Conceiving the cosmic choruses – Constraints on f(R) gravity from gravitational radiation emitted by pulsar systems

Masterarbeit

zur Erlangung des akademischen Grades
Master of Science (M.Sc.) in Physik
am Institut für Theoretische Physik (ITP)
der Goethe-Universität Frankfurt am Main

vorgelegt von Yannik Schaper
Matrikelnr.: 6345947
Erstgutachterin: Prof. Dr. Laura Sagunski
Zweitgutachter: Prof. Dr. Jürgen Schaffner-Bielich
eingereicht am 23. Jul. 2024

Ehrenwörtliche Erklärung

nach §39 (15) Prüfungsordnung 2020 für den BAMA-Studiengang Physik

Hiermit erkläre ich, dass ich die Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus Veröffentlichungen oder aus anderen fremden Texten entnommen wurden, sind von mir als solche kenntlich gemacht worden. Ferner erkläre ich, dass die Arbeit nicht – auch nicht auszugsweise – für eine andere Prüfung verwendet wurde.

Eppstein, den 23. Jul. 2024

Yannik Schaper

Zusammenfassung

Die Beobachtungen von Pulsaren markierten den Beginn der Messung von Gravitationswellen. Sie sind in jüngster Zeit durch die Veröffentlichungen von Pulsar-Timing-Array-Organisationen wieder in den Fokus gerückt. Nicht nur die Transmission von Gravitationswellen, sondern insbesondere auch ihre Emission kann mit diesen Systemen untersucht werden. Mit der wachsenden Zahl sehr präziser Beobachtungen von Pulsaren in Doppelsternsystemen wurde es möglich, ihre Massen sowie die Änderung der Umlaufzeiten mit hoher Genauigkeit zu messen. Dies macht sie zu idealen astrophysikalischen Laboratorien, um die allgemeine Relativitätstheorie sowie mögliche Modifikationen dieser über die Summe der Strahlungsleistung zu überprüfen.

Dabei ist die $f(R)$-Gravitation, bei der der Ricci-Skalar durch eine beliebige Funktion dieses ersetzt wird, eine gute Möglichkeit, um nach einem breiten Spektrum von Korrekturen der allgemeinen Relativitätstheorie bei hohen Krümmungen zu suchen. Über die dynamische Äquivalenz zu der Allgemeinen Relativitätstheorie plus einem massiven Skalarfeld ermöglicht sie auch die Suche nach Dunkler Materie in Form einer großen Gruppe von Modellen leichter Skalarteilchen. Die Beobachtung der Summe der tensorartigen Gravitationswellen, welche schon in der Allgemeinen Relativitätstheorie anwesend sind, und der skalaren Strahlung, welche von diesen Modellen eingeführt wird, ermöglicht es, diese auf vielen Größenordnungen der enthaltenen Teilchenmasse einzuschränken. Die Analysetechnik stellt eine wichtige Ergänzung zu anderen Methoden dar, insbesondere am unteren Ende des möglichen Massenbereichs.

Abstract

Pulsar observations marked the beginning of gravitational wave measurements. Recently they got into the focus again with the publications from pulsar timing array organisations. Not only the transmission of gravitational waves, but especially their emission can be investigated using these systems. With the growing number of very precise observations of pulsars in binary systems, it became possible to measure their masses as well as the change of the orbital periods to high accuracy. This makes them prefect astrophysical laboratories to test general relativity as well as its possible modifications by studying the total radiated power.

Considering $f(R)$ gravity, which replaces the Ricci scalar with an arbitrary function of it, for this is a practical way to look for a broad range of high curvature corrections to general relativity. Due to the dynamical equivalence to general relativity with an additional massive scalar field, it also enables the search for dark matter in form of a large family of models of light scalar particles. The observations determining the sum of tensor radiation, also present in general relativity, and scalar radiation introduced by these extensions enables constraining them on many orders of magnitude for their masses with competitive results especially at the lower end of feasible mass interval.

Acronyms & Abbreviations

ALP	axion-like particle
CompOSE	CompStar Online Supernovae EoS
DM	dark matter
EoS	equation of state
FBS	fermion-boson star
GR	general relativity
GS	golden search
GW	gravitational wave
MSP	millisecond pulsar
NS	neutron star
pulsar	pulsating source of radio emission
PTA	pulsar timing array
ToA	time of arrival
TOV	Tolmann-Volkhoff-Oppenheimer
TT	transverse-traceless
QFT	quantum field theory
VEV	vacuum expectation value
WD	white dwarf

Contents

1 Introduction

> Pulsar: a dying star spinning under its own exploding anarchic energy, like a lighthouse on speed. A star the size of a city, a city the size of a star, whirling round and round, its death-song caught by a radio receiver, light years later, like a recorded message nobody heard, back-played now into infinity across time. Love and loss.
>
> Jeanette Winterson, The Stone Gods [1]

In this work the investigated objects are pulsars. These are strange phenomena; they are not live-bringing like a well behaved main sequence star, but the aftermath of the violent death of a star radiating its tragic story far in the universe. But on another level, they also become relatable; their stability and regularity is fascinating, it creates structure in the vast landscapes of our galaxy, which brings about a feeling of orientation. Having this in mind it is no wonder that Jeanette Winterson uses them in her post-apocalyptic love story "The Stone Gods" to describe the relation of the protagonists.

In a more pragmatic way they are also used to guide contingent intergalactic travellers the passage to earth. On the Voyager golden records this was designed in form of a map of several bright pulsars by Sagan et al. [2]. Their regular, time dependent, signal is very close to how we as humans construct our world with calendars, music etc. So it seems reasonable that, together with the decoding of the actual record, an alien species will decipher this information and recognise them as lighthouses in the night sky, too.

If one goes back in time to the early ages of radio astronomy, it is also no surprise that Jocelyn Bell-Burnell became exited when she noticed the very small but regular patterns appearing during observations in 1968. There appeared periodic spikes in the radio signal returning in a time span of the order of 1 s. This led to the name of this phenomena: *Pulsating source of radio emission (pulsar)*. The pattern becomes even more impressive, when it is seen in the optical range as it became possible with the latest generation of telescopes [3]. The aerial resolution leads to a locatable source in the sky showing its distinct pattern like it can be seen in a time sequence in figure 1.1.

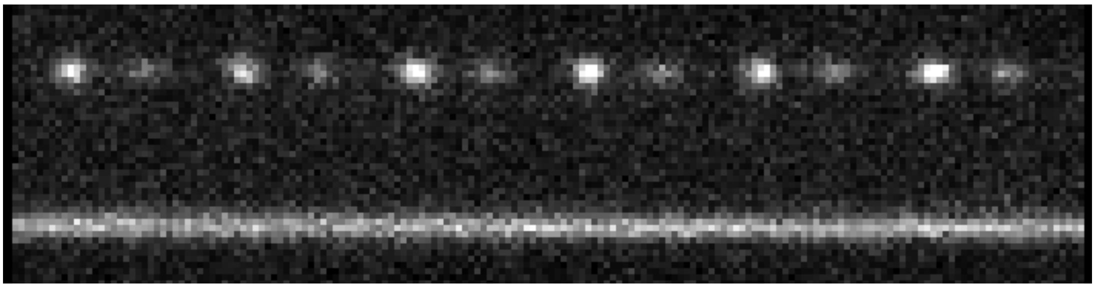
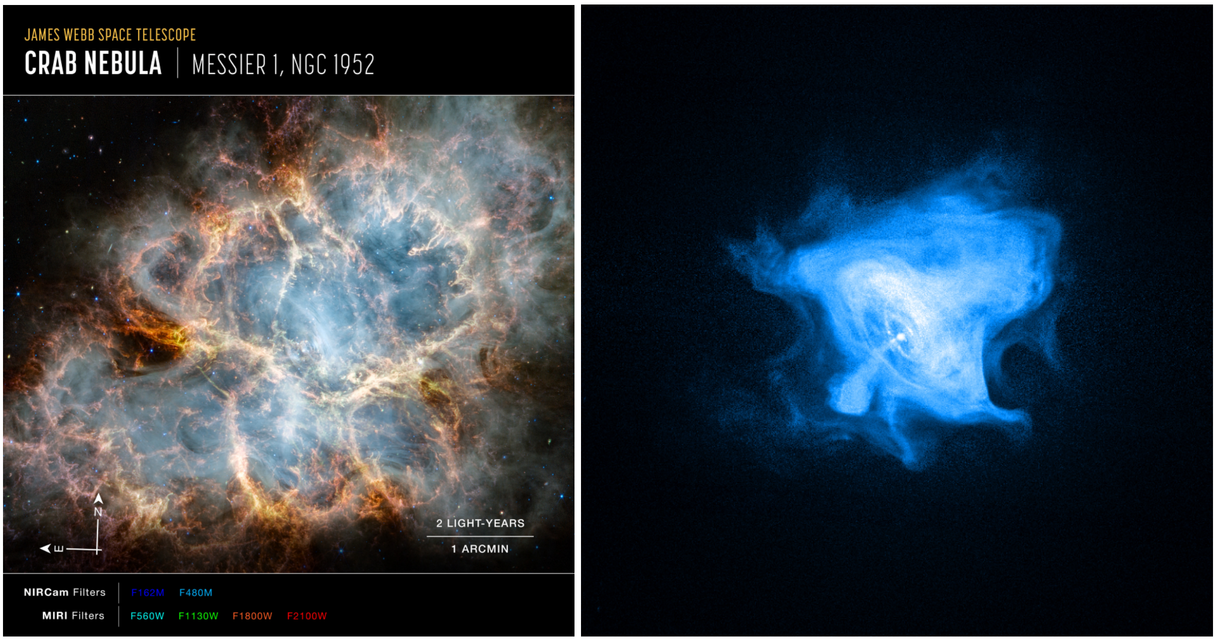


Figure 1.1: Time sequence of the Crab Pulsar and a close by non-variable star observed at the VLT (data from the release [4]).

When studying these objects further, they do not become less impressive. It was found that they consist of neutron stars, that are rapidly spinning. These are producing a lot of radiation along two light-cones due to their strong and spiralised magnetic field. This uses the energy that is left over from one of the most violent events in the universe: A supernova at the end of the lifetime of a star.

For example is the depicted object on the last page linked to the Crab Nebula. This was formed from the outer shells of a star that appeared on the sky during the Supernova event in 1054, when it could be observed with the naked eye by astronomers in China and Japan. It was captured in its form today by the James Webb Space Telescope, which is shown in figure 1.2a together with an x-ray image of the contained pulsar in figure 1.2b.



(a) The Crab Nebula (Messier 1) as depicted by JWST (reprinted from [5]).

(b) The x-ray image constructed from Chandra data of PSR J0534+2200 (reprinted from [6]).

Figure 1.2: Pictures of the Crab Nebula and its associated pulsar in different electromagnetic bands.

The possibility of precise measurements of the signal, especially the times of arrival of the pulses and the interesting environment with very high energy densities makes pulsars perfect astrophysical laboratories. Here one key observable is the emission gravitational waves, that is enabled due to the heavy masses and dynamic system. Together with systems containing solar mass black holes, binary neutron stars are the only ones for which their gravitational wave emission has been directly detected by the LIGO observatories [7].

But even before the last stages of an inspiral that can be observed with these, gravitational waves are radiated away in a relatively high intensity. They are too weak to be detected directly. However, they can be traced due to the energy they carry away from the system. The first indirect proof for the existence of gravitational wave was achieved using radio observations of a pulsar. Only seven years after the first pulsar detections, Hulse and Taylor [8] were measuring the change of the orbit of a binary pulsar, which imprinted its signature in the delay of the times of arrival of the pulses.

If one studies the effect in these high curvature systems, where also other relativistic effects like the periastron shift are very distinct, it is likely that a deviation in the behaviour of gravity from the theory of general relativity will be noticeable. This is why pulsars are very interesting objects to study, if one is interested in theories of modified gravity, for which many different models are feasible. One common approach is to extend the influence of the spacetime curvature being described with the Ricci scalar R by using a function $f(R)$ instead. This function can in principle be arbitrary. But because it has to describe all the systems and phenomena as well as the very precise and verified predictions of general relativity, only small deviations at higher orders in R are possible.

They are still interesting, because for example they could capture quantum gravity effects in form of an effective theory.Exhibiting an equivalence to models of dark matter brings these theories in the focus of the community which is studying possible dark matter accumulations around neutron stars, too. In these particular models the candidates are scalar particles that dynamically influence the ordinary matter in the equivalent way to a correction of general relativity a high curvature. To study these neutron stars are especially interesting, since they are very compact and, in contradiction to black holes, can carry a scalar charge induced by the contained dark matter.

This enables a secondary interaction between two neutron stars, which introduces a new form of radiation in addition to gravitational waves. The deviation in the waveform for the so-called *scalar-tensor theories* can be seen in detail in [9]. But the important point for the pulsar observations is that more energy can be radiated away, if these theories are correct. This would be a smoking gun for the existence of dark matter halos around neutron stars or a necessary new theory of gravity at high curvatures.

In the search for suitable candidates, a good way is to compare the predictions of the modification to the increasing number of very precise pulsar measurements, that have been made in recent years [10]. This way one can constrain the allowed range of parameters in the function $f(R)$. Because of the above described two kinds of radiation, their dependency on the selected model respectively mass range is not trivial.

What also comes into play here, is that for the scalar radiation the compactness of the neutron star becomes important. To calculate this, one has to consider its inner microphysics, that is described by a for neutron stars yet unknown equation of state. The calculation of the constraints that emerge from these dependencies has to be done numerically, which is the goal of this work and the corresponding created code bases.

2 Pulsars

Looking at the reoccurring spikes in the radio signal with a very regular period about 1.337 s in figure 2.1, the name of the class of objects first found in observations by Hewisch et al. [11] becomes very clear. The so called *pulsating sources of radio emission (pulsars)* as phenomena of radio observations were thought of as too rapid to be linked to oscillations of regular variable stars. That lead to neutron stars (NSs) or white dwarfs (WDs) being preferred as a possible explanation. Calculations showed that they can oscillate with high frequencies of up to the order of 10 ms [12]. The width of the peaks is often even smaller than 10 ms corresponding to a causally connected area over the emission time with a diameter of below 3000 km. This matter of fact is also pointing towards these compact objects.

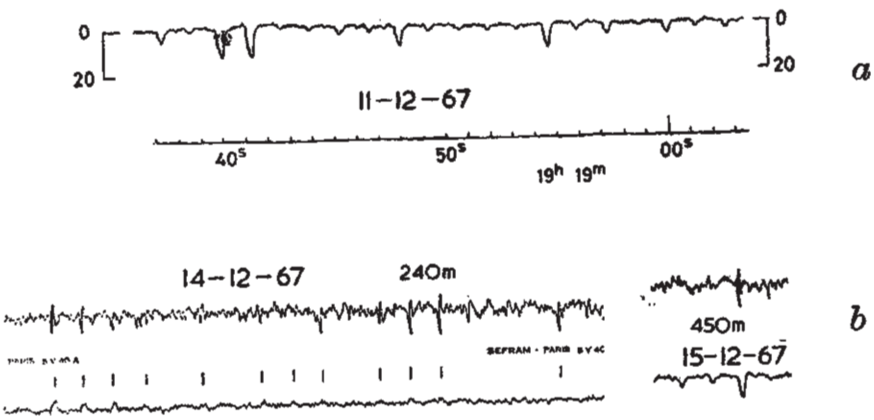


Figure 2.1: Pulsar signal from the observations at the Mullard Radio Astronomy Observatory with the 4 m-band interferometer by Hewish and Bell-Burnell (modified reprint from [11]).

Already in the same year, the identification with rapidly rotating NSs became clear due to works from Gold [13] and Pacini [14]: Taking a compact object with the mass M and radius R, the maximum angular velocity Ω_{max} is determined by the centrifugal force at the equator. This has to be balanced by gravity resulting in

$$\Omega_{max}^2 R = G \frac{M}{R^2} . \tag{2.1}$$

Assuming a constant density $\rho := M/R^3$ the minimal rotational period P_{min} is then described by

$$P_{min} = \frac{2\pi}{\Omega_{max}} = \sqrt{\frac{2\pi}{G\,\rho}} . \tag{2.2}$$

This result is leading to a density, which has to be grater than 10^{11} kg/m^3 [15] and because an actual physical object will become oblate in response to the rotation, the true minimal density will even exceed this estimate.

In order to generate a pulsed signal, these systems have to emit a beam of radiation which is swept across the observer. Measurements of the polarisation of the radio waves in the following year by Radhakrishnan and Cooke [16] supported that kind of behaviour similar to a *lighthouse*. They suggest that the emission comes from a co-rotating magnetosphere leading to magnetic dipole radiation, which is forming the beam of radio waves.

In addition to that, it causes a reduction of the rotational energy. Together with the large angular momentum this gives a good explanation for the very stable period that is slowly becoming larger. This is the reason why only young pulsars like PSR B0531+21 (formed in 1054 at the supernova event that also formed the Crab Nebula) [17] can have short periods around 100 ms and will spin down during their lifetime until the radio emission ends at a period up to 10 s [18].

Contradicting this idea, PSR B1937+21 detected by Backer et al. [19] in 1982 has only a period of 1.56 ms. The value is still above the limit derived in equation (2.2) for NSs, but it it much higher than expected from having only the angular momentum after the core-collapse process. Hence, this pulsar has been the first detection of the class of *millisecond pulsars (MSPs)*.

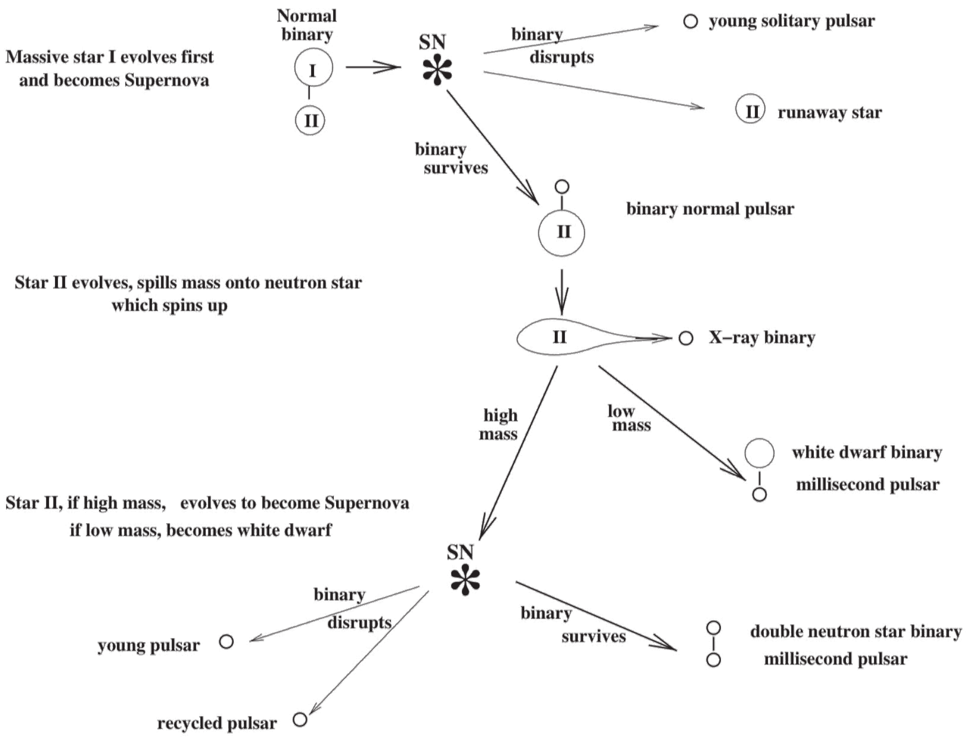


Figure 2.2: Evolution pathways of a binary system containing a MSP (reprint from [20]).

The fact that these MSPs are almost always found in close binary systems gives an important hint on where the additional angular momentum comes from: If the companion object is a star at the end of its lifetime, the expansion of the outer atmosphere will lead to an accretion onto the pulsar. In this process a part of the angular momentum is transferred onto the pulsar resulting in a rapid spin-up of it [21]. An overview of the possible development paths for these systems is depicted in figure 2.2.

2.1 Compact objects as pulsar constituents

Before the pulsar signal to be observed as well as its analysis is discussed further explaining their usage as astrophysical laboratories for gravity, the compact objects contained in them are in the focus of the following sections. The central part for this will be the derivation of their equation of state (EoS) determining the effect of their composition on the mass distribution.

2.1.1 White Dwarfs

The first kind of compact objects, *white dwarfs (WDs)* play an important role as companions of pulsars and by this become relevant for this work. Despite recent foundings where they can also act as emitters of radio waves resulting in an object very similar to regular pulsar only with periods around 1 min [22], here only the case will be discussed where WDs are present as companion objects.

They were the first compact objects to be described theoretical as well as being observed. Being the remnant of stars, which is left after its outer shells have been ejected when the energy production from nuclear fusion has ended, they can not be stabilised by the radiation pressure from the resulting radiation like regular stars. The only outward force comes from the degeneracy pressure of the densely packed electron gas in the plasma.

It is determined by the so called Fermi energy ϵ_F, which describes the thermal energy electrons have to obtain to occupy a higher state than the lowest possible due to the Pauli exclusion principle. This can be calculated via their de Broglie wavelength like shown in [23] with the electron mass m_e and their number density n_e as

$$\epsilon_F = \frac{\hbar^2}{2m_e}\left(3\pi^2\, n_e\right)^{2/3} . \tag{2.3}$$

Assuming full ionisation, this is the same as the proton number, which can be expressed by the nuclear number density $n_c := \rho/m_h$ and the fraction of protons in the nucleus described by the factor of the ordinal and mass number of $\mu := Z/A$. From this the equation can be rewritten as

$$\epsilon_F = \frac{\hbar^2}{2m_e}\left(3\pi^2\,\frac{\mu\,\rho}{m_h}\right)^{2/3} . \tag{2.4}$$

As long as the electrons are non-relativistic and their speed is given by $v = p/m_e$ the corresponding momentum p is

$$p = \hbar\left(3\pi^2\, n_e\right)^{1/3} = \hbar\left(3\pi^2\,\frac{\mu\,\rho}{m_h}\right)^{1/3} , \tag{2.5}$$

resulting in a pressure P generated by the degeneracy of

$$P = \frac{1}{3}\, n_e\, p\, v = \frac{\left(3\pi^2\right)^{2/3}}{5}\,\frac{\hbar^2}{2m_e}\left(\frac{\mu\,\rho}{m_h}\right)^{5/3} . \tag{2.6}$$

This is a simplified version of the EoS of a WD, in which all electrons would have the same momentum. It has the polytropic form of $P \propto \rho^{5/3}$ and if one inserts this into the *Tolmann-Volkhoff-Oppenheimer (TOV) equation* [24], which describes the equilibrium of the pressure in a spherical symmetric star, one obtains a solution for a stable object.

Because pressure and temperature will increase with the central density, there exists a point at which the electrons will become relativistic and their velocity will instead be described by $v \approx c$ resulting in a modified EoS:

$$P = \frac{\left(3\pi^2\right)^{1/3}}{4} \hbar\, c \left(\frac{\mu\, \rho}{m_h}\right)^{4/3} . \tag{2.7}$$

Compared to equation (2.6) there is a reduction of the polytropic index to 4/3 resulting in lower central pressures and by this instability. The dynamic caused here is a collapse of the core leading to a supernova. Knowing this no WDs can exist over a certain mass linked to the central pressure, which is called the Chandrasekhar limit with the maximal mass M_{Ch}. Its value can be described by

$$M_{Ch} = \frac{\sqrt{3}}{8\pi} M_{Pl}^3 \left(\frac{\mu}{m_H}\right)^2 {}^1, \tag{2.8}$$

which is $1.46\, M_\odot$, if the WD is dominated by ${}^4_2$He resulting in $\mu \approx 1/2$.

The derivation described up until here was also done by Nauenberg [25], where he used a more realistic velocity distribution via using the internal energy of an electron gas. Even without additional corrections this leads to the quite accurate mass-radius relation

$$R = \frac{0.0225}{\mu} R_\odot \sqrt{\left(\frac{M}{M_{Ch}}\right)^{-2/3} - \left(\frac{M}{M_{Ch}}\right)^{2/3}} \approx 3450\,\mathrm{km} * \sqrt{1.65 \left(\frac{M}{M_\odot}\right)^{-2/3} - \left(\frac{M}{M_\odot}\right)^{2/3}} . \tag{2.9}$$

Here for the approximation on the right-hand side a He-WD is assumed. The resulting curves for different compositions are depicted in figure 2.3. For these the effects of β-decay inside the WD and further corrections due to the more complex inner structure depending on the stars history are being neglected. But WDs in pulsars in almost all cases can not be observed directly and by that the Helium domination is often the only information over their composition that can be reconstructed. Hence, it would not be reasonable to use these more complex models.

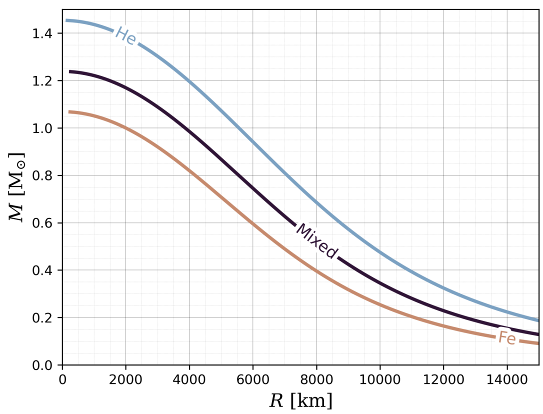


Figure 2.3: Mass-radius relation of a no-β-decay WD with no electrostatic corrections dominated by ${}^4_2$He, ${}^{56}_{26}$Fe and for an equal mixture of these.

[1] $M_{Pl} := \sqrt{\frac{\hbar c}{4\pi G}}$ is the Planck mass coming from the unit of mass in natural or Planck units.

2.1.2 Neutron Stars

If a WD obtains a mass above the Chandrasekhar limit discussed in the previous section, the remnant of the resulting supernova is called a *neutron star (NS)* [26]. Inside its core the matter changes from a dense plasma of regular matter to a free gas of only very closely packed neutrons [23]. This uses less phase space as well as physical one compared to regular matter leading to an even more compact object. Because of the electric neutrality of the neutrons it again can only be stabilised by the degeneracy pressure resulting in central densities of the order of 10^{18} kg/m^3 similar to an atomic nucleus.

To describe their intrinsic nature more accurately an EoS is needed relating the composition of the matter to the energy density at the different distances to the centre of the NS. Even though the form of this is still unknown, typical features also depicted in figure 2.4 exists, such as an outer crust of heavy, neutron-rich nuclei and relativistic electrons, an inner crust, where a superfluid of free neutrons are present as an additional component, and the interior, where besides the super-fluid of neutrons only a small number of protons and electrons. There also may be a solid core of other sub-nuclear particles, but this as well as other details are sensitive to the micro-physics and total mass of the NS.

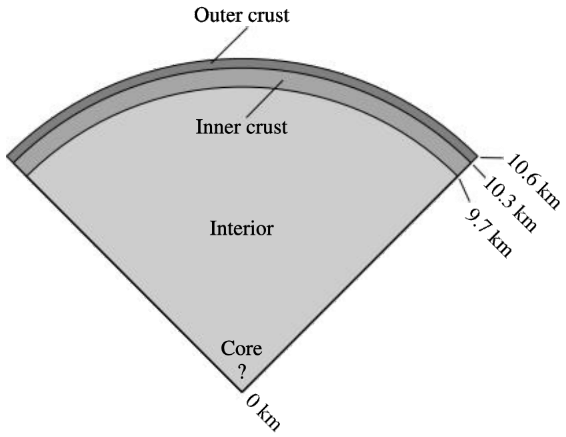


Figure 2.4: Model for a typical NS with a mass of 1.4 $M_{\odot}$(reprinted from [23]).

Another thing which is clear independent of the actual model, is the existence of a maximum mass where the pressure at the centre will not be enough to stabilise the system and the NS will collapse into a black hole. The limit will show up in the mass-radius relation as a point, where the mass is not growing anymore and its derivative with respect to the radius changes its sign.

For the emission of the pulsar signal the detailed structure of the interior does not play an important role, but what will be important is the mass and the radius of the NS. There are suggestions for universal relations, which will capture the dependencies very well, at least at the maximum mass [27]. But for describing NSs at masses down to 1 $M_{\odot}$ as well the EoS is determining the mass-radius relation [28]. Some examples for these are depicted in figure 2.5 showing the broad range of possible radii.

To capture this, several EoSs are used trying to sample the whole range of soft to stiff realistic ones based on possible EoSs in the analysis of the NS events done by the LIGO collaboration [30], the later discussed modification of gravity [31] and a general review about realistic EoSs [29]. The ones selected have to allow for high enough maximum mass like shown in [32] for their consideration in

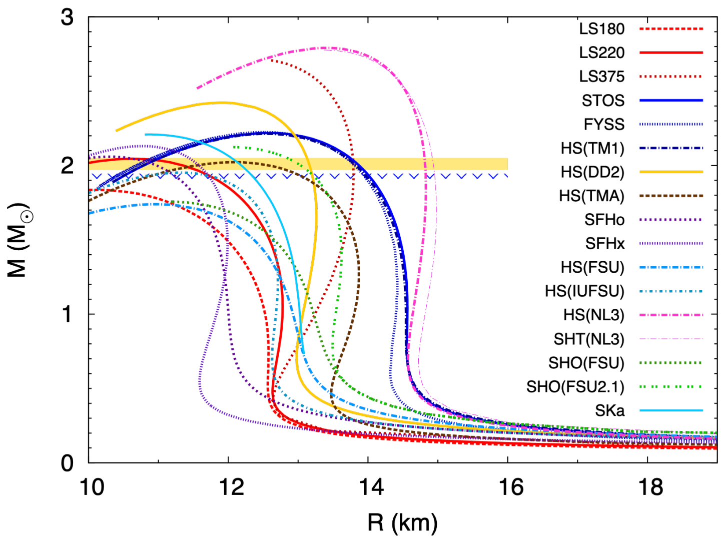


Figure 2.5: Mass-radius relation of spherically symmetric NSs for a selection off EoSs, where the two horizontal bars indicate the mass measurements of PSR J1614-2230 and J0348+0432 (reprinted from [29]).

describing pulsars.

These are taken from the CompStar Online Supernovae EoS (CompOSE) service [33, 34] and listed in table 2.1. Together with the provided code [35] it is possible to generate tables for them containing the energy density and pressure depending on the number density of neutron. The application of these as input for the actual calculations will be later described in section 4.1.

Name	CompOSE index	Category	Reference papers
SFH(SFHo)	14	hadronic	[36–38]
HS(DD2)	18	hadronic	[36–40]
APR(APR)	68	hadronic	[41–43]
RG(SLY9)	86	nuclear	[44–46]
RG(Skl3)	88	nuclear	[44, 45, 47]
GMSR(FSU2H)	213	inner crust-core unified	[48–51]
GPPVA(FSU2)	215	inner crust-core unified	[48–50, 52]
GPPVA(DD2)	217	inner crust-core unified	[40, 48, 49]
GPPVA(DDME2)	218	inner crust-core unified	[48, 49, 53]
GPPVA(TW)	219	inner crust-core	[48, 49, 54]
GMSR(NL3wrL55)	220	inner crust-core unified	[48, 49, 55–57]
GMSR(H4)	231	nucleonic / unified	[58]

Table 2.1: EoSs used in the description of NSs taken from the CompOSE service [33].

2.2 Pulsar timing

For the usage of pulsars as astrophysical laboratories the very consistent pulse periods are an obvious choice, because their *times of arrival (ToAs)* are relatively simple to study. For this it is necessary to isolate the true pulse from influences like the movement of the earth or the proper motion of the pulsar itself, but also gravitational and other effects influencing the transmission of the signal on its path through the galaxy.

The first one is well studied by observations inside the solar system. From the Jupiter satellite eclipse observations by Rømer [59] the delay t_R is directly given by the speed of light together with the distance to the sun $d_\oplus$ and the angular velocity of the earth ω as

$$t_R = c\, d_\oplus \cos(\omega t - \lambda) \cos\beta \,, \tag{2.10}$$

where λ is the ecliptic longitude and β the latitude of the observed object. This leads to a sinusoidal variation of the pulse's ToAs over the year [60]. In analogy to this, a correction due to the rotation of the Earth and by this observational position of the order of 21 ms is necessary.

An effect of the theory of general relativity (GR) also leads to a delay, the so called Shapiro delay Δ_S. It is caused by the passage of the radiation though the solar system with its curved spacetime. Using the angle pulsar-Sun-Earth θ this can be expressed as

$$\Delta_S = \frac{2G\,M_\odot}{c^3} \ln(1 + \cos\theta) \,, \tag{2.11}$$

which has a maximum when the pulsar appears close to the Sun with around 120 ms. There can also be an increase of this delay by other objects, if they are close to the path of the radiation. The most prominent is here the delay due to a companion object, if observing along the ecliptic plane of the system. Because their masses appear in the formula this allows for their measurement, which makes these binaries very interesting systems to study their dynamics.

In addition, if one compares the the time of a clock on earth t_E with the coordinate time at infinite distance to the Sun t, there is an effect given by

$$\frac{\mathrm{d}t}{\mathrm{d}t_E} = 1 + \frac{2GM_\odot}{c^2}\left(\frac{1}{r} - \frac{1}{4a}\right) . \tag{2.12}$$

Here the deviation of the distance to earth from the semi-mayor axis a plays a key roll. Assuming a Keplerian orbit this is a function of the true anomaly f, which is the angle between the Earth at perihelion and its instantaneous position [61]. Given the eccentricity e, it can be expressed as

$$\Delta t_E = 1.661\,45\,\mathrm{ms} * \left(\left(1 - \frac{e^2}{8}\right)\sin(f) + \frac{e^2}{2}\sin(2f) + \frac{3e^2}{8}\sin(3f)\right) . \tag{2.13}$$

With all the corrections described above the ToAs can be reduced to the ToA observed at a static position at infinity in an otherwise empty universe.The only effects left are on the emitter side. Here for example the proper motion of the system will lead to a linear growing deviation from the expected ToAs. Although, this is very small in comparison to the period timescale and will only be detected in observations over several years.

Another effect and the last one large enough to be mentioned here is the change of the observed derivative of the period due to a gravitational acceleration of the pulsar caused by its surroundings. This can often be seen for pulsars in globular clusters, where an acceleration towards the observer for systems at the far side of the cluster will lead to an artificial slowdown of the pulses and vice versa.

Considering all these effects, one now can model the intrinsic behaviour of the pulsar. This is typically done by expressing the expected pulse number N as a Taylor series of the pulsation frequency or as it will be done here angular velocity $\Omega := 2\pi/P$ over the observation time t_O:

$$N = \Omega\, t_O + \frac{\dot{\Omega}}{2}\, t_O^2 + \frac{\ddot{\Omega}}{6}\, t_O^3 + \mathcal{O}\left(t_O^4\right) . \tag{2.14}$$

From this model the evolution of the angular velocity is determined. The most important parameter that can be derived of it is the braking index $n := \Omega\ddot{\Omega}/\dot{\Omega}^2$ [62], that can also theoretically be computed from the slow down due to emission of angular momentum. It is described by the time derivative of the radiated energy $\dot{E}_M$ as a function of the magnetic field strength on the surface B and the angle between the dipole and the rotational axis α like

$$\dot{E}_M = - \frac{2}{3c^3}\, \Omega^4\, R^6\, \sin^2\alpha \left(\frac{B}{\mu_0}\right)^2 . \tag{2.15}$$

Using the definition for the rotational energy $E_R := 1/2\, I\, \Omega^2$ and assuming a constant moment of inertia I as well as inclination α and strength of the magnetic field, there is the relation for the change of the angular velocity of $\dot{\Omega} \propto \Omega^3$. Differentiating this and inserting it into the definition of the braking index the proportionality constant depending on the quantities mentioned above will drop out leading to the simple result $n = 3$.

The described model can be used for the actual observations, folding the signal according to the difference of concurrent ToAs to separate the signal of an individual pulsar from other sources and the detector noise. However, it also allows to look for very small deviations from effects not captured above.

2.2.1 Timing of binary systems

In the previous section one possible deviation from the modelled signal was left out. As mentioned in the beginning of this chapter a significant amount of pulsars are part of a binary system. For these additional delays similar to equation (2.10) from the change of emission position and equation (2.11) from the gravitational effect of the companion object are present [63]. This is shown for three observations near conjunction of the binary in figure 2.6, where the difference of the epochs comes from the relativistic precession of the orbital ellipse.

Additionally, there is an effect called the Einstein delay ΔE, which comes from relativistic modifications of the orbital phase θ (deviating from Keplers second law) given by

$$\Delta_E = \frac{G}{c^2}\frac{\Omega\, e}{a}\, \mu\left(1 + \frac{2}{q}\right) \sin\theta\, , \tag{2.16}$$

where besides the reduced mass μ the mass ratio $q := m_p/m_c$ [2] plays a role as well.

[2] All quantities with an index p belong to the pulsar and with c to its companion.

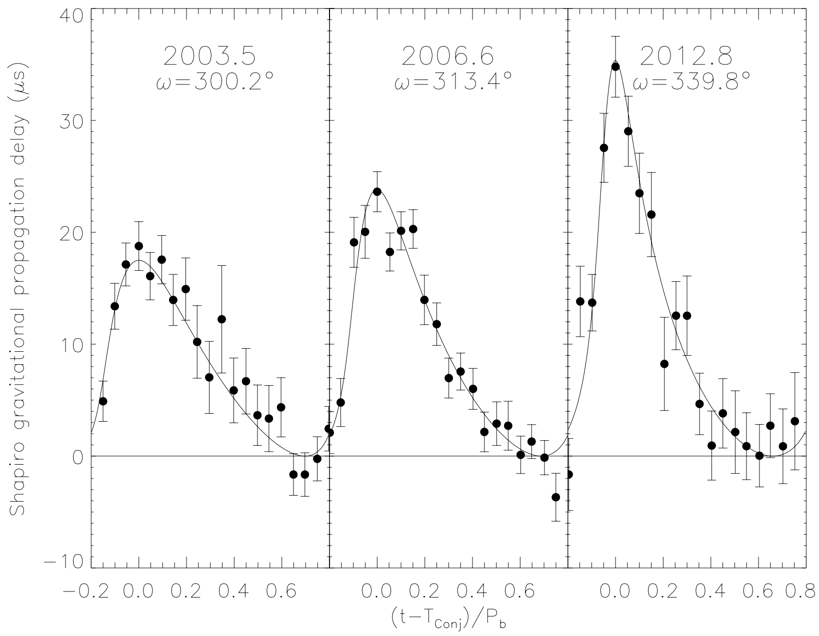


Figure 2.6: The variation of the Shapiro delay at three epochs, where the time is given as fraction of the period P_b centred at the pulsar's superior conjunction time T_{Conj}, for which the longitude of periastron ω is given as well (reprinted from [64]).

From these effects the semi-major axis, inclination, eccentricity of the orbit and even their masses can be computed. By knowing these orbit parameters one is able to test GR, for example by observing the periastron advance in analogy to the central prediction for the Mercury orbit used to verify GR, which is also shown in figure 2.7. Even before this observation was done by Weisberg and Huang [64], the change of the orbital period due to emission of gravitational waves (GWs) was observed for the Hulse-Taylor-Binary-pulsar [8] as shown in figure 2.8.

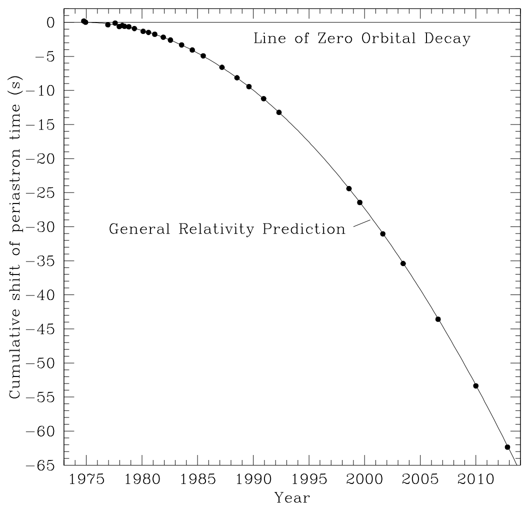


Figure 2.7: Periastron shift for the Hulse-Taylor pulsar and its GR prediction (reprinted from [64]).

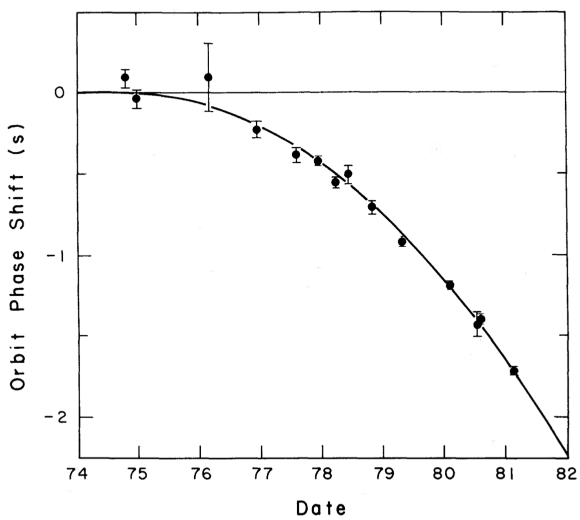


Figure 2.8: Orbit phase shift due to the emission of GW for the Hulse-Taylor pulsar (reprinted from [65]).

2.3 Measurements of pulsar masses and orbital periods

The above described observations have been done for more and more systems in recent years. A list with the observed pulsar and all computed quantities, methods and references is provided by Freire [10]. It was used to expand the quite limited sample in the paper by [66] fundamental to this work to enable the search for dependencies on the companion type, eccentricity and the stage in their development. All pulsars that are used can be found in table 2.2 and a complete documentation of their data is given in the section about the implementation in code in table 4.1.

Name	Companion	MSP	Eccentricity	High mass	Reference
PSR B1534+12	NS		high		[67]
PSR B1913+16	NS		high		[64]
PSR J0337+1715	WD	✓			[68]
PSR J0348+0432	WD		low	✓	[69]
PSR J0437-4715	WD	✓	low		[70]
PSR J0453+1559	NS	✓	high		[71]
PSR J0740+6620	WD	✓	low	✓	[72]
PSR J1141-6545	WD		high		[73]
PSR J1518+4904	NS	✓	high		[74]
PSR J1738+0333.	WD		low		[75]
PSR J1756-2251	NS		high		[76]
PSR J1757-1854	NS		high	✓	[77]
PSR J2222-0137	WD			✓	[78]

Table 2.2: All selected pulsars with their affiliations to different subgroups from interesting key characteristics [10].

3 Gravitational waves

In the previous chapter GR already appeared in form of effects on the orbits and radiation transmission. All this as well as gravity as a fundamental force with all its phenomena can be derived from its central part, the Einstein equation for the gravitational field:

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu} \ , \tag{3.1}$$

which relates the curvature and by this the geometry of the universe to its content represented by the energy-momentum tensor $T_{\mu\nu}$ [79]. This is done via the metric $g_{\mu\nu}$ of the spacetime contained in the Einstein tensor

$$G_{\mu\nu} := \overline{R}_{\mu\nu} = R_{\mu\nu} - {}^1\!/_2\, g_{\mu\nu}\, R \ {}^{1,2}, \tag{3.2}$$

where the Ricci tensor is a contraction of the Riemann tensor $R_{\mu\nu} := R^{\rho}{}_{\mu\rho\nu}$, which again can be written as a function of only the metric like

$$R^{\mu}{}_{\nu\rho\sigma} = -{}^1\!/_2 \left(\partial_\nu \partial_\rho g^{\mu}{}_{\sigma} + \partial^\mu \partial_\sigma g_{\nu\rho} - \partial_\nu \partial_\sigma g^{\mu}{}_{\rho} - \partial^\mu \partial_\rho g_{\nu\sigma} \right) . \tag{3.3}$$

3.1 Gravitational waves in the theory of general relativity

Considering the field equations one has a set of second order differential equations. But it is not obvious in this form that a solution for radiation similar to the wave equation in electromagnetism ($\Box A^\mu = 0$) will exist. Looking for wavelike solutions Maggiore [80] suggests considering spacetimes with small curvature. These are represented by a line element which consists of a flat spacetime $\eta_{\mu\nu}$ plus a small deviation $h_{\mu\nu}$ having the form

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} + \mathcal{O}\left((h_{\mu\nu})^2\right) \ , \ \left|h_{\mu\nu}\right| \ll 1 \ . \tag{3.4}$$

Using this in the Einstein tensor all terms higher than linear order in $h_{\mu\nu}$ will also vanish in the derivatives of the metric. By this, it simplifies to:

$$G_{\mu\nu} = \Box \overline{h}_{\mu\nu} - \partial_\mu \partial_\rho \overline{h}^{\rho}{}_{\nu} + {}^1\!/_2\, \eta_{\mu\nu} \partial_\rho \partial_\sigma \overline{h}^{\rho\sigma} \ . \tag{3.5}$$

The first term can be identified as the searched for d'Alembert operator acting on the trace-free perturbation $\overline{h}_{\mu\nu}$. This is also the only non-vanishing term, if we use the gauge freedom inherent in GR and the so called Lorenz gauge ($\partial_\rho h^{\rho}{}_{\nu} = 0$). The other terms are zero in this scheme and with the additional condition of $h_{\mu\nu}$ being traceless, $G_{\mu\nu}$ reduces to the right-hand-side of an ordinary wave equation.

[1] The "bar-operator" is used as a shorthand notation for trace-free tensors defined as $\overline{x}_{\mu\nu} := x_{\mu\nu} - {}^1\!/_2\, g_{\mu\nu}\, x^{\rho}{}_{\rho}$.
[2] To denote the trace of a tensor in a compact way the indices are left out like $x := x^{\mu}{}_{\mu}$.

As a third condition the energy momentum tensor is also set to zero, which fits well to our assumption of a (besides GW contributions) flat spacetime, at least outside the source. The three conditions together define the so called transverse-traceless (TT) gauge, for which the wave equation has the clear form

$$\Box h_{\mu\nu}^{\mathrm{TT}} = 0\,. \tag{3.6}$$

3.2 Creation of gravitational waves in compact binaries

For the existence of GWs a possibility of their creation is also necessary. An obvious candidate is here a binary system of compact stars leading to a strong and dynamically changing spacetime curvature.

To describe this system the action for the graviton field can be constructed using the Feynman rules resulting from the treatment of linearised gravity [81]. The starting point for this is the Einstein-Hilbert action [82] that is also used to derived the field equation given in equation (3.1). Together with the Lagrangian for the matter content of the universe $\mathcal{L}_M$ it has the form:

$$S = \int \sqrt{-g}\, \mathrm{d}^4x \left(-\frac{c^4}{16\pi G} R + \mathcal{L}_M \left(g_{\mu\nu}, \Psi \right) \right) {}^{3}. \tag{3.7}$$

Using these and the universal coupling with matter $\kappa := \sqrt{8\pi\, G/c^4}$ [83] the Ricci tensor and scalar can be expressed based on the split in equation (3.4) as:

$$R_{\mu\nu} = 1/2\, \kappa \left(\partial_\rho \partial_{[\mu} h^\rho{}_{\nu]} - \partial_\mu \partial_\nu h - \Box h_{\mu\nu} \right) + \mathcal{O}\left(h^2\right)\,, \tag{3.8}$$

$$R = \kappa \left(\partial_\mu \partial_\nu h^{\mu\nu} - \Box h \right) + \mathcal{O}\left(h^2\right)\,. \tag{3.9}$$

From this the effective action can be written in analogy to equation (3.7) like

$$S = \int \mathrm{d}^4x\, 1/2 \left[h^{\mu\nu} \Box h_{\mu\nu} - h \Box h - h^{\mu\nu} \partial_\mu \partial_\rho h^\rho{}_\nu + h \partial_\mu \partial_\nu h^{\mu\nu} + h^{\mu\nu} \partial_\mu \partial_\nu h + \kappa\, h^{\mu\nu}\, \overline{T}_{\mu\nu} \right]\,, \tag{3.10}$$

where the interaction term describes the graviton emission from a classical source for which the Feynman diagram defined like in quantum field theory (QFT) is depicted in figure 3.1.

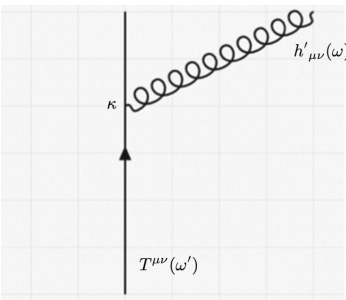


Figure 3.1: Vertex $\kappa\, h_{\mu\nu}^{\mathrm{TT}}\, \overline{T}_{\mu\nu}$ for the graviton emission from a classical source (reprinted from [81]).

[3] For the determinant of the metric here the shorthand notation $g := \det\left(g_{\mu\nu}\right)$ is used.

Using this action, the rate of graviton emission $d\Gamma^h$ can then be given by a sum containing the polarisation tensor $\epsilon^{\mu\nu}_{(\lambda)}$ to be integrated over the wave number space d^3k [66] as

$$d\Gamma^h = \frac{\kappa^2}{4} \sum_{\lambda=1}^{2} \left| \overline{T}_{\mu\nu}(k')\, \epsilon^{\mu\nu}_{(\lambda)}(k) \right|^2 \delta\left(\omega - \omega'\right) \frac{\pi}{\omega} \frac{\mathrm{d}^3 k}{(2\pi)^3} \,. \tag{3.11}$$

Because of the relation for the sum of all polarisation modes [84]

$$\sum_{\lambda=1}^{2} \epsilon^{(\lambda)}_{\mu\nu}(k)\, \epsilon^{(\lambda)}_{\rho\sigma}(k) = 1/2 \left(\eta_{\mu\rho}\eta_{\nu\sigma} + \eta_{\mu\sigma}\eta_{\nu\rho} - \eta_{\mu\nu}\eta_{\rho\sigma}\right) \tag{3.12}$$

this can be expressed as

$$d\Gamma^h = \frac{\kappa^2}{20\pi} \left(T_{ij}(\omega')\, T^*_{ji}(\omega') - 1/3 \left| T(\omega') \right|^2 \right) \delta\left(\omega - \omega'\right) \omega\, \delta\omega \,, \tag{3.13}$$

which as a last step leads to the energy loss

$$\dot{E}^h = \frac{\kappa^2}{20\pi} \int \left(T_{ij}(\omega')\, T^*_{ji}(\omega') - 1/3 \left| T(\omega') \right|^2 \right) \delta\left(\omega - \omega'\right) \omega^2 \, \mathrm{d}\omega \,. \tag{3.14}$$

For the calculation of the energy-momentum tensor in this equation the binary system is described as a classical current determined by its dynamics being described by Kepler orbits of the shape

$$r(\theta) = \frac{a\left(1 - e^2\right)}{1 + e\cos(\theta)} \,, \tag{3.15}$$

in which a, e are again part of the orbital elements introduced in section 2.2.1.

From writing the orbit and its velocity Ω (given by Kepler's third law $\Omega = \sqrt{GM/a^3}$ [85]) in a parametric form as

$$x = a\left(\cos\theta - e\right), \quad y = a\sqrt{1-e^2}\,\sin\theta, \quad \Omega t = \theta - e\,\sin\theta \,, \tag{3.16}$$

the velocity components can be written over their Fourier transforms resulting in

$$\begin{aligned} \dot{x}_n &= \frac{\Omega}{2\pi} \int_0^{\Omega/2\pi} \exp\left(i\, n\, \Omega\, t\right) \dot{x}\, \mathrm{d}t = -\, i\, a\, \Omega\, J'_n(n\, e) \,, \\ \dot{y}_n &= \frac{\Omega}{2\pi} \int_0^{\Omega/2\pi} \exp\left(i\, n\, \Omega\, t\right) \dot{y}\, \mathrm{d}t = a\, \frac{\sqrt{1-e^2}}{e}\, \Omega\, J_n(n\, e) \,, \end{aligned} \tag{3.17}$$

containing the Bessel function of first kind $J_n(x)$ [86] via their identity

$$J_n(x) = \frac{1}{2\pi} \int_0^{2\pi} \exp\left(i\left(n\,\theta - x\,\sin\theta\right)\right) \mathrm{d}\theta \,. \tag{3.18}$$

Using these to construct the energy-momentum tensor, which for a system orbiting in the x-y plane is simply given by the reduced mass μ and four velocity $U_\mu := (1,\ \dot{x},\ \dot{y},\ 0)$ [87] as

$$T_{ij}(\omega')T^*_{ji}(\omega') - {}^1\!/\!_3 \left|T(\omega')\right|^2 = \mu\ \delta\left(\vec{x} - \vec{x}'\right)\ U_\mu U_\nu\ , \tag{3.19}$$

it can be expressed depending on the parameter $\omega := n\,\Omega$ in the simple shape of a sum of Bessel functions with different orders:

$$\begin{aligned} \overline{T}_{\mu\nu}(\omega') &= \mu^2\,\omega^4\,a^4\,b(n,e)\ , \\ b(n,e) := \frac{1}{8n^2}\Bigg[&\Big(J_{n-2}(n\,e) - 2e\,J_{n-1}(n\,e) + {}^2\!/\!_n\,J_n(n\,e) + 2e\,J_{n+1}(n\,e) - J_{n+2}(n\,e)\Big)^2 \\ &+ \left(1-e^2\right)\Big(J_{n-2}(n\,e) - 2J_n(n\,e) + J_{n+2}(n\,e)\Big)^2 + \frac{4}{3n^2}\,J_n^2(n\,e)\Bigg]\ . \end{aligned} \tag{3.20}$$

Inserting this into equation (3.14) the energy loss can be expressed as

$$\begin{aligned} \dot{E}^h &= \frac{\kappa}{5\pi}\sum_{n=1}^{2}(n\,\Omega)^2\,\mu^2\,a^4\,(n\,\Omega)^4\,b(n,e) \\ &= \frac{32G}{5}\,\mu^2\,a^4\,\Omega^6\left(1-e^2\right)^{-7/2}\left(1 + \frac{73}{24}\,e^2 + \frac{37}{96}\,e^4\right)\ , \end{aligned} \tag{3.21}$$

which is the so called *Peter-Mathews formula* [88] extending the quadrupole formula for gravitational wave emission by Einstein [89] to non-circular orbits.

This energy loss has also an influence back on the Kepler orbit assumed above. The increase in binding energy E_b [85], which itself is given by

$$E_b = -G\,\frac{M\,\mu}{2\,a} = -\,\mu\left(2\pi\,\frac{G\,M}{P_b}\right)^{2/3}\ , \tag{3.22}$$

leads to a decrease in the distance of the objects which also causes an decrease in the orbital period of the form

$$\dot{P}_b = -\,6\pi\,\sqrt{\frac{\mu\,a^5}{(G\,M)^3}}\,\dot{E}\ . \tag{3.23}$$

3.3 Modifications induced by f(R) gravity

There are many possibilities to extend GR. An interesting candidate modifying the behaviour at very high curvature and big scales is $f(R)$ gravity, in which the Ricci scalar R in the Einstein-Hilbert action iss being replaced by a function of it. It can be specified as in [90, 91] or used to describe a dark matter (DM) model like [92] via a dynamical equivalence, or be an effective theory for the corrections by quantum gravity. Because of the large realm of possibilities with models motivated differently well, in this work an agnostic ansatz is chosen, where general constraints for the shape of $f(R)$ are calculated. These can later be fit to an individual model [93, 94].

3.3.1 The linearised action for f(R) gravity

To get the linearised action for this modified theory, we take the function $f(R)$ and expand it around $R = 0$. Because we have to resemble GR with only a small deviation at higher orders one can set $f(0) = 0$ and $f'(0) = 1$. [66] By that the function simplifies to

$$f(R) = R + \alpha R^2 + \mathcal{O}\left(R^3\right) , \tag{3.24}$$

where $\alpha := 1/2\, f''(0)$ is the free parameter of the lowest order deviation from GR. Expanding the gravitational action with respect to $h_{\mu\nu}$ like in section 3.2, we get the action

$$\begin{aligned} S = \int \mathrm{d}^3x \Big[1/2 \left(h^{\mu\nu} \Box h_{\mu\nu} - h \Box h - h^{\mu\nu}\partial_\mu \partial_\rho h^\rho{}_\nu + h \partial_\mu \partial_\nu h^{\mu\nu} + h^{\mu\nu} \partial_\mu \partial_\nu h \right) \\ +2\alpha \left(h \Box^2 h + h^{\mu\nu} \partial_\mu \partial_\nu \partial_\rho \partial_\sigma h^{\rho\sigma} - h^{\mu\nu} \partial_\mu \partial_\nu \Box h - h \Box \partial_\mu \partial_\nu h^{\mu\nu} \right) + 1/2\, \kappa\, h^{\mu\nu} \overline{T}_{\mu\nu} \Big] . \end{aligned} \tag{3.25}$$

An elegant way to deal with the new terms proportional to 2α is to perform a conformal transformation from the *Jordan frame*, in which the Lagrangian looks like derived above, to the *Einstein frame*. For these it can be shown that they are dynamically equivalent [95].

The transformation itself is defined by the relation of the metrics over the conformal factor $A(\phi) := (f'(R))^{-1/2}$. By this the Einstein frame metric is $\tilde{g}_{\mu\nu} = A^2(\phi)\, g_{\mu\nu}$ [96] and the corresponding action has the form

$$S = \int \sqrt{-\tilde{g}}\, \mathrm{d}^4x \left(-\frac{\tilde{R}}{2\kappa} - 1/2\, \partial_\mu \phi\, \partial^\mu \phi + V(\phi) + \mathcal{L}_M(A^2(\phi)\, \tilde{g}_{\mu\nu}\, , \, \Psi) \right) , \tag{3.26}$$

which looks like the action for GR plus a scalar field $\phi \propto \ln\left(f'(R)\right)$ with the potential $V(\phi)$ given by

$$V(\phi) = \frac{R\, f'(R) - f(R)}{2\kappa\, (f'(R))^2} . \tag{3.27}$$

This resembles a DM candidate for example axion-like particles (ALPs) and the linearisation for it can be done analogue to the Jordan frame by expanding each term in the sum individually. The first term will again be the same as in the GR case, while the scalar field part is determined by the dynamic part expressed as

$$\sqrt{-\tilde{g}}\, \partial_\mu \phi\, \partial^\mu \phi = (1 + 1/2\, \kappa\, h) \left(\eta^{\mu\nu} - \kappa\, h^{\mu\nu} \right) \partial_\mu \phi\, \partial^\mu \phi \tag{3.28}$$

and the potential term that can be written like

$$\sqrt{-\tilde{g}}\, V(\phi) = (1 + 1/2\, \kappa\, h)\, V_{\mathrm{VEV}} + 1/\sqrt{24}\, \kappa\, \rho_{\mathrm{VEV}} (\phi - \phi_{\mathrm{VEV}}) + 1/2\, m_\phi (\phi - \phi_{\mathrm{VEV}})^2 + \mathcal{O}\left(\phi^3\right)^{4}, \tag{3.29}$$

where the density of the cosmological background due to the presence of the scalar field ρ_{VEV} appears as well as the mass of this field $m_\phi := \sqrt{V''(\phi_{\mathrm{VEV}})}$.

[4] The subscript VEV denotes the vacuum expectation value of the scalar field ϕ or indicates that this should be used for quantities depending on this field.

Because in the first term in the expansion V_{VEV} acts like an cosmological constant and ρ_{VEV} is the cosmological background density, the terms can be neglected, since the timescale of the evolution of the compact binary orbits is much smaller than of the cosmological evolution and the densities that are much larger. Neglecting them in the scalar field part this simplifies to

$$S_\phi = \int \mathrm{d}^4 x \left({}^1\!/\!{}_2\, \partial_\mu \left(\phi - \phi_{\mathrm{VEV}}\right) \partial^\mu \left(\phi - \phi_{\mathrm{VEV}}\right) - {}^1\!/\!{}_2\, m_\phi^2 \left(\phi - \phi_{\mathrm{VEV}}\right)^2 \right) . \tag{3.30}$$

The last complication to this transformation is that the matter part effectively still depends in the untransformed metric $g_{\mu\nu}$. To resolve this the transformation of the energy-momentum tensor can be used, which is

$$\tilde{T}^{\mu\nu} = - \left(2\partial_{\eta_{\mu\nu}} + \eta^{\mu\nu} \right) \mathcal{L}_M = A_{\mathrm{VEV}}^2\, \overline{T}^{\mu\nu} , \tag{3.31}$$

leading to the expression of the matter part depending on this

$$S_M = \int \mathrm{d}^4 x A_{\mathrm{VEV}}^4 \left({}^1\!/\!\sqrt{24}\, \kappa\, \tilde{T} \left(\phi - \phi_{\mathrm{VEV}}\right) + {}^1\!/\!{}_2\, \kappa\, h^{\mu\nu}\, \tilde{T}_{\mu\nu} \right) . \tag{3.32}$$

3.3.2 Modifications to the pulsar system

For the behaviour of the pulsars the first important thing is that even though the new scalar field could act as a fifth force altering the orbit, there are very strong constraints from solar system tests like from the Cassini mission [97]. Because of the higher surface gravity of compact objects the possible deviations are here even smaller and it is still valid to model the orbits as Keplerian.

For the tensor radiation we get almost the same emission rate like in the GR case. Only the conformal factor in the transformed energy momentum tensor in equation (3.31) leads to an additional dependency on it resulting in

$$d\Gamma^h = \frac{\kappa^2}{4} A_{\mathrm{VEV}}^8 \sum_{\lambda=1}^{2} \left| \overline{T}_{\mu\nu}(k')\, \epsilon^{\mathrm{TT}\mu\nu}_{(\lambda)}(k) \right|^2 \delta\left(\omega - \omega'\right) \frac{\pi}{\omega} \frac{\mathrm{d}^3 k}{(2\pi)^3} . \tag{3.33}$$

This leads to the same modification of for the energy loss, which expressed with $f'(R_{\mathrm{VEV}}) = A_{\mathrm{VEV}}^{-2}$ is

$$\dot{E}_h = \frac{32G}{5} f'(R_{\mathrm{VEV}})^{-4}\, \mu^2\, a^4\, \Omega^6 \left(1 - e^2\right)^{-7/2} \left(1 + \frac{73}{24} e^2 + \frac{37}{96} e^4\right) . \tag{3.34}$$

Assuming the objects to be of constant mass density for the inner solution of the scalar field one can match an exterior solution [66] given by

$$\phi(r) = \phi_{\mathrm{VEV}} - M_{Pl}\, G\, \epsilon\, \frac{M}{R} \exp\left(- m_\phi\, r\right) , \tag{3.35}$$

where the screened parameter ϵ is introduced. This is given by the Newtonian surface gravitational potential $\Phi = G\,\beta$ containing the compactness $\beta := {}^M\!/\!{}_R$ and the external VEV of the field resulting in

$$\epsilon = M_{Pl} \frac{\phi_{\mathrm{VEV}}}{\Phi} = - \sqrt{3/2}\, \frac{\ln f'(R_{\mathrm{VEV}})}{\Phi} . \tag{3.36}$$

From equation (3.35) one can additionally identify the scalar charge of the objects that is given by the linear dependency on the mass $Q := 4\pi\, M_{Pl}\, G\, \epsilon\, M$ and by being in the non-relativistic limit the energy density is simply $\rho = -\tilde{T}$. From this and the relation for the scalar charge $\rho_s = {}^{M_{Pl}}\!/\!\sqrt{6}\,\rho$ the interaction Lagrangian is

$$\mathcal{L}_S = {}^1\!/\!_2\, \kappa\, M_{Pl}\, A^4_{\text{VEV}}\, (\phi - \phi_{\text{VEV}})\, \rho\,, \tag{3.37}$$

resulting in the scalar part of the radiation emission then of the form

$$d\Gamma^s = ({}^1\!/\!_2\, \kappa\, M_{Pl})^2\, A^8_{\text{VEV}} \int |\rho_s(\omega')|^2\, \delta\,(\omega - \omega')\, \frac{\pi}{\omega}\, \frac{\mathrm{d}^3 k}{(2\pi)^3}\,. \tag{3.38}$$

Fourier transforming the scalar density and expanding it regarding the wave number leads to

$$\rho_s(\omega') = (Q_p + Q_c)\, \delta\,(\omega') + i\, \mu \left(\frac{Q_p}{M_p} + \frac{Q_c}{M_c}\right) \left(k_x\, x(\omega') + k_y\, y(\omega')\right) + \mathcal{O}\left(k^2\right)\,, \tag{3.39}$$

where the density is expressed in terms of the charges of the individual objects that are described as point particles with $\rho_s = Q_p\, \delta^3\,(x - x_p) + Q_c\, \delta^3\,(x - x_c)$. Because the first term does not contribute in the radiation formula one can use the transformed coordinates in equation (3.17) and the angular average $\langle k_i \rangle = {}^1\!/\!_3\, \Omega^2 \left(n^2 - ({}^{m_\phi}\!/\!_\Omega)^2\right)$ [98] to express the scalar density as

$$|\rho_s|^2 = {}^1\!/\!_3\, \mu^2\, a^2\, \Omega^2 \left(\frac{Q_p}{m_p} + \frac{Q_c}{m_c}\right)^2 \frac{1}{n^2} \left(n^2 - \left(\frac{m_\phi}{\Omega}\right)^2\right) \left(J_n'(n\,e)^2 + \frac{1-e^2}{e^2} J_n^2(n\,e)\right)\,. \tag{3.40}$$

With this and equation (3.38) the energy loss can be described for the scalar part as well like

$$\begin{aligned}\dot{E}_s &= {}^1\!/\!_6\, \kappa^2\, M_{Pl}\, A^8_{\text{VEV}} \left(\frac{Q_p}{m_p} + \frac{Q_c}{m_c}\right)^2 \mu^2\, a^2\, \Omega^4 \sum_{n=1}^{\infty} \frac{1}{n} \left(n^2 - \left(\frac{m_\phi}{\Omega}\right)^2\right)^{3/2} \left(J_n'(n\,e)^2 + \frac{1-e^2}{e^2} J_n^2(n\,e)\right) \\ &= {}^1\!/\!_3\, G\, A^8_{\text{VEV}}\, (\epsilon_p + \epsilon_c)^2\, \mu^2\, a^2\, \Omega^4 * S(m_\phi)\,,\end{aligned} \tag{3.41}$$

where the term of the sum over the radiation written as $S(m_\phi)$ can be investigated further:

$$\begin{aligned} S(m_\phi) &:= \sum_{n=1}^{\infty} n^{-1} \left(n^2 - \left(\frac{m_\phi}{\Omega}\right)^2\right)^{-3/2} \left(J_n'(n\,e)^2 + (e^{-2} - 1)\, J_n(n\,e)^2\right) \\ &= \begin{cases} \left(1 - e^2\right)^{-5/2} \left(1 + {}^1\!/\!_2\, e^2\right) & ,\ m_\phi \ll \Omega \\ \sum_{n=\lceil m_\phi/\hbar\Omega \rceil}^{\infty} n^{-1} \left(n^2 - \left(\frac{m_\phi}{\Omega}\right)^2\right)^{-3/2} \exp\left(-2n\,e\right) & ,\ m_\phi \gg \Omega\,. \end{cases}\end{aligned} \tag{3.42}$$

This shows that for larger masses the scalar radiation is exponentially suppressed until it reaches a constant level for very small masses, where the formula becomes independent of it.

From there only the definition of the screened parameter ϵ is needed from equation (3.36) to find the simpler expression containing only the derivative of $f(R)$

$$\dot{E}_s = {}^1\!/\!_2\, f'(R_{\text{VEV}})^{-4} \left(\frac{\ln f'(R_{\text{VEV}})}{\Delta\beta}\right)^2 \mu^2\, a^2\, \Omega^4\, S(m_\phi)\,, \tag{3.43}$$

where for the resulting compactness difference of the surface gravity terms the quantity $\Delta\beta$ is defined in analogy to the reduced mass as

$$\Delta\beta := \left|\frac{\beta_p\, \beta_c}{\beta_p - \beta_c}\right| = \left|\frac{R_p}{M_p} - \frac{R_c}{M_c}\right|^{-1}\,. \tag{3.44}$$

3.4 Constraints on the deviation from general relativity

To see the effect of this modification of GR in the systems described in chapter 2, the power radiated away can be linked to the orbital period shift as shown in equation (3.23). If this is done for the theory with and without the modification, one gets predictions on this quantity that can be compared to each other and the observed values listed in table 4.1. To do this on the level of one individual pulsar, the used inequality is

$$1 \overset{!}{>} \frac{\Delta_{f(R)}}{\Delta_{GR}} = \left| \frac{\dot{P}_{b,f(R)} - \dot{P}_{b,GR}}{\dot{P}_{b,intr} - \dot{P}_{b,GR}} \right| . \tag{3.45}$$

By this a modification is ruled out if the pair of $f'(R)$ and m_ϕ values result in a larger deviation from the GR prediction than the measured shift. This may seem counter-intuitive because the GR prediction serves as the null hypothesis and not the actual measurement. But because here individual measurements are used, which are random manifestations of the distribution of possible values due to the uncertainties of the measurement, these cannot function as such. Hence, $\dot{P}_{b,f(R)} \in \left(\dot{P}_{b,intr}, 2\dot{P}_{b,GR} - \dot{P}_{b,intr}\right)$ is the allowed interval for the modified gravity predictions.

This still does not rule out that a single pulsar, which because of the uncertainty of the measure accidentally would be observed like predicted by GR, could exclude reasonable parts of the parameter space. Because of this constraints are always calculated and compared for a group of pulsars. In addition to this, the broadening of the allowed interval due to the uncertainties of the measurements are calculated.

For the calculations regarding the solving of the inequality for $f'(R)$ and m_ϕ, that are discussed in detail in the following chapter, it is useful that one can set an upper bound on the value of $f'(R)$ from solar system constraints as well. The timing of the radio communication of the Cassini mission leads to $|\epsilon| < 2.3 \times 10^{-5}$ [99], which together with equation (3.36) will constrain $f'(R)$. Especially values which are not of the order of 1 can be excluded a priori.

4 Numerical implementation

The implementation of the numerical calculations necessary for the constraints on $f(R)$ is the last part missing for creating these. The code written for this work can be found in the GitHub repositories from the DMGW-Goethe organisation [100] *NSMassRadiusCurves* and *PulsarConstraints*. Besides the standard C++ and Python libraries, it features the libraries used for this work as well.

The following sections will provide a detailed description of each individual code base. For the latter only excerpts of the code are listed in this chapter. The complete files can be found in the repositories and all objects will be linked to there or an official documentation if possible.

4.1 Bounds on the neutron star compactnesses

As introduced in section 3.3.1, the compactnesses of the objects come into play when scalar radiation is considered. To set a bound on its possible values for the case of a NS, the *FBS-Solver* [101] is used. This code from the DMGW group has its origin in solving the TOV equation for *fermion-boson stars (FBSs)*. But by setting the input parameters for the additional bosonic field to zero, mass-radius curves can also be calculated for regular NSs quite easily. The input used for this calculation is the CompOSE data listed in table 2.1 together with a range of values for the central density, which then corresponds to one point on the M-R plane for each value.

The solver is written as a python module, as which it was imported in the code calculating the compactness data provided in *NSMassRadiusCurves*. This consist of the bash script *exeThis.sh* for structure, logging and command line options as well as its central part, the *calculation* function shown in listing 4.1. This function ensures the build and correct linking of the module with its *cython* part, in its own subsidiary function. The path to this together with the list of data files is handed to the calculating python file at the end in line 100.

```
73 # install fbs solver with make
74 makeFBS() {
75     local -r goBack=$(pwd)
76     cd $fbsLocP
77     make clean
78     make
79     cd $goBack
80 }
```

```
# part that should be loged
calculation() {
    echo -e "log of exeThis.sh::calculation()\n${separator}\n"

    # stop whole script if error from single command
    setError

    # ensure compiled fbs
    local -r buildExe="${fbsLocP}/main.out"
    [ -f "$buildExe" ] && echo "existing fbs installation" || makeFBS
    echo -e "${separator}\n"

    # get comma seperated list of data files and pyFBS location
    local dataFs="${dataLoc}*.ose"
    dataFs=$(echo $dataFs | sed 'y/ /,/')
    local -r pyFBSLoc="${fbsLocP}/pyfbs/"

    # calculate MR-curves with pyhton interface of fbs-solver
    python3 $pyF $dataFs $pyFBSLoc $resLocP $tmpLocP $plotLocP $massGrid

    echo -e "${separator}\ncalculation finished\n"
}
```

Listing 4.1: Part of the *exeThis.sh* script with the *calculation* function and its subsidiary *makeFBS* that builds the *cython* module. It is executed before the file *main.py* is called by *$pyF*. The other variables ending on *LocP* hold absolute paths to locations in the repository, *massGrid* the user defined parameters for the mass range on the x-axis and *setError* is a function that just sets the *-e* flag for error handling and a trap for the error message.

The called central python file *main.py* named after the contained *main* function shown in listing 4.2 has its most important task in opening a process *Pool* from the *multiprocessing* module in lines 35 to 45. In this the *calcBetaCurves* calculations can be performed asynchronous for each EoS. This parallelisation resulted in a reduction of the computation time with a factor of 6.3 for the number of 12 used EoSs and 8 used threads, which is quite close to the theoretical maximum taking into consideration the small ratio of tasks to threads. If more EoSs were used, it would approach the theoretical limit even closer.

```
27 # main function for plotting of data in file given as argument
@timer
def main() -> int:
30     # parse input from bash script
    argInputs = parseInput()
    print("input parsed, start calculations in integrate.py")

    # do calculations for each EoS in pool of processes
35     procPool = Pool()
    dataList: list[curveData] = []
    for dataFile in argInputs.filenames:
        # perform "dataList.append(calcBetaCurves(dataFile, pyFBSPath, tablePath, \\
            gridParams))" in parallel
        procPool.apply_async(
40             calcBetaCurves,
            args=(dataFile, argInputs.pyFBSPath, argInputs.tablePath, \\
                argInputs.gridParams),
            callback=lambda res: dataList.append(res),
        )
```

```
    procPool.close()
    procPool.join()

    # get boundary curves from all results
    commonData = boundCurves(dataList)

    # do the actual plotting and save data
    saveToFile(dataList, commonData, argInputs.plotPath, argInputs.resultPath)
    print("\ncompactness curve calculations done")

    return os.EX_OK
```

Listing 4.2: The *main* function in *main.py* handling the partially parallelised computation of the compactness curves with *Pool()* included from *multiprocessing*. It is decorated with *timer*, which is a wrapper tracking the execution time of the function and relies on the *parseInput()* function to convert the command line arguments to a *collections.namedtuple* as well as *calcBetaCurves* and *boundCurves* for the calculations itself.

Based on the example code of the included *FBS-Solver*, the calculation of the β-M curves in *integrate.py* is implemented. As depicted in listing 4.3 lines 120 ff. the function *MRCalc* is doing an initial calculation of one curve based on inputted *gridParams* for the central density after loading the EoS data in line 131. The calculation of M-R pairs for a given density is then repeated for a new list generated by *missingRho* in lines 82 ff. until the mass phase space is explored with at least the same granularity as the later applied mass grid at all values. At last it is using *filterStablePart* to only return a section cut off at the maximal stable mass for that particular EoS.

```
81 # get additionally needed densities
def missingRho(rhoMRSets: ary, massStep: float):
    # iterate over points to see, if masses are covered well
    missingRhos: list[float] = []
85  for pos in range(1, rhoMRSets.size):
        # add intermediate point if mass gap to large
        if abs(rhoMRSets[pos]["mass"] - rhoMRSets[pos - 1]["mass"]) > massStep:
            missingRhos.append((rhoMRSets[pos]["rho"] + rhoMRSets[pos - \\
                1]["rho"]) / 2)

90      # break if over maximal mass
        if rhoMRSets[pos]["mass"] < rhoMRSets[pos - 1]["mass"]:
            break

        # if no maximum mass found, broaden range
95      if pos + 1 == rhoMRSets.size:
            missingRhos.append(1.5 * rhoMRSets[pos]["rho"])

    return np.array(missingRhos)

100
# look for the stable part up to the maximal mass
def filterStablePart(rhoMRSets: ary):
    # iterate over all positions of curve
    M, R = [], []
105 for pos in range(1, rhoMRSets.size):
        # break if over maximal mass
        if rhoMRSets[pos]["mass"] < rhoMRSets[pos - 1]["mass"]:
            break
```

```
        # add masses and radii at this position
        M.append(rhoMRSets[pos]["mass"])
        R.append(rhoMRSets[pos]["radius"])

    print("maximal mass = ", M[-1], " at radius ", R[-1])
    # return curve as two arrays
    return QtyGrid(["M", "R"], np.array(M), np.array(R))


# calculate M-r-curve from EoS datafile
def MRCalc(EoSData: str, pyFBSPath: str, tableName: str, gridParams: list[str]):
    # import from local FBS submodule
    try:
        sys.path.append(pyFBSPath)
        import pyfbs_cython as cyfbs
    except ImportError:
        print("pyfbs package not found at " + pyFBSPath)
    else:
        print("pyfbs succsesfully imported")

    # load EoS from Data dir
    EoS = cyfbs.PyEoStable(EoSData)

    # set parameters of solver for no DM
    mu = 1.0
    lam = 0.0
    phi_c = np.array([0.0])
    rho_c = np.geomspace(4e-4, 1e-1, int(float(gridParams[2]) / 10))

    # define aditional variables
    rhoMRSets = dictifyResults(rho_c)
    tableName += "Mu" + str(mu) + "Lam" + str(lam) + ".tb"
    granularity = (float(gridParams[1]) - float(gridParams[0])) / \\
        float(gridParams[2])

    # start actual calculations
    print("start calculation of M-r-curve")
    while True:
        # calculate M-r-curve
        pyMROutput = cyfbs.PyMRcurve.from_rhophi_list(mu, lam, EoS, rho_c, phi_c, \\
            tableName)

        # add to result list in correct positions
        rhoMRSets = np.sort(np.concatenate((rhoMRSets, dictifyResults(rho_c, \\
            pyMROutput))), order="rho")

        # look which densities are also needed, stop calculation if none
        rho_c = missingRho(rhoMRSets, granularity)
        if rho_c.size == 0:
            break

    return filterStablePart(rhoMRSets)
```

Listing 4.3: The part of the call tree of *integrate.py* containing the *MRCalc* function for calculating the mass-radius curves with an algorithm exploring the mass range equally well at all values. This imports *pyfbs_cython* from the *FBS-Solver* and relies heavily on the *rhoMRSets* type, which is a *structured array* setting `dtype=[("rho", float), ("mass", float), ("radius", float)]`.

After completing this task, the post processing of the curves in the second part of *calcBetaCurves* in *integrate.py* starts with calculating the compactnesses for each M-R pair and returning the β-M curve with *calc-Beta*. Thereafter, the created β-M curve is moved to a grid with a constant step size, where the values are interpolated linearly in the *gridify* function. As a next step the β'-M curve can be calculated as well. For this the function *differentiate* calculates the derivative with respect to the mass using *numpy.gradient* [102] with a constant spacing $\Delta M := M_i - M_{i+1}\ \forall i$ and $edge\_order = 2$. This corresponds to the symmetric 5-point-stencil [103]

$$\frac{\mathrm{d}\beta}{\mathrm{d}M}(M_i) \approx \frac{\beta(M_i - 2\Delta M) - 8\beta(M_i - \Delta M) + 8\beta(M_i + \Delta M) - \beta(M_i + 2\Delta M)}{12\Delta M}\,. \tag{4.1}$$

Here the *gaussian_filter1d* from the *scipy.ndimage* module is used to reduce numeric artefacts introduced by moving the curve on a fixed grid.

After the calculations for all EoSs are finished in line 48 of listing 4.2, the calculation of the bounds from the extreme values at each mass point is initiated with *boundCurves*. To do this for both curves, the minimal and maximal compactness at each value is selected and an additional margin of 25 % of the orthogonal difference between these two is added. Because only a limited set of EoSs were used, this is done to ensure not to underestimate the constraints.

The resulting curves are shown in the following two plots in figure 4.1. With *saveToFile* defined in *plotting.py* they are created using *label-lines* [104] and at the same time stored as *HDF5* data [105] to be used in further calculations.

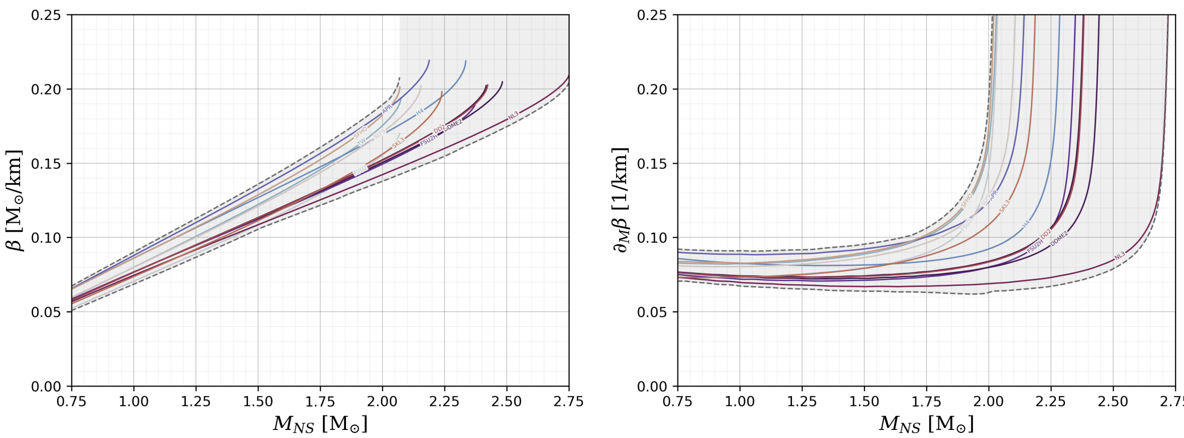


Figure 4.1: Bounds on the compactness on the left-hand-side and its derivative on the right-hand-side shaded in grey with the individual β-M curves for all included EoSs.

4.2 f′(R) constraints for the mass and compactness space

For the implementation of the constraints on f(R) gravity itself, an own code base was written. Like all branches of *PulsarConstraints*, *BetaPlots* uses C++ for the majority of the computations, especially the solving for $f'(R)$. Only the resulting data is transferred with the included *highFive* [105, 106] library to plot the constraints with *Matplotlib* [107] in *plotting.py* and supporting files. This structure is again defined together with the logging and possible user interaction in a file called *exeThis.sh*.

The used data from pulsar measurements that can also be found in table 4.1 is stored as csv-data in *ObservationalData.csv*. When it is loaded to the C++ code, *dataIO::readCSV* makes it possible to filter for companion object type and group tags to only use a subset in the calculations. This function is part of the *dataIO* namespace handling the *HDF5* data transfer and parsing of the command line inputs as well.

The parsing function *dataIO::parseMainInput* is also the first thing the *main* function of the central C++ file *main.cpp* is executing after it is called. After this and the loading of the data is done the calculations here are parallelised, too. This is done by an array of *std::thread*s that executes a *lambda* function defined in lines 35 to 43 of listing 4.4, which again creates a *Pulsar* object and triggers the solving for this. By this, the calculations for as many pulsars as available threads can be done at the same time.

```
31      // define vector for constraints and lambdaExpression to calculate these
32      using dataIO::dTypes::PlotData;
33      vector<PlotData> constraintData(pulsarData.data.size());
34      std::mutex coutMutex;
35      auto oneThreadCalc = [&](uint i)
36      {
37          // lock cout until thread is destructed
38          std::lock_guard<std::mutex> coutLock(coutMutex);
39
40          // create pulsar object and solve for constraints
41          Pulsar pulsar(pulsarData.data.at(i), pulsarData.csvStructure);
42          constraintData.at(i) = pulsar.solving(inputs);
43      };
44
45      // create vector of threads and let them join again
46      vector<std::thread> threads;
47      for (uint i = 0; i < constraintData.size(); i++)
48          threads.push_back(std::thread(oneThreadCalc, i));
49      for (modRef<std::thread> oneThread : threads)
50          oneThread.join();
51      cout << "\nsolving for all pulsars done" << endl;
```

Listing 4.4: lambda expression containing *Pulsar::solving* for each pulsar and returning a vector of *dTypes::PlotData* objects defined in *dataIO.hpp*.]Part of *main.cpp* for the parallel calculations, that is executing a *lambda* expression containing *Pulsar::solving* for each pulsar and returning a vector of *dTypes::PlotData* objects defined in *dataIO.hpp*.

The mentioned *Pulsar* class is defined to calculate the orbital elements, reduced masses and similar in its constructor. In case of a WD companion its compactness from equation (2.9) is calculated here as well. It has only one public member function *Pulsar::solving* printed in listing 4.5 which creates an object from the *Solver* class and defines the solving process with functions from this class.

pulsar		**companion**		**orbit**		
name	m_p $[M_\odot]$	type	m_c $[M_\odot]$	e	P_b [d]	$\lvert\dot{P}_b\rvert$
PSR B1534+12	1.3330(2)	NS	1.3455(2)	0.273 677 40(4)	4.207 372 988 81(2)	$5.59(3) \times 10^{-14}$
PSR J0337+1715	1.44(1)	WD	0.197(8)	$6.98(9) \times 10^{-4}$	1.629 40(6)	$4.79(19) \times 10^{-16}$
PSR J0348+0432	2.01(4)	WD	0.172(3)	$2.0(10) \times 10^{-6}$	0.102 424 062 722(7)	$2.73(45) \times 10^{-13}$
PSR J0437-4715	1.44(7)	WD	0.224(7)	$1.9180(3) \times 10^{-5}$	5.741 045 9(7)	$3.728(6) \times 10^{-12}$
PSR J0453+1559	1.44(1)	NS	1.174(4)	0.112 518 32(4)	4.072 468 588(4)	$3.3(8) \times 10^{-15}$
PSR J0740+6620	2.14(1)	WD	0.251(5)	$5.68(3) \times 10^{-6}$	4.766 944 619 33(8)	$1.2(2) \times 10^{-12}$
PSR J1141-6545	1.27(1)	WD	1.02(1)	0.171 884(1)	0.197 650 959 3(1)	$4.03(25) \times 10^{-13}$
PSR J1518+4904	1.47(4)	NS	1.248(18)	0.249 484 383(9)	8.634 004 961 160(15)	$1.176(5) \times 10^{-15}$
PSR J1756-2251	1.341(70)	NS	1.230(7)	0.180 569 4(2)	3.196 339 014 3(3)	$2.34(9) \times 10^{-13}$
PSR J1757-1854	1.3412(4)	NS	1.3917(4)	0.605 817 40(3)	0.183 537 831 626(4)	$5.294(5) \times 10^{-12}$
PSR J2222-0137	1.831(10)	WD	1.319(4)	$3.8092(1) \times 10^{-4}$	2.445 759 995 469(5)	$1.6(8) \times 10^{-14}$
PSR J2222-0137	1.831(10)	WD	1.319(4)	$3.8092(1) \times 10^{-4}$	2.445 759 995 469(5)	$1.6(8) \times 10^{-14}$

Table 4.1: The data of all used pulsars in the input file *ObservationalData.csv*.

```
// initiate solving process taking range of compactness values to solve for
dataIO::dTypes::PlotData Pulsar::solving(ref<dataIO::dTypes::Input> inputs)
{
    cout << "start solving for pulsar " << name << endl;

    // initialise solver object
    Solver solution(orbitParams, orbitParamErrors);

    // initialise data Structure and copy name of pulsar
    dataIO::dTypes::PlotData results;
    results.pulsarName = name;

    // calculate xAxes values and those to solve for
    vecDouble solvingRange;
    std::tie(results.input, solvingRange) = \\
        solution.calcCompactnessAxes(inputs.betaSettings.betaRange, \\
        inputs.betaSettings.DelBetaRange, inputs.betaSettings.length, \\
        orbitParams.compType, orbitParams.compBeta);

    // add calculated const density compactness if requested
    if (inputs.calcLegacy == true)
    {
        cout << "compactness calculation with uniform density used" << endl;
        solution.addLegacyBeta(results.input.values, solvingRange, \\
            orbitParams.compType, orbitParams.legacyPulBeta, \\
            orbitParams.legacyCompBeta);
    }

    // create mass term
    vecDouble massTerms, solvingMassRange;
    std::tie(massTerms, solvingMassRange) = \\
        solution.calcMassSum(orbitParams.rotVel, orbitParams.eccent, \\
        inputs.fieldMass.range, inputs.fieldMass.length);

    // create setup for inequality
    arr2Double shiftBounds = solution.calcShiftConstraints(observe.intrShift);
    // solve non-analytic equation
    results.bounds = solution.solveEquation(solvingRange, massTerms, shiftBounds);

    // solve with values modified by errors, only if selected
    if (inputs.calcError == false)
        return results;

    // calculate error from mass term
    const vecDouble massTermErrors = \\
        solution.calcMassSumError(orbitParamErrors.rotVel, \\
        orbitParamErrors.eccent, orbitParams.rotVel, orbitParams.eccent, \\
        results.input.values, massTerms);
    // create errors for upper and lower bound and solve for these
    const arr2Double constraintBoundErrors = \\
        solution.calcConstraintErrors(shiftBounds, observe.intrShiftError, \\
        observe.intrShift);
    results.errors = solution.solveErrorEquation(massTermErrors, \\
        constraintBoundErrors);
```

```
70     return results;
}
```

Listing 4.5: The public *Pulsar::solving* function from the *Pulsar* class taking a *dTypes::Input* object and returning one of the form *dTypes::PlotData* defined in *dataIO.hpp*. It creates a Solver object and uses its member variable *orbitParams* to prepare for the solving with the function *solveEquation* inherited from the *Newton* class.

In its constructor the parameters are rearranged such that the equations from section 3.3.1 can be easily constructed. To do this, the total period shift can be written like

$$\left|\dot{P}_b\right| = C_0 \left(C_1 + C_2 \, \ln^2 \mathcal{F}\right) \, [1], \tag{4.2}$$

where the three constants C_i are defined as

$$\begin{aligned} C_0 &= 6\pi \, \mu \, M^{-3/2} \, a^{9/2} \, \Omega^4 \, , \qquad C_2 = 1/2 \, \Delta\beta^{-2} \, S(m_\phi) \, , \\ C_1 &= 32/5 \, a^2 \, \Omega^2 \left(1 - e^2\right)^{-7/2} \left(1 + 73/24 \, e^2 + 37/96 \, e^4\right) \, . \end{aligned} \tag{4.3}$$

From these new constants, the GR prediction for the period shift can be calculated in line 56 of listing 4.5 and from that together with equation (3.45) the allowed interval for $\mathcal{F}$ in modified gravity. Beforehand it is decided depending on the companion type, if β or $\Delta\beta$ will be the x-axis for the plot where this pulsar will be placed in. At the same time the $\Delta\beta$ range for the solving will be calculated by *Solver::calcCompactnessAxes* in lines 41 f. Additionally, a potential additional compactness point from the calculation with a constant density of Narang, Mohanty and Jana [66] can be calculated in lines 45 ff.

4.2.1 Calculation of the term $S(m_\phi)$

The last part missing to fully calculate C_2 is $S(m_\phi)$, the term depending on the mass of the scalar field. When calculating the sum in equation (3.42), the condition that modes with imaginary wave numbers do not contribute can lead to very high orders of n even for the leading ones in the Bessel functions, if $m_\phi \gg \Omega$. Since their implementation in the *boost* library is only reliable up to $n = 127$ [108], an alternative way for calculating these has to be found.

The form $J_n(ne)$ demands an expansion for both high orders and large arguments [109] in the *Support::Sum* class defined in *solverSupport.hpp*. A useful form for this is

$$J_n(n \, \operatorname{sech} \alpha) = \exp\left(n \left(\tanh\alpha - \alpha\right)\right) \, \left(2\pi \, n \, \tanh\alpha\right)^{-1/2} \sum_k (8n)^{-k} \, U_k(\coth\alpha) \, . \tag{4.4}$$

[1] From here on $f'(R_{\mathrm{VEV}})$ is abbreviated as $\mathcal{F}$ for better comprehensibility.

In this expression $U_k(p)$ are polynomials of order k^3 given by

$$
\begin{aligned}
U_0(p) &= 1\,, \\
U_1(p) &= \frac{p}{3}\left(1-5\,p^2\right)\,, \\
U_2(p) &= \frac{p^2}{18}\left(81-462\,p^2+385\,p^4\right)\,, \\
U_3(p) &= \frac{p^3}{810}\left(30\,375-369\,603\,p^2+765\,765\,p^4-425\,425\,p^6\right)\,, \\
U_4(p) &= \frac{p^4}{9720}\Big(4\,465\,125-94\,121\,676\,p^2+349\,922\,430\,p^4 \\
&\qquad -446\,185\,740\,p^6+185\,910\,725\,p^8\Big)\,, \\
U_5(p) &= \frac{p^5}{204\,120}\Big(1\,519\,035\,525-49\,286\,948\,607\,p^2+284\,499\,769\,554\,p^4 \\
&\qquad -614\,135\,872\,350\,p^6+566\,098\,157\,625\,p^8-188\,699\,385\,875\,p^{10}\Big)\,, \\
U_6(p) &= \frac{p^6}{18\,370\,800}\Big(2\,757\,049\,477\,875-127\,577\,298\,354\,750\,p^2+1\,050\,760\,774\,457\,901\,p^4 \\
&\qquad -3\,369\,032\,068\,261\,860\,p^6+5\,104\,696\,716\,244\,125\,p^8 \\
&\qquad -3\,685\,299\,006\,138\,750\,p^{10}+1\,023\,694\,168\,371\,875\,p^{12}\Big)\ \ldots
\end{aligned}
\tag{4.5}
$$

Similarly the derivatives of the Bessel functions $J_n'(ne)$ can be expressed with only some modifications to the expansion [110] as

$$
J_n'(n\ \mathrm{sech}\,\alpha) = \exp\left(n\left(\tanh\alpha-\alpha\right)\right)\ \left(2\pi\,n\ \sinh\alpha\ \cosh\alpha\right)^{-1/2}\sum_k (8n)^{-k}\,V_k(\coth\alpha)\,,
\tag{4.6}
$$

containing the modified polynomials $V_k(p)$ calculated in *polynomials.numbers*:

$$
\begin{aligned}
V_0(p) &= 1\,, \\
V_1(p) &= \frac{p^2}{3}\left(-9+7\,p^2\right)\,, \\
V_2(p) &= \frac{p^2}{18}\left(-135+594\,p^2-455\,p^4\right)\,, \\
V_3(p) &= \frac{p^3}{810}\left(-42\,525+451\,757\,p^2-883\,575\,p^4+475\,475\,p^6\right)\,, \\
V_4(p) &= \frac{p^4}{9720}\Big(-5\,740\,875+111\,234\,708\,p^2-39\,657\,875\,p^4 \\
&\qquad +493\,152\,660\,p^6-202\,076\,875\,p^8\Big)\,, \\
V_5(p) &= \frac{p^5}{204\,120}\Big(-1\,856\,598\,975+56\,869\,556\,085\,p^2-317\,970\,330\,678\,p^4 \\
&\qquad +672\,625\,003\,050\,p^6-611\,386\,010\,235\,p^8+201\,713\,136\,625\,p^{10}\Big)\,,
\end{aligned}
\tag{4.7}
$$

$$V_6(p) = \frac{p^6}{18\,370\,800} \Big(-3\,258\,331\,201\,125 + 144\,587\,604\,802\,050\, p^2 - 1\,161\,367\,171\,769\,260\, p^4$$
$$+3\,661\,991\,378\,545\,500\, p^6 - 5\,482\,822\,398\,928\,870\, p^8$$
$$+3\,923\,060\,232\,341\,250\, p^{10} - 1\,082\,190\,977\,993\,120\, p^{12} \Big) \ldots$$

By using these two expressions and after some simplifications [111, 112], the sum can be written as

$$\begin{aligned} S(m_\phi) = \sum_{n=\lceil m_\phi/\hbar\Omega \rceil}^{\infty} & \frac{\left(n^2 - (m_\phi/\Omega)^2\right)^{-3/2}}{2\pi\, n^2} \frac{\sqrt{1-e^2}}{e\left(1+\sqrt{1-e^2}\right)} \exp\left(2n\, e\right) \\ & * \left[\left(\sum_{k=0}^{\infty} (8n)^{-k}\, U_k\left(1/e\right) \right)^2 + \left(\sum_{k=0}^{\infty} (8n)^{-k}\, V_k\left(1/e\right) \right)^2 \right] . \end{aligned} \tag{4.8}$$

Making use of this formula and breaking the series of polynomials at 6th order like written in equations (4.5, 4.7) only leads to an error of around 2 % even at the third order. If the expansion is used like in the code for $n > 100$, no deviations down to machine precision are noticeable. It can be used until the minimal value of the used *double* datatype is reached for the result. By then the scalar radiation is so much suppressed, that the whole term can be neglected without issues. But it is important to take this into consideration as an edge case at very high masses.

4.2.2 Solving of the inequality with a Newton algorithm

Now everything is set up for the last step in *Pulsar::solving* in line 58 of listing 4.5. The function *solveEquation* is inherited from the *Newton* class. It provides everything needed for the numerical solving of the transcendental equation containing the square of the natural logarithm. For this a Newton algorithm is used because of its quadratic convergence leading to very good performance from the moment it starts converging. Due to the simple form of the function, it is very unlikely to be trapped in a false minimum or loop as long as the starting position is chosen well. Due to this fact, the immediate starting can be expected.

To enable the use of the aforementioned algorithm, the equation (4.2) must be written in the form

$$0 = A - B\,\mathcal{F}^4 + \ln^2 \mathcal{F} \;, \tag{4.9}$$

where there are only two constants left: $A := C_1/C_2$ quantifying the ratio of tensor to scalar radiation and $B := \dot{P}_b/C_0 C_2$. This can be understood via the ratio B/A that describes the ratio of the bound on $\dot{P}_b$ to its GR prediction. From that it can be seen that if $A \gg 1$, only the tensor radiation contributes significantly and the solution simplifies to $\sqrt[4]{A/B}$.

If this is not the case, the solution has to be found iteratively, where one treats the right-hand-side of equation (4.9) as a function $g(\mathcal{F})$ and calculates its derivative, as it is usual for Newton algorithms, resulting in

$$g'(\mathcal{F}) = -3B\,\mathcal{F}^3 + 2\,\frac{\ln \mathcal{F}}{\mathcal{F}} \;. \tag{4.10}$$

With these two the series elements are iteratively given [113] using the condition for the next element $\mathcal{F}_{n+1}$

$$\mathcal{F}_n - \mathcal{F}_{n+1} = \frac{g(\mathcal{F})}{g'(\mathcal{F})}\,, \tag{4.11}$$

resulting in the iteration condition implemented in *Newton::calcStep* at the top of listing 4.6 of the form

$$\mathcal{F}_{n+1} = \mathcal{F}_n \left(1 + \frac{A - B\,\mathcal{F}^4 + \ln^2 \mathcal{F}}{3B\,\mathcal{F}^4 - 2\ln \mathcal{F}} \right) . \tag{4.12}$$

To ensure that no cyclic behaviour is hindering the convergence, after only 100 steps another try with a slightly altered starting position is initiated, which can be found in lines 519 ff. in the definition of *Newton::calcSolution*.

```
479 // algorithm step propto f(R)/f'(R)
480 double Newton::calcStep(ref<NewtStruct> parameters, ref<double> lastPosition)
    {
        const double termProp = parameters.B * pow(lastPosition, 4);
        const double termLog = log(lastPosition);

485     // from x_(n+1) = x_n - f(x_n) / f'(x_n)
        return 1 + (parameters.A - termProp + pow2(termLog)) / (3 * termProp - 2 * \\
            termLog);
    }

    // find solution for all compactnesses
490 double Newton::calcSolution(ref<NewtStruct> parameters, ref<double> start)
    {
        // variables for steps and result set at starting point
        double step, root = start;

495     // set bounds of looping
        const double nbOfSteps = 100;
        const double nbOfStarts = 1e3;

        // iterate over steps until retried too often
500     for (uint stepIndex = 0, retryIndex = 0;; stepIndex++)
        {
            // calculate multiplier for next step
            step = calcStep(parameters, root);
            root *= step;
505
            // throw error if multiplier is 0
            if (step == 0)
                throw std::runtime_error{"multiplier in Newton algorithm is 0, can't \\
                    continue to calculate result!"};

510         // return if root isn't changing anymore
            if (std::abs(1 / step - 1) <= 1e-12)
                return root;

            // try again, if inside of iterative cycle
515         if (stepIndex < nbOfSteps)
                continue;
```

```
            // set new starting point and reset step counter
            root = start * 1.01 * (retryIndex + 1);
        stepIndex = 0;
        retryIndex++;

        // throw exception in case of no convergence
        if (retryIndex == nbOfStarts)
            throw std::runtime_error{"Newton algorithm isn't converging"};
    }
    throw;
}
```

Listing 4.6: The implementation of the Newton algorithm in *Newton.cpp* for a single set of parameters A and B in the *NewtStruct* object and a corresponding starting position handed to *Newton::calcSolution*.

4.2.3 Treatment of different cases for most conservative bounds

Because of the additional scalar radiation it is not clear that there is only one set of parameters for which the predicted period shift is equal to the edges of the allowed range in the inequality equation (3.45). Particularly it is not directly clear which of these are crossing either the upper or lower one. The possible shapes of the curve for $\dot{P}$ and its implications for the resulting intervals in the constraint are depicted in figure 4.2. To get a feel for the behaviour, especially its dependency on the mass of the scalar field, the branch *plotRatio* can be used to produce plots similar to these using the actual data.

Considering this, it becomes necessary to calculate the case from the parameters A and B and set the start as well as the constraint on $\dot{P}$ accordingly. It is done by the *Newton::Cases* subclass defined in *Newton.hpp*. It features the condition defined in *Newton::Cases::decideCase*

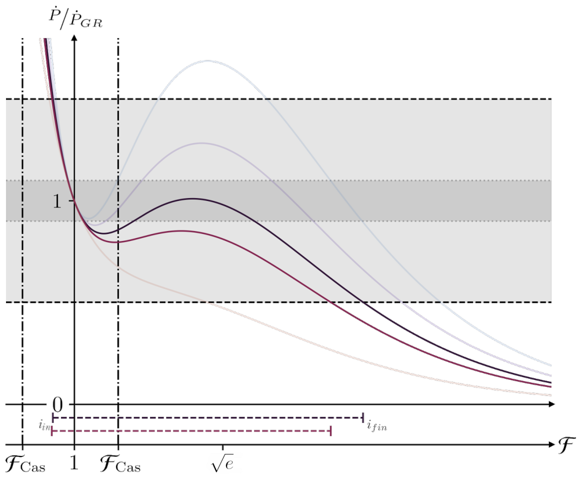
$$B(\dot{P}_{max}) \leq \frac{1+\sqrt{1-16A}}{8\exp\left(1+\sqrt{1-16A}\right)} \tag{4.13}$$

for the maximum due to the $\ln^2$ term at $\mathcal{F}_{max} = \exp\left({}^1\!/\!{}_4\left(1+\sqrt{1-16A}\right)\right)$ being equal to the upper end of the allowed $\dot{P}_b$ interval. There also is the analogue condition for the resulting minimum at smaller values of $\mathcal{F}$,

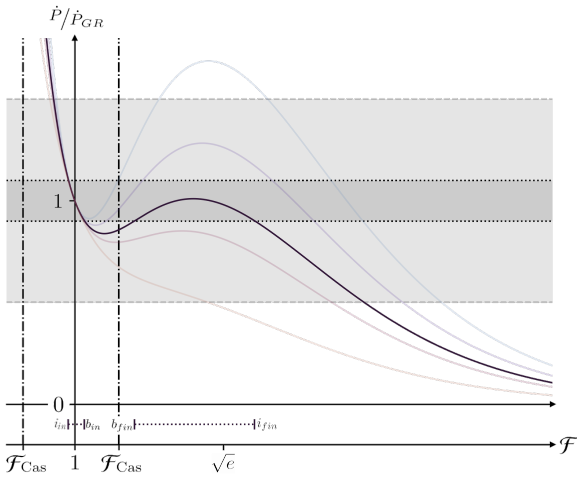
$$B(\dot{P}_{min}) \geq \frac{1-\sqrt{1-16A}}{8\exp\left(1-\sqrt{1-16A}\right)}\,. \tag{4.14}$$

Depending on these conditions, the five cases implemented in the enumeration *Newton::Cases::SolvCases* can be defined as

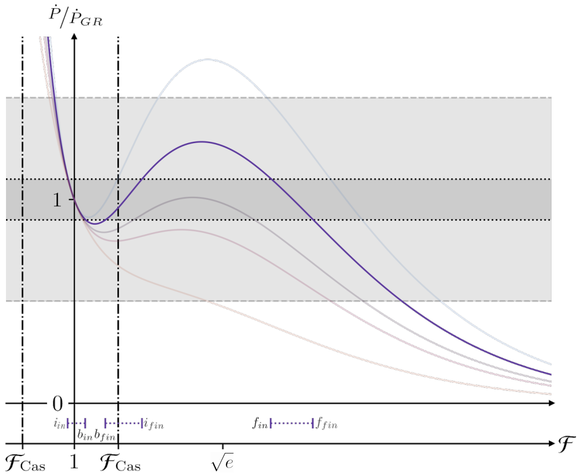
- **I** both conditions are false,
- **II** the upper condition is false, the lower one is true,
- **III** the upper and lower condition are true,
- **IV** the upper condition is true, but the lower one is false,
- **V** the function is monotonous (if $A > {}^1\!/\!{}_{16}$).



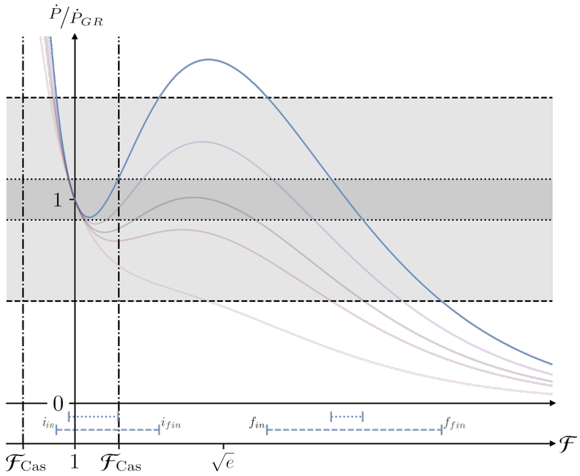
(a) *SolvCase* I.



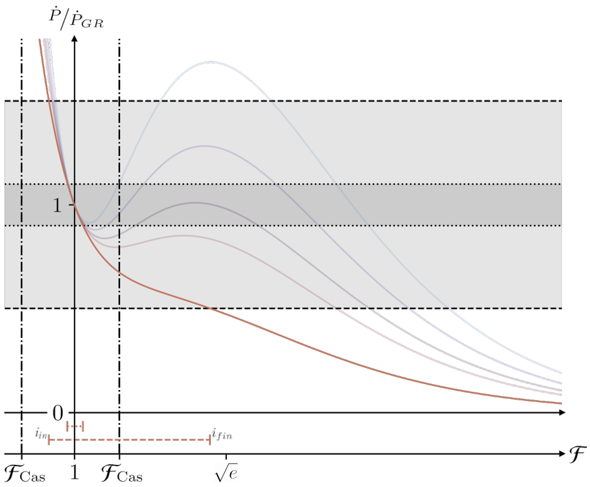
(b) *SolvCase* II.



(c) *SolvCase* III.



(d) *SolvCase* IV.



(e) *SolvCase* V.

Figure 4.2: The allowed interval i, its break b and the far interval f for the different cases. It is depending on the crossings of the interval of $\dot{P}_b$, which here is depicted for a small deviation from the GR prediction and a larger one, and the curves in the f(R) case, here depicted for 5 values of A.

For a growing value of $\Delta\beta$, the cases will always transition in the order V–I–IV or V–II–III–IV and in the moment the upper condition is met, there is a second allowed interval appearing. This lays at values larger that the position of the maximum as it can be seen in figure 4.2c and figure 4.2d. Because there is already the condition $\mathcal{F}_{max} \geq \sqrt{e} \approx 1.6$, the whole interval (f_{in}, f_{fin}) will be at way larger values than allowed by the solar system constraints [97]. This is the reason, why it can be neglected here.

If the lower condition is met, another split of the interval is present. In these cases, besides the endpoints i_{in} and i_{fin}, two more points are added to the list of positions to solve for, the start and end of the breaking interval (b_{in}, b_{fin}). The combination of these four parameter sets and their starting positions is stored in a *Newton::Cases::fullSolverSet*, resulting in an array of these to solve for, stored in the *solverSets* member variable. The solver then uses this list to perform all necessary calculations.

For the positions of possible crossings, the correct constants have to be chosen, or more precisely the correct bound on the period shift has to be used calculating these in *Newton::Cases::setShiftConst*. They are the upper bound for the initial position in all cases and the lower bound for the final position in all cases besides III and IV, where the crossing will be at $\dot{P}/\dot{P}_{GR} > 1$ as well. If there is a break, it will always be present, because the predicted shift will be below the lower bound as it can be seen in figure 4.2b and figure 4.2c. This also shows that it is located at the lower bound and the value of that has to be chosen here, too.

Choosing the starting points for the algorithm is easy as well. This is done in the function *Newton::Cases::setShiftStarts* and the only condition is to be in the surrounding region of the solution in which the first derivative does not change its sign. By that for the first point at i_{in} a simple starting position, which always can be used, is 1. Because the $\mathcal{F}^{-4}$ term as well as the logarithmic one are monotonous at these values, there is no significant difference in the shape of the function between the different cases.

For the end point i_{fin}, this start can also be chosen for case V, because the function behaves monotonous at $\mathcal{F} > 1$ as well. In case III and IV a suitable choice is $\sqrt{e}$, where the second derivative of the function changes sign. This will always be located between the minimum and maximum and by that the convergence to the right position is ensured. Finally, for cases I and II this has to be large enough to be at the tail behind the maximum. This can be acchieved by setting the starting position $A^{-\ln 2/\ln 10}$, which is mimicking the scaling of the maximum at small values of A. The dependency on this parameter ensures not to be at way larger values than necessary for some values of A.

Finally, the boundaries of the break interval 1 for b_{in} and $\sqrt{e}$ for b_{fin} are suitable as for their neighbouring points of the interval i. Because these lay on the same slope, they are leading to the same arguments for the usage of these starting positions. The treatment of the different solution cases together with the solver itself makes it possible to calculate for a position in the M-β phase space, which has been the goal at the beginning of section 4.2.2 and the *Newton::solveEquation* function.

4.3 f′(R) constraints using the bounded compactness range

To calculate the constraints with only $\mathcal{F}$ and m_ϕ as free parameters, there is another branch in the repository called *massPlots*. This features minor changes to the data structure and generation of the x-axis data done in the commit *3bd4145*. Besides that, the most important modification in there is the addition of the data in the file *betaBound.hdf5* depicted in figure 4.1 and the implementation of the lookup mechanism with the class *DelBeta* in *solverSupport.hpp*.

This class enables the calculation of bounds for $\Delta\beta$ for both types of companion objects. As member functions, *lookup* and *diffLookup* are returning the corresponding extreme values at the masses of the system and the other two calculation functions called inside *Solver::calcCompactnessBounds* are calculating $\Delta\beta$ using the latter. Because for the case of a WD companion only the compactness of the pulsar is not fixed, the definition of $\Delta\beta$ in equation (3.44) can directly be used in this case by inserting the extreme compactnesses for the pulsar.

In the case of a NS companion the value can be constructed with a combination of the extreme values of the compactness and its derivative in the interval bounded by the two masses $\left[m_p, m_c\right]$,

$$\Delta\beta = \max_{m=m_p}(\beta)\left|1 + \frac{\max_{m=m_p}(\beta)}{\left(m_p - m_c\right)\min_{m\in\left[m_p,m_c\right]}(\beta')}\right| \tag{4.15}$$

By using the extreme derivative times the mass interval, this will give a value which is at least as large as the actual difference. Therefore, the calculated constraints overestimate the possible range of $\Delta\beta$ values. However, the slow change of the derivative makes this not very severe.

4.3.1 Generalising cases for intervals of Newton constants

Using the described range of $\Delta\beta$ values, there is one new challenge to face. It becomes possible that there is more than one case possible in regards to the allowed values for the constants in the Newton solver A and B. This makes it non-trivial to decide which value corresponding to which case will result in the most conservative bounds on $\mathcal{F}$.

From figure 4.2 one can see that the farthest endpoint will be at the crossing of cases II–III respectively I–IV, where the upper condition from equation (4.13) is on the edge of being true. Because the interval beyond the maximum can be neglected as discussed in section 4.2.3, the side of this transition that is chosen for the calculation is III respectively IV. The definition of this can also be found in line 241 of listing 4.7.

The algorithm used to calculate the value of $\Delta\beta$ that corresponds to this position is a golden search (GS). This is usually implemented for finding a local minimum of a function h by evaluating a it at one point and compare this with its neighbouring values. The special aspect here is that the three points used for comparison have distances in the ratio $\Delta := {}^{3-\sqrt{5}}/_{2}$ [114]. To start the value at the point in the middle, b has to be smaller then the value at the edges at point a and c. A new value at position x is then evaluated. Determined by Δ the position b and of the a point x using its ratio δ is:

$$b = a + \Delta\left(c - a\right) = c - \left(1 - \Delta\right)\left(c - a\right)\ , \quad x := b + \delta\left(c - a\right) = a + \left(\delta + \Delta\right)\left(c - a\right)\ . \tag{4.16}$$

```
// single step for golden search
void Newton::Cases::searchStep(modRef<lmrSet> currentPos)
{
    // choose new point (left + right - middle) with corresponding case
    casePair next = {.consts = {currentPos.front().consts.A + \\
        currentPos.back().consts.A - currentPos.at(1).consts.A,
                                 currentPos.front().consts.B + \\
                                     currentPos.back().consts.B - \\
                                     currentPos.at(1).consts.B}};
    next.sCase = decideCase(next.consts);

    // look for resulting distances in A range
    const double middleDist = abs(currentPos.front().consts.A - \\
        currentPos.at(1).consts.A) / currentPos.at(1).consts.A,
                  nextDist = abs(currentPos.at(1).consts.A - next.consts.A) / \\
                      currentPos.at(1).consts.A;
    const bool wantedCase = (currentPos.at(1).sCase == SolvCases::I || \\
        currentPos.at(1).sCase == SolvCases::II);

    // return for vanishing distance to break recursion
    if (middleDist + nextDist < 10 * doubleLim::epsilon() && wantedCase == true)
        return;

    // next smaller than middle A (more scalar radiation)
    if (next.consts.A < currentPos.at(1).consts.A)
    {
        switch (currentPos.at(1).sCase)
        {
        // below convergent transition
        case SolvCases::IV:
        case SolvCases::III:
            currentPos.front() = next; // set next to new left
            break;

        // above convergent transition
        default:
            currentPos.back() = currentPos.at(1); // set middle to new right
            currentPos.at(1) = next;              // set next to new middle
            break;
        }
    }

    // next larger than middle A (less scalar radiation)
    else
    {
        switch (currentPos.at(1).sCase)
        {
        // below convergent transition
        case SolvCases::IV:
        case SolvCases::III:
            currentPos.front() = currentPos.at(1); // set middle to new left
            currentPos.at(1) = next;               // set next to new middle
            break;

        // above convergent transition
        default:
            currentPos.back() = next; // set next to new right
            break;
        }
```

```
    }

    // recurse and return
    return searchStep(currentPos);
}

// golden search for most conservative bounds
Newton::Cases::casePair Newton::Cases::goldenSearch(ref<NewtPair> fullConsts)
{
    // calculate initial middle position and set up start data
    const double delta = 0.5 * (3 - sqrt(5));
    const NewtStruct lmrConsts[3] = {fullConsts.front(),
                                     {fullConsts.front().A + delta * \\
                                         (fullConsts.back().A - \\
                                         fullConsts.front().A),
                                      fullConsts.front().B + delta * \\
                                          (fullConsts.back().B - \\
                                          fullConsts.front().B)},
                                     fullConsts.back()};
    lmrSet SearchSet;
    for (uint i = 0; i < SearchSet.size(); i++)
    {
        SearchSet.at(i).consts = lmrConsts[i];
        SearchSet.at(i).sCase = decideCase(lmrConsts[i]);
    }

    // search and return middle position
    searchStep(SearchSet);
    return SearchSet.at(1);
}
```

Listing 4.7: The implementation of the GS algorithm with *Newton::Cases::goldenSearch* and *Newton::Cases::searchStep* in *newton.cpp* to recursively find a case transition based on the pair of boundary values *fullConsts* for A and B. In this, *lmrSet* (defined as pairs of *SolvCase* and *NewtStruct* from the *Newton* class as well) is used to hold the data and the function *decideCase* described in section 4.2.3 for their determination.

Depending on whether $h(x)$ is smaller or larger then $h(b)$, the interval used for the next step is either $[a, b]$ or $[b, c]$. After both being as likely as the other, if the shape of $h(x)$ is not known, preserving the golden ratio for the new position by

$$1-\Delta \overset{!}{=} \delta+\Delta \wedge \delta := \frac{\Delta}{1-\Delta} \Leftrightarrow \Delta = \frac{1-\delta}{2} = \frac{1-\Delta\,(1-\Delta)}{2} \Leftrightarrow \Delta = \frac{3-\sqrt{5}}{2} \Rightarrow x = c-b+a\,, \quad (4.17)$$

results in the fastest convergence towards the minimum.

In order to use this algorithm, the case transition orders described in section 4.2.3 play the role of the function $h(x)$ determining if one is below or above the transition, where the convergence should lead to. This is defined in the *Newton::Cases::goldenSearch* function printed in listing 4.7, which is called inside *Newton::Cases::decideShiftCases* as long as the compactnesses don't imply that for their whole range one is above or below the most conservative transition.

For the starting point and the possible break of the interval this extra step does not need to be done. Here looking at figure 4.2 is again enough to determine the most conservative bounds: Because of the additive nature of the scalar radiation, which grows with smaller values of A, this leads to the maximal value of it for the smallest starting position and the minimal one for the break start and end like above described.

Because of the continuous behaviour of the resulting interval with regards to the change of $\Delta\beta$, one can also be sure that all values between a possible break end using the maximal A and furthest end of the interval with a lower value of A are allowed for some value between these and no further computations are necessary.

4.3.2 The creation of the two sets of plots

After the solutions are found and the data is transferred to the plotting script, it is restructured as a new data class in *data.py*. Besides reordering as a list of pulsar data for each mass value in case of these calculations and separating the possible constant density datapoint with *__separatePoint*, the important part here is to categorise the data depending on β or $\Delta\beta$ being the x-axis for the plots to be created. This is done in *__decideXQty*, which sets a boolean variable to trigger the categorisation later.

Then in *plotting.py* shown in listing 4.8 two kinds of colour lists from the *twilight* colour map of *matplotlib* and *pyplot.Figure* objects [107] are prepared. With these the data is fed to *onePulsarPlot* where the data of one pulsar is added to the corresponding set of plots. When all pulsars are plotted, the last thing to do is adding the external constraint from the Cassini mission [115] before the plots are polished and saved.

```
def main() -> int:
    # parse input from bash script
    argInputs = parseInput()

    # read-in data file
    structData = DataLists(filenameInput=argInputs.dataPath, \\
        plotPoint=argInputs.paperRes)

    # create figure objects and colours for plotting
    figLists = iniFigs(len(structData.pulsarList[0].massValues))
    colourLists = createColourLists(structData.pulsarList)

    # plot everything for one pulsar
    print("\nstart plotting")
    for pulsar in structData.pulsarList:
        # get colour pair for plotting and remove from list of available ones
        if pulsar.figIndex == False:
            colourPair = (colourLists.betaColours.lightColours.pop(), \\
                colourLists.betaColours.darkColours.pop())
        else:
            colourPair = (colourLists.DelBetaColours.lightColours.pop(), \\
                colourLists.DelBetaColours.darkColours.pop())

        # plot lines in correct fig with chosen colours
        onePulsarPlot(pulsar, figLists[pulsar.figIndex], colourPair, \\
            argInputs.paperRes, argInputs.errors)
```

```
    print("all data plotted", end="\n\n")

    # do design modifications and save each plot
    filename = argInputs.dataPath.split("/")[-1]
    for figIndex, figList in enumerate(figLists):
        for index, figure in enumerate(figList):
            savePolished(figure, filename, \\
                structData.pulsarList[0].massValues[index], figIndex, \\
                argInputs.plotPath)
            mpl.pyplot.close(figure)
s
        print("generated all plots concerning " + filename + " and " + \\
            str(xQuantities[figIndex]), end="\n\n")

    return os.EX_OK
```

Listing 4.8: The *main* function in the script *plotting.py* in which the restructuring defined in *data.py* is done with the imported *DataLists* before several plotting steps from *plotSup.py* are called.

4.3.3 Including the uncertainties of the measurements

Until now it was not considered for the values used as input from table 4.1 that these come with an uncertainty related to their measurement. This means that the true constraints using this data will be weaker, because there are combinations of that parameters that allow for a broader parameter range.

To capture this, the uncertainties are propagated with Gaussian error propagation for the quantities with known dependencies. For the constants C_i these can be easy calculated from the uncertainties of the depending values x_j by

$$\Delta C_i(x_j) = \sqrt{\sum_j \left(\partial_{x_j} C_j * \Delta x_j\right)^2} \,. \tag{4.18}$$

Most dependencies are just multiplicative, but one difficulty is the uncertainty of the sum $S(m_\phi)$ in equation (4.3). The analogue calculation also using partial derivatives times the corresponding uncertainties becomes

$$\begin{aligned}\Delta S(m_\phi) = \Bigg[&\left[\Delta e * \sum_{n=1}^{\infty} 4n^{-2}\left(n^2 - (m_\phi/\Omega)^2\right)^{3/2}\left(n\left(e^{-1} - e\right) J_n'(ne) - e^{-2} J_n(ne)\right) J_n(ne)\right]^2 \\ &+ \left[\Delta\Omega * \sum_{n=1}^{\infty} 3(m_\phi/\Omega)^2 n^{-1}(n^2 - (m_\phi/\Omega)^2)^{1/2}\left(J_n'(ne)^2 + (e^{-2} - 1) J_n(ne)^2\right)\right]^2\Bigg]^{1/2} ,\end{aligned} \tag{4.19}$$

which is implemented in *Support::Sum::calcEccentPartial* a member of Support.

With these *Newton::Cases::addShiftErrors* ensures to alter the constants A and B by adding or subtracting the uncertainties in such a way the most conservative result possible in this interval is calculated similar to section 4.3.1. The only difficulty emerges for the plots that use the constraint on the compactness. Here there are not only independent A and B uncertainties, but also the $\Delta\beta$ interval which modifies both constants at the same time.

By looking on their dependency on the case defining conditions in equation (4.13) and 4.14, this shows that an increase in A will only allow for a possible transition from *true* to *false* for the lower bound condition and from *false* to *true* for the upper one. For an increase of the value of B as well as $\Delta\beta$ it will be the other way around. So taking the broadened interval using $(A_{max}, B_{min}, \Delta\beta_{min})$ and vice versa will lead to the most conservative constraints.

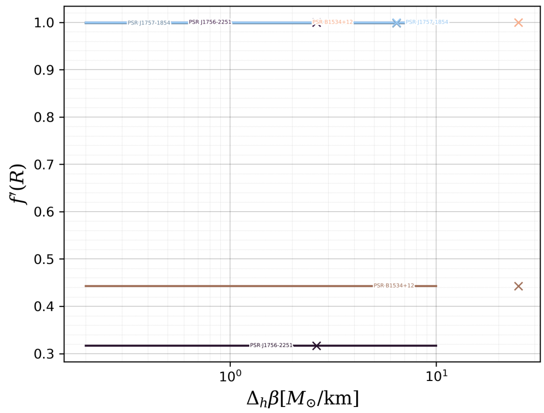
Here the only issue is the case where the GS is used, or more precisely, where it would have to be used when including the uncertainties. It can happen that this is not triggered, because the the alteration of the parameters in the Newton solver can only be done after deciding for a case. If the case then would change depending on these, they would again make a different modification from the uncertainties necessary. However, this will only lead to an overestimation of the allowed values of $\mathcal{F}$ and additionally only for a very small edge case, where the constraints will by far not be competitive in comparison to the solar system constraint. Because of that matter of fact, a more complex treatment would not improve the results and hence, it is left like described above.

By this treatment, the actual Newton solver can stay unchanged and the data is simply added when transferring to the python script. It is plotted as an outer broader area together with the constraints for the mean values from the used measurements.

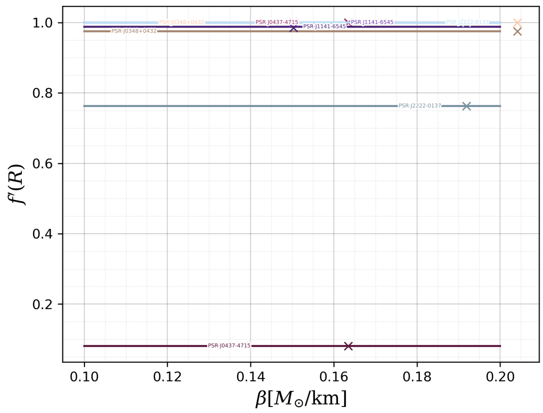
5 Calculated Constraints

5.1 Compactness dependency of the constraints

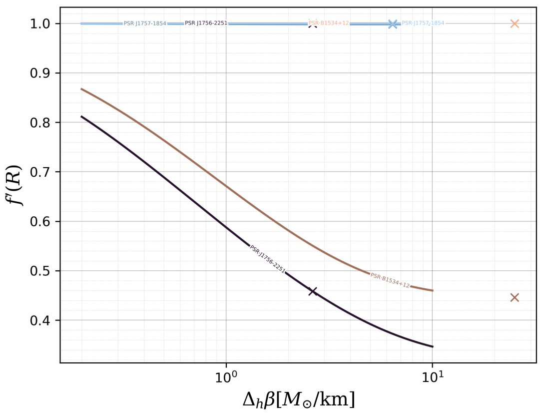
If one calculates the constraints, one has $\Delta\beta$ and m_ϕ as dependent parameters of $f'(R)$. To investigate these, the dependency on the compactness parameter is looked at for a set mass first. This is shown for selected masses in figure 5.1.



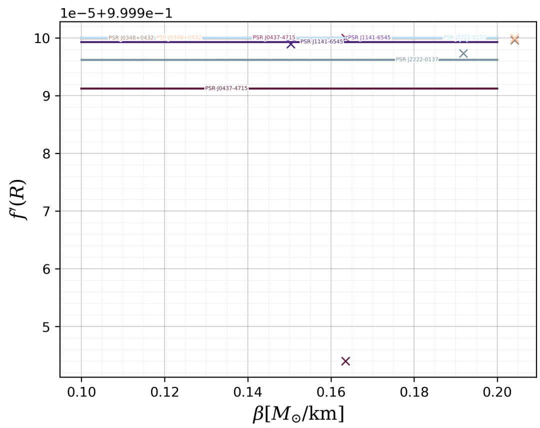
(a) NS-NS binaries and $m_\phi = 10^{-17}$ eV.



(b) NS-WD binaries and $m_\phi = 10^{-17}$ eV.



(c) NS-NS binaries and $m_\phi = 10^{-19}$ eV.



(d) NS-WD binaries and $m_\phi = 10^{-20}$ eV.

Figure 5.1: Compactness dependent constraints for selected masses of the scalar field.

As expected one can see here that there is no mass dependency at larger masses, where the amount of scalar radiation is negligible. This changes when the mass becomes of the order of $\hbar\Omega$ and hence, the scalar radiation becomes relevant via the term $S(m_\phi)$. Because it is suppressed by the factor $\Delta\beta^{-2}$, the constraints in this case are getting weaker for large values of the compactness parameter.

In principle $\Delta\beta$ can go up to infinity, at least in NS-NS systems, because of its definition in equation (3.44). However, this would need a very fine tuned shape of the EoS to get very similar values for the compactnesses of two NSs with different masses. The values using constant densities as they are used by Narang, Mohanty and Jana [66] are at the upper end of a more realistic range for $\Delta\beta$.

For NS-WD systems it can only have even lower values, because its value is approximately the compactness of the WD, which is of the order of $10^{-3}\, M_\odot/\text{km}$. This leads to stronger constraints for those as long as the mass is small enough.

5.2 Constraints using the calculated compactness ranges

To further study the dependency on m_ϕ, constraints were calculated using the *massPlots* branch as described in section 4.3. This results in the plot shown in figure 5.2, where one obtains strong constraints for small masses and much weaker ones for larger masses as described in the previous section.

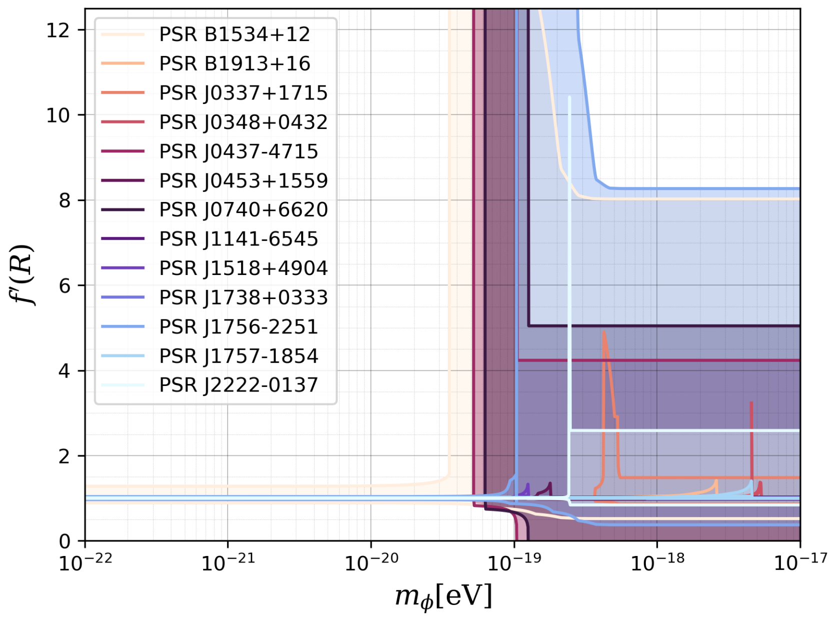
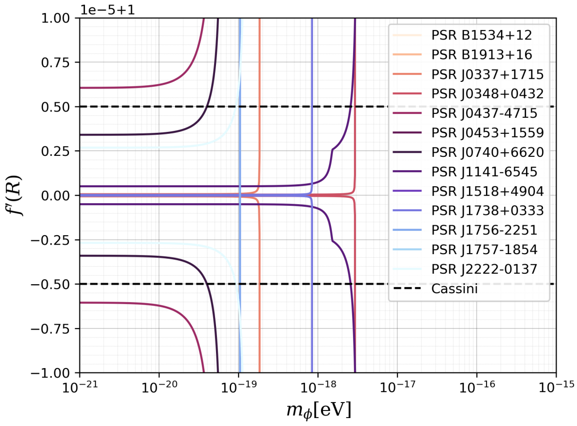


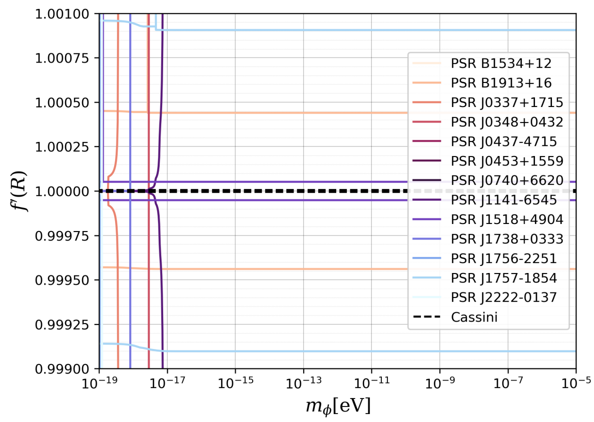
Figure 5.2: $f(R_{\text{VEV}})$-m_ϕ constraints for $f(R)$ gravity using all pulsar data in table 4.1.

A new feature is a sharp spike between these extreme cases, which can be explained by looking at the different cases in figure 4.2. This predicts very weak constraints at the II-III respectively I-IV crossing from the flat shape of the function $g(\mathcal{F})$ for the corresponding parameter values. The vertical section at the end of the spike is an artefact of neglecting the interval f as described in section 4.2.3. This would go on in a continuous manner with a separating interval growing very fast. The lower end of this second break would be the end of the interval, as it is shown here.

In figure 5.3b one can see a zoomed-in depiction of the constraints for the region of no significant scalar radiation. In this case none of the systems produce constraints which are competitive in comparison to the solar system constraint from the Cassini mission. Opposite to this, in figure 5.3a one can see that competitive results are possible for some of them. The mass up to which this is the case depends on the orbital frequency of these as it is discussed in the previous section.



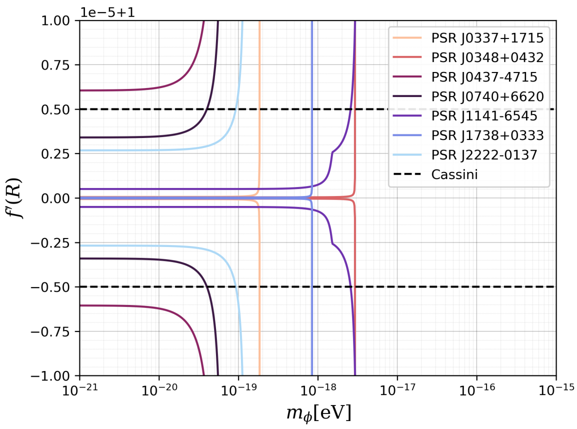
(a) Stronger constraints at low masses.



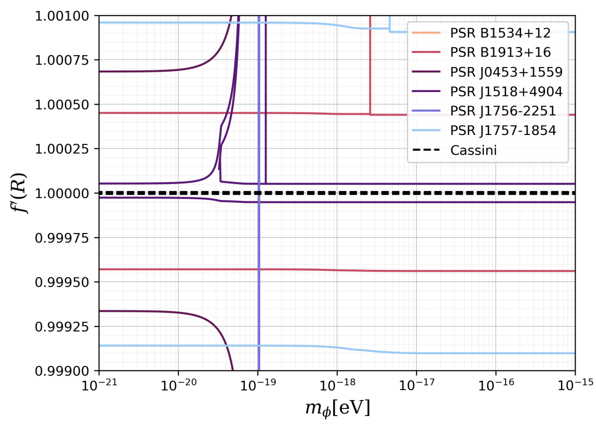
(b) Mass independent constraints at high masses.

Figure 5.3: Zoomed in constraints for the extreme cases of the mass of the field m_ϕ together with the solar system constraint from the Cassini mission as a comparison.

If one separates the systems depending on the type of the companion object like in the previous section, one can see that almost all WD systems produce competitive results while none of the systems with a NS companion, for which the constraints often are not much stronger than in the mass independent case. This is the dominant dependency on the strength of the constraints and can be seen in figure 5.4.



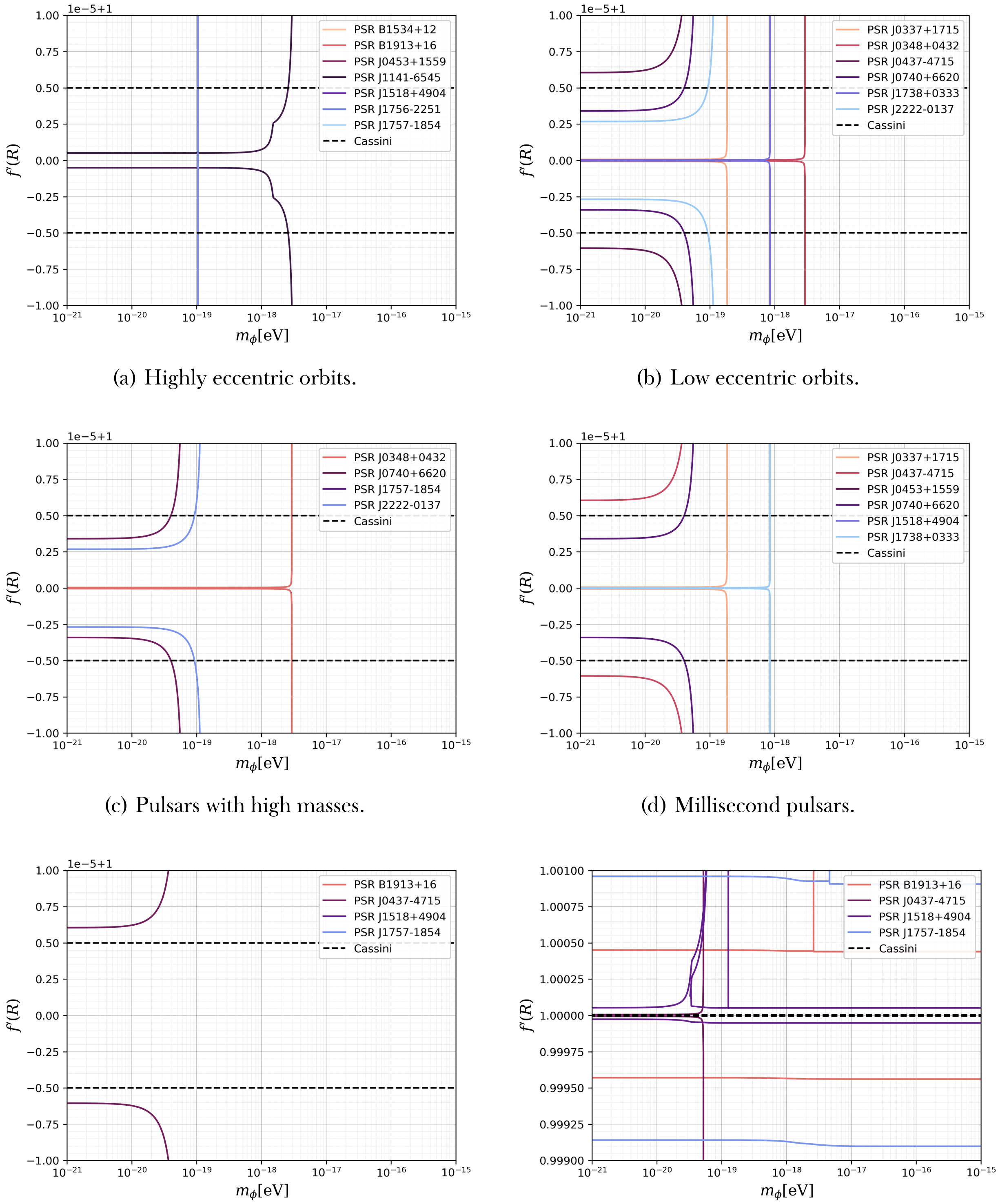
(a) NS-WD binaries.



(b) NS-NS binaries.

Figure 5.4: Comparison of the low mass constraints depending on the companion type.

Additionally, the dependencies on other key features defined in table 2.2 were examined in figure 5.5. For this it becomes clear that between systems with highly eccentric orbits and ones with a very low eccentricity no significant differences were found as in [66], if one takes into account that almost all systems with high eccentricity have NS companions.



(a) Highly eccentric orbits.

(b) Low eccentric orbits.

(c) Pulsars with high masses.

(d) Millisecond pulsars.

(e) Systems measured with high precision.

(f) Extended picture of high precision constraints.

Figure 5.5: Constraints using only certain subsets of the data depending on key features of the system defined in table 2.2.

This is also the case for only using MSPs and systems whose orbital shift was measured with comparatively high precision. Here again the important point is that this was only achieved for NS-NS binaries. Because of the weakness of the constraints, the picture for these systems becomes clearer, if they are depicted with a broader range in figure 5.5f.

5.3 Constraints including the uncertainties of measurements

To provide believe in the trustworthiness and competitiveness of the constraints, it is important to consider how the uncertainties of the measurements used for the data input affect the results. This is depicted in figure 5.6, in which one can see that small differences between the systems depending on the precision of measurements are present.

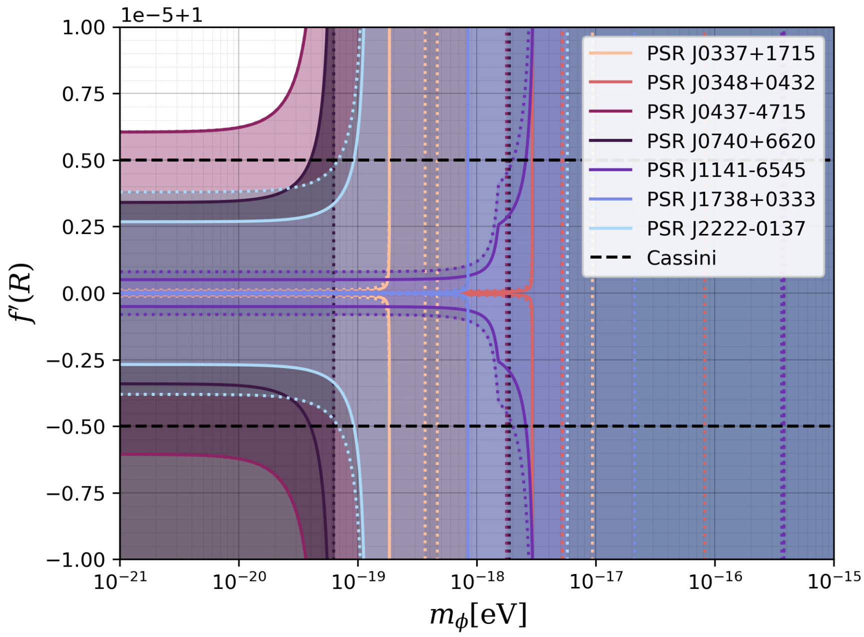


Figure 5.6: Constraints for the NS-WD systems including the uncertainties of the measurements in form of the dashed lines and lighter shaded regions.

However, if one looks at the complete picture, these are way smaller than the differences between the results for similar systems. The issue is that a system which by chance replicates the exact prediction of $\dot{P}_b$ will lead to a very strong constraint. But this is not to be believed, because it is just an arbitrary manifestation of the uncertainty distribution.

Nevertheless, the general picture is not as bad. One can see a number of constraints with small uncertainty bands around $1 \pm 2.5 \times 10^{-6}$ with some being even 1 order of magnitude better, but for building trust in the constraints a combination of the different measurements with their uncertainties in one common constraint would be preferable.

5.4 Conclusion and possible further steps

From looking at the plots in this chapter, it becomes clear that it is the right decision to step away from modelling the compact objects with a constant density. The realistic WDs description massively improves the constraints for these systems and also the treatment of the NS EoS, as it is described in section 5.1, results in a better picture for these.

In addition, the non-trivial shape in figure 5.2 shows that it is worth to not only look at the simplified case at very small masses as it has been done by [66]. Even if the constraints are not competitive enough at the whole mass range, the insights regarding the difference the scalar radiation makes for different systems is very useful.

It also is clear that only looking at three systems – one with a NS companion and one with high and one with low eccentricity – like in the paper mentioned above is not enough to ensure that the constraints are not heavily influenced by the measurement process with its random and systematic errors. Here using multiple pulsars and comparing their individual constraints are also not the ideal solution.

Because of the large number of well observed systems that became available in recent years, a solution could be to abolish the inequality for the constraints in equation (3.45) completely. Instead, the free parameters can be fitted to the measured shift of the orbital period and after that a distribution of the resulting values can be computed to get an allowed region depending on the standard deviation.

The only challenge would be the inclusion of the measurement uncertainties and especially the compactness parameter, which is only given as an allowed interval. But if there is a way to include these in the standard deviation of the results of the individual systems, this could be the right move to step away from GR tests by individual measured pulsars as they are very common [116] and start the era of constraints based on the whole population of them.

The goal for this is to produce a single constraint, which than can be compared with results from LIGO events or pulsar timing arrays (PTAs) [117] shown in figure 5.7. This would be particularly interesting, because they are sensitive to the transmission of gravitational waves and the results in this work are sensitive to their emission in pulsars.

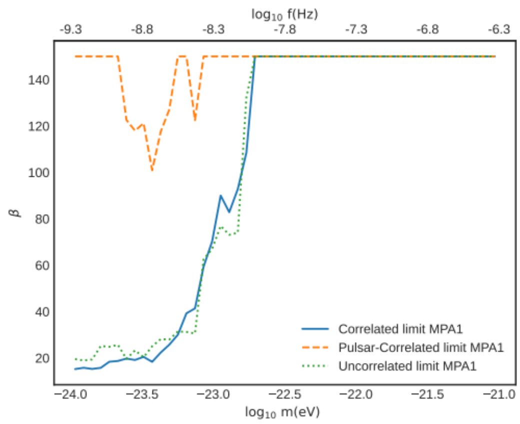


Figure 5.7: Constraints on the self-interaction of ultralight DM for several orders of magnitude of their mass based on the observations by the European Pulsar Timing Array (modified reprint from [117]).

Another improvement could be a better constraint on the compactness parameter. For example, one could go away from looking at individual EoSs and try to systematically include all possible ones with a parametric approach like by Ecker et al. [118]. This could improve the results mainly for NS-NS systems at small masses. However, it will have almost no effect in case of a WD companion, because for them $\Delta\beta$ is dominated by the compactness of the WD and hence, no large improvements are expected. For all systems considered the uncertainty of the compactness parameter was relatively small compared to the ones of the measured quantities, following that this is not the aspect which has the most relevance with regards to further improvements.

When the general constraints are set, one interesting next step would be to look at constraints for individual models. From the relation of $f'(R_{\mathrm{VEV}})$ to the interaction rate of DM models it is possible to constrain them like in [92, 115, 119, 120]. This would also enable a closer comparison to the paper the calculations are based on [66]. Over the second derivative of $f(R)$, which is linked to m_ϕ, it is also possible to constraint individual GR extensions like in [121, 122]. From these the code created here can become a powerful tool to investigate many different models with only modest computation resources and be a valuable complement to the merger and PTA constraints.

Acknowledgements

When achieving important steps in ones life, being in the the spotlight often is not preventable. But the people which surround me, made me who I am and enabled that I could go on this journey, studying and hopefully becoming a physicist soon. This is why I want to mention them here.

First of all, I want to thank Prof. Dr. Laura Sagunski for making the research in this work possible. Being a member of your working group has been a great experience during the year writing this thesis as well as the time of my bachelor thesis, which I am also very happy to have done under your supervision. During these and even in the time before, your enthusiasm for the physics as well as things beyond that and your ability to connect with people made a huge difference for me.

Another incredible valuable support for this work was Tamara, who enabled me to make it presentable to the reader in numerous ways. Moreover, she also deserves many thanks for being such a great office mate. You, like all members of our office and the whole DMGW group, were that open and welcoming, which made some days way more productive and others just encouraging for continuing with trying be this. I do not know, how I could have achieved the research presented in this work without all of you.

And lastly, I cannot thank my parents enough for supporting me during the endless years of my studies, as they have done my whole life. You made it possible for me to do what I am passionate about, giving me a place to be and were I always can come to. You are the foundation to who I am and will become.

Bibliography

1. J. Winterson. *The Stone Gods*. Vintage Canada Edition. Mariner Books, 2007. ISBN: 978-0-307-39723-2.
2. C. Sagan et al. *Murmurs of earth : the Voyager interstellar record*. First Ballatine Books Edition. Random House Inc., New York, 1978. ISBN: 0-345-31536-7. URL: https://archive.org/details/murmursofearthvo00saga/page/n5/mode/2up.
3. E. S. O. (ESO). *The VLT White Book*. "This version of the VLT White Book has been prepared in conjunction with the VLT UT1 First Light event that was successfully achieved during the night of May 25-26, 1998. It is based on a large number of documents and articles of varying scope and level, and the presentation is not yet in the final state." [reverse t.p.] Garching near Munich: European Southern Observatory (ESO), 1998. URL: http://www.eso.org/sci/libraries/historicaldocuments/VLT_White_Book/VLT_Whitebook_A1b.pdf.
4. *A Powerful Twin Arrives – First Images from FORS2 at VLT KUEYEN on Paranal*. Press Release. Observation with FORS2 instrument at VLT Kueyen. The European Southern Observatory, Nov. 1999. URL: https://www.eso.org/public/news/eso9948/.
5. *The Crab Nebula Seen in New Light by NASA's Webb*. Press Release. NASA, ESA, CSA, STScI, Tea Temim (Princeton University), Oct. 2023. URL: https://webbtelescope.org/contents/news-releases/2023/news-2023-137.
6. *Crab Nebula*: *Fingers, Loops and Bays in The Crab Nebula*. Press Release. NASA, CXC, SAO, F.Seward et al, Nov. 2008. URL: https://chandra.harvard.edu/photo/2008/crab/.
7. R. Abbott et al. 'GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run'. In: *Physical Review X* 13.4, 041039 (Oct. 2023), p. 041039. DOI: 10.1103/PhysRevX.13.041039.
8. R. A. Hulse and J. H. Taylor. 'Discovery of a pulsar in a binary system.' In: *The Astrophysical Journal Letters* 195 (Jan. 1975), pp. L51–L53. DOI: 10.1086/181708.
9. R. F. Diedrichs, D. Schmitt and L. Sagunski. 'Binary Systems in Massive Scalar-Tensor Theories: Next-to-Leading Order Gravitational Waveform from Effective Field Theory'. In: (2023). arXiv: 2311.04274 [gr-qc]. URL: https://arxiv.org/abs/2311.04274.
10. P. Freire. *Pulsar mass measurements and tests of general relativity*. Max Planck Institute for Radio Astronomy. URL: https://www3.mpifr-bonn.mpg.de/staff/pfreire/NS_masses.html (visited on 05/06/2024).
11. A. Hewisch et al. 'Observation of a Rapidly Pulsating Radio Source'. In: *Nature* 217.5130 (Feb. 1968), pp. 709–713. DOI: 10.1038/217709a0.
12. D. W. Meltzer and K. S. Thorne. 'Normal Modes of Radial Pulsation of Stars at the End Point of Thermonuclear Evolution'. In: *The Astrophysical Journal* 145 (Aug. 1966), p. 514. DOI: 10.1086/148792.

13. T. Gold. 'Rotating Neutron Stars and the Nature of Pulsars'. In: *Nature* 221.5175 (Jan. 1969), pp. 25–27. DOI: 10.1038/221025a0.

14. F. Pacini. 'Rotating Neutron Stars, Pulsars and Supernova Remnants'. In: *Nature* 219.5150 (July 1968), pp. 145–146. DOI: 10.1038/219145a0.

15. A. Lyne and F. Graham-Smith. '1 The discovery of pulsars'. In: *Pulsar Astronomy*. 4th ed. Cambridge Astrophysics. Cambridge University Press, 2012, 1–15 . DOI: 10.1017/9781108861656.

16. V. Radhakrishnan and D. J. Cooke. 'Magnetic Poles and the Polarization Structure of Pulsar Radiation'. In: *Astrophysical Letters* 3 (Jan. 1969), p. 225.

17. F. D. Drake. 'Radio Observations of the Crab Nebula Pulsar'. In: *The Crab Nebula*. Ed. by R. D. Davies and F. G. Smith. Springer Netherlands, 1971, pp. 73–83. ISBN: 978-94-010-3087-8. DOI: 10.1007/978-94-010-3087-8_9.

18. V. S. Beskin and A. Y. Istomin. 'Pulsar death line revisited – II. 'The death valley''. In: *Monthly Notices of the Royal Astronomical Society* 516.4 (Aug. 2022), pp. 5084–5091. ISSN: 1365-2966. DOI: 10.1093/mnras/stac2423.

19. D. C. Backer et al. 'A millisecond pulsar'. In: *Nature* 300.5893 (Dec. 1982), pp. 615–618. DOI: 10.1038/300615a0.

20. A. Lyne and F. Graham-Smith. '11 Millisecond and binary pulsars'. In: *Pulsar Astronomy*. 4th ed. Cambridge Astrophysics. Cambridge University Press, 2012, pp. 151–169. DOI: 10.1017/9781108861656.

21. R. N. Manchester. 'Millisecond Pulsars, their Evolution and Applications'. In: *Journal of Astrophysics and Astronomy* 38.3 (Sept. 2017). ISSN: 0973-7758. DOI: 10.1007/s12036-017-9469-2.

22. Stanway, E. R. et al. 'VLA radio observations of AR Scorpii∗'. In: *Astronomy & Astrophysics* 611 (2018), A66. DOI: 10.1051/0004-6361/201732380.

23. B. W. Carroll and D. A. Ostlie. 'The Degenerate Remnants of Stars'. In: *An Introduction to Modern Astrophysics*. 2nd edition. Cambridge University Press, 2017, pp. 557–608. ISBN: 978-1-108-42216-1. DOI: 10.1002/andp.19273892002.

24. J. R. Oppenheimer and G. M. Volkoff. 'On Massive Neutron Cores'. In: *Phys. Rev.* 55 (4 Feb. 1939), pp. 374–381. DOI: 10.1103/PhysRev.55.374.

25. M. Nauenberg. 'Analytic Approximations to the Mass-Radius Relation and Energy of Zero-Temperature Stars'. In: *The Astrophysical Journal* 175 (July 1972), p. 417. DOI: 10.1086/151568.

26. W. Baade and F. Zwicky. 'Cosmic Rays from Super-Novae'. In: *Proceedings of the National Academy of Sciences* 20.5 (1934), pp. 259–262. DOI: 10.1073/pnas.20.5.259.

27. D. D. Ofengeim, P. S. Shternin and T. Piran. *A three-parameter characterization of neutron stars' mass-radius relation and equation of state*. 2024. arXiv: 2404.17647 [astro-ph.HE].

28. P. Haensel, A. Y. Potekhin and D. G. Yakovlev, eds. *Neutron Stars 1 – Equation of State and Structure*. Vol. 326. Astrophysics and Space Science Library. Springer-Verlag New York, 2007. DOI: 10.1007/978-0-387-47301-7.

29. M. Oertel et al. 'Equations of state for supernovae and compact stars'. In: *Reviews of Modern Physics* 89.1 (Mar. 2017). ISSN: 1539-0756. DOI: 10.1103/revmodphys.89.015007.

30. B. P. Abbott et al. 'Model comparison from LIGO-Virgo data on GW170817's binary components and consequences for the merger remnant'. In: *Classical and Quantum Gravity* 37.4 (Jan. 2020), p. 045006. DOI: 10.1088/1361-6382/ab5f7c.

31. S. S. Yazadjiev et al. 'Non-perturbative and self-consistent models of neutron stars in R-squared gravity'. In: *Journal of Cosmology and Astroparticle Physics* 2014.06 (June 2014), p. 003. DOI: 10.1088/1475-7516/2014/06/003.

32. H. Hu et al. 'Constraining the dense matter equation-of-state with radio pulsars'. In: *Monthly Notices of the Royal Astronomical Society* 497.3 (July 2020), pp. 3118–3130. ISSN: 1365-2966. DOI: 10.1093/mnras/staa2107.

33. S. Typel et al. *CompOSE Reference Manual*. 2022. arXiv: 2203.03209 [astro-ph.HE].

34. M. Oertel et al. 'Equations of state for supernovae and compact stars'. In: *Rev. Mod. Phys.* 89 (1 Mar. 2017), p. 015007. DOI: 10.1103/RevModPhys.89.015007.

35. C. Ishizuka et al. *code-compose*. gitlab. CompStar European network, June 2022. URL: https://gitlab.obspm.fr/data_and_software_compose/code-compose.

36. M. Hempel and J. Schaffner-Bielich. 'A statistical model for a complete supernova equation of state'. In: *Nuclear Physics A* 837.3 (2010), pp. 210–254. ISSN: 0375-9474. DOI: 10.1016/j.nuclphysa.2010.02.010.

37. M. Hempel et al. 'New Equations of State in Simulations of Core-Collapse Supernovae'. In: *The Astrophysical Journal* 748.1 (Mar. 2012), p. 70. DOI: 10.1088/0004-637X/748/1/70.

38. A. W. Steiner, M. Hempel and T. Fischer. 'Core-Collapse Supernovae Equations of State Based on Neutron Star Observations'. In: *The Astrophysical Journal* 774.1 (Aug. 2013), p. 17. DOI: 10.1088/0004-637X/774/1/17.

39. P. Möller, J. Nix and K.-L. Kratz. 'Nuclear Properties for Astrophysical and Radioactive-Ion-Beam Applications'. In: *Atomic Data and Nuclear Data Tables* 66.2 (1997), pp. 131–343. ISSN: 0092-640X. DOI: https://doi.org/10.1006/adnd.1997.0746.

40. S. Typel et al. 'Composition and thermodynamics of nuclear matter with light clusters'. In: *Phys. Rev. C* 81 (1 Jan. 2010), p. 015803. DOI: 10.1103/PhysRevC.81.015803.

41. A. Akmal, V. R. Pandharipande and D. G. Ravenhall. 'Equation of state of nucleon matter and neutron star structure'. In: *Phys. Rev. C* 58 (3 Sept. 1998), pp. 1804–1828. DOI: 10.1103/PhysRevC.58.1804.

42. G. Baym, C. Pethick and P. Sutherland. 'The Ground State of Matter at High Densities: Equation of State and Stellar Models'. In: *The Astrophysical Journal* 170 (Dec. 1971), p. 299. DOI: 10.1086/151216.

43. Douchin, F. and Haensel, P. 'A unified equation of state of dense matter and neutron star structure'. In: *Astronomy & Astrophysics* 380.1 (2001), pp. 151–167. DOI: 10.1051/0004-6361:20011402.

44. P. Danielewicz and J. Lee. 'Symmetry energy I: Semi-infinite matter'. In: *Nuclear Physics A* 818.1 (2009), pp. 36–96. ISSN: 0375-9474. DOI: 10.1016/j.nuclphysa.2008.11.007.

45. F. Gulminelli and A. R. Raduta. 'Unified treatment of subsaturation stellar matter at zero and finite temperature'. In: *Phys. Rev. C* 92 (5 Nov. 2015), p. 055803. DOI: 10.1103/PhysRevC.92.055803.

46. E. Chabanat. 'Interactions effectives pour des conditions extrêmes d'isospin'. Examiner: Jacques Meyer. ID: 1995LYO10016. PhD thesis. Université de Lille, 1995. URL: `http://www.theses.fr/1995LYO10016`.

47. P.-G. Reinhard and H. Flocard. 'Nuclear effective forces and isotope shifts'. In: *Nuclear Physics A* 584.3 (1995), pp. 467–488. ISSN: 0375-9474. DOI: `https://doi.org/10.1016/0375-9474(94)00770-N`.

48. F. Grill et al. 'Equation of state and thickness of the inner crust of neutron stars'. In: *Phys. Rev. C* 90 (4 Oct. 2014), p. 045803. DOI: `10.1103/PhysRevC.90.045803`.

49. J. M. Pearson et al. 'Erratum: Unified equations of state for cold non-accreting neutron stars with Brussels-Montreal functionals. I. Role of symmetry energy'. In: *Monthly Notices of the Royal Astronomical Society* 486.1 (Mar. 2019), pp. 768–768. ISSN: 0035-8711. DOI: `10.1093/mnras/stz800`.

50. C. Providência et al. 'Hyperonic Stars and the Nuclear Symmetry Energy'. In: *Frontiers in Astronomy and Space Sciences* 6 (2019). ISSN: 2296-987X. DOI: `10.3389/fspas.2019.00013`.

51. R. Negreiros et al. 'Cooling of Small and Massive Hyperonic Stars'. In: *The Astrophysical Journal* 863.1 (Aug. 2018), p. 104. DOI: `10.3847/1538-4357/aad049`.

52. W.-C. Chen and J. Piekarewicz. 'Building relativistic mean field models for finite nuclei and neutron stars'. In: *Phys. Rev. C* 90 (4 Oct. 2014), p. 044305. DOI: `10.1103/PhysRevC.90.044305`.

53. G. A. Lalazissis et al. 'New relativistic mean-field interaction with density-dependent meson-nucleon couplings'. In: *Phys. Rev. C* 71 (2 Feb. 2005), p. 024312. DOI: `10.1103/PhysRevC.71.024312`.

54. S. Typel and H. Wolter. 'Relativistic mean field calculations with density-dependent meson-nucleon coupling'. In: *Nuclear Physics A* 656.3 (1999), pp. 331–364. ISSN: 0375-9474. DOI: `https://doi.org/10.1016/S0375-9474(99)00310-3`.

55. C. J. Horowitz and J. Piekarewicz. 'Neutron radii of ^{208}Pb and neutron stars'. In: *Phys. Rev. C* 64 (6 Nov. 2001), p. 062802. DOI: `10.1103/PhysRevC.64.062802`.

56. C. J. Horowitz and J. Piekarewicz. 'Neutron Star Structure and the Neutron Radius of ^{208}Pb'. In: *Phys. Rev. Lett.* 86 (25 June 2001), pp. 5647–5650. DOI: `10.1103/PhysRevLett.86.5647`.

57. H. Pais and C. Providência. 'Vlasov formalism for extended relativistic mean field models: The crust-core transition and the stellar matter equation of state'. In: *Phys. Rev. C* 94 (1 July 2016), p. 015808. DOI: `10.1103/PhysRevC.94.015808`.

58. G. Grams et al. 'Confronting a set of Skyrme and χ_{EFT} predictions for the crust of neutron stars: On the origin of uncertainties in model predictions'. In: *The European Physical Journal A* 58.3 (Mar. 2022). ISSN: 1434-601X. DOI: `10.1140/epja/s10050-022-00706-w`.

59. O. C. Rømer. 'Démonstration touchant le mouvement de la lumière trouvé'. In: *l'Académie royale des sciences* (1676), pp. 233–236. URL: `https://gallica.bnf.fr/ark:/12148/bpt6k56527v/f234.item`.

60. A. Lyne and F. Graham-Smith. '5 Pulsar timing'. In: *Pulsar Astronomy*. 4th ed. Cambridge Astrophysics. Cambridge University Press, 2012, pp. 61–75. DOI: `10.1017/9781108861656`.

61. R. Blandford and S. A. Teukolsky. 'Arrival-time analysis for a pulsar in a binary system.' In: *The Astrophysical Journal* 205 (Apr. 1976), pp. 580–591. DOI: `10.1086/154315`.

62. O. Hamil et al. 'Braking index of isolated pulsars'. In: *Physical Review D* 91.6 (Mar. 2015). ISSN: 1550-2368. DOI: 10.1103/physrevd.91.063007.

63. A. Lyne and F. Graham-Smith. '6 Timing and astrometry of binary pulsars'. In: *Pulsar Astronomy*. 4th ed. Cambridge Astrophysics. Cambridge University Press, 2012, pp. 76–87. DOI: 10.1017/9781108861656.

64. J. M. Weisberg and Y. Huang. 'Relativistic Measurements from Timing the Binary Pulsar PSR B1913+16'. In: *The Astrophysical Journal* 829.1 (Sept. 2016), p. 55. ISSN: 1538-4357. DOI: 10.3847/0004-637x/829/1/55.

65. J. H. Taylor and J. M. Weisberg. 'A new test of general relativity - Gravitational radiation and the binary pulsar PSR 1913+16'. In: *The Astrophysical Journal* 253 (Feb. 1982), pp. 908–920. DOI: 10.1086/159690.

66. A. Narang, S. Mohanty and S. Jana. 'Gravitational radiation from binary systems in f(R) gravity: A semi-classical approach'. In: *Journal of Cosmology and Astroparticle Physics* 2023.03 (Mar. 2023), p. 008. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2023/03/008.

67. E. Fonseca, I. H. Stairs and S. E. Thorsett. 'A Comprehensive Study of Relativistic Gravity Using PSR B1534+12'. In: *The Astrophysical Journal* 787.1 (May 2014), p. 82. DOI: 10.1088/0004-637X/787/1/82.

68. G. Voisin et al. 'An improved test of the strong equivalence principle with the pulsar in a triple star system'. In: *Astronomy and Astrophysics* 638, A24 (June 2020), A24. DOI: 10.1051/0004-6361/202038104.

69. J. Antoniadis et al. 'A Massive Pulsar in a Compact Relativistic Binary'. In: *Science* 340.6131 (Apr. 2013). ISSN: 1095-9203. DOI: 10.1126/science.1233232.

70. D. J. Reardon et al. 'Timing analysis for 20 millisecond pulsars in the Parkes Pulsar Timing Array'. In: *Monthly Notices of the Royal Astronomical Society* 455.2 (Nov. 2015), pp. 1751–1769. ISSN: 0035-8711. DOI: 10.1093/mnras/stv2395.

71. J. G. Martinez et al. 'Pulsar J0453+1559: A Double Neutron Star System with a Large Mass Asymmetry'. In: *The Astrophysical Journal* 812.2 (Oct. 2015), p. 143. DOI: 10.1088/0004-637x/812/2/143.

72. E. Fonseca et al. 'Refined Mass and Geometric Measurements of the High-mass PSR J0740+6620'. In: *The Astrophysical Journal* 915.1, L12 (July 2021), p. L12. DOI: 10.3847/2041-8213/ac03b8.

73. N. D. R. Bhat, M. Bailes and J. P. W. Verbiest. 'Gravitational-radiation losses from the pulsar-white-dwarf binary PSR J1141–6545'. In: *Physical Review D* 77.12 (June 2008). ISSN: 1550-2368. DOI: 10.1103/physrevd.77.124017.

74. C. M. Tan et al. 'High-cadence Timing of Binary Pulsars with CHIME'. In: *The Astrophysical Journal* 966.1, 26 (May 2024), p. 26. DOI: 10.3847/1538-4357/ad28b2.

75. P. C. C. Freire et al. 'The relativistic pulsar–white dwarf binary PSR J1738+0333 – II. The most stringent test of scalar–tensor gravity'. In: *Monthly Notices of the Royal Astronomical Society* 423.4 (June 2012), pp. 3328–3343. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2012.21253.x.

76. R. D. Ferdman et al. 'PSR J1756–2251: a pulsar with a low-mass neutron star companion'. In: *Monthly Notices of the Royal Astronomical Society* 443.3 (Sept. 2014), pp. 2183–2196. DOI: 10.1093/mnras/stu1223.

77. A. D. Cameron et al. 'New constraints on the kinematic, relativistic, and evolutionary properties of the PSR J1757–1854 double neutron star system'. In: *Monthly Notices of the Royal Astronomical Society* 523.4 (Aug. 2023), pp. 5064–5085. DOI: 10.1093/mnras/stad1712.

78. Y. J. Guo et al. 'PSR J2222–0137: I. Improved physical parameters for the system'. In: *Astronomy & Astrophysics* 654 (Oct. 2021), A16. ISSN: 1432-0746. DOI: 10.1051/0004-6361/202141450.

79. L. Rezzolla and O. Zanotti. *Relativistic Hydrodynamics*. Oxford University Press, 2013. ISBN: 978-0-19-852890-6.

80. M. Maggiore. *Gravitational Waves*. Vol. 1. Oxford University Press, 2008. ISBN: 978-0-19-857074-5.

81. M. J. VELTMAN. 'Quantum Theory of Gravitation'. In: *Methods in Field Theory*. Ed. by R. Balian and J. Zinn-Justin. World Scientific, 1993, pp. 265–328. DOI: 10.1142/9789814412674_0007.

82. D. Hilbert. 'Die Grundlagen der Physik . (Erste Mitteilung.)' In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* (1915), pp. 395–408. URL: http://eudml.org/doc/58946.

83. S. Mohanty. 'Gravitational Waves'. In: *Astroparticle Physics and Cosmology: Perspectives in the Multimessenger Era*. Springer International Publishing, 2020, pp. 191–219. ISBN: 978-3-030-56201-4. DOI: 10.1007/978-3-030-56201-4_7.

84. C. W. Misner, K. S. Thorne and J. A. Wheeler. *Gravitation*. Twenty Fifth Printing. W. H. Freeman and Company, 1973. ISBN: 0-7167-0344-0.

85. B. W. Carroll and D. A. Ostlie. 'Celestial Mechanics'. In: *An Introduction to Modern Astrophysics*. 2nd edition. Cambridge University Press, 2017, pp. 557–608. ISBN: 978-1-108-42216-1. DOI: 10.1002/andp.19273892002.

86. O. I. Marichev. *Handbook of Integral Transforms of Higher Transcendental Functions: Theory and Algorithmic Tables*. Vol. 0-85312-528-7. Edited by F. D. Gakhov, Translated from the Russian by L. W. Longdon. Ellis Horwood Ltd./John Wiley & Sons, Inc, 1983, p. 336. ISBN: 0-85312-528-7.

87. T. K. Poddar, S. Mohanty and S. Jana. 'Gravitational radiation from binary systems in massive graviton theories'. In: *Journal of Cosmology and Astroparticle Physics* 2022.03 (Mar. 2022), p. 019. DOI: 10.1088/1475-7516/2022/03/019.

88. P. C. Peters and J. Mathews. 'Gravitational Radiation from Point Masses in a Keplerian Orbit'. In: *Phys. Rev.* 131 (1 1963), pp. 435–440. DOI: 10.1103/PhysRev.131.435.

89. A. Einstein. 'Über Gravitationswellen'. In: *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften* (Jan. 1918), pp. 154–167. DOI: 10.1007/978-3-662-57411-9_17.

90. F. Payandeh and M. Fathi. 'R2 theory of gravity'. In: *Journal of Physics: Conference Series* 442.1 (June 2013), p. 012053. DOI: 10.1088/1742-6596/442/1/012053.

91. S. Alexander and N. Yunes. 'Chern-Simons modified general relativity'. In: *Physics Reports* 480.1–2 (Aug. 2009), pp. 1–55. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2009.07.002.

92. G. Alonso Alvarez. 'Axions and other light dark matter candidates'. PhD thesis. Combined Faculty of Natural Sciences and Mathematics of the Ruprecht-Karls-Universität of Heidelberg, July 2020. DOI: 10.11588/heidok.00028671.

93. T. P. Sotiriou and V. Faraoni. ‘$f(R)$ theories of gravity’. In: *Reviews of Modern Physics* 82.1 (Mar. 2010), pp. 451–497. DOI: 10.1103/revmodphys.82.451.

94. A. De Felice and S. Tsujikawa. ‘f(R) Theories’. In: *Living Reviews in Relativity* 13.1 (June 2010). ISSN: 1433-8351. DOI: 10.12942/lrr-2010-3.

95. C. T. Hill and G. G. Ross. ‘Gravitational contact interactions and the physical equivalence of Weyl transformations in effective field theory’. In: *Phys. Rev. D* 102 (12 Dec. 2020), p. 125014. DOI: 10.1103/PhysRevD.102.125014.

96. T. Faulkner et al. ‘Constraining $f(R)$ gravity as a scalar-tensor theory’. In: *Physical Review D* 76.6 (Sept. 2007). ISSN: 1550-2368. DOI: 10.1103/physrevd.76.063505.

97. C. M. Will. ‘The Confrontation between General Relativity and Experiment’. In: *Living Reviews in Relativity* 17.1 (June 2014). ISSN: 1433-8351. DOI: 10.12942/lrr-2014-4.

98. J. F. Donoghue, M. M. Ivanov and A. Shkerin. *EPFL Lectures on General Relativity as a Quantum Field Theory*. 2017. arXiv: 1702.00319 [hep-th].

99. X. Zhang, T. Liu and W. Zhao. ‘Gravitational radiation from compact binary systems in screened modified gravity’. In: *Physical Review D* 95.10 (May 2017). ISSN: 2470-0029. DOI: 10.1103/physrevd.95.104027.

100. *DMGW-Goethe*. Github organisation for the corresponding DMGW working group @ Goethe University Frankfurt. URL: https://github.com/DMGW-Goethe.

101. R. F. Diedrichs et al. ‘Tidal deformability of fermion-boson stars: Neutron stars admixed with ultralight dark matter’. In: *Physical Review D* 108.6 (Sept. 2023). ISSN: 2470-0029. DOI: 10.1103/physrevd.108.064009.

102. ‘Array programming with NumPy’. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

103. B. Fornberg. ‘Generation of Finite Difference Formulas on Arbitrarily Spaced Grids’. In: *Mathematics of Computation* 51.184 (1988), pp. 699–706. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/2008770.

104. C. Cadiou. *Matplotlib label lines*. Version v0.5.1. Dec. 2022. DOI: 10.5281/zenodo.7428071.

105. The HDF Group. *Hierarchical Data Format, version 5*. URL: https://portal.hdfgroup.org/documentation/.

106. The Blue Brain Project. *HighFive*. Header-only C++ HDF5 interface. URL: https://github.com/BlueBrain/HighFive.

107. J. D. Hunter. ‘Matplotlib: A 2D graphics environment’. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

108. N. A. A. Bikineev et al. *boost 1.67.0, Math Toolkit 2.6.0*. 2017. URL: https://www.boost.org/doc/libs/1_67_0/libs/math/doc/html/index.html.

109. F. W. J. Olver. *Tables for Bessel functions of moderate or large orders*. Mathematical Tables. London: Her Majesty’s Stationery Office, 1962.

110. B. A. for the Advancement of Science - Committee for the Calculation of Mathematical Tables. *Bessel functions : 2. Functions of positive integer order*. Cambridge, 1960.

111. Ahmose. *The Rhind Mathematical Papyrus*. problem 50. The British Museum, ID: EA10058. year 33 of Apophis. URL: https://www.britishmuseum.org/collection/object/Y_EA10058.

112. A. B. Chace. 'Egyptian Geometry'. In: *The Rhind mathematical papyrus: free translation and commentary with selected photographs, transcriptions, transliterations, and literal translations*. Vol. I. Mathemathical Association of America, 1927, pp. 86–99. ISBN: 978-0873531337.

113. P. Deuflhard. 'Systems of Equations: Local Newton Methods'. In: *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 45–107. ISBN: 978-3-642-23899-4. DOI: 10.1007/978-3-642-23899-4_2.

114. W. H. Press et al. *Numerical Recipes – The Art of Scientific Computing*. Third Edition. Cambridge University Press, 2007. ISBN: 978-0-521-88068-8. URL: https://numerical.recipes/book.html.

115. W. Hu and I. Sawicki. 'Models of $f(R)$ cosmic acceleration that evade solar system tests'. In: *Physical Review D* 76.6 (Sept. 2007). ISSN: 1550-2368. DOI: 10.1103/physrevd.76.064004.

116. G. G. L. Nashed and S. Capozziello. *Constraining $f(\mathcal{R})$ gravity by Pulsar SAX J1748.9-2021 observations*. 2024. arXiv: 2405.09590 [gr-qc].

117. C. Smarra et al. *Constraints on conformal ultralight dark matter couplings from the European Pulsar Timing Array*. May 2024. arXiv: 2405.01633 [astro-ph.HE].

118. C. Ecker et al. *Listening to the long ringdown: a novel way to pinpoint the equation of state in neutron-star cores*. 2024. arXiv: 2403.03246.

119. J. Preskill, M. B. Wise and F. Wilczek. 'Cosmology of the invisible axion'. In: *Physics Letters B* 120.1 (1983), pp. 127–132. DOI: 10.1016/0370-2693(83)90637-8.

120. B. K. Yadav and M. M. Verma. 'Dark matter as scalaron in f(R) gravity models'. In: *Journal of Cosmology and Astroparticle Physics* 2019.10 (Oct. 2019), pp. 052–052. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2019/10/052.

121. E. Berti et al. 'Testing general relativity with present and future astrophysical observations'. In: *Classical and Quantum Gravity* 32.24 (Dec. 2015), p. 243001. ISSN: 1361-6382. DOI: 10.1088/0264-9381/32/24/243001.

122. E. D. Barr et al. 'A pulsar in a binary with a compact object in the mass gap between neutron stars and black holes'. In: *Science* 383.6680 (Jan. 2024), pp. 275–279. ISSN: 1095-9203. DOI: 10.1126/science.adg3005.