UNIVERSITY OF LJUBLJANA

FACULTY OF MATHEMATICS AND PHYSICS,
DEPT. OF ASTROPHYSICS

Prša Andrej

PHOEBE tutorial

August 01, 2002

Contents

1 Introduction

This document is a part of the ongoing effort to bring PHOEBE closer to its users. Here you will find step-by-step instructions for using PHOEBE for the first time. When you get acquainted with the main ideas and design, you may find the PHOEBE Reference Manual (which doesn't exist at the time) more instructive.

1.1 PHOEBE Licence

PHOEBE is released under a GNU General Public Licence. A copy of this licence is distributed along PHOEBE and you may find it in a file COPYING in a base PHOEBE directory. To quote the most important part:

> The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.
>
> When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
>
> To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.
>
> For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.
>
> We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

> Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.
>
> Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification may be found in a file `COPYING`.

1.2 Some general remarks about PHOEBE

Although this document is a tutorial and as such doesn't even tend to cover all aspects of PHOEBE, a few introductory lines are in order. Let us start by explaining the name: PHOEBE stands for PHysics Of Eclipsing BinariEs.

Probably the most important thing for any new PHOEBE user is to understand the limitations and capabilities of this software. Although PHOEBE aims to be as useful as possible, it still tends to follow the basic principle of simplicity. In other words: although PHOEBE may assist you in solving your specific problem, it won't write an article for you or even make you a good cup of coffee; that is still up to you.

The basic goal of PHOEBE is to aid you with intuitive interface through the pain of solving the inverse problem for eclipsing binaries. Since the dimension of the parameter phase-space is tremendous and the problem itself highly non-linear, this task is hard to live up to and PHOEBE is thus subject to constant changes and improvements. Hopefully each new version converges to a global optimum.

To describe PHOEBE in one sentence: for a set of user-supplied experimental light curves and radial velocity curves PHOEBE estimates physical parameters of an eclipsing binary that theoretically best matches the data.

Physical parameters PHOEBE is working with are divided in the two categories: *curve independent* parameters and *curve dependent* parameters. When certain parameters hold a single value which is the **same** for all light and radial velocity curves, we denote it as *curve independent*. Such are, in example, surface temperatures, mass ratio, semi-major axis etc. On the other hand, some parameters have a value, which depends on a specific curve, thus holding **different** values for different curves. These parameters are *curve dependent*; such are limb darkening coeficients, luminosities, third light etc.

1.3 PHOEBE structure

The main PHOEBE window is divided in 10 **tabs** – screen pages with selected content that fits into a specific category. Data tab contains the information on experimental data, which is supplied by the user. Basic and Advanced tabs contain curve-independent parameters, sorted in physically most meaningful order. Orbit tab contains parameters such as orbital ellipticity and will shortly contain Roche geometry graphs and star plotting functions. Next tab, Luminosities, contains curve-dependent parameters, which means that they have different values for different filters. Limb darkening and Spots tabs contain perturbative orders of corrections to curves. Switches tab is obsolete and will eventually be deleted; for now it contains switches that aren't yet placed anywhere else. Finally, the Fitting tab contains functions and parameters that define and support minimization algorithms. Utilities tab is currently disabled, but will hold functions that are not directly connected to PHOEBE.

Besides the main window there are three other types of PHOEBE windows. **Calculation** windows help the user to quickly estimate the values of some specific parameters, such as limb darkening coeficients and gravity potentials. **Plotting** windows are meant for quick and simple eye inspections of both experimental and synthetic light and radial velocity curves. Finally, a **fitting** window guides the user through the minimization process and iterative solution convergence.

All windows may peacefully coexist because of the modular design of PHOEBE. It is up to you to decide the layout that suits you best, PHOEBE poses no limitations in this respect. You may not, however, have more copies of the same window. That would seemingly make PHOEBE bloated and mundane, since the contents of one window might not be the same as the contents of another instance of the same window and this might lead to ambiguity problems.

On the bottom of the main PHOEBE window you may see a status bar (labelled Last Event), where PHOEBE notifies you of all events when they are initiated. This is for your convenience only. There is a planned support for history browsing and undo-ing unwanted changes, which will be closely related to events in the status bar.

Finally, the lowermost part of the main PHOEBE window is occupied by a button box. The buttons in this box let you quickly perform some specific operations. Open and Save buttons let you manipulate PHOEBE keyword files – files containing all the information about a certain star and are usually recognisable by the suffix .phoebe. The Defaults button restores all parameter values and entries to a state they were in the beginning. LC Plot and RV Plot buttons open the plotting windows, Fitting takes you to the fitting window, Help button will eventually bring up the online help screen, but this is not enabled yet. Finally, the Exit button closes all PHOEBE windows.

2 First steps with PHOEBE

In this section we will take PHOEBE for a spin to get a feeling about what it can do for us. We shall base our introductory procedure on physical insights rather than just reach formal solutions to our problem.

2.1 Configuring PHOEBE

We shall presume that your PHOEBE installation was successful; if not, please turn to installation instructions found in the top-level PHOEBE directory, in a README file. Further on, we shall presume (since this is a testing version of PHOEBE) that the installation resides in a temporary directory (as suggested and explained in a README file).

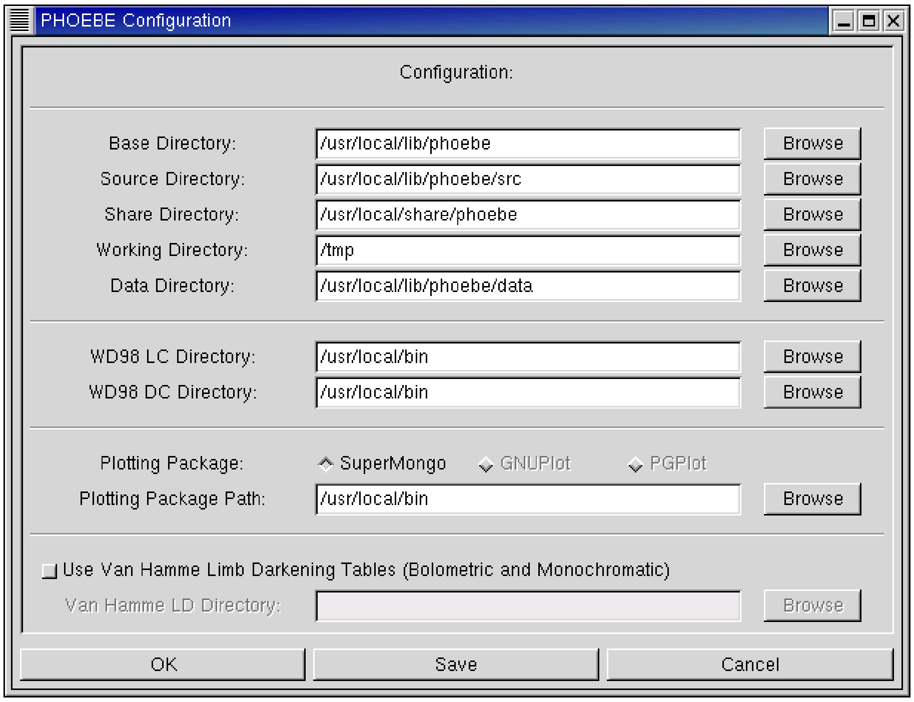
As most computer programs, PHOEBE must be set up before the first use. When you first start PHOEBE, a following warning will appear:

PHOEBE *configuration warning.*

This warning lets you know that PHOEBE was unable to find its configuration directory; this is a hidden directory in your personal $HOME directory (usually /home/name/.phoebe). By clicking on OK button this directory will be created and PHOEBE will take you to the configuration window (figure on the next page).

The default settings in the configuration window are usually OK for a complete PHOEBE installation, which by our initial presumption isn't (yet) the case. This is why we want to change these default settings. You may enter new pathnames either by typing in the new values directly to the entry box, or by clicking on the Browse button and then selecting the right directory through a file selector. The latter is preferable, since it is easy to make typing errors otherwise. Please note that fool-proofing isn't yet complete and that PHOEBE might crash if these pathnames aren't valid.

Let's presume you have installed PHOEBE in /home/name/phoebe. The Base directory is the directory of your current distribution, meaning, in our case, /home/name/phoebe. By clicking on the Browse button next to the entry, a file selector will be opened. Double-click on .. to get to the base directory and click on OK. This will put the path to that directory to the appropriate entry box. Next, set the Source directory to /home/name/phoebe/src by following the same method as above.



PHOEBE *configuration window.*

Please make sure not to assign filenames, only directories to these entries! Set the `Share directory` to `/home/name/phoebe/share`, leave the `Working directory` as is, `/tmp`, and change both `WD98` directories to `/home/name/phoebe/bin`.

Right now the only plotting package `PHOEBE` supports is P. Monger's and R. Lupton's `Super Mongo`. We understand that his software is propriatery and as such not freely available, but (1) to our understanding it is widely spread in astronomical community and (2) there is planned support for `GNUPlot` and `PGPlot` in the near future.

If you have access to limb darkening coeficients calculated and tabulated by W. Van Hamme, you may use them to obtain good initial approximation for those coeficients. In that case turn on the appropriate switch and set the path to the tables in the designated entry field.

When finished, click on the `Save` button. The settings will be saved, so you need to set it up only this one time. If you want to review or modify your settings later, you may find them in `Settings->Configuration` menu.

When you are finished with setting up, it's time to get some real work done!

2.2 Getting started

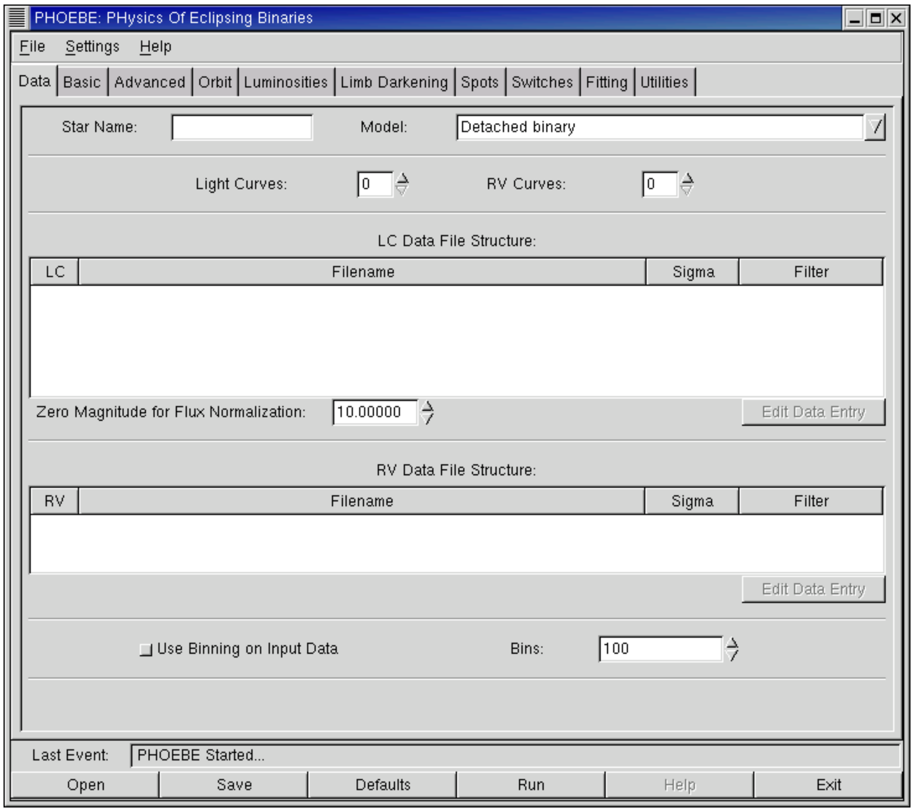
During this tutorial, we shall try and find the preliminary solution of the `UV Leonis` binary system. `UV Leonis` is a detached binary with coordinates $\alpha_{2000} = 10^h 38^m 20.7699^s$ and $\delta_{2000} = +14^o 16' 03.670''$. Data files for this system are acompanying every `PHOEBE` release and reside in `data/UVLeo` directory. There are two photometric light curves[1] in B and V filters (both from Johnson series): `UVLeo.B` and `UVLeo.V` and two radial velocity curves[2]: `UVLeo.rv1` and `UVLeo.rv2` – one for each star.

2.3 Setting experimental data

First and foremost, we have to tell `PHOEBE` where to find our experimental data.

On top of the `Data` tab (figure below) you may see `Star Name` entry field. This field is for your comfort only and carries no modeling significance. Click on it and enter `UV Leonis`.

Assign a model to your star by choosing the most appropriate from the given choices. The choice of a model is very important, since it poses specific restrictions on your solution. Take a minute to browse through the available models, but since our star in fact is a detached system, leave it



`PHOEBE` *data tab.*

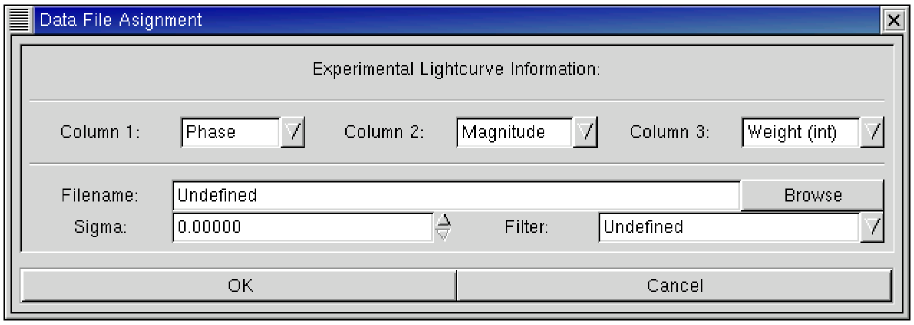
[1]Photometric measurements have been acquired and are the property of the Črni Vrh observatory, Slovenia.

[2]Spectroscopic measurements have been acquired and are the property of the Asiago observatory, Italy.

on default. If in doubt what to choose, you may start with a general binary system model and estimate the type of your binary as you get closer to the solution; then set the appropriate model.

Tell `PHOEBE` that we have two light curves and two radial velocity curves by entering number 2 in both fields. You may change the value directly, by clicking on the entry and typing 2 instead of 0, or use spin arrows. As you change the number of experimental data curves, empty records are added either to light curves list or to radial velocity curves list. Note that the used abreviation for light curves is `LC` and for radial velocity curves is `RV`.

Assign a first light curve to `PHOEBE`. Edit the first LC entry by either double-clicking on it, or by clicking on the `Edit Data Entry` button on the lower right part of the list. A following window will appear:



`PHOEBE` *light curve assignment.*

Input files should contain two or three columns. The first column should contain the **independent** variable, which may be phase or heliocentric Julian date for both light curves and radial velocity curves. The second column should contain the **dependent** variable, which may be a flux or a magnitude for light curves, or RV in km/s or RV in 100km/s for radial velocity curves. The third column is optional; if present, it should contain weights for individual data points either as integers or as real numbers. The normalization is arbitrary, higher the number, stronger the weight.

Now inspect your input files from the terminal; simply write something like:

```
[name@machine src]$ head ../data/UVLeo/UVLeo.B
2451602.395411   10.130
2451602.397101   10.129
2451602.398802   10.157
2451602.400527   10.177
2451602.402251   10.212
2451602.403953   10.197
2451602.405689   10.202
2451602.407414   10.200
2451602.409103   10.215
2451602.410828   10.178
```

Although there is no header line in this file (any line that starts with a # delimeter is ignored when reading in the values and it is thus a common practice to use it as a header line), it is obvious that the independent variable is heliocentric Julian date (PHOEBE refers to it simply as Time) and that the dependent variable is magnitude. The third column is missing. Therefore set all three column definitions accordingly. Assign a filename by clicking on the Browse button and selecting UVLeo.B from the data/UVLeo directory. Since we are not sure about dispersion of our data, set Sigma to 0.1. We shall fix this during the course of seeking preliminary solutions. Finally, choose apropriate filter: 440nm (B).

When you are happy with the settings, click the OK button. This will close the assignment window and update the light curves list. Repeat the above assignment for the other light curve file; this time set the filter to 550nm (V).

Be aware that all internal calculations are performed in fluxes, not magnitudes. Therefore PHOEBE automatically transforms magnitudes to fluxes, if magnitudes are dependent variables, which in our case they are. The conversion is done by using the user-supplied zero magnitude. If the zero magnitude corresponds to the magnitude at quarter phase, then the flux will be normalized at quarter phase. This is what the user should tend to, but not rigorously. By this we mean the following: don't spend your time seeking the right zero magnitude, leave this to PHOEBE. Just be sure you are close enough: $\pm$ 2 magnitudes or so. In our case leaving zero magnitude at 10.0 works just fine.

Next, assign radial velocity curves to PHOEBE. In much the same manner as above, inspect the contents of a radial velocity file:

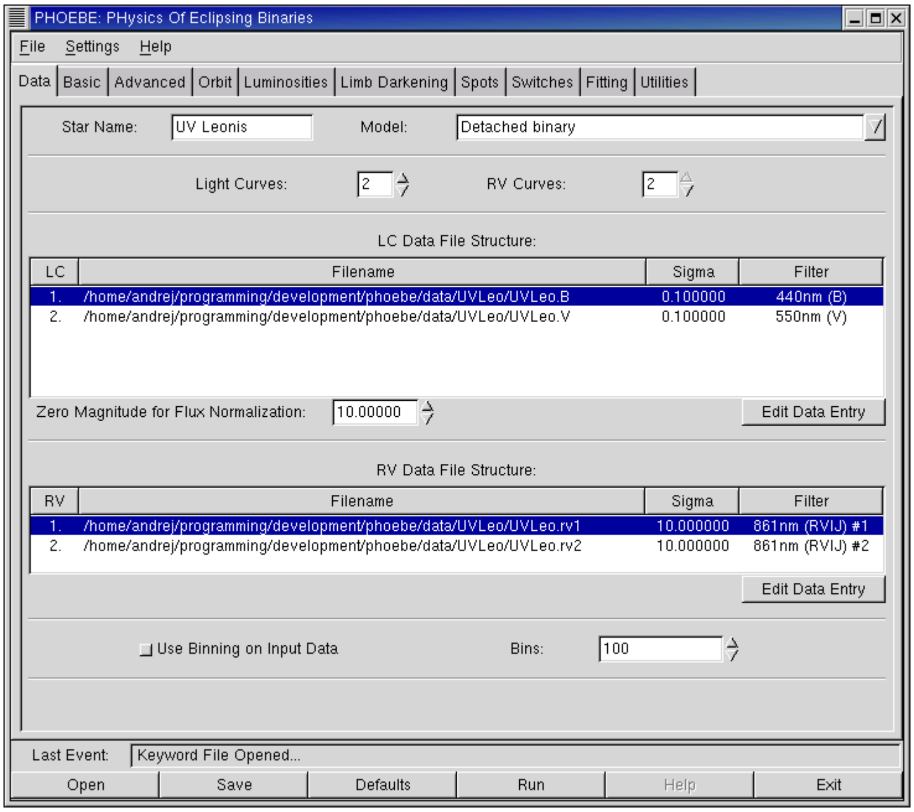
```
[name@machine src]$ head ../data/UVLeo/UVLeo.rv1
2451209.5077  -151.4
2451210.4258   166.6
2451216.4831   116.0
2451217.4545    18.9
2451221.5130  -156.3
2451225.4237   179.4
2451269.4314  -110.2
2451269.4640  -149.6
2451269.4965  -164.4
2451275.4386  -121.7
```

This suggests that the independent variable is again heliocentric Julian date and that the dependent variable is radial velocity in units km/s. Edit both RV entries as explained above. Set both Sigmas to 10 km/s and choose a filter 861nm (RVIJ). Radial velocities should theoretically be independent of the filter choice and in most cases they are; this specific filter tells PHOEBE to use a straight flux file and not think of radial velocities as wavelength dependent values.

After assigning a second radial velocity curve, note that filter names in RV columned list have changed: two extra numbering delimeters #1 and #2 have been added to make a distinction in curve dependent assignments.

If you want to use bins instead of complete data arrays, you may select data binning next. This is advisable only when you deal with humongous number of data points (10.000 or more), and that only when seeking pre-pre-preliminary solution. Be aware that you are degrading light curves heavily this way.

If you followed the described procedure correctly, you should end up with a `Data` tab looking like this:

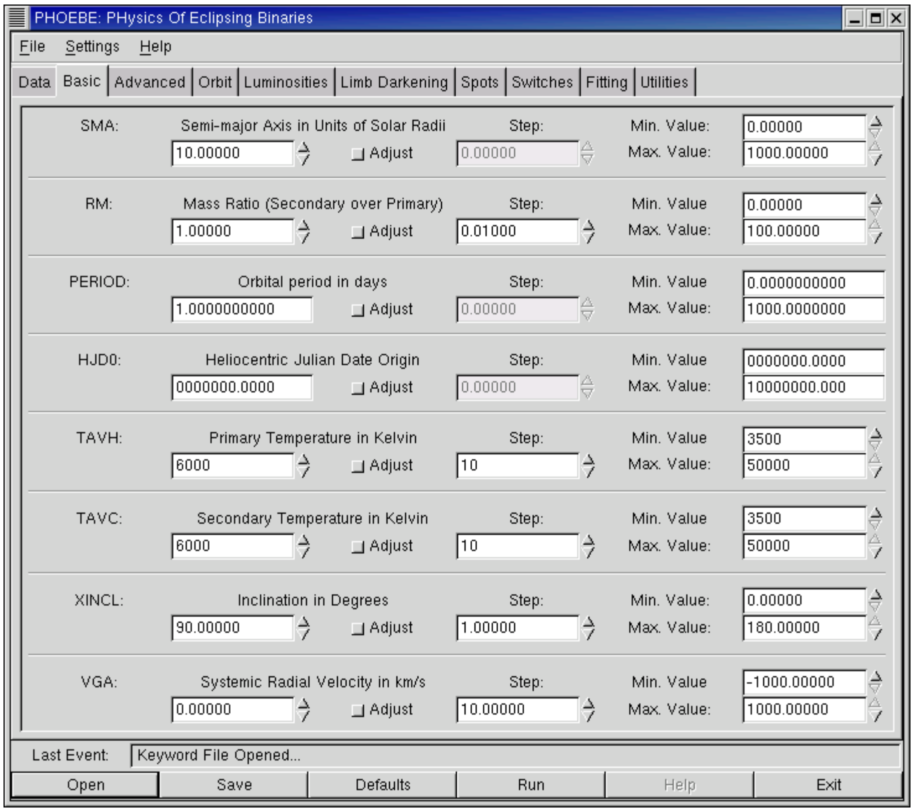


How your `Data` *tab should eventually look.*

2.4 Basic parameters

There are two most important physical binary star parameters that must be set: period and the origin of time, `HJD0`. Before we modify anything, though, take a few moments to get familiar with the overall structure.

The whole tab is built of 8 records, each setting up one physical parameter. The notation written in capitals of each record signifies a `WD98` parameter name; this is intended for those users who already have experience working with Wilson–Devinney code. To the right you may see a concise description and their assigned value underneath. The adjustment button, `Adjust`, lets you tell PHOEBE that you wish that parameter to be adjusted when minimization algorithm is initiated. The `Step` values are the fitting steps of the *differential corrections* minimization routine; when tinkering with them, be careful about their estimations – they shouldn't be too small, for you sacrifice computing time, or too large, when you sacrifice accuracy. Some steps you cannot set (semi-major axis and period, in example) and their entries are shaded. PHOEBE uses *Levenberg-Marquardt* minimization routine for these parameters for a better convergence and the optimal step size is computed automatically. Finally, on the far right you may see minimum and maximum allowed values for a certain parameter; these limitations are reserved for *non-linear simplex* minimization algorithm, which is not yet implemented.



PHOEBE *basic tab.*

Now try and change the value for period and the origin of time! Change the period to 0.600086 (in units of days) and `HJD0` to 2448500.562. These two numbers you should estimate yourself independently of using PHOEBE. There is a planned support for a light curves minimum finding routine. Since we have no physical indications about the values of the other parameters, leave them be as they are.

2.5 What else may we presume?

Since we know virtually nothing in advance of our star, it makes no solid sense to make *any* further presumptions. We cannot safely say that the orbit is either circular or elliptical, until we examine the shape of radial velocity curves. We cannot be sure about the temperatures until we examine light curves and measure the color index. We have no idea about the parameters that merely influence shapes of synthetic curves, such as the inclination, mass ratio or the semi-major axis; we definitely have no clues about the secular changes (time derivatives of the period and the periastron); these are of such perturbative order that it is rarely obvious how to change these values. As the fit brings you closer to the solution, more of the star's nature will be revealed; take it one step at the time.

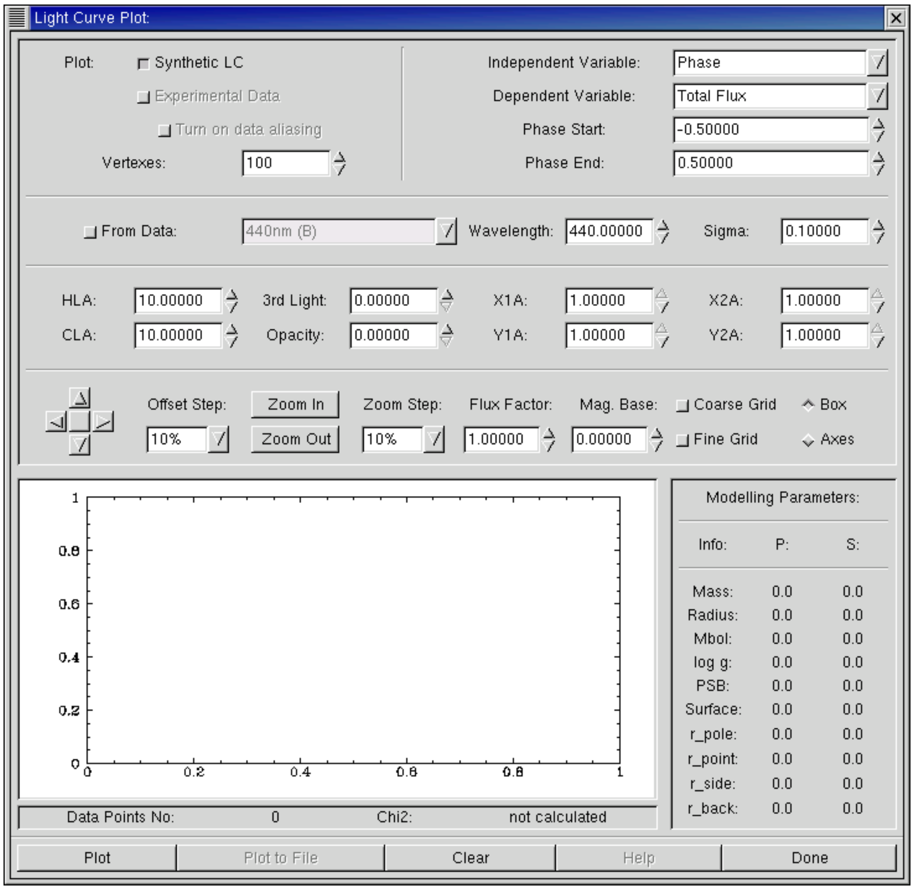
2.6 Inspecting data files closely

The next logical step would usually be eye inspection of light curves and radial velocity curves. `PHOEBE` enables you to plot experimental curves in a separate window, which you access by clicking on the `Plot LC` button on the button box for light curves and `Plot RV` button for radial velocity curves.

Let us start with light curves. Click on the `Plot LC` button on the button-box.

The top left part of the LC plotting window is used to select what it is we want plotted. Synthetic light curves are computed from theory, based on the current palette of physical parameters. Experimental light curves are taken from our input files.

Please note a very important fact: curve independent parameters for synthetic light curve calculation are taken from the *main* `PHOEBE` window and curve dependent parameters are taken from the *plotting* window: the wavelength, dispersion (`Sigma`), primary and secondary star luminosities (`HLA` and `CLA`), third light, opacity and limb darkening coeficients (`X1A`, `Y1A`, `X2A` and `Y2A`) all have their entries in the plotting window. Take a moment to locate them in the plotting window. If you wish to use curve dependent parameters that are assigned to a specific experimental curve, you must tell `PHOEBE` what curve it is you want the data from. You achieve this by turning the `From Data` switch on and choosing a filter of your choice from the list. Try doing this and observe what happens! By assigning a specific filter to `PHOEBE`, (1) the `Experimental Data` switch in the top left part of the plotting window becomes available, since now `PHOEBE` knows what data you want to plot, and (2) the values for curve dependent parameters are copied from the main `PHOEBE` window to the corresponding place in the LC plotting window. These values cannot be edited when the `From Data` switch is turned on; this is to avoid ambiguity problems — does the change in the plotting window change the actual values of parameters and alike.



The light curves plotting window.

The resolution of a synthetic light curve is defined by the number of vertexes that we use for plotting. This value is held by an entry labelled `Vertexes`. If we have steep and narrow eclipses, we need finer resolution, especially in eclipse regions. The number of vertexes defaults to 100, but sometimes this does not suffice for accurate inspections. Leave it at 100 for the time being.

The top right part of the LC plotting window is used to assign the variables of our plot. As usual, the independent variable is plotted on x-axis and the dependent variable is plotted on y-axis.

Independent variable may be time (HJD) or phase. Dependent variable is a bit tricky. It may be:

- `Magnitude`; for synthetic light curves the magnitude level at quarter phase is set by the `Mag. Base` value[3]. The scaling is handled internally by `PHOEBE`, so only a vertical shift should be accomodated by the user. Experimental data is plotted directly if input files contain magnitudes; if they contain fluxes, the values are transformed using the normalization value in the `Data` tab, which we already explained.

[3]Keep in mind that `PHOEBE` knows nothing about magnitudes and magnitude plots are your your convenience only. Thus magnitude base value has no modeling significance.

- `Total Flux`; this is the most significant quantity for `PHOEBE`. All calculations are performed with respect to the total flux. By total flux we mean the sum of the flux from the first star, the flux from the second star and the (optional) third light. The units of total flux are arbitrary and are made physically meaningful through luminosity values (`HLA`s and `CLA`s), as explained in detail in the next section. Experimental data is copied verbatim if the input files contain fluxes and is transformed using the normalization value in the `Data` tab, if they contain magnitudes.

- `Primary Star Flux`; this is the calculated contribution of the primary star. It has no relevant counterpart in experimental data, so that part of the output is suppressed.

- `Secondary Star Flux`; this is the calculated contribution of the secondary star. This too has no relevant counterpart in experimental data and is thus suppressed.

- `Normalized flux`; this quantity is again for your convenience only. Some `WD98` users prefer to adjust the shape of a light curve first and fix the flux level at the very end. This approach seems a bit vague and is deprecated in `PHOEBE`, but it is still available. The normalized flux is the same as the total flux by all means, except that it is re-normalized to the value given by the `Flux Factor` (usually 1) at the phase given by `Phase Norm`[4] (usually 0.25).

If the independent variable for our plot is phase and what we have in our input files is heliocentric Julian date, then `PHOEBE` automatically transforms HJD to phase according to the origin of time and period that we set in the main `PHOEBE` window. This automatic procedure transforms all data points to have phases between -0.5 and 0.5. Thus, if we use a plotting range wider than this, the regions outside this interval will contain no data points. That is why `PHOEBE` allows you to *map* existing points to the whole region of the plot. This means that a single data point will occur multiple times, wherever the phase of that point is the same. This procedure is referred to as *aliasing*[5]. Aliasing is turned on through a `Turn on data aliasing` switch.

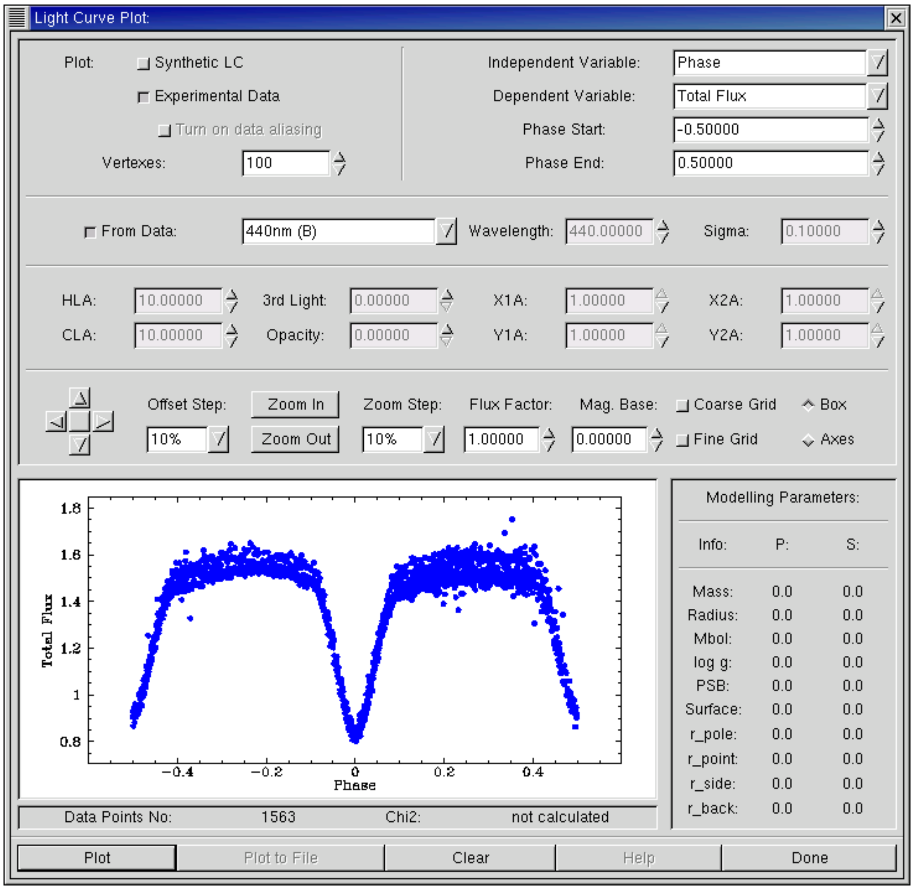
Let's first see whether our period and the origin of time make sense; thus we do not want to plot synthetic light curve, so turn the button `Synthetic LC` off. Set the data filter to `440nm (B)` by turning the `From Data` switch on and picking that filter from the list. Now turn the `Experimental Data` switch on. We shall plot the total flux against phase (which is the default). The phase range lets you select what phases you want plotted. This too we shall

[4] This parameter may be currently found in the `Switches` tab, but this is obsolete; it will soon be moved to the plotting window.

[5] Alias stands for "also listed as".

at this time leave at their current values. Flux factor and magnitude base values play no role at total flux plots, so just ignore them.

When we are content with the plotting settings, click on the `Plot` button. The following plot appears:

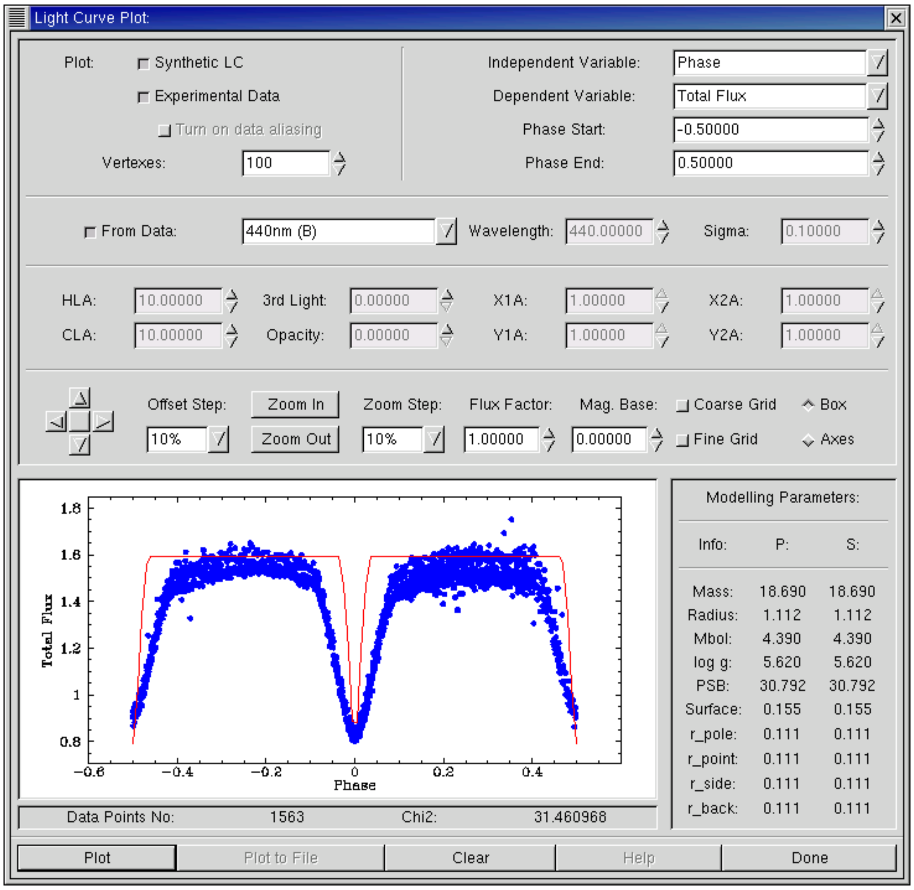


Plotted 440nm (B) light curve.

Whenever you click on the `Plot` button, the plot is displayed; underneath it you may find the number of experimental data points, in our case 1563. The value of χ^2 (denoted as `Chi2`) is calculated between experimental data and synthetic light curve, so it is naturally not calculated now that we have no synthetic lightcurve. On the right of the plot you may see the `Modeling parameters` part of the plotting window; these parameters are calculated from synthetic light curve parameters, so again they are not displayed for our first plot.

Now select the other light curve by choosing `550nm (V)` filter instead of `440nm (B)`. You may see that both light curves seem quite as expected.

Let us check how big is the difference between the initial synthetic light curve and our data! Turn on synthetic light curve (`Synthetic LC` switch) and click on `Plot` again! You are presented with the following picture:



Synthetic light curve with plotted 440nm (B) light curve.

Synthetic curve is denoted with a solid red line and experimental data with blue filled circles.

As you may see, the χ^2 is now calculated: 31.460968. This value is not normalized with dispersion or weighted; it is the difference between the consecutive plots that count. The value itself is the sum of squares of differences between the synthetic and experimental data.

Synthetic light curve computation fills in the `Modeling Parameters` window (on the right of the plot itself). Browse through the parameters to see what is contained in the output. Abreviations `P` and `S` stand for primary and secondary star, respectively. The masses are expressed in units of solar mass, radii (their mean value, to be exact) in units of solar radius. Bolometric magnitude `Mbol` and surface gravity potential `log g` have standard units[6]. `PSB` stands for polar surface brightness, `Surface` is in units of the Roche lobe surface size and all radii informations are given with respect to the semi-major axis. The values you see make no sense, but this is of course to be expected, since parameter values are way wrong. Don't worry about it, it will converge eventually.

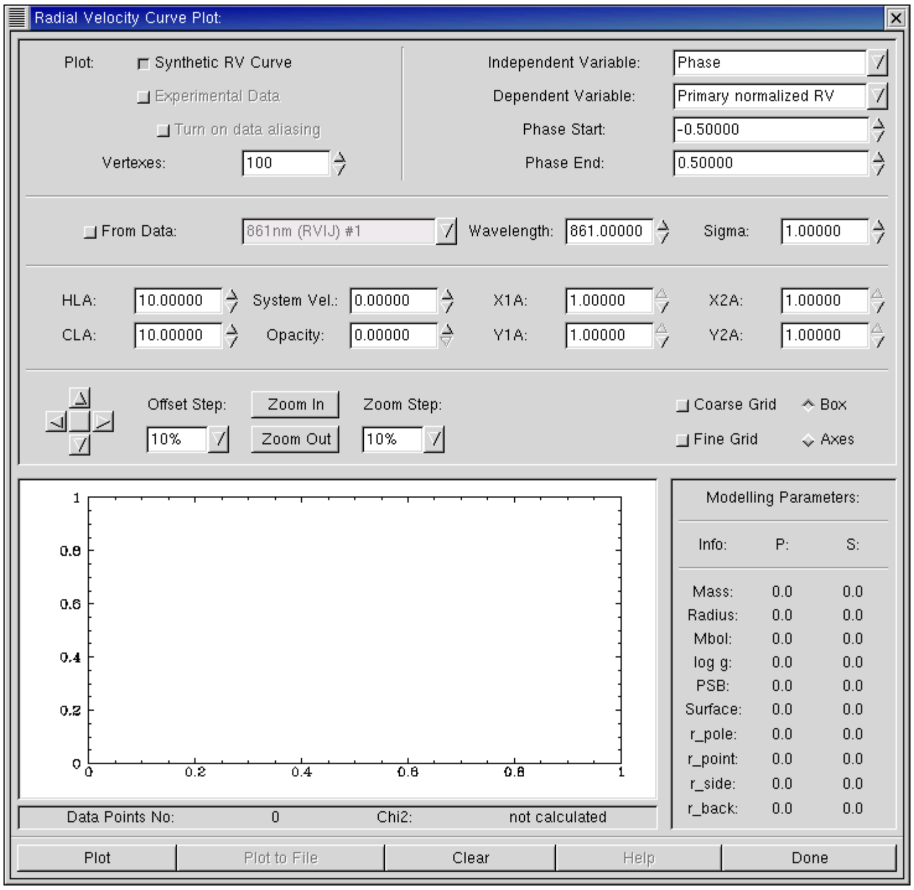
[6]Don't forget that `log g` is defined in the `cgs` system.

The fit between experimental points and the synthetic curve is, for the lack of a better word, unsatisfying. Inspect the same for the other light curve; this time the fit would better be described as catastrophic.

Just above the plot you may see plot control arrows and zooming buttons. These controls enable you to move your plot according to your needs. Try zooming in on the eclipse minimum, move the plot around and, all in all, get a feeling of the plotting window as a whole. A button in between the direction arrows is the reset button; clicking on it will restore original plot settings. On the far right of that part of the plotting window you see the grid options and the coordinate options for your plot. The grid (both coarse and fine) is used to easily read out the values from the plot directly. Some users might prefer axes to a boxed coordinate system and this option is supported too.

When you had your share of playing with the LC plotting window and have examined the results, click on the `Done` button to close the window. Note that you may keep the window open if you prefer – this is usually sound to do when your monitor resolution is very high (1600×1200 or so).

We shall now do much the same for radial velocity curves; open the RV plotting window by clicking on the `RV Plot` button on the button box.

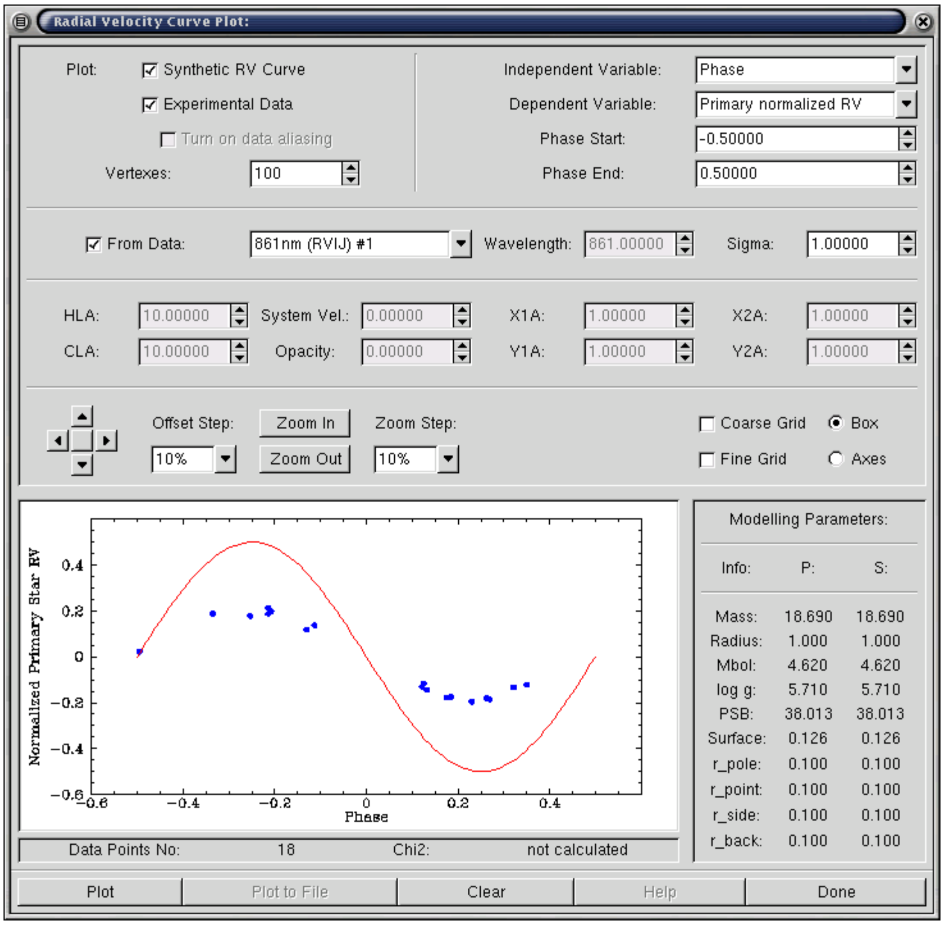


The radial velocities plotting window.

The shape and feel of the RV plotting window is much the same as the LC plotting window, although few differences should be stressed. Dependent plot variables are as follows:

- `Primary (Secondary) Normalized RV`; synthetic radial velocity curves normalized to orbit, meaning they are transformed to a dimensionless value by multiplying their value in km/s with $P_0[\mathrm{s}]/2\pi a[\mathrm{km}]$. Same is done with experimental radial velocity data. This means that we expect the value of normalized velocity around 0.5 if the mass ratio is around 1.

- `Primary (Secondary) eclipse proximity corrections`; this shows the effects of the eclipse proximity correction on radial velocities in same dimensionless units as explained above. Experimental data has no counterpart and its output is thus suppressed.



Synthetic RV curve with plotted primary star RV curve.

- `Primary (Secondary) RV in adopted units`; these are much like the first two choices, only that RVs are transformed to physically more meaningful units: the ones set by the `Velocity Unit` switch. Typically this is set to 1.0 if we want km/s as our adopted unit or 100.0 if we want 100km/s as our adopted unit. The transformation applies to both synthetic and experimental radial velocities.

The third light has no significance on radial velocity curves. Its place is thus switched by the system radial velocity, although this is **not** a curve dependent parameter. It causes the vertical shift to the radial velocity curve plot.

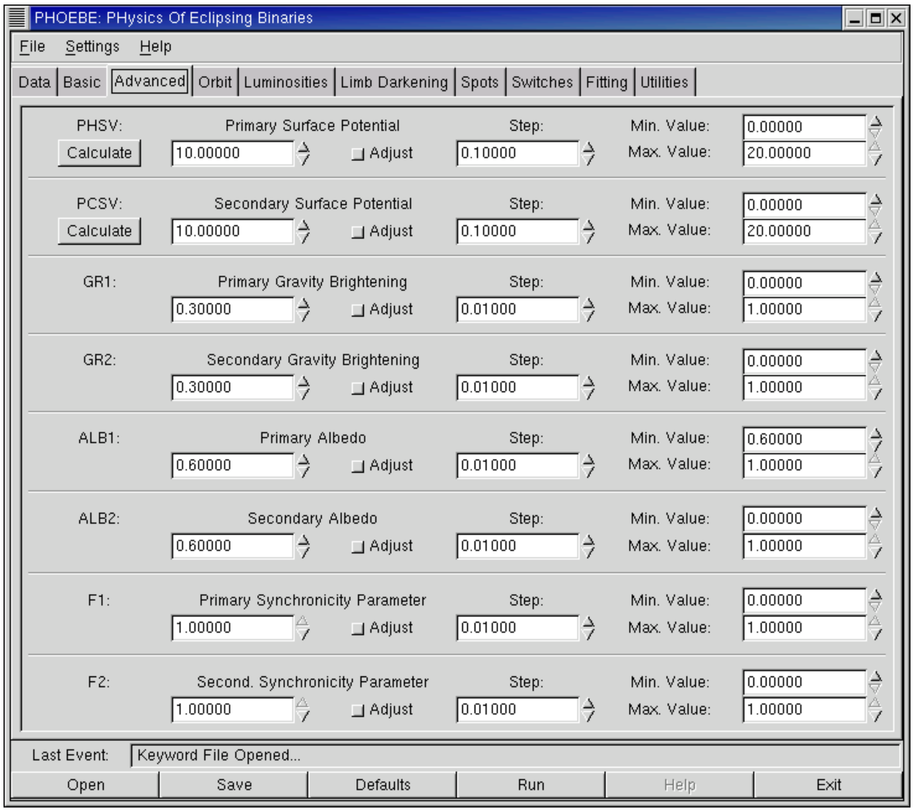
This time leave both experimental data and synthetic RV switches on; if you select the first filter (carrying #1 in its name), set `Primary Normalized RV` as the dependent variable. If you select the other RV curve, set `Secondary Normalized RV` as the dependent variable. Acknowledge that again the fit is miserable.

Feel free to examine other options of the plotting window. See what is accomplished when different choices are made. Move the plot around, turn the grid on and off, change the coordinate system shape to axes or boxes. Make your `PHOEBE` session as comfortable as possible. Again, if you wish, you may leave your RV plotting window open or close it by clicking on the `Done` button.

2.7 Advanced parameters

`UV Leonis` is a detached binary system and as such we may estimate the gravity potential. We are referring to modified Kopal potentials as defined in (Wilson 1979). Their `WD98` names are `PHSV` for a primary star and `PCSV` for a secondary star. There are supplied calculation routines which you may access by clicking on the `Calculate` button.

After clicking on the `Calculate` button, a new window (figure on the next page) will be shown. There is built-in support for both circular and elliptical orbits. You may quickly calculate the value of the potential by supplying the instantaneous separation between both stars, normalized to the semi-major axis (thus always 1.0 for circular orbits), the mass ratio and the radius-vector to the desired point in space, again in units of semi-major axis. To calculate the modified Kopal potential, set this radius-vector to the star radius towards the companion star (if tidal forces are strong, the star is deformed into a dumbell shape, so stellar radius is direction-dependent). In case of elliptical orbit the potential will depend on synchronicity parameter and direction cosines, but this is out of scope of this tutorial. Please refer to `PHOEBE` Reference Manual for further assistance.

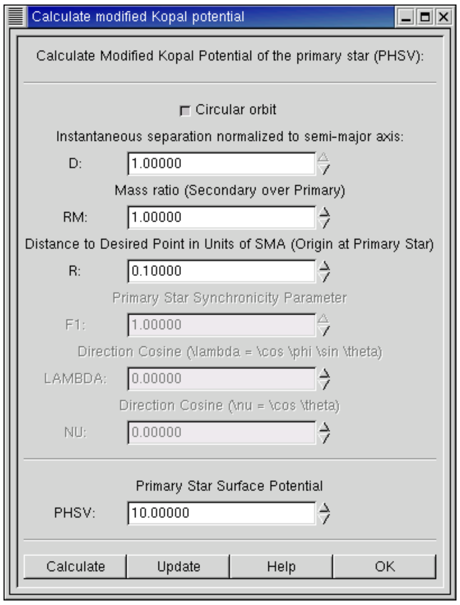


PHOEBE *advanced tab.*

On the lowermost part of the calculation window you may see a button box. The `Calculate` button calculates the value of potential, but does nothing else. PHOEBE lets the user inspect it and decide whether the calculated value is acceptable or not. If not, you may close the calculation window by clicking on the `OK` button and there will have been no changes made. If so, clicking on the `Update` button will copy the calculated value to the main PHOEBE window.

`UV Leonis` seems to have a pretty circular orbit and in the first approximation it seems that the default values in the calculation window are as might be expected (we will be proven wrong shortly, but this in fact is the realistic presumption). Therefore click on the `Calculate` button. The value 11.00504 will show up in `PHSV` entry. Since we are happy with it, click on `Update` button. The value will be copied to the `Advanced` tab of the main PHOEBE window. Close the calculation window by clicking on the `OK` button.

Repeat the same procedure for the secondary potential. The value will be the same.



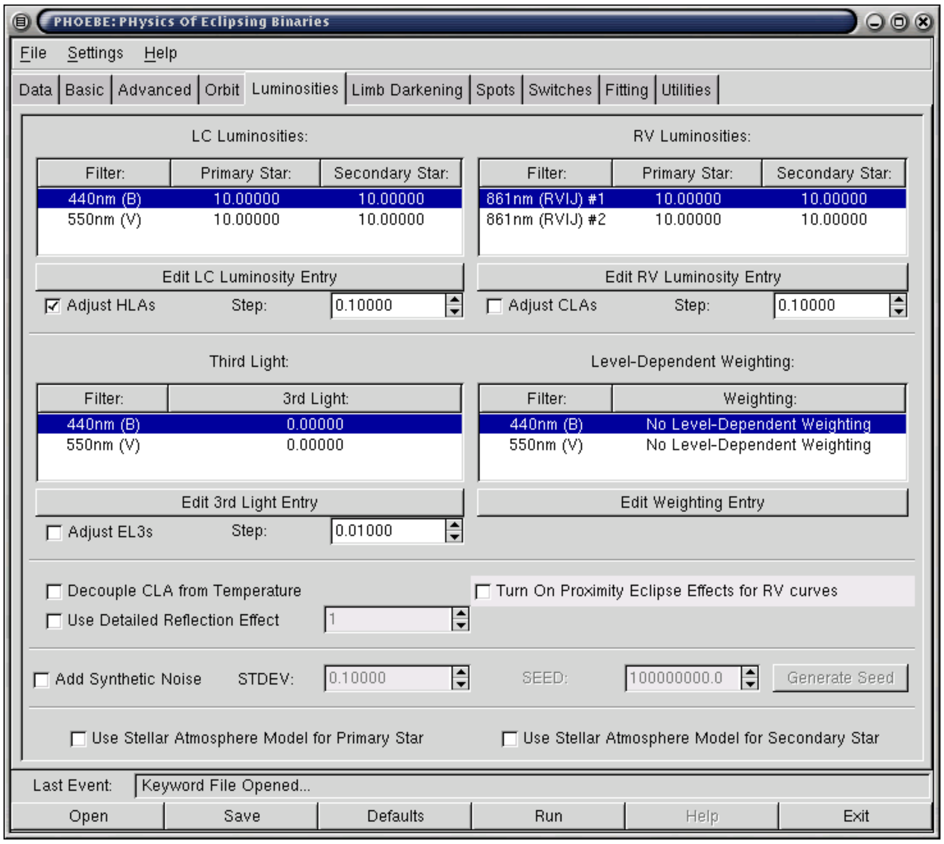
The calculation window of modified Kopal potentials.

At this time we cannot safely presume anything about the values of gravity brightening, albedo and synchronicity parameters, so we continue without modifying anything.

3 First steps of fitting

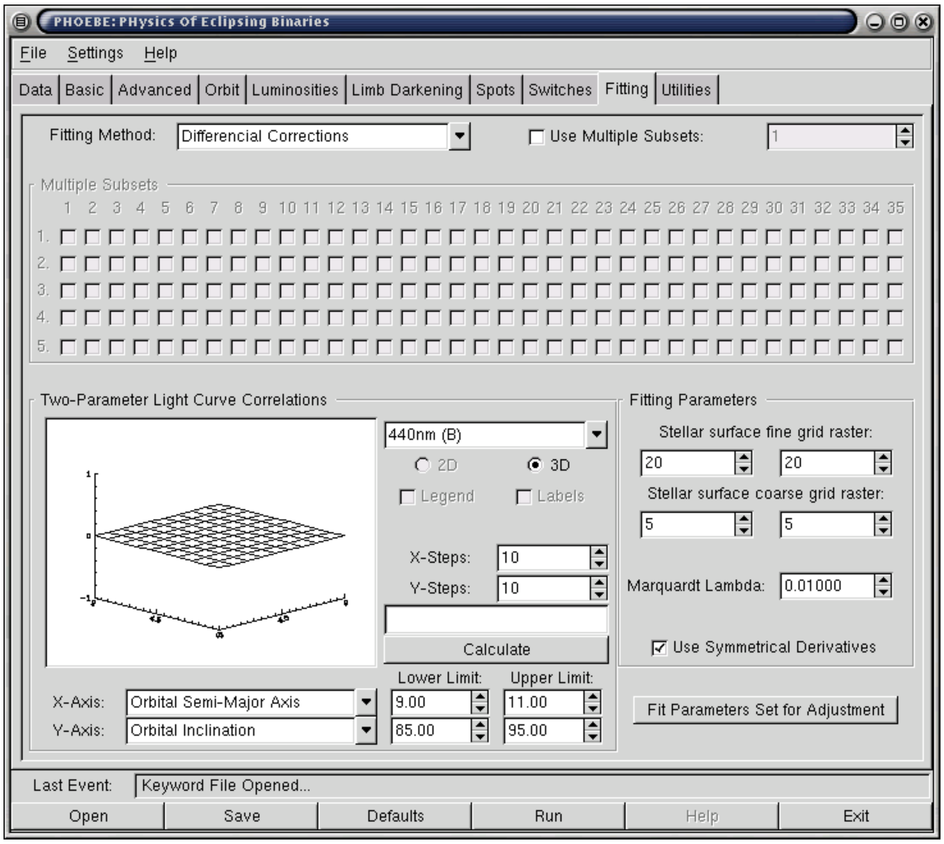
There is a preferable order of what parameters should be fitted first and what last, what should be coupled with what, what changes when some global change is imposed. The easiest way is to advise you with patience: as you gain experience, you will get the feeling of how to approach certain types of problems. But the usual recipe would be this: **always** start with luminosities! These quantities, although carrying physical name, are actually "normalizers" of your experimental data; original minimization routines work with fluxes, not with magnitudes, so `PHOEBE` internally does all the necessary conversions. These conversions are based on the normalizing magnitude that is set in the `Data` tab. It defaults to 10, but should be set to the magnitude of the quarter phase; as this cannot (and shouldn't) be done too accurately, luminosities jump in and normalize the data so that the normalization is actually done internally and you don't have to worry about it! The luminosities are in `WD98` style referred to as `HLA` and `CLA`. Let us demonstrate!

Go to the `Luminosities` tab (picture below) and turn on the `Adjust HLAs` switch. Leave the step as it is, since we are going to use Levenberg-Marquardt's λ multiplicator routine, which automatically chooses a proper step size when close to the solution.



`PHOEBE` *luminosities tab.*

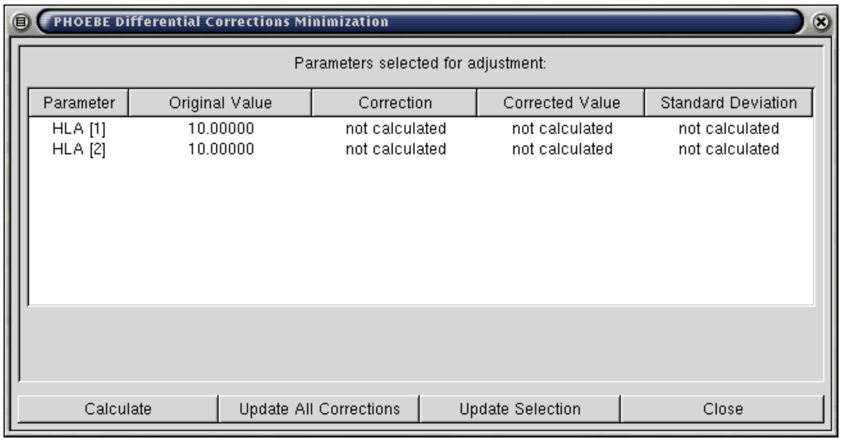
Now go to the `Fitting` tab; the upper part of the tab is dedicated to the method of multiple subsets, which we will not using in this tutorial; The lower left part belongs to the two-dimensional part of the whole parameter phase-space, which helps us analyse parameter correlations. We shall demonstrate its use in the course of this tutorial. On the right, there are fitting parameters, which are generally a good choice. Change grid sizes only if stellar shapes deviate significantly from spheroids, change Marquardt's λ only if convergence isn't satisfactory. Choosing symmetrical derivatives is a better approximation, but at expence of computing time. Since this time isn't that important at the moment, turn on symmetrical derivatives! After you have gotten yourself comfortable, click on the `Fit Parameters Set For Adjustment` switch!



`PHOEBE` *fitting tab.*

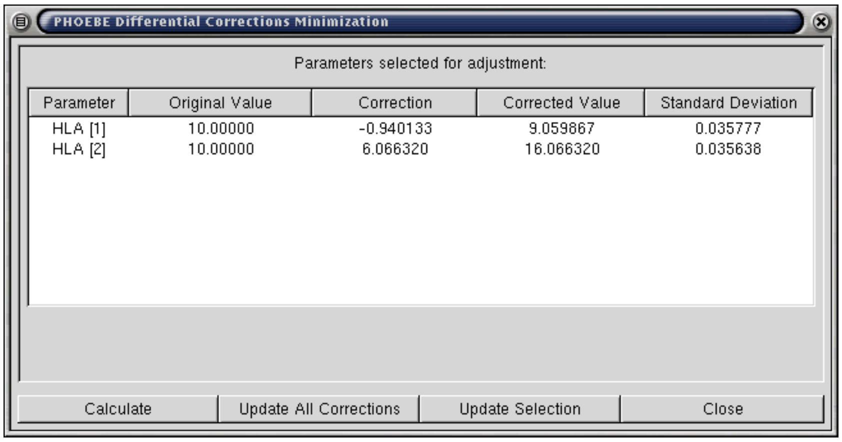
After you have clicked on that button, a fitting window (a picture on the next page) will show up. The columned list contains all parameters selected for adjustment. In case of curve-independent parameters there is only one entry and in case of curve-dependent parameters there are as many entries as there are affected curves. Light curves and radial velocity curves differ in this respect and have each their own adjustment switches.

Along with parameter names there are their current values. The fitting algorithm will calculate the correction of **one** iteration. Be very careful about these values, for they are the best marker to show if the fit converges or diverges. Luminosities essentially always converge, but you can never be too sure. The window also shows the corrected value and standard deviation of the fit of the particular parameter.



`PHOEBE` *fitting tab.*

Now click on the `Calculate` button! After some seconds of calculating (depending on the speed of your computer), the following window will arise:

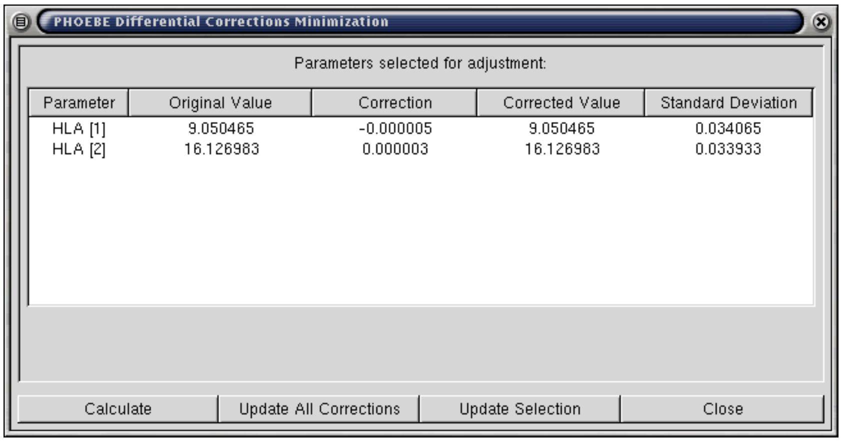


`PHOEBE` *fitting window.*

You may see that both corrections (and their corrected values) are in agreement with what we expect (recall what offsets we saw in plotting windows). Since we are happy with the fit results, we want to keep them. Thus click on `Update all corrections`. This particular columned list is designed in a way that enables you to update only those parameters out of all displayed that you see fit, but it is in its testing stages and therefore not available at this time; you will have clicked (selected) those you wish and clicked on `Update Selection`.

When you clicked on `Update all corrections`, the corrected values replaced the original in both the fitting window and in the main `PHOEBE` window. The fitting function is now ready for the second iteration, so click on

`Calculate` button again! The corrections will now be even smaller and the convergence obvious. Update the corrected values and recalculate the values for as many times as the corrections don't become small enough to your satisfaction. This means that as long as you are far away from the solution, be sure to let the values converge on a decimal or two, but as the fit gets better and better, try to be as accurate as possible. If you persisted completely through the convergence, you should have come up with these values:



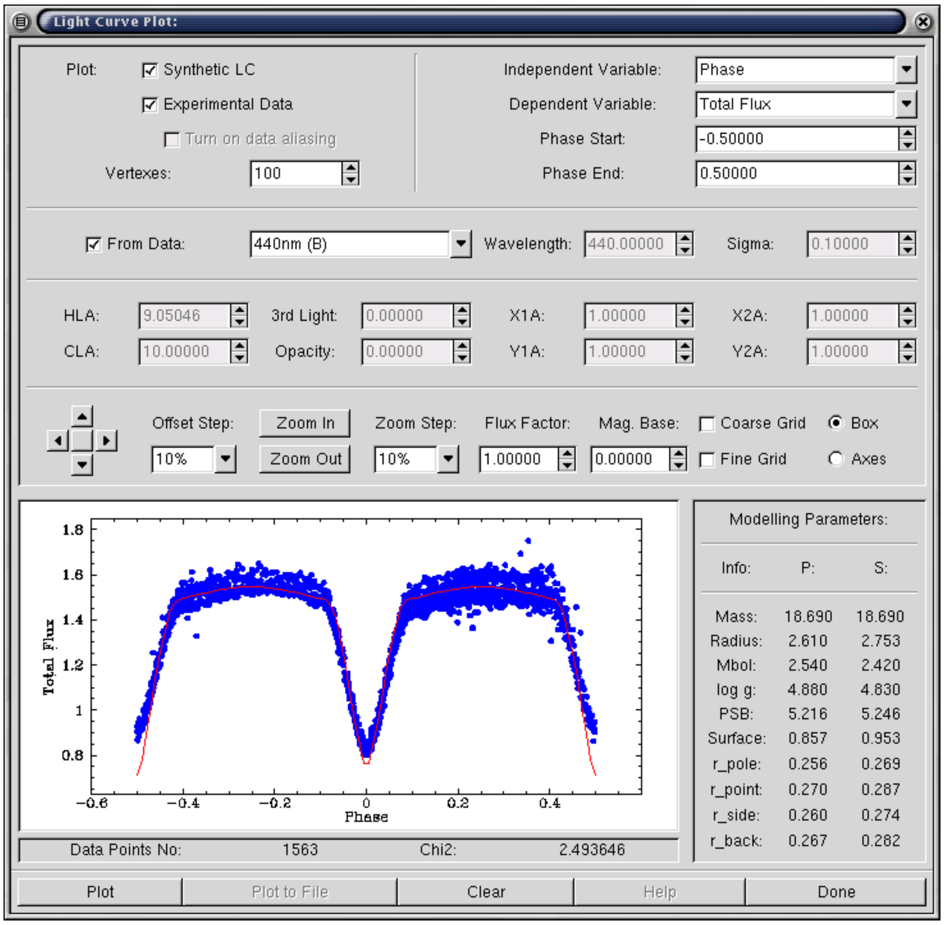
`PHOEBE` *fitting window.*

When you've finished (and are happy) with fitting, close the window by clicking on the `Close` button. Go to the `Luminosities` tab, turn off the adjustment switch of the luminosities and inspect the results, as you have before, for both light curves. Notice any difference? Still not perfect, but much better.

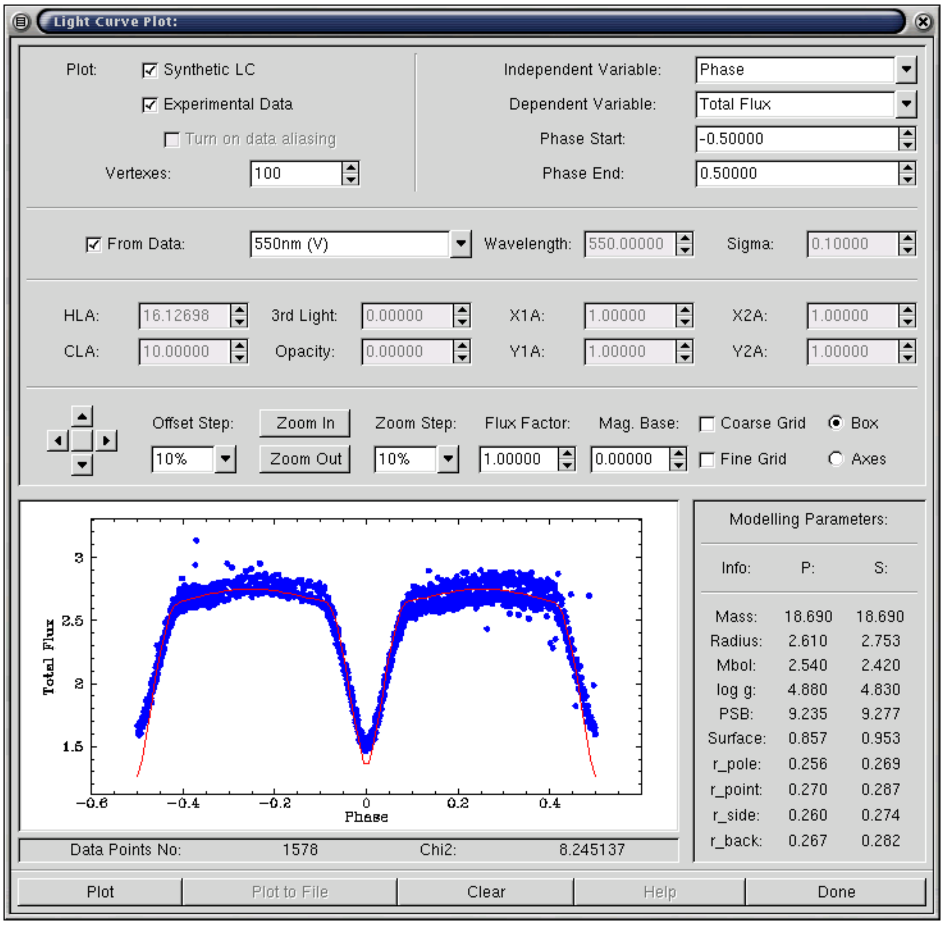
4 I feel comfortable, what next?

What's next basically depends on your particular problem. In our case, it seems that the shape of the light curve is way off. The most influencing parameters are Kopal potentials, so let us try and fit those. Go to the `Advanced` tab and mark both primary and secondary star potentials (`PHSV`, `PCSV`) to be adjusted. Go to the `Fitting` tab and do as before: correct the values of potentials one step at the time. Note that the convergence is slower than that of the luminosities, but be patient and keep going until you have reached satisfactory values. A piece of advice: keep a plotting window open and inspect what each iteration does to your synthetic light curve! To do this, turn on and off the `From Data` switch to update the values (this will be simplified in the next version of `PHOEBE`: a before-and-after images will be present in the fitting window itself) and then click on `Plot` to get the actual updated plot. And I just discovered a bug: don't update parameters when there is no update calculated, it will set your parameters to 0. I will fix it shortly.

If you have followed the above scheme, you should have landed with the primary and secondary potential values of 4.878 and 4.686, respectively. We now see how way off we have been with our calculated values. You may play with the values in the potential calculation window to see what presumption was wrong. The corrected values produce the following for both light curves:



A 440nm (B) light curve after the fit of potentials.



A 550nm (B) light curve after the fit of potentials.

This is much better already!

... TO BE CONTINUED ...