

Pavol Jozef Šafárik University in Košice,

Faculty of Science

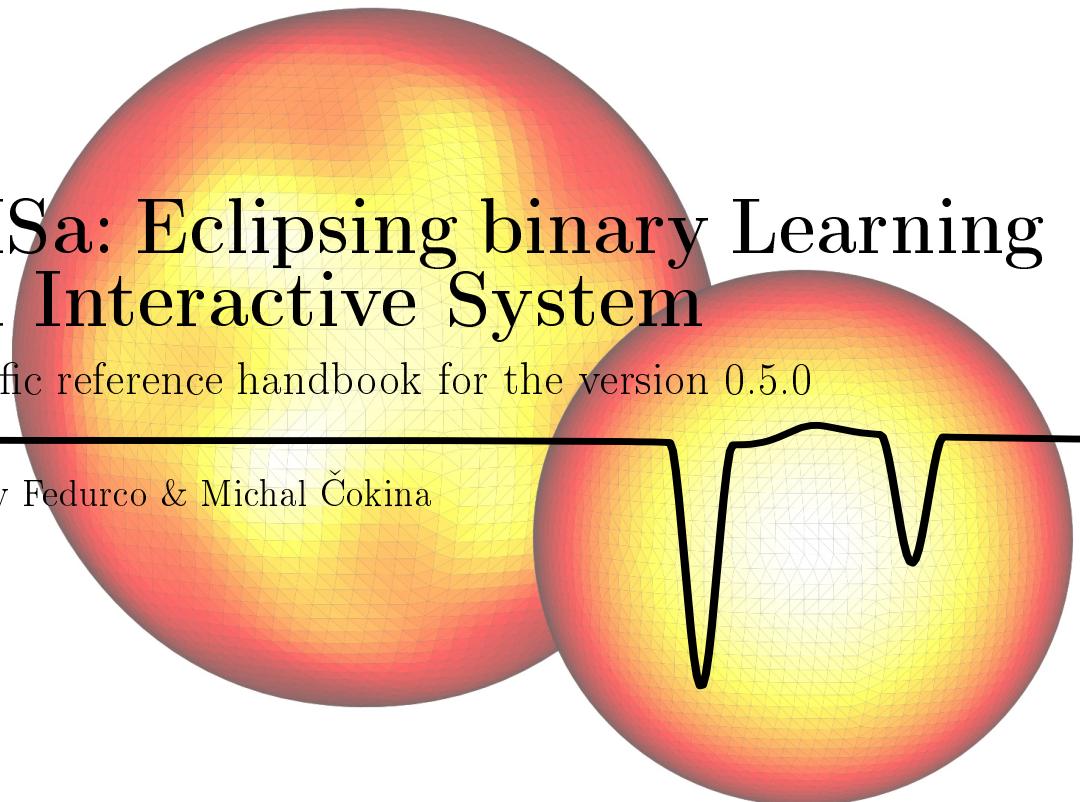
Department of Theoretical Physics and Astrophysics

# ELISa: Eclipsing binary Learning and Interactive System

Scientific reference handbook for the version 0.5.0

---

Miroslav Fedurco & Michal Čokina



Košice 2021

## Acknowledgements

I would like to express a great deal of gratitude to my supervisor doc. Mgr. Štefan Parimucha, PhD. for the continuous support of my PhD. study, insightful comments and also for providing me with the opportunities to widen my research in many directions. A great deal of gratitude goes to my colleague and good friend Michal Čokina for the endless effort put into the co-development of this software package.

## **Abstract**

We present a new, fast and easy to use tool for modelling light and radial velocity curves of single and close binary stars with the built-in method for solving an inverse problem and the capability to model low amplitude non-radial oscillations. The main goal of ELISa (Eclipsing binary Learning and Interactive System) is to provide an acceptable compromise between computational speed and precision. The package is entirely written in Python programming language in a modular fashion, making it easy to install, modify, and run various operating systems. ELISa implements Roche geometry and triangulation process to model a surface of the close binaries with a tidal distortion and single-star systems with centrifugal distortion. Surface parameters of triangulated surface are treated separately to increase the numerical precision of synthetic observations. ELISa also can introduce small perturbations into the equilibrium surface to model photometric and radial velocity variations caused by stellar pulsations. Surface symmetries and approximations based on the similarity between surface geometries were used to reduce runtime during the calculation of synthetic observations. ELISa provides various optimisation methods to provide the built-in capability to determine parameters of the eclipsing binaries from photometric observations and radial velocities. Based on various benchmarks evaluating the precision and computational speed, we conclude that the ELISa maintains an acceptable level of accuracy to analyse data from ground-based and space-based observations. In conclusion, users can find the package useful for applications where higher computational speed is desired to achieve a set goal in a reasonable amount of time.

# Contents

<b>1 Motivation and outline of the problem</b>	<b>5</b>
<b>2 Introduction to the ELISa package</b>	<b>8</b>
2.1 Package modules . . . . .	8
2.2 System module . . . . .	9
2.2.1 Initializing a system in ELISa . . . . .	10
2.2.2 Position container . . . . .	10
2.3 DataSet module . . . . .	11
2.4 Observer module . . . . .	12
<b>3 Stellar surface in hydrostatic equilibrium</b>	<b>14</b>
3.1 Equipotential surfaces . . . . .	14
3.1.1 Single stars . . . . .	14
3.1.2 Binary stars . . . . .	15
3.2 Generating surface points . . . . .	17
3.2.1 Generating angular coordinates . . . . .	17
3.2.2 Generating radial coordinates . . . . .	20
3.3 Triangulation of surface points . . . . .	20
3.3.1 Single and detached binary systems . . . . .	21
3.3.2 Over-contact binaries . . . . .	21
3.4 Discretization and triangulation of surfaces with spots . . . . .	22
3.5 Properties of the surface discretization . . . . .	24
3.5.1 Homogeneity of the surface discretization . . . . .	24
3.5.2 Estimation of the horizon . . . . .	26
<b>4 Properties of the equilibrium stellar surfaces</b>	<b>27</b>
4.1 Velocities . . . . .	27
4.2 Surface acceleration . . . . .	29
4.3 Effective temperature . . . . .	30
4.3.1 Gravity darkening . . . . .	30
4.3.2 Reflection effect . . . . .	32
4.4 Limb-darkening . . . . .	35
4.5 Normal radiance . . . . .	37
<b>5 Non-radial pulsations</b>	<b>40</b>
5.1 Implementation of stellar pulsations in ELISa package . . . . .	42

5.2	Pulsations as a perturbation of an equilibrium stellar surface . . . . .	42
5.3	Kinematics of the surface points . . . . .	43
5.3.1	Displacement of the surface element on a complex plane . . . . .	43
5.3.2	Kinematic quantities of the pulsating stellar surface . . . . .	45
5.4	Perturbation of the surface temperature distribution . . . . .	46
<b>6</b>	<b>Synthetic observations</b>	<b>48</b>
6.1	Managing time series . . . . .	48
6.1.1	Reduction to a single observation . . . . .	48
6.1.2	Using phased time series . . . . .	49
6.1.3	Phase curve symmetry . . . . .	49
6.2	Orientation of the model with respect to the observer . . . . .	50
6.2.1	Light-travel time effect in the observer's reference frame . . . . .	51
6.3	Visibility of the surface elements . . . . .	52
6.3.1	Selecting the surface elements facing the observer . . . . .	52
6.3.2	Determination of the surface visibility during the eclipse . . . . .	53
6.4	Light curve integration . . . . .	55
6.5	Radial velocities . . . . .	56
6.5.1	Kinematic method . . . . .	56
6.5.2	Radiometric method . . . . .	57
<b>7</b>	<b>Integration techniques for synthetic observations</b>	<b>59</b>
7.1	Integration using a single model . . . . .	59
7.2	Synthetic observations of binaries with eccentric orbits . . . . .	60
7.2.1	Interpolation between symmetrical orbital positions . . . . .	61
7.2.2	Assumption of identical geometry on nearly symmetrical orbital positions . . . . .	63
7.2.3	Assumption of similar geometry on neighboring orbital positions . . . . .	65
7.2.4	Precision versus speed comparison . . . . .	66
<b>8</b>	<b>Inverse problem</b>	<b>68</b>
8.1	Solution to inverse problem . . . . .	68
8.1.1	Least Square Trust Region Reflective algorithm . . . . .	69
8.1.2	Markov chain Monte Carlo . . . . .	69
8.2	General guidelines for solving the inverse problem . . . . .	70
8.3	AnalyticsTask module . . . . .	71
8.4	Determination of parameters in case of eclipsing binary KIC 4851217 . . . . .	74
8.4.1	Fitting of radial velocities . . . . .	75

8.4.2 Light curve fitting . . . . .	76
<b>Appendices</b>	<b>79</b>
<b>A List of valid initial parameters combinations.</b>	<b>79</b>
<b>B Mathematical forms of the modified potential <math>\Omega</math></b>	<b>81</b>
B.1 Spherical coordinates . . . . .	81
B.2 Cylindrical coordinates . . . . .	82
B.3 Cartesian coordinates . . . . .	83
<b>C Spherical harmonics and their derivatives</b>	<b>85</b>
<b>D Performance of eccentric orbit approximations</b>	<b>86</b>
<b>E Fit parameters for AnalyticsTask module</b>	<b>87</b>
<b>F Inverse problem of KIC 4851217</b>	<b>89</b>

---

## List of commonly used variables

Label	Description	Value (if applicable)
$a$	semi-major axis	
$c$	speed of light in a vacuum	$299\,792\,458 \text{ m s}^{-1}$
$e$	eccentricity	
$F$	synchronicity	$\omega/\omega_{orb}$
$G$	gravitational constant	$6.674 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$
$q$	mass ratio	$M_2/M_1$
$r_{backward}$	backward radius	
$r_{equator}$	equatorial radius	
$r_{eq}$	equivalent radius	
$r_{forward}$	forward radius	
$r_{polar}$	polar radius	
$r_{side}$	side radius	
$\mathbb{Z}$	set of all integers	
$\nabla_{ad}$	adiabatic temperature gradient	$\frac{d \ln T}{d \ln p}$
$\omega$	angular speed of rotation	
$\omega_{orb}$	angular speed of orbital motion	
$\omega_p$	angular frequency of pulsations	

## 1 Motivation and outline of the problem

In recent years, technological advancement in observational techniques substantially increased the amount of available data gathered from ground-based observations. Especially automated ground-based surveys such as ASAS (Pojmanski et al., 2005), OGLE (Udalski et al., 2015), PanSTARRS (Chambers et al., 2016), SuperWASP (Pollacco et al., 2006) and others contributed to the total volume of the observational data. Even the general public, namely astronomy enthusiasts, have access to the equipment capable of producing high-quality photometric observations. Further increase in the volume of observations is expected with forthcoming surveys such as Vera Rubin Observatory (LSST) (Ivezic et al., 2019). An important part of the current observational capabilities is also space-based observatories such as GAIA (Gaia Collaboration et al., 2016), Kepler (Borucki, 2016) or TESS (Ricker et al., 2015) (and eventually PLATO). These observations are a “gold mine” of publicly available high-quality photometric data with excellent precision and duration. Light curves of the stellar objects are the main type of the gathered data, and a significant portion of them belong to the light curves of eclipsing binaries.

Photometric observations represent the principal source of knowledge about eclipsing binaries. Increasing accuracy of observations needs to be accompanied by the implementation of relevant modelling capabilities. Such a tool would enable us to find system parameters from observations and fully utilise the observations’ precision. The volume of the light curves produced by the mentioned instruments means that only a minuscule fraction of observations can be processed using a standard customised object-to-object approach. Hence, an efficient and autonomous solution to the inverse problem needs to be developed to increase the number of eclipsing binaries with derived parameters.

Numerous software packages focused on the modelling of eclipsing binaries have been developed. A significant fraction of presently available modelling tools such as ELC (Orosz and Hauschildt, 2000) or Phoebe v1 (Prsa and Zwitter, 2005) is inspired by the Wilson-Devinney code (Wilson and Devinney, 1971). The main shortcoming of the Wilson-Devinney code is the implementation of trapezoidal discretisation, where surface properties are calculated on the surface represented by a discrete set of points. Consequently, WD produces unwanted artefacts in the synthetic observations, mainly during eclipses. Additionally, the calculation of surface areas corresponding to the surface point also raises few concerns.

Other modelling tools may use different approaches, such as modelling binary components as triaxial ellipsoids as done in the ELLC package Maxted (2016). However,

## 1 MOTIVATION AND OUTLINE OF THE PROBLEM

---

the triaxial ellipsoid approximation cannot be used to model close eclipsing binaries, where the shape of the components departs significantly from the shape of the triaxial ellipsoid.

Although computationally fast, previously mentioned packages are limited in terms of precision and modelling capabilities. On the other side, probably the most advanced tool for modelling eclipsing binaries available is Phoebe v2 (Prsa et al., 2016), which contains a wide variety of functionalities and achieve exceptional precision. However, these features come at the price of relatively slower runtime speed (Čokina et al., 2021).

As a result, one of the main goals of this handbook is to introduce a reader to a software package ELISa. ELISa has been designed as an acceptable compromise between precision and computational speed. The main application of the ELISa package are tasks where a large number of synthetic observations have to be evaluated in a reasonable amount of time for machine learning application or for solving an inverse problem, i.e. inferring parameters of eclipsing binaries from observational data. The package provides the built-in capability to solve an inverse problem using various optimisation methods. The package has been developed using a high-level programming language, Python, commonly used in astronomy. Therefore, user scripts are written natively in Python, enabling the utilisation of a vast library of tools. Finally, the package is written with standard coding practices in mind, which should provide a smooth learning process for users with basic programming skills in Python.

Current research in asteroseismology focuses on a research of the pulsating variables as members of close eclipsing binaries (e.g. Sekaran et al. (2021); Stassun et al. (2021); Rappaport et al. (2021); etc.). The presence of the companions enables us to infer the absolute parameters of the pulsator with high precision and thus significantly improve the starting point of the asteroseismological analysis that will allow us to infer internal structure profiles such as internal mixing or rotation profiles. This analysis is performed with the state of the art modelling tools such as MESA (Modules for Experiments in Stellar Astrophysics) (Paxton et al., 2018) and GYRE (Townsend and Teitler, 2013) that utilise advanced one-dimensional models of stellar structure. However, components of close binary systems are often deformed from the spherical shape by tidal and centrifugal forces. Deformation of the surface significantly impacts how the pulsations manifest themselves in the photometric observations. Therefore, a suitable tool is necessary to model stellar oscillations on a tidally distorted star capable of dealing with eclipses, ellipsoidal variations and reflection effect on the resulting shape of the oscillations.

Moreover, the distorted shape of the component also causes a misalignment of the pulsation mode axis with the rotational axis. The pulsation modes are often tidally

## 1 MOTIVATION AND OUTLINE OF THE PROBLEM

---

locked relative to the position of the companion star. Therefore, the ELISa provides a capability to model tidally locked modes with an arbitrary orientation of the pulsation mode axis. Tidally distorted surfaces defined by a standard theory of modelling close binary surfaces (Wilson, 1979) were used as a quasi-equilibrium model that non-radial modes can subsequently perturb.

This handbook discusses in detail the functionality of the ELISa package. Section 2 outlines a basic structure of the package. Afterwards, Sections 3 and 4 describe the process of construction of the equilibrium models of single and binary systems. Section 5 then describes the modelling of non-radial oscillations on the equilibrium surface models defined in the previous sections. A complete model of a binary system is then used to produce synthetic observations, and the process of the calculation of the observables is described in Section 6.

The numerical model can achieve significant improvements to the computational speed by utilising approximations used during the process of integration of the synthetic observations outlined in Section 7 along with their evaluation in terms of their speed and precision. Finally, the module implementing binary system modelling capabilities to solve an inverse problem using radial velocity and photometric data is described in detail in Section 8.

## 2 Introduction to the ELISa package

ELISa (Eclipsing binary Learning and Interactive System) (Čokina et al., 2021) is a multi-platform software package developed to generate synthetic observations of single and binary stars containing surface features as spots and pulsations. The package is written in Python programming language, and it is freely available at <https://github.com/mikecokina/elisa>, or it can be installed with the simple command `pip install elisa`.

The most important features of the ELISa package are:

- ELISa is a pure Python software package that makes it easy to install and modify,
- ELISa is a cross-platform package tested and developed on Windows and Linux operating systems,
- utilizing a modern approach to the modelling of the stellar surfaces with the emphasis on computational speed while maintaining a sufficient level of precision to process a modern ground-based and space-based observations,
- integrated fitting of the light curve and the radial velocity curves based on the Least Square Trust Region Reflective algorithm and the Markov-Chain Monte Carlo.

Due to a relatively wide range of capabilities, the introduction of a reader to the package structure is required. Therefore, the rest of this chapter is dedicated to explaining the philosophy behind the design of modules. The accompanying set of Jupyter tutorials is available on the GitHub page of the project.

### 2.1 Package modules

ELISa combines a wide range of functionalities ranging from constructing a model and production of synthetic observations to managing user's observations and solving an inverse problem. Therefore, it was necessary to partition the package into separate modules performing specialized tasks. As a result, ELISa consists of 4 different modules:

- **System**: module responsible for the construction of a complete model of the observed object.
- **Observer**: module responsible for producing synthetic observations.
- **Dataset**: unified data frame for storing observations.

## 2 INTRODUCTION TO THE ELISA PACKAGE

---

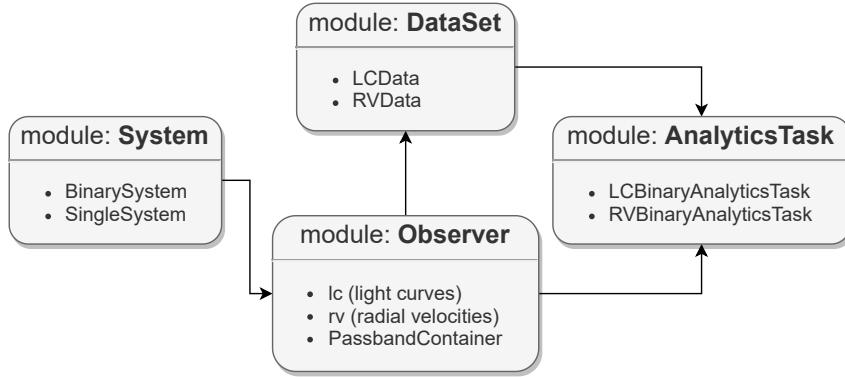


Fig. 1: This figure shows a simplified diagram depicting four principal modules within the Elisa and their major components.

- **AnalyticsTask**: module dedicated to solving an inverse problem.

Diagram in Figure 1 helps to visualize the general structure of the package. The arrows indicate the dependencies between the different modules. The overall structure of the **System**, **Dataset** and **Observer** modules is discussed in the following sections. Approach to the solution of the inverse problem implemented in the **AnalyticsTask** module is discussed later in Section 8.

### 2.2 System module

An essential role of the **System** module is the initialization process and management of the object properties. Figure 2 depicts the basic structure of this module. Astrophysical phenomena modelled in ELISa consist of the three main groups:

- *System* - abstract class of objects responsible for the overall system hierarchy and the interaction of its components. Currently supported types of systems are single and binary star systems.
- *Body* - abstract class representing the component(s) of the system. The primary task of this class is the management of all component's physical properties (mass, effective temperature, gravity darkening factor, etc.).
- *Features* - inhomogeneities present on the surface of the system components. Currently, ELISa is capable of modelling temperature spots and pulsations.

Currently, the ELISa supports the modelling of single star systems and binary systems. A complete description of the physics used to model both types of objects is described in detail in Sections 3 and 4.

## 2 INTRODUCTION TO THE ELISA PACKAGE

---

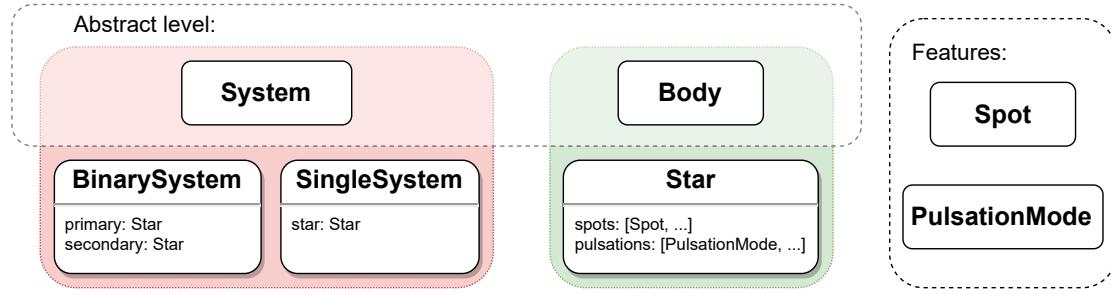


Fig. 2: Description of the overall structure of the System module. Objects derived from the abstract class `System` contain instances of the abstract class `Body` as their components. Each component can additionally contain an arbitrary number of surface features such as spots or pulsation modes.

### 2.2.1 Initializing a system in ELISa

Initializing a system in ELISa requires providing certain combinations of parameters that can unambiguously define a system. Therefore, ELISa allows passing only a specific combination of input parameters to initialize a single or a binary system successfully. Appendix A lists available combinations of input parameters.

User can initialize an `SingleSystem` instance using two possible combinations of initial parameters. Combinations differ from each other in a manner in how they define the size of the modelled star. The “standard” parameter combination establishes the size of the star by the value of the polar gravity that usually correlates with the evolution stage of the star. On the other hand, the “radius” - based parameter combination accepts the desired polar radius as a model parameter.

Instances of `BinarySystem` can be initialized using “standard” or “community” combination of parameters. The standard format of the input parameters requires providing component’s masses. On the other hand, the “community” format offers the option to provide necessary constraints with the mass ratio and semi-major axis of the system instead of the component’s masses. Community format is useful to define dimensionless systems scaled with the reasonable value of the semi-major axis. Defining the system in such a manner is very useful when only limited (mainly photometric) observational data are available. Thus, the absolute parameters of the system cannot be derived.

### 2.2.2 Position container

Surface properties of the single and binary star components are in general time-dependent. Therefore, a structure called `PositionContainer` is used to manage complete models of the system at various moments corresponding to desired times of observation. The `PositionContainer` is constructed based on a corresponding `System`

## 2 INTRODUCTION TO THE ELISA PACKAGE

---

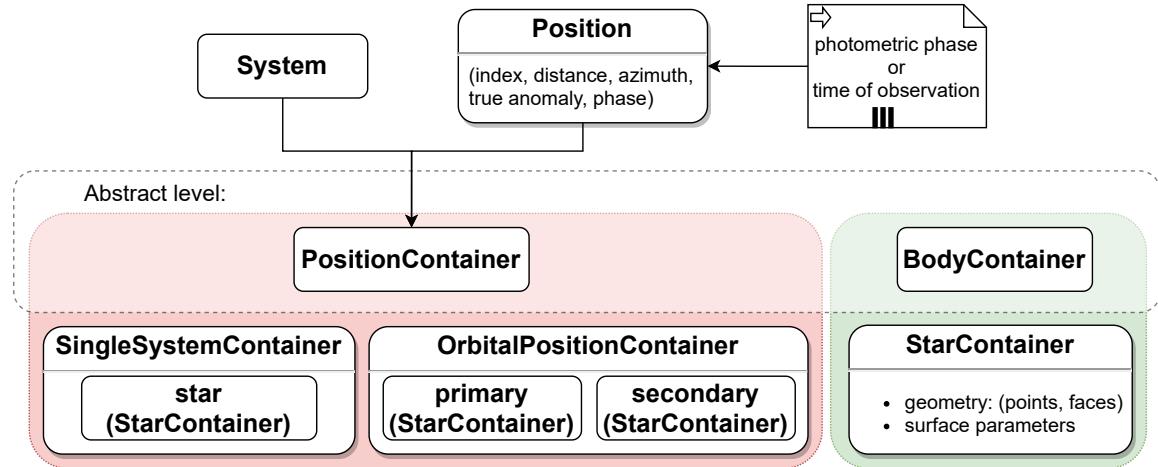


Fig. 3: The basic structure of the `PositionContainer` used to build and store a complete model of a modelled system. A full model of a system is built based on the system parameters stored in a `System` instance and position of the system.

instance defined in Section 2.2.1 and a precise moment in which to built the model is defined by the `Position` described in Section 6.2. The parameters connected with system components such as surface points mesh, faces, surface temperature distribution, etc. are stored within the `PositionContainer` in form of a `StarContainer` defined for each component. Figure 3 describes a structure of a `PositionContainer` instances for single and binary systems.

### 2.3 DataSet module

The principal task of the ELISa package is to be able to model synthetic observations and analyze existing real-world observational data. The ELISa itself can provide such observations, or real-world observations from various instruments have to be imported from outside of the framework. Therefore, we had developed a dedicated `DataSet` object to transform various input data formats to the form suitable for further processing (e.g. converting magnitudes to relative fluxes). The current version of the module supports importing the observations from any file type with the data ordered in columns. Currently, the module supports handling photometric and radial velocity observations either in the time or phase domain (see Figure 4). The module also supports operations such as JD to phase conversion (and vice versa) and the ability to visualize the stored data in the form of a 2D plot.

## 2 INTRODUCTION TO THE ELISA PACKAGE

---

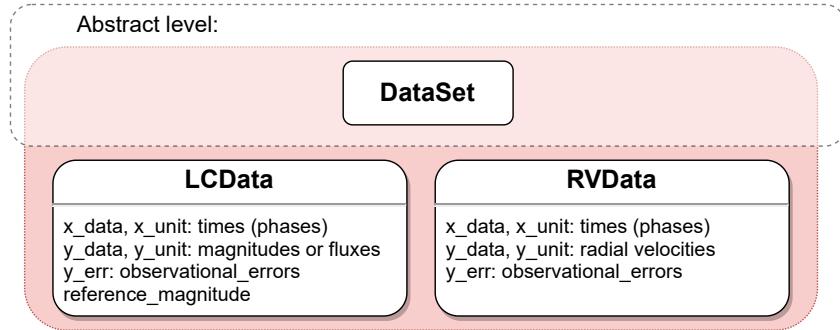


Fig. 4: Basic parameters of the `LCData` and `RVData` objects responsible for handling of photometric and radial velocity measurements.

### 2.4 Observer module

The observer module enables the user to generate synthetic observations of systems defined in 2.2. The observer module contains all the necessary information to perform the synthetic observation at an arbitrary moment. Parameters related to the modelled object are contained in the provided `System` instance built a priori. The observer-related parameters are also provided during the initialization of the `Observer` class instance. One of them is a photometric passband, used to integrate synthetic light curves observed in commonly used photometric filters. A list of the passbands supported by ELISa is accessible in configuration parameter `settings.PASSBANDS`.

After initialization, the `Observer` instance can produce two types of synthetic observations: light curves and radial velocity curves calculated by corresponding functions `lc()` and `rv()` as illustrated in Figure 5. One of the main tasks of this module is to classify the modelled system and assign the most efficient method for integration of the synthetic observations. This process is discussed in detail in Section 7.

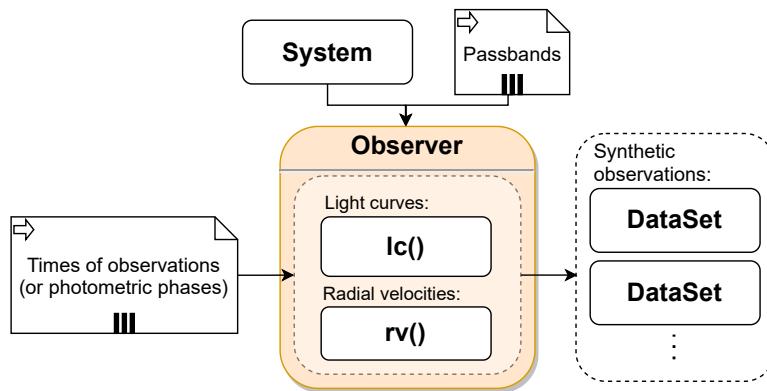


Fig. 5: The figure displays a general overview of the structure and functionality of the `Observer` class. An instance of `System` and passband filters are provided for the `Observer` class to perform observation at moments defined by the specific observational time stamps or phases.

### 3 Stellar surface in hydrostatic equilibrium

#### 3.1 Equipotential surfaces

The shape of the stellar surfaces can vary significantly from the almost spherical shape of a slowly rotating single star to the heavily distorted dumbbell shape of a component in over-contact binary systems. However, every star tends to form an equilibrium surface with equalized pressures of the stellar matter on the surface. In such a case, the surface creates an equipotential surface (Kopal, 1959), i.e. a surface on which given general potential function  $\Psi$  maintains the same value. A potential function suitable for constructing the stellar surface should include the effects of all forces acting on a surface element of the star, such as tidal and centrifugal forces. The potential functions used to model stellar surfaces within this package were derived using the following assumptions:

1. Stars are regarded as point masses with negligible mass stored in their envelopes.
2. An instantaneous force field defines the star's surface, i.e. the effect of non-radial oscillations is negligible.
3. Star rotates as a rigid body without any differential rotation.

##### 3.1.1 Single stars

We must take the only gravitational force into account in the most simple case of a single non-rotating star with the mass  $M$ . Therefore, the star has a spherical shape and potential function has a straightforward form in spherical coordinates:

$$\Psi_{sph} = -\frac{GM}{r}. \quad (1)$$

Due to the imparted angular momentum, e.g. during a star formation, a vast majority of the stars rotate, which will cause deformation of the stellar surface in the equatorial region. The deformation is caused by the additional term in the potential function taking into account a centrifugal force:

$$\Psi_{rot} = -\frac{GM}{r} - \frac{1}{2}(\omega r \sin \vartheta)^2, \quad (2)$$

where  $\omega$  is an angular speed of the rotation. The surface described by the Equation 2 is still axially symmetric around its axis of rotation which we can utilize during the modelling of the surface. The example of the stellar surfaces generated using the Equation 2 is displayed in Figure 6, which demonstrates the shapes of the solar model with rotational periods of 30 d and 0.22 d respectively.

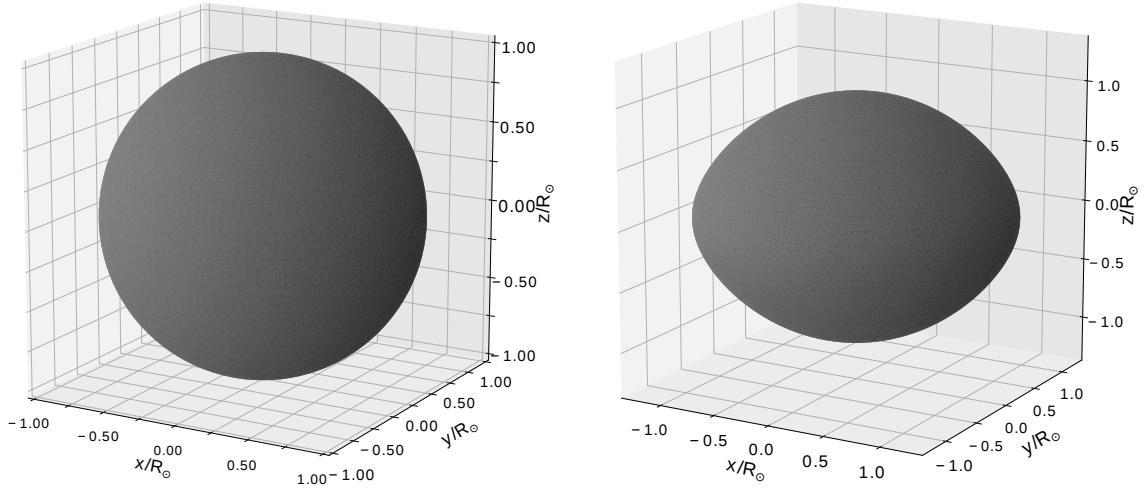


Fig. 6: Shapes of the stellar surfaces of star models using solar parameters in Equation 2, except for the rotational period. The model on the left figure rotates with 30 d period and 0.22 d (5.3 h) rotational period on the right side. The second model demonstrates a significant deformation in the equatorial regions due to the substantial centrifugal forces.

### 3.1.2 Binary stars

Modelling of binary system components follows the same principles as described in Section 3.1. In this case, we must consider the gravitational forces of both components and centrifugal forces of the orbital and rotational motion. Additional forces will cause deformation in the direction of the binary companion, which grows more pronounced with the decreasing relative distance between the components. By far, the most common approach to the modelling of close binaries was described by Wilson (1979). This method is suitable for modelling binary systems with eccentric orbit and asynchronous rotation where the component's axis of rotation is perpendicular to the orbital plane. The equipotential function describing the surface can be, in this case, described in the form of the *modified Kopal potential* (from now on referred to as modified potential) in the co-rotating reference frame centred to the primary component:

$$\Omega_1(\varrho; q, d) = \frac{1}{\varrho} + q \left[ \frac{1}{\sqrt{d^2 - 2d\varrho\lambda + \varrho^2}} - \frac{\varrho\lambda}{d^2} \right] + \frac{1}{2} F_1^2(q+1)\varrho^2(1-\nu^2), \quad (3)$$

where  $q = M_2/M_1$  is a mass ratio of the components,  $d$  is an instantaneous distance between the components normalized by semi-major axis (also known as SMA units)  $a$ .

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

The parameters  $\varrho$ ,  $\lambda$ ,  $\nu$  are substitutions for the spherical coordinates:

$$\begin{aligned}\varrho &= r/a, \\ \lambda &= \cos\varphi \sin\vartheta, \\ \nu &= \cos\vartheta.\end{aligned}\tag{4}$$

Finally, the parameter  $F_1$  represents the *synchronicity parameter* of the primary component, which is generally defined as a ratio between the rotational angular speed  $\omega$  of a given component and the orbital angular speed  $\omega_{orb}$ :

$$F = \frac{\omega}{\omega_{orb}}.\tag{5}$$

The components of the close eclipsing binaries with circular orbits are almost always fully synchronized ( $F = 1$ ) due to the very rapid synchronization timescale compared to the component's nuclear time scale (Zahn, 1975). Synchronization occurs rapidly in the case of eccentric orbits as well. However, due to the strong dependence of the magnitude of the tidal forces on distance, the final rotation is periastron-synchronized with synchronicity parameter given by Hut (1981):

$$F = \frac{(1+e)^2}{(1-e^2)^{\frac{3}{2}}}.\tag{6}$$

Equation 3 can be used to calculate equipotential  $\Omega'$  in the reference frame of the secondary component, if inverse mass ratio  $q' = 1/q$  is used instead of  $q$ . However, this modified potential needs to be modified to be consistent with the modified potential calculated in the reference frame of the primary component using the following transformation:

$$\Omega(\varrho; q, d) = q\Omega'(\varrho; q', d) - \frac{1}{2}(q-1),\tag{7}$$

which leads to the final form of the modified potential for the equipotential function, which we will use to model surfaces of the secondary components:

$$\Omega_2(\varrho; q, d) = \frac{q}{\varrho} + \frac{1}{\sqrt{d^2 - 2d\varrho\lambda + \varrho^2}} - \frac{\varrho\lambda}{d^2} + \frac{1}{2}F_2^2(q+1)\varrho^2(1-\nu^2) + \frac{1}{2}(1-q).\tag{8}$$

We can transform the modified potential back to the regular potential  $\Psi$ , which will be used later during the calculation of the surface acceleration:

$$\Psi = -\frac{GM_1}{a} \left( \Omega + \frac{1}{2} \frac{q^2}{q+1} \right).\tag{9}$$

Equations 3 and 8 can be solved implicitly (see Section 3.2) for the parameter  $\varrho$  and the resulting shapes of the surfaces can be divided into the three major groups that reflect the degree of filling of the component's Roche lobes:

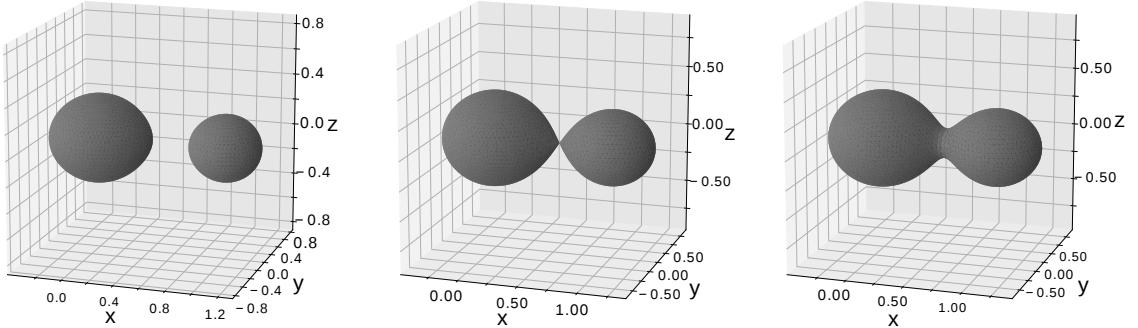


Fig. 7: Figure illustrating different morphologies of close binary stars based on the fulfilment of the component's Roche lobes. Examples of detached, contact and over-contact systems are displayed in order.

- *Detached* - Components are physically detached, and the effective potentials of both components are below the *critical potential*, lower than the potential at Lagrangian point  $L_1$ .
- *Semi-detached* - Components are physically detached, and the effective potentials of both components are below the *critical potential*, lower than the potential at Lagrangian point  $L_1$ .
- *Over-contact* - The components overflow their respective Roche lobes, and the components are physically connected with the structure, which we will refer to as the *neck*.

Examples of the stellar surfaces from each group are displayed in the Figure 7.

### 3.2 Generating surface points

The initial step in generating a stellar surface is producing a discrete set of surface points, creating a mesh that will define a surface geometry. This set of points should be covering the whole stellar surface homogeneously and with sufficient fidelity to not produce any unwanted artefacts during the curve modelling. The deciding factor governing the density of surface points is, in our case, the *discretization factor* set automatically or defined by the user. The discretization factor represents the desired angular separation of the surface points in degrees.

#### 3.2.1 Generating angular coordinates

Generating surface points is done in two steps. Initially, the angular coordinates  $\varphi$  and  $\vartheta$  are generated using modified forms of a trapezoidal discretization. These variants vary

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

for the different morphologies to accommodate their specifics and aim to maintain relatively homogeneous sampling. However, in every case, the sampling of the surface is reduced to the symmetrically unique part that significantly reduces the computational time necessary to generate a stellar surface. In general, the discretization method follows the trapezoidal method, which can be subsequently improved using corrections listed below. ELISa implements both trapezoidal and improved trapezoidal method that user can switch between using configuration parameter `settings.MESH_GENERATOR` with possible values: 'trapezoidal', 'improved\_trapezoidal' and 'auto'. Value 'auto' will evaluate the deformation of the model by comparing the relative difference between the smallest and the largest pre-calculated stellar radii with a threshold value stored in parameter

`settings.DEFORMATION_TOL`. If the surface exceeds the preset deformation tolerance, the improved trapezoidal method is implemented.

#### Single systems

In the case of single star surfaces, axial symmetry around the rotational axis can generate points for a given set of latitudes  $\vartheta$  which will be rotated around the angular coordinate  $\varphi$  to form an equidistantly spaced ring of longitude points. Due to the equatorial bulge in spherically squashed stars, the sampling of  $\vartheta$  can not be linear. That would cause sparse sampling of points near the equator, which would degrade the homogeneity of the surface discretization. This issue can be solved using the following correction to an equidistantly sampled coordinate  $\vartheta_{linear}$ :

$$\vartheta = \vartheta_{linear} + \arctan \frac{(r_{equator} - r_{polar}) \tan \vartheta}{r_{polar} + r_{equator} \tan^2 \vartheta}, \quad (10)$$

where  $r_{equator}$  and  $r_{polar}$  are equatorial and polar radii of the star already calculated during the initialization of the system. The result of this process is illustrated in Figure 8a.

#### Detached binary systems

Surface discretization of binary systems was separated into different approaches for detached and over-contact binaries due to significant changes in their respective geometries. However, in both cases, the planar symmetries of the equipotential surface were utilized to reduce the number of evaluated surface points to one-quarter of the total number of surface points. Subsequently, the process mirrored the generated points to fill the remaining three quadrants of the surface.

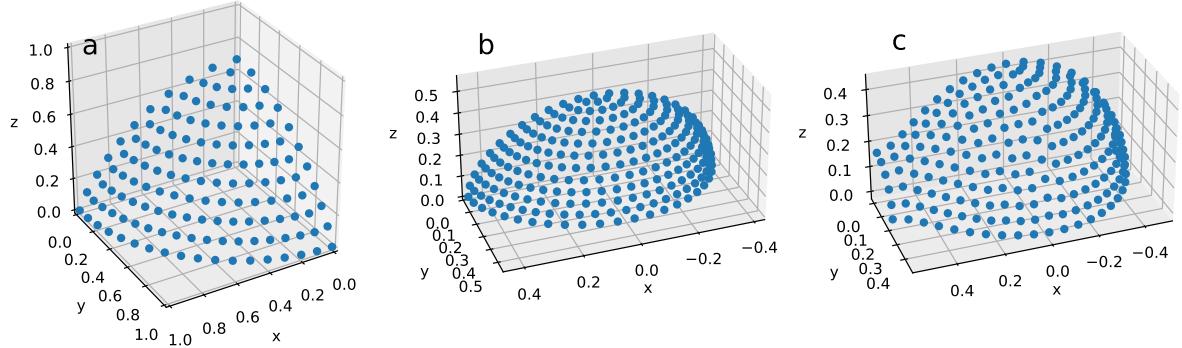


Fig. 8: Surface discretization of a single system (a), detached (b) and over-contact binary component (c). Points do not cover the whole surface of the stars due to the utilization of surface symmetries.

Detached system surfaces were generated using a process similar to the trapezoidal method utilized in the original Wilson-Devinney code (WD) (Wilson and Devinney, 1971) used on a unique one-quarter of the surface. However, in cases of highly deformed close binaries, this method will sample the surface unevenly with the considerable differences between surface element sizes (see Section 3.5). However, we can substantially improve the homogeneity of the sampling. This goal can be achieved by the introduction of the following adjustments to the angular coordinates generated by the original trapezoidal discretization  $\varphi_t$  and  $\vartheta_t$ :

$$\begin{aligned} \vartheta &= \theta_t + \arctan \frac{(\bar{r}_{\text{equator}} - r_{\text{polar}}) \tan \vartheta_t}{r_{\text{polar}} + \bar{r}_{\text{equator}} \tan^2 \vartheta_t}, \\ \varphi &= \varphi_t + \arctan \frac{(r_{\text{side}} - r_{\text{forward}}) \sin \vartheta \tan \varphi_t}{r_{\text{side}} + r_{\text{forward}} \tan^2 \varphi_t}; \quad \varphi_t \in \left(0, \frac{\pi}{2}\right), \\ \varphi &= \varphi_t + \arctan \frac{(r_{\text{side}} - r_{\text{backward}}) \sin \vartheta \tan \varphi_t}{r_{\text{side}} + r_{\text{backward}} \tan^2 \varphi_t}; \quad \varphi_t \in \left(\frac{\pi}{2}, \pi\right), \end{aligned} \quad (11)$$

where  $\bar{r}_{\text{equator}}$  is an “average” equatorial radius estimated as the mean of forward, side and backward radii  $r_{\text{forward}}$ ,  $r_{\text{side}}$ ,  $r_{\text{backward}}$ . The example of the discretization for the detached binary surface is Figure 8b.

### Over-contact binary systems

The previously described method is not suitable for the discretisation of over-contact surfaces due to the presence of the neck, where even corrected trapezoidal method breaks down. Therefore, the over-contact surface is split into the surface facing the companion star, i.e. the near-side and the surface facing away from the companion, i.e. the far-side. As a result, we can generate the angular coordinates for a far-side surface in the same manner as in the detached systems. On the other side, we used a

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

discretisation method utilising cylindrical symmetry along an  $x$ -axis for the near-side part of the surface containing the neck. The discretisation consists of concentric rings of points around  $x$ -axis. The rings were separated equidistantly along the neck, defined as a 15th-degree polynomial fit of the surface points on the  $\varphi = 0$  meridian. The cylindrical angular coordinate  $\varphi_{c,linear}$  measured from the direction of the cartesian  $z$  coordinate, where  $z_c \equiv x$  is then generated equidistantly along with the generated concentric rings. Finally, the  $\varphi_{c,linear}$  is then corrected for obliqueness using the following correction:

$$\varphi_c = \varphi_{c,linear} + \arctan \frac{(\delta_r - 1) \tan \vartheta}{1 + \delta_r \tan^2 \vartheta}; \quad \delta_r = \frac{r_{side}}{r_{polar}}. \quad (12)$$

Resulting surface discretization for over-contact surface is displayed in Figure 8c.

#### 3.2.2 Generating radial coordinates

In the second step, each combination of angular coordinates  $\varphi, \vartheta$  is solved for the corresponding  $\varrho$  using implicit equations for modified potential (see Equation 2 and appendix B). Since the number of surface points to be solved is in the range of thousands, the sequential point-by-point approach would require a substantial amount of computational time. Instead, we decided to solve all surface points utilizing a parallel approach where we solved Equations 3 and 8 using  $n$ -dimensional vectors  $\varphi, \vartheta$  and  $\varrho$ . Straightforward but effective Newton-Raphson method (Kantorovich, 1948) was used to iteratively improve the radial coordinate estimates of the potential function using the following formula:

$$6\varrho_{i+1} = \varrho_i - \frac{\Omega(\varrho_i)}{\Omega'(\varrho_i)}; \quad \Omega' = \frac{\partial \Omega}{\partial \varrho}. \quad (13)$$

The iterative process was performed until the desired absolute precision of  $10^{-10}$  in SMA units was reached for each component of  $\varrho$ . The Newton method always converges for the stellar surfaces modelled in ELISa since our potential functions are strictly monotonic on the desired interval of surface potentials.

### 3.3 Triangulation of surface points

One of the most significant differences between the original WD code and an ELISa is surface triangulation that tessellates the surfaces into a set of triangles that cover the surface without any gaps and overlaps. Since the general shape of the surface depends on the system morphology, the triangulation process differs slightly between single systems, detached and over-contact systems.

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

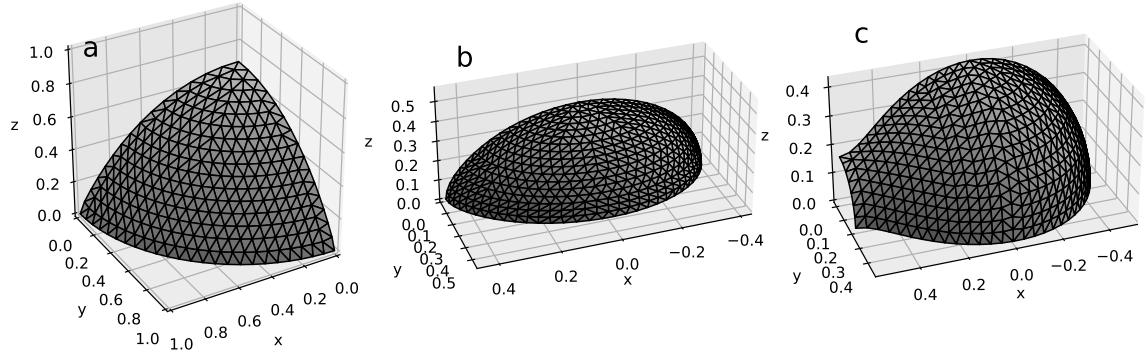


Fig. 9: Surface triangulation of symmetrical parts of a single system (a), detached (b) and over-contact binary component (c).

#### 3.3.1 Single and detached binary systems

Surfaces of components in single and detached binaries are, from a geometrical perspective, convex objects in three-dimensional space. Probably one of the most effective triangulation methods for convex objects is the Delaunay triangulation algorithm (Delaunay, 1934). The Delaunay triangulation in two dimensions is based on the criterion of empty circumcircle described to three points of the dataset belonging to the same triangle. The two-dimensional version of this algorithm can be adapted in three dimensions where the circumsphere for the given tetrahedron should not contain any additional points. The desired surface is then obtained as the convex hull of such triangulation. Therefore, we can use the method directly to describe the surface of the detached binary component since they always form a convex surface.

Additionally, we can reduce the size of the triangulated surface without surface inhomogeneities to its symmetrical component (see Figure 9a, b), which is still a convex object. Finally, triangulation can be extended to the rest of the surface.

#### 3.3.2 Over-contact binaries

The surface of over-contact binaries differs significantly from detached binary system counterparts. Components of over-contact binaries are physically connected with the so-called "neck", which is not a convex object. Thus we cannot find the surface of the component with the Delaunay triangulation alone. There are several numerical methods capable of discretization of concave surfaces, such as the Poisson surface reconstruction (Kazhdan et al., 2006) or marching method described by (Hartmann, 1998). These methods are implemented in various, primarily graphic software libraries. However, we adapted the Delaunay triangulation to deal with the over-contact binary surfaces to

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

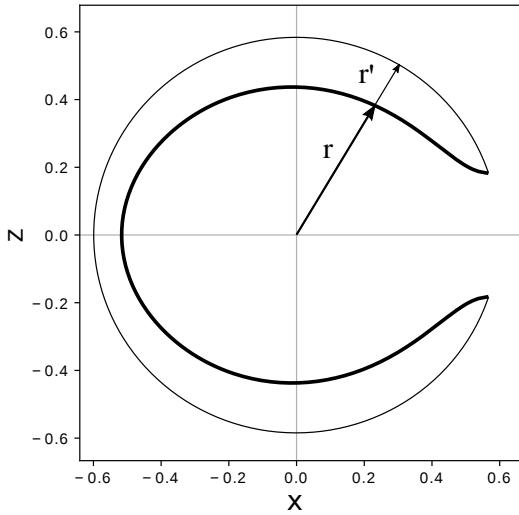


Fig. 10: This figure visualizes  $xz$  cross-section of the transformation from over contact component surface (thick inner curve) to convex shape suitable for Delaunay triangulation (thin outer curve). Transformations defined by the Equation 14 is illustrated by position vector  $\mathbf{r}$  and its transformed counterparts  $\mathbf{r}'$ .

utilize its efficiency.

The main idea behind our adaptation of the Delaunay triangulation algorithm to over contact surfaces is a suitable transformation of the original surface points into a convex object while maintaining relative positions of transformed surface points. Subsequently, the standard Delaunay triangulation can be used to triangulate the transformed points. Finally, the resulting simplices (i.e. indices of triangle vertices) can be applied to the original set of surface points.

In our treatment, surface points  $\mathbf{r}$  are transformed into a sphere with a radius equal to the distance of the component's neck to the centre of mass of the star  $x_{neck}$  using the following transformation:

$$\mathbf{r}' = x_{neck} \frac{\mathbf{r}}{|\mathbf{r}|}. \quad (14)$$

Spherical transformation described in Equation 14 is visualized in Figure 10 and the resulting surface is located in Figure 9c.

#### 3.4 Discretization and triangulation of surfaces with spots

We have devised a different discretization method for the points inside the spots to ensure a clear boundary between the spot and the surrounding surface. Due to the circular shape of spots modelled in ELISa, we decided to generate spherical coordinates of the surface spots symmetrically around the centre of the spot defined by spherical coordinates  $\varphi_{spot}$ ,  $\vartheta_{spot}$ . Therefore, spot points are generated on concentric circles with

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

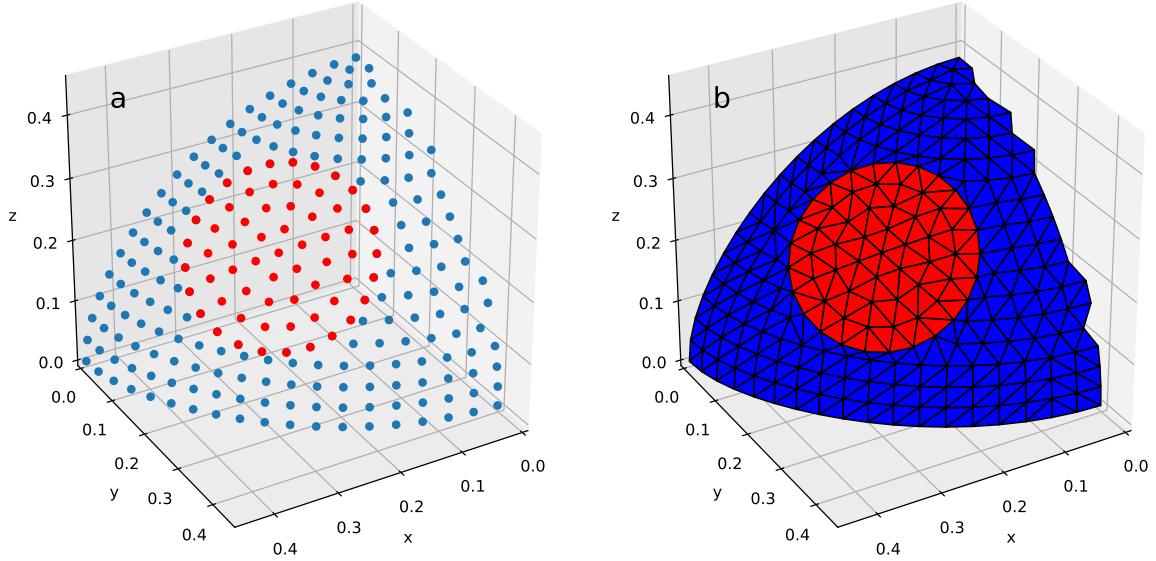


Fig. 11: A surface mesh of a binary system component with a single spot (a) and the resulting triangulation (b). Circular symmetry of a discretization enables to maintain a well-defined edge between a spot and the rest of the surface.

radii up to the angular radius of the spot  $r_{spot}$  in increments close to the discretization factor (see Figure 11a). Afterwards, the radial coordinates of spot points are generated as in Section 3.2.2. Additionally, star points (or points from the previously defined spot) located underneath the currently modelled spot are removed from the model to smooth out the surface coverage of points.

Nevertheless, the described discretization method has its downsides, namely the deformation of the spot boundary in case of highly deformed components from the desired circular shape, inability to model spots intersecting with the neck in case of over-contact binaries and finally, the spots breaking the surface symmetry that decreases computational speed. Therefore, further improvements to the discretization of spots are required.

Due to the inability to implement a surface symmetry in cases of surfaces containing spots, the resulting surface points of the whole star and spot are triangulated together. Finally, each surface element is assigned to the star or the spot according to its angular distance from the centre of the spot.

ELISa is also capable of modelling multiple overlapping spots. The discretization algorithm layers different spot points on top of each other where spot defined later will be on top of the previously defined spot (see Figure 12).

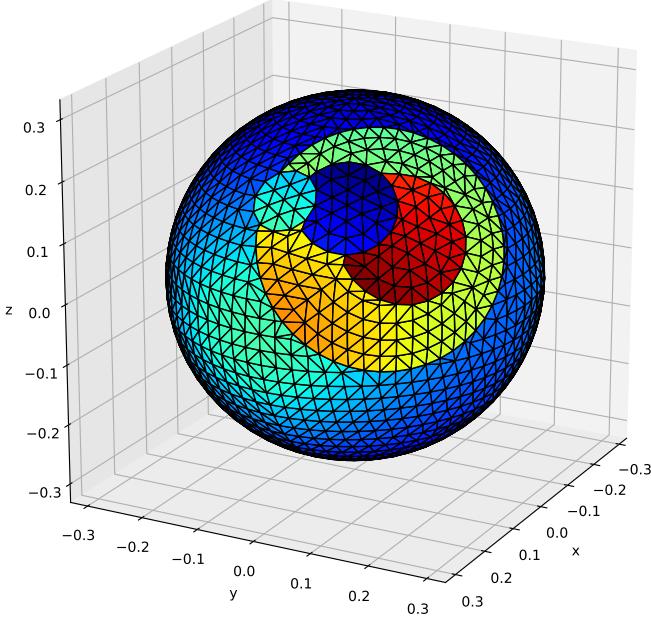


Fig. 12: Example of stacking of multiple overlapping spots. The order in which the spots were defined were from the largest to the smallest.

### 3.5 Properties of the surface discretization

Splitting the surface into the separate surface elements (i.e. surface discretization) contributes significantly to the resulting precision of the observables, such as a measured flux or radial velocity. The following sections describe some of the important properties of any given discretization method.

#### 3.5.1 Homogeneity of the surface discretization

Homogeneous distribution of the points across the surface crucially contributes to the overall precision of the model. Homogenous distribution of the surface points across the surface leads to a precise and most efficient treatment of surface parameter gradients for a given number of surface points. A very good indication of the homogeneity in the distribution of points is a distribution of surface element areas which we can use to detect sparsely or densely covered regions. To demonstrate the suitability of the improved trapezoidal method, Figure 13 illustrates distributions of surface element areas on an example of a single, detached and over-contact system visualized in Figure 9. Results show that the improved trapezoidal method is capable of improving the spacing between the points in case of surfaces distorted due to rotation or tidal forces as demonstrated in Table 1.

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

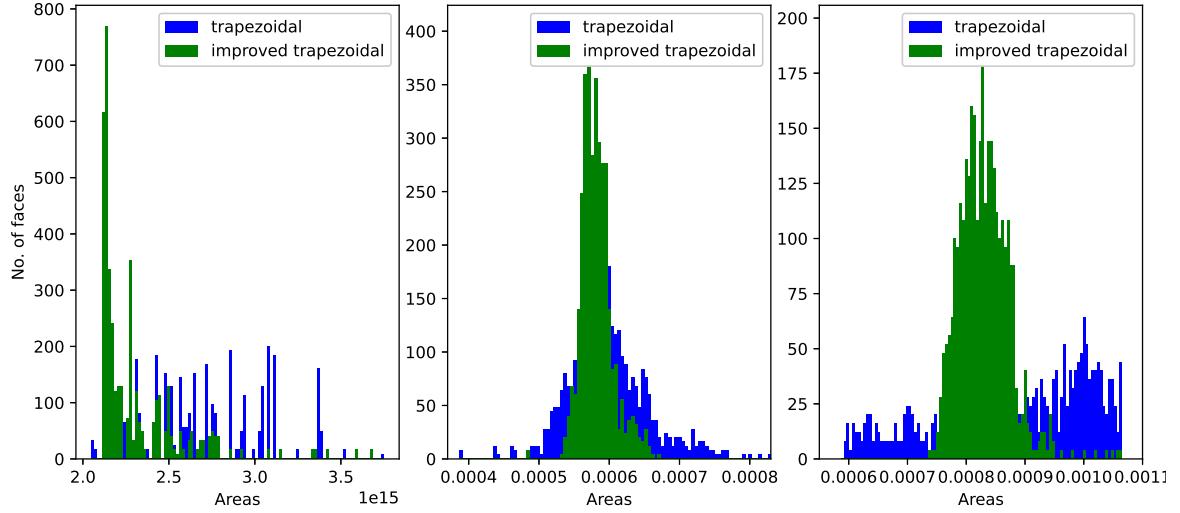


Fig. 13: Distribution of surface areas for single, detached and over-contact systems displayed in order from left to right.

Tab. 1: Relative dispersion of surface element areas (in %) for systems displayed in Figure 13 using the improved trapezoidal method.

Discretization method	trapezoidal	improved trapezoidal		
Percentile interval	(32, 68)	(1, 99)	(32, 68)	(1, 99)
Fast rotating single star	9.0	20	3.5	15
Close detached binary	3.5	25	1.5	9.0
Over-contact binary	14	110	2.5	11

### 3 STELLAR SURFACE IN HYDROSTATIC EQUILIBRIUM

---

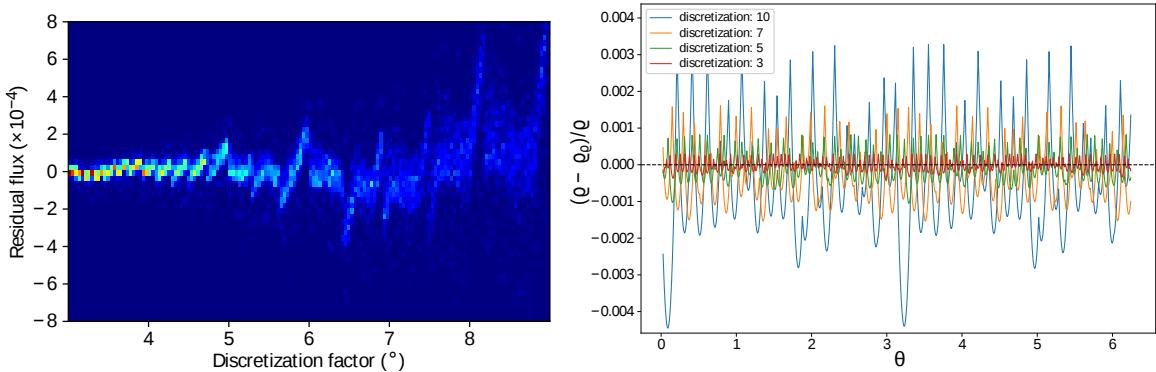


Fig. 14: Conservation of the observed flux with the changing discretization factor tested on 10 000 randomly drawn eclipsing binaries visualized in the left figure. The right figure shows the difference between the synthetic and true horizon for the detached binary star component calculated during photometric phase 0.25 and inclination 60°.

#### 3.5.2 Estimation of the horizon

The second important factor is the precision in the estimation of the stellar horizon. This factor is critically important to model the eclipses and the amount of flux in the observer's direction. The triangulated surfaces tend to underestimate the surface since only the triangle vertices are located on the true surface, and the rest of the triangle is below the true surface. This issue, if uncorrected, can lead to measurable differences between the expected and the measured synthetic flux. Additionally, this discrepancy is scaled non-linearly with the surface discretization.

The introduction of a correction factor in the following form rectifies this issue:

$$C = \sqrt{\frac{E\alpha}{\sin E\alpha}}, \quad (15)$$

where  $\alpha$  is a discretization factor, and a factor  $E$  accounts for the effect of non-equilateral triangles produced by discretization methods described above. We calibrated the value of  $E$  for each discretization method on a grid of discretization factors  $\alpha$ . The correction factor slightly expands the stellar surface and ensures that the resulting surface area of the model's surface is equal to the surface area of the true surface. We tested the conservation of the outgoing flux on a set of 10 000 randomly drawn binary systems. The conservation of the outgoing flux is demonstrated in the left side of Figure 14. The actual precision of the horizon estimation compared to the true horizon is displayed on the right side of Figure 14 for four different values of the discretization factor.

## 4 Properties of the equilibrium stellar surfaces

The stellar surfaces described in Section 3 can be used as a basis on which we can define surface quantities. Division of the stellar surface into the surface elements enables us to assign unique surface properties to each surface element and store them in `StarContainer` in a vectorized form, significantly improving the computational speed. The discretization method used in ELISa also enables utilizing surface symmetries for the calculation of stellar surfaces. In the following subsections, the processes of calculating the various surface distributions of parameters such as surface acceleration, effective temperature, etc., are described in detail.

### 4.1 Velocities

As a system performs its motion, each surface elements moves differently with respect to the observer. As a consequence, the velocity of each surface element needs to be calculated separately. Determination of surface velocity of each surface element is essential for modelling a wide variety of phenomena such as the Rossiter-McLaughlin effect (Rossiter, 1924; McLaughlin, 1924) or the impact of spots on the radial velocity curves. Each surface element of the stellar surface performs a motion compositing from 3 components:

- *Secular velocity ( $\gamma$ )*: also known as a centre of mass velocity. Only in radial component of the  $\gamma$  is relevant to model observable quantities of single and binary systems. Therefore, ELISa uses a scalar value of  $\gamma$  as an input value where  $\gamma = [\gamma, 0, 0]$ .
- *Orbital velocity ( $\mathbf{v}_{orb}$ )*: also known as velocity of the component's centre of mass. In the case of a binary system,  $\mathbf{v}_{orb}$  can be obtained as a solution to a two-body problem. In the case of the primary component, its orbital speed can be expressed as (Neslušan, 2018):

$$v_{orb,1} = M_1 \sqrt{\frac{G}{(M_1 + M_2)} \frac{1}{a} \left( \frac{2}{d} - 1 \right)} \quad (16)$$

Subsequently, we can define orbital speed vector for the primary component in a reference frame originating in the system's centre of mass with its  $x$ -axis aligned with the current join vector connecting the components:

$$\mathbf{v}_{orb,1} = \begin{bmatrix} \frac{1+e \cos \nu}{\sqrt{1+e^2+e \cos \nu}} \\ \frac{-e \sin \nu}{\sqrt{1+e^2+e \cos \nu}} \\ 0 \end{bmatrix} v_{orb,1} \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \\ \hat{\mathbf{z}}, \end{bmatrix} \quad (17)$$

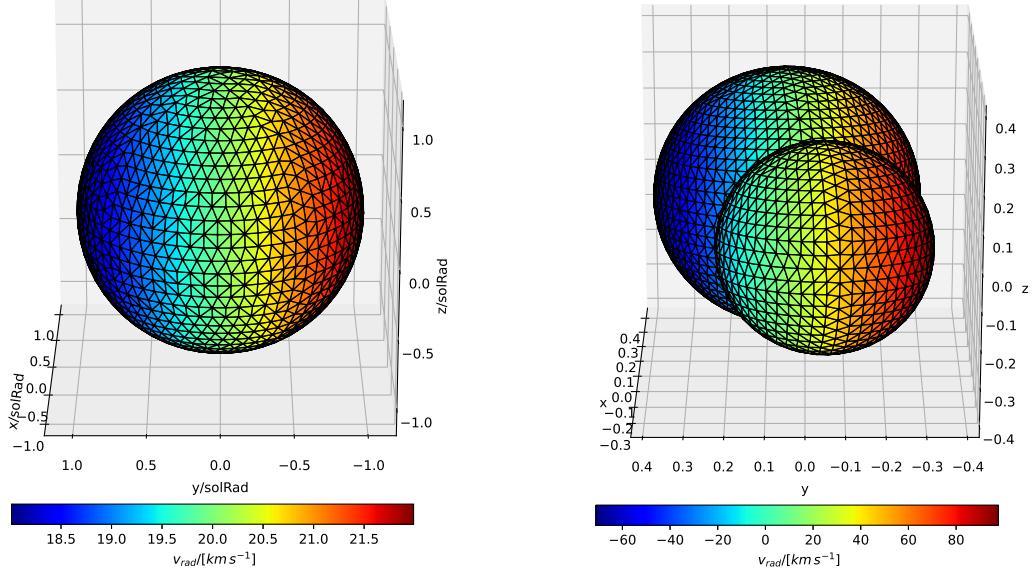


Fig. 15: This figure illustrates the distribution of radial velocities across the surface of a single and binary system with secular velocity  $\gamma = 20 \text{ km s}^{-1}$ . The binary system is captured at photometric phase 0.02.

where  $\nu$  is the true anomaly of the primary component. Finally, the orbital speed vector for the secondary component can be expressed using the conservation of momentum:

$$\mathbf{v}_{\text{orb},2} = -\frac{\mathbf{v}_{\text{orb},1}}{q}. \quad (18)$$

- *Rotational velocity ( $\mathbf{v}_{\text{rot}}$ ):* describes the rotational motion of each surface element around the component's axis of rotation with angular speed  $\omega$  (see Equation 5). Due to the rotational axis being perpendicular to the orbital plane, we can reduce the calculation of the rotational velocity of the point  $\mathbf{X}$  to:

$$\mathbf{v}_{\text{rot}} = \mathbf{X} \times \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix}. \quad (19)$$

In summary, the resulting velocity of the surface element  $i$  of component  $j$  is a sum of secular, orbital and rotational velocities:

$$\mathbf{v}_{j,i} = \boldsymbol{\gamma} + \mathbf{v}_{\text{orb},j} + \mathbf{v}_{\text{rot},j,i}. \quad (20)$$

Note that the first two terms are constant for each component. In case of pulsating stars, the velocity perturbation caused by the pulsations calculated by the Equation 55 in

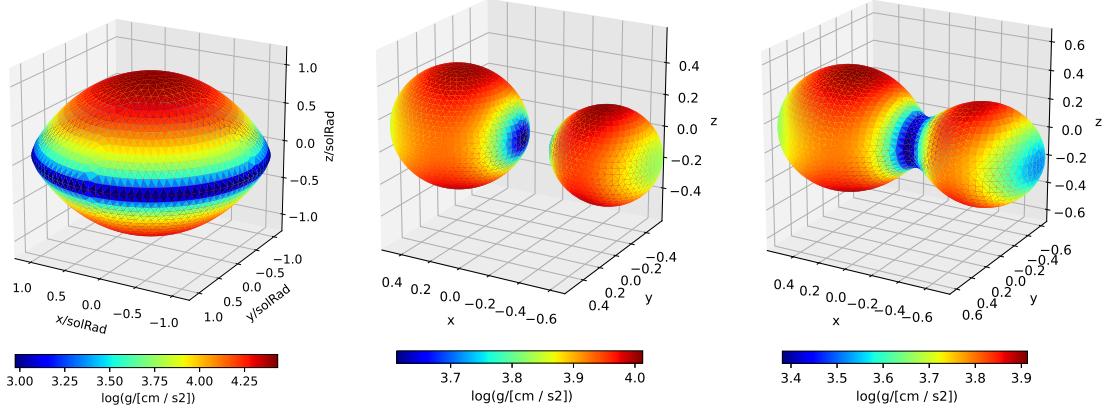


Fig. 16: Variations in surface acceleration in  $\log g$  on rapidly rotating star, detached and over-contact binary.

Section 5.3.2 is added to result in Equation 20. Examples of radial velocity distributions for single and binary systems are displayed in Figure 15.

## 4.2 Surface acceleration

Surface acceleration is distributed unevenly across the surface due to the tidal effects of the companion and centrifugal forces caused by the rotation. A proper evaluation of the surface acceleration is necessary to precisely infer parameters that are dependent on the local properties of the surface, such as a surface distribution of the effective temperature discussed in Section 4.3. Surface acceleration can be calculated as a gradient of the potential function  $\Psi$ :

$$\mathbf{g} = -\nabla\Psi. \quad (21)$$

In theory, Equation 21 needs to be evaluated for every surface element. However, in cases discussed in Section 3.2.1, we can reduce the evaluation of the surface acceleration to a symmetrical part of the surface. Acceleration vectors can be then assigned to the remaining part of the surface using simple transformation.

In the case of a single system, Equations 2 and 21 will lead us to the formula for the surface acceleration on a surface of a rotating star in Cartesian coordinates:

$$\mathbf{g}_{single} = \left( -\frac{GM}{r^3} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \omega^2 \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \right) \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}. \quad (22)$$

Similarly, equation for modified potential 3 can be applied to derive surface acceleration on a surface of binary component:

$$\mathbf{g}_{binary} = -\frac{GM_1}{a^2} \nabla\Omega, \quad (23)$$

## 4 PROPERTIES OF THE EQUILIBRIUM STELLAR SURFACES

---

where factor  $GM_1/a^2$  converts modified potential to the potential function  $\Psi$  suitable for calculating surface acceleration. Corresponding derivatives of the modified potential  $\Omega$  for both components are listed in Appendix B.3. Resulting surface distributions of  $\log g$  with low gravity regions due to rotation and tidal forces are visible in Figure 16.

### 4.3 Effective temperature

Stars radiate a significant portion of their produced energy in the form of electromagnetic radiation. To help parametrize the radiative properties of the star, we can refer to a parameter called *effective temperature* ( $T^{eff}$ ), i.e. a temperature of a black body of the same size as the star that would radiate the same total amount of energy (Rouan, 2011). The stellar surface does not radiate the energy uniformly. Thus, the local value of the effective temperature varies across the surface. ELISa takes into account two significant contributions to the  $T^{eff}$  inhomogeneity: gravity darkening and reflection effect. Treatment of both effects is described in the following sections.

#### 4.3.1 Gravity darkening

Due to the uneven distribution of surface acceleration, the distribution of the effective temperature will vary across the stellar surface. This effect is called *gravity darkening* (also known as gravity brightening). The apparent brightening of the surface in regions with higher gravity is caused by the steeper temperature gradient that puts hotter layers into the same optical depth.

A power law can approximate the dependence of the emergent flux, and thus, the effective temperature on the surface acceleration:

$$T^{eff} \propto g^{\frac{\beta}{4}} \quad (24)$$

where parameter  $\beta$  refers to the *gravity darkening factor*. Initial estimation of gravity darkening was done by von Zeipel (1924) for radiative stellar envelopes of hot stars. According to von Zeipel theorem, the outgoing emergent intensity of radiation is proportional to the surface acceleration and thus,  $\beta = 1.0$ . On the other hand, the upper convective layer present in cooler stars significantly reduces the dependence of effective temperature on surface acceleration. Lucy (1967) estimated the value of the gravity darkening factor to be close to 0.32 based on the convective models of stellar envelopes. A more recent study (Espinosa Lara and Rieutord, 2012) demonstrated that the  $\beta$  does not stay constant for a given type of stellar envelope, but it is a slowly changing function of mass ratio and filling factor. Therefore, ELISa supports custom values of

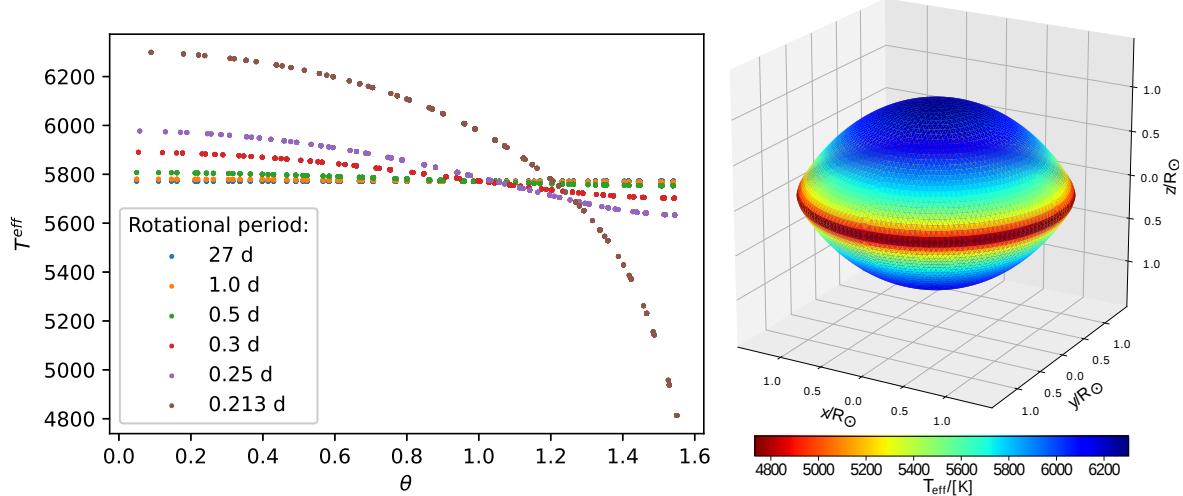


Fig. 17: The left figure illustrates the distribution of  $T^{eff}$  as a function of co-latitude angle  $\theta$  on a single star model with solar parameters. Decreasing the rotational period increases temperature variation until the star reaches the breakup period of rotation. The figure on the right side displays the temperature distribution on a heavily distorted single star model with solar parameters except for a rotational period of 0.213 d. All models adopted the gravity darkening coefficient  $\beta = 0.32$  for convective envelopes.

$\beta$  that can be set as fixed, variable or constrained during a fitting process (see Section 8).

Standard approach in modelling surface distribution of  $T^{eff}$  (Wilson, 1979) is to use a power law in Equation 24 to express  $T_i^{eff}$  at any given surface element  $i$  using polar effective temperature  $T_{pole}^{eff}$ :

$$T_i^{eff} = T_{pole}^{eff} \left( \frac{g_i}{g_{pole}} \right)^{\frac{\beta}{4}}, \quad (25)$$

where local and polar surface accelerations  $g_i$  and  $g_{pole}$  are calculated according to the process described in Section 4.2. Additionally,  $T_{pole}^{eff}$  can be derived using the definition of the stellar effective temperature:

$$T_{pole}^{eff} = T^{eff} \left( \frac{\sum_{i=1}^n S_i}{\sum_{i=1}^n \left( \frac{g_i}{g_{pole}} \right)^{\beta} S_i} \right)^{\frac{1}{4}}, \quad (26)$$

where  $S_i$  is a surface area of the surface element  $i$  and the summations iterate over all  $n$  surface elements. Examples of the gravity darkening on the surface distribution  $T^{eff}$  are listed in Figure 17 in the case of a single star with solar parameters and varying rotational period.

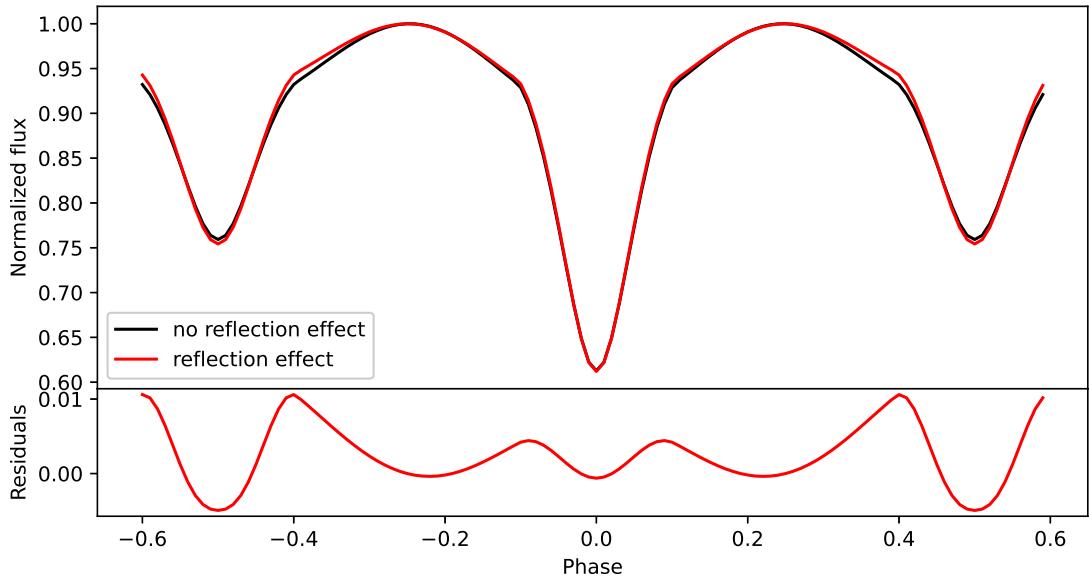


Fig. 18: Impact of reflection effect on the shape of the light curve in case of close binary system with  $T_{primary}^{eff} = 7000\text{ K}$  and  $T_{secondary}^{eff} = 6000\text{ K}$ .

#### 4.3.2 Reflection effect

The gravity darkening effect alone is not sufficient to explain surface temperature variations in close binary stars. The proximity of the stellar components will cause additional heating of the stellar surface facing the companion that will significantly affect the shape of the light curve as demonstrated in Figure 18. ELISa adopts treatment of reflection effect published by Wilson (1990). In this case, the bolometric albedo  $A$  of the irradiated component plays an important role in the extent of the heating caused by the reflection effect. The *bolometric albedo* is, in this case, defined as a local ratio of re-radiated energy to the irradiance on the given surface element caused by the companion. This section describes in detail how is the reflection effect implemented in ELISa and what steps were made to improve computational speed.

Different values of the  $A$  for the radiative and convective envelopes should be used for radiative and convective envelopes representing different mechanisms of energy redistribution within the upper layers of the star. In the case of a radiative envelope,  $A$  should be generally kept very close to 1, and the value decreases for the convective envelope to  $\sim 0.5$  (Rucinski, 1969). Switching between the radiative and convective envelope occurs at an approximate effective temperature of 7000 K (Figure 3.9 in Aerts et al. (2010)).

The main goal of the following approach is to describe the correction factors for the local effective temperature  $T_i$  of the irradiated surface element  $i$ . The correction factor considers heating caused by the flux contributions from each visible surface element on

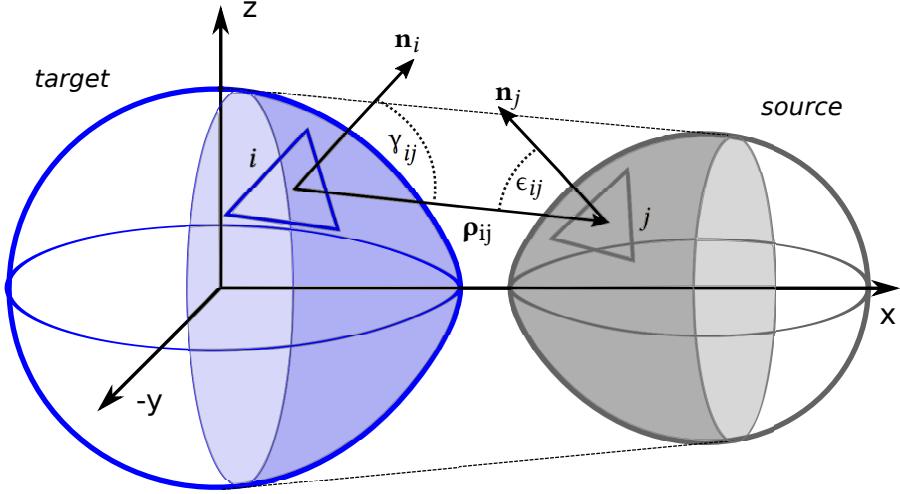


Fig. 19: Detailed description of variables mentioned in Equation 29.

the binary counterpart, called "source". Initially, we can describe the temperature of the irradiated surface element  $T'_i$  as following:

$$T'_i = \sqrt[4]{R_i} T_i, \quad (27)$$

where  $R_i$  is the reflection factor and describes the ratio between the bolometric fluxes from the given surface element with and without the reflection effect taken into account:

$$R_i = 1 + A_t \frac{E_i}{M_{0,i}}, \quad (28)$$

where  $A_t$  is the albedo of the irradiated "target" component,  $E_i$  is integrated *irradiance* (ISO, 2019) from the source, irradiating our target surface element  $i$  and  $M_{0,i}$  is a *radiant exitance* (ISO, 2019) of the surface element without the reflection effect taken into account. Note that  $E_i$  needs to be calculated for each surface element  $i$  due to the change in orientation of the target surface element relative to surface elements of the source star.

The most computationally demanding step during the calculation of the reflection effect is to obtain irradiance flux ratio  $E_i/M_{0,i}$  for each surface element of the target. The ratio is calculated by summation of flux contributions over  $N'$  visible surface elements of the source star:

$$\frac{E_i}{M_{0,i}} = \frac{1}{\mathcal{L}_i^{int} T_i^4} \sum_{j=1}^{N'} T_j^4 R_j \mathcal{L}_{ij} (\cos \epsilon_{ij}) \frac{\cos(\gamma_{ij}) \cos(\epsilon_{ij}) S_j}{\rho_{ij}^2}, \quad (29)$$

where  $R_j$ ,  $\mathcal{L}_{ij}$  and  $S_j$  is the reflection effect correction factor, limb darkening factor and surface area of the given surface element on the source.  $\rho_{ij}$  is the mutual distance of

## 4 PROPERTIES OF THE EQUILIBRIUM STELLAR SURFACES

---

visible pairs of surface elements.  $\mathcal{L}_i^{int}$  is the bolometric integrated limb darkening factor of the target surface element. Finally,  $\gamma_{ij}$  and  $\epsilon_{ij}$  are angles between surface normals and their mutual join vector (see Figure 19 for visualization). Areas of the surface elements were already calculated during the surface discretization process. Cosines of angles  $\gamma_{ij}$  and  $\epsilon_{ij}$  can be easily calculated from already known outward-facing normal vectors for each surface element and join vector  $\rho_{ij}$ :

$$\cos \gamma_{ij} = \frac{\mathbf{n}_i \cdot \boldsymbol{\rho}_{ij}}{\rho_{ij}}; \quad \cos \epsilon_{ij} = -\frac{\mathbf{n}_j \cdot \boldsymbol{\rho}_{ij}}{\rho_{ij}}. \quad (30)$$

Calculation of the reflection effect can be therefore summarized into the following steps. The first set of steps is performed only once:

1. Indices of surface elements  $i, j$  facing the companion stars are determined based on distance and relative size of components (light blue and grey parts of the surfaces in Figure 19).
2. Reflection factors on both components are set to 1.0 across all visible surface elements determined in step 1.
3. Parameters in Equation 29 staying constant during the whole process such as  $\cos \gamma_{ij}$ ,  $\cos \epsilon_{ij}$  and  $\boldsymbol{\rho}_{ij}$  are calculated in advance and stored in a auxiliary variable  $Q$ :

$$Q_{ij} = \frac{\cos(\gamma_{ij})\cos(\epsilon_{ij})}{\rho_{ij}^2}. \quad (31)$$

The following steps should be performed iteratively for each component until the desired precision is reached. In ELISa, number iterations can be specified using variable `settings.REFLECTION_EFFECT_ITERATIONS` with the default value of 2:

4. Irradiance flux ratio  $E_i/M_{0,i}$  in Equation 29 is calculated.
5. Surface distribution of reflection factor  $R$  is updated (Equation 28).

Finally, the last step in the process is to calculate effective temperature distribution using Equation 27 and newly derived distributions of reflection factor  $R$ .

A significant decrease in computational time can be achieved using surface symmetries described in Section 3. In the case of stellar surfaces where symmetry can be utilized, we can reduce the calculation of the temperature distribution on the target component to one quarter. Subsequently, we can symmetrically map the calculated temperature distribution to the rest of the surface. Therefore, surface symmetry has significant consequences on the number of operations required to calculate two-dimensional matrices such as limb darkening factor  $\mathcal{L}_{ij}$  and geometry related quantity  $Q_{ij}$  as explained in Figure 20.

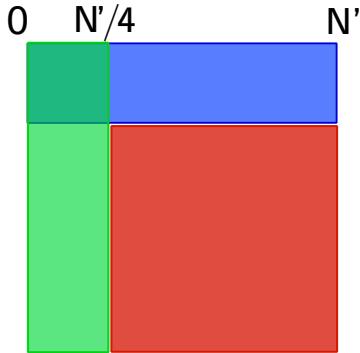


Fig. 20: Diagram illustrating a reduction in the number of operations necessary to evaluate matrix parameters when utilizing surface symmetry. The vertical and horizontal sides of the rectangle correspond to the number of mutually visible surface elements on the primary and secondary components. Blue and green regions point to values used to calculate the reflection effect on a primary and secondary component. The red region indicates matrix elements corresponding to surface element combinations belonging to symmetrical counterparts star quadrants that are not used to calculate the reflection effect.

#### 4.4 Limb-darkening

*Limb-darkening* (LD in short) is an apparent dimming of the stellar surface towards the edge of the stellar disc from the observer's viewpoint. The LD factor in the following form is commonly used to describe the decrease in the surface intensity:

$$\mathcal{L}(\theta) = \frac{I(\theta)}{I(0)}, \quad (32)$$

where  $I(0)$  stands for the intensity of radiation in the direction of the outward-facing normal vector, and  $I(\theta)$  is the intensity of the surface element observed at an incidence angle  $\theta$  measured between the line-of-sight vector and the surface element normal vector.

The extent and the properties of the dimming depend significantly on the structure of the stellar atmosphere in question (Milne, 1921). The opacity of the stellar material causes the majority of radiation to originate in a particular geometrical depth that depends on the incidence angle  $\theta$ . Due to the temperature gradient present in the stellar atmosphere, the disk's centre appears brighter since the emergent radiation is generated in deeper and hotter layers. The impact of the limb-darkening on the surface radiance distribution in the direction to the observer is displayed in Figure 21.

Modelling of stellar atmospheres is a computationally heavy task that would significantly prolong the calculation of outgoing flux. Therefore, a standard approach is to calculate *limb-darkening factor*  $\mathcal{L}$  in form of empirical limb darkening law with coefficients derived from atmospheric models. There are numerous LD laws and ELISa implements 3 of them: linear (cosine), logarithmic (Klinglesmith and Sobieski, 1970)

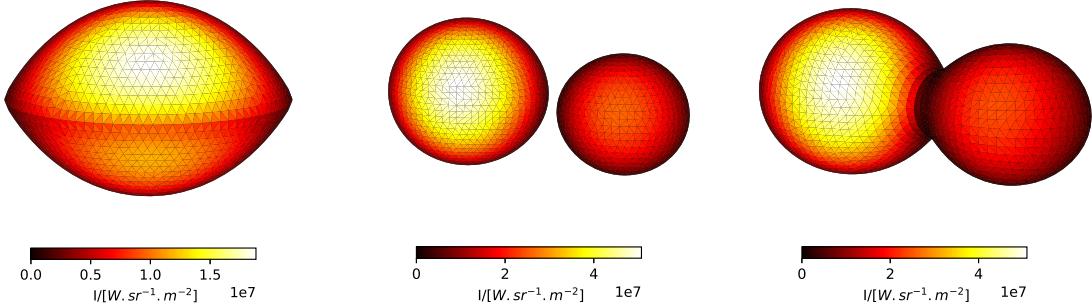


Fig. 21: Distributions of radiance in the observer's direction for rapidly rotating single star, detached and over-contact binary components using linear limb darkening law. Systems are captured at photometric phase 0.14 and inclination 80°.

and square-root law (Diaz-Cordoves and Gimenez, 1992). ELISa enables to switch between different LD laws using variable settings.LIMB\_DARKENING\_LAW. The most simple LD law is linear utilizing single coefficient  $x_{lin}$ :

$$\mathcal{L}_{lin}(\theta) = 1 - x_{lin} + x_{lin} \cos \theta. \quad (33)$$

The logarithmic and square-root laws depend on two coefficients  $x$  and  $y$ :

$$\mathcal{L}_{log}(\theta) = 1 - x_{log}(1 - \cos \theta) - y_{log} \cos \theta \log(\cos \theta), \quad (34)$$

$$\mathcal{L}_{sqrt}(\theta) = 1 - x_{sqrt}(1 - \cos \theta) - y_{sqrt} \left(1 - \sqrt{\cos \theta}\right). \quad (35)$$

The second limb-darkening related parameter is *integrated limb-darkening factor*  $\mathcal{L}^{int}$  defined as:

$$\mathcal{L}^{int} = \int_{\Omega/2} \mathcal{L}^{bol}(\theta) \cos(\theta) d\omega \quad (36)$$

where integral is performed over a hemisphere. Factor  $\mathcal{L}^{int}$  accounts for anisotropy of the outgoing radiation due to LD. Its primary role is to renormalize the radiance of the surface element in the normal direction to conserve element's total radiosity. Definitions of the LD laws in Equations 33, 34 and 35 applied to the integral 36 will lead to the corresponding relations for the integrated LD factor:

$$\mathcal{L}_{lin}^{int} = \pi \left(1 - \frac{1}{3}x_{lin}\right), \quad (37)$$

$$\mathcal{L}_{log}^{int} = \pi \left(1 - \frac{1}{3}x_{log} + \frac{2}{9}y_{log}\right), \quad (38)$$

$$\mathcal{L}_{sqrt}^{int} = \pi \left(1 - \frac{1}{3}x_{sqrt} - \frac{1}{5}y_{sqrt}\right). \quad (39)$$

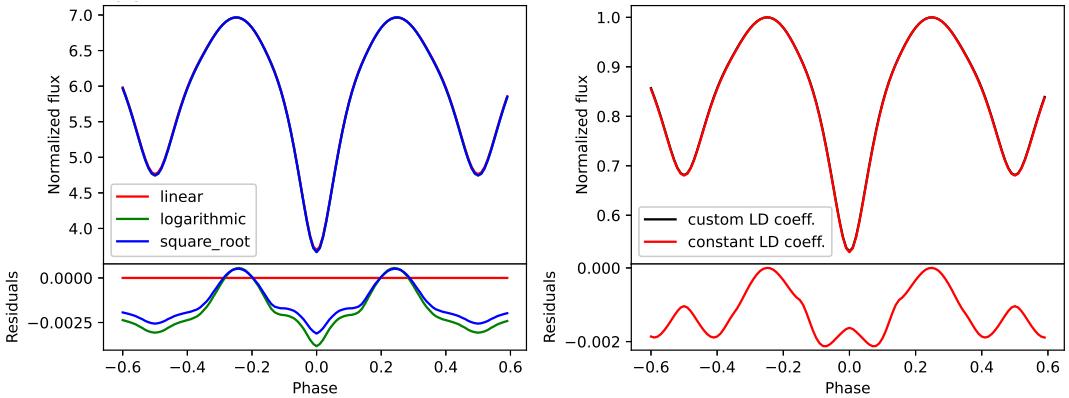


Fig. 22: The left panel demonstrates the impact of different LD laws in the case of a semi-detached system. The panel on the right shows the impact of using custom LD coefficients versus using a single set of LD coefficients for the whole surface of the star.

Corresponding coefficients are interpolated from a pre-calculated grid of passband-specific coefficients obtained for different values of effective temperatures, surface accelerations and metallicities. Effect of different LD laws on the shape of the light curve of a binary system with tidally distorted components is visualized on the left side of Figure 22. Due to variations in  $T_{\text{eff}}$  and surface acceleration across the stellar surface, unique LD coefficients are interpolated for each surface element which has a significant impact on the precision for systems with severely deformed components, as demonstrated on the right panel of Figure 22. There are numerous available tables with LD coefficients published in recent decades (e.g. Claret (2016); Reeve and Howarth (2016); Claret et al. (2012)), and ELISa has a prepared framework to use various LD tables if necessary. Currently, ELISa uses limb darkening coefficients from table<sup>1</sup> of coefficients calculated for LD laws described in van Hamme (1993) based on Castelli and Kurucz (2003) models of stellar atmospheres.

#### 4.5 Normal radiance

Due to the properties of the real stellar atmospheres, bound-bound and bound-free absorption and emission processes will cause the stellar spectrum to depart from the black-body thermal radiation profile. Changes in the shape of the light curves of a close binary system in Figure 23 demonstrates the impact of using a black-body approximation for modelling of stellar atmosphere instead of more sophisticated models available in pre-calculated tables.

*Spectral radiant exitance* (ISO, 2019)  $M_{\lambda,i}$ , i.e. the amount of radiant flux leaving a unit of a surface element  $i$  per unit of wavelength is obtained by interpolating the

<sup>1</sup><http://faculty.fiu.edu/~vanhamme/limb-darkening/>

## 4 PROPERTIES OF THE EQUILIBRIUM STELLAR SURFACES

---

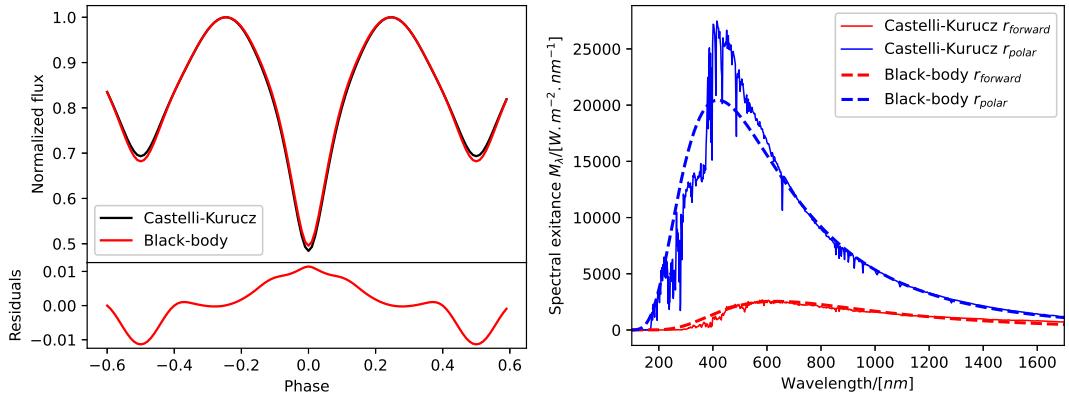


Fig. 23: The necessity to use a proper model of the stellar atmosphere is highlighted on examples of the light curve, and synthetic spectra of semi-detached binary observed in the Bessel U passband. The residual curve in the bottom left figure shows a huge discrepancy in observed flux between models using black-body and synthetic model of an atmosphere from Castelli and Kurucz (2003). The figure on the right displays significant differences in synthetic spectra obtained from polar and equatorial regions of the primary component caused by the gravity darkening effect.

pre-calculated table of atmosphere models such as published in Castelli and Kurucz (2003). This process is similar to the determination of the LD coefficients. It provides a sufficiently precise yet fast determination of the  $J_\lambda$  without the need to resort to a crude black-body approximation or time-consuming modelling of stellar atmospheres. In general, the atmosphere model used on each surface element is defined by three quantities:  $T^{eff,i}$ ,  $\log g_i$  and metallicity  $M/H$ , which is assumed constant across the surface.

As a result, normal *spectral radiance* (ISO, 2019)  $L_{n,\lambda,i}$ , i.e. the amount of radiant flux emitted on a given wavelength by the unit of area on a surface element  $i$  in the direction of the outward-facing normal vector, is derived from the spectral radiant exitance  $M_{\lambda,i}$  in the following manner:

$$L_{n,\lambda,i} = \frac{M_{\lambda,i}}{\mathcal{L}_{\lambda,i}^{int}}, \quad (40)$$

Subsequently, normal *radiance* (ISO, 2019)  $L_{n,i}$  integrated over each passband can be calculated as:

$$L_{n,i} = \int_{\lambda} L_{n,\lambda,i} \tau_{\lambda} d\lambda, \quad (41)$$

where  $\tau_{\lambda}$  represents the response function of the passband.

Due to practical issues, integrated LD factor  $\mathcal{L}^{int}$  for the whole passband is provided instead of the wavelength-dependent counterpart  $\mathcal{L}_{\lambda}^{int}$ . Therefore,  $\mathcal{L}^{int}$  can be carried out of the integral in Equation 41 with spectral intensity replaced by the Equation 40:

$$L_{n,i} = \frac{1}{\mathcal{L}_i^{int}} \int_{\lambda} M_{\lambda,i} \tau_{\lambda} d\lambda. \quad (42)$$

## 4 PROPERTIES OF THE EQUILIBRIUM STELLAR SURFACES

---

Normal radiances  $L_{n,i}$  calculated for each surface element using Equation 42 are stored in the corresponding container. Now completed **StarContainer**, storing all necessary surface parameters, can be used to calculate observables (e.g. flux, radial velocity curves, etc.). As mentioned in previous sections, surface symmetry can reduce the number of surface elements where normal radiance needs to be evaluated.

## 5 Non-radial pulsations

Sections 3 and 4 discussed in great details the construction process of a stellar surface according to the assumptions defined in Section 3.1. These assumptions are, however, not true in a plethora of classes of pulsating variable stars dispersed across the Hertzsprung–Russell (HR) as visualized in Figure 24. In the case of pulsating variables, the hydrostatic equilibrium is perturbed by the various internal processes, i.e. “driving mechanisms” powerful enough to overcome the internal friction and induce a periodic motion of a stellar matter around the equilibrium.

There are several proposed or established driving mechanisms that can induce pulsations, such as  $\kappa$ -mechanism Eddington (1917), based on the modulation of the opacity of the stellar matter during the pulsation cycle. In this regime, the star is functioning as a heat engine, and this mechanism is also used to explain the instability in the area of the HR diagram called *instability strip* where Cepheids, RR Lyr and  $\delta$  Sct type of variables reside. Additionally, slowly pulsating B-type (SPB) stars located outside the instability strip are also excited by the  $\kappa$ -mechanism. A similar proposed excitation mechanism is an  $\epsilon$ -mechanism based on the modulation in the energy production rate within the stellar core. In theory, this excitation mechanism could be responsible for the instability of extremely massive stars (Aerts et al., 2010).

Next resident of the instability strip, the  $\gamma$  Dor type of variables, utilize a *convective blocking* mechanism (Guzik et al., 2000) where the bottom of the convective zone does not have time to adjust for any change in the total luminosity passing through the given layer. Finally, there is a *stochastic* driving mechanism responsible for driving the solar-like oscillations due to the turbulent convective motion of the material on the top of the convective layer.

Previously described types of pulsators can also be observed as members of binary stars (Rappaport et al., 2021; Skarka et al., 2019). The presence of the pulsator in the binary system can aid the further asteroseismological analysis due to the much more precise determination of absolute parameters such as mass, size and age. However, on the other side, studying such objects requires taking additional steps during the data processing to remove the binary signature from the observations. Moreover, the identification of the source of the pulsations needs to take place as well.

A mere presence of the stellar companion, especially in the case of systems with eccentric orbits, can lead to tidally induced oscillations, as may be the case of IM Per (Lee et al., 2021) or KIC 3858884 (Maceroni et al., 2014). Additionally, tidal deformation of close components causes the mode axis to align with the tidal axis (axis pointing towards the companion) instead of the axis of rotation (Handler et al., 2020).

## 5 NON-RADIAL PULSATIONS

---

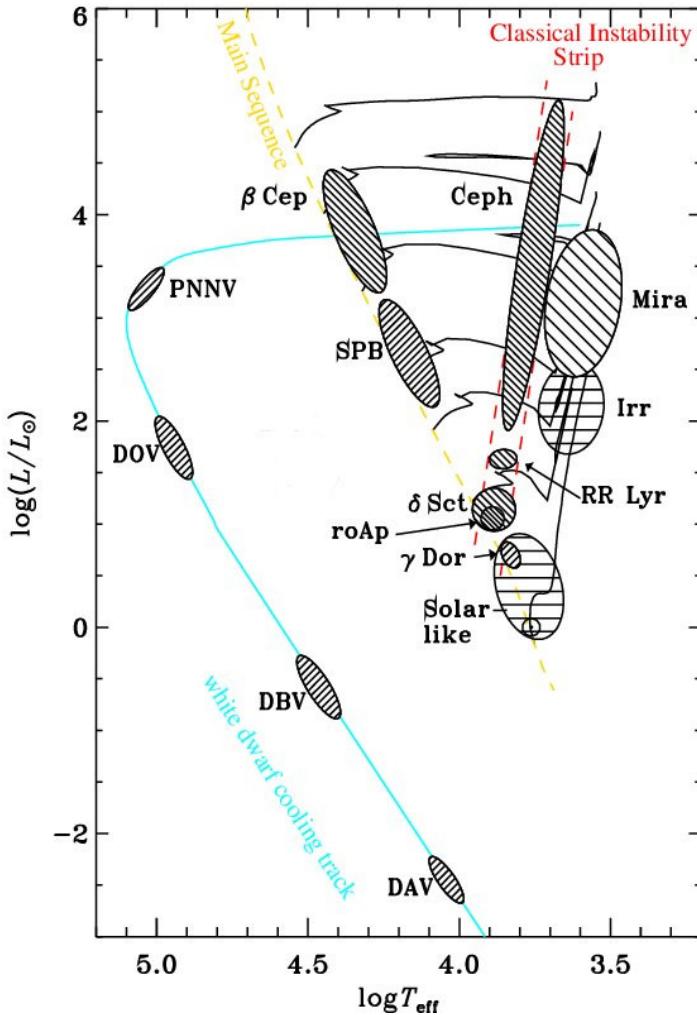


Fig. 24: Overview of the distribution of the different classes of pulsating variables across the HR diagram (Kim et al., 2006).

The “re-alignment” occurs mainly because the tidal effects have a much more influence on the resulting shape of the star than the effects of rotation.

One of the main goals of our efforts is to develop a module capable of modelling non-radial oscillations on a surface of a tidally deformed component of an eclipsing binary. Therefore, such a module should be capable of dealing with the following phenomena:

- Distortion of the pulsation mode due to the presence of the tidal forces.
- Modulation of the modal amplitude during eclipses due to the partial visibility of the stellar surface.
- Modelling non-radial modes aligned according to an arbitrary axis. This enables the alignment of the modal axis either with the rotational axis, tidal axis or arbitrarily selected axis.

## 5 NON-RADIAL PULSATIONS

---

- Expressing a perturbed quantity  $\delta f$  as a sum of partial perturbations from all modes defined as a function of an equilibrium model  $M_0$  and pulsation mode  $m$ :

$$\delta f = \sum_{i=1}^n F(M_0, m_i) \quad (43)$$

The following sections describe in detail the implementation of non-radial pulsations in the framework of the ELISa package.

### 5.1 Implementation of stellar pulsations in ELISa package

Pulsations are defined within the framework of ELISa as attribute of the `Star` instance, similar to the stellar spots (see Table 6 in Appendix A). Each pulsation mode is defined by the instance of `PulsationMode` that contains attributes characterizing a given mode. It is possible to initialize one or multiple pulsation modes in a form of a list of `PulsationMode` instances passed through `pulsations` attribute of a parent star. The order in which the modes are defined is not important, as the resulting perturbation of the stellar surface is calculated as a superposition of the partial perturbations from each mode as expressed in Equation 43. The list of all available attributes of the `PulsationMode` is located in Table 8 in Appendix A.

### 5.2 Pulsations as a perturbation of an equilibrium stellar surface

Even though driving mechanisms can perturb the star away from the hydrostatic equilibrium, these deviations are almost always small enough to treat them as small compared to the equilibrium values. The extreme cases of high amplitude pulsators such as the classical Cepheids are out of the scope of this package since their radius can change substantially during the pulsation period (Rodgers, 1957). However, the surface quantities of the low amplitude pulsators such as  $\delta$  Sct,  $\gamma$  Dor and or SPB stars change in the order of %. Such low amplitudes enable us to treat oscillations as a small perturbation  $\delta f$  of the quasi-equilibrium quantity  $f$  corresponding to the surface element described by the angular coordinates  $\theta, \phi$  aligned with the axis of the pulsation mode at a given moment  $t$ :

$$f(\theta, \phi, t) = f(\theta_0, \phi_0, t) + \delta f(\theta_0, \phi_0, t), \quad (44)$$

where  $\delta f \ll f$ . This equation utilizes the Lagrangian approach, i.e. the approach following the motion of the surface element between equilibrium and perturbed position:

$$\mathbf{r}(\theta, \phi) = \mathbf{r}(\theta_0, \phi_0) + \boldsymbol{\delta r}(\theta_0, \phi_0), \quad (45)$$

## 5 NON-RADIAL PULSATIONS

---

where the position vector  $\mathbf{r}$  also depends on time  $t$ . From now on, the time dependence is taken as implicit unless otherwise specified.

The low amplitude pulsations enable us to approximate the perturbations of the surface quantities in the form of spherical harmonics  $Y_\ell^m$  derived from the solution for the non-rotating spherical star:

$$\delta f(\theta, \phi) = \tilde{f} Y_\ell^m(\theta, \phi) \exp \{-i[\omega_p(t - t_{ltt}) + \varphi_0]\}, \quad (46)$$

where  $\tilde{f}$  is the root mean square (RMS) of the given pulsation mode. Angular degree  $\ell$  describes the number of surface nodal lines, and azimuthal order  $m$  defines the number of nodal lines passing through mode axis, where the sign of  $m$  describes the direction of the travel of the mode (counter-clockwise direction for positive  $m$ ). Parameter  $\omega_p$  is the angular frequency of the pulsation mode along with the initial phase  $\varphi_0$ . Observational time  $t$  is in case of binary star components corrected for the light travel time effect (LTT) with the correction  $t_{ltt}$  due to the changing distance between the orbiting star and the observer (see Section 6.2 for calculation of the  $t_{ltt}$ ). For the sake of brevity, the time exponential will be from now on assumed as a part of the spherical harmonics  $Y_\ell^m$  (see Appendix C).

### 5.3 Kinematics of the surface points

Surface elements of the pulsating star will perform a periodic motion around the equilibrium position established in Section 3. We can describe this motion using an Eulerian (fixed position) approach or the Lagrangian (fixed element) approach. ELISa uses the second formalism and adapts the modelling method described by Townsend (2003) within the scope of the linear approximation.

The description of the perturbations is more convenient within the spherical coordinate system  $r, \theta, \phi$  aligned with the axis of the pulsation mode. The mode axis is defined with the angular coordinates  $\vartheta_{mode}, \varphi_{mode}$  defined in the coordinate system aligned with the axis of the rotation (see Figure 25). Additionally, the mode axis can be set as tidally locked, which will fix the position of the mode axis with respect to the companion regardless of the rotation of the parent star (see Table 8 in Appendix A).

#### 5.3.1 Displacement of the surface element on a complex plane

This section aims to describe the displacement vector  $\delta\mathbf{r}$  in a complex representation that will later enable us to derive the other kinematic quantities such as the velocity and the acceleration of a surface element. The total displacement  $\delta\mathbf{r}$  can be divided

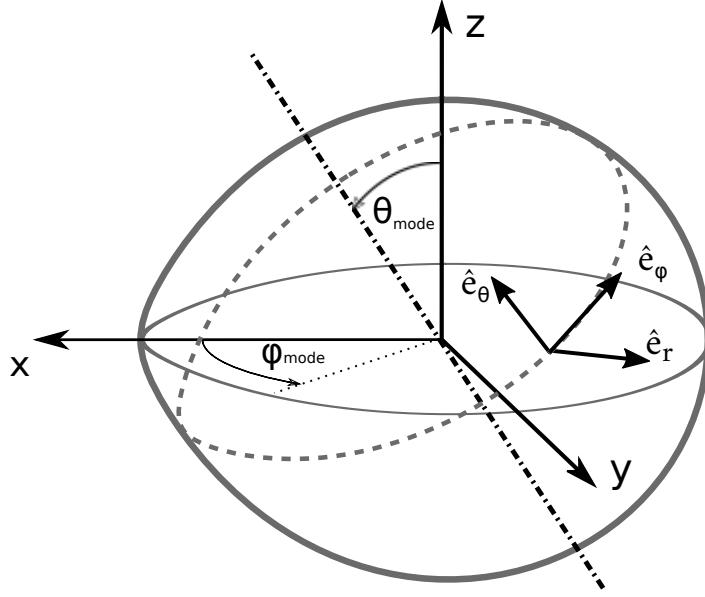


Fig. 25: Illustration of the orientation of the pulsation mode axis (dash-dotted line) defined by the angles  $\vartheta_{mode}$ ,  $\varphi_{mode}$  with respect to the axis of rotation. Figure also shows the local orientation of the unit vectors  $\{\hat{e}_r, \hat{e}_\theta, \hat{e}_\phi\}$ .

into its components within the previously described coordinate system:

$$\delta\mathbf{r}(\theta, \phi) = \delta r(\theta, \phi)\hat{e}_r + r\delta\theta(\theta, \phi)\hat{e}_\theta + r \sin \theta \delta\phi(\theta, \phi)\hat{e}_\phi, \quad (47)$$

where the subscript 0 was omitted from the equilibrium quantities,  $\delta r$ ,  $\delta\phi$ ,  $\delta\theta$  are perturbations of the spherical coordinates and  $\{\hat{e}_r, \hat{e}_\theta, \hat{e}_\phi\}$  are the unit vectors along the corresponding axes (see Figure 25). Based on approach such as Unno et al. (1989), the radial part of the displacement vector can be expressed as:

$$\delta r(\theta, \phi) = A_r Y_\ell^m(\theta, \phi), \quad (48)$$

where  $A_r$  is the RMS of the radial displacement.

The horizontal component of the displacement vector  $\delta\mathbf{r}_\perp$ , containing the last two terms of Equation 47, can be expressed according to Townsend (2003) as:

$$\delta\mathbf{r}_\perp(\theta, \phi) = A_\perp r \nabla_\perp Y_\ell^m(\theta, \phi), \quad (49)$$

where  $A_\perp$  is the RMS of the horizontal displacement and  $\nabla_\perp$  is the horizontal component of the gradient operator in spherical coordinates. The ratio between the horizontal and the radial RMS can be expressed as (Aerts et al., 2010):

$$\frac{A_\perp}{A_r} = \sqrt{l(l+1)} \frac{GM}{r_{eq}^3 \omega_p^2}. \quad (50)$$

## 5 NON-RADIAL PULSATIONS

---

The ratio  $A_{\perp}/A_r$  can be also set manually (see Table 8 in Appendix A). The ratio is inversely proportional to the square of the mode frequency. Therefore, the majority of the displacement in the case of low-frequency g-modes occurs in the horizontal direction. In contrast, in the case of the p-modes, the motion also has a substantial radial component. The horizontal and radial displacement RMSs also have to satisfy the condition:

$$\tilde{r} = \sqrt{A_r^2 + A_{\perp}^2}, \quad (51)$$

where  $\tilde{r}$  is the RMS of the absolute value of the displacement  $|\delta\mathbf{r}|$  derived from the user defined RMS of surface speed  $\tilde{v}$  as:

$$\tilde{r} = \frac{\tilde{v}}{\omega_p}. \quad (52)$$

Taking into account Equations 45 and 49, we can derive the angular displacements  $\delta\theta$  and  $\delta\phi$  along the unit vectors  $\hat{\mathbf{e}}_{\theta}, \hat{\mathbf{e}}_{\phi}$ :

$$\begin{aligned} \delta\theta &= \frac{A_{\perp}}{r} C(\ell, m) \frac{\partial Y_{\ell}^m}{\partial \theta}, \\ \delta\phi &= \frac{A_{\perp}}{r \sin^2 \theta} C(\ell, m) \frac{\partial Y_{\ell}^m}{\partial \phi}, \end{aligned} \quad (53)$$

where  $C(\ell, m)$  is a spherical harmonics derivative renormalization constant and its form along with both derivatives of  $Y_{\ell}^m$  are listed in the Appendix C.

### 5.3.2 Kinematic quantities of the pulsating stellar surface

Equations 48 and 53 completely describe the behavior of the displaced surface element. The absolute value of the complex displacement vector  $\delta\mathbf{r}$  describes a local amplitude of a surface element motion due to the pulsations. Subsequently displacement  $\delta\mathbf{r} = [\delta r, \delta\theta, \delta\phi]$  is used to generate the displacement, velocity and acceleration. The displacement  $\xi$  represents a real component of the complex displacement and it describes a position of the surface element:

$$\xi = \text{Re}\{\delta\mathbf{r}\}. \quad (54)$$

Similarly, the velocity  $\mathbf{v}$  and acceleration  $\mathbf{a}$  of the surface element can be derived using real components of the first and the second time derivatives of complex surface displacement:

$$\begin{aligned} \mathbf{v} &\equiv \text{Re} \left\{ \frac{\partial \delta\mathbf{r}}{\partial t} \right\} = \omega_p \text{Im} \{ \delta\mathbf{r} \}, \\ \mathbf{a} &\equiv \text{Re} \left\{ \frac{\partial^2 \delta\mathbf{r}}{\partial t^2} \right\} = -\omega_p^2 \text{Re} \{ \delta\mathbf{r} \}. \end{aligned} \quad (55)$$

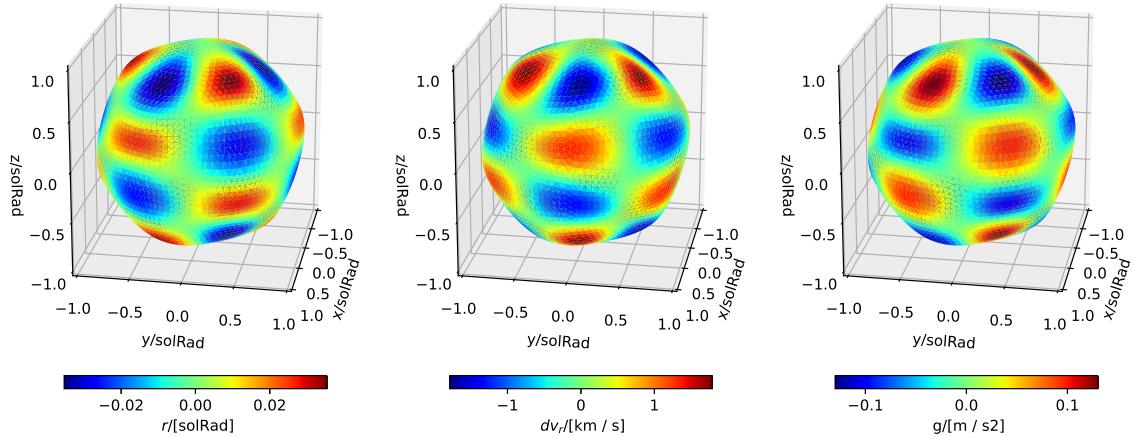


Fig. 26: This set of figures illustrates from left to right the radial components of the displacement, velocity and acceleration perturbation (from left to right) caused by a pulsation mode with  $\ell = 6$  and  $m = 3$ , velocity amplitude  $\tilde{v} = 2.5 \text{ km s}^{-1}$  and frequency  $f_p = 1 \text{ d}^{-1}$  on a solar model. Each image captures the single-star model at the same moment.

The derived values of  $\boldsymbol{\xi}$ ,  $\mathbf{v}$ ,  $\mathbf{a}$  are subsequently transformed from tilted coordinate system aligned with the pulsation mode into the coordinate system aligned with the axis of rotation. Those values can be finally added to the equilibrium quantities calculated in Sections 3 and 4 to form a perturbed surface (see Figure 26).

#### 5.4 Perturbation of the surface temperature distribution

The deformation of the surface geometry is accompanied by the perturbation in the surface distribution of effective temperature  $T$ . In many cases, such as  $\kappa$ -mechanism or convective blocking, the surface deformation is a consequence of heat entrapment, which causes the surface region to heat up and expand. The maximum temperature perturbation  $\delta T$  usually precedes the moment of the maximum radial displacement and usually occurs near the point of the maximum speed of expansion. Due to the non-adiabaticity of the pulsations, especially in the case of long-period oscillations, the user-defined phase difference  $\Delta\varphi_T$  between the displacement  $\delta r$  and temperature perturbation  $\delta T$  can be set to a custom value, different from a default value  $\pi/3$  (Townsend, 2003). The relation between the complex displacement  $\delta r$  and relative temperature perturbation  $\delta T/T$  can be defined as:

$$\frac{\delta T}{T} = \frac{\tilde{T}}{T^{eff}} \frac{1}{A_r} \operatorname{Re} \{ \delta r \exp(i\Delta\varphi_T) \}. \quad (56)$$

The relative temperature perturbation amplitude  $\tilde{T}/T^{eff}$  can be calculated with-in the Cowling (Cowling, 1941) and adiabatic approximation, i.e. assumptions that the

## 5 NON-RADIAL PULSATIONS

---

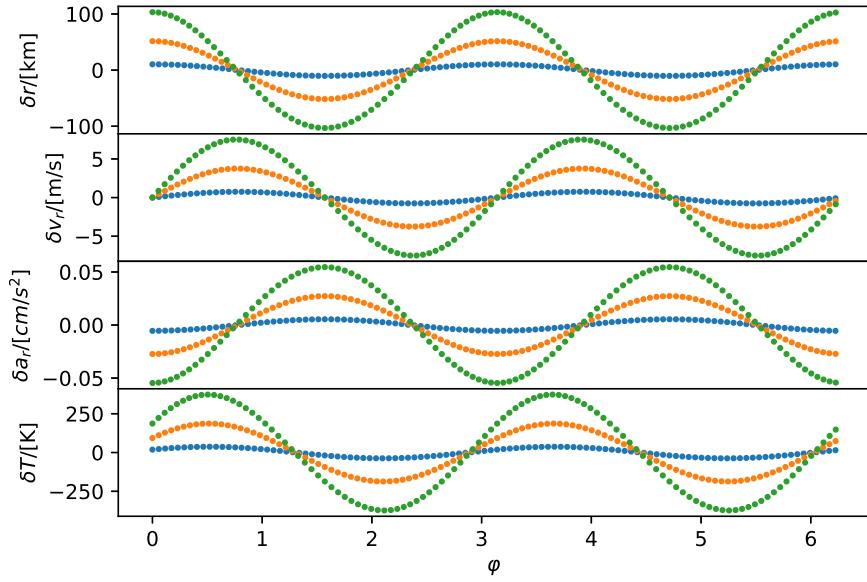


Fig. 27: Cross-section of the equatorial region of the solar model pulsating in a quadrupole sectoral mode with  $\ell = 2$  and  $m = 2$  and frequency  $f = 1 d^{-1}$  for different values of the velocity amplitude  $\tilde{v} = 0.1, 0.5$  and  $1.0 \text{ km}.s^{-1}$  indicated by the blue, orange and green points.  $y$ -axis of each figure displays a different perturbed surface quantity (from top to bottom: radial components of the displacement  $\delta r$ , velocity  $\delta v_r$  and acceleration  $\delta a_r$  and finally the perturbation in surface temperature  $\delta T$ ).

gravitational potential is not disturbed, and the surface mass elements do not transfer a significant amount of heat between each other during the period of the pulsations (Buta and Smith, 1979):

$$\frac{\tilde{T}}{T_{\text{eff}}} = \nabla_{ad} \left[ \frac{A_\perp}{A_r} l(l+1) - 4 - \frac{A_r}{A_\perp} \right] \frac{A_r}{r_{eq}}, \quad (57)$$

where  $\nabla_{ad}$  is the adiabatic temperature gradient equal to 0.4 for monoatomic ideal gas. Relative amplitude  $\tilde{T}/T_{\text{eff}}$  can be also set to custom value (see table 8 in Appendix A). The distributions of kinematic quantities and the temperature perturbation on the equator of a solar model pulsating in the quadrupole mode are listed in Figure 27.

## 6 SYNTHETIC OBSERVATIONS

---

# 6 Synthetic observations

Section 4 describes how to construct a complete model of a single or binary star system at a given moment in the form of a data structure called `PositionContainer` (Section 2.2.2). However, such a container does not provide the necessary information about the observer. One of the most important pieces of missing information is the mutual orientation between the observer and the system.

A single `PositionContainer` represents the state of the system at a single moment in time and thus cannot be used to produce time-series observations. Therefore, we solved the issues raised in the previous paragraph by implementing a separate `Observer` module (Section 2.4). Main tasks of the `Observer` class instance are:

1. Storing and managing different passbands that are used to perform synthetic observations.
2. Managing time series defined by the user, used for the production of synthetic observations.
3. Selecting the most efficient method for the synthesis of observables based on the system morphology.

Following sections will discuss the tasks 2 and 3 in detail.

## 6.1 Managing time series

ELISa produces synthetic observations at the specific times or photometric phases defined by the user. In many cases, the synthetic observation module can greatly reduce the total number of unique synthetic observations in which ELISa must evaluate the model. There are numerous methods applicable to certain types of systems that can achieve a reduction in computational time. The most efficient methods for each type of system modelled in ELISa is listed in Table 2.

### 6.1.1 Reduction to a single observation

In the most simple case of a single star system without surface inhomogeneities, we can reduce the calculation of the observable quantity to a single observation due to the axial symmetry of such surface. This simplification becomes very handy while incorporating surface perturbations such as pulsations where the same equilibrium model can be used repeatedly for the whole observation.

## 6 SYNTHETIC OBSERVATIONS

---

Tab. 2: Use cases for methods aimed to reduce the number of model evaluations. Sections 6.1.1, 6.1.2, 6.1.3 discusses the mentioned methods in detail.

Method		Single observation	Phasing	Curve symmetry
Single system	no spots	yes	no	no
	spots	no	yes	no
Binary System	circular	fixed features	no	yes
		moving features	no	no
	eccentric	no spots	no	yes
		spots	no	no

### 6.1.2 Using phased time series

Objects modelled in ELISa usually perform a periodic motion (either orbital or rotational) described by an ephemeris. In cases of the model with stationary surface features, we can phase the desired time series and the duplicate phases  $\phi'$  are possible to evaluate in a single synthetic observation at phase  $\phi$  due to the symmetry:

$$\phi' = n\phi; \quad n \in \mathbb{Z}; \quad \phi \in \langle 0, 1 \rangle \quad (58)$$

Phasing a time series can significantly reduce the number of model evaluations in case of observation spanning many epochs and containing a large number of observations, such as in the case of space-based observations. To fully utilize this feature, we strongly advise the user, wherever possible, to generate phased synthetic observations with sufficient density, which will be subsequently interpolated to match the desired observations.

### 6.1.3 Phase curve symmetry

This specific type of symmetry is applicable in case light curves of binary systems with circular orbits without surface inhomogeneities. Light curves of such systems are symmetrical around photometric phases 0.0 and 0.5. The method is used simultaneously with the approach described in Section 6.1.2 and cuts the number of model evaluations further in half in the following manner:

$$\phi' = 1 - \phi; \quad \phi \in \langle 0, 0.5 \rangle; \quad \phi' \in (0, 1.0). \quad (59)$$

## 6 SYNTHETIC OBSERVATIONS

---

### 6.2 Orientation of the model with respect to the observer

`PositionContainer` constructed in Sections 3 and 4 is generated in the co-rotating frame of reference. Therefore, the vector quantities stored in the container need to be transformed into an inertial reference frame aligned with the observer. Such transformation should enable to perform synthetic observations easily from a position of an observer, which will be fixed in such an inertial reference frame. In the case of ELISA, the line-of-sight vector coincides with the direction of the negative  $x$ -axis (i.e. the observer is fixed at  $x=-\infty$ ). The described reference frame is consistent with the convention where the radial velocity of a body moving towards the observer has a negative value. This arrangement also provides a convenient way of obtaining radial components of vector quantities as the negative  $x$ -component of a vector. Moreover, the cosine of the incidence angle between line-of-sight and surface element normal vector is obtained in the same manner.

The orientation of the `PositionContainer` in 3D space is stored in a Python data structure called a named tuple `Position` containing the following identifiers:

- `idx`: unique index characterizing the observation within the time-series,
- `distance ( $d$ )`: distance between the components in SMA units. This identifier is not used for `SingleSystem`.
- `azimuth ( $A$ )`: angle between  $x$ -axis of co-rotating frame of reference and  $y$ -axis in the fixed observer's reference frame. In the case of binary systems, the azimuth  $A$  can be expressed as:

$$A = \nu + \omega, \quad (60)$$

where  $\omega$  is an argument of periastron (see Figure 28 for more information). In the case of a single star system, the azimuth is defined as an angle of rotation:

$$A = 2\pi \frac{t - T_0}{P}, \quad (61)$$

of a star from its starting position at  $T_0$ .

- `true_anomaly ( $\nu$ )`: true anomaly of the secondary component. This identifier is not valid for `SingleSystem`.
- `phase`: photometric phase of the system defined by the system's ephemeris.

The `Position` is used to reorient the `PositionContainer` into the final position. The transformation of the `PositionContainer` from a co-rotating frame into the observer's reference frame consists of two rotations. In the first step, the system is rotated

## 6 SYNTHETIC OBSERVATIONS

---

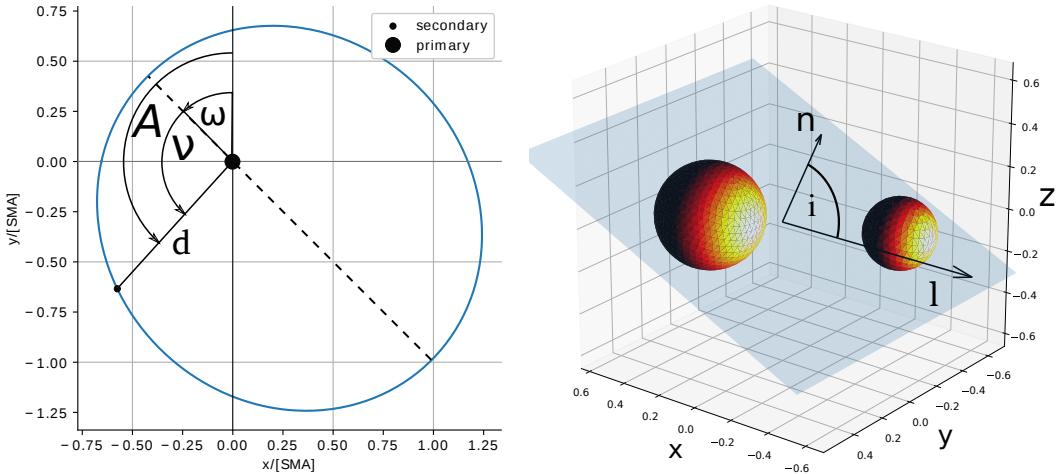


Fig. 28: The left panel illustrates the orbit orientation in the orbital plane where the  $y$ -axes of the orbital plane and the observer's reference frame coincide. The panel on the right side shows the orientation of the system in 3D space with highlighted line-of-sight vector  $\mathbf{l}$ , orbital plane normal vector  $\mathbf{n}$  and inclination  $i$ . Colourmap on the component's surfaces corresponds to the bolometric radiance in the direction towards the observer.

around the  $z$ -axis by the angle  $\delta\phi$  in the counter-clockwise direction:

$$\Delta\phi = A + \frac{\pi}{2}. \quad (62)$$

In the case of the binary systems, this rotation puts the orbit into the orientation indicated on the left side of Figure 28. In the second step, the system is tilted into the final position by the rotation around the  $y$ -axis in the clockwise direction by the amount:

$$\Delta\delta = \pi - i. \quad (63)$$

After the transformation, the orbital (rotational) motion for inclinations  $0^\circ$  to  $90^\circ$  appears in the counter-clockwise direction from the vantage point of the observer (Kallrath and Milone, 2009) and the  $y$ -axis coincides with ascending node where the component crosses the plane-of-sky towards the observer.

Finally, the system container is reoriented according to the corresponding Position, and it is now ready to produce corresponding synthetic observation.

### 6.2.1 Light-travel time effect in the observer's reference frame

The light-travel time effect (LTT) occurs due to the change in distance between the observed object and the observer, thus modifying the time of travel of the light beam. The LTT affects primarily periodically occurring phenomena such as the timing of

## 6 SYNTHETIC OBSERVATIONS

---

eclipses in multicomponent systems (Fedurco and Parimucha, 2018) or pulsations on a surface of a star performing an orbital motion (Shibahashi and Kurtz, 2012).

As a result, the timing of the periodic event will be modulated by the amount  $t_{ltt}$  that will correspond to the additional path that light has to cross. In the case of binary systems with a pulsating component, the time modulation for primary and secondary component  $t_{ltt,p}$ ,  $t_{ltt,s}$  can be expressed as:

$$\begin{aligned} t_{ltt,p} &= -\frac{d}{c} \frac{q}{1+q} \cos\left(A - \frac{\pi}{2}\right) \sin i, \\ t_{ltt,s} &= \frac{d}{c} \frac{1}{1+q} \cos\left(A - \frac{\pi}{2}\right) \sin i, \end{aligned} \quad (64)$$

where  $c$  is the speed of light in a vacuum. The resulting corrected time  $t - t_{ltt}$  is used to generate a solution to perturbation of the surface quantity in Equation 46. In conclusion, correction  $t - t_{ltt}$  represents a time in the reference frame of the pulsating star that will produce an observation corresponding to a time  $t$  in the observer's reference frame.

### 6.3 Visibility of the surface elements

Modelling of observable quantities from three-dimensional objects requires an ability to evaluate the visibility of surface elements from the observer's point of view. Determination of visibility of the surface elements belongs to one of the most challenging issues in the process of modelling binary systems. Improper treatment of this issue will significantly affect the precision of the observables. Establishing the visibility of the surface element can be split into two major tasks discussed in Sections 6.3.1 and 6.3.2.

#### 6.3.1 Selecting the surface elements facing the observer

In general, only the side of the object facing the observer can contribute to the calculation of the observable quantities. For the arbitrary orientation of the object, the most straightforward way to determine whether the surface element is oriented towards the observer is to examine the following condition for the cosine of the angle  $\theta_i$  between the surface element normal  $\mathbf{n}$  and the line-of-sight vector  $\mathbf{l}$ :

$$\cos \theta_i \in (0, 1) \quad (65)$$

(see Figure 31 for further information). Orientation of the observer's reference frame described in Section 6.2 ensures that verifying condition 65 is computationally inexpensive task.

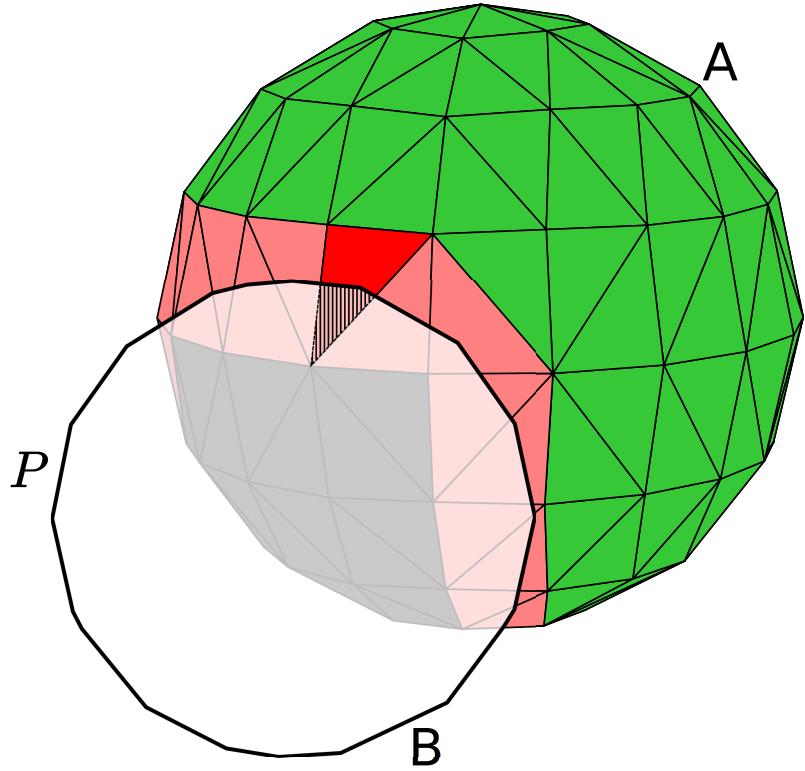


Fig. 29: This diagram illustrates the process of determining the visibility of surface elements of star A partially covered by star B during the eclipse. Green colour designates surface elements fully visible to the observer, whereas grey elements represent hidden surface elements. Pink (and red) are partially visible surface elements based on the algorithm described in Section 6.3. As an example, the red surface element illustrates the treatment of partially visible surface elements. In this case, the striped area marking the outline  $P$  and surface element overlap is calculated. Finally, we can calculate the area visible to the observer.

### 6.3.2 Determination of the surface visibility during the eclipse

During the orbital motion, components of eclipsing binary obscure each other periodically from the observer's perspective. In such cases, certain parts of the eclipsed component are not visible despite them fulfilling the condition defined in Equation 65. In such cases, we have to disregard the contributions of the obscured surface elements while calculating observables. The following paragraph discusses how to discern visible surface elements during eclipses.

From now on, let us refer to the eclipsed component as the star A and the component obscuring the star A as the star B. Initially, a model of the binary system rotated into the reference frame of the observer (see Section 6.2) is projected into the  $yz$  plane. Afterwards, a polygon  $P$  - the convex hull of projected points of star B acts as its horizon. The following approach depends on the comparative sizes of the surface

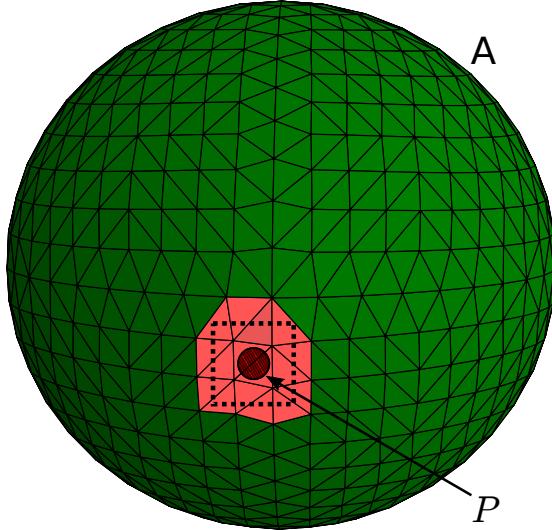


Fig. 30: Treatment of stellar eclipses produced by compact objects with a similar or smaller diameter than the size of the surface elements of the eclipsed star. Colour schema and designations are identical as in Figure 29.

elements of the star A and the outline P.

The first case will outline the situation illustrated in Figure 29, where the outline P is sufficiently larger than the average size of the surface element. In such a case, projected points of star B facing the observer are sorted into two groups lying either inside or outside the outline  $P$ . The sorting is performed by an efficient algorithm implemented in matplotlib<sup>2</sup> package (Hunter, 2007). Afterwards, the surface elements are divided into three groups:

- *Visible*: each point of the given surface element lies outside of the outline  $P$ . Therefore, the whole area of the surface element is visible to the observer.
- *Obscured*: each point of the given surface element lies inside the outline  $P$ . Therefore, the whole surface element is entirely obscured by star B. This surface element does not contribute to the calculation of the observable.
- *Partially visible*: surface element contains points both inside and outside of the outline  $P$ .

Surface elements designated as partially visible require further attention. The surface area (projected into the  $yz$  plane) visible to the observer is determined indirectly by calculating the surface area of the overlap between the projection of the surface

---

<sup>2</sup>[https://matplotlib.org/3.1.3/api/path\\_api.html](https://matplotlib.org/3.1.3/api/path_api.html)

## 6 SYNTHETIC OBSERVATIONS

---

element and the outline  $P$ . This process is illustrated in Figure 29 where the overlap area of the partially eclipsed triangle is marked with stripes and the remaining visible part with red colour.

The second scenario concerns compact objects orbiting relatively large companions with similar or larger surface elements compared to the diameter of the compact star. Such a case is illustrated in Figure 30 and it demonstrates a case where neither of the projected points of the star B lies inside the outline  $P$  even though the stars are clearly in eclipse. The previous algorithm would erroneously identify no obscured or partially visible surface elements. Therefore, a search box (dotted square in Figure 30) with twice the size of star A surface elements is formed around the centre of the outline  $P$ . Subsequently, surface elements with at least one point inside the box will be treated as partially visible. Finally, the surface elements flagged by this procedure as partially visible but outside the outline  $P$  will still contribute fully to the outgoing flux. The reason for that is the mentioned overlap algorithm will find no overlapping area between the outline  $P$  and the surface element in question.

Finally for both cases, the visible area is corrected for the tilt of the surface element from the line-of-sight vector. The resulting value is used to calculate outgoing flux from the given surface element. As a result, the described treatment of the partially visible surface elements significantly increases the numerical precision of the computed LC for the given level of the surface discretization (i.e. the given number of surface elements).

### 6.4 Light curve integration

Perhaps the most helpful feature of the package ELISa is modelling light curves of single stars and close binary systems. Photometry is, after all, a very affordable and information-dense data messenger. Photometric observations of the variable stars, such as close binary stars or pulsating stars, provide valuable insight into the physical processes causing the variability.

ELISa calculates and stores normal radiances  $L_{n,i}$  for each surface element photometric passband, as described in Section 4.5. Subsequently,  $L_{n,i}$  is used to calculate the *radiant intensity* (ISO, 2019)  $I$  in the direction to the observer.  $I$ , also known as *photometric flux*  $F$  in astrophysics, can be obtained by the summation of  $N'$  partial intensities  $I_i$  for each at least partially visible surface element  $i$ :

$$I \equiv F = \sum_{i=1}^{N'} I_i, \quad (66)$$

where the corresponding normal radiance  $L_{n,i}$ , visible surface area  $\Delta S_i$  and limb dar-

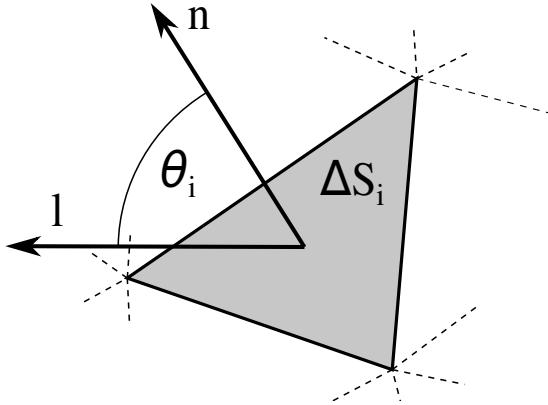


Fig. 31: This diagram clarifies the variables used in the Equations 65 and 67 used to determine the orientation of the surface element and calculate the emergent radiant intensity  $I_i$  in the direction to the observer  $l$ . Vectors  $l$  and the surface element outward-facing normal vector  $n$  form the angle  $\theta_i$  that reduces the cross-sectional area visible in the direction of the observer and also governs the extent of the limb darkening  $\mathcal{L}$ . Parameter  $\Delta S_i$  describes the area of the surface element.

kening factor  $\mathcal{L}_\lambda$  define the partial radiant intensity  $I_i$ :

$$I_i = \mathcal{L}_\lambda(\theta_i) L_{n,i} \Delta S_i \cos \theta_i \quad (67)$$

Figure 31 illustrates the calculation of  $I_i$  in the direction of the observer  $l$ .

## 6.5 Radial velocities

Determination of radial velocities from spectroscopy provides another useful window into the research of stellar variability. In eclipsing binaries, the constraints provided by the combination of radial velocity and photometric observations are enough to derive a complete set of binary system parameters.

ELISa performs synthetic observations of radial velocities using two different methods: kinematic and radiometric. They significantly differ in their computational speed and their use cases. The following sections provide detailed descriptions of both methods.

### 6.5.1 Kinematic method

The kinematic method assumes the measured radial velocity of the star to be identical with the radial component of the centre-of-mass velocity. In such case, the radial component of centre-of-mass velocities  $v_p$ ,  $v_s$  for the primary and secondary component can

## 6 SYNTHETIC OBSERVATIONS

---

be obtained analytically from the solution of the two-body problem (Hilditch, 2001):

$$v_p = 2\pi a \sin(i) \frac{q}{1+q} \frac{\cos(\nu + \omega) + e \cos(\omega)}{P\sqrt{1-e^2}} + \gamma, \quad (68)$$

$$v_s = -2\pi a \sin(i) \frac{1}{q+1} \frac{\cos(\nu + \omega) + e \cos(\omega)}{P\sqrt{1-e^2}} + \gamma. \quad (69)$$

This method is very helpful in binary systems where the rotation (the Rossiter - McLaughlin effect) and surface features (spots, pulsations) are not significant enough to waste computational resources. The main advantage of this method is a substantial increase in computational speed compared to the radiometric method. A fast approach to modelling radial velocities is convenient during solving the inverse problem discussed in Section 8. Depending on the optimization method, the inverse problem may require many model evaluations to produce a solution. In this and similar cases, the computational time saved can be significant.

### 6.5.2 Radiometric method

Surface inhomogeneities and rotation of the binary may significantly alter the measured radial velocities. In such cases, a prediction provided by the kinematic method is no longer adequate. As illustrated in the example provided in Figure 32, a more sophisticated approach to the modelling of radial velocities is sometimes required for high-precision measurements of radial velocities.

The radiometric method implemented in ELISa calculates the resulting radial velocity  $v_r$  from radial velocities  $v_{r,i}$  of all at least partially visible surface elements. Each contributing velocity is weighted by the intensity  $I_i$  of the corresponding surface element in the direction towards the observer (Equation 3.5.4 in Kallrath and Milone (2009)):

$$v_r = \frac{\sum_{i=1}^{N'} I_i v_{r,i}}{\sum_{i=1}^{N'} I_i}. \quad (70)$$

Intensity  $I_i$  is calculated in a narrow wavelength interval specified by the configuration parameter `settings.RV_LAMBDA_INTERVAL`.

## 6 SYNTHETIC OBSERVATIONS

---

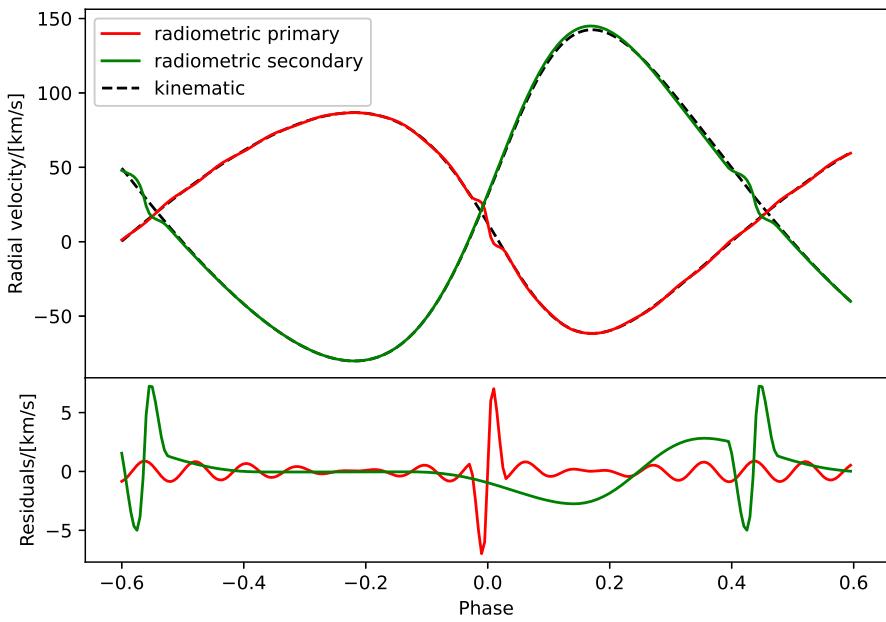


Fig. 32: Impact of rotation, spots and oscillations on the shape of the radial velocity curve. This example shows predicted radial velocities of the eclipsing binary with tidally locked dipole mode ( $\ell=1, m=0$ ) and amplitude of 1 km/s on the primary component and a cool spot on the secondary component using both kinematic and radiometric methods. Red and green radial velocity curves correspond to the primary and secondary component.

## 7 Integration techniques for synthetic observations

In general, each point in a time-series observation represents a separate measurement of the observable quantity produced from a unique model of a system. Each model has an unique orientation in the observer's reference frame described by the `Position` named tuple (see Section 6.2). Therefore, a separate `PositionContainer` encapsulating a complete model of the system is required for each synthetic observation in a time series defined by the user. Information necessary to construct such a container is carried in the `Position` tuples generated for each observation.

Selecting the most suitable way to construct a model is crucial to the increased efficiency of the algorithm. The model can be built from scratch, or it can be adapted from a different `PositionContainer`, saving a substantial amount of computational time. ELISa contains several optimizations listed in Table 3 along with their use cases. The following sections provide a detailed description of the mentioned methods. The method based on using a single model is applicable for single and certain types of binary system with circular orbits. The remaining methods are designed to speed up synthetic observations for binary systems with eccentric orbits.

### 7.1 Integration using a single model

The models of single star systems and binary systems with circular orbits do not change with the time or rotational/orbital phase. This package utilizes the time-independent nature of such surfaces to accelerate the calculation of synthetic observations substantially. The integrator builds only one model of a single star system from scratch, which is subsequently rotated into the positions corresponding to desired observational times. This method significantly accelerates the calculation of synthetic observation

Method		Single model	Inter- polation	Orbital symmetry	Similar neighbours
Single system		yes	no	no	no
Binary	circular	fixed features	yes	no	no
		moving features	no	no	no
System	eccentric	no spots	no	yes	yes
		spots	no	no	no

Tab. 3: Use cases for methods implemented to optimize the calculation of synthetic observations. Sections 7.1, 7.2.1, 7.2.2 and 7.2.3 discuss the mentioned methods in detail.

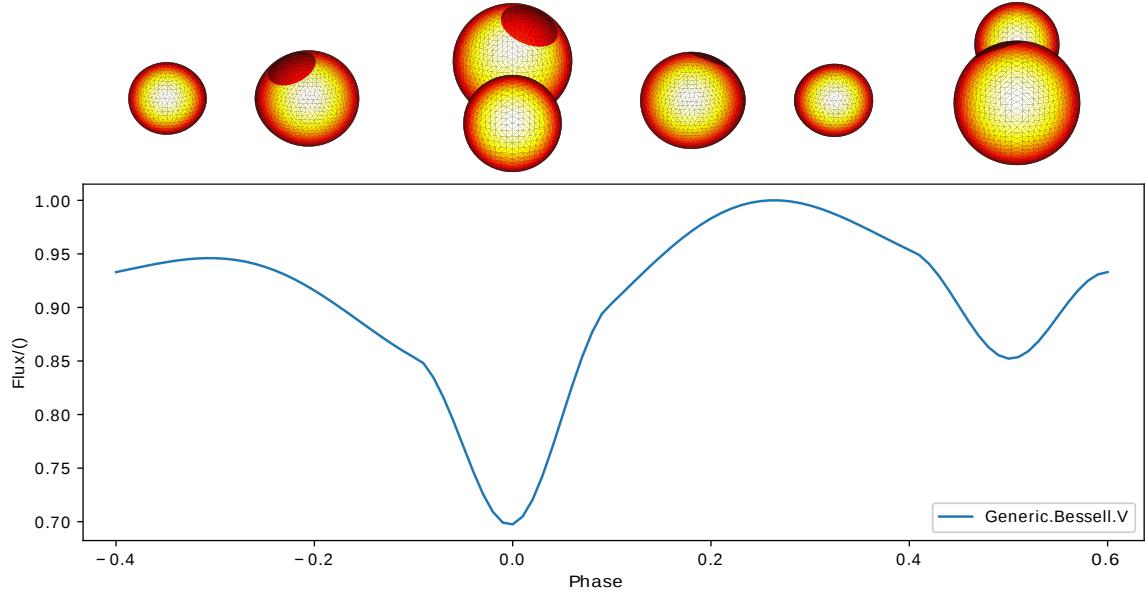


Fig. 33: The figure displays a light curve of a binary system with circular orbit, synchronous rotation and spot on the primary component. The light curve was obtained by rotation of a single binary model and subsequent integration of the flux in the observer's direction. Binary models in the top row are shown in orientations corresponding to photometric phases -0.25, 0.0, 0.25, 0.5.

time-series and can be used for models containing spots.

Binary systems with circular orbits also conserve their surface geometry during the orbital motion. Therefore, we can use the single model approach in cases where surface features remain fixed in the co-rotating frame of reference (see Figure 33). Unfortunately, spots on asynchronously rotating binary components need to be re-incorporated into a clean surface during each phase. Thus, the single-model method cannot be applied to this type of binary.

## 7.2 Synthetic observations of binaries with eccentric orbits

Calculating observables of binary models with eccentric orbits suffers from the necessity to recalculate surface geometry due to a change in component's distance during the orbital motion. Additionally, the corrections to the surface potentials of the components need to be derived by numerical means to conserve volumes of the components. These issues significantly slow down the evaluation of synthetic observations compared to binaries with circular orbits discussed in Section 7.1. However, the following three approximations applicable for eccentric binaries without spots can significantly accelerate the calculation of synthetic time-series observation while maintaining a reasonable precision.

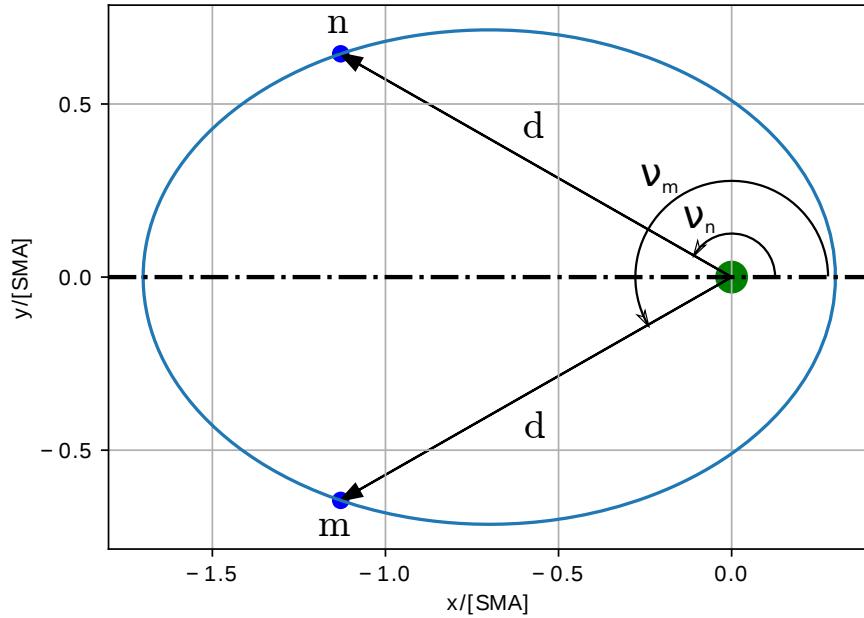


Fig. 34: This figure displays an orbit of a binary star with a pair of symmetrical orbital positions that share the same surface geometry and the component's distance  $d$ . True anomalies  $\nu_n$  and  $\nu_m$  satisfy condition in Equation 71.

The approximation methods described in 7.2.1 and 7.2.2 utilize the symmetry of an eccentric orbit along its apsidal line. In such case, the surface geometries and thus the binary models are identical for the orbital positions  $n$  and  $m$  if their true anomalies  $\nu_n$  and  $\nu_m$  satisfy the following condition:

$$\nu_m = 360^\circ - \nu_n, \quad (71)$$

(see Figure 34 for clarification). The symmetry 71 between orbital positions  $n$  and  $m$  means that corresponding binary models without surface inhomogeneities are identical and thus interchangeable. The binary model for symmetrical orbital positions can be calculated once in the co-rotating frame of reference and then rotated into the final positions in the observer's reference frame.

The main issue arises from the fact that orbital positions corresponding to the user-defined time series are generally not aligned symmetrically around the apsidal line. Following Sections 7.2.1 and 7.2.2 describe two different solution to this issue.

### 7.2.1 Interpolation between symmetrical orbital positions

In this approximation method, also called the interpolation approximation, we solved the asymmetry of orbital positions around the apsidal line with the following process. Initially, the orbital positions were split into halves using the apsidal line. Subsequently, orbital positions from one side, indicated with blue points in Figure 35, were

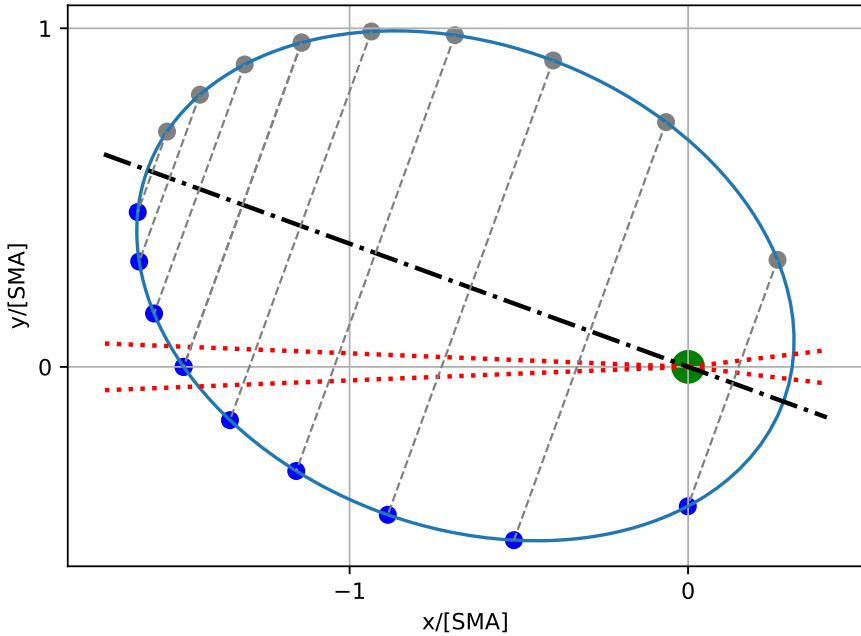


Fig. 35: Visualization of the interpolation approximation described in Section 7.2.1. The method is used for the integration of synthetic observations of eccentric orbits. Orbital positions located on one half of the orbit (blue points) are mirrored on the symmetrical counterpart of the orbit indicated by the grey points. The red dotted points indicate eclipse boundaries.

mirrored across the apsidal line indicated by the grey points. Each symmetric couple (connected by the grey dashed line) shares the same surface geometry and thus can be created in a single step. Afterwards, synthetic observations are produced using symmetrically coupled orbital positions indicated by blue and grey points in Figure 35. Finally, the synthetic observations at the user-defined times were interpolated. The interpolation utilized cubic splines between the synthetic observations calculated from the symmetrical orbital positions.

The shortcoming of this approximation is a requirement of sufficient density of the LC points that ensures the validity of the interpolation. Therefore, we introduced constraints to judge whether the approximation can be used while maintaining an acceptable level of precision. One of them is the maximum separation of the adjacent orbital positions in terms of true anomaly defined by the configuration parameter `settings.MAX_NU_SEPARATION`. If the condition is satisfied, the density of points is sufficient to prevent the appearance of unwanted artefacts as a result of interpolation on a sparsely sampled data set. In the opposite case, where the number of points does not satisfy the condition, this approximation is not used.

Eclipses deserve special attention during the evaluation of the viability of this approximation. Parts of the orbit located between the eclipse's outer and inner contact

point produce the most rapid changes in the observed quantities. Therefore, a synthetic observation should contain at least a predefined minimum number of points in the descending and ascending portion of the eclipse to ensure the precision of the interpolated data points. The minimum number of synthetic observations inside the critical parts of the eclipse is defined in the configuration parameter `settings.MIN_POINTS_IN_ECLIPSE`. If the minimum number of points is satisfied, the approximation is performed according to the description above. In the opposite case, the points within the eclipse are calculated exactly. They are then appended to the rest of the points where the approximation was implemented.

We evaluated the overall precision of the interpolation approximation in Section 7.2.4. The evaluation process adopted the default configuration values used by the ELISa.

### 7.2.2 Assumption of identical geometry on nearly symmetrical orbital positions

The symmetrical counterparts approximation attempts to find pairs of orbital positions from the opposite sides of the apsidal line closest to satisfying Equation 71. Coupled orbital positions share the same model of the binary system that is subsequently rotated into the respective symmetrical orbital positions to produce corresponding synthetic observations.

Compared to the interpolation approximation discussed in Section 7.2.1, this approximation improves the precision during eclipses since the symmetrical synthetic observation is calculated directly from the model at the desired orbital position instead of being interpolated from observations on adjacent orbital positions. The main disadvantage of this approximation is that the paired orbital positions do not perfectly satisfy the Equation 71. Thus, the symmetrical counterpart binary model does not perfectly correspond to the actual surface geometry.

We, therefore, evaluated relative change in surface geometry and mutual irradiation of components in the paired orbital positions. This step is made to ensure that the use of a single model for both orbital positions will not significantly impact the precision of the synthetic observations. The condition for the relative change in the surface geometry is evaluated using the conservative estimation of the change in the outgoing photometric flux  $\delta F$ . Flux change is estimated using the difference between the forward

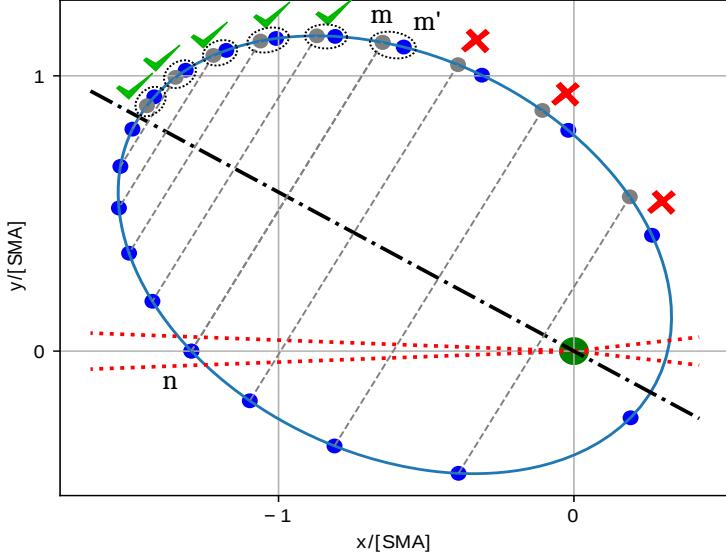


Fig. 36: Visualization of the symmetrical counterparts approximation described in Section 7.2.2. The method searches for the closest symmetrical counterparts satisfying Equation 71 and then evaluates the similarity of mirrored orbital position  $m'$  and user-defined position  $m$  and. A single binary model is used for both orbital positions if the similarity in the surface geometry and mutual irradiation between the orbital pair members is sufficient. The red dotted lines indicate eclipse boundaries.

radii  $\delta r_{n,m'}$  of the components in the orbital positions  $n$  and  $m'$ :

$$\frac{2r_{eq} * \delta r_{n,m'} + \delta r_{n,m'}^2}{\sum r_{eq}^2} \leq \delta F, \quad (72)$$

$$\delta r_{n,m'} = |r_{forward,n} - r_{forward,m'}|,$$

where summation in the denominator goes through both components and  $\delta F$  is governed by the configuration parameter `settings.MAX_D_FLUX`. Orbital position  $m'$  forms a similar orbital position couple with orbital position  $m$  that satisfies Equation 71.

The second constraint takes into account the relative change in mutual irradiation that has a significant effect on the synthetic observations, especially photometric flux (as demonstrated in Section 4.3.2). The change in the distance  $d$  of the components between the orbital positions  $n$  and  $m'$  will affect the flux arriving from the companion. Therefore, we introduced the constraint to the difference between fractions of the surface integrated irradiance  $L_{irr}$  to the total luminosity of the system  $L$ :

$$\left| \left[ \frac{L_{irr}}{L} \right]_{m'} - \left[ \frac{L_{irr}}{L} \right]_n \right| \leq \delta F. \quad (73)$$

The  $L_{irr}/L$  ratio describes a ratio of irradiated energy and the energy outputted by the star. Ratio  $L_{irr}/L$  also represents a surface-integrated variant of the  $E/M_0$  term in the Equation 28 describing an amplitude of the reflection effect. Subsequently, the

irradiation from the stellar companion  $L_{irr}$  can be estimated as a fraction of the companion's luminosity captured by the irradiated star based on the components distance  $d$ , equivalent radii  $r_{eq}$  and effective temperatures of the components  $T^{eff}$ . Therefore, the ratio  $L_{irr}/L$  can be approximated for the “target” component  $t$  while being irradiated by the “source” component  $s$ :

$$\frac{L_{irr}}{L} \approx \left( \frac{r_{eq,s}}{d} \right)^2 \left( 1 + \left( \frac{r_{eq,s}}{r_{eq,t}} \right)^2 \left( \frac{T_s^{eff}}{T_t^{eff}} \right)^4 \right)^{-1} \quad (74)$$

Constraints put on the orbital position pairs ensure that the errors of the synthetic observations are below the pre-defined threshold. Section 7.2.4 provides the evaluation of precision and speed relative to the exactly calculated model for the symmetrical counterparts approximation along with the interpolation approximation.

### 7.2.3 Assumption of similar geometry on neighboring orbital positions

Binary stars with eccentric orbits spend a majority of the time near the apastron of their orbits. Therefore, observations equidistant in time tend to over-sample parts of the orbit with the maximum separation of the components where the orbital motion is the slowest. As a result, the binary components undergo the slowest change in surface geometry and mutual irradiation near the apastron passage. A similar neighbour approximation is supposed to take advantage of this property by searching for similar binary models on adjacent orbital positions satisfying conditions for similar orbital positions listed in Equations 73 and 74. Figure 37 illustrates the resulting distribution of similar orbital positions across the example of highly eccentric orbit.

The similar neighbour approximation also proves extremely useful in cases with a high density of synthetic observations and cases of binary systems with an eccentricity close to zero. For near-circular orbits, the surface geometry does not change rapidly nor significantly during the orbital motion. Thus, a complete recalculation of the binary for each orbital position is wasteful.

The main advantage of this approximation is the ability to combine the similar neighbours approximation with the interpolation approximation or symmetrical counterparts approximation which creates a compounding effect on the saved computational time. Therefore, unless directed otherwise, ELISa always attempts to utilize the similar neighbours approximation along with the previous methods. For this reason, the precision versus speed benchmark performed in Section 7.2.4 already uses the similar neighbours approximation along with interpolation and symmetrical counterparts approximation.

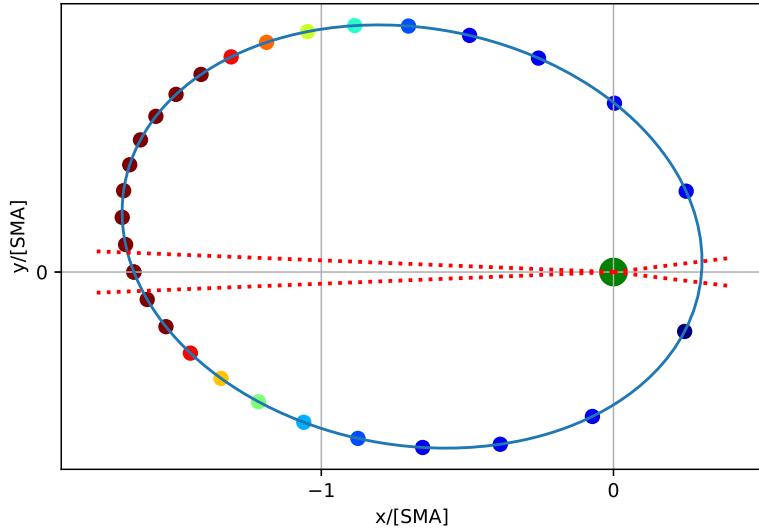


Fig. 37: Visualization of the similar neighbor approximation where the similar orbital positions satisfying Equations 73 and 74 share the same color.

#### 7.2.4 Precision versus speed comparison

The efficiency of approximation methods mentioned above depends on multiple factors such as density of synthetic observations, eccentricity, filling factors of binary components, etc. The interpolation approximation is slightly more efficient for synthetic observations with higher point densities. On the other hand, the similar counterparts approximation performs well for binaries with orbital eccentricities above 0.6, where the interpolation approximation generally cannot be implemented due to the large angular separation of the orbital positions near the periastron. Additionally, approximation utilizing adjacent orbital positions can be stacked with either remaining method and thus provide a compounding effect on the resulting reduction of the computational time.

We evaluated the performance of the approximation methods on a sample of 10 000 randomly generated binary systems. Eccentricities, arguments of periastron were drawn from their full range, i.e.  $e \in \langle 0, 1 \rangle$ ,  $\omega \in \langle 0, 360 \rangle^\circ$ . The inclinations were picked from the interval  $\langle 50, 90 \rangle^\circ$  and mass ratio was restricted to the interval  $(0, 1)$ . The benchmark drew both surface potentials from values below 50, with the lower boundary being the component's critical potential at periastron, where the component fills its Roche lobe. Discretization factors of both components were set to 5 during the whole procedure. Subsequently, the synthetic observations of such systems were generated using interpolation approximation and symmetrical counterparts approximation. Random number of  $N \in \langle 100, 900 \rangle$  equidistantly spaced synthetic observations were calculated on phase interval  $\langle 0, 1 \rangle$ . Both methods were also supplemented with the similar neighbour approximation as implemented in ELISa code. The results were then compared to the exactly

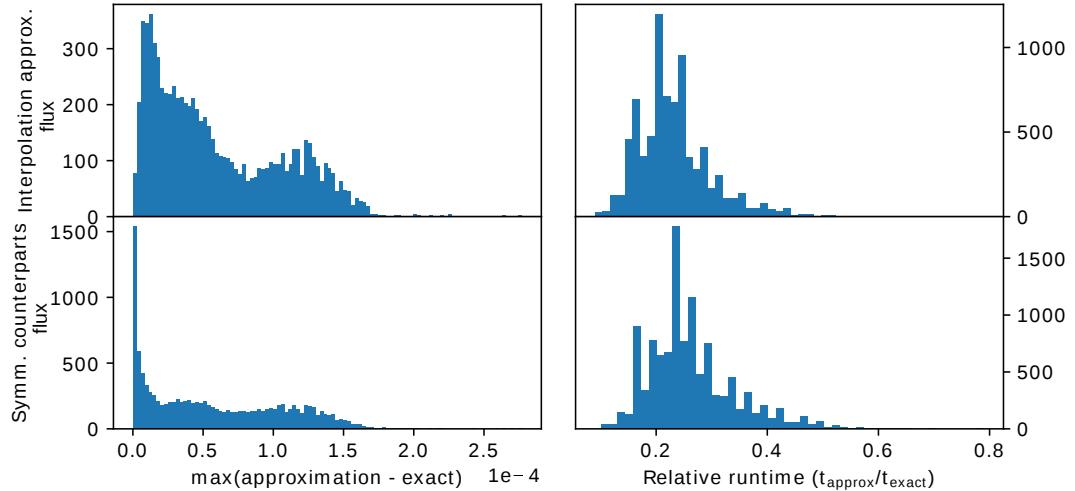


Fig. 38: The figure shows a set of histograms evaluating the precision (left column) and relative speed (right column) of the approximations used while calculating the light curves of binary systems with eccentric orbits. Combined effect of the interpolation and similar neighbors approximation (Sections 7.2.1 and 7.2.3) were evaluated in the top row whereas bottom row describes the performance of the symmetrical counterparts and similar neighbors approximation (Sections 7.2.2 and 7.2.3). We evaluated the precision and speed of both approximations on a sample size of approximately 10 000 randomly generated binary systems.

calculated light curve of the same binary system. The left panel in Figure 38 shows the distributions of maximum deviations of approximated observations from observations calculated exactly for interpolation approximation and symmetrical counterparts approximation. Detailed distribution of the deviations along the parameters  $e$ ,  $i$ ,  $q$ ,  $\omega$  and  $N$  are displayed in the Appendix D. Results show that an overwhelming portion of the modelled LCs did not deviate from the exactly calculated reference synthetic observation by more than  $2 \times 10^{-4}$  in normalized photometric flux. On the other side, the right panel of Figure 38 points to the quadrupling of the mean computational speed if the approximation was utilized. Precision  $2 \times 10^{-4}$  reached by this benchmark is similar to the precision achieved by the currently used surface discretization method discussed in Section 3. Such performance makes this set of approximations, where applicable, a very useful tool for accelerating the evaluation of synthetic observations of binary systems with eccentric orbits, which usually require the largest amount of computational power.

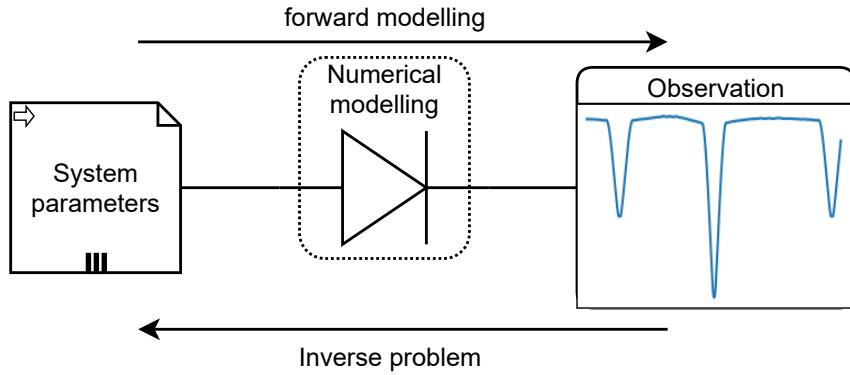


Fig. 39: Numerical modelling provides a one-way capability to produce synthetic observations based on the system parameters and our current understanding of such objects. However, the same numerical model is not usable in the opposite direction: inferring system parameters from the available observations.

## 8 Inverse problem

Sections 3 - 7 discussed in great detail the process of constructing the model and subsequent integration of synthetic observations based on the set of user-defined parameters. Due to the complexity of the task, this is done almost exclusively utilizing numerical modelling. However, the most common problems in astrophysics run in the opposite direction - from observations to object parameters. Only in a very limited number of cases, mostly in theoretical astrophysics, will the researcher inquire what observations a certain set of parameters produce. However, while dealing with astronomical observations, we have access only to a couple of observational quantities (e.g. photometric flux, spectra) that depend non-trivially on the system parameters. The challenge of the inverse problem is illustrated in the diagram located in Figure 39.

### 8.1 Solution to inverse problem

We can express the set of the unknown model parameters defining a system in the form of a  $n$ -dimensional vector  $\mathbf{x}$ :

$$\mathbf{x} = \{x_1, x_2, \dots, x_n\}. \quad (75)$$

The inverse problem can be then defined as a process of finding the combination of parameters  $\mathbf{x}$  that produces the best fitting synthetic observations  $y^{mod}(\mathbf{x})$  to the available observations  $y^{obs}$ . The similarity between the model and observation is evaluated with a cost function depending on the used optimization method. The complexity of the inverse problem increases disproportionately with the number of model parameters  $n$ . In our case,  $n$  ranges approximately from 4 to 12. Primarily, for this reason, the straightforward application of brute force to search the parameter space is out

## 8 INVERSE PROBLEM

---

of the question. Such an approach is currently far beyond the capabilities of modern hardware, and it would be incredibly wasteful on computational resources.

In theory, there are a plethora of optimization methods potentially suitable for the task at hand. However, in this case, two selected optimization methods have been used, and each of them is suitable to perform a slightly different role discussed in detail in the following Sections 8.1.1 and 8.1.2. These methods were selected based on the ability to quickly find the optimal solution and capability to provide a robust estimation of the parameter's confidence intervals.

### 8.1.1 Least Square Trust Region Reflective algorithm

ELISa uses the Least Squares Trust Region Reflective<sup>1</sup> (LSTRR) algorithm implemented in SciPy (Jones et al., 2001). LSTRR algorithm utilizes a local description of the objective function on a specified “trust-region” to search for a local minimum of a function  $\xi^2$  defined as:

$$\xi^2 = \sum_{i=1}^n \frac{[y_i^{obs} - y_i^{mod}(\boldsymbol{x})]^2}{\sigma_i^2}, \quad (76)$$

representing the weighted sum of squares of the residuals.

This optimization method is ideal for performing exploratory searches in the parameters space in the general vicinity of the starting parameters  $\boldsymbol{x}_0$  due to the relatively low number of model evaluations required to arrive at the optimal local solution. LSTRR was selected over the more commonly used Levenberg–Marquardt algorithm (Levenberg, 1944; Marquardt, 1963) or other unbounded methods due to the better performance at larger distances from the optimal solution (Berghen, 2004). Unfortunately, the success rate of the method depends significantly on the initial parameters. Therefore, the user must provide a starting point for the algorithm in a general vicinity of the optimal solution to prevent the algorithm from settling in a sub-optimal local minimum.

### 8.1.2 Markov chain Monte Carlo

Along with the LSTRR method, ELISa also uses a pure Python implementation of the affine-invariant Markov chain Monte Carlo ensemble sampler (Foreman-Mackey et al., 2012) available in the emcee<sup>1</sup> package. In this case, the goodness of the fit is evaluated by the logarithm of the likelihood function  $\ln p$ . The likelihood function was

---

<sup>1</sup>[https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least\\_squares.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html)

<sup>1</sup><https://emcee.readthedocs.io/en/stable/>

## 8 INVERSE PROBLEM

---

defined using an assumption that the observational data  $y^{obs}$  are drawn from a Gaussian distribution around the actual values  $y^{mod}$  with the standard deviation  $\sigma$ :

$$\ln p(y^{obs}|y^{mod}, \sigma) = -\frac{1}{2} \sum_i^n \left\{ \frac{[y_i^{obs} - y_i^{mod}(\mathbf{x})]^2}{\sigma_i^2 + s_i^2} + \ln [2\pi(\sigma_i^2 + s_i^2)] \right\}, \quad (77)$$

where  $s_i$  is an error underestimation parameter that accounts for the fact that the synthetic model cannot explain 100% of the variability present in the observations. Parameter  $s_i$  can be expressed as:

$$s_i = \log(f)y_i^{mod}(\mathbf{x}), \quad (78)$$

where  $\ln f$  is a marginalization parameter fitted alongside the rest of the variables.

Alongside the likelihood function, prior knowledge about the model parameters is also used in the process of accepting or rejecting the generated states of the sampler. Selecting a suitable prior distribution for each variable parameter can greatly aid in producing reliable estimations of the confidence intervals. As for version 0.5, ELISa supports the sampling from uniform or normal prior distribution, which we can use to utilize the prior knowledge about the given parameter (i.e. previous estimation of the parameter using other independent methods). Parameters with uniform priors require to specify the upper and lower boundary of the sampling. On the other hand, the parameters with normal prior distribution are drawn around the most probable value with the expected standard deviation. Additionally, the normal prior distribution can be bounded with the lower and upper boundary similarly to the uniform distribution to prevent the sampler from reaching the undesired areas of the parameter space. The examples of the discussed types of prior distributions are demonstrated in the case of orbital eccentricity in Figure 40.

### 8.2 General guidelines for solving the inverse problem

The user should keep several guidelines in mind during the solution of the inverse problem. They are designed to maximize the chances of the optimizer finding a valid solution and help minimize the computational time.

**Reduce the number of fitted parameters:** An unnecessary high number of the optimized parameters will dramatically increase the volume of parameter space that the optimizer has search through. Therefore, the user should attempt to reduce the number of fitted parameters to a minimum. Parameters that do not significantly affect the shape of the synthetic curve (e.g. gravity darkening factor, albedo, metallicity, etc.)

## 8 INVERSE PROBLEM

---

should be kept fixed at some reasonable value during the initial runs. The primary task of the initial part of the fitting procedure is to search for a parameter combination that reproduces a general shape of the observations. Subsequently, when a rough solution is found, the rest of the parameters can be set as a variable to relax the model into a local minimum. We should stress that parameters possible to obtain from less computationally demanding methods should be calculated apriori and kept fixed during the fit (e.g. ephemeris). Finally, if the parameter is a function of other variable model parameters, the user should utilize the capability to set the model parameter as constrained instead of allowing them variable. One of the examples is the stellar synchronicity  $F$ , which in the case of close eccentric binary systems quickly synchronizes to a value expressed by Equation 6. In the case of variable model parameter  $e$ , Equation 6 can be used as a constraint for stellar synchronicity.

**Combine the optimizers:** Compared to the LSTRR algorithm, MCMC provides an additional capability to provide robust estimates of the parameter's confidence intervals using a posterior distribution of the sampler's accepted states. However, this comes at the price of worse performance in terms of convergence speed to the equilibrium and the necessity to produce a statistically significant sample after the equilibrium has been reached. Therefore, the user should use the MCMC and the LSTRR method together in stages. The LSTRR should be used primarily to search for a local minimum in the general vicinity of the user-provided starting parameters. Subsequently, the parameter space near the found solution can be sampled using the MCMC method to estimate confidence intervals of the fitted parameters.

**Utilize the prior knowledge about fitted parameters:** Improving the reliability of the MCMC-derived confidence intervals can be achieved by propagating the uncertainties of parameters derived from an independent analysis. For example, a mass ratio  $q$  derived from analysis of radial velocities should be used during the MCMC sampled from the normal prior distribution with a corresponding standard deviation. The same holds for estimates of effective temperatures derived from spectroscopy or colour photometry.

### 8.3 AnalyticsTask module

The ELISa package contains an `AnalyticsTask` module responsible for the implementation of the solution to the inverse problem described in the Section 8.1. `AnalyticsTask` module contains dedicated classes for analysis of radial velocity and photometric data

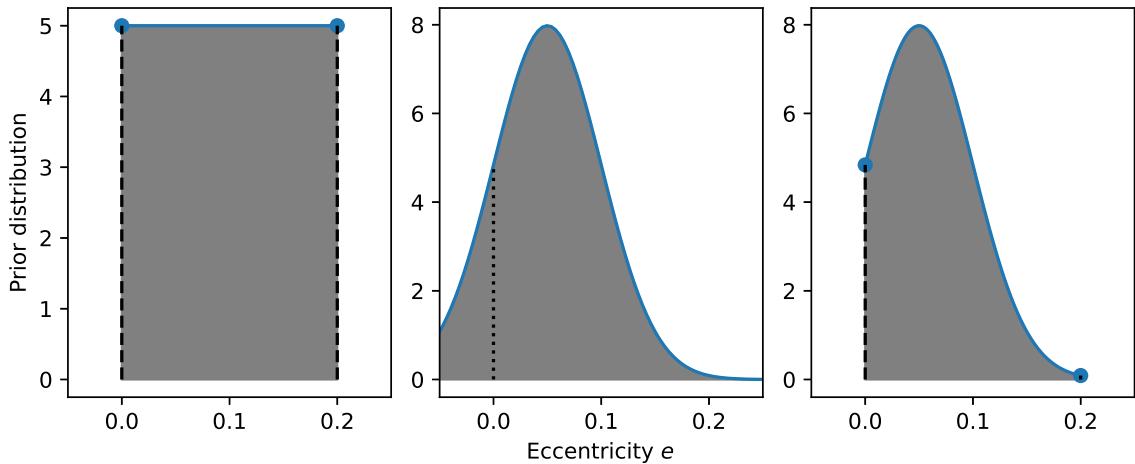


Fig. 40: Example of a uniform (left) and normal prior distribution functions (centre). The normal prior distribution is suitable when the prior estimate of the parameter value is available from an independent analysis. This distribution can be further clipped to avoid sampling invalid or unwanted regions of the parameter space as demonstrated in the panel on the right in case of negative values of the orbital eccentricity.

of binary systems as illustrated in Figure 41. Both fitting classes can utilize LSTRR and MCMC optimizers and can accept user-provided observations as an instances of the `DataSet` module (see Section 2.3). Each fitting class requires input in a dictionary (or json object) containing model parameters to initialize a fitting procedure. Fitting of radial velocities and photometric observations require a different set of model parameters listed in Appendix E.

`AnalyticsTask` module recognizes 3 types of model parameters:

- *variable*: Parameters are actively adjusted by an optimization method to provide a model with the best agreement with the supplied observations.
- *fixed*: Parameters remain constant at a preset value defined by the user.
- *constrained*: The parameter's value depends on the value of one or more variable parameters via a formula defined by the user.

Each fitting parameter is defined as a dictionary of configuration settings that define the parameter's behaviour during a fitting procedure. User can configure each model parameter with the following settings:

- **value**: value of the fixed parameter. Here is an example of a definition of the eccentricity parameter fixed on zero during the fitting of binary systems with circular orbits:

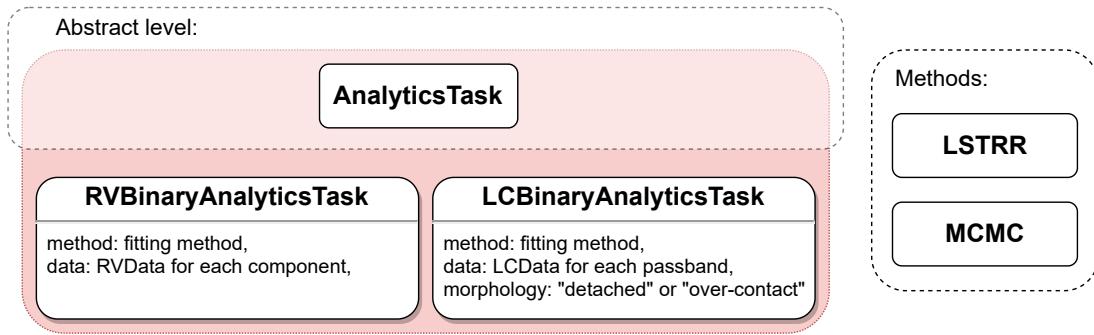


Fig. 41: Overview of the `DataSet` module containing classes for inferring binary model parameters from radial velocity observations and photometric observations. Both classes require to specify optimization methods (right panel) and the input data in the form of `RVData` instances for each component in case of the radial velocities and `LCDATA` instances for each passband in case of photometric data.

```

"eccentricity": {
    "value": 0.0,
    "fixed": True,
    "unit": None
}

```

In the case of variable parameters, this parameter is used as a starting value for the optimizer in the LSTRR method. It also acts as an expected value if the variable is drawn from a normal prior distribution during an MCMC sampling.

- `fixed`: True/False. This parameter defines whether a corresponding model parameter is regarded as fixed or variable.
- `min`, `max`: a lower (upper) boundary for a corresponding variable model parameter. Lets us now demonstrate the definition of an albedo as a variable parameter bounded on a given interval:

```

"albedo": {
    "value": 1.0,
    "min": 0.4,
    "max": 1.0,
    "fixed": False,
    "unit": None
}

```

## 8 INVERSE PROBLEM

---

- **sigma**: the expected standard deviation of a parameter drawn from a normal prior distribution during MCMC sampling. Here is an example of an effective temperature  $T^{eff} = 6700 \pm 400$  K derived from a colour index used as a prior distribution for variable parameter during an MCMC sampling:

```
"t_eff": {  
    "value": 6700,  
    "sigma": 400,  
    "fixed": false,  
    "unit": "K"  
}
```

- **constraint**: defines a relation between constrained parameters and any given variable parameter. As an example, optimizer can utilize prior knowledge about parameter  $a \sin i = 11.3 R_{\odot}$  obtained from radial velocity fit to constrain semi-major axis during a light curve fit. A constraint is provided in a string representation of a mathematical formula:

```
"semi_major_axis": {  
    "constraint": "11.3 / sin(radians(system@inclination))",  
    "unit": "solRad"  
}
```

where the names of the variable parameters are created by concatenating a name of the parent object (system, primary, secondary, etc.) with the parameter name as defined in Appendix E using character @. List of allowed mathematical operators is accessible in the attribute **CONSTRAINT\_OPERATORS** of each optimizer class instance.

- **unit**: defines a unit in which the **value** or **constraint** configuration parameters are supplied.

### 8.4 Determination of parameters in case of eclipsing binary KIC 4851217

We want to demonstrate ELISa's capability to solve an inverse problem on the example of close eclipsing binary KIC 4851217. This object was observed during an initial

Parameter	Value	Status
$q$	$1.08^{+0.03}_{-0.03}$	Variable
$a \sin i/[AU]$	$11.3^{+0.2}_{-0.3}$	Variable
$e$	$0.02^{+0.1}_{-0.1}$	Variable
$\omega/[^{\circ}]$	$230^{+20}_{-30}$	Variable
$\gamma/[km.s^{-1}]$	$-24.8^{+0.9}_{-1.2}$	Variable
$P/[d]$	2.47028	Fixed
$T_0/[BJD]$	2 454 953.900507	Fixed

Tab. 4: This table shows the results of the radial velocity fitting procedure, which corresponds to a resulting radial velocity fit in Figure 42.

observing campaign of the Kepler spacecraft (Borucki, 2016) and included in the third revision of the Kepler Eclipsing Binary Catalog (Kirk et al., 2016). Fortunately, follow-up spectroscopic observations were performed by Matson et al. (2017) that allows us in combination with photometric data to determine absolute parameters of the system.

#### 8.4.1 Fitting of radial velocities

We used the “kinematic” method (Section 6.5.1) to model radial velocities due to precision and the lack of observations near the eclipses. Therefore, instead of full set binary system parameters that would otherwise end up heavily correlated, only five parameters were optimised (see Table 4). The most critical parameters are the mass ratio  $q$  and factor  $a \sin i$  that will later help us determine the dimensions of the eclipsing binary. We kept the ephemeris fixed on the value derived from the timing of primary eclipses in the photometric data. We made this step according to the suggestion in Section 8.2 to reduce problem dimensionality.

In the initial step, the radial velocity (RV) data were processed using a two-step method. In the first step, we used the LSTRR method to find an optimal solution, and the results were used as starting parameters for the second stage. Subsequently, the MCMC method with 40 000 iterations were utilised to sample the vicinity of the obtained solution to estimate confidence intervals from the posterior distribution of the equilibrium section of the MCMC chain after the convergence to the equilibrium. To achieve this goal, we utilised built-in capabilities to discard the chain’s thermalisation stage and calculate the one  $\sigma$  confidence intervals derived from the (16, 84) percentile intervals of the posterior distribution. Resulting RV fit parameters are listed in Table 4 and a posterior distribution of the samples can be found in 49 in Appendix F.

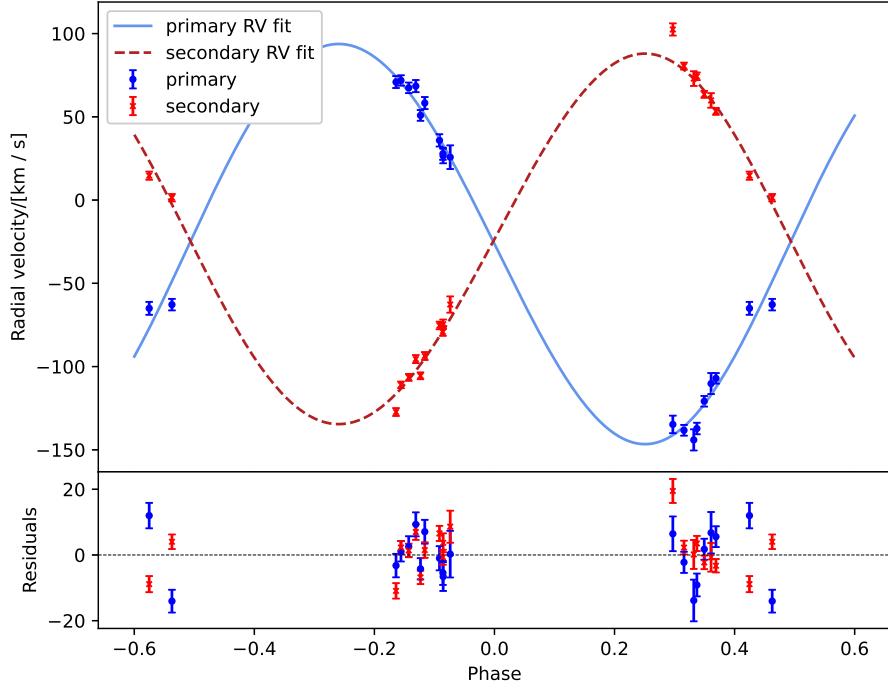


Fig. 42: Radial velocity fit of KIC 4851217 based on the parameters derived in Table 4.

#### 8.4.2 Light curve fitting

Detrended long cadence photometric observations with the pre-whitened pulsations were used to determine the geometry and orientation of the binary model in space. Like in RV fit, we initially used the LSTRR solver to optimize the user-provided set of starting parameters. A few additional steps are described in the following text that accelerated the light curve (LC) fitting process.

In initial runs, six parameter were set as free for the LSTRR optimizer: inclination  $i$ , eccentricity  $e$ , argument of periastron  $\omega$ , secondary component effective temperature  $T_2^{eff}$  and surface potentials  $\Omega_1$ ,  $\Omega_2$ . Initial values of  $e$  and  $\omega$  were provided from the RV fit, and starting values of remaining parameters  $i$ ,  $T_2^{eff}$ ,  $\Omega_1$  and  $\Omega_2$  were found heuristically to match a general shape of the binary system's LC. On the other hand, the primary component's effective temperature was set to  $T_1^{eff} = 6694$  K (Kirk et al., 2016) and stayed fixed during initial runs along with  $q$  derived from the RV fit. Due to the short orbital period, both synchronicities  $F_1$ ,  $F_2$  were kept constrained to  $e$  according to Equation 6 since we expected the rotation of the components to be fully synchronized. Determination of parameter  $a \sin i$  enabled us to also constrain the semi-major axis  $a$  to the orbital inclination  $i$ . Atmospheric parameters, such as albedos and gravity darkening factors were initially fixed to values corresponding to radiative envelope for primary component ( $A_1=1.0$ ,  $\gamma_1=1.0$ ) and convective component for the secondary

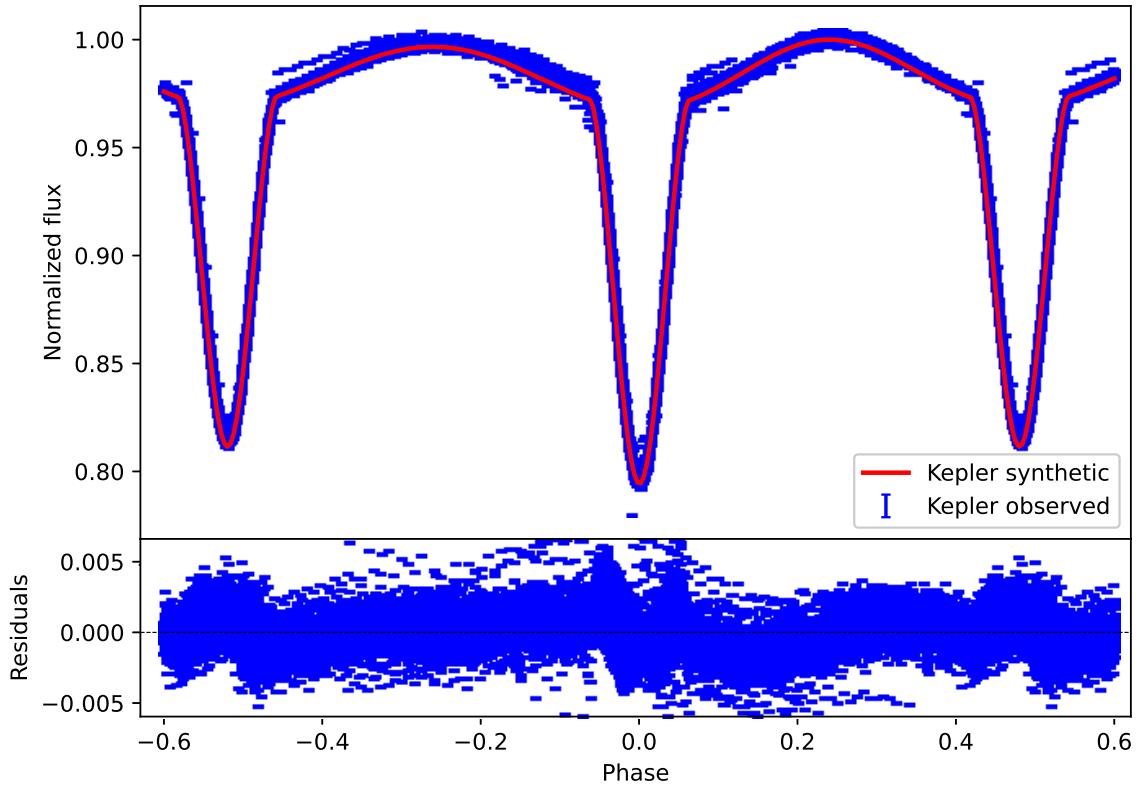


Fig. 43: Light curve fit of KIC 4851217 based on the parameters derived in Table 5.

( $A_2=0.6$ ,  $\gamma_2=0.32$ ). Additionally, the logarithmic limb-darkening coefficients and atmospheric models were interpolated from pre-calculated grids adapted from Claret et al. (2012) and Castelli and Kurucz (2003). Finally, the preliminary solution was refined by the LSTRR method by allowing the model to relax into the local optimum by setting parameters  $T_1^{eff}$ , both albedos and gravity darkening factors, LC phase shift  $\Delta\phi$  and the third light  $l_3$  as a variable.

The vicinity of optimized solution was than explored by the MCMC sampler to primarily establish the confidence intervals of the model parameters. An uninformed priors were assumed for all parameters set as variable in the previous paragraph. Additionally, a prior knowledge from RV fit was used to sample the  $q$  from an normal distribution constructed from the derived value of  $q$  in Table 4. The 400 walkers with 400 steps each was used to produce a chain that after discarding 100 000 iterations belonging to the thermalization stage was considered as the posterior distribution of the modelled parameters. The  $1\sigma$  confidence intervals were derived from the posterior distribution similarly to an RV fit. Additionally, the values of constrained and derived parameters such as various radii and surface polar accelerations were derived using an error propagation technique built in ELISa. This technique produces a set of `BinarySystem`

## 8 INVERSE PROBLEM

---

instances for each member of the resulting MCMC chain and the derived parameters calculated during the initialization process are then collected. Finally, the confidence intervals of the derived parameters are then calculated from the obtained distributions. The resulting parameters along with their confidence interval are listed in Table 5 and the resulting synthetic curve is on display in Figure 43 along with posterior distribution of the samples displayed in Figure 50 located in Appendix F.

Tab. 5: Parameters of the KIC 4851217 derived during the fitting procedure. The status of the constrained parameters contains constraint formulas that were used to evaluate the constrained parameter during each iteration.

Parameter	Value	Status
System		
Photometric mass ratio $q$	$1.12^{+0.02}_{-0.03}$	Variable
Inclination $i[^{\circ}]$	$76.7^{+0.3}_{-0.1}$	Variable
Semi-major axis $a$	$11.61^{+0.05}_{-0.015}$	$11.3 / \sin i$
Eccentricity $e$	$0.032^{+4 \times 10^{-4}}_{-3 \times 10^{-4}}$	Variable
Argument of periastron $\omega[^{\circ}]$	$199^{+2}_{-2}$	Variable
Orbital period $P[d]$	2.47028	Fixed
Primary minimum $T_0[d]$	2454953.900507	Fixed
Component		
	primary	secondary
Mass $M$	$1.62^{+0.02}_{-0.02}$	$1.82^{+0.02}_{-0.02}$
Surface potential $\Omega$	$5.59^{+0.06}_{-0.13}$	$6.34^{+0.4}_{-0.1}$
Synchronicity $F$	$1.0675^{+8 \times 10^{-4}}_{-7 \times 10^{-4}}$	$1.0675^{+8 \times 10^{-4}}_{-7 \times 10^{-4}}$
Equivalent radius $R_{eq}[R_{\odot}]$	$2.61^{+0.1}_{-0.03}$	$2.42^{+0.03}_{-0.1}$
Polar gravity log ( $g[cgs]$ )	$3.835^{+0.008}_{-0.034}$	$3.94^{+0.05}_{-0.01}$
Bolometric luminosity $L[L_{\odot}]$	$10^{+1}_{-1}$	$7.1^{+0.5}_{-0.5}$
Atmospheric parameters		
Effective temperature $T^{eff}/[K]$	$6360^{+70}_{-180}$	$6100^{+50}_{-150}$
Gravity darkening $\beta$	$0.87^{+0.03}_{-0.04}$	$0.88^{+0.06}_{-0.14}$
Albedo $A$	$0.62^{+0.05}_{-0.04}$	$0.56^{+0.05}_{-0.03}$
Metallicity $M/H$	0.0	0.0
Radii at periastron $[R_{\odot}]$		
Polar radius	$2.57^{+0.09}_{-0.03}$	$2.40^{+0.03}_{-0.12}$
Backward radius	$2.66^{+0.10}_{-0.03}$	$2.45^{+0.03}_{-0.13}$
Side radius	$2.61^{+0.10}_{-0.03}$	$2.42^{+0.03}_{-0.13}$
Forward radius	$2.68^{+0.11}_{-0.03}$	$2.46^{+0.04}_{-0.14}$

---

A LIST OF VALID INITIAL PARAMETERS COMBINATIONS.

# Appendices

## A List of valid initial parameters combinations.

Tab. 6: List of available input parameter combinations for initialization of the modelled system. The stellar size in the single system can be defined either by the polar surface acceleration in the “standard” parameter format or using the equivalent radius in the “radius” format. In the case of binary systems, the “standard” format using component’s masses or “community” format using a mass ratio is available.

system	SingleSystem		BinarySystem	
parameter format	standard	radius	standard	community
system parameters	inclination: $i$ , rotation_period: $P$ , reference_time: $T_0$ , phase_shift: $\Delta\phi$ , additional_light: $l_2$ , gamma: $\gamma$			semi_major_axis: $a$ , mass_ratio: $q$ ,
component parameters	polar_log_g: $\log g_{pole}$ , mass: $M$ , t_eff: $T^{eff}$ , gravity_darkening: $\beta$ , polar_log_g: $\log g_{pole}$ , metallicity: $\log(M/H)$ , discretization_factor: $\alpha$ , spots: [Spot, ...], pulsations: [PulsationMode, ...]	equivalent_radius: $r_{eq}$	mass: $M$ ,	surface_potential: $\Omega$ , synchronicity: $F$ , t_eff: $T^{eff}$ , gravity_darkening: $\beta$ , albedo: $A$ , metallicity: $\log(M/H)$ , discretization_factor: $\alpha$ , spots: [Spot, ...], pulsations: [PulsationMode, ...]

## A LIST OF VALID INITIAL PARAMETERS COMBINATIONS.

---

Tab. 7: List of available input parameter combinations for initialization of a `Spot` instance.

instance	<code>Spot</code>
parameters	<code>longitude</code> : $A$ , <code>latitude</code> : $\delta$ , <code>angular_radius</code> : $r_{spot}$ , <code>temperature_factor</code> : $T_{spot}/T_{star}$ , <code>discretization_factor</code> : $\alpha$

Tab. 8: List of available input parameter combinations for initialization of a `PulsationMode` instance.

instance	<code>PulsationMode</code>
parameters	<code>l</code> : angular degree $\ell$ , <code>m</code> : azimuthal order $m$ , <code>amplitude</code> : $\tilde{v}$ , <code>frequency</code> : $f_{mode} = \omega_p/2\pi$ , <code>start_phase</code> : $\varphi_0$ , <code>mode_axis_theta</code> : $\vartheta_{mode}$ , <code>mode_axis_phi</code> : $\varphi_{mode}$ , <code>temperature_perturbation_phase_shift</code> : $\Delta\varphi_T$ , <code>horizontal_to_radial_amplitude_ratio</code> : $A_\perp/A_r$ , <code>temperature_amplitude_factor</code> : $\tilde{T}/T^{eff}$ , <code>tidally_locked</code> : True/False

## B Mathematical forms of the modified potential $\Omega$

This appendix will list all forms of modified potential and its derivatives used in ELISa code for the sake of posterity. Listed potential functions will make use of substitutions to speed up the computations to reduce the number of required operations:

### B.1 Spherical coordinates

Equation 3 for modified potential for primary component can be rewritten in the following manner:

$$\Omega_1 = \frac{1}{\varrho} + \frac{q}{\sqrt{B + \varrho^2 - C\varrho}} - D_1\varrho + E_1\varrho^2, \quad (79)$$

where following substitutions were made:

$$\begin{aligned} B &= d^2, \\ C &= 2d\lambda, \\ D_1 &= \frac{q\lambda}{B}, \\ E_1 &= \frac{1}{2}F_1^2(1+q)(1-\nu^2). \end{aligned} \quad (80)$$

Similarly, equation 8 can be rewritten:

$$\Omega_2 = \frac{q}{\varrho} + \frac{1}{\sqrt{B + \varrho^2 - C\varrho}} - D_2\varrho + E_2\varrho^2 + F, \quad (81)$$

where couple of additional substitutions were introduced:

$$\begin{aligned} D_2 &= \frac{\lambda}{B}, \\ E_2 &= \frac{1}{2}F_2^2(1+q)(1-\nu^2), \\ F &= \frac{1}{2}(1-q) \end{aligned} \quad (82)$$

Defined substitutions can be also utilized to calculate the derivatives of Equations 79 and 81 according to radial coordinate used in the Newton-Rhapson solver:

$$\frac{\partial\Omega_1}{\partial\varrho} = -\frac{1}{\varrho^2} + \frac{1}{2}\frac{q(C-2\varrho)}{(B+\varrho^2-C\varrho)^{\frac{3}{2}}} - D_1 + 2E_1\varrho \quad (83)$$

$$\frac{\partial\Omega_2}{\partial\varrho} = -\frac{q}{\varrho^2} + \frac{1}{2}\frac{(C-2\varrho)}{(B+\varrho^2-C\varrho)^{\frac{3}{2}}} - D_2 + 2E_2\varrho \quad (84)$$

Note that spherical coordinates  $\varrho, \varphi, \vartheta$  are centered on the centre of mass of given component and aligned in such manner that  $\varphi = 0, \vartheta = \pi/2$  points to the companion star and  $\varphi$  is measured in anti-clockwise direction while  $\vartheta = 0$  points to the north pole of the component as visualized in figure 44.

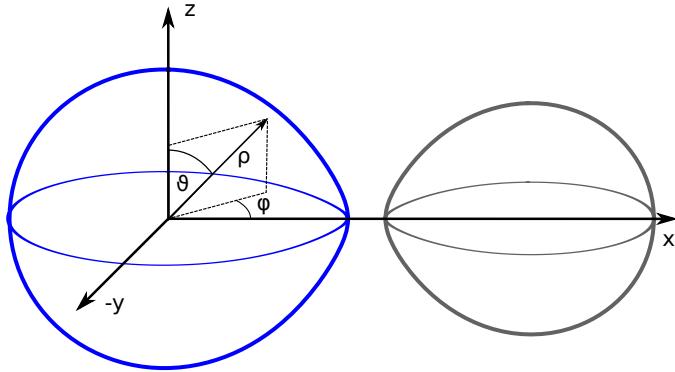


Fig. 44: Orientation of spherical coordinate system used to model detached systems and far side of over-contact systems.

## B.2 Cylindrical coordinates

Cylindrical symmetry is used to model the “neck” of the over-contact surface since the surface discretization in spherical symmetry results in a very uneven sampling of this part of the surface. Therefore, to generate surface points in cylindrical symmetry, we need to transform the modified potential  $\omega$  into cylindrical coordinates with  $z$  axis pointing toward the companion star and  $\varphi = 0$ ,  $\vartheta = \pi/2$  points to the north pole of the star as indicated in Figure 45. Making the use of suitable substitutions, the resulting modified potential for the primary component is:

$$\Omega_1 = \frac{1}{\sqrt{A + \varrho^2}} + \frac{q}{\sqrt{B + \varrho^2}} - C_1 + D_1(A + E_1\varrho^2), \quad (85)$$

where:

$$\begin{aligned} A &= z^2 \\ B &= (d - z)^2, \\ C_1 &= \frac{qz}{d^2}, \\ D_1 &= \frac{1}{2}F_1^2(1 + q), \\ E_1 &= (1 - \nu^2). \end{aligned} \quad (86)$$

Similar transformation can be made for secondary modified potential:

$$\Omega_2 = \frac{q}{\sqrt{A + \varrho^2}} + \frac{1}{\sqrt{B + \varrho^2}} - C_2 + D_2(A + E_2\varrho^2) + F, \quad (87)$$

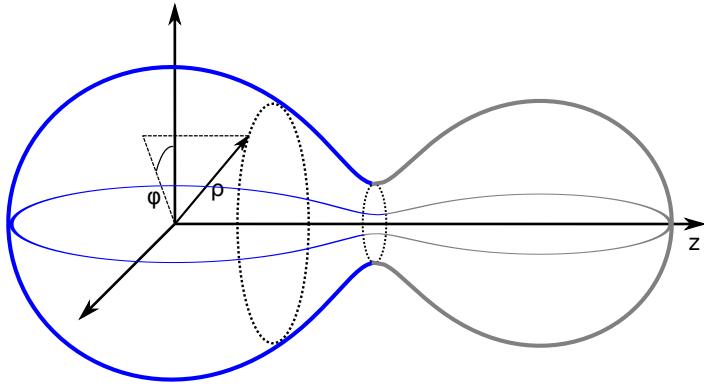


Fig. 45: Orientation of cylindrical coordinate system used to model near side of over-contact system components.

with substitution defined as:

$$\begin{aligned} C_2 &= \frac{z}{d^2}, \\ D_2 &= \frac{1}{2}F_2^2(1+q), \\ E_2 &= (1-\nu^2), \\ F &= \frac{1}{2}(1-q). \end{aligned} \tag{88}$$

The derivatives of the modified potentials in cylindrical coordinates according to  $\varrho$  are defined from the equations above to be used in the surface solver:

$$\frac{\partial\Omega_1}{\partial\varrho} = -\frac{\varrho}{(A+\varrho^2)^{\frac{3}{2}}} - \frac{\varrho q}{(B+\varrho^2)^{\frac{3}{2}}} + 2D_1E_1\varrho, \tag{89}$$

$$\frac{\partial\Omega_2}{\partial\varrho} = -\frac{\varrho q}{(A+\varrho^2)^{\frac{3}{2}}} - \frac{\varrho}{(B+\varrho^2)^{\frac{3}{2}}} + 2D_2E_2\varrho. \tag{90}$$

### B.3 Cartesian coordinates

The cartesian coordinate system is very convenient for numerical modelling due to the simplicity of performing differential operations. Therefore, ELISa calculates and stores the vast majority of the vector quantities in the Cartesian coordinates, such as surface acceleration, depending on the modified potential  $\Omega$ . Thus, we can rewrite the primary modified potential (Equation 3) and define substitutions in a Cartesian coordinate system centred on the primary component as indicated in Figure 44:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \bar{r} &= \sqrt{(d-x)^2 + y^2 + z^2} \end{aligned}$$

## B MATHEMATICAL FORMS OF THE MODIFIED POTENTIAL $\Omega$

---

$$\Omega_1 = \frac{1}{r} + q \left[ \frac{1}{\bar{r}} - \frac{x}{d^2} \right] + \frac{1}{2} F_1^2(q+1)(x^2 + y^2). \quad (91)$$

Similarly, secondary modified potential (Equation 8) can be rewritten into the same Cartesian coordinate system:

$$\Omega_2 = \frac{q}{r} + \frac{1}{\bar{r}} - \frac{x}{d^2} + \frac{1}{2} F_1^2(q+1) [(d-x)^2 + y^2] + \frac{1}{2}(1-q). \quad (92)$$

The prerequisite quantity to calculate surface acceleration on an element is the gradient of the modified potential function. That requires from us to define derivatives of the modified potentials 91 and 92 along all three coordinates:

$$\nabla \Omega_1 = \begin{bmatrix} \frac{\partial \Omega_1}{\partial x} \hat{\mathbf{x}} \\ \frac{\partial \Omega_1}{\partial y} \hat{\mathbf{y}} \\ \frac{\partial \Omega_1}{\partial z} \hat{\mathbf{z}} \end{bmatrix}; \quad \nabla \Omega_2 = \begin{bmatrix} \frac{\partial \Omega_2}{\partial x} \hat{\mathbf{x}} \\ \frac{\partial \Omega_2}{\partial y} \hat{\mathbf{y}} \\ \frac{\partial \Omega_2}{\partial z} \hat{\mathbf{z}} \end{bmatrix}, \quad (93)$$

$$\begin{aligned} \frac{\partial \Omega_1}{\partial x} &= -\frac{x}{r^3} + q \left[ \frac{(d-x)}{\bar{r}^3} - \frac{1}{d^2} \right] + F_1^2(q+1)x, \\ \frac{\partial \Omega_1}{\partial y} &= -y \left[ \frac{1}{r^3} + \frac{q}{\bar{r}^3} - F_1^2(q+1) \right], \\ \frac{\partial \Omega_1}{\partial z} &= -z \left[ \frac{1}{r^3} + \frac{q}{\bar{r}^3} \right], \end{aligned} \quad (94)$$

$$\begin{aligned} \frac{\partial \Omega_2}{\partial x} &= -\frac{x}{r^3} + \frac{q(d-x)}{\bar{r}^3} + \frac{1}{d^2} - F_2^2(q+1)(d-x)x, \\ \frac{\partial \Omega_2}{\partial y} &= -y \left[ \frac{1}{r^3} + \frac{q}{\bar{r}^3} - F_2^2(q+1) \right], \\ \frac{\partial \Omega_2}{\partial z} &= -z \left[ \frac{1}{r^3} + \frac{q}{\bar{r}^3} \right]. \end{aligned} \quad (95)$$

## C SPHERICAL HARMONICS AND THEIR DERIVATIVES

---

### C Spherical harmonics and their derivatives

Spherical harmonics provides the basis of the solution to the non-radial oscillations implemented in ELISa. The spherical harmonics are defined by their angular degree  $\ell$  and azimuthal order  $m$  on a sphere described by the spherical coordinates  $\theta, \phi$ :

$$Y_\ell^m(\theta, \phi, t) = \sqrt{(2\ell+1) \frac{(l-m)!}{(l+m)!}} P_\ell^m(\cos \theta) \exp [i(m\phi - \omega_p t - \varphi_0)], \quad (96)$$

where  $P_\ell^m$  are the associated Legendre polynomials,  $\omega_p$  is the angular frequency of the mode, and  $\varphi_0$  is the initial phase at  $t = 0$ . Additionally, angular degree  $\ell$  is a positive integer and azimuthal order  $m$  is an integer within  $\langle -l, l \rangle$  interval. Time exponential is incorporated into the  $Y_\ell^m$  for the sake of brevity except for Equation 44, where time-dependency is explicitly expressed.

Contrary to the form of the spherical harmonics widely used in quantum mechanics, the spherical harmonics used for the modelling of stellar oscillations are normalized in a way that defines their root mean square (RMS) value over the whole surface as equal to one:

$$\frac{1}{4\pi} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} Y_\ell^m Y_\ell^{m*} d\Omega = 1. \quad (97)$$

This normalization enables us to set the amplitude of the perturbed quantity very close to its mean amplitude averaged over the whole surface just by multiplying the  $Y_\ell^m$  with the desired amplitude. This is very useful while attempting to match the synthetic models of the oscillating star with the observations such as in case of radial velocity observations.

The horizontal displacement of the surface element (Equation 53) utilizes the derivatives of the  $Y_\ell^m$  along the  $\theta$  and  $\phi$  coordinates that can be expressed in the following form using the Equation 96:

$$\begin{aligned} \frac{\partial Y_\ell^m}{\partial \theta} &= m Y_\ell^m \cot \theta + \sqrt{(l-m)(l+m+1)} Y_\ell^{m+1} \exp(i\phi) \\ \frac{\partial Y_\ell^m}{\partial \phi} &= im Y_\ell^m, \end{aligned} \quad (98)$$

However, the derivatives of the  $Y_\ell^m$  along the  $\theta$  and  $\phi$  utilized in the horizontal displacement (Equation 53) are not normalized. Therefore, the renormalization constant  $C(l, m)$  was utilized to separately re-normalize the angular derivatives of the  $Y_\ell^m$  to produce a horizontal displacement with an RMS equal to one:

$$C(l, m) = 4\pi \left[ \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \sqrt{\left| \frac{\partial Y_\ell^m}{\partial \theta} \right|^2 + \frac{1}{\sin^2 \theta} \left| \frac{\partial Y_\ell^m}{\partial \phi} \right|^2} d\Omega \right]^{-1}, \quad (99)$$

where the value of  $C(l, m)$  is calculated numerically for each mode.

## D PERFORMANCE OF ECCENTRIC ORBIT APPROXIMATIONS

### D Performance of eccentric orbit approximations

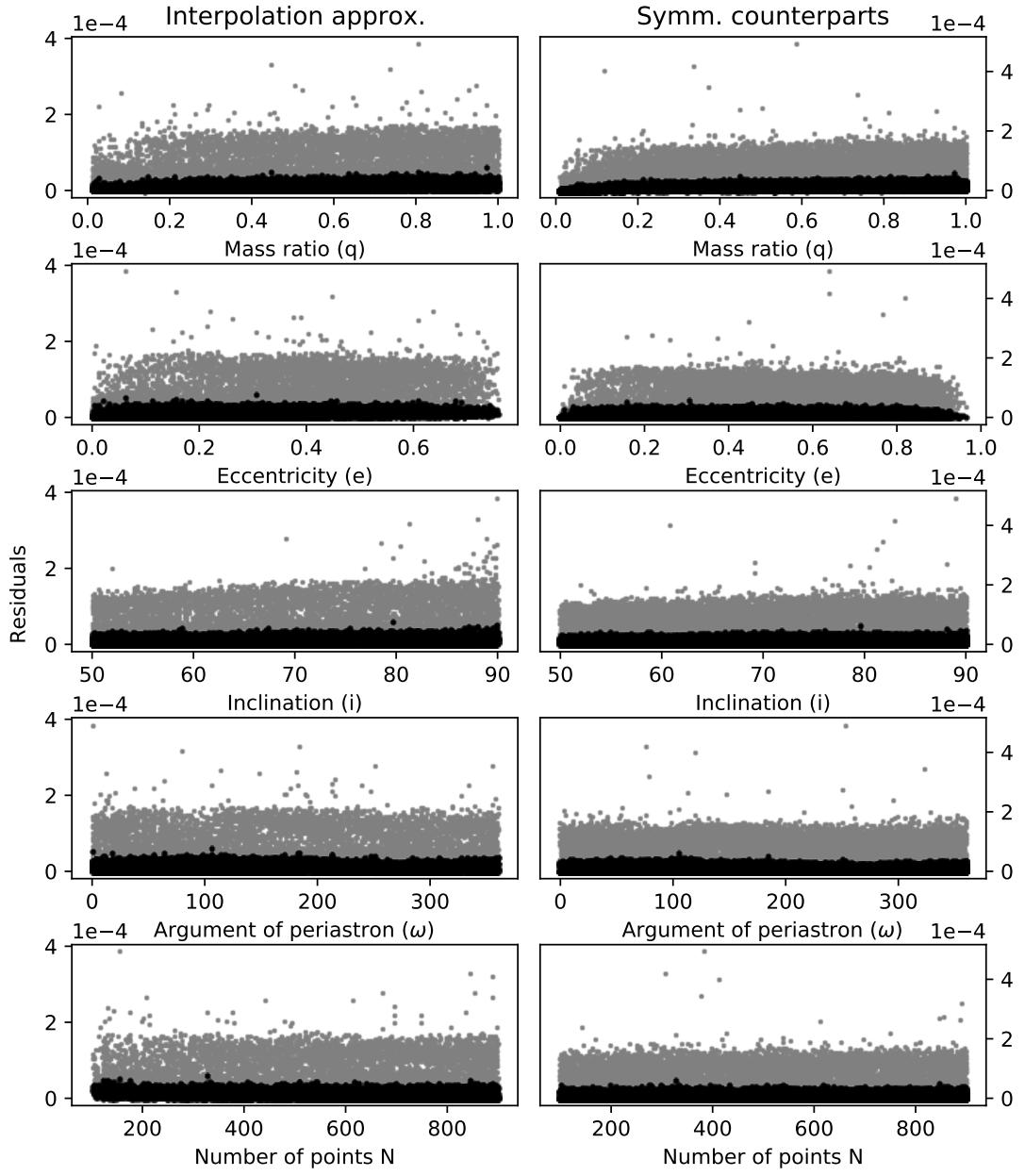


Fig. 46: Maximum deviations (grey points) and average deviations (black points) of the approximated light curves from the exactly calculated control sample. Panels illustrate distributions of the residuals along each parameter space from which we generated the 10 000 random binary system. Left column displays result of the combination of interpolation (Section 7.2.1) and the similar neighbours approximation (Section 7.2.3) whereas the right column shows the results for the combination of symmetrical counterpart (Section 7.2.2) and similar neighbour approximation.

## E FIT PARAMETERS FOR ANALYTICSTASK MODULE

### E Fit parameters for AnalyticsTask module

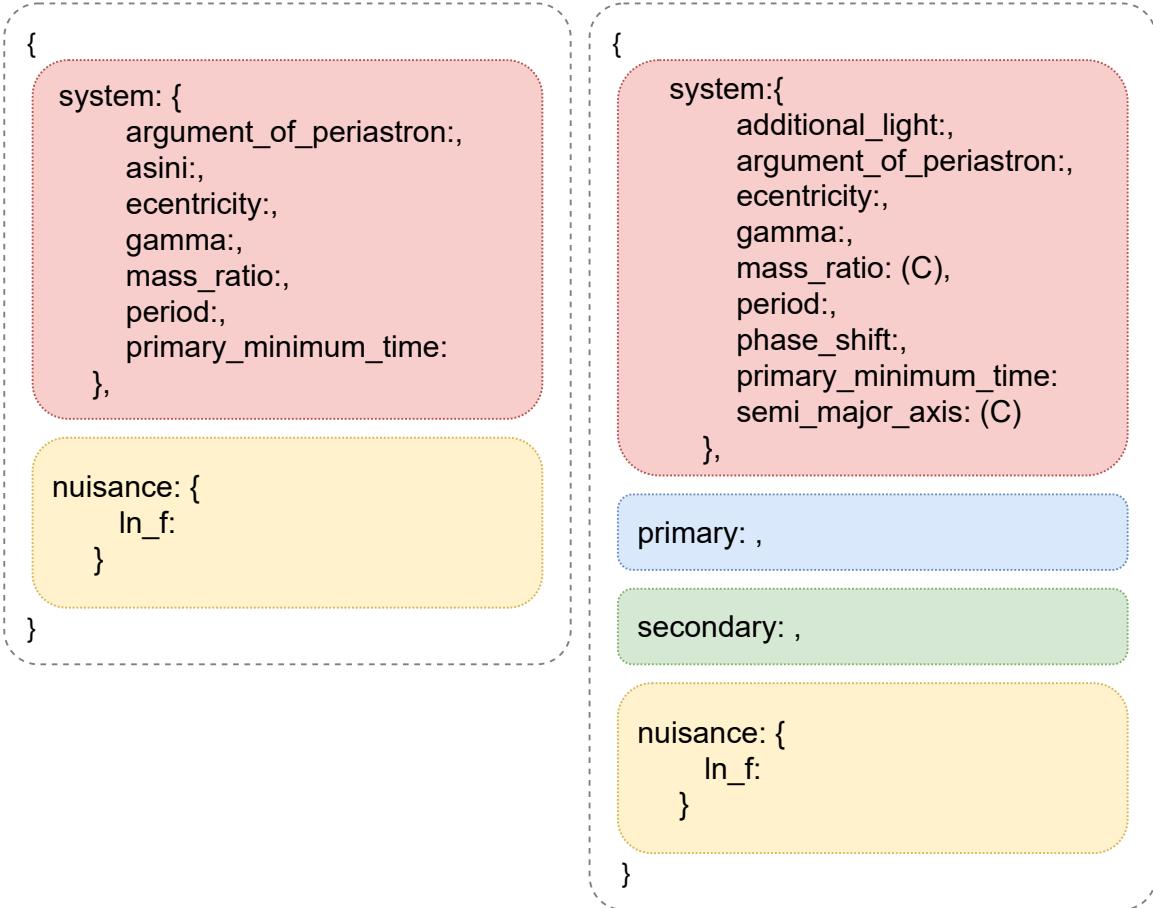


Fig. 47: Available model parameters for `RVBinaryAnalyticsTask` class (left) and `LCBinaryAnalyticsTask` class (right). Letters in the parenthesis describe whether given parameter belongs to the “standard” combination of parameters (S) or community parameter combination (C).

```
star_parameters: {  
    mass: (S),  
    t_eff:,  
    surface_potential:,  
    gravity_darkening:,  
    albedo:,  
    synchronicity:,  
    metallicity:,  
  
    spots: {  
        spot_name1:,  
        ...  
    },  
  
    pulsations: {  
        mode_name1:,  
        ...  
    }  
}
```

```
spot_name: {  
    longitude:,  
    latitude:,  
    radius:,  
    temperature_factor:  
}  
  
mode_name: {  
    l,  
    m,  
    amplitude,  
    frequency,  
    start_phase,  
    mode_axis_theta,  
    mode_axis_phi  
}
```

Fig. 48: List of available star parameters, including spots and pulsation mode parameters.

## F INVERSE PROBLEM OF KIC 4851217

### F Inverse problem of KIC 4851217

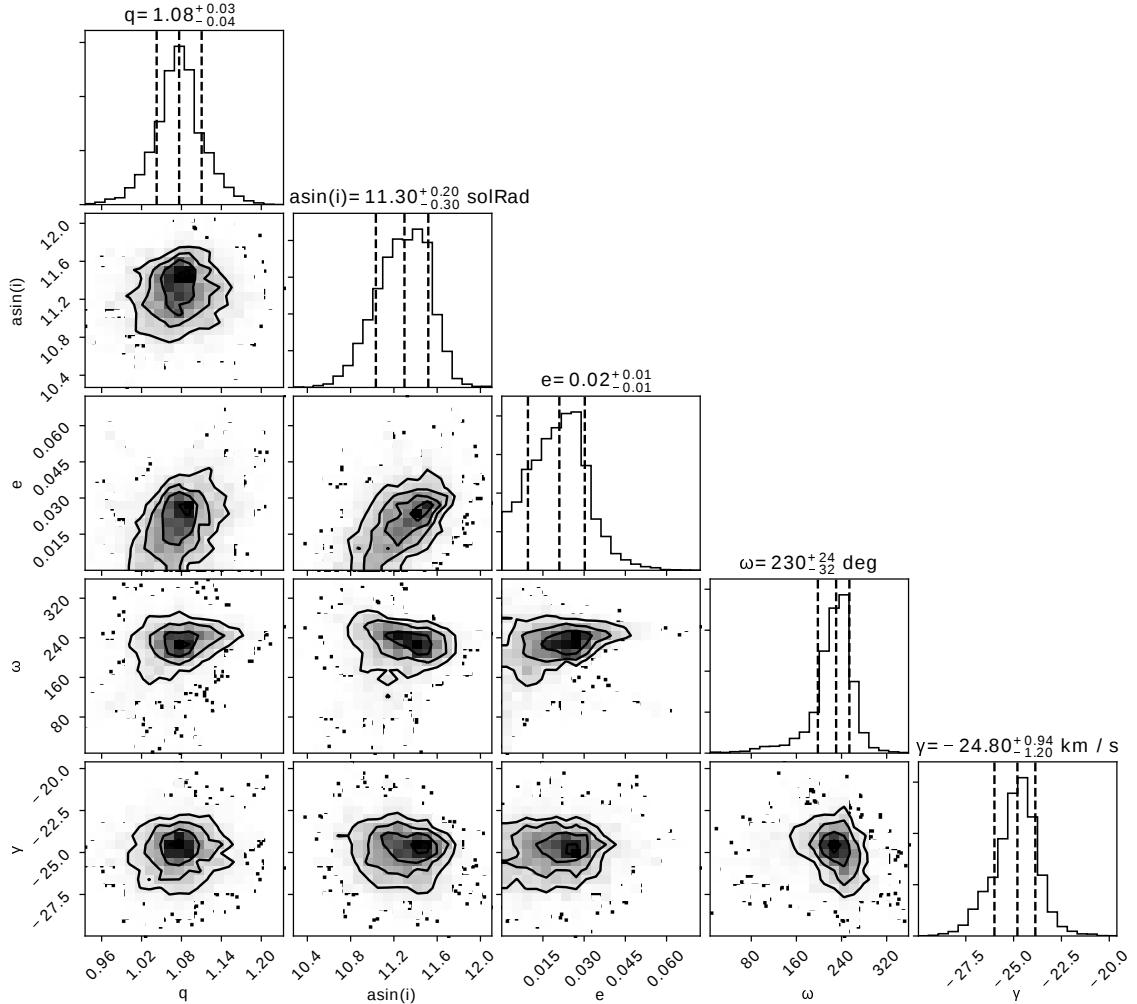


Fig. 49: Posterior distribution of MCMC sampling for the radial velocity curve fitted in Section 8.4.1.

## F INVERSE PROBLEM OF KIC 4851217

---

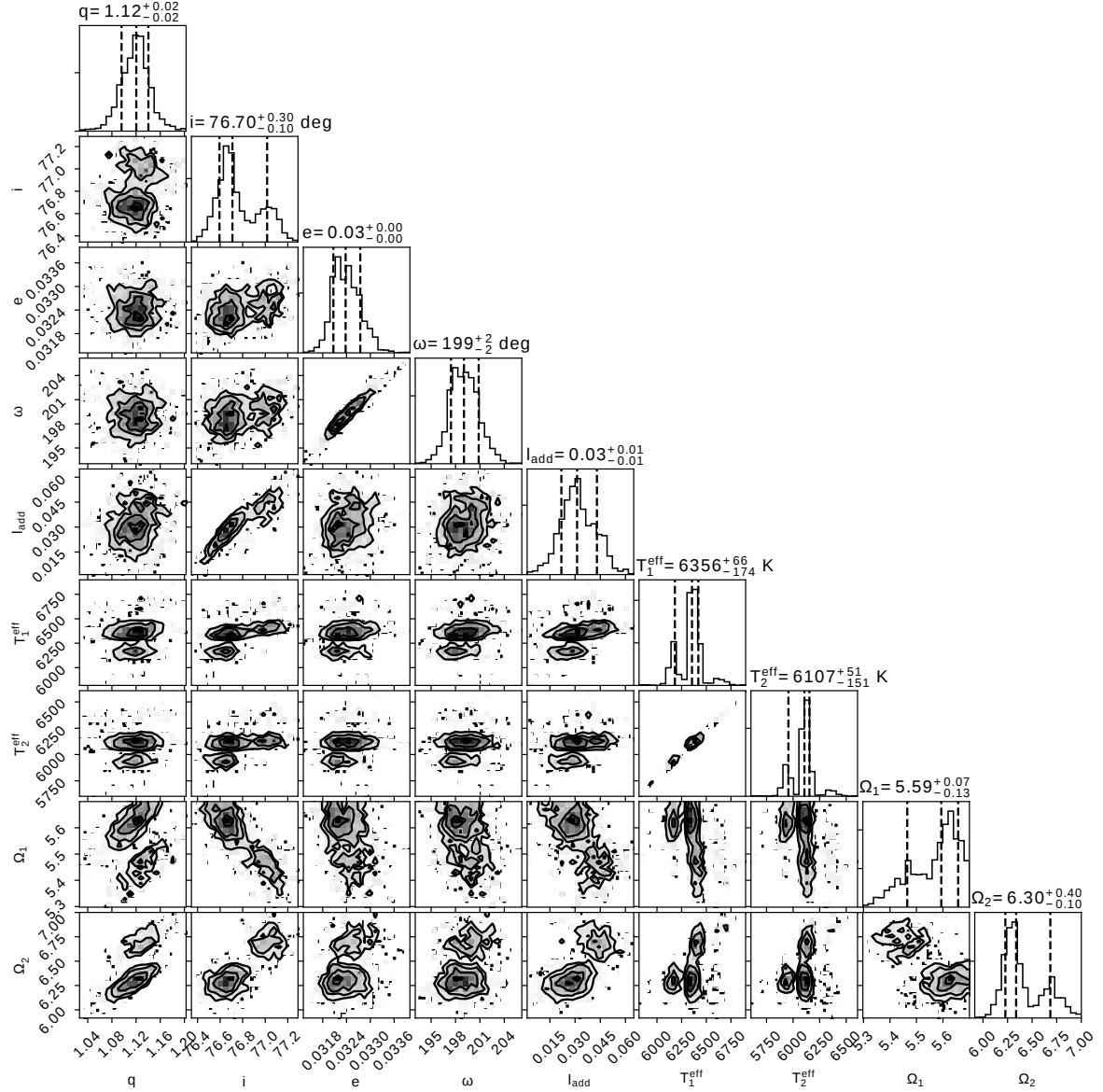


Fig. 50: Posterior distribution of MCMC sampling for the light curve fitted in Section 8.4.2.

## References

- Aerts, C., Christensen-Dalsgaard, J., and Kurtz, D. (2010). *Asteroseismology*. Astronomy and Astrophysics Library. Springer Netherlands.
- Berghen, F. W. (2004). *CONDOR: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions*. PhD thesis, Universite Libre de Bruxelles, Avenue Franklin Roosevelt 50, 1050 Bruxelles, Belgium.
- Borucki, W. J. (2016). KEPLER Mission: development and overview. *Reports on Progress in Physics*, 79(3):036901.
- Buta, R. J. and Smith, M. A. (1979). The light variations of nonradial pulsators: theory and application to the line profile variable 53 Persei. *ApJ*, 232:213–235.
- Castelli, F. and Kurucz, R. L. (2003). New Grids of ATLAS9 Model Atmospheres. In Piskunov, N., Weiss, W. W., and Gray, D. F., editors, *Modelling of Stellar Atmospheres*, volume 210, page A20.
- Chambers, K. C., Magnier, E. A., Metcalfe, N., et al. (2016). The Pan-STARRS1 Surveys. *arXiv e-prints*, page arXiv:1612.05560.
- Claret, A. (2016). Limb and gravity-darkening coefficients for the TESS satellite at several metallicities, surface gravities and microturbulent velocities. *VizieR Online Data Catalog*, pages J/A+A/600/A30.
- Claret, A., Hauschildt, P. H., and Witte, S. (2012). New limb-darkening coefficients for PHOENIX/1D model atmospheres. I. Calculations for  $1500 \text{ K} \leq T_{\text{eff}} \leq 4800 \text{ K}$  Kepler, CoRot, Spitzer, uvby, UBVRIJHK, Sloan, and 2MASS photometric systems. *A&A*, 546:A14.
- Cowling, T. G. (1941). The non-radial oscillations of polytropic stars. *MNRAS*, 101:367.
- Delaunay, B. (1934). Sur la sphère vide. *Bulletin de l'Académie des Sciences de l'URSS, Classe des sciences mathématiques et naturelles*, 6:793–800.
- Diaz-Cordoves, J. and Gimenez, A. (1992). A new nonlinear approximation to the limb-darkening of hot stars. *A&A*, 259(1):227–231.
- Eddington, A. S. (1917). The pulsation theory of Cepheid variables. *The Observatory*, 40:290–293.

- Espinosa Lara, F. and Rieutord, M. (2012). Gravity darkening in binary stars. *A&A*, 547:A32.
- Fedorco, M. and Parimucha, Š. (2018). Double eclipsing binary system KIC 3832716. *Astrophysics and Space Science*, 363(12):267.
- Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J. (2012). emcee: The mcmc hammer. *ArXiv e-prints*.
- Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., Brown, A. G. A., et al. (2016). The gaia mission. *A&A*, 595:A1.
- Guzik, J. A., Kaye, A. B., Bradley, P. A., Cox, A. N., and Neuforge, C. (2000). Driving the Gravity-Mode Pulsations in  $\gamma$  Doradus Variables. *ApJ*, 542(1):L57–L60.
- Handler, G., Kurtz, D. W., Rappaport, S. A., Saio, H., Fuller, J., et al. (2020). Tidally trapped pulsations in a close binary star system discovered by TESS. *Nature Astronomy*, 4:684–689.
- Hartmann, E. (1998). A marching method for the triangulation of surfaces. *Visual Comput.*, 14(3):95–108.
- Hilditch, R. (2001). *An Introduction to Close Binary Stars*. Cambridge University Press.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- Hut, P. (1981). Tidal evolution in close binary systems. *A&A*, 99:126–140.
- ISO (2019). Quantities and units — part 7: Light and radiation. ISO 80000-7:2019, International Organization for Standardization, Geneva, Switzerland.
- Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., et al. (2019). LSST: From Science Drivers to Reference Design and Anticipated Data Products. *ApJ*, 873:111.
- Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python.
- Kallrath, J. and Milone, E. (2009). *Eclipsing Binary Stars: Modeling and Analysis*. Astronomy and Astrophysics Library. Springer New York.
- Kantorovich, L. V. (1948). Functional analysis and applied mathematics. *Usp. Mat. Nauk*, 3(6(28)):89–185.

- Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson Surface Reconstruction. In Sheffer, A. and Polthier, K., editors, *Symposium on Geometry Processing*. The Eurographics Association.
- Kim, A., Winget, D. E., Montgomery, M. H., and Kepler, S. O. (2006). Driving in ZZ Ceti stars - Problem solved? . *Memorie della Societa Astronomica Italiana*, 77:376.
- Kirk, B., Conroy, K., Prsa, A., Abdul-Masih, M., Kochoska, A., et al. (2016). VizieR Online Data Catalog: Kepler Mission. VII. Eclipsing binaries in DR3 (Kirk+, 2016). *VizieR Online Data Catalog*, page J/AJ/151/68.
- Klinglesmith, D. A. and Sobieski, S. (1970). Nonlinear Limb Darkening for Early-Type Stars. *AJ*, 75:175–182.
- Kopal, Z. (1959). *Close Binary Systems*. International astrophysics series. Wiley.
- Lee, J. W., Hong, K., and Kim, H.-Y. (2021). Tidally Excited Modes and  $\delta$  Scuti Pulsations in the Eclipsing Triple Star IM Persei. *AJ*, 161(3):129.
- Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.
- Lucy, L. B. (1967). Gravity-Darkening for Stars with Convective Envelopes. *Zeitschrift fuer Astrophysik*, 65:89.
- Maceroni, C., Lehmann, H., da Silva, R., Montalbán, J., Lee, C. U., et al. (2014). KIC 3858884: a hybrid  $\delta$  Scuti pulsator in a highly eccentric eclipsing binary. *A&A*, 563:A59.
- Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.
- Matson, R. A., Gies, D. R., Guo, Z., and Williams, S. J. (2017). Radial Velocities of 41 Kepler Eclipsing Binaries. *AJ*, 154(6):216.
- Maxted, P. F. L. (2016). ellc: A fast, flexible light curve model for detached eclipsing binary stars and transiting exoplanets. *A&A*, 591:A111.
- McLaughlin, D. B. (1924). Some results of a spectrographic study of the Algol system. *ApJ*, 60:22–31.

- Milne, E. A. (1921). Radiative Equilibrium in the Outer Layers of a Star: the Temperature Distribution and the Law of Darkening.
- Neslušan, L. (2018). *Elementárny úvod do nebeskej mechaniky*. Veda, vydavateľstvo SAV.
- Orosz, J. A. and Hauschildt, P. H. (2000). The use of the NextGen model atmospheres for cool giants in a light curve synthesis code. *A&A*, 364:265–281.
- Paxton, B., Schwab, J., Bauer, E. B., Bildsten, L., Blinnikov, S., et al. (2018). Modules for Experiments in Stellar Astrophysics (MESA): Convective Boundaries, Element Diffusion, and Massive Star Explosions. *ApJS*, 234(2):34.
- Pojmanski, G., Pilecki, B., and Szczygiel, D. (2005). The All Sky Automated Survey. Catalog of Variable Stars. V. Declinations 0 arcdeg - +28 arcdeg of the Northern Hemisphere. *Acta Astronomica*, 55:275–301.
- Pollacco, D. L., Skillen, I., Collier Cameron, A., et al. (2006). The WASP Project and the SuperWASP Cameras. *Publications of the Astronomical Society of the Pacific*, 118(848):1407–1418.
- Prsa, A., Conroy, K., and Horvat, M. a. a. (2016). Physics of eclipsing binaries. ii. towards the increased model fidelity. *The Astrophysical Journal Supplement Series*, 227(2).
- Prsa, A. and Zwitter, T. (2005). Photometric study of the contact binary star v861 herculis[\*]. *Astrophysical Journal*, 628:426–438.
- Rappaport, S. A., Kurtz, D. W., Handler, G., Jones, D., Nelson, L. A., et al. (2021). A tidally tilted sectoral dipole pulsation mode in the eclipsing binary TIC 63328020. *MNRAS*, 503(1):254–269.
- Reeve, D. C. and Howarth, I. D. (2016). Limb-darkening coefficients from line-blanketed non-LTE hot-star model atmospheres. *MNRAS*, 456(2):1294–1298.
- Ricker, G. R., Winn, J. N., Vanderspek, R., et al. (2015). Transiting Exoplanet Survey Satellite (TESS). *Journal of Astronomical Telescopes, Instruments, and Systems*, 1:014003.
- Rodgers, A. W. (1957). Radius variation and population type of cepheid variables. *MNRAS*, 117:85.

- Rossiter, R. A. (1924). On the detection of an effect of rotation during eclipse in the velocity of the brighter component of beta Lyrae, and on the constancy of velocity of this system. *ApJ*, 60:15–21.
- Rouan, D. (2011). *Effective Temperature*, pages 479–480. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Rucinski, S. M. (1969). The proximity effects in close binary systems. ii. the bolometric reflection effect for stars with deep convective envelopes. *Acta Astronomica*, 19:245.
- Sekaran, S., Tkachenko, A., Johnston, C., and Aerts, C. (2021). A comparison of the dynamical and model-derived parameters of the pulsating eclipsing binary KIC 9850387. *A&A*, 648:A91.
- Shibahashi, H. and Kurtz, D. W. (2012). FM stars: a Fourier view of pulsating binary stars, a new technique for measuring radial velocities photometrically. *MNRAS*, 422(1):738–752.
- Skarka, M., Kabáth, P., Paunzen, E., Fedurco, M., Budaj, J., et al. (2019). HD 99458: First time ever Ap-type star as a  $\delta$  Scuti pulsator in a short period eclipsing binary? *MNRAS*, 487(3):4230–4237.
- Stassun, K. G., Torres, G., Johnston, C., Stevens, D. J., Feliz, D. L., et al. (2021). Discovery and Characterization of a Rare Magnetic Hybrid  $\beta$  Cephei Slowly Pulsating B-type Star in an Eclipsing Binary in the Young Open Cluster NGC 6193. *ApJ*, 910(2):133.
- Townsend, R. H. D. (2003). Surface modelling of non-radial pulsators: alternative formalisms within the linear approximation. *MNRAS*, 343(3):863–870.
- Townsend, R. H. D. and Teitler, S. A. (2013). GYRE: an open-source stellar oscillation code based on a new Magnus Multiple Shooting scheme. *MNRAS*, 435(4):3406–3418.
- Udalski, A., Szymański, M. K., and Szymański, G. (2015). OGLE-IV: Fourth Phase of the Optical Gravitational Lensing Experiment. *Acta Astronomica*, 65(1):1–38.
- Unno, W., Osaki, Y., Ando, H., Saio, H., and Shibahashi, H. (1989). *Nonradial oscillations of stars*.
- van Hamme, W. (1993). New limb-darkening coefficients for modeling binary star light curves. *AJ*, 106:2096–2117.

- von Zeipel, H. (1924). The radiative equilibrium of a rotating system of gaseous masses. *MNRAS*, 84:665–683.
- Wilson, R. E. (1979). Eccentric orbit generalization and simultaneous solution of binary star light and velocity curves. *ApJ*, 234:1054–1066.
- Wilson, R. E. (1990). Accuracy and Efficiency in the Binary Star Reflection Effect. *ApJ*, 356:613.
- Wilson, R. E. and Devinney, E. J. (1971). Realization of Accurate Close-Binary Light Curves: Application to MR Cygni. *ApJ*, 166:605.
- Zahn, J. P. (1975). The dynamical tide in close binaries. *A&A*, 41:329–344.
- Čokina, A., Fedurco, M., and Š., P. (2021). ELISa: a new tool for fast modelling of eclipsing binaries. *accepted in A&A*.