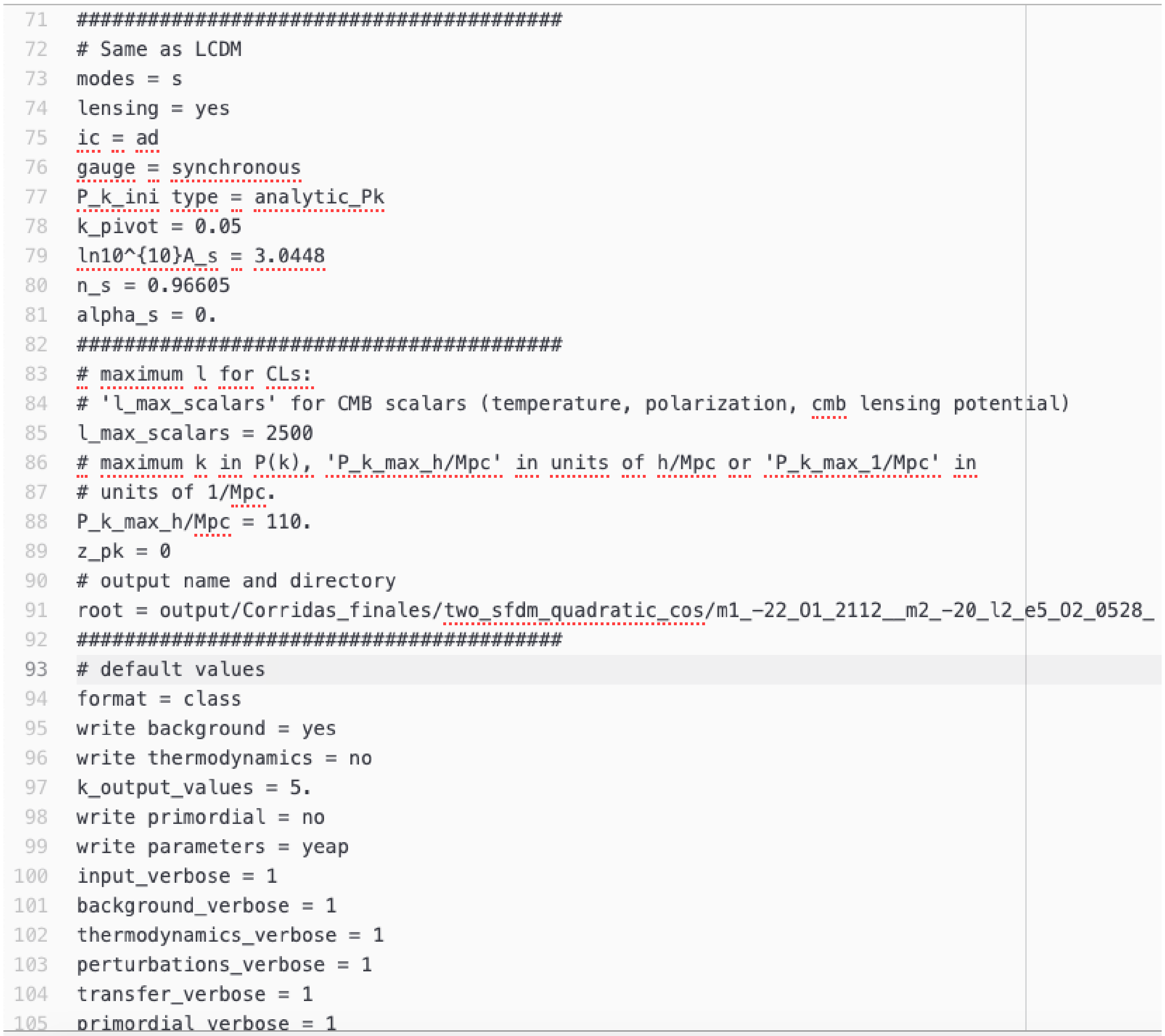
depends on the value of lambda. If the attractor is off, we need to specify the value of the remaining inputs. This is valid for both fields.

In Figure K.7, we see the result of running CLASS with the .ini file we just reviewed. All of these messages are found in .c files inside the source folder. For example, the information of the scalar fields is in the file background.c as we saw in the appendix E.



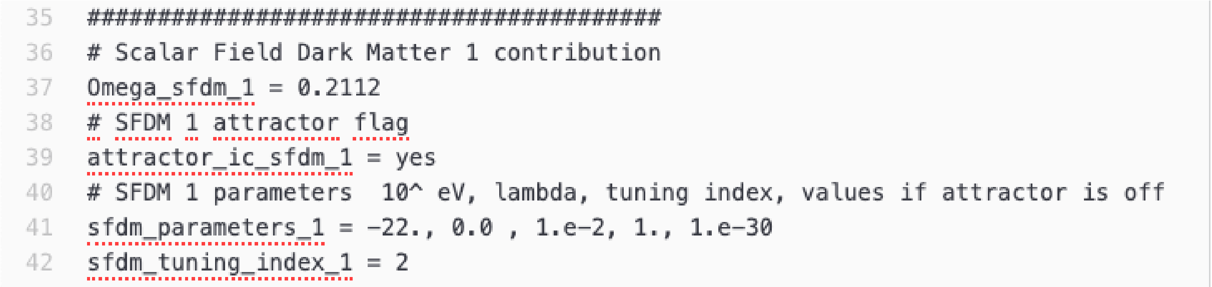
```
# Scalar Field Dark Matter 1 contribution
Omega_sfdm_1 = 0.2112
# SFDM 1 attractor flag
attractor_ic_sfdm_1 = yes
# SFDM 1 parameters  10^ eV, lambda, tuning index, values if attractor is off
sfdm_parameters_1 = -22., 0.0 , 1.e-2, 1., 1.e-30
sfdm_tuning_index_1 = 2
#########################################
# Scalar Field Dark Matter 2 contribution
Omega_sfdm_2 = 0.0528
# SFDM 2 attractor flag
attractor_ic_sfdm_2 = yes
# SFDM 2 parameters  10^ eV, lambda, tuning index, values if attractor is off
sfdm_parameters_2 = -20., 1.e5, 1.e-2, 1.e-16, 1.e-30
sfdm_tuning_index_2 = 2
#########################################
# Same as LCDM
YHe = BBN
recombination = RECFAST
reio_parametrization = reio_camb
tau_reio = 0.054
reionization_exponent = 1.5
reionization_width = 0.5
helium_fullreio_redshift = 3.5
helium_fullreio_width = 0.5
annihilation = 0.
decay = 0.
#########################################
# list of output spectra requested:
# 'tCl' for temperature Cls,
# 'pCl' for polarization Cls,
# 'lCl' for CMB lensing potential Cls,
# 'mPk' for total matter power spectrum P(k) infered from gravitational potential,
# 'dTk' (or 'mTk') for density transfer functions for each species,
output = tCl,pCl,lCl,mPk,mTk
```

Figure K.2: Double Scalar Field Dark Matter input file.



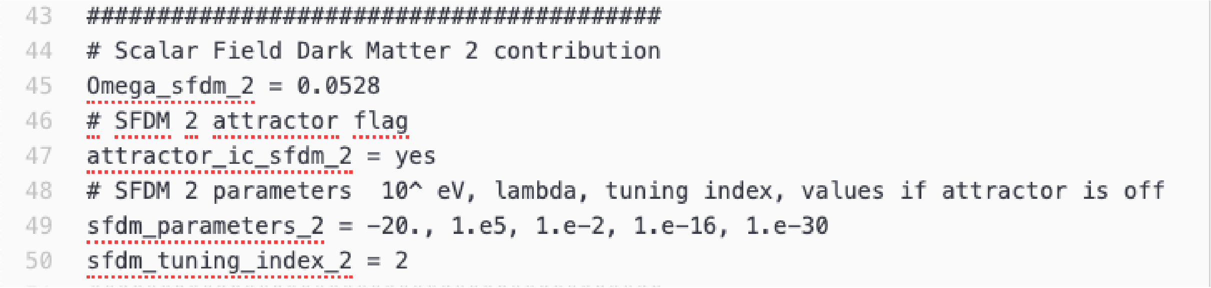
```
#########################################
# Same as LCDM
modes = s
lensing = yes
ic = ad
gauge = synchronous
P_k_ini type = analytic_Pk
k_pivot = 0.05
ln10^{10}A_s = 3.0448
n_s = 0.96605
alpha_s = 0.
#########################################
# maximum l for CLs:
# 'l_max_scalars' for CMB scalars (temperature, polarization, cmb lensing potential)
l_max_scalars = 2500
# maximum k in P(k), 'P_k_max_h/Mpc' in units of h/Mpc or 'P_k_max_1/Mpc' in
# units of 1/Mpc.
P_k_max_h/Mpc = 110.
z_pk = 0
# output name and directory
root = output/Corridas_finales/two_sfdm_quadratic_cos/m1_-22_01_2112__m2_-20_l2_e5_02_0528_
#########################################
# default values
format = class
write background = yes
write thermodynamics = no
k_output_values = 5.
write primordial = no
write parameters = yeap
input_verbose = 1
background_verbose = 1
thermodynamics_verbose = 1
perturbations_verbose = 1
transfer_verbose = 1
primordial_verbose = 1
```

Figure K.3: Double Scalar Field Dark Matter input file.



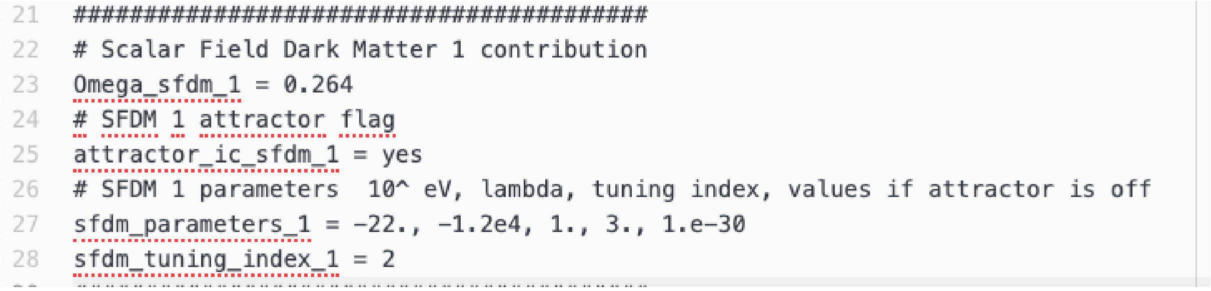
```
#########################################
# Scalar Field Dark Matter 1 contribution
Omega_sfdm_1 = 0.2112
# SFDM 1 attractor flag
attractor_ic_sfdm_1 = yes
# SFDM 1 parameters  10^ eV, lambda, tuning index, values if attractor is off
sfdm_parameters_1 = -22., 0.0 , 1.e-2, 1., 1.e-30
sfdm_tuning_index_1 = 2
```

Figure K.4: Parameter values for the quadratic case.



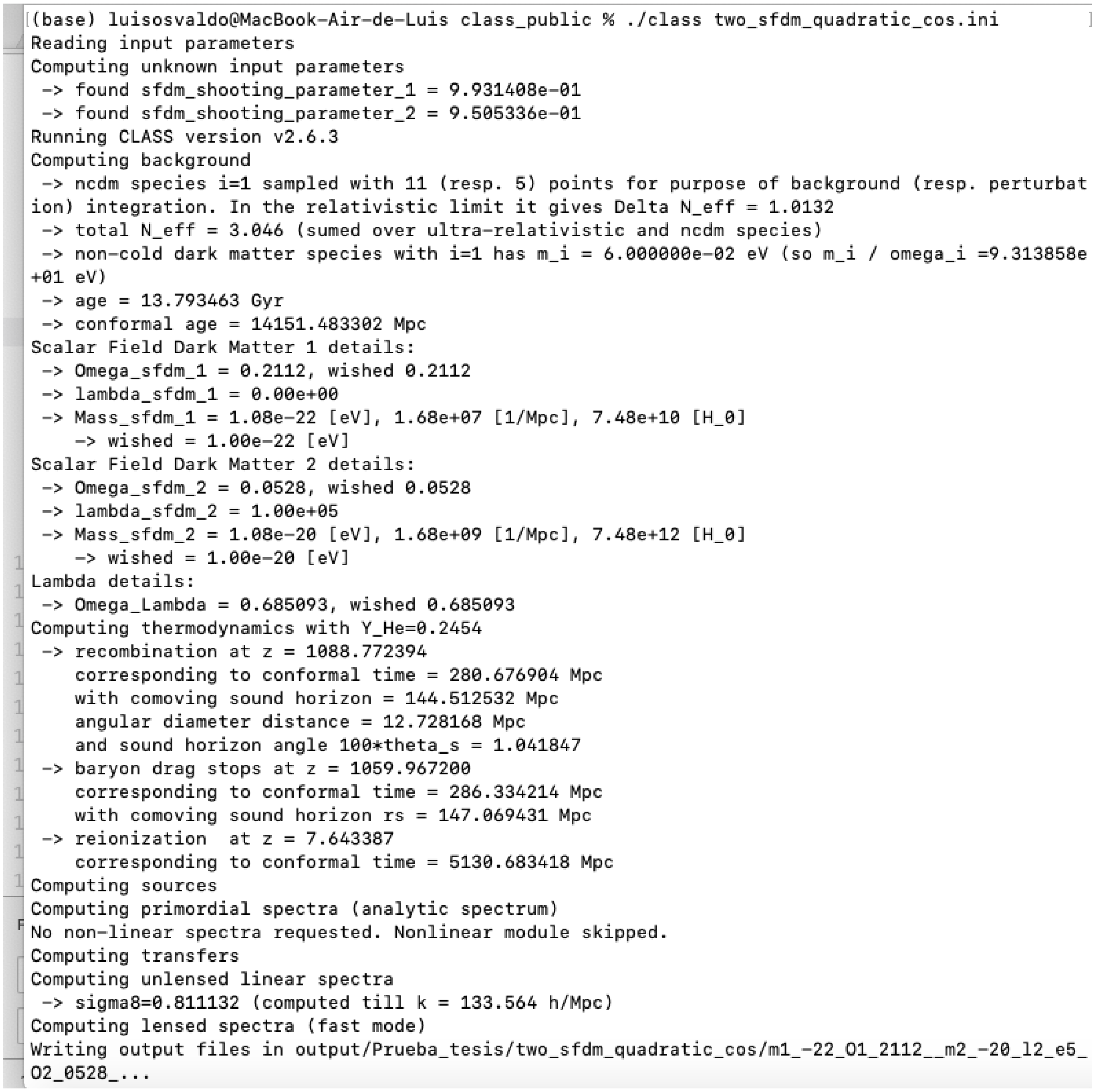
```
##########################################
# Scalar Field Dark Matter 2 contribution
Omega_sfdm_2 = 0.0528
# SFDM 2 attractor flag
attractor_ic_sfdm_2 = yes
# SFDM 2 parameters  10^ eV, lambda, tuning index, values if attractor is off
sfdm_parameters_2 = -20., 1.e5, 1.e-2, 1.e-16, 1.e-30
sfdm_tuning_index_2 = 2
```

Figure K.5: Parameter values for the cos case.



```
##########################################
# Scalar Field Dark Matter 1 contribution
Omega_sfdm_1 = 0.264
# SFDM 1 attractor flag
attractor_ic_sfdm_1 = yes
# SFDM 1 parameters  10^ eV, lambda, tuning index, values if attractor is off
sfdm_parameters_1 = -22., -1.2e4, 1., 3., 1.e-30
sfdm_tuning_index_1 = 2
```

Figure K.6: Parameter values for the cosh case.



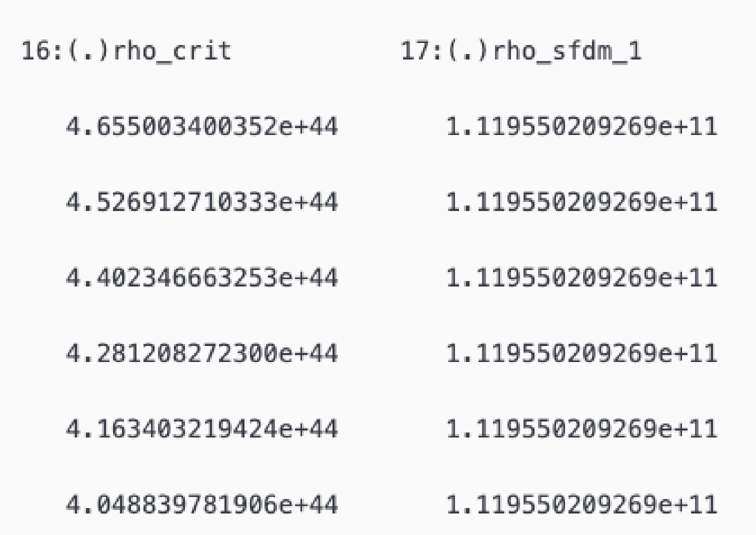
```
(base) luisosvaldo@MacBook-Air-de-Luis class_public % ./class two_sfdm_quadratic_cos.ini
Reading input parameters
Computing unknown input parameters
 -> found sfdm_shooting_parameter_1 = 9.931408e-01
 -> found sfdm_shooting_parameter_2 = 9.505336e-01
Running CLASS version v2.6.3
Computing background
 -> ncdm species i=1 sampled with 11 (resp. 5) points for purpose of background (resp. perturbat
ion) integration. In the relativistic limit it gives Delta N_eff = 1.0132
 -> total N_eff = 3.046 (sumed over ultra-relativistic and ncdm species)
 -> non-cold dark matter species with i=1 has m_i = 6.000000e-02 eV (so m_i / omega_i =9.313858e
+01 eV)
 -> age = 13.793463 Gyr
 -> conformal age = 14151.483302 Mpc
Scalar Field Dark Matter 1 details:
 -> Omega_sfdm_1 = 0.2112, wished 0.2112
 -> lambda_sfdm_1 = 0.00e+00
 -> Mass_sfdm_1 = 1.08e-22 [eV], 1.68e+07 [1/Mpc], 7.48e+10 [H_0]
    -> wished = 1.00e-22 [eV]
Scalar Field Dark Matter 2 details:
 -> Omega_sfdm_2 = 0.0528, wished 0.0528
 -> lambda_sfdm_2 = 1.00e+05
 -> Mass_sfdm_2 = 1.08e-20 [eV], 1.68e+09 [1/Mpc], 7.48e+12 [H_0]
    -> wished = 1.00e-20 [eV]
Lambda details:
 -> Omega_Lambda = 0.685093, wished 0.685093
Computing thermodynamics with Y_He=0.2454
 -> recombination at z = 1088.772394
    corresponding to conformal time = 280.676904 Mpc
    with comoving sound horizon = 144.512532 Mpc
    angular diameter distance = 12.728168 Mpc
    and sound horizon angle 100*theta_s = 1.041847
 -> baryon drag stops at z = 1059.967200
    corresponding to conformal time = 286.334214 Mpc
    with comoving sound horizon rs = 147.069431 Mpc
 -> reionization  at z = 7.643387
    corresponding to conformal time = 5130.683418 Mpc
Computing sources
Computing primordial spectra (analytic spectrum)
No non-linear spectra requested. Nonlinear module skipped.
Computing transfers
Computing unlensed linear spectra
 -> sigma8=0.811132 (computed till k = 133.564 h/Mpc)
Computing lensed spectra (fast mode)
Writing output files in output/Prueba_tesis/two_sfdm_quadratic_cos/m1_-22_O1_2112__m2_-20_l2_e5_
O2_0528_...
```

Figure K.7: Running CLASS.

Appendix L

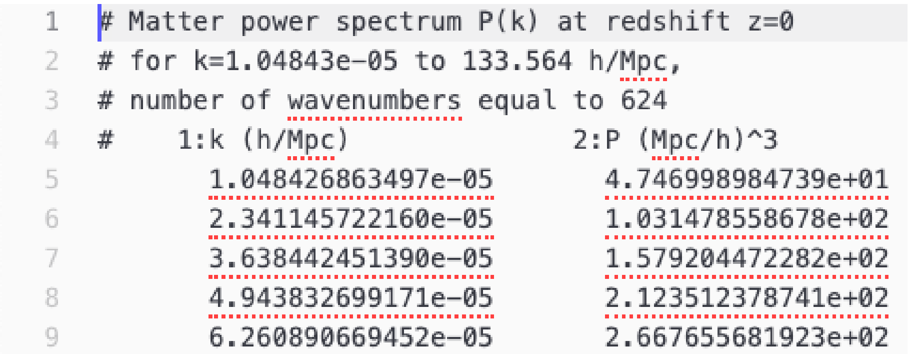
Output files

The output files that are important to us are the background files (name_background.dat), the MPS files (name_pk.dat), and the CMB files (name_cl.dat). From the name_background.dat file we will use the critical density and field densities columns, see Figure L.1. From the name_pk.dat file we will use the only two columns, L.2; and from the name_cl.dat file we will only use the first two columns that give us information on the CMB temperature, L.3.



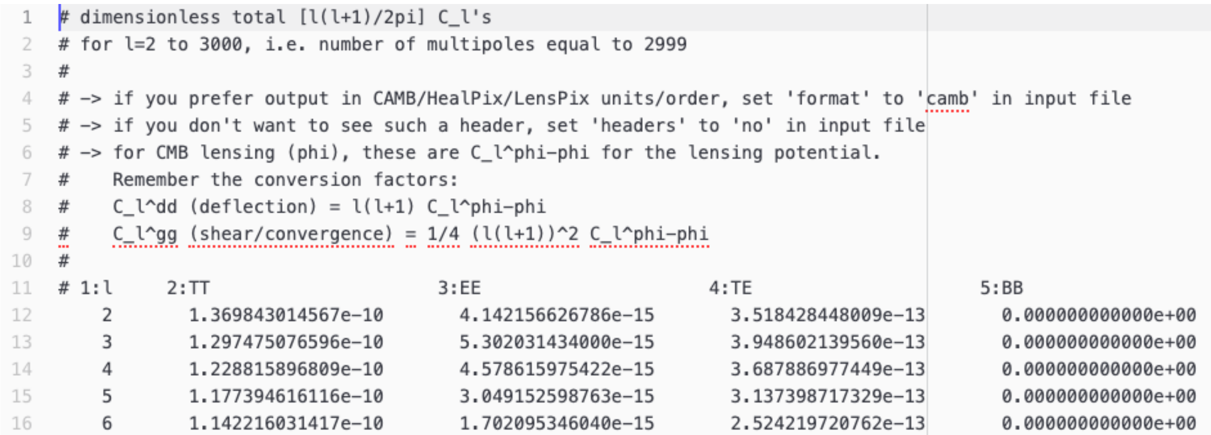
```
16:(.)rho_crit          17:(.)rho_sfdm_1

   4.655003400352e+44      1.119550209269e+11

   4.526912710333e+44      1.119550209269e+11

   4.402346663253e+44      1.119550209269e+11

   4.281208272300e+44      1.119550209269e+11

   4.163403219424e+44      1.119550209269e+11

   4.048839781906e+44      1.119550209269e+11
```

Figure L.1: Example of file name_background.dat.



```
# Matter power spectrum P(k) at redshift z=0
# for k=1.04843e-05 to 133.564 h/Mpc,
# number of wavenumbers equal to 624
#    1:k (h/Mpc)              2:P (Mpc/h)^3
        1.048426863497e-05       4.746998984739e+01
        2.341145722160e-05       1.031478558678e+02
        3.638442451390e-05       1.579204472282e+02
        4.943832699171e-05       2.123512378741e+02
        6.260890669452e-05       2.667655681923e+02
```

Figure L.2: Example of file name_pk.dat.



```
# dimensionless total [l(l+1)/2pi] C_l's
# for l=2 to 3000, i.e. number of multipoles equal to 2999
#
# -> if you prefer output in CAMB/HealPix/LensPix units/order, set 'format' to 'camb' in input file
# -> if you don't want to see such a header, set 'headers' to 'no' in input file
# -> for CMB lensing (phi), these are C_l^phi-phi for the lensing potential.
#    Remember the conversion factors:
#    C_l^dd (deflection) = l(l+1) C_l^phi-phi
#    C_l^gg (shear/convergence) = 1/4 (l(l+1))^2 C_l^phi-phi
#
# 1:l     2:TT                  3:EE                  4:TE                  5:BB
     2      1.369843014567e-10      4.142156626786e-15      3.518428448009e-13      0.000000000000e+00
     3      1.297475076596e-10      5.302031434000e-15      3.948602139560e-13      0.000000000000e+00
     4      1.228815896809e-10      4.578615975422e-15      3.687886977449e-13      0.000000000000e+00
     5      1.177394616116e-10      3.049152598763e-15      3.137398717329e-13      0.000000000000e+00
     6      1.142216031417e-10      1.702095346040e-15      2.524219720762e-13      0.000000000000e+00
```

Figure L.3: Example of file name_cl.dat.

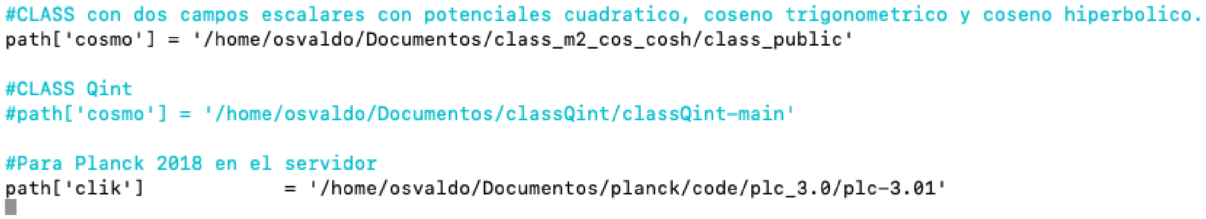
Appendix M

Monte Python code

As its name says, Monte Python [158, 159] is a code written in python that does parameter estimation by taking cosmological information from CLASS using MCMC.

M.1 Installing Monte Python

The first thing we need to do is download the code from [192], just like we do with CLASS. Then, inside the montepython_public-3.3 directory, we must create a new file called default.conf by copying and adapting the contents of the default.conf.template file to our computer. In this file we tell Monte Python where the CLASS is so it can use it. It is important that it is a new file because if we only copy and paste the file default.conf.template, the copy will keep the extension .template and not .conf. We show the default.conf file for ekbek server in Figure M.1. clik refers to the Planck 18 likelihood that we will review later. The python libraries needed by Monte Python have already been installed in appendix A.1. As it is written in python we don't need to compile anything, we just have to run the code.

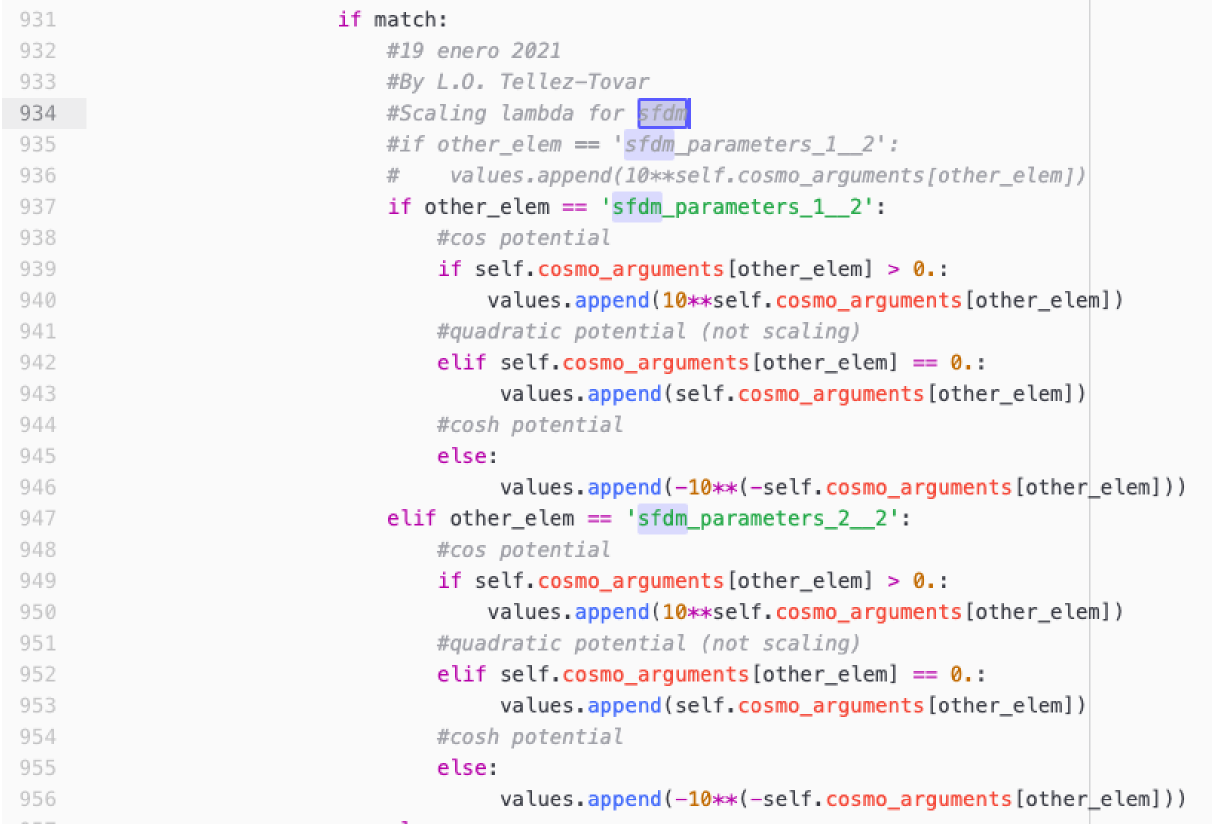


```
#CLASS con dos campos escalares con potenciales cuadratico, coseno trigonometrico y coseno hiperbolico.
path['cosmo'] = '/home/osvaldo/Documentos/class_m2_cos_cosh/class_public'

#CLASS Qint
#path['cosmo'] = '/home/osvaldo/Documentos/classQint/classQint-main'

#Para Planck 2018 en el servidor
path['clik']          = '/home/osvaldo/Documentos/planck/code/plc_3.0/plc-3.01'
```

Figure M.1: default.conf file for the ekbek server.

M.2 Modifying Monte Python

As we saw in section 3.4, $\lambda_{\phi i}$ can range from zero to values like 10^6 so, to allow Monte Python to cover the parameter space, we redefine this parameter to be read logarithmically. For this, we modified the data.py file inside the montepython_public-3.3/montepython directory, see Figure M.2.

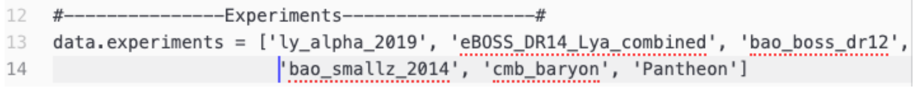


```
if match:
    #19 enero 2021
    #By L.O. Tellez-Tovar
    #Scaling lambda for sfdm
    #if other_elem == 'sfdm_parameters_1__2':
    #    values.append(10**self.cosmo_arguments[other_elem])
    if other_elem == 'sfdm_parameters_1__2':
        #cos potential
        if self.cosmo_arguments[other_elem] > 0.:
            values.append(10**self.cosmo_arguments[other_elem])
        #quadratic potential (not scaling)
        elif self.cosmo_arguments[other_elem] == 0.:
             values.append(self.cosmo_arguments[other_elem])
        #cosh potential
        else:
             values.append(-10**(-self.cosmo_arguments[other_elem]))
    elif other_elem == 'sfdm_parameters_2__2':
        #cos potential
        if self.cosmo_arguments[other_elem] > 0.:
            values.append(10**self.cosmo_arguments[other_elem])
        #quadratic potential (not scaling)
        elif self.cosmo_arguments[other_elem] == 0.:
             values.append(self.cosmo_arguments[other_elem])
        #cosh potential
        else:
             values.append(-10**(-self.cosmo_arguments[other_elem]))
```

Figure M.2: Modifying the data.py file.

M.3 Input files

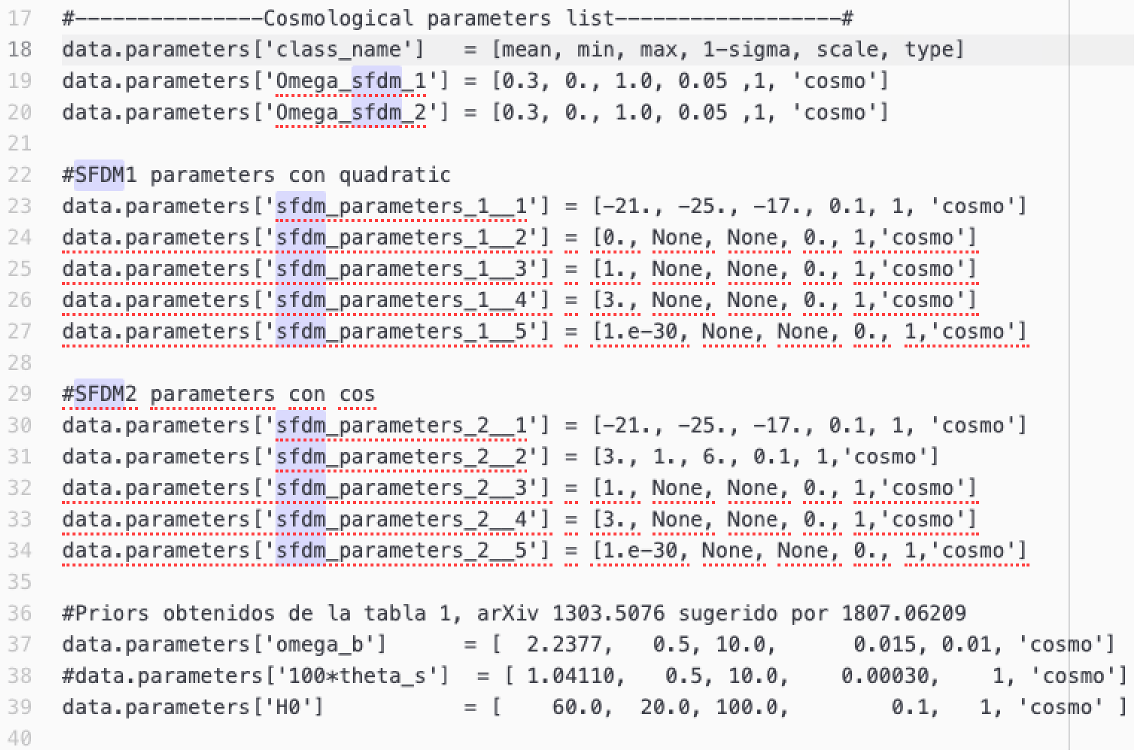
Like CLASS, Monte Python requires an input file. In this case, it has the .param extension. It contains the information of the parameters to be adjusted, the data to be used and the conditions under which the inference process will be carried out. Here we review one of the used files. First we must include the data, the name of each set must match the names of the directories that are in the montepython_public-3.3/montepython/likelihoods directory, see Figure M.3.



```
#---------------Experiments------------------#
data.experiments = ['ly_alpha_2019', 'eBOSS_DR14_Lya_combined', 'bao_boss_dr12',
                     'bao_smallz_2014', 'cmb_baryon', 'Pantheon']
```

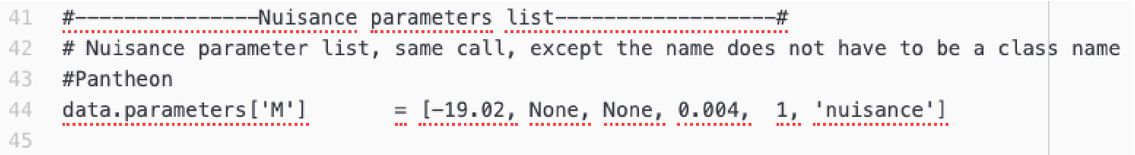
Figure M.3: Data sets in .param file.

Then we put the parameters, there are three types. The cosmo types are the ones that will enter the inference process, the nuissance types are for the data and depend on the data set; and the derived types that will not participate in the inference process but their value will be calculated using the cosmo type data. First, we put the name which, for the first and third type of parameter, must match the one in the CLASS .ini file. In the first entry, after the equals sign, we put the mean value that we believe the parameter may have. In the second and third entries are the lower and upper bounds, respectively, of the parameter prior. If we don't have a specific prior we put none on both entries. In the fourth entry goes the value of $1-\sigma$. If we want a parameter to remain unchanged, we leave this entry as zero. In the fifth input we put the scale value. For scalar fields we will leave it at 1 because we will not do any scaling. Otherwise, for omega_b this entry is equal to 0.01 because Monte Python does scale this parameter. In the last entry we defined the type, in this example, we will put the cosmo types first, see Figure M.4. Note that, since in CLASS the scalar field parameters appear in an array, in Monte Python we must put them in the same way. However, if the attractor is on, we put 0 as the $1-\sigma$ value in inputs 3, 4 and 5 of the parameter array. If we only put, for example, the scalar field mass and we do not put the others parameters even if they are not going to vary, Monte Python will take it as an error and will not be executed, see Figure M.4. In this case, Pantheon requires nuisance parameters so we need to add them, see Figure M.5. Then we define the derived parameters and assign values to parameters that we want to be fixed, see Figure M.6. Finally, we set the maximum value of k at which we want the MPS to be calculated and default values for Monte Python in case they are not specified when running the code, see Figure M.7.



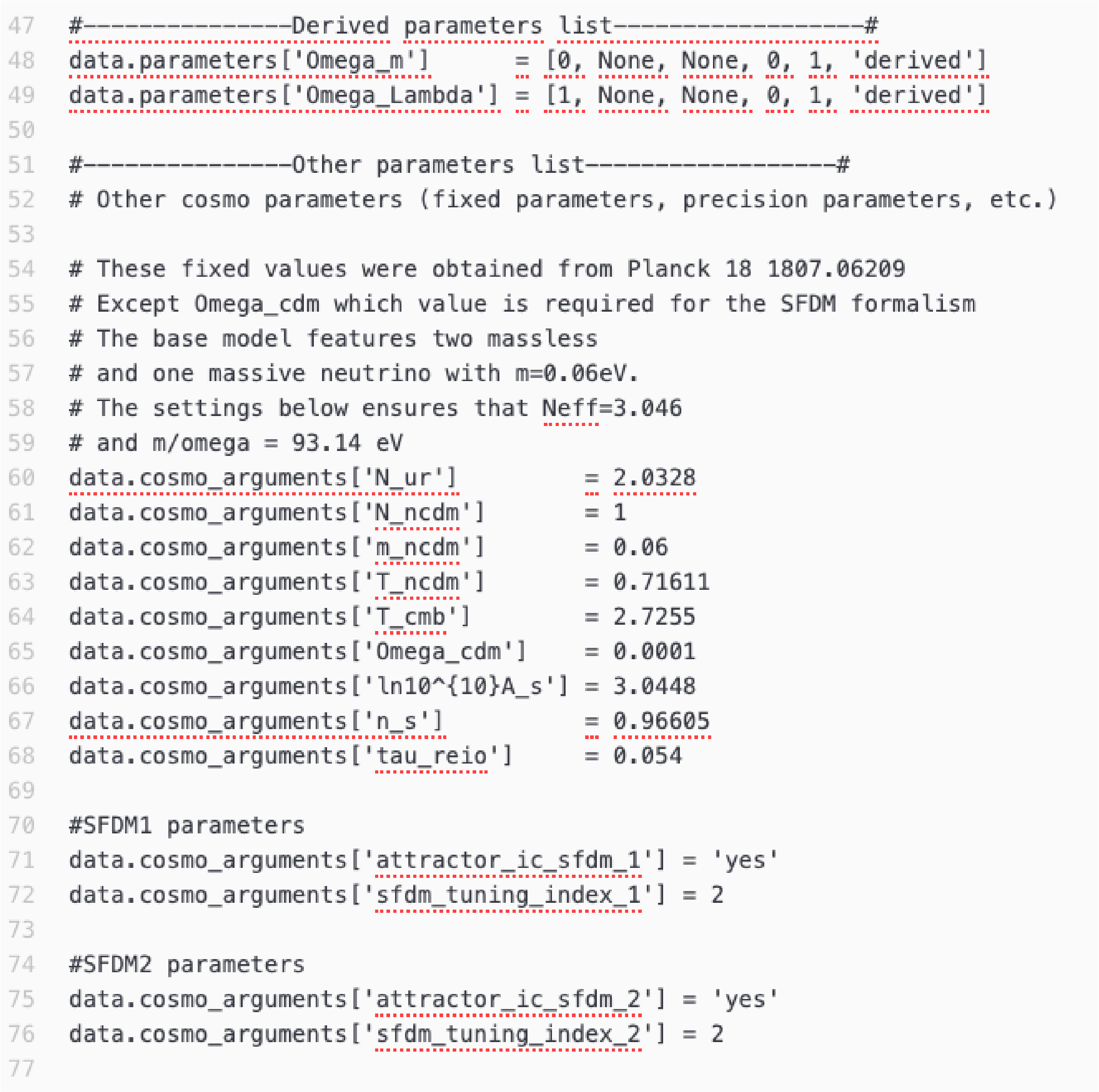
```
#---------------Cosmological parameters list------------------#
data.parameters['class_name']   = [mean, min, max, 1-sigma, scale, type]
data.parameters['Omega_sfdm_1'] = [0.3, 0., 1.0, 0.05 ,1, 'cosmo']
data.parameters['Omega_sfdm_2'] = [0.3, 0., 1.0, 0.05 ,1, 'cosmo']

#SFDM1 parameters con quadratic
data.parameters['sfdm_parameters_1__1'] = [-21., -25., -17., 0.1, 1, 'cosmo']
data.parameters['sfdm_parameters_1__2'] = [0., None, None, 0., 1,'cosmo']
data.parameters['sfdm_parameters_1__3'] = [1., None, None, 0., 1,'cosmo']
data.parameters['sfdm_parameters_1__4'] = [3., None, None, 0., 1,'cosmo']
data.parameters['sfdm_parameters_1__5'] = [1.e-30, None, None, 0., 1,'cosmo']

#SFDM2 parameters con cos
data.parameters['sfdm_parameters_2__1'] = [-21., -25., -17., 0.1, 1, 'cosmo']
data.parameters['sfdm_parameters_2__2'] = [3., 1., 6., 0.1, 1,'cosmo']
data.parameters['sfdm_parameters_2__3'] = [1., None, None, 0., 1,'cosmo']
data.parameters['sfdm_parameters_2__4'] = [3., None, None, 0., 1,'cosmo']
data.parameters['sfdm_parameters_2__5'] = [1.e-30, None, None, 0., 1,'cosmo']

#Priors obtenidos de la tabla 1, arXiv 1303.5076 sugerido por 1807.06209
data.parameters['omega_b']      = [  2.2377,    0.5, 10.0,      0.015, 0.01, 'cosmo']
#data.parameters['100*theta_s']  = [ 1.04110,    0.5, 10.0,    0.00030,    1, 'cosmo']
data.parameters['H0']           = [    60.0,   20.0, 100.0,       0.1,    1, 'cosmo' ]

```

Figure M.4: Defining cosmo parameters.



```
#---------------Nuisance parameters list------------------#
# Nuisance parameter list, same call, except the name does not have to be a class name
#Pantheon
data.parameters['M']       = [-19.02, None, None, 0.004,  1, 'nuisance']

```

Figure M.5: Defining nuisance parameters.

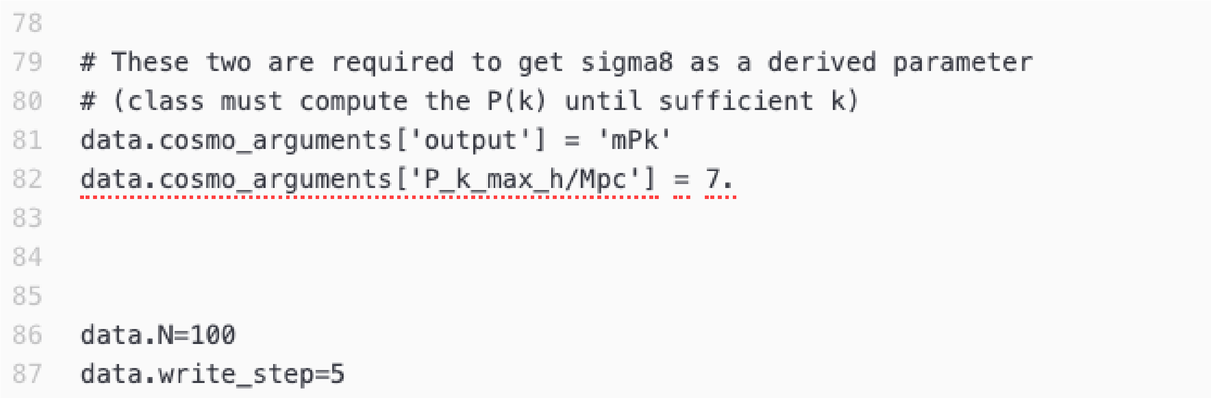


```
#---------------Derived parameters list------------------#
data.parameters['Omega_m']       = [0, None, None, 0, 1, 'derived']
data.parameters['Omega_Lambda'] = [1, None, None, 0, 1, 'derived']

#---------------Other parameters list------------------#
# Other cosmo parameters (fixed parameters, precision parameters, etc.)

# These fixed values were obtained from Planck 18 1807.06209
# Except Omega_cdm which value is required for the SFDM formalism
# The base model features two massless
# and one massive neutrino with m=0.06eV.
# The settings below ensures that Neff=3.046
# and m/omega = 93.14 eV
data.cosmo_arguments['N_ur']         = 2.0328
data.cosmo_arguments['N_ncdm']       = 1
data.cosmo_arguments['m_ncdm']       = 0.06
data.cosmo_arguments['T_ncdm']       = 0.71611
data.cosmo_arguments['T_cmb']        = 2.7255
data.cosmo_arguments['Omega_cdm']    = 0.0001
data.cosmo_arguments['ln10^{10}A_s'] = 3.0448
data.cosmo_arguments['n_s']          = 0.96605
data.cosmo_arguments['tau_reio']     = 0.054

#SFDM1 parameters
data.cosmo_arguments['attractor_ic_sfdm_1'] = 'yes'
data.cosmo_arguments['sfdm_tuning_index_1'] = 2

#SFDM2 parameters
data.cosmo_arguments['attractor_ic_sfdm_2'] = 'yes'
data.cosmo_arguments['sfdm_tuning_index_2'] = 2

```

Figure M.6: Defining the derived parameters and assigning values to parameters that we want to be fixed.

M.4 Running Monte Python

For the process that we are going to explain, we recommend using tmux, this is a virtual terminal inside the “real” terminal in which we are working. The advantage of tmux is



```

# These two are required to get sigma8 as a derived parameter
# (class must compute the P(k) until sufficient k)
data.cosmo_arguments['output'] = 'mPk'
data.cosmo_arguments['P_k_max_h/Mpc'] = 7.



data.N=100
data.write_step=5
```

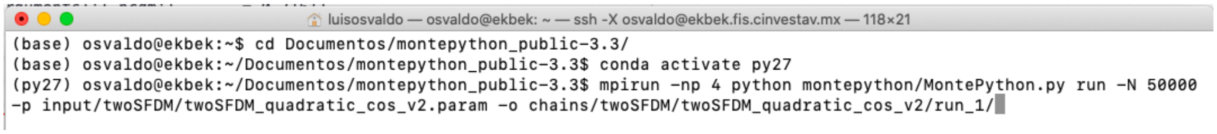
Figure M.7: Setting the maximum value of k at which we want the MPS to be calculated and default values for run Monte Python.

that we will leave the processes running and we can close both the main terminal and the remote server session. We install it with sudo apt-get install tmux.

The process of running Monte Python consists of three parts. The first consists of generating chains of few steps. The second is to analyze these chains and obtain the covariance matrix. The third part is to run a new set of chains with many more steps than the first. Finally, these chains are analyzed.

We first get into tmux by simply typing tmux in the terminal. Next, we move to the main Monte Python directory and activate the python environment. There, we write the line shown in Figure M.8. With mpirun -np we tell the code how many chains we want to run, that is, we are going to run the code in parallel. In this case we will use 4 chains. Then, with python montepython/MontePython.py we will run the main file. With run -N we tell it that we will do the inference process. After this, we put the number of steps per chain. For this first part we use 50000. With -p we pass the input file and with -o we pass the address where the output files will be saved.

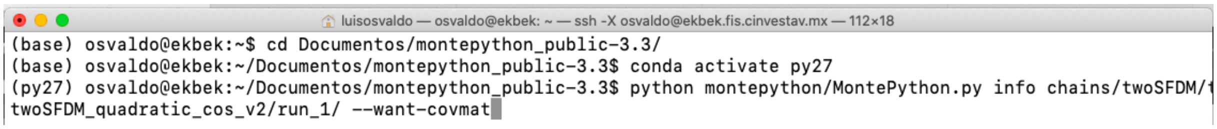
When these chains are done, we're going to analyze them to get the covariance matrix. With python montepython/MontePython.py info we tell the code that we want to analyse the chains. Then, we put the output directory of the previous part. And with –want-covmat we tell Monte Python to get the covariance matrix will be saved in the same


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 118×21

```
(base) osvaldo@ekbek:~$ cd Documentos/montepython_public-3.3/
(base) osvaldo@ekbek:~/Documentos/montepython_public-3.3$ conda activate py27
(py27) osvaldo@ekbek:~/Documentos/montepython_public-3.3$ mpirun -np 4 python montepython/MontePython.py run -N 50000
-p input/twoSFDM/twoSFDM_quadratic_cos_v2.param -o chains/twoSFDM/twoSFDM_quadratic_cos_v2/run_1/
```

Figure M.8: Running the first part of the Monte Python process.

output directory. See Figure M.9.

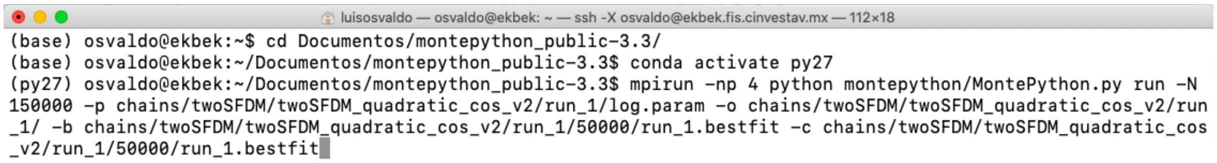

luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 112×18

```
(base) osvaldo@ekbek:~$ cd Documentos/montepython_public-3.3/
(base) osvaldo@ekbek:~/Documentos/montepython_public-3.3$ conda activate py27
(py27) osvaldo@ekbek:~/Documentos/montepython_public-3.3$ python montepython/MontePython.py info chains/twoSFDM/t
twoSFDM_quadratic_cos_v2/run_1/ --want-covmat
```

Figure M.9: Obtainign the covariance matrix.

As a good programming practice, we move the previous chains and the files obtained from the analysis to a new folder to avoid being overwritten. In this case, we create a new folder, called 50000, inside the output folder and move all the files except the log.param into it. This last file has all the information from the run we just did and we will use it for the next part. Once this is done, we move on to the final part.

We start again with mpirun -np 4, then with python montepython/MontePython.py run -N 150000. We also pass the input file with -p, this time it will not be the one in the input directory but the log.param file that is in the output directory. With -o we give the address of the output directory. With -b we pass the .besfit file which, as its name indicates, contains the best fit values of the previous chains. These values will be used as the new mean from which Monte Python will start the inference process. And with -c we pass the covariance matrix. See Figure M.10.


luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 112×18

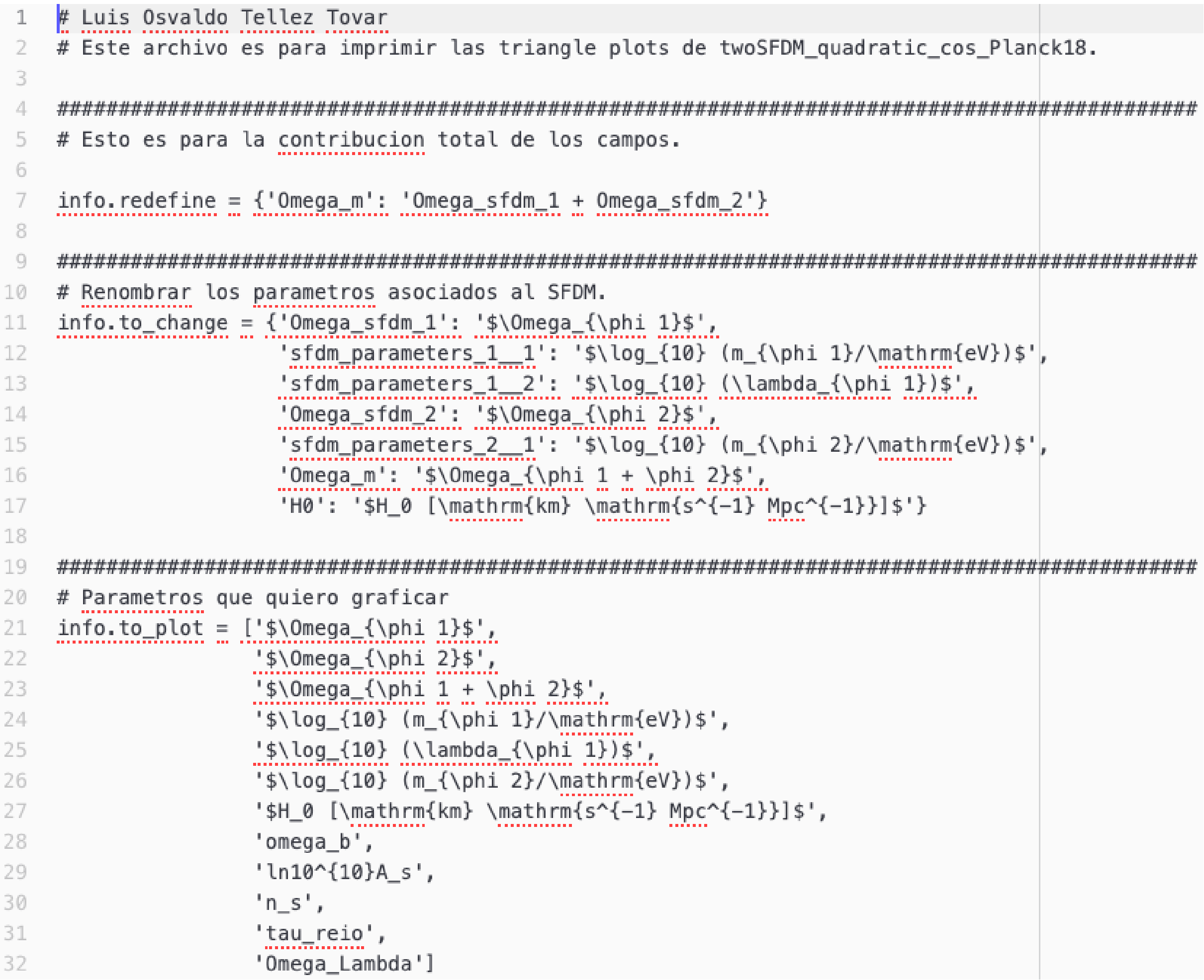
```
(base) osvaldo@ekbek:~$ cd Documentos/montepython_public-3.3/
(base) osvaldo@ekbek:~/Documentos/montepython_public-3.3$ conda activate py27
(py27) osvaldo@ekbek:~/Documentos/montepython_public-3.3$ mpirun -np 4 python montepython/MontePython.py run -N
150000 -p chains/twoSFDM/twoSFDM_quadratic_cos_v2/run_1/log.param -o chains/twoSFDM/twoSFDM_quadratic_cos_v2/run
_1/ -b chains/twoSFDM/twoSFDM_quadratic_cos_v2/run_1/50000/run_1.bestfit -c chains/twoSFDM/twoSFDM_quadratic_cos
_v2/run_1/50000/run_1.bestfit
```

Figure M.10: Running the last part of the Monte Python process.

M.5 Analyze the output

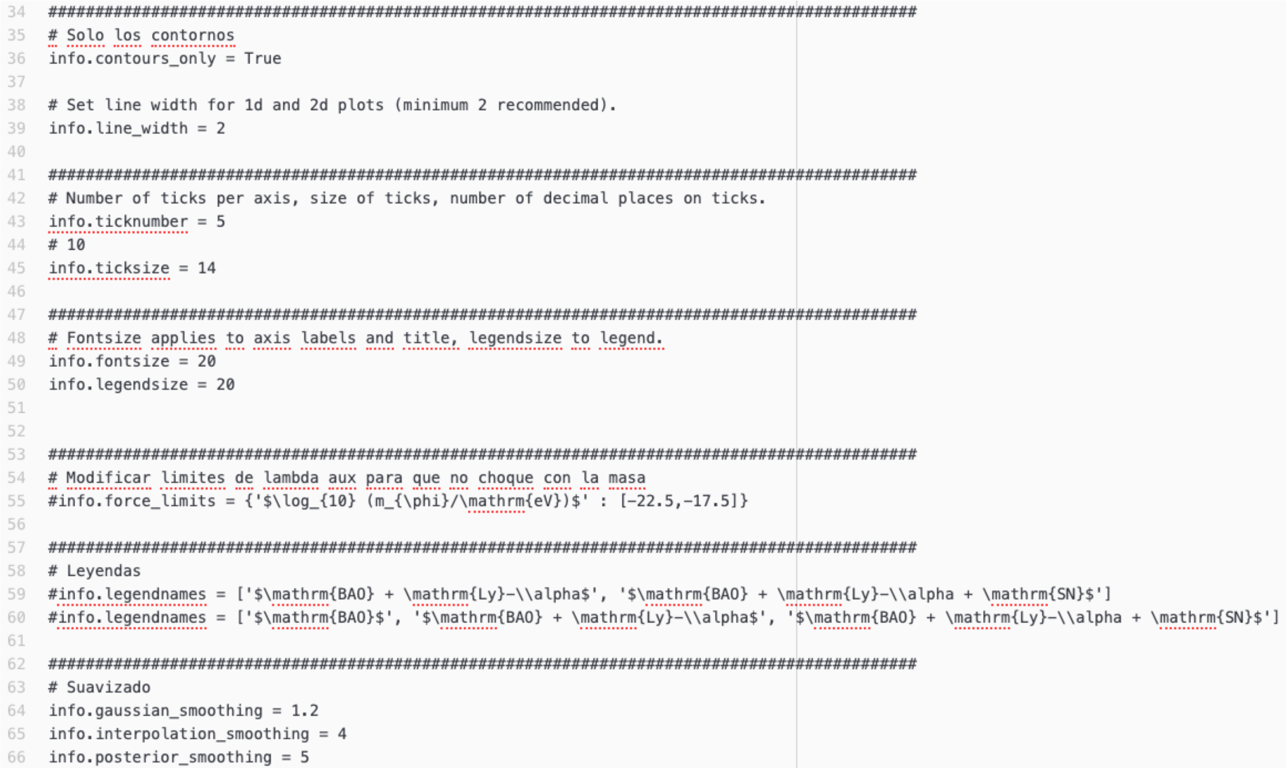
To analyze the final chains we can write the lines in Figure M.9 without the –want-covmat option. This will give us the R-1 value (Gelman-Rubin criterion) and the 1-D and 2-D posteriors plots with all the parameters. However, it is possible to customize the graphs and the parameters that we want to plot. For this we need a file with a .plot extension that is located in the montepython_public-3.3/plot_files directory. The file that contains all the customization options is called example.plot. Here we will show an example of the file that we use. We can redefine and rename parameters and indicate which ones we want to plot, see Figure M.11. We can plot contours only, change ticks and fonts size, change plot limits, put legends, and change curve smoothing, see Figure M.12.

We know that our chains are good enough when the Gelman-Rubin criterion is met; when the acceptance rate is in the range 0.2-0.3, this value can be seen in the .log file in the output directory; and when the plots look smooth.



```
# Luis Osvaldo Tellez Tovar
# Este archivo es para imprimir las triangle plots de twoSFDM_quadratic_cos_Planck18.

##############################################################################################
# Esto es para la contribucion total de los campos.

info.redefine = {'Omega_m': 'Omega_sfdm_1 + Omega_sfdm_2'}

##############################################################################################
# Renombrar los parametros asociados al SFDM.
info.to_change = {'Omega_sfdm_1': '$\Omega_{\phi 1}$',
                  'sfdm_parameters_1__1': '$\log_{10} (m_{\phi 1}/\mathrm{eV})$',
                  'sfdm_parameters_1__2': '$\log_{10} (\lambda_{\phi 1})$',
                  'Omega_sfdm_2': '$\Omega_{\phi 2}$',
                  'sfdm_parameters_2__1': '$\log_{10} (m_{\phi 2}/\mathrm{eV})$',
                  'Omega_m': '$\Omega_{\phi 1 + \phi 2}$',
                  'H0': '$H_0 [\mathrm{km} \mathrm{s^{-1} Mpc^{-1}}]$'}

##############################################################################################
# Parametros que quiero graficar
info.to_plot = ['$\Omega_{\phi 1}$',
                '$\Omega_{\phi 2}$',
                '$\Omega_{\phi 1 + \phi 2}$',
                '$\log_{10} (m_{\phi 1}/\mathrm{eV})$',
                '$\log_{10} (\lambda_{\phi 1})$',
                '$\log_{10} (m_{\phi 2}/\mathrm{eV})$',
                '$H_0 [\mathrm{km} \mathrm{s^{-1} Mpc^{-1}}]$',
                'omega_b',
                'ln10^{10}A_s',
                'n_s',
                'tau_reio',
                'Omega_Lambda']
```

Figure M.11: Example of a .plot file.



```
#############################################################################################
# Solo los contornos
info.contours_only = True

# Set line width for 1d and 2d plots (minimum 2 recommended).
info.line_width = 2

#############################################################################################
# Number of ticks per axis, size of ticks, number of decimal places on ticks.
info.ticknumber = 5
# 10
info.ticksize = 14

#############################################################################################
# Fontsize applies to axis labels and title, legendsize to legend.
info.fontsize = 20
info.legendsize = 20


#############################################################################################
# Modificar limites de lambda aux para que no choque con la masa
#info.force_limits = {'$\log_{10} (m_{\phi}/\mathrm{eV})$' : [-22.5,-17.5]}

#############################################################################################
# Leyendas
#info.legendnames = ['$\mathrm{BAO} + \mathrm{Ly}-\\alpha$', '$\mathrm{BAO} + \mathrm{Ly}-\\alpha + \mathrm{SN}$']
#info.legendnames = ['$\mathrm{BAO}$', '$\mathrm{BAO} + \mathrm{Ly}-\\alpha$', '$\mathrm{BAO} + \mathrm{Ly}-\\alpha + \mathrm{SN}$']

#############################################################################################
# Suavizado
info.gaussian_smoothing = 1.2
info.interpolation_smoothing = 4
info.posterior_smoothing = 5
```

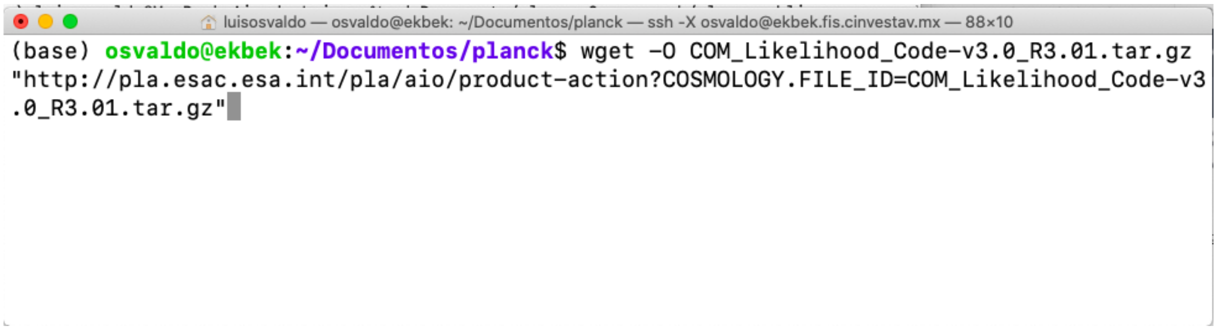
Figure M.12: Example of a .plot file.

Appendix N

Installing the Planck 18 likelihood

In order to use the data from the Planck 18 collaboration in Monte Python we must install the likelihood and the data. We recommend doing this on a server and not on your personal computer because inference with this dataset is computationally intensive. The steps we will show here are for Ubuntu only.

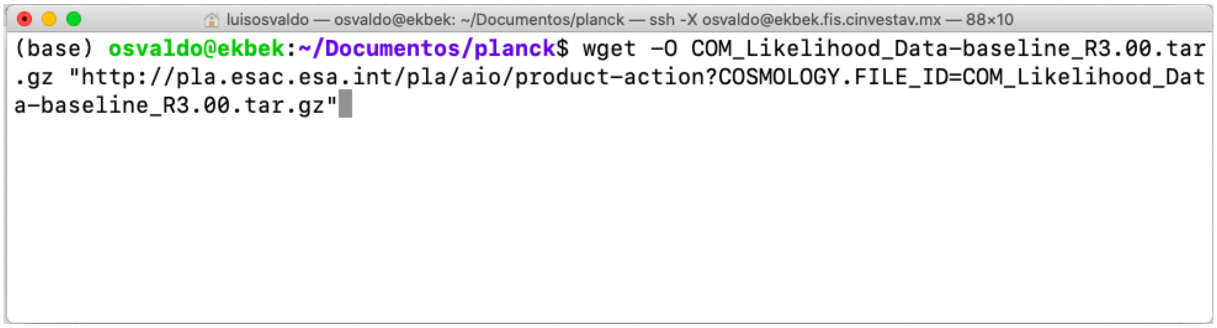
First we move to the folder where we want to download Planck. Then we download the code data with the lines wget -O COM_Likelihood_Code-v3.0_R3.01.tar.gz ”http://pla.esac.esa.int/pla/aio/product-action?COSMOLOGY.FILE_ID=COM_Likelihood_Code-v3.0_R3.01.tar.gz”, see Figure N.1.


luisosvaldo — osvaldo@ekbek: ~/Documentos/planck — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 88×10

```
(base) osvaldo@ekbek:~/Documentos/planck$ wget -O COM_Likelihood_Code-v3.0_R3.01.tar.gz
"http://pla.esac.esa.int/pla/aio/product-action?COSMOLOGY.FILE_ID=COM_Likelihood_Code-v3
.0_R3.01.tar.gz"
```

Figure N.1: Downloading the code files.

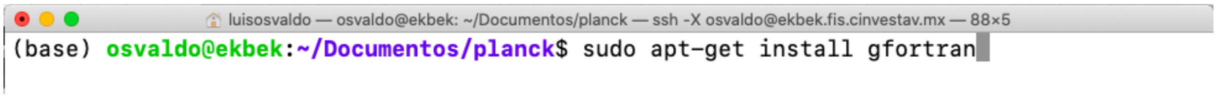
We download the baseline data with wget -O COM_Likelihood_Data-baseline_R3.00.tar.gz ”http://pla.esac.esa.int/pla/aio/product-action?COSMOLOGY.FILE_ID=COM_Likelihood_Data-baseline_R3.00.tar.gz”, see Figure N.2.


luisosvaldo — osvaldo@ekbek: ~/Documentos/planck — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 88×10

```
(base) osvaldo@ekbek:~/Documentos/planck$ wget -O COM_Likelihood_Data-baseline_R3.00.tar
.gz "http://pla.esac.esa.int/pla/aio/product-action?COSMOLOGY.FILE_ID=COM_Likelihood_Dat
a-baseline_R3.00.tar.gz"
```

Figure N.2: Downloading the baseline files.

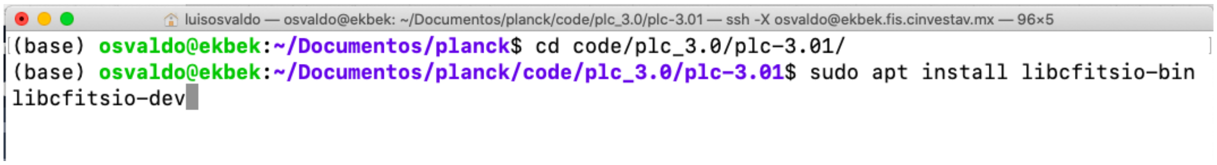
Since these codes are written in fortran, we need to install a compiler for that programming language. We use gfortran which is installed by typing in the terminal sudo apt-get install gfortran as can be seen in Figure N.3.


luisosvaldo — osvaldo@ekbek: ~/Documentos/planck — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 88×5

```
(base) osvaldo@ekbek:~/Documentos/planck$ sudo apt-get install gfortran
```

Figure N.3: Installing the fortran compiler.

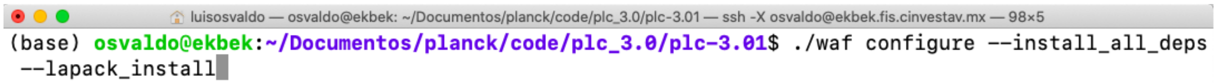
Then we move to the directory code/plc_3.0/plc-3.01 and install the math functions library cfitsio with sudo apt install libcfitsio-bin libcfitsio-dev , as is Figure N.4.


luisosvaldo — osvaldo@ekbek: ~/Documentos/planck/code/plc_3.0/plc-3.01 — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 96×5

```
(base) osvaldo@ekbek:~/Documentos/planck$ cd code/plc_3.0/plc-3.01/
(base) osvaldo@ekbek:~/Documentos/planck/code/plc_3.0/plc-3.01$ sudo apt install libcfitsio-bin
libcfitsio-dev
```

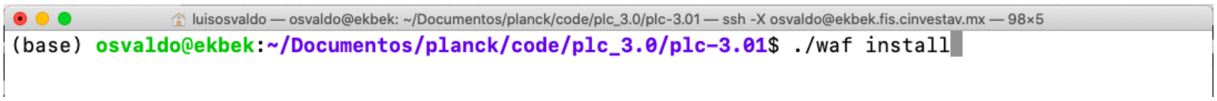
Figure N.4: Installing the library cfitsio.

When this process is done, we are going to write ./waf configure –install_all_deps –lapack_install to install all the dependencies that Planck likelihood needs, see Figure N.5. If there are no errors, a message will be displayed asking us to type ./waf install to finish the installation as in Figure N.6.


luisosvaldo — osvaldo@ekbek: ~/Documentos/planck/code/plc_3.0/plc-3.01 — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 98×5

```
(base) osvaldo@ekbek:~/Documentos/planck/code/plc_3.0/plc-3.01$ ./waf configure --install_all_deps
 --lapack_install
```

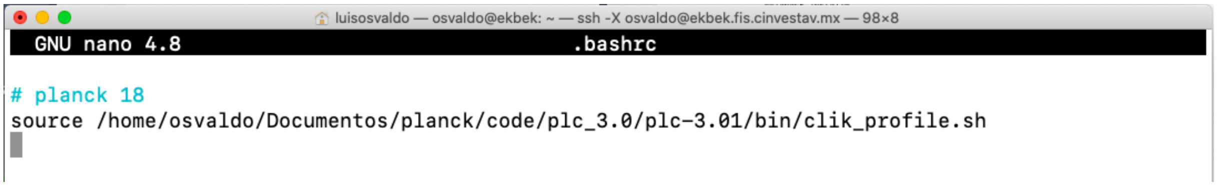
Figure N.5: Installing all the dependencies that Planck likelihood needs.


luisosvaldo — osvaldo@ekbek: ~/Documentos/planck/code/plc_3.0/plc-3.01 — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 98×5

```
(base) osvaldo@ekbek:~/Documentos/planck/code/plc_3.0/plc-3.01$ ./waf install
```

Figure N.6: Finishing the installation.

Finally, we must add the Planck likelihood address to the PATH. In this case, we need to modify the .bashrc file as in Figure N.7.

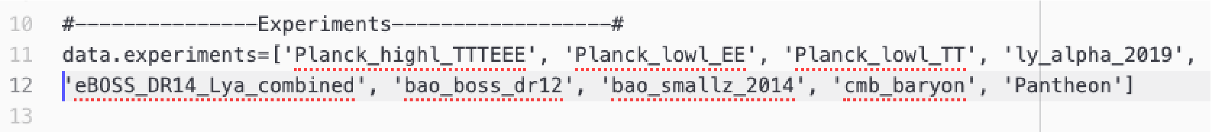

luisosvaldo — osvaldo@ekbek: ~ — ssh -X osvaldo@ekbek.fis.cinvestav.mx — 98×8

```
  GNU nano 4.8                            .bashrc

# planck 18
source /home/osvaldo/Documentos/planck/code/plc_3.0/plc-3.01/bin/clik_profile.sh
```

Figure N.7: Adding the clik likelihood to the PATH.

N.1 Running Monte Python with Planck 18

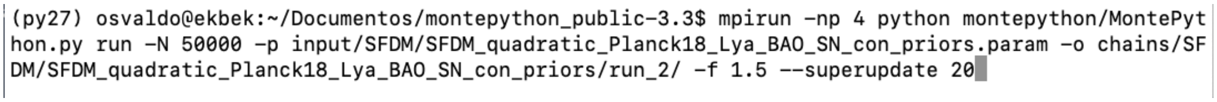
Planck 18 data is added to .param files as shown in Figure N.8 and can be combined with data from other cosmological observations.



```
#---------------Experiments------------------#
data.experiments=['Planck_highl_TTTEEE', 'Planck_lowl_EE', 'Planck_lowl_TT', 'ly_alpha_2019',
'eBOSS_DR14_Lya_combined', 'bao_boss_dr12', 'bao_smallz_2014', 'cmb_baryon', 'Pantheon']

```

Figure N.8: Adding the Planck 18 data to the .param file.

Also, it is recommended to add the -f 1.5 –superupdate 20 options in the first part of the Monte Python process to have a better acceptance rate, see Figure N.9. Although, as in any numerical process of Bayesian inference, this is somewhat empirical.



```
(py27) osvaldo@ekbek:~/Documentos/montepython_public-3.3$ mpirun -np 4 python montepython/MontePyt
hon.py run -N 50000 -p input/SFDM/SFDM_quadratic_Planck18_Lya_BAO_SN_con_priors.param -o chains/SF
DM/SFDM_quadratic_Planck18_Lya_BAO_SN_con_priors/run_2/ -f 1.5 --superupdate 20
```

Figure N.9: Special options for Planck 18.

Appendix O

Extracting the Ly-α data

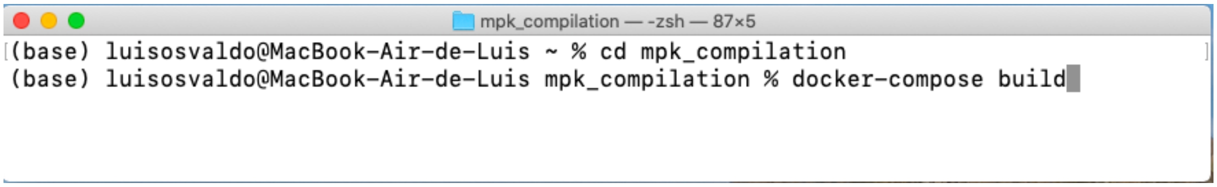
To obtain the Ly-α MPS points that we used in section 3.4, it was necessary to modify a jupyter notebook, which accompanies the reference [86], because this only allowed us to reproduce the plots in that paper but not extract the data or the errors. In this appendix we will review the modification.

The authors of this code saved it in a Docker container. This works as a type of virtual machine in which all the dependencies necessary to run the notebook are already included, thus avoiding compatibility problems between versions and features of different computers. It is important to clarify that Docker was only available for Windows and MasOS at the time of writing this thesis, therefore the process presented here was carried out in MacOS and not in Ubuntu as in the previous appendices.

The first step is to install Docker by following the instructions on its page [193]. We must also install julia because the code is written in this programming language. To install it we just have to follow the instructions that come on its page [194].

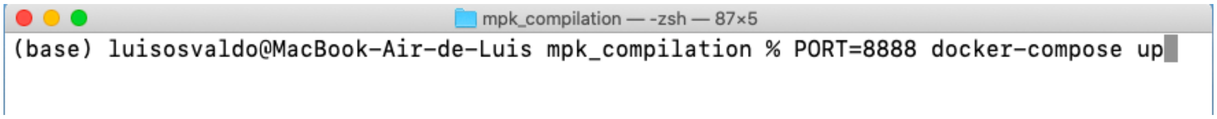
The process that runs this notebook is very heavy, so it is recommended to change the Docker configuration so that it supports at least 8 Gb of RAM. If the computer on which this process is going to be carried out has a higher capacity RAM, we can increase the amount allowed to Docker.

We download the code from [195]. Then we enter the directory where we downloaded it with cd mpk_compilation. Next, we will build the docker container with docker-compose build. See Figure O.1. After the process completes, we type PORT=8888 docker-compose up to activate the Docker container, see Figure O.2. Finally we will use jupyter to open the notebook. Since we installed python before, we must check that the jupyter kernel is julia because if it is python it will not execute the code.



```
[(base) luisosvaldo@MacBook-Air-de-Luis ~ % cd mpk_compilation                        ]
(base) luisosvaldo@MacBook-Air-de-Luis mpk_compilation % docker-compose build
```

Figure O.1: Building the docker container.



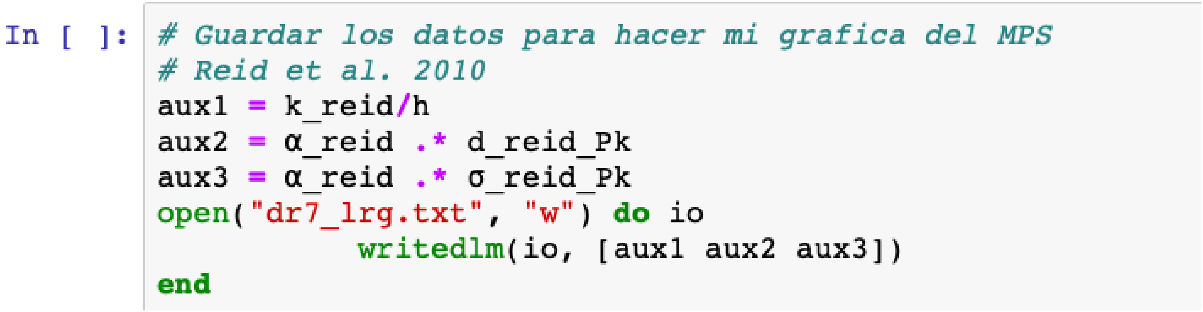
```
(base) luisosvaldo@MacBook-Air-de-Luis mpk_compilation % PORT=8888 docker-compose up
```

Figure O.2: Activating the docker container.

We are not going to review all of the code, just the cells we added to extract the points and save them to .txt files. Before running the added cells, we ran all the previous ones because, as already mentioned, obtaining Ly- α information depends on the model and this process is done in the original cells of the notebook. However, we strongly recommend not running cells below the sections χ^2's and "Make plot" because they require such a large amount of RAM that 8 Gb is not enough. This causes the kernel to die and we have to run again. Since the whole process is slow, we must save as much time as possible.

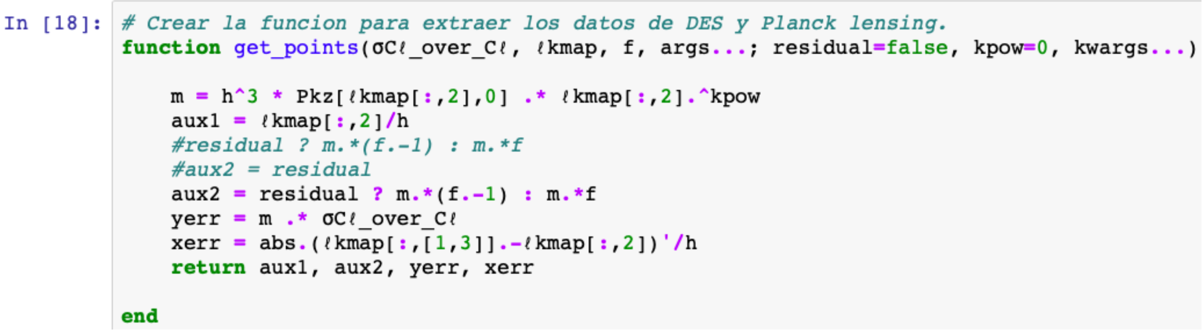
First we are going to save the data from the SDSS DR7 LGR collaboration. Here we only save the results of the section corresponding to this data set in arrays and, in turn, we save these arrays in a .txt file. See Figure O.3.

For the DES and Planck data we need to define a helper function that allows us to extract the data, see Figure O.4.



```
In [ ]: # Guardar los datos para hacer mi grafica del MPS
        # Reid et al. 2010
        aux1 = k_reid/h
        aux2 = α_reid .* d_reid_Pk
        aux3 = α_reid .* σ_reid_Pk
        open("dr7_lrg.txt", "w") do io
                   writedlm(io, [aux1 aux2 aux3])
        end
```

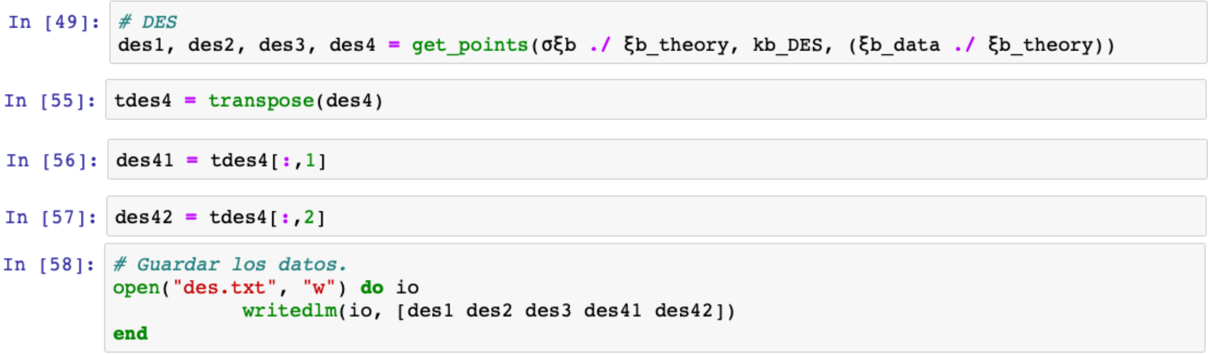
Figure O.3: Saving the SDSS DR7 LGR data in a .txt file.



```
In [18]: # Crear la funcion para extraer los datos de DES y Planck lensing.
         function get_points(σCℓ_over_Cℓ, ℓkmap, f, args...; residual=false, kpow=0, kwargs...)

             m = h^3 * Pkz[ℓkmap[:,2],0] .* ℓkmap[:,2].^kpow
             aux1 = ℓkmap[:,2]/h
             #residual ? m.*(f.-1) : m.*f
             #aux2 = residual
             aux2 = residual ? m.*(f.-1) : m.*f
             yerr = m .* σCℓ_over_Cℓ
             xerr = abs.(ℓkmap[:,[1,3]].-ℓkmap[:,2])'/h
             return aux1, aux2, yerr, xerr

         end
```

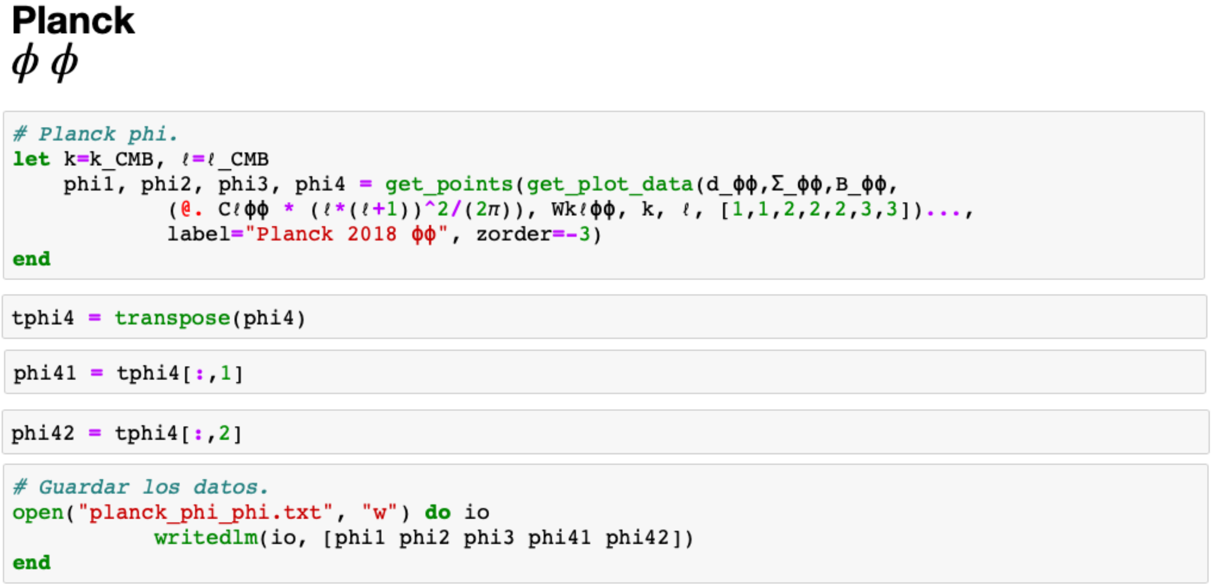
Figure O.4: Defining the auxiliary function to extract the DES and Planck Ly-α data.

The general process is: extract the data with the function we just defined, this will return column arrays (one-dimensional) and matrices. We then get the transpose of these matrices and store each column of the transposed matrix in one-dimensional arrays. Finally, we save all these arrays in a .txt file. For DES data, see Figure O.5. For Planck $\phi\phi$, see Figure O.6. For Planck EE, see Figure O.7. For Planck TT, see Figure O.8.



```
In [49]: # DES
         des1, des2, des3, des4 = get_points(σξb ./ ξb_theory, kb_DES, (ξb_data ./ ξb_theory))

In [55]: tdes4 = transpose(des4)

In [56]: des41 = tdes4[:,1]

In [57]: des42 = tdes4[:,2]

In [58]: # Guardar los datos.
         open("des.txt", "w") do io
                   writedlm(io, [des1 des2 des3 des41 des42])
         end
```

Figure O.5: Extracting the DES data.



Planck
ϕ ϕ

```
# Planck phi.
let k=k_CMB, ℓ=ℓ_CMB
    phi1, phi2, phi3, phi4 = get_points(get_plot_data(d_ϕϕ,Σ_ϕϕ,B_ϕϕ,
            (@. Cℓϕϕ * (ℓ*(ℓ+1))^2/(2π)), Wkℓϕϕ, k, ℓ, [1,1,2,2,2,3,3])...,
            label="Planck 2018 ϕϕ", zorder=-3)
end
```

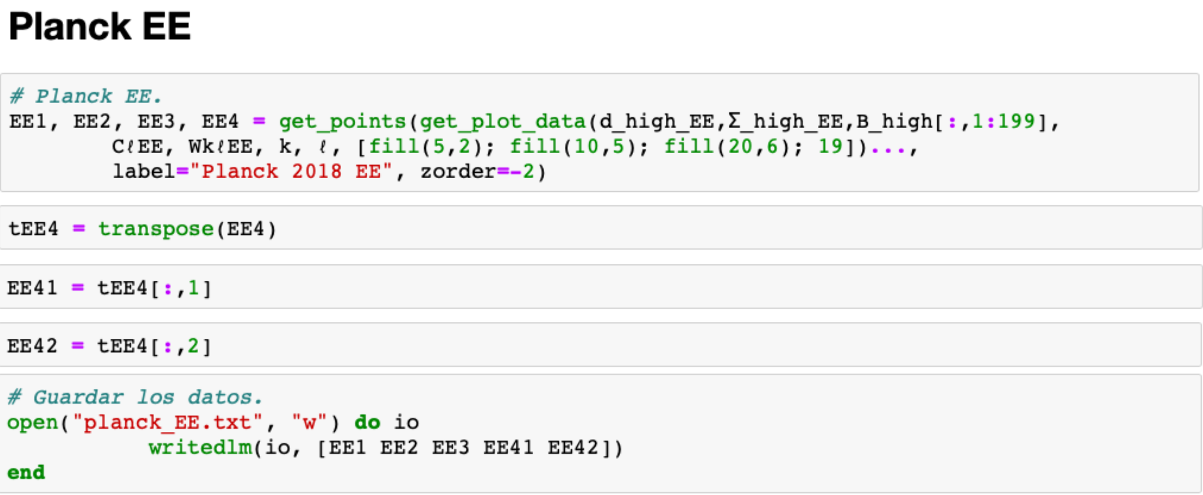
```
tphi4 = transpose(phi4)
```

```
phi41 = tphi4[:,1]
```

```
phi42 = tphi4[:,2]
```

```
# Guardar los datos.
open("planck_phi_phi.txt", "w") do io
         writedlm(io, [phi1 phi2 phi3 phi41 phi42])
end
```

Figure O.6: Extracting the Planck $\phi\phi$ data.



Planck EE

```
# Planck EE.
EE1, EE2, EE3, EE4 = get_points(get_plot_data(d_high_EE,Σ_high_EE,B_high[:,1:199],
        CℓEE, WkℓEE, k, ℓ, [fill(5,2); fill(10,5); fill(20,6); 19])...,
        label="Planck 2018 EE", zorder=-2)
```

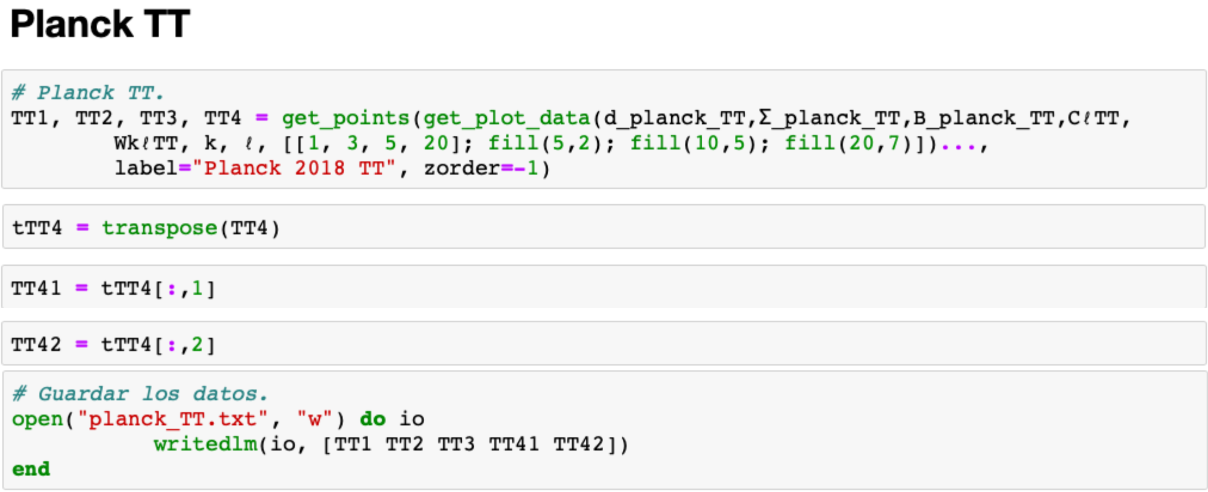
```
tEE4 = transpose(EE4)
```

```
EE41 = tEE4[:,1]
```

```
EE42 = tEE4[:,2]
```

```
# Guardar los datos.
open("planck_EE.txt", "w") do io
           writedlm(io, [EE1 EE2 EE3 EE41 EE42])
end
```

Figure O.7: Extracting the Planck EE data.



Planck TT

```
# Planck TT.
TT1, TT2, TT3, TT4 = get_points(get_plot_data(d_planck_TT,Σ_planck_TT,B_planck_TT,CℓTT,
        WkℓTT, k, ℓ, [[1, 3, 5, 20]; fill(5,2); fill(10,5); fill(20,7)])...,
        label="Planck 2018 TT", zorder=-1)
```

```
tTT4 = transpose(TT4)
```

```
TT41 = tTT4[:,1]
```

```
TT42 = tTT4[:,2]
```

```
# Guardar los datos.
open("planck_TT.txt", "w") do io
           writedlm(io, [TT1 TT2 TT3 TT41 TT42])
end
```

Figure O.8: Extracting the Planck TT data.

Bibliography

[1] F. Zwicky, *On the Masses of Nebulae and of Clusters of Nebulae*, *Astrophys. J.* **86** (1937) 217.

[2] V.C. Rubin and J. Ford, W. Kent, *Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions*, **159** (1970) 379.

[3] V.C. Rubin, *Dark matter in spiral galaxies*, *Scientific American* **248** (1983) 96.

[4] A. Liddle, *An Introduction to Modern Cosmology*, John Wiley and Sons Ltd, United Kingdom (2003).

[5] E. Hubble, *A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae*, *Proceedings of the National Academy of Science* **15** (1929) 168.

[6] Supernova Search Team collaboration, *Observational evidence from supernovae for an accelerating universe and a cosmological constant*, *Astron. J.* **116** (1998) 1009 [astro-ph/9805201].

[7] Supernova Cosmology Project collaboration, *Measurements of Ω and Λ from 42 high redshift supernovae*, *Astrophys. J.* **517** (1999) 565 [astro-ph/9812133].

[8] D.H. Weinberg, M.J. Mortonson, D.J. Eisenstein, C. Hirata, A.G. Riess and E. Rozo, *Observational probes of cosmic acceleration*, *Phys. Rept.* **530** (2013) 87 [1201.2434].

[9] M. Li, X.-D. Li, S. Wang and Y. Wang, *Dark Energy: A Brief Review*, *Front. Phys. (Beijing)* **8** (2013) 828 [1209.0922].

[10] Planck collaboration, *Planck 2018 results. VI. Cosmological parameters*, *Astron. Astrophys.* **641** (2020) A6 [1807.06209].

[11] L.E. Padilla, L.O. Tellez, L.A. Escamilla and J.A. Vazquez, *Cosmological Parameter Inference with Bayesian Statistics*, *Universe* **7** (2021) 213 [1903.11127].

[12] L.O. Téllez-Tovar, T. Matos and J.A. Vázquez, *Cosmological constraints on the Multi Scalar Field Dark Matter model*, 2112.09337.

[13] T. Matos and L.O. Té llez-Tovar, *The cosmic microwave background and mass power spectrum profiles for a novel and efficient model of dark energy*, *Revista Mexicana de Física* **68** (2022) .

[14] V. Trimble, *The 1920 shapley-curtis discussion: background, issues, and aftermath*, *Publications of the Astronomical Society of the Pacific* **107** (1995) 1133.

[15] http://www.soloentendidos.com/wp-content/uploads/2015/12/04-12-2015-12-13-58.jpg.

[16] R. Wald, *General relativity. the university of chicago*, *Chicago. Sect* **6** (1984) .

[17] D. Baumann, *Cosmology, part iii mathematical tripos*, *University lecture notes* (2014) .

[18] L. Iorio, *Editorial for the Special Issue 100 Years of Chronogeometrodynamics: The Status of the Einstein's Theory of Gravitation in Its Centennial Year*, *Universe* **1** (2015) 38 [1504.05789].

[19] I. Debono and G.F. Smoot, *General Relativity and Cosmology: Unsolved Questions and Future Directions*, *Universe* **2** (2016) 23 [1609.09781].

[20] A. Vázquez-González and T. Matos, *La materia oscura del universo: retos y perspectivas*, *Revista mexicana de física E* **54** (2008) 193.

[21] K. Arun, S. Gudennavar and C. Sivaram, *Dark matter, dark energy, and alternate models: A review*, *Advances in Space Research* **60** (2017) 166.

[22] O. Lahav and A.R. Liddle, *The Cosmological Parameters 2014*, `1401.1389`.

[23] D.H. Weinberg, J.S. Bullock, F. Governato, R. Kuzio de Naray and A.H.G. Peter, *Cold dark matter: controversies on small scales*, *Proc. Nat. Acad. Sci.* **112** (2015) 12249 [`1306.0913`].

[24] P. Bull et al., *Beyond ΛCDM: Problems, solutions, and the road ahead*, *Phys. Dark Univ.* **12** (2016) 56 [`1512.05356`].

[25] S.Y. Kim, A.H.G. Peter and J.R. Hargis, *Missing Satellites Problem: Completeness Corrections to the Number of Satellite Galaxies in the Milky Way are Consistent with Cold Dark Matter Predictions*, *Phys. Rev. Lett.* **121** (2018) 211302 [`1711.06267`].

[26] DES collaboration, *Milky Way Satellite Census. III. Constraints on Dark Matter Properties from Observations of Milky Way Satellite Galaxies*, *Phys. Rev. Lett.* **126** (2021) 091101 [`2008.00022`].

[27] T. Matos, J.-R. Luevano, I. Quiros, L.A. Urena-Lopez and J.A. Vazquez, *Dynamics of Scalar Field Dark Matter With a Cosh-like Potential*, *Phys. Rev. D* **80** (2009) 123521 [`0906.0396`].

[28] S.-J. Sin, *Late-time phase transition and the galactic halo as a bose liquid*, *Physical Review D* **50** (1994) 3650 [`hep-ph/9409267`].

[29] F.S. Guzman, T. Matos and H. Villegas, *Scalar fields as dark matter in spiral galaxies: comparison with experiments*, *Astronomische Nachrichten: News in Astronomy and Astrophysics* **320** (1999) 97.

[30] T. Matos and F.S. Guzman, *Scalar fields as dark matter in spiral galaxies*, *Class. Quant. Grav.* **17** (2000) L9 [gr-qc/9810028].

[31] J.-w. Lee and I.-g. Koh, *Galactic halos as boson stars*, *Phys. Rev. D* **53** (1996) 2236 [hep-ph/9507385].

[32] T. Matos, A. Vazquez-Gonzalez and J. Magana, *ϕ^2 as Dark Matter*, *Mon. Not. Roy. Astron. Soc.* **393** (2009) 1359 [0806.0683].

[33] D.N. Spergel and P.J. Steinhardt, *Observational evidence for selfinteracting cold dark matter*, *Phys. Rev. Lett.* **84** (2000) 3760 [astro-ph/9909386].

[34] T. Gonzalez, T. Matos, I. Quiros and A. Vazquez-Gonzalez, *Self-interacting Scalar Field Trapped in a Randall-Sundrum Braneworld: The Dynamical Systems Perspective*, *Phys. Lett. B* **676** (2009) 161 [0812.1734].

[35] L.E. Padilla, J.A. Vázquez, T. Matos and G. Germán, *Scalar Field Dark Matter Spectator During Inflation: The Effect of Self-interaction*, *JCAP* **05** (2019) 056 [1901.00947].

[36] A. De Felice and S. Tsujikawa, *f(R) theories*, *Living Rev. Rel.* **13** (2010) 3 [1002.4928].

[37] S. Nojiri, S. Odintsov and V. Oikonomou, *Modified gravity theories on a nutshell: inflation, bounce and late-time evolution*, *Physics Reports* **692** (2017) 1 [1705.11098].

[38] M.A. Garcia-Aspeitia, A. Hernandez-Almada, J. Magaña, M.H. Amante, V. Motta and C. Martínez-Robles, *Brane with variable tension as a possible solution to the*

problem of the late cosmic acceleration, Phys. Rev. D **97** (2018) 101301 [1804.05085].

[39] J.A. Vázquez, D. Tamayo, A.A. Sen and I. Quiros, *Bayesian model selection on scalar ϵ-field dark energy*, Phys. Rev. D **103** (2021) 043506 [2009.01904].

[40] O. Akarsu, N. Katirci, A.A. Sen and J.A. Vazquez, *Scalar field emulator via anisotropically deformed vacuum energy: Application to dark energy*, 2004.14863.

[41] S. Tsujikawa, *Quintessence: A Review*, Class. Quant. Grav. **30** (2013) 214003 [1304.1961].

[42] J. Yoo and Y. Watanabe, *Theoretical Models of Dark Energy*, Int. J. Mod. Phys. D **21** (2012) 1230002 [1212.4726].

[43] B. Feng, *The quintom model of dark energy*, astro-ph/0602156.

[44] O. Akarsu, N. Katırcı, N. Özdemir and J.A. Vázquez, *Anisotropic massive Brans-Dicke gravity extension of the standard ΛCDM model*, Eur. Phys. J. C **80** (2020) 32 [1903.06679].

[45] M. Sharif and S. Azeem, *Dark energy models and cosmic acceleration with anisotropic universe in $f(t)$ gravity*, Communications in Theoretical Physics **61** (2014) 482.

[46] D. Saadeh, S.M. Feeney, A. Pontzen, H.V. Peiris and J.D. McEwen, *How isotropic is the universe?*, Physical review letters **117** (2016) 131302 [1605.07178].

[47] S. Linden and J.-M. Virey, *A test of the CPL parameterization for rapid dark energy equation of state transitions*, Phys. Rev. D **78** (2008) 023526 [0804.0389].

[48] R.J. Scherrer, *Mapping the Chevallier-Polarski-Linder parametrization onto Physical Dark Energy Models*, Phys. Rev. D **92** (2015) 043001 [1505.05781].

[49] D. Tamayo and J.A. Vazquez, *Fourier-series expansion of the dark-energy equation of state*, *Mon. Not. Roy. Astron. Soc.* **487** (2019) 729 [1901.08679].

[50] J. Alberto Vazquez, M. Bridges, M.P. Hobson and A.N. Lasenby, *Reconstruction of the Dark Energy equation of state*, *JCAP* **09** (2012) 020 [1205.0847].

[51] S. Hee, J.A. Vázquez, W.J. Handley, M.P. Hobson and A.N. Lasenby, *Constraining the dark energy equation of state using Bayes theorem and the Kullback–Leibler divergence*, *Mon. Not. Roy. Astron. Soc.* **466** (2017) 369 [1607.00270].

[52] J.A. Vazquez, S. Hee, M.P. Hobson, A.N. Lasenby, M. Ibison and M. Bridges, *Observational constraints on conformal time symmetry, missing matter and double dark energy*, *JCAP* **07** (2018) 062 [1208.2542].

[53] S.-J. Sin, *Late-time phase transition and the galactic halo as a bose liquid*, *Phys. Rev. D* **50** (1994) 3650.

[54] J.-w. Lee and I.-g. Koh, *Galactic halos as boson stars*, *Phys. Rev. D* **53** (1996) 2236 [hep-ph/9507385].

[55] F. Guzmán, T. Matos and H. Villegas, *Scalar fields as dark matter in spiral galaxies: comparison with experiments*, *Astronomische Nachrichten* **320** (1999) 97.

[56] T. Matos and F.S. Guzman, *Scalar fields as dark matter in spiral galaxies*, *Class. Quant. Grav.* **17** (2000) L9 [gr-qc/9810028].

[57] W. Hu, R. Barkana and A. Gruzinov, *Cold and fuzzy dark matter*, *Phys. Rev. Lett.* **85** (2000) 1158 [astro-ph/0003365].

[58] C.G. Boehmer and T. Harko, *Can dark matter be a Bose-Einstein condensate?*, *JCAP* **06** (2007) 025 [0705.4158].

[59] L. Hui, J.P. Ostriker, S. Tremaine and E. Witten, *Ultralight scalars as cosmological dark matter*, *Phys. Rev. D* **95** (2017) 043541 [1610.08297].

[60] T. Matos and F.S. Guzman, *Scalar fields as dark matter in spiral galaxies*, *Class. Quant. Grav.* **17** (2000) L9 [gr-qc/9810028].

[61] J. Magaña and T. Matos, *A brief Review of the Scalar Field Dark Matter model*, in *Journal of Physics Conference Series*, vol. 378 of *Journal of Physics Conference Series*, p. 012012, Aug., 2012, DOI [1201.6107].

[62] T. Matos, A. Vazquez-Gonzalez and J. Magana, ϕ^2 *as Dark Matter*, *Mon. Not. Roy. Astron. Soc.* **393** (2009) 1359 [0806.0683].

[63] A. Suárez, V.H. Robles and T. Matos, *A Review on the Scalar Field/Bose-Einstein Condensate Dark Matter Model*, *Astrophys. Space Sci. Proc.* **38** (2014) 107 [1302.0903].

[64] L.A. Ureña López, *Brief Review on Scalar Field Dark Matter Models*, *Front. Astron. Space Sci.* **6** (2019) 47.

[65] L.A. Ureña López and A.X. Gonzalez-Morales, *Towards accurate cosmological predictions for rapidly oscillating scalar fields as dark matter*, *JCAP* **07** (2016) 048 [1511.08195].

[66] E. Aubourg et al., *Cosmological implications of baryon acoustic oscillation measurements*, *Phys. Rev. D* **92** (2015) 123516 [1411.1074].

[67] Planck collaboration, *Planck 2013 results. XVI. Cosmological parameters*, *Astron. Astrophys.* **571** (2014) A16 [1303.5076].

[68] "What is a Supernova." https://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-a-supernova.html, 2021.

[69] A.G. Riess, A.V. Filippenko, P. Challis, A. Clocchiatti, A. Diercks, P.M. Garnavich et al., *Observational evidence from supernovae for an accelerating universe and a*

cosmological constant, *The Astronomical Journal* **116** (1998) 1009 [astro-ph/9805201].

[70] Supernova Search Team collaboration, *Supernova limits on the cosmic equation of state*, *Astrophys. J.* **509** (1998) 74 [astro-ph/9806396].

[71] Supernova Cosmology Project collaboration, *Measurements of Ω and Λ from 42 high redshift supernovae*, *Astrophys. J.* **517** (1999) 565 [astro-ph/9812133].

[72] SDSS collaboration, *Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples*, *Astron. Astrophys.* **568** (2014) A22 [1401.4064].

[73] R.G. Vishwakarma and J.V. Narlikar, *Is it no Longer Necessary to Test Cosmologies with Type Ia Supernovae?*, *Universe* **4** (2018) .

[74] J.C. Mather et al., *A Preliminary measurement of the Cosmic Microwave Background spectrum by the Cosmic Background Explorer (COBE) satellite*, *Astrophys. J. Lett.* **354** (1990) L37.

[75] C.L. Bennett et al., *Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Are There Cosmic Microwave Background Anomalies?*, *Astrophys. J. Suppl.* **192** (2011) 17 [1001.4758].

[76] Planck collaboration, *Planck 2018 results. VII. Isotropy and Statistics of the CMB*, *Astron. Astrophys.* **641** (2020) A7 [1906.02552].

[77] Planck collaboration, *Planck 2018 results. X. Constraints on inflation*, *Astron. Astrophys.* **641** (2020) A10 [1807.06211].

[78] S. Staggs, J. Dunkley and L. Page, *Recent discoveries from the cosmic microwave background: a review of recent progress*, *Reports on Progress in Physics* **81** (2018) 044901.

[79] D. Samtleben, S. Staggs and B. Winstein, *The cosmic microwave background for pedestrians: A review for particle and nuclear physicists*, *Annu. Rev. Nucl. Part. Sci.* **57** (2007) 245.

[80] M. Kamionkowski and A. Kosowsky, *The cosmic microwave background and particle physics*, *Annual Review of Nuclear and Particle Science* **49** (1999) 77.

[81] A.L. Erickcek, S.M. Carroll and M. Kamionkowski, *Superhorizon perturbations and the cosmic microwave background*, *Physical Review D* **78** (2008) 083012.

[82] M. Kamionkowski, A. Kosowsky and A. Stebbins, *Statistics of cosmic microwave background polarization*, *Physical Review D* **55** (1997) 7368.

[83] https://w.astro.berkeley.edu/~jcohn/lya.html.

[84] https://escholarship.org/content/qt3mx4f1bs/qt3mx4f1bs.pdf?t=msyiak.

[85] D.H. Weinberg, R. Dave, N. Katz and J.A. Kollmeier, *The Lyman - alpha forest as a cosmological tool*, *AIP Conf. Proc.* **666** (2003) 157 [astro-ph/0301186].

[86] S. Chabanier, M. Millea and N. Palanque-Delabrouille, *Matter power spectrum: from Lyα forest to CMB scales*, *Mon. Not. Roy. Astron. Soc.* **489** (2019) 2247 [1905.08103].

[87] V. Iršič, M. Viel, M.G. Haehnelt, J.S. Bolton and G.D. Becker, *First constraints on fuzzy dark matter from Lyman-α forest data and hydrodynamical simulations*, *Phys. Rev. Lett.* **119** (2017) 031302 [1703.04683].

[88] K.K. Rogers and H.V. Peiris, *Strong Bound on Canonical Ultralight Axion Dark Matter from the Lyman-Alpha Forest*, *Phys. Rev. Lett.* **126** (2021) 071302 [2007.12705].

[89] V.H. Robles and T. Matos, *Exact Solution to Finite Temperature SFDM: Natural Cores without Feedback*, **763** (2013) 19 [1207.5858].

[90] A. Hernández-Almada and M.A. García-Aspeitia, *Multistate scalar field dark matter and its correlation with galactic properties*, *Int. J. Mod. Phys. D* **27** (2017) 1850031 [1711.01388].

[91] T. Matos, A. Vazquez-Gonzalez and J. Magana, *Study of several potentials as scalar field dark matter candidates*, *AIP Conf. Proc.* **1083** (2008) 144.

[92] B. Li, T. Rindler-Daller and P.R. Shapiro, *Cosmological Constraints on Bose-Einstein-Condensed Scalar Field Dark Matter*, *Phys. Rev. D* **89** (2014) 083536 [1310.6061].

[93] A. Suárez and P.-H. Chavanis, *Hydrodynamic representation of the Klein-Gordon-Einstein equations in the weak field limit: General formalism and perturbations analysis*, *Phys. Rev. D* **92** (2015) 023510 [1503.07437].

[94] A. Suárez and P.-H. Chavanis, *Cosmological evolution of a complex scalar field with repulsive or attractive self-interaction*, *Phys. Rev. D* **95** (2017) 063515 [1608.08624].

[95] F.X.L. Cedeño, A.X. González-Morales and L.A. Ureña López, *Cosmological signatures of ultralight dark matter with an axionlike potential*, *Phys. Rev. D* **96** (2017) 061301 [1703.10180].

[96] F.X. Linares Cedeño, A.X. González-Morales and L.A. Ureña López, *Ultralight DM bosons with an axion-like potential: scale-dependent constraints revisited*, *JCAP* **01** (2021) 051 [2006.05037].

[97] G.G. Ross, G. German and J.A. Vazquez, *Hybrid Natural Inflation*, *JHEP* **05** (2016) 010 [1601.03221].

[98] T. Matos, J.-R. Luévano, I. Quiros, L.A. Ureña-López and J.A. Vázquez, *Dynamics of scalar field dark matter with a cosh-like potential*, *Physical Review D* **80** (2009) .

[99] L.A. Ureña López, *Scalar field dark matter with a cosh potential, revisited*, *JCAP* **06** (2019) 009 [1904.03318].

[100] R. Hlozek, D. Grin, D.J.E. Marsh and P.G. Ferreira, *A search for ultralight axions using precision cosmological data*, *Phys. Rev. D* **91** (2015) 103512 [1410.2896].

[101] A. Paredes and H. Michinel, *Interference of Dark Matter Solitons and Galactic Offsets*, *Phys. Dark Univ.* **12** (2016) 50 [1512.05121].

[102] A.X. González-Morales, D.J.E. Marsh, J. Peñarrubia and L.A. Ureña López, *Unbiased constraints on ultralight axion mass from dwarf spheroidal galaxies*, *Mon. Not. Roy. Astron. Soc.* **472** (2017) 1346 [1609.05856].

[103] V. Lora, J. Magana, A. Bernal, F.J. Sanchez-Salcedo and E.K. Grebel, *On the mass of ultra-light bosonic dark matter from galactic dynamics*, *JCAP* **02** (2012) 011 [1110.2684].

[104] E. Calabrese and D.N. Spergel, *Ultra-Light Dark Matter in Ultra-Faint Dwarf Galaxies*, *Mon. Not. Roy. Astron. Soc.* **460** (2016) 4397 [1603.07321].

[105] A. Sarkar, R. Mondal, S. Das, S.K. Sethi, S. Bharadwaj and D.J.E. Marsh, *The effects of the small-scale DM power on the cosmological neutral hydrogen (HI) distribution at high redshifts*, *JCAP* **04** (2016) 012 [1512.03325].

[106] E. Armengaud, N. Palanque-Delabrouille, C. Yèche, D.J.E. Marsh and J. Baur, *Constraining the mass of light bosonic dark matter using SDSS Lyman-α forest*, *Mon. Not. Roy. Astron. Soc.* **471** (2017) 4606 [1703.09126].

[107] D.J.E. Marsh, *Axion Cosmology*, *Phys. Rept.* **643** (2016) 1 [1510.07633].

[108] P. Mocz et al., *First star-forming structures in fuzzy cosmic filaments*, *Phys. Rev. Lett.* **123** (2019) 141301 [1910.01653].

[109] O. Müller, M.S. Pawlowski, H. Jerjen and F. Lelli, *A whirling plane of satellite galaxies around Centaurus A challenges cold dark matter cosmology*, *Science* **359** (2018) 534 [1802.00081].

[110] L.E. Strigari, J.S. Bullock, M. Kaplinghat, J.D. Simon, M. Geha, B. Willman et al., *A common mass scale for satellite galaxies of the Milky Way*, *Nature* **454** (2008) 1096 [0808.3772].

[111] I.J. Allali, M.P. Hertzberg and F. Rompineve, *Dark sector to restore cosmological concordance*, *Phys. Rev. D* **104** (2021) L081303 [2104.12798].

[112] K. Blum and L. Teodori, *Gravitational lensing H0 tension from ultralight axion galactic cores*, *Phys. Rev. D* **104** (2021) 123011 [2105.10873].

[113] A. Arvanitaki, S. Dimopoulos, S. Dubovsky, N. Kaloper and J. March-Russell, *String Axiverse*, *Phys. Rev. D* **81** (2010) 123530 [0905.4720].

[114] H.N. Luu, S.H.H. Tye and T. Broadhurst, *Multiple Ultralight Axionic Wave Dark Matter and Astronomical Structures*, *Phys. Dark Univ.* **30** (2020) 100636 [1811.03771].

[115] A. Paliathanasis and M. Tsamparlis, *Two scalar field cosmology: Conservation laws and exact solutions*, *Phys. Rev. D* **90** (2014) 043529 [1408.1798].

[116] D. Benisty and E.I. Guendelman, *Two scalar fields inflation from scale-invariant gravity with modified measure*, *Class. Quant. Grav.* **36** (2019) 095001 [1809.09866].

[117] K. Bamba, S.D. Odintsov and P.V. Tretyakov, *Inflation in a conformally-invariant two-scalar-field theory with an extra R^2 term*, *Eur. Phys. J. C* **75** (2015) 344 [1505.00854].

[118] J.A. Vázquez, L.E. Padilla and T. Matos, *Inflationary Cosmology: From Theory to Observations*, 1810.09934.

[119] O. Bertolami, P. Carrilho and J. Paramos, *Two-scalar-field model for the interaction of dark energy and dark matter*, *Phys. Rev. D* **86** (2012) 103522 [1206.2589].

[120] V.M. Mehta, M. Demirtas, C. Long, D.J.E. Marsh, L. McAllister and M.J. Stott, *Superradiance in string theory*, *JCAP* **07** (2021) 033 [2103.06812].

[121] M. Cicoli, V. Guidetti, N. Righi and A. Westphal, *Fuzzy Dark Matter Candidates from String Theory*, 2110.02964.

[122] E. Gutiérrez-Luna, B. Carvente, V. Jaramillo, J. Barranco, C. Escamilla-Rivera, C. Espinoza et al., *Scalar field dark matter with two components: combined approach from particle physics and cosmology*, 2110.10258.

[123] NOvA collaboration, *Constraints on Oscillation Parameters from ν_e Appearance and ν_μ Disappearance in NOvA*, *Phys. Rev. Lett.* **118** (2017) 231801 [1703.03328].

[124] P.F. de Salas, D.V. Forero, C.A. Ternes, M. Tortola and J.W.F. Valle, *Status of neutrino oscillations 2018: 3σ hint for normal mass ordering and improved CP sensitivity*, *Phys. Lett. B* **782** (2018) 633 [1708.01186].

[125] Super-Kamiokande collaboration, *Atmospheric neutrino oscillation analysis with external constraints in Super-Kamiokande I-IV*, *Phys. Rev. D* **97** (2018) 072001 [1710.09126].

[126] F. Capozzi, E. Lisi, A. Marrone and A. Palazzo, *Current unknowns in the three neutrino framework*, *Prog. Part. Nucl. Phys.* **102** (2018) 48 [1804.09678].

[127] L.A. Ureña López, *New perturbative method for analytical solutions in single-field models of inflation*, *Phys. Rev. D* **94** (2016) 063532 [1512.07142].

[128] T. Cookmeyer, J. Cookmeyer, D. Grin and T.L. Smith, *How sound are our ultralight axion approximations?*, *Phys. Rev. D* **101** (2020) 023501 [1909.11094].

[129] S. Passaglia and W. Hu, *Accurate effective fluid approximation for ultralight axions*, 2201.10238.

[130] C.-P. Ma and E. Bertschinger, *Cosmological perturbation theory in the synchronous and conformal Newtonian gauges*, *Astrophys. J.* **455** (1995) 7 [astro-ph/9506072].

[131] D. Blas, J. Lesgourgues and T. Tram, *The cosmic linear anisotropy solving system (class). part ii: Approximation schemes*, *Journal of Cosmology and Astroparticle Physics* **2011** (2011) 034–034.

[132] https://github.com/losvaldote/two_sfdm_class.

[133] S. Dodelson, *Modern Cosmology* (2003).

[134] E.W. Kolb and M.S. Turner, *The early universe*, vol. 69 (1990).

[135] Planck collaboration, *Planck 2018 results. I. Overview and the cosmological legacy of Planck*, *Astron. Astrophys.* **641** (2020) A1 [1807.06205].

[136] http://pla.esac.esa.int/pla/#home.

[137] B. Kleijin, *Bayesian statistics*, *University of Amsterdam lecture notes* **11** (2013) .

[138] A. Heavens, *Statistical techniques in cosmology*, *arXiv e-prints* (2009) [0906.0664].

[139] R. Trotta, *Bayes in the sky: Bayesian inference and model selection in cosmology*, *Contemp. Phys.* **49** (2008) 71 [0803.4089].

[140] L. Verde, *Statistical methods in cosmology*, vol. 800, pp. 147–177 (2010), DOI [0911.3105].

[141] R. Trotta, *Bayesian Methods in Cosmology*, 1701.01467.

[142] A.H. Jaffe, *H0 and odds on cosmology*, *Astrophys. J.* **471** (1996) 24 [astro-ph/9501070].

[143] WMAP collaboration, *First year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Parameter estimation methodology*, *Astrophys. J. Suppl.* **148** (2003) 195 [astro-ph/0302218].

[144] G. D'Agostini, *Probability and measurement uncertainty in physics: A Bayesian primer*, hep-ph/9512295.

[145] S. Sharma, *Markov Chain Monte Carlo Methods for Bayesian Data Analysis in Astronomy*, **55** (2017) 213 [1706.01629].

[146] A.R. Liddle, *How many cosmological parameters?*, *Mon. Not. Roy. Astron. Soc.* **351** (2004) L49 [astro-ph/0401198].

[147] J.A. Vazquez, A.N. Lasenby, M. Bridges and M.P. Hobson, *A Bayesian study of the primordial power spectrum from a novel closed universe model*, *Mon. Not. Roy. Astron. Soc.* **422** (2012) 1948 [1103.4619].

[148] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*, Cambridge university press (2007).

[149] R. Medel, I. Gomez, J.A. Vazquez and R. Garcia, *An introduction to markov chain monte carlo*, *Boletin de Estadistica e Investigacion Operativa* **37** (2021) 47.

[150] M.A. Tanner, *Tools for statistical inference*, Springer (2012).

[151] W.R. Gilks, S. Richardson and D. Spiegelhalter, *Markov chain Monte Carlo in practice*, Chapman and Hall/CRC (1995).

[152] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari and D.B. Rubin, *Bayesian data analysis*, Chapman and Hall/CRC (2013).

[153] S.M. Ross, *Introduction to probability models*, Academic press (2014).

[154] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, *Equation of state calculations by fast computing machines*, *The journal of chemical physics* **21** (1953) 1087.

[155] W.K. Hastings, *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, *Biometrika* **57** (1970) 97.

[156] A. Gelman, D.B. Rubin et al., *Inference from iterative simulation using multiple sequences*, *Statistical science* **7** (1992) 457.

[157] S.P. Brooks and A. Gelman, *General methods for monitoring convergence of iterative simulations*, *Journal of computational and graphical statistics* **7** (1998) 434.

[158] T. Brinckmann and J. Lesgourgues, *MontePython 3: boosted MCMC sampler and other features*, *Phys. Dark Univ.* **24** (2019) 100260 [1804.07261].

[159] B. Audren, J. Lesgourgues, K. Benabed and S. Prunet, *Conservative constraints on early cosmology with montepython*, *Journal of Cosmology and Astroparticle Physics* **2013** (2013) 001–001.

[160] A. Cuceu, J. Farr, P. Lemos and A. Font-Ribera, *Baryon Acoustic Oscillations and the Hubble Constant: Past, Present and Future*, *JCAP* **10** (2019) 044 [1906.11628].

[161] BOSS collaboration, *The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: cosmological analysis of the DR12 galaxy sample*, *Mon. Not. Roy. Astron. Soc.* **470** (2017) 2617 [1607.03155].

[162] F. Beutler, C. Blake, M. Colless, D.H. Jones, L. Staveley-Smith, L. Campbell et al., *The 6df galaxy survey: baryon acoustic oscillations and the local hubble constant*, *Monthly Notices of the Royal Astronomical Society* **416** (2011) 3017–3032.

[163] A.J. Ross, L. Samushia, C. Howlett, W.J. Percival, A. Burden and M. Manera, *The clustering of the SDSS DR7 main Galaxy sample – I. A 4 per cent distance measure at* $z = 0.15$, *Mon. Not. Roy. Astron. Soc.* **449** (2015) 835 [1409.3242].

[164] D.M. Scolnic et al., *The Complete Light-curve Sample of Spectroscopically Confirmed SNe Ia from Pan-STARRS1 and Cosmological Constraints from the Combined Pantheon Sample*, *Astrophys. J.* **859** (2018) 101 [1710.00845].

[165] A. Heavens, Y. Fantaye, A. Mootoovaloo, H. Eggers, Z. Hosenie, S. Kroon et al., *Marginal Likelihoods from Monte Carlo Markov Chains*, *arXiv e-prints* (2017) arXiv:1704.03472 [1704.03472].

[166] R. Trotta, *Bayes in the sky: Bayesian inference and model selection in cosmology*, *Contemp. Phys.* **49** (2008) 71 [0803.4089].

[167] A.R. Liddle, *Information criteria for astrophysical model selection*, *Mon. Not. Roy. Astron. Soc.* **377** (2007) L74 [astro-ph/0701113].

[168] M. Kunz, R. Trotta and D. Parkinson, *Measuring the effective complexity of cosmological models*, *Phys. Rev. D* **74** (2006) 023503 [astro-ph/0602378].

[169] A. Lewis, A. Challinor and A. Lasenby, *Efficient computation of CMB anisotropies in closed FRW models*, **538** (2000) 473 [astro-ph/9911177].

[170] https://camb.readthedocs.io/en/latest/.

[171] https://github.com/dgrin1/axionCAMB.

[172] J. Lesgourgues and S. Pastor, *Massive neutrinos and cosmology*, *Phys. Rept.* **429** (2006) 307 [astro-ph/0603494].

[173] Y.Y.Y. Wong, *Neutrino mass in cosmology: status and prospects*, *Ann. Rev. Nucl. Part. Sci.* **61** (2011) 69 [1111.1436].

[174] S. Agarwal and H.A. Feldman, *The effect of massive neutrinos on the matter power spectrum*, *mnras* **410** (2011) 1647 [1006.0689].

[175] J. Liu and M.S. Madhavacheril, *Constraining neutrino mass with the tomographic weak lensing one-point probability distribution function and power spectrum*, *Phys. Rev. D* **99** (2019) 083508 [1809.10747].

[176] KATRIN collaboration, *Improved Upper Limit on the Neutrino Mass from a Direct Kinematic Method by KATRIN*, *Phys. Rev. Lett.* **123** (2019) 221802 [1909.06048].

[177] A.E. Bayer, F. Villaescusa-Navarro, E. Massara, J. Liu, D.N. Spergel, L. Verde et al., *Detecting Neutrino Mass by Combining Matter Clustering, Halos, and Voids*, *Astrophys. J.* **919** (2021) 24 [2102.05049].

[178] KATRIN collaboration, *Direct neutrino-mass measurement with sub-electronvolt sensitivity*, *Nature Phys.* **18** (2022) 160 [2105.08533].

[179] Planck collaboration, *Planck 2018 results. V. CMB power spectra and likelihoods*, *Astron. Astrophys.* **641** (2020) A5 [1907.12875].

[180] R. Murgia, A. Merle, M. Viel, M. Totzauer and A. Schneider, *"Non-cold" dark matter at small scales: a general approach*, *JCAP* **11** (2017) 046 [1704.07838].

[181] R. Murgia, V. Iršič and M. Viel, *Novel constraints on noncold, nonthermal dark matter from Lyman- α forest data*, *Phys. Rev. D* **98** (2018) 083540 [1806.08371].

[182] M. Archidiacono, D.C. Hooper, R. Murgia, S. Bohr, J. Lesgourgues and M. Viel, *Constraining Dark Matter-Dark Radiation interactions with CMB, BAO, and Lyman-α*, *JCAP* **10** (2019) 055 [1907.01496].

[183] T. Matos and L. L-Parrilla, *The graviton Compton mass as Dark Energy*, *Rev. Mex. Fis.* **67** (2021) 040703 1.

[184] S. del Campo, J.C. Fabris, R. Herrera and W. Zimdahl, *On holographic dark-energy models*, *Phys. Rev. D* **83** (2011) 123006 [1103.3441].

[185] S. Weinberg, *Gravitation and Cosmology*, Wiley, 1 ed. (1972).

[186] A.G. Riess et al., *A 2.4% Determination of the Local Value of the Hubble Constant*, *Astrophys. J.* **826** (2016) 56 [1604.01424].

[187] https://docs.conda.io/en/latest/miniconda.html.

[188] https://docs.conda.io/en/latest/miniconda_hashes.html.

[189] https://github.com/lesgourg/class_public.

[190] https://rogerdudler.github.io/git-guide/index.html.

[191] https://brew.sh.

[192] https://github.com/brinckmann/montepython_public/tree/3.3.

[193] https://docs.docker.com/engine/install/.

[194] https://julialang.org/downloads/.

[195] https://github.com/marius311/mpk_compilation.