Manual for the Eclipsing Light Curve (ELC) Code
Version 7

Jerome A. Orosz
Department of Astronomy
San Diego State University
5500 Campanile Drive
San Diego, CA 92182-1221

November 20, 2019

Contents

1. Citation

The proper reference for the ELC code is:

Orosz, J. A., & Hauschildt, P. H. 2000, A&A, 364, 265

2. Acknowledgments

My involvement with this code started in 1994 during graduate school at Yale. I wish to thank my thesis adviser Charles Bailyn, as well as Jeff McClintock and Ron Remillard for their support during those years. The "modern" ELC code started during my postdoc days in Utrecht, and special thanks go to Marten van Kerkwijk and Frank Verbunt for support during those times. Don Short at San Diego State University developed many of the algorithms at the heart of the most recent version of the ELC family of codes.

3. Introduction

Yorham Avni was interested in, among other things, mass determinations of the compact objects in high mass X-ray binaries (e.g. Cyg X-1, Cen X-3, etc.). During the course of his research he wrote a FORTRAN code to compute the ellipsoidal light curve of a single (usually) Roche-lobe filling star (Avni & Bahcall 1975; Avni 1978). Avni passed on this code to Jeff McClintock and Ron Remillard shortly before his death in March of 1988 (see McClintock & Remillard 1990). The code came to me via Ron Remillard sometime in 1994 when I was a graduate student at Yale. During the spring of 1995 it became clear that the black hole binary GRO J1655-40 was an eclipsing system and that the Avni code in its original form was not adequate. I substantially modified the code in the summer of 1995 to include light from an accretion disk and to account for eclipses (Orosz & Bailyn 1997; OB97).

The code described in OB97 is somewhat unwieldy. It is very difficult to read and modify. I decided it was time to have a new code that is more general, more modular, easier to read, and most importantly, easier to modify and build upon. This present code goes a long way in meeting this goal. This present code is nearly a completely new code, written from the ground up. Although I have used much of Avni's notation, and I have followed his basic method of setting up the Roche geometry and integrating the observable flux, most of the code is my own (any errors are due to me). I have been liberal with comments in the code, so hopefully it will be much more readable to others.

The major features of this current version include:

- the ability to model any binary star system with a circular orbit, except for over-contact binaries (i.e. W UMa stars)
- the extension to eccentric orbits
- the ability to add an accretion disk around the second star
- the detailed treatment of the reflection effect (Wilson 1990)

- the use of local intensities computed from detailed model atmospheres
- more accurate integration, especially during eclipses and transits with extreme ratios of radii
- fast analytic approximations for modeling transit light curves of extra-solar planets
- more variety in the ways one can specify the geometry
- multi-body systems (circumbinary planets, triple star systems, etc.)
- the ability to calculate light curves in time units
- better graphics (using separate programs)
- more optimizer choices

4. Exhortation and Disclaimer

Apparently you have observations of some close (or not so close) binary star and you want to derive some quantities of astrophysical interest. You may use my codes to do so. However, please understand that I cannot guarantee that the codes are 100% error free or that they are appropriate for your situation. These codes seem to work well for me. I have experience in modeling binary light curves, and I can usually tell when something does not make sense. If you find a bug or find something that does not make sense, please let me know.

I don't mind if you modify the code, provided that (i) you *really* know what you are doing; and (ii) you don't give out modified copies to others. I cannot overemphasize the importance of really knowing what you are doing when tinkering with the code. Changes in the code in one place may cause errors in a different part of the code, and in some cases these errors may not be obvious.

5. Outline of the Methods

As time permits, I will include a detailed outline of the algorithms used in the code and details on their implementation. Until then, you can read any number of papers on the subject (e.g. Wilson & Devinney 1971; Wilson 1979; Wilson 1990; Avni & Bahcall 1975; Avni 1978; OB97; Orosz & Hauschildt 2000).

6. Compiling the Codes

The main program is **`ELC.f90.`** It includes the routines in **`lcsubs.f90`** using the FORTRAN **`include`** statement. It is written in free-format FORTRAN 90.

To compile ELC, the best option at the moment is to use the gfortran compiler, which works on most Linux systems and also MacOS:

```
gfortran -O3 -o ELC ELC.f90
```

The `-O3` flag is for the optimization (this is the letter O). You should be able to compile the other optimizer codes (discussed in Section **8.4**) using similar commands.

The codes will also compile using the PGI compilers:

```
pgfortran -fast -o ELC ELC.f90
```

Additional flags are available for more aggressive optimization:

```
pgfortran -fast -Mfprelaxed -Mipa=fast,inline -o ELC ELC.f90
```

Note, however, that the codes won't compile for parallel operation (see Section **8.7**) with PGI versions 19.x. The PGI-compiled executable is one or two percent faster than the executables compiled with gfortran.

I don't have much experience with the Intel FORTRAN compiler, but this command works:

```
ifort -O3 -xHost -assume buffered_io -fp-model fast=2 -no-prec-div -o ELC ELC.f90
```

You should be able to compile the other optimizer codes (discussed in Section **8.4**) using similar commands.

7. Running the ELC Code

7.1 Overview of ELC

ELC computes a "forward" model. One specifies the geometry of the binary, the radiative properties of the stars, and ELC computes observable quantities like light and velocity curves. In particular, the code can compute:

- The light curve and radial velocity from one visible star (star 1) using Roche geometry (there is obviously another body present but it might not be optically visible, think X-ray binary or a red giant/main sequence star system where the light from the red giant dominates). The most common use for this option would be ellipsoidal variables. The orbit can be circular or eccentric.

- The light and radial velocity curves from two stars (star 1 and star 2), either using Roche geometry for distorted stars or fast analytic expressions for spherical stars. The orbit can be circular or eccentric. If you are using Roche geometry gravity darkening and the reflection effect can be included, as can star spots.

- The light and radial velocity curves from one or two stars and the light curve from an accretion disk, using Roche geometry for the stars. If you have a disk, it *must* be around star 2. Note that star 2 can be invisible, as in an X-ray binary. Reflection effects, gravity darkening, and X-ray heating can be included, as can spots on the star or on the disk.

- The light and radial velocity curves of multi-body systems, like circumbinary planetary systems or hierarchical triples, or "double-binary" systems. In this case, the stars (and planets) are assumed to be spherical and fast analytic expressions are used to compute eclipses and transits.

If you have a single star, it is always referred to as "star 1". The phase convention is as follows: for circular orbits, star 1 is closest to the observer at phase 0.0. Star 2 (if present) and the disk (if present) would be eclipsed (for sufficiently large inclinations) at this phase. For eccentric orbits, phase 0.0 corresponds to the time of periastron passage. For cases with three or more bodies, the light curves are always computed in time units.

7.2 The input files

Here is a list of input files that ELC needs:

- **`ELC.inp`**: The main file for the input parameters, always required. Section **7.2.1** gives a line-by-line description of the required entries.
- **`ELC.atm`**: Optional file that contains the model atmosphere table of specific intensities (don't mess with this file under any circumstances!). This atmosphere file usually has the NextGen models for the stars cooler than 10,000 K, and Kurucz models thereafter, for the standard UBVRIJHK filters. Files with different model atmospheres and/or filters can be made upon request.
- **`ELCgap.inp`**: Optional file that specifies which portions of a long (in time) light curve that can be skipped. See Section **7.2.2**.
- **`ELCSC.inp`**: Optional file that specifies which portions of a long (in time) light curve contain *Kepler* short cadence (one minute sampling) data. See Section **7.2.3**.
- **`ELCdynwin.inp`**: If you are computing the light curves of a system with three or more bodies, a dynamical integrator is used to find the positions of the bodies as a function of time. Many intermediate quantities from this integration can be saved. This optional file specifies the time ranges where the intermediate quantities should be retained. See Sections **7.2.4** and **7.4.4**.

7.2.1 `ELC.inp`

The parameters in **`ELC.inp`** *must* be in the order given. There is no input format statement, so you can have leading spaces. Also, the variable names printed by default are ignored by the code. If you try to run the code when the file `ELC.inp` does not exist, the program will write a sample file with default parameters.

Note that starting with version 6 of the code the arrangement of `ELC.inp` has changed. The code should be able to read the "old" style input files, and Appendix A discusses the parameters for input files of this format. In addition, the code **`convertELCinp.f90`** will read an input file in the old format and output a suitable file in the new format. There are many new input parameters that are not available in the old input files, so using old input files does not make sense.

The top line of **`ELC.inp`** should contain this:

`#1 New version 7 input, this line required`

In the new style input file, comment lines can be inserted anywhere past line 1. To specify a comment line, put `!` or `#` in the starting position of that line. Apart from comment lines, the parameters given in `ELC.inp` must be given in the correct order. How many lines you will need will depend on the number of bodies. In the following I give the "sections" of `ELC.inp` in **bold**, followed by the required lines for that section using bullet points (•). Generally there is one parameter per line, with exceptions noted in red. Parameters that start with the letters `a` through `h` or `o` through `z` (not case sensitive) should be *real* numbers and parameters that start with the letters `i` through `n` (not case sensitive) should be *integers*.

Units

- `iunit`: Use 0 (the number zero).

Parameters for light curve sampling

- `itime`: Set to 0 (zero) for input data in phase units; set to 1 for input data in time units, but model computation in phase units; set to 2 for input data in time units and model computation in time units.
- `t_start`: Starting time for model in days, if `itime=2`.
- `t_end`: Ending time for model in days, if `itime=2`.
- `t_step`: Step size for model in days, if `itime=2`.
- `dphase`: Phase interval in degrees for the model curves if `itime=0` or `1`.

Control flags for filters and limb darkening

- `iatm`: Set to 0 (zero) for black body intensities; set to 1 for specific intensities from `ELC.atm` and Roche geometry; set to 2 to compute out-of-eclipse fluxes from `ELC.atm` when using the fast analytic expressions. If `iatm=0` or `iatm=2` you need to specify a limb darkening law and supply the appropriate coefficients.
- `icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK`: Need 8 integers for this line. These 8 integers, separated by spaces, are switches for the "filter wheel". Set `icnU=1` to compute the light curve for the U band, or `icnU=0` to skip; set `icnB=1` to compute the B-band light curve, `icnB=0` to skip, etc. When using the model atmosphere table (`ELC.atm`), the bandpasses may not necessarily be the Johnson $UBVRIJHK$ filters, but the labels for the output files are always `U`, `B`, etc.
- `iRVfilt`: This is the index of the wavelength (or filter) used to compute the radial velocity curves. Note that the form of the radial velocity curves is practically independent of the input wavelength. This switch also controls which light curve is output in the `lcstar?.linear` and `lcdisk.linear` files (described below). Thus set `iRVfilt=3` to compute radial velocity curves for the V band and to output individual V-band light curves, etc.

- **ilaw**: This controls the form of the limb darkening law. Set **ilaw=1** for the linear law:

  $$I(\mu) = I_0(1 - x + x\mu)$$

  (where μ is the cosine of the foreshortening angle of the grid element). Set **ilaw=2** for the logarithmic law:

  $$I(\mu) = I_0(1 - x + x\mu - y\mu \ln \mu).$$

  Set **ilaw=3** for the square root law:

  $$I(\mu) = I_0 \left[1 - x + x\mu - y(1 - \sqrt{\mu})\right].$$

  Set **ilaw=4** for the quadratic law:

  $$I(\mu) = I_0 \left[1 - x + x\mu - y(1 - \mu)^2\right].$$

  Set **ilaw=5** for the quadratic law with Kipping's (2013) reformulation that uses "triangular" resampling. Set **ilaw=6** for the logarithmic law with triangular resampling (Espinoza & Jordán 2016). Finally, set **ilaw=7** for the square root law with triangular resampling (Kipping 2013).

  You can find tables of the coefficients in Van Hamme (1993). Note that the same law is used in the computation of the reflection effect and in the wavelength dependent intensity computations. If you have a binary with nearly identical stars, you can add 10 to the **ilaw** flag of one of the above values to force the limb darkening laws for both stars to be the same. Thus, if **ilaw=12**, then both stars will have the same logarithmic law with the same coefficients.

- **lambda_1**: Central wavelength (in Å) for filter 1 when **iatm=0**.

- **lambda_2**: Central wavelength (in Å) for filter 2 when **iatm=0**.

- **lambda_3**: Central wavelength (in Å) for filter 3 when **iatm=0**.

- **lambda_4**: Central wavelength (in Å) for filter 4 when **iatm=0**.

- **lambda_5**: Central wavelength (in Å) for filter 5 when **iatm=0**.

- **lambda_6**: Central wavelength (in Å) for filter 6 when **iatm=0**.

- **lambda_7**: Central wavelength (in Å) for filter 7 when **iatm=0**.

- **lambda_8**: Central wavelength (in Å) for filter 8 when **iatm=0**.

- **Nref**: The number of iterations for the reflection effect (see Wilson 1990). To skip the reflection effect routines entirely, set **NREF=-1**. If **NREF=0** a "simplified" computation is used. This approximation is exact for spherical stars and becomes worse as the stars deviate from spherical symmetry. If **NREF>0**, then the "detailed" reflection is called **NREF** times. **NREF=1** is adequate for most purposes. In most situations in the black body mode, the computation of the reflection effect takes most of the time.

  If you have an X-ray binary and want to compute the effects of X-ray heating, set **Nref=0** and set the parameter **Lx/Lopt** accordingly. By default, the X-ray source is assumed to be a thin disk. If you want to make the X-ray source a point source, set **iXheat=1** (see below).

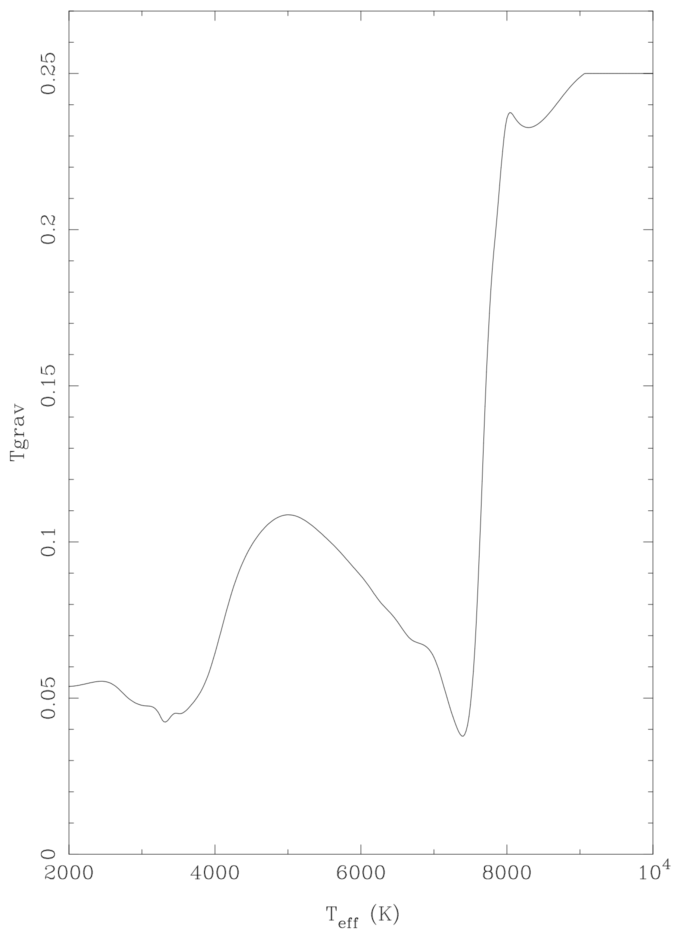

Figure 1: The values of **Tgrav1**, **Tgrav2** set by the code when **igrav>0**. From Claret (2000).

- **Lx/Lopt**: This is the logarithm (base 10) of the X-ray luminosity of star 2 in ergs s^{-1} in the cases where it is invisible—i.e. **Teff2<0**, tag Lx. This number is used in the computation of X-ray heating.

- **iXheat**: If **iXheat=1**, the X-ray source is assumed to be a point source when computing X-ray heating. If **iXheat=0**, then a foreshortening correction for the X-ray heating is applied, assuming the X-ray emitting part of the accretion is a thin disk in the orbital plane.

- **idark1**: If **idark1=1** then star 1 will not contribute to the light curve of the system.

- **idark2**: If **idark2=1** then star 1 will not contribute to the light curve of the system.

- **ispotprof**: This controls how the temperature over a star spot varies. If **ispotprof=0** then the temperature over the spot is uniform. If **ispotprof=1** then the temperature over the spot increases linearly in the radial direction from zero at the outside to the maximum temperature at the center. If **ispotprof=2** then the temperature profile over the spot area is Gaussian.

- **igrav**: Set **igrav=0** to use the input values of **Tgrav1,Tgrav2**, set **igrav=1** to have the code choose **Tgrav1** for you based on the input value of **Teff1**, set **igrav=2** to have the code choose `Tgrav2` for you based on the input value of `Teff2`, and set `igrav=3` to have the code choose *both* `Tgrav1` and `Tgrav2` for you based on the input values of `Teff1` and `Teff2` (Claret 2000, see Figure 1).

- `iflux`: Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline fluxes directly, using the variables `flux1_U`, `flux1_B`, etc. If `iflux=2`, then the parameters give flux ratios for stars 2 through 5, where the ratios are `flux2_U`/`flux1_U`, etc. (e.g. star 2's flux divided by star 1's flux, etc.).

Third light parameters

- `T3`: Third light temperature in Kelvin, tag t3.
- `g3`: Gravity of the third light star ($\log g$ with g in cgs units), tag g3.
- `SA3`: Ratio of the area of star 1 to the area of the third light star (both projected on the sky plane), tag SA.

Control flags related to speed and accuracy

- `Nterms`: This flag, when set to 1 or larger, activates the fast analytic mode whereby the bodies are assumed to be spheres. If you are using the Gimenez (2006a) routines (`mandel=0`) `Nterms` is the number of terms in the polynomial expansion. `Nterms=300` is a good number to start with.
- `mandel`: When `Nterms>0`, this switch controls which algorithm to use for the analytic expressions. Set `mandel=0` for the Gimenez (2006a) routines, `mandel=1` for the Mandel & Agol (2002) routines, `mandel=2` for the routines based on the work of Don Short for light curves only, and `mandel=3` for both light curves and Rossiter effects using Don Short's algorithms. The Mandel & Agol (2002) and the Short routines are faster than that of Gimenez (2006a). Note, however, that the computation of the Rossiter effect is implemented in the Gimenez (2006a) routine and in the Short routines, but *not* in the Mandel & Agol (2002) routine. Finally, the Gimenez (2006a) routine and the Mandel & Agol (2002) routine are both restricted to the linear and quadratic limb darkening laws, whereas the Short routine implements all of the limb darkening laws.
- `Ngap`: When `itime=2`, this switch allows the user to skip various segments in the light curve. If `Ngap > 0`, the code looks for a file called `ELCgap.inp` (see Section **7.2.2** below). This file should contain `Ngap` lines giving the starting and ending times of the gaps that the code should skip. If there is a problem reading this file, the code defaults to `Ngap=0`, the result of which being that the entire light curve between `t_start` and `t_end` is computed.
- `iecheck`: This flag applies only when Roche geometry is used to compute light curves (`Nterms=0`). If the accretion disk is axisymmetric (i.e. there are no spots), the integrated flux of the disk does not depend on the orbital phase, except for possible eclipses. Thus you need to compute the disk flux during the eclipse phases, and only once thereafter. Set `iecheck=0` to compute the disk flux at each phase, or `iecheck=1` to skip the computation

at the phases outside of eclipse. If you set **iecheck=-1** the code will not check for eclipses. Thus you can see how much difference the eclipses make. The increase in the execution speed is quite small when the eclipse checking is turned off. Note: if you have **iecheck=1**, then the intensity maps for the disk (see the **idraw** option below) will be *incorrect* for most phases (the light curve will of course be correct).

When you have a normal binary star with no disk that uses Roche geometry, the flag **iecheck** can be used to cut down the execution time. Set **iecheck=5** and the code will compute the light curve at 2 degree intervals outside of eclipse. Set **iecheck=9** and the code will compute the light curves only during eclipses. A constant "reference" flux will be used for the out-of-eclipse light curves. This is useful for the computation of transits of extrasolar planets where the ratio of radii is often extreme. If you have an eccentric orbit, set **iecheck=1**.

- **ism1**: This flag applies only when Roche geometry is used to compute light curves (**Nterms=0**). Set **ism1=0** to force the code to compute the entire light curve and **ism1=1** to only compute "half" of the light curve. If the orbit is circular, only the phases 0 to 180 degrees are computed if **ism1=1**, and the rest of the phases are filled in by symmetry. Note that the time savings is far less than a factor of two on average since the "setup" time dominates (the initial geometry and reflection effect computations). If the orbit is eccentric, the program makes use of a symmetry of the *geometry* around periastron if **ism1=1**. For example, the geometry (instantaneous orbital separation, filling factors, etc.) of the binary 10 degrees before periastron is the same as the geometry 10 degrees after periastron. Since the time to compute a light curve is dominated by the "setup", setting **ism1=1** for eccentric cases almost always saves a factor of two in computing. Note if you have spots, then you must set **ism1=0** since you no longer have symmetry. **If you get a strange looking radial velocity curve for an eccentric orbit, set the ism1 flag back to zero.**

- **ifasttrans**: This flag was part of an effort to speed up the computation of transit light curves of extrasolar planets when using Roche geometry (**Nterms=0**). If the star has symmetry (e.g. no spots or the like), the flux from the parts of the star that are not eclipsed will be the same always. Setting **ifasttrans=1** will have the code use fewer pixels in the parts of the star that are not eclipsed. However, it turns out that this flag does not really help, so using **ifasttrans=0** is recommended.

- **MonteCarlo**: This flag is mainly for the computation of accurate transit light curves for extrasolar planets when using Roche geometry (**Nterms=0**). If **MonteCarlo=0**, then a simple interpolation scheme is used to correct for partially eclipsed pixels. When **MonteCarlo>10**, a Monte Carlo integration scheme is used to correct for fractionally eclipsed pixels. In practice, **MonteCarlo** should be set to something between 200 and 1000.

- **phaselow**: This parameter can be used along with **phasehigh** to specify a restricted range of phases to compute (when **itime=0**). Set **phaselow>0**, **phasehigh>0**, and **phaselow<phasehigh**. The units are degrees.

- **phasehigh**: This parameter can be used along with **phaselow** to specify a restricted range of phases to compute (when **itime=0**). Set **phaselow>0**, **phasehigh>0**, and **phaselow<phasehigh**. The units are degrees.

Control flags for Roche geometry

- **`isquare`**: This flag is used to control the number of longitude points per latitude row. **`isquare=0`** gives you the "elliptical" arrays that are used in the Wilson & Devinney code (fewer longitude points near the poles), and **`isquare=1`** gives you "square" arrays (the same number of longitude points at all latitude rows). The code might choke if **`isquare=0`** and if the number of latitude and longitude points (**`Nalph, Nbet`**) points is small. In this case, set **`isquare=1`**. If you want to draw the binary, then set **`isquare=1`**.

- **`iusepot`**: Controls the input variables for the filling factors of the two stars when using Roche geometry. Set **`iusepot=0`** to specify the filling factors using **`fill1`**, and **`fill2`** (the usual mode). Set **`iusepot=1`** to use the Ω potentials **`usepot1`** and **`usepot2`**. The use of the Ω potentials makes it much easier to compare the ELC light curves with Wilson Devinney light curves.

- **`usepot1`**: This is the "Ω-potential" for star 1, which is the Wilson Devinney equivalent of the **`fill1`**. If **`iusepot=1`** (see above), then the filling factor of the two stars are computed using these potentials. If the entered value of a potential is smaller than the critical potential, then the filling factor is set to 1.

- **`usepot2`**: This is the "Ω-potential" for star 2, which is the Wilson Devinney equivalent of the **`fill1`**. If **`iusepot=1`** (see above), then the filling factor of the two stars are computed using these potentials. If the entered value of a potential is smaller than the critical potential, then the filling factor is set to 1.

Drawing flags

- **`idraw`**: The control integer for writing output files. Use **`idraw=1`** and use Roche geometry (**`Nterms=0`**) to write output files that the separate graphics programs use. See Section **7.4.7** below.

- **`ionephase`**: If the value of the integer flag **`ionephase`** is 1, then the program will compute the flux for only one phase given by **`onephase`** below.

- **`onephase`**: If the value of the integer flag **`ionephase`** is 1, then the program will compute the flux for only one phase whose value is set by this parameter. This feature is useful if you want to make a picture of the binary at a specific phase.

Control flags for orbital elements

- **`pshift`**: The phase shift to be applied to the output light and velocity curves when **`itime=0`** or **`itime=1`**. The shift must be between -1.0 and 1.0. This parameter can be used to effectively switch stars 1 and 2 (e.g. put **`pshift=0.5`** to have the same phase convention as the Wilson-Devinney code). It is also useful when fitting phase-folded data folded on an ephemeris with uncertainties (tag ps).

- **`ikeep`**: Set **`ikeep=1`** to place the eclipse of star 2 at phase 0.0, and **`ikeep=2`** to place the eclipse of star 1 at phase 180.0. If **`ikeep=0`**, then the phase of the primary eclipse depends on the eccentricity and argument of periastron, as in the Wilson & Devinney code.

- **isynch**: Controls whether the stars rotate synchronously at periastron for eccentric orbits. Set **isynch=0** to have the code use the input values of **omega1** and **omega2** and **isynch=1** to have the code compute the values of **omega1** and **omega2** for you:

$$\Omega = \sqrt{\frac{1+e}{(1-e)^3}}.$$

- **ialign**: Set **ialign=1** to have the rotational axis of star 1 aligned with the angular momentum vector of the orbit. Set **ialign=0** to have the rotational axis misaligned. Adjust the parameters **axis_I** and **axis_beta** below accordingly. This parameter will mainly make subtle changes to the radial velocity curve of star 1 during transit (e.g. the Rossiter-McLaughlin effect).

- **iuseTconj**: Set this flag to 1 to use a time of primary eclipse as the reference time rather than a time of periastron passage.

- **iecosflag**: Set this flag to 1 to specify the eccentricity and argument of periastron of the binary by $e\cos\omega$ and $e\sin\omega$.

Control flags for use with optimizers

- **imag**: Set **imag=0** to input light curve data in magnitude units, or set **imag=1** to input light curve data in flux units when using the optimizer codes. See Section **8.4**.

- **ielite**: Integer flag (zero or larger) that is used to insert specific models into the initial populations of some of the optimizer codes. See Section **8.5**.

- **ifixgamma**: Set **ifixgamma=0** to have the input value of **gamma** used in the fitting one or both velocity curves. Set **ifixgamma=1** to have the optimizer codes adjust the **gamma** velocities of each star separately when fitting the radial velocity curves. Set **ifixgamma=2** to have a common value of **gamma** fit to *both* velocity curves.

- **iwriteeclipse**: Set **iwriteeclipse=1** to fit for eclipse times when **Nbody>2** and **itime=2**. Requires an additional configuration file described in Section **8.2.1** below.

- **ifracswitch**: Set **ifracswitch=1** to enable the optimizers to write the "luminosity ratios" and "disk fractions" to output files (called **ELCratio.???????**) for further analysis.

- **isuppressscreen**: Set **isuppressscreen=1** to suppress most of the screen output produced by many of the optimizer codes.

- **isuppressfile**: Set **isuppressfile=1** to suppress certain optimizer output files (**ELCdynparm.SUFFIX**, **demcmc_dynparm.SUFFIX**, and **hammer_dynparm.SUFFIX**, see Section **8.3**).

- **chi_thresh**: If **chi_thresh>0**, these optimizers won't update the generation.*, ELCparm.*, etc. files when $\chi^2 > \chi_{\rm small} + \chi_{\rm thresh}$: demcmcELC, geneticELC, markovELC, randomELC.

- **rmed**: If **rmed>0**, the optimizer codes will use the absolute deviation as the measure of fitness, rather than the normal χ^2.

- **jdum**: Random number seed for **demcmcELC**, **geneticELC**, **markovELC**, **nestedELC**, and **randomELC**. If you wish to use a starting seed other than the default value, set **jdum** to a negative integer.

Control flags and parameters for use with the dynamical integrator (`Nbody>=3`)

- **`Ndynwin`**: Number of time segments in the output of the dynamical integrator (see Section **7.4.4**) where the results are stored. If **`Ndynwin=0`** all results are stored. **`Ndynwin>0`** requires an additional input file, see Section **7.2.4**.

- **`Tref`**: Reference time (in days) for the dynamical integrator (see Section **7.4.4**).

- **`hh`**: Step size (in days) for the dynamical integrator (see Section **7.4.4**).

- **`iGR`**: Control flag to include effects of General Relativity and/or tides in the dynamical integrator (see Section **7.4.4**) Set **`iGR=0`** for pure Newtonian, **`iGR=1`** to include GR, **`iGR=2`** to include tides, and **`iGR=3`** to include both. If you set **`iGR=4`**, then the integration is the same as the **`iGR=3`** option, but with **`rk2`** forced to be the same as **`rk1`**.

- **`ibinbin`**: Set **`ibinbin=1`** to model a double binary system. Also set **`Nbody=4`**, see Section **7.4.4**.

- **`itconj`**: This controls what the time the parameter **`P1T0`** (see below) refers to: **`itconj=0`** for reference to the time of periastron passage of the third body, **`itconj=1`** for reference to the time of the third body transit of the *barycenter of the binary*, and **`itconj=2`** for reference to the time when the third body goes directly behind the *barycenter of the binary*. The most common use is **`itconj=1`**.

- **`ilog`**: If **`ilog=1`** the mass ratios for the extra bodies are assumed to be logarithms (base 10) of the actual mass ratios. Thus **`P1Q=5`** actually implies a mass ratio of 10^5 for the third body. If **`ilog=2`** the mass ratios for the extra bodies are assumed to be the actual masses in Earth masses. Thus if **`P2Q=36.0`** then the mass of body 4 is $36\,M_\oplus$.

- **`iEarthrad`**: Set **`iEarthrad=1`** to have planet radius ratios be actual radii in Earth radii. Not yet implemented.

- **`itran_occul`**: Set **`itran_occul=1`** to treat transits and occultations together (see Section **7.4.4**).

- **`itran_penal`**: Set **`itran_penal=1`** to impose a χ^2 penalty on models that produce transits of either star by one or more of the planets (see Section **7.4.4**).

- **`isec_penal`**: Set **`isec_penal=1`** to impose a χ^2 penalty on models that produce a secondary eclipse (see Section **7.4.4**).

- **`tran_penal`**: The χ^2 penalty for a transit when **`itran_penal=1`**.

- **`inform`**: Set **`inform=1`** to produce informational output from the dynamical integrator (see Section **7.4.4**). This works only for **`ELC`** and not any of the optimizer codes.

Features for Kepler data

- **`contam`**: This is the contamination to be applied to models of *Kepler* data, tag co. See Section **7.4.5**.

- **`Iseason`**: Set **`Iseason=1`** to use seasonal contamination parameters. See Section **7.4.5**.

- `contamS0`: Season 0 contamination, tag s0 (that is the number zero). See Section **7.4.5**.
- `contamS1`: Season 1 contamination, tag s1. See Section **7.4.5**.
- `contamS2`: Season 2 contamination, tag s2. See Section **7.4.5**.
- `contamS3`: Season 3 contamination, tag s3. See Section **7.4.5**.
- `Nseg`: Number of segments to fit for contamination. Use 0, not yet implemented.
- `ifastbin`: Set `ifastbin=1` for fast Kepler binning. See Section **7.4.5**.
- `lcbin`: Bin size for light curves (minutes). See Section **7.4.5**.
- `rvbin`: Bin size for radial velocity curves (minutes). See Section **7.4.5**.
- `NSC`: Number of short cadence intervals. See Section **7.2.3** for information on the input file and Section **7.4.5** for information on the algorithm.

Number of bodies

- `Nbody`: Number of bodies, from 2 to 10.

Binary orbit parameters

- `Period`: Orbital period in days, tag pe.
- `T0`: Time of periastron passage, tag T0 (that is the number zero).
- `Tconj`: Time of primary eclipse in days, tag Tc.
- `PbpTc`: $P + T_{\rm conj}$ for binary (days), tag bp. See Section **7.4.2**.
- `PbmTc`: $P - T_{\rm conj}$ for binary (days), tag bm. See Section **7.4.2**.
- `binqTc`: Slope in the $P, T_{\rm conj}$ for binary, tag bq. See Section **7.4.2**.
- `Tbinoff`: Offset in the $P, T_{\rm conj}$ for binary, tag bt. See Section **7.4.2**.
- `ecc`: Eccentricity of binary orbit, tag ec.
- `argper`: Argument of periastron of binary orbit (deg), tag ar.
- `ocose`: $e\cos\omega$ for binary orbit, tag oc. See Section **7.4.2**.
- `osine`: $e\sin\omega$ for binary orbit, tag os. See Section **7.4.2**.
- `sqcos`: $\sqrt{e}\cos\omega$ for binary orbit, tag bc. See Section **7.4.2**.
- `sqsin`: $\sqrt{e}\sin\omega$ for binary orbit, tag bs. See Section **7.4.2**.
- `finc`: Inclination of binary orbit (deg), tag in.
- `Omega_bin`: Nodal angle of binary orbit (deg), tag ob.

- **primK**: K-velocity of star 1 (km sec^{-1}), tag pk.
- **separ**: semi-major axis of binary orbit ($R_\odot$), tag se.
- **gamma1**: γ-velocity of binary orbit (km sec^{-1}), tag ga.
- **ecosw**: The phase difference between secondary and primary eclipses for an eccentric orbit, tag dp. The units are normalized phase units such that 2π**ecosw** is the phase difference in radians. If **ecosw>0** and **ecc>0**, the value of **argper** is computed and set. If you have been fitting for this parameter to get initial guesses for **ecc** and **argper**, don't forget to set **ecosw=0** if you wish to fit for **ecc** and **argper**.

Body 1 parameters

- **Nalph1**, **Nbet1**: Need two integers for this line, separated by spaces. These specify the number of grid elements used to integrate the intensity of star 1, if you are using Roche geometry (**Nterms=0**). The "**alpha**" direction is the latitude on the star, running from the "north pole" (**ialf=1**) to the "south pole" of the star (**ialf=Nalph1**) and represents equal steps in the angle. At each **alpha**, you loop over the "**beta**" direction which is the longitude on the star. The **beta** steps are equal in longitude, and there are a total of **4*Nbet1** points. Minimal values for these parameters are **Nalph1=10** and **Nbet1=6**.
- **Teff1**: $T_{\rm eff}$ for star 1 in K, tag T1 (the number one).
- **Tgrav1**: This is the "gravity darkening" exponent for star 1, tag g1 (the number one). The value of **Tgrav1** should be 0.25 for stars with radiative envelopes (von Zeipel 1924) and roughly 0.08 for stars with convective envelopes (Lucy 1967). If the flag **igrav** is larger than 0, then this input value might be ignored.
- **alb1**: The bolometric albedo for star 1 used in the computation of the reflection effect, tag l1 (the letter L and the number one). The albedo should be **alb1=1.0** for stars with a radiative envelope, and **alb1=0.5** for stars with a convective envelope.
- **omega1**: This parameter specifies the ratio of the rotational frequency of star 1 to the orbital frequency of the binary, tag o1 (the number one). If the star is tidally locked, then **omega1=1.0**.
- **fill1**: The Roche-lobe filling factor for star 1 when **Nterms=0**, tag f1 (the number one). **fill1=1.0** means the star exactly fills its Roche lobe. The filling factor must be larger than 0.0 and less than or equal to 1.0. This parameter may be ignored depending on other parameters, see Section **7.4.1**.
- **primrad**: Radius of star 1 ($R_\odot$), tag pr. See Section **7.4.1**.
- **frac1**: The fractional radius of star 1, R_1/a, where a is the orbital separation, tag q1 (the number one). See Section **7.4.1**.
- **radfill1**: If **radfill1>0**, it sets the Roche lobe filling factor of star 1 in terms of $R_{\rm eff}$, rather than in terms of the distance to the L_1 point.
- **primmass**: Mass of star 1 ($M_\odot$), tag pm. See Section **7.4.1**.

- **axis_I1**: This is the inclination of the rotational axis of star 1 if **ialign=0**, tag ai. See Hosokawa (1953) for helpful diagrams and equations.

- **axis_beta1**: This is the angle of the rotational axis of star 1 with respect to the orbit if **ialign=0**, tag ab. See Hosokawa (1953) for helpful diagrams and equations.

- **x1U**, **y1U**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 1, tag x1, y1 (the number one).

- **x1B**, **y1B**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 2, tag x2, y2.

- **x1V**, **y1V**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 3, tag x3, y3.

- **x1R**, **y1R**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 4, tag x4, y4.

- **x1I**, **y1I**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 5, tag x5, y5.

- **x1J**, **y1J**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 6, tag x6, y6.

- **x1H**, **y1H**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 7, tag x7, y7.

- **x1K**, **y1K**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 1, filter 8, tag x8, y8.

- **xbol1**, **ybol1**: Need two real numbers for this line, separated by spaces. Bolometric darkening coefficients for star 1, used in the computation of the reflection effect (Wilson 1990). Van Hamme (1993) has tabulated these coefficients for most temperatures and gravities. These bolometric limb darkening coefficients do not depend on the wavelength.

- **Tspot11**: The "temperature" factor of the first spot on star 1, tag b1 (the number one). Make this number larger than 1.0 for a hot spot and a positive number less than 1.0 for a cool (or dark) spot. The temperature of a surface element within a spot is the underlying temperature times the temperature factor of the spot, subject to the underlying scaling function determined by the **ispotprof** flag above. Set **Tspot11** to a negative number to have no spot.

- **latspot11**: The latitude of the spot for the first spot on star 1, tag b2. The units are *degrees*, with latitude of 0 degrees at the "north pole" (positive z-value), 90 degrees at the equator (orbital plane), and 180 degrees at the "south pole."

- **longspot11**: The longitude of the spot for the first spot on star 1, tag b3. The units are *degrees*, where the longitude is zero at the inner Lagrangian point and 180 degrees at the back end. A latitude of 90 degrees corresponds to the side of star 1 that is seen directly at orbital phase 90 degrees (circular orbit), or the side of star 2 that is seen directly at orbital phase 270 degrees (circular orbit).

- **radspot11**: The angular radius (in degrees) of the first spot on star 1, tag b4.

- **Tspot21**: The "temperature" factor of the second spot on star 1, tag b5. Make this number larger than 1.0 for a hot spot and a positive number less than 1.0 for a cool (or dark) spot. The temperature of a surface element within a spot is the underlying temperature times the temperature factor of the spot, subject to the underlying scaling function determined by the **ispotprof** flag above. Set **Tspot21** to a negative number to have no spot.

- **latspot21**: The latitude of the spot for the second spot on star 1, tag b6. The units are *degrees*, with latitude of 0 degrees at the "north pole" (positive z-value), 90 degrees at the equator (orbital plane), and 180 degrees at the "south pole."

- **longspot21**: The longitude of the spot for the second spot on star 1, tag b7. The units are *degrees*, where the longitude is zero at the inner Lagrangian point and 180 degrees at the back end. A latitude of 90 degrees corresponds to the side of star 1 that is seen directly at orbital phase 90 degrees (circular orbit), or the side of star 2 that is seen directly at orbital phase 270 degrees (circular orbit).

- **radspot21**: The angular radius (in degrees) of the second spot on star 1, tag b8.

- **beam1**: The Doppler boosting factor for star 1, tag e1 (the number one). If this parameter is zero, no correction to the light curve is applied. A typical value is **beam1=2.5** for an A-star, see van Kerkwijk et al. (2010).

- **rk1**: The value of the tidal apsidal constant for star 1, tag a1 (the number one). Needs to have **Nterms>0**, **Nbody>2**, and **iGR>1**, see Section **7.4.4**.

- **flux11**: Flux of star 1, filter 1, tag 11 (the number eleven). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 1 using **flux11**.

- **flux12**: Flux of star 1, filter 2, tag 12 (the number twelve). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 2 using **flux12**.

- **flux13**: Flux of star 1, filter 3, tag 13 (the number thirteen). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 3 using **flux13**.

- **flux14**: Flux of star 1, filter 4, tag 14 (the number fourteen). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 4 using **flux14**.

- **flux15**: Flux of star 1, filter 5, tag 15 (the number fifteen). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 5 using **flux15**.

- **flux16**: Flux of star 1, filter 6, tag 16 (the number sixteen). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux16**.

- **flux17**: Flux of star 1, filter 7, tag 17 (the number seventeen). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux17**.

- **flux18**: Flux of star 1, filter 8, tag 18 (the number eighteen). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux18**.

Body 2 parameters

- **Nalph2**, **Nbet2**: Need two integers for this line, separated by spaces. These specify the number of grid elements used to integrate the intensity of star 2, if you are using Roche geometry (**Nterms=0**). The "**alpha**" direction is the latitude on the star, running from the "north pole" (**ialf=1**) to the "south pole" of the star (**ialf=Nalph2**) and represents equal steps in the angle. At each **alpha**, you loop over the "**beta**" direction which is the longitude on the star. The **beta** steps are equal in longitude, and there are a total of **4*Nbet2** points. Minimal values for these parameters are **Nalph2=10** and **Nbet2=6**.

- **Teff2**: $T_{\rm eff}$ for star 2 in K, tag T2.

- **temprat**: The ratio of effective temperature of star 2 to the effective temperature of star 1, tag te. If **temprat>0**, then the value of **teff2** is computed from **temprat** and **Teff1**.

- **Tgrav2**: This is the "gravity darkening" exponent for star 2, tag g2. The value of **Tgrav2** should be 0.25 for stars with radiative envelopes (von Zeipel 1924) and roughly 0.08 for stars with convective envelopes (Lucy 1967). If the flag **igrav** is larger than 0, then this input value might be ignored.

- **alb2**: The bolometric albedo for star 2 used in the computation of the reflection effect, tag l2 (the letter L). The albedo should be **alb2=1.0** for stars with a radiative envelope, and **alb2=0.5** for stars with a convective envelope.

- **omega2**: This parameter specifies the ratio of the rotational frequency of star 2 to the orbital frequency of the binary, tag o2. If the star is tidally locked, then **omega2=1.0**.

- **fill2**: The Roche-lobe filling factor for star 2 when **Nterms=0**, tag f2. **fill2=1.0** means the star exactly fills its Roche lobe. The filling factor must be larger than 0.0 and less than or equal to 1.0. This parameter may be ignored depending on other parameters, see Section **7.4.1**.

- **fillsum**: This parameter gives **fill1+fill2**, tag sf. See Section **7.4.1**.

- **filldiff**: This parameter gives **fill1−fill2**, tag sd. See Section **7.4.1**.

- **secrad**: Radius of star 2 ($R_\odot$), tag sr. See Section **7.4.1**.

- **frac2**: The fractional radius of star 2, R_2/a, where a is the orbital separation, tag q2. See Section **7.4.1**.

- **ratrad**: **primrad/secrad**, ratio of star 1 radius to star 2 radius, tag ra. See Section **7.4.1**.

- **radsum**: **primrad+secrad**, sum of radii of star 1 and star 2 ($R_\odot$), tag rs. See Section **7.4.1**.

- **raddiff**: **primrad−secrad**, difference of radii of star 1 and star 2 ($R_\odot$), tag rd. See Section **7.4.1**.

- **fracsum**: **frac1+frac2**, sum of fractional radii, tag fs. See Section **7.4.1**.

- **fracdiff**: **frac1−frac2**, difference of fractional radii, tag fd. See Section **7.4.1**.

- **radfill2**: If **radfill2>0**, it sets the Roche lobe filling factor of star 2 in terms of $R_{\rm eff}$, rather than in terms of the distance to the L_1 point.

- **Q**: **secmass/primmass**, mass ratio of binary, tag ma. See Section **7.4.1**.

- **secmass**: Mass of star 2 ($M_\odot$), tag sm. See Section **7.4.1**.

- **masssum**: **primmass+secmass**, sum of the masses of star 1 and star 2 ($M_\odot$), tag ms. See Section **7.4.1**.

- **massdiff**: **primmass−secmass**, difference of the masses of star 1 and star 2 ($M_\odot$), tag md. See Section **7.4.1**.

- **axis_I2**: This is the inclination of the rotational axis of star 2 if **ialign=0**, tag bi. See Hosokawa (1953) for helpful diagrams and equations.

- **axis_beta2**: This is the angle of the rotational axis of star 2 with respect to the orbit if **ialign=0**, tag bb. See Hosokawa (1953) for helpful diagrams and equations.

- **x2U**, **y2U**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 1, tag z1, w1 (the number one).

- **x2B**, **y2B**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 2, tag z2, w2.

- **x2V**, **y2V**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 3, tag z3, w3.

- **x2R**, **y2R**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 4, tag z4, w4.

- **x2I**, **y2I**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 5, tag z5, w5.

- **x2J**, **y2J**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 6, tag z6, w6.

- **x2H**, **y2H**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 7, tag z7, w7.

- **x2K**, **y2K**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for star 2, filter 8, tag z8, w8.

- **xbol2**, **ybol2**: Need two real numbers for this line, separated by spaces. Bolometric darkening coefficients for star 2, used in the computation of the reflection effect (Wilson 1990). Van Hamme (1993) has tabulated these coefficients for most temperatures and gravities. These bolometric limb darkening coefficients do not depend on the wavelength.

- **Tspot12**: The "temperature" factor of the first spot on star 2, tag c1 (the number one). Make this number larger than 1.0 for a hot spot and a positive number less than 1.0 for a cool (or dark) spot. The temperature of a surface element within a spot is the underlying temperature times the temperature factor of the spot, subject to the underlying scaling function determined by the **ispotprof** flag above. Set **Tspot12** to a negative number to have no spot.

- **latspot12**: The latitude of the spot for the first spot on star 2, tag c2. The units are *degrees*, with latitude of 0 degrees at the "north pole" (positive z-value), 90 degrees at the equator (orbital plane), and 180 degrees at the "south pole."

- **longspot12**: The longitude of the spot for the first spot on star 2, tag c3. The units are *degrees*, where the longitude is zero at the inner Lagrangian point and 180 degrees at the back end. A latitude of 90 degrees corresponds to the side of star 1 that is seen directly at orbital phase 90 degrees (circular orbit), or the side of star 2 that is seen directly at orbital phase 270 degrees (circular orbit).

- **radspot12**: The angular radius (in degrees) of the first spot on star 2, tag c4.

- **Tspot22**: The "temperature" factor of the second spot on star 2, tag c5. Make this number larger than 1.0 for a hot spot and a positive number less than 1.0 for a cool (or dark) spot. The temperature of a surface element within a spot is the underlying temperature times the temperature factor of the spot, subject to the underlying scaling function determined by the **ispotprof** flag above. Set **Tspot22** to a negative number to have no spot.

- **latspot22**: The latitude of the spot for the second spot on star 2, tag c6. The units are *degrees*, with latitude of 0 degrees at the "north pole" (positive z-value), 90 degrees at the equator (orbital plane), and 180 degrees at the "south pole."

- **longspot22**: The longitude of the spot for the second spot on star 2, tag c7. The units are *degrees*, where the longitude is zero at the inner Lagrangian point and 180 degrees at the back end. A latitude of 90 degrees corresponds to the side of star 1 that is seen directly at orbital phase 90 degrees (circular orbit), or the side of star 2 that is seen directly at orbital phase 270 degrees (circular orbit).

- **radspot22**: The angular radius (in degrees) of the second spot on star 2, tag c8.

- **beam2**: The Doppler boosting factor for star 2, tag e2. If this parameter is zero, no correction to the light curve is applied. A typical value is **beam1=2.5** for an A-star, see van Kerkwijk et al. (2010).

- **rk2**: The value of the tidal apsidal constant for star 2, tag a2. Needs to have **Nterms>0**, **Nbody>2**, and **iGR>1**, see Section **7.4.4**.

- **flux21**: Flux of star 2, filter 1, tag 21 (the number twenty one). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 1 using **flux21**.

- **flux22**: Flux of star 2, filter 2, tag 22. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 2 using **flux22**.

- `flux23`: Flux of star 2, filter 3, tag 23. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 3 using `flux23`.
- `flux24`: Flux of star 2, filter 4, tag 24. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 4 using `flux24`.
- `flux25`: Flux of star 2, filter 5, tag 25. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 5 using `flux25`.
- `flux26`: Flux of star 2, filter 6, tag 26. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 6 using `flux26`.
- `flux27`: Flux of star 2, filter 7, tag 27. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 7 using `flux27`.
- `flux28`: Flux of star 2, filter 8, tag 28. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 8 using `flux28`.

Accretion disk parameters

- `idint`: Set `idint=1` to put the accretion disk around star 2.
- `Ntheta`: The number of grid points on the disk in the azmithual direction.
- `Nradius`: The number of grid points on the disk in the radial direction. You need a relatively large number of points in the radial direction to get the flux integration to converge reasonably well.
- `beta_rim`: This is the opening angle (in degrees) of the disk rim above the plane, tag be.
- `rinner`: This is the inner radius of the disk, in the same units as `fill2`, tag ri. Hence `rinner` should be equal to or larger than `fill2`. Currently the optimizer codes assume `fill2=rinner` for cases when star 2 is visible and there is a disk.
- `router`: This is the outer radius of the disk, expressed as a fraction of star 2's effective Roche lobe radius, tag ro. This parameter should be less than 1. If the inner disk radius is larger than the outer disk radius in the internal program units, the program will stop and complain.
- `tdisk`: This is the temperature of the *inner* disk, tag td. Note that on the old code (OB97) the temperature of the outer rim was specified. If `tdisk` is negative, the disk will be dark. It will still cause eclipses for high inclinations, but it will not contribute to the total light.

- `xi`: This is the power-law exponent on the temperature profile of the disk:

$$T(r) \propto T_d(r/r_{\rm inner})^{\xi},$$

tag xi. For a "steady-state" disk, `xi` should be -0.75. For disks that are irradiated by the central source, `xi` might be larger, perhaps on the order of -0.425 (i.e. the temperature profile on the disk is flatter and the outer parts of the disk are hotter than they would be otherwise).

- `Tspotdisk1`: The "temperature" factor of the first spot on the accretion disk, tag d1 (the number one). Make this number larger than 1.0 for a hot spot and a positive number less than 1.0 for a cool (or dark) spot. The temperature of a surface element within a spot is the underlying temperature times the temperature factor of the spot. Set `Tspotdisk1` to a negative number to have no spot.

- `diskspotaz1`: The azimuth of the first spot on the accretion disk, tag d2. For the disk, a spot at azimuth 90 degrees is seen directly ($\mu = 1$) at orbital phase 90 degrees (circular orbit), a spot at azimuth 200 degrees is seen directly at orbital phase 200 degrees, etc.

- `rspot_cut1`: Radial cut-off of the first spot on the accretion disk, tag d3. For the disk, the spot is always on the rim. How far down the disk is controlled by the cutoff radius. Set the cutoff radius to 0.9 to have the spot fill the outer 10% of the disk sector, etc.

- `angsizespot1`: Angular size of the first spot on the accretion disk (in degrees), tag d4.

- `Tspotdisk2`: The "temperature" factor of the second spot on the accretion disk, tag d5. Make this number larger than 1.0 for a hot spot and a positive number less than 1.0 for a cool (or dark) spot. The temperature of a surface element within a spot is the underlying temperature times the temperature factor of the spot. Set `Tspotdisk1` to a negative number to have no spot.

- `diskspotaz2`: The azimuth of the second spot on the accretion disk, tag d6. For the disk, a spot at azimuth 90 degrees is seen directly ($\mu = 1$) at orbital phase 90 degrees (circular orbit), a spot at azimuth 200 degrees is seen directly at orbital phase 200 degrees, etc.

- `rspot_cut2`: Radial cut-off of the second spot on the accretion disk, tag d7. For the disk, the spot is always on the rim. How far down the disk is controlled by the cutoff radius. Set the cutoff radius to 0.9 to have the spot fill the outer 10% of the disk sector, etc.

- `angsizespot2`: Angular size of the second spot on the accretion disk (in degrees), tag d8.

- `disk_frac_ref`: If `disk_frac_ref>0`, its value gives the reference phase for the disk fraction when using the optimizer codes (see below). Otherwise, the median of the disk fraction over the whole orbit is used.

Pulsar parameters

- `asini`: For use with pulsar binaries. This parameter is the projected semi-major axis of the pulsar's orbit *in light seconds*. If `Teff2 < 0` and `asini > 0`, then the orbital separation `separ` is computed from the values of the mass ratio `Q`, the inclination `finc`, and `asini`:

$$a = \frac{\texttt{asini}(1+Q)c}{\sin i}.$$

- **asini_error**: In the "millisecond pulsar mode", the parameter **asini** is the projected semi-major axis of the pulsar in light seconds. **asini_error** is the uncertainty in **asini**, and if nonzero will be used in some of the optimizer codes. Values of **asini** will be drawn from a Gaussian distribution with a standard deviation of **asini_error**.

If **Nbody=2**, then this concludes **ELC.inp**. Otherwise, parameters related to the third body follow.

Body 3 orbital parameters

- **P1Tconj**: Time of barycentric inferior conjunction of body 3, tag tj.
- **P1period**: Orbital period of body 3 in days, tag tt.
- **P1pTc**: **P1period**+**P1Tconj** ($P + T_{\rm conj}$ for body 3) in days, tag 1p (the number one). See Section **7.4.2**.
- **P1mTc**: **P1period**−**P1Tconj** ($P - T_{\rm conj}$ for body 3) in days, tag 1m (the number one). See Section **7.4.2**.
- **P1qTc**: Slope parameter for the $P - T_{\rm conj}$ plane for body 3, tag 1q (the number one). See Section **7.4.2**.
- **T1off**: Offset parameter for the $P - T_{\rm conj}$ plane for body 3, tag 1t (the number one). See Section **7.4.2**.
- **P1T0**: Time of periastron passage of body 3, tag tu. See above for the note on **itconj**.
- **P1ecos**: $e\cos\omega$ for body 3, tag tv. See Section **7.4.2**.
- **P1esin**: $e\sin\omega$ for body 3, tag tw. See Section **7.4.2**.
- **P1sqecos**: $\sqrt{e}\cos\omega$ for body 3, tag 1c. See Section **7.4.2**.
- **P1sqesin**: $\sqrt{e}\sin\omega$ for body 3, tag 1s. See Section **7.4.2**.
- **P1incl**: Inclination of body 3 orbit in degrees, tag tx.
- **P1Omega**: Nodal angle of body 3 orbit in degrees, tag ty.
- **angsum1**: Sum of inclination and nodal angle of body 3 orbit in degrees, tag 1a (the number one). See Section **7.4.2**.
- **angdiff1**: Difference of inclination and nodal angle of body 3 orbit in degrees, tag 1d (the number one). See Section **7.4.2**.

Body 3 parameters

- **Nalph3**, **Nbet3**: Need two integers for this line, separated by spaces. These specify the number of grid elements used to integrate the intensity of star 3, if you are using Roche geometry (**Nterms=0**).
- **Teff3**: $T_{\rm eff}$ for star 3 in K, tag T3.

- `g3`: Logarithm of the surface gravity of star 3 in c.g.s. units, tag g3.
- `P1ratrad`: Ratio of radius of star 1 to radius of body 3, tag tb.
- `P1Q`: Ratio of $(M_1 + M_2)/M_3$, tag tz. This parameter can have different meanings depending on the flag `ilog` above.
- `omega3`: This parameter specifies the ratio of the rotational frequency of star 3 to the orbital frequency, tag o3. This is needed when `Nterms>0` and `iGR>1` (e.g. when corrections for tides are computed in the dynamical integrator).
- `axis_I3`: This is the inclination of the rotational axis of body 3 if `ialign=0`, tag ci. See Hosokawa (1953) for helpful diagrams and equations.
- `axis_beta3`: This is the angle of the rotational axis of body 3 with respect to the orbit if `ialign=0`, tag cb. See Hosokawa (1953) for helpful diagrams and equations.
- `x3U`, `y3U`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 1, tag m1, n1 (the number one).
- `x3B`, `y3B`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 2, tag m2, n2.
- `x3V`, `y3V`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 3, tag m3, n3.
- `x3R`, `y3R`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 4, tag m4, n4.
- `x3I`, `y3I`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 5, tag m5, n5.
- `x3J`, `y3J`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 6, tag m6, n6.
- `x3H`, `y3H`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 7, tag m7, n7.
- `x3K`, `y3K`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 3, filter 8, tag m8, n8.
- `rk3`: The value of the tidal apsidal constant for star 3, tag a3. Needs to have `Nterms>0`, `Nbody>2`, and `iGR>1`, see Section **7.4.4**.
- `flux31`: Flux of body 3, filter 1, tag 31 (the number thirty one). Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 1 using `flux31`.
- `flux32`: Flux of body 3, filter 2, tag 32. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 2 using `flux32`.
- `flux33`: Flux of body 3, filter 3, tag 33. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 3 using `flux33`.

- **flux34**: Flux of body 3, filter 4, tag 34. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 4 using **flux34**.

- **flux35**: Flux of body 3, filter 5, tag 35. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 5 using **flux35**.

- **flux36**: Flux of body 3, filter 6, tag 36. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux36**.

- **flux37**: Flux of body 3, filter 7, tag 37. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux37**.

- **flux38**: Flux of body 3, filter 8, tag 38. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux38**.

If **Nbody=3**, then this concludes **ELC.inp**. Otherwise, parameters related to the fourth body follow.

Body 4 orbital parameters

- **P2Tconj**: Time of barycentric inferior conjunction of body 4, tag uj.

- **P2period**: Orbital period of body 4 in days, tag ut.

- **P2pTc**: **P2period**+**P2Tconj** ($P + T_{\rm conj}$ for body 4) in days, tag 2p. See Section **7.4.2**.

- **P2mTc**: **P2period**−**P2Tconj** ($P - T_{\rm conj}$ for body 4) in days, tag 2m. See Section **7.4.2**.

- **P2qTc**: Slope parameter for the $P - T_{\rm conj}$ plane for body 4, tag 2q. See Section **7.4.2**.

- **T2off**: Offset parameter for the $P - T_{\rm conj}$ plane for body 4, tag 2t. See Section **7.4.2**.

- **P2T0**: Time of periastron passage of body 4, tag uu. See above for the note on **itconj**.

- **P2ecos**: $e \cos\omega$ for body 4, tag uv. See Section **7.4.2**.

- **P2esin**: $e \sin\omega$ for body 4, tag uw. See Section **7.4.2**.

- **P2sqecos**: $\sqrt{e} \cos\omega$ for body 4, tag 2c. See Section **7.4.2**.

- **P2sqesin**: $\sqrt{e} \sin\omega$ for body 4, tag 2s. See Section **7.4.2**.

- **P2incl**: Inclination of body 4 orbit in degrees, tag ux.

- **P2Omega**: Nodal angle of body 4 orbit in degrees, tag uy.

- **angsum2**: Sum of inclination and nodal angle of body 4 orbit in degrees, tag 2a. See Section **7.4.2**.

- **angdiff2**: Difference of inclination and nodal angle of body 4 orbit in degrees, tag 2d. See Section **7.4.2**.

Body 4 parameters

- **`Teff4`**: $T_{\rm eff}$ for star 4 in K, tag T4.
- **`g4`**: Logarithm of the surface gravity of star 4 in c.g.s. units, tag g4.
- **`P2ratrad`**: Ratio of radius of star 1 to radius of body 4, tag ub.
- **`P2Q`**: Ratio of $(M_1+M_2)/M_4$, tag uz. This parameter can have different meanings depending on the flag **`ilog`** above.
- **`omega4`**: This parameter specifies the ratio of the rotational frequency of star 4 to the orbital frequency, tag o4. This is needed when **`Nterms>0`** and **`iGR>1`** (e.g. when corrections for tides are computed in the dynamical integrator).
- **`axis_I4`**: This is the inclination of the rotational axis of body 4 if **`ialign=0`**, tag di. See Hosokawa (1953) for helpful diagrams and equations.
- **`axis_beta4`**: This is the angle of the rotational axis of body 4 with respect to the orbit if **`ialign=0`**, tag db. See Hosokawa (1953) for helpful diagrams and equations.
- **`x4U`**, **`y4U`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 1, tag i1, j1 (the number one).
- **`x4B`**, **`y4B`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 2, tag i2, j2.
- **`x4V`**, **`y4V`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 3, tag i3, j3.
- **`x4R`**, **`y4R`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 4, tag i4, j4.
- **`x4I`**, **`y4I`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 5, tag i5, j5.
- **`x4J`**, **`y4J`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 6, tag i6, j6.
- **`x4H`**, **`y4H`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 7, tag i7, j7.
- **`x4K`**, **`y4K`**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 4, filter 8, tag i8, j8.
- **`rk4`**: The value of the tidal apsidal constant for star 4, tag a4. Needs to have **`Nterms>0`**, **`Nbody>2`**, and **`iGR>1`**, see Section **7.4.4**.
- **`flux41`**: Flux of body 4, filter 1, tag 41 (the number forty one). Set **`iflux=0`** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **`iflux=1`** to specify the baseline flux for filter 1 using **`flux41`**.
- **`flux42`**: Flux of body 4, filter 2, tag 42. Set **`iflux=0`** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **`iflux=1`** to specify the baseline flux for filter 2 using **`flux42`**.

- **flux43**: Flux of body 4, filter 3, tag 43. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 3 using **flux43**.

- **flux44**: Flux of body 4, filter 4, tag 44. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 4 using **flux44**.

- **flux45**: Flux of body 4, filter 5, tag 45. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 5 using **flux45**.

- **flux46**: Flux of body 4, filter 6, tag 46. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux46**.

- **flux47**: Flux of body 4, filter 7, tag 47. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux47**.

- **flux48**: Flux of body 4, filter 8, tag 48. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux48**.

Binary+binary stellar parameters

- **bin2masssum**: $M_3 + M_4$ ($M_\odot$) in binary+binary mode, tag 91 (the number ninety one). See Section **7.4.4**.

- **bin2massdiff**: $M_3 - M_4$ ($M_\odot$) in binary+binary mode, tag 92. See Section **7.4.4**.

- **bin2Q**: M_4/M_3 in binary+binary mode, tag 93. See Section **7.4.4**.

- **bin2radsum**: $R_3 + R_4$ ($R_\odot$) in binary+binary mode, tag 94. See Section **7.4.4**.

- **bin2raddiff**: $R_3 - R_4$ ($R_\odot$) in binary+binary mode, tag 95. See Section **7.4.4**.

- **bin2ratrad**: R_3/R_4 in binary+binary mode, tag 96. See Section **7.4.4**.

- **bin2M3**: M_3 ($M_\odot$) in binary+binary mode, tag 97. See Section **7.4.4**.

- **bin2M4**: M_4 ($M_\odot$) in binary+binary mode, tag 98. See Section **7.4.4**.

- **bin2R3**: R_3 ($R_\odot$) in binary+binary mode, tag 99. See Section **7.4.4**.

- **bin2R4**: R_4 ($R_\odot$) in binary+binary mode, tag 90 (the number ninety). See Section **7.4.4**.

If **Nbody=4**, then this concludes **ELC.inp**. Otherwise, parameters related to body 5 follow.

Body 5 orbital parameters

- **P3Tconj**: Time of barycentric inferior conjunction of body 5, tag vj.
- **P3period**: Orbital period of body 5 in days, tag vt.
- **P3pTc**: **P3period** + **P3Tconj** ($P + T_{\rm conj}$ for body 5) in days, tag 3p. See Section **7.4.2**.
- **P3mTc**: **P3period** − **P3Tconj** ($P - T_{\rm conj}$ for body 5) in days, tag 3m. See Section **7.4.2**.
- **P3qTc**: Slope parameter for the $P - T_{\rm conj}$ plane for body 5, tag 3q. See Section **7.4.2**.
- **T3off**: Offset parameter for the $P - T_{\rm conj}$ plane for body 5, tag 3t. See Section **7.4.2**.
- **P3T0**: Time of periastron passage of body 5, tag vu. See above for the note on **itconj**.
- **P3ecos**: $e\cos\omega$ for body 5, tag vv. See Section **7.4.2**.
- **P3esin**: $e\sin\omega$ for body 5, tag vw. See Section **7.4.2**.
- **P3sqecos**: $\sqrt{e}\cos\omega$ for body 5, tag 3c. See Section **7.4.2**.
- **P3sqesin**: $\sqrt{e}\sin\omega$ for body 5, tag 3s. See Section **7.4.2**.
- **P3incl**: Inclination of body 5 orbit in degrees, tag vx.
- **P3Omega**: Nodal angle of body 5 orbit in degrees, tag vy.
- **angsum3**: Sum of inclination and nodal angle of body 5 orbit in degrees, tag 3a. See Section **7.4.2**.
- **angdiff3**: Difference of inclination and nodal angle of body 5 orbit in degrees, tag 3d. See Section **7.4.2**.

Body 5 parameters

- **Teff5**: $T_{\rm eff}$ for star 5 in K, tag T5.
- **g5**: Logarithm of the surface gravity of star 5 in c.g.s. units, tag g5.
- **P3ratrad**: Ratio of radius of star 1 to radius of body 5, tag vb.
- **P3Q**: Ratio of $(M_1 + M_2)/M_5$, tag vz. This parameter can have different meanings depending on the flag **ilog** above.
- **omega5**: This parameter specifies the ratio of the rotational frequency of star 5 to the orbital frequency, tag o5. This is needed when **Nterms>0** and **iGR>1** (e.g. when corrections for tides are computed in the dynamical integrator).
- **axis_I5**: This is the inclination of the rotational axis of body 5 if **ialign=0**, tag .... See Hosokawa (1953) for helpful diagrams and equations.
- **axis_beta5**: This is the angle of the rotational axis of body 5 with respect to the orbit if **ialign=0**, tag .... See Hosokawa (1953) for helpful diagrams and equations.

- **x5U**, **y5U**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 1, tag k1, p1 (the number one).
- **x5B**, **y5B**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 2, tag k2, p2.
- **x5V**, **y5V**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 3, tag k3, p3.
- **x5R**, **y5R**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 4, tag k4, p4.
- **x5I**, **y5I**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 5, tag k5, p5.
- **x5J**, **y5J**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 6, tag k6, p6.
- **x5H**, **y5H**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 7, tag k7, p7.
- **x5K**, **y5K**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 5, filter 8, tag k8, p8.
- **rk5**: The value of the tidal apsidal constant for star 5, tag a5. Needs to have **Nterms>0**, **Nbody>2**, and **iGR>1**, see Section **7.4.4**.
- **flux51**: Flux of body 5, filter 1, tag 51 (the number fifty one). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 1 using **flux51**.
- **flux52**: Flux of body 5, filter 2, tag 52. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 2 using **flux52**.
- **flux53**: Flux of body 5, filter 3, tag 53. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 3 using **flux53**.
- **flux54**: Flux of body 5, filter 4, tag 54. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 4 using **flux54**.
- **flux55**: Flux of body 5, filter 5, tag 55. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 5 using **flux55**.
- **flux56**: Flux of body 5, filter 6, tag 56. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux56**.
- **flux57**: Flux of body 5, filter 7, tag 57. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux57**.

- **flux58**: Flux of body 5, filter 8, tag 58. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux58**.

If **Nbody=5**, then this concludes **ELC.inp**. Otherwise, parameters related to body 6 follow.

Body 6 orbital parameters

- **P4Tconj**: Time of barycentric inferior conjunction of body 6, tag wj.
- **P4period**: Orbital period of body 6 in days, tag wt.
- **P4pTc**: **P4period**+**P4Tconj** ($P + T_{\rm conj}$ for body 6) in days, tag 4p. See Section **7.4.2**.
- **P4mTc**: **P4period**−**P4Tconj** ($P - T_{\rm conj}$ for body 6) in days, tag 4m. See Section **7.4.2**.
- **P4qTc**: Slope parameter for the $P - T_{\rm conj}$ plane for body 6, tag 4q. See Section **7.4.2**.
- **T4off**: Offset parameter for the $P - T_{\rm conj}$ plane for body 6, tag 4t. See Section **7.4.2**.
- **P4T0**: Time of periastron passage of body 6, tag wu. See above for the note on **itconj**.
- **P4ecos**: $e\cos\omega$ for body 6, tag wv. See Section **7.4.2**.
- **P4esin**: $e\sin\omega$ for body 6, tag ww. See Section **7.4.2**.
- **P4sqecos**: $\sqrt{e}\cos\omega$ for body 6, tag 4c. See Section **7.4.2**.
- **P4sqesin**: $\sqrt{e}\sin\omega$ for body 6, tag 4s. See Section **7.4.2**.
- **P4incl**: Inclination of body 6 orbit in degrees, tag wx.
- **P4Omega**: Nodal angle of body 6 orbit in degrees, tag wy.
- **angsum4**: Sum of inclination and nodal angle of body 6 orbit in degrees, tag 4a. See Section **7.4.2**.
- **angdiff4**: Difference of inclination and nodal angle of body 6 orbit in degrees, tag 4d. See Section **7.4.2**.

Body 6 parameters

- **Teff6**: $T_{\rm eff}$ for star 6 in K, tag T6.
- **g6**: Logarithm of the surface gravity of star 6 in c.g.s. units, tag g6.
- **P4ratrad**: Ratio of radius of star 1 to radius of body 6, tag wb.
- **P4Q**: Ratio of $(M_1+M_2)/M_6$, tag wz. This parameter can have different meanings depending on the flag **ilog** above.
- **omega6**: This parameter specifies the ratio of the rotational frequency of star 6 to the orbital frequency, tag o6. This is needed when **Nterms>0** and **iGR>1** (e.g. when corrections for tides are computed in the dynamical integrator).

- `axis_I6`: This is the inclination of the rotational axis of body 6 if `ialign=0`, tag .... See Hosokawa (1953) for helpful diagrams and equations.

- `axis_beta6`: This is the angle of the rotational axis of body 6 with respect to the orbit if `ialign=0`, tag .... See Hosokawa (1953) for helpful diagrams and equations.

- `x6U`, `y6U`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 1, tag ..., ...

- `x6B`, `y6B`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 2, tag ..., ...

- `x6V`, `y6V`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 3, tag ..., ...

- `x6R`, `y6R`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 4, tag ..., ...

- `x6I`, `y6I`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 5, tag ..., ...

- `x6J`, `y6J`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 6, tag ..., ...

- `x6H`, `y6H`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 7, tag ..., ...

- `x6K`, `y6K`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 6, filter 8, tag ..., ...

- `rk6`: The value of the tidal apsidal constant for star 6, tag a6. Needs to have `Nterms>0`, `Nbody>2`, and `iGR>1`, see Section **7.4.4**.

- `flux61`: Flux of body 6, filter 1, tag 61 (the number sixty one). Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 1 using `flux61`.

- `flux62`: Flux of body 6, filter 2, tag 62. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 2 using `flux62`.

- `flux63`: Flux of body 6, filter 3, tag 63. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 3 using `flux63`.

- `flux64`: Flux of body 6, filter 4, tag 64. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 4 using `flux64`.

- `flux65`: Flux of body 6, filter 5, tag 65. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 5 using `flux65`.

- **flux66**: Flux of body 6, filter 6, tag 66. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux66**.
- **flux67**: Flux of body 6, filter 7, tag 67. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux67**.
- **flux68**: Flux of body 6, filter 8, tag 68. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux68**.

If **Nbody=6**, then this concludes **ELC.inp**. Otherwise, parameters related to body 7 follow.

Body 7 orbital parameters

- **P5Tconj**: Time of barycentric inferior conjunction of body 7, tag xj.
- **P5period**: Orbital period of body 7 in days, tag xt.
- **P5pTc**: **P5period**+**P5Tconj** ($P + T_{\rm conj}$ for body 7) in days, tag 5p. See Section **7.4.2**.
- **P5mTc**: **P5period**−**P5Tconj** ($P - T_{\rm conj}$ for body 7) in days, tag 5m. See Section **7.4.2**.
- **P5qTc**: Slope parameter for the $P - T_{\rm conj}$ plane for body 7, tag 5q. See Section **7.4.2**.
- **T5off**: Offset parameter for the $P - T_{\rm conj}$ plane for body 7, tag 5t. See Section **7.4.2**.
- **P5T0**: Time of periastron passage of body 7, tag xu. See above for the note on **itconj**.
- **P5ecos**: $e\cos\omega$ for body 7, tag xv. See Section **7.4.2**.
- **P5esin**: $e\sin\omega$ for body 7, tag xw. See Section **7.4.2**.
- **P5sqecos**: $\sqrt{e}\cos\omega$ for body 7, tag 5c. See Section **7.4.2**.
- **P5sqesin**: $\sqrt{e}\sin\omega$ for body 7, tag 5s. See Section **7.4.2**.
- **P5incl**: Inclination of body 7 orbit in degrees, tag xx.
- **P5Omega**: Nodal angle of body 7 orbit in degrees, tag xy.
- **angsum5**: Sum of inclination and nodal angle of body 7 orbit in degrees, tag 5a. See Section **7.4.2**.
- **angdiff5**: Difference of inclination and nodal angle of body 7 orbit in degrees, tag 5d. See Section **7.4.2**.

Body 7 parameters

- `Teff7`: $T_{\rm eff}$ for star 7 in K, tag T7.
- `g7`: Logarithm of the surface gravity of star 7 in c.g.s. units, tag g7.
- `P5ratrad`: Ratio of radius of star 1 to radius of body 7, tag xb.
- `P5Q`: Ratio of $(M_1+M_2)/M_7$, tag xz. This parameter can have different meanings depending on the flag `ilog` above.
- `omega7`: This parameter specifies the ratio of the rotational frequency of star 7 to the orbital frequency, tag o7. This is needed when `Nterms>0` and `iGR>1` (e.g. when corrections for tides are computed in the dynamical integrator).
- `axis_I7`: This is the inclination of the rotational axis of body 7 if `ialign=0`, tag .... See Hosokawa (1953) for helpful diagrams and equations.
- `axis_beta7`: This is the angle of the rotational axis of body 7 with respect to the orbit if `ialign=0`, tag .... See Hosokawa (1953) for helpful diagrams and equations.
- `x7U`, `y7U`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 1, tag ..., ...
- `x7B`, `y7B`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 2, tag ..., ...
- `x7V`, `y7V`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 3, tag ..., ...
- `x7R`, `y7R`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 4, tag ..., ...
- `x7I`, `y7I`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 5, tag ..., ...
- `x7J`, `y7J`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 6, tag ..., ...
- `x7H`, `y7H`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 7, tag ..., ...
- `x7K`, `y7K`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 7, filter 8, tag ..., ...
- `rk7`: The value of the tidal apsidal constant for star 7, tag a7. Needs to have `Nterms>0`, `Nbody>2`, and `iGR>1`, see Section **7.4.4**.
- `flux71`: Flux of body 7, filter 1, tag 71 (the number seventy one). Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 1 using `flux71`.
- `flux72`: Flux of body 7, filter 2, tag 72. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 2 using `flux72`.

- `flux73`: Flux of body 7, filter 3, tag 73. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 3 using `flux73`.
- `flux74`: Flux of body 7, filter 4, tag 74. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 4 using `flux74`.
- `flux75`: Flux of body 7, filter 5, tag 75. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 5 using `flux75`.
- `flux76`: Flux of body 7, filter 6, tag 76. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 6 using `flux76`.
- `flux77`: Flux of body 7, filter 7, tag 77. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 7 using `flux77`.
- `flux78`: Flux of body 7, filter 8, tag 78. Set `iflux=0` to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set `iflux=1` to specify the baseline flux for filter 8 using `flux78`.

If `Nbody=7`, then this concludes `ELC.inp`. Otherwise, parameters related to body 8 follow.

Body 8 orbital parameters

- `P6Tconj`: Time of barycentric inferior conjunction of body 8, tag sj.
- `P6period`: Orbital period of body 8 in days, tag st.
- `P6pTc`: `P6period`+`P6Tconj` ($P + T_{\rm conj}$ for body 8) in days, tag 6p. See Section **7.4.2**.
- `P6mTc`: `P6period`−`P6Tconj` ($P - T_{\rm conj}$ for body 8) in days, tag 6m. See Section **7.4.2**.
- `P6qTc`: Slope parameter for the $P - T_{\rm conj}$ plane for body 8, tag 6q. See Section **7.4.2**.
- `T6off`: Offset parameter for the $P - T_{\rm conj}$ plane for body 8, tag 6t. See Section **7.4.2**.
- `P6T0`: Time of periastron passage of body 8, tag su. See above for the note on `itconj`.
- `P6ecos`: $e\cos\omega$ for body 8, tag sv. See Section **7.4.2**.
- `P6esin`: $e\sin\omega$ for body 8, tag sw. See Section **7.4.2**.
- `P6sqecos`: $\sqrt{e}\cos\omega$ for body 8, tag 6c. See Section **7.4.2**.
- `P6sqesin`: $\sqrt{e}\sin\omega$ for body 8, tag 6s. See Section **7.4.2**.
- `P6incl`: Inclination of body 8 orbit in degrees, tag sx.
- `P6Omega`: Nodal angle of body 8 orbit in degrees, tag sy.

- **angsum6**: Sum of inclination and nodal angle of body 8 orbit in degrees, tag 6a. See Section **7.4.2**.

- **angdiff6**: Difference of inclination and nodal angle of body 8 orbit in degrees, tag 6d. See Section **7.4.2**.

Body 8 parameters

- **Teff8**: $T_{\rm eff}$ for star 8 in K, tag T8.

- **g8**: Logarithm of the surface gravity of star 8 in c.g.s. units, tag g8.

- **P6ratrad**: Ratio of radius of star 1 to radius of body 8, tag sb.

- **P6Q**: Ratio of $(M_1+M_2)/M_8$, tag sz. This parameter can have different meanings depending on the flag **ilog** above.

- **omega8**: This parameter specifies the ratio of the rotational frequency of star 8 to the orbital frequency, tag o8. This is needed when **Nterms>0** and **iGR>1** (e.g. when corrections for tides are computed in the dynamical integrator).

- **axis_I8**: This is the inclination of the rotational axis of body 8 if **ialign=0**, tag .... See Hosokawa (1953) for helpful diagrams and equations.

- **axis_beta8**: This is the angle of the rotational axis of body 8 with respect to the orbit if **ialign=0**, tag .... See Hosokawa (1953) for helpful diagrams and equations.

- **x8U**, **y8U**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 1, tag ..., ...

- **x8B**, **y8B**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 2, tag ..., ...

- **x8V**, **y8V**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 3, tag ..., ...

- **x8R**, **y8R**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 4, tag ..., ...

- **x8I**, **y8I**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 5, tag ..., ...

- **x8J**, **y8J**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 6, tag ..., ...

- **x8H**, **y8H**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 7, tag ..., ...

- **x8K**, **y8K**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 8, filter 8, tag ..., ...

- **rk8**: The value of the tidal apsidal constant for star 8, tag a8. Needs to have **Nterms>0**, **Nbody>2**, and **iGR>1**, see Section **7.4.4**.

- **flux81**: Flux of body 8, filter 1, tag 81 (the number eighty one). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 1 using **flux81**.
- **flux82**: Flux of body 8, filter 2, tag 82. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 2 using **flux82**.
- **flux83**: Flux of body 8, filter 3, tag 83. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 3 using **flux83**.
- **flux84**: Flux of body 8, filter 4, tag 84. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 4 using **flux84**.
- **flux85**: Flux of body 8, filter 5, tag 85. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 5 using **flux85**.
- **flux86**: Flux of body 8, filter 6, tag 86. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux86**.
- **flux87**: Flux of body 8, filter 7, tag 87. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux87**.
- **flux88**: Flux of body 8, filter 8, tag 88. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux88**.

If **Nbody=8**, then this concludes **ELC.inp**. Otherwise, parameters related to body 9 follow.

Body 9 orbital parameters

- **P7Tconj**: Time of barycentric inferior conjunction of body 9, tag hj.
- **P7period**: Orbital period of body 8 in days, tag ht.
- **P7pTc**: **P7period**+**P7Tconj** ($P + T_{\rm conj}$ for body 9) in days, tag 7p. See Section **7.4.2**.
- **P7mTc**: **P7period**−**P7Tconj** ($P - T_{\rm conj}$ for body 9) in days, tag 7m. See Section **7.4.2**.
- **P7qTc**: Slope parameter for the $P - T_{\rm conj}$ plane for body 9, tag 7q. See Section **7.4.2**.
- **T7off**: Offset parameter for the $P - T_{\rm conj}$ plane for body 9, tag 7t. See Section **7.4.2**.
- **P7T0**: Time of periastron passage of body 9, tag hu. See above for the note on **itconj**.
- **P7ecos**: $e\cos\omega$ for body 9, tag hv. See Section **7.4.2**.
- **P7esin**: $e\sin\omega$ for body 9, tag hw. See Section **7.4.2**.

- `P7sqecos`: $\sqrt{e}\cos\omega$ for body 9, tag 7c. See Section **7.4.2**.
- `P7sqesin`: $\sqrt{e}\sin\omega$ for body 9, tag 7s. See Section **7.4.2**.
- `P7incl`: Inclination of body 9 orbit in degrees, tag hx.
- `P7Omega`: Nodal angle of body 9 orbit in degrees, tag hy.
- `angsum7`: Sum of inclination and nodal angle of body 9 orbit in degrees, tag 7a. See Section **7.4.2**.
- `angdiff7`: Difference of inclination and nodal angle of body 9 orbit in degrees, tag 7d. See Section **7.4.2**.

Body 9 parameters

- `Teff9`: $T_{\rm eff}$ for star 9 in K, tag T9.
- `g9`: Logarithm of the surface gravity of star 9 in c.g.s. units, tag g9.
- `P7ratrad`: Ratio of radius of star 1 to radius of body 9, tag hb.
- `P7Q`: Ratio of $(M_1+M_2)/M_9$, tag hz. This parameter can have different meanings depending on the flag `ilog` above.
- `omega9`: This parameter specifies the ratio of the rotational frequency of star 9 to the orbital frequency, tag o9. This is needed when `Nterms>0` and `iGR>1` (e.g. when corrections for tides are computed in the dynamical integrator).
- `axis_I9`: This is the inclination of the rotational axis of body 8 if `ialign=0`, tag .... See Hosokawa (1953) for helpful diagrams and equations.
- `axis_beta9`: This is the angle of the rotational axis of body 8 with respect to the orbit if `ialign=0`, tag .... See Hosokawa (1953) for helpful diagrams and equations.
- `x9U`, `y9U`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 1, tag ..., ...
- `x9B`, `y9B`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 2, tag ..., ...
- `x9V`, `y9V`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 3, tag ..., ...
- `x9R`, `y9R`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 4, tag ..., ...
- `x9I`, `y9I`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 5, tag ..., ...
- `x9J`, `y9J`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 6, tag ..., ...

- **x9H**, **y9H**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 7, tag ..., ...

- **x9K**, **y9K**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 9, filter 8, tag ..., ...

- **rk9**: The value of the tidal apsidal constant for star 9, tag a9. Needs to have **Nterms>0**, **Nbody>2**, and **iGR>1**, see Section **7.4.4**.

- **flux91**: Flux of body 9, filter 1, tag 91 (the number ninety one). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 1 using **flux91**.

- **flux92**: Flux of body 9, filter 2, tag 92. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 2 using **flux92**.

- **flux93**: Flux of body 9, filter 3, tag 93. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 3 using **flux93**.

- **flux94**: Flux of body 9, filter 4, tag 94. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 4 using **flux94**.

- **flux95**: Flux of body 9, filter 5, tag 95. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 5 using **flux95**.

- **flux96**: Flux of body 9, filter 6, tag 96. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux96**.

- **flux97**: Flux of body 9, filter 7, tag 97. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux97**.

- **flux98**: Flux of body 9, filter 8, tag 98. Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux98**.

If **Nbody=9**, then this concludes **ELC.inp**. Otherwise, parameters related to body 10 follow.

Body 10 orbital parameters

- **P8Tconj**: Time of barycentric inferior conjunction of body 10, tag kj.

- **P8period**: Orbital period of body 10 in days, tag kt.

- **P8pTc**: **P8period** + **P8Tconj** ($P + T_{\rm conj}$ for body 10) in days, tag 8p. See Section **7.4.2**.

- **P8mTc**: **P8period** − **P8Tconj** ($P - T_{\rm conj}$ for body 10) in days, tag 8m. See Section **7.4.2**.

- `P8qTc`: Slope parameter for the $P - T_{\rm conj}$ plane for body 10, tag 8q. See Section **7.4.2**.
- `T8off`: Offset parameter for the $P - T_{\rm conj}$ plane for body 10, tag 8t. See Section **7.4.2**.
- `P8T0`: Time of periastron passage of body 10, tag ku. See above for the note on **`itconj`**.
- `P8ecos`: $e\cos\omega$ for body 10, tag kv. See Section **7.4.2**.
- `P8esin`: $e\sin\omega$ for body 10, tag kw. See Section **7.4.2**.
- `P8sqecos`: $\sqrt{e}\cos\omega$ for body 10, tag 8c. See Section **7.4.2**.
- `P8sqesin`: $\sqrt{e}\sin\omega$ for body 10, tag 8s. See Section **7.4.2**.
- `P8incl`: Inclination of body 10 orbit in degrees, tag kx.
- `P8Omega`: Nodal angle of body 10 orbit in degrees, tag ky.
- `angsum8`: Sum of inclination and nodal angle of body 10 orbit in degrees, tag 8a. See Section **7.4.2**.
- `angdiff8`: Difference of inclination and nodal angle of body 10 orbit in degrees, tag 8d. See Section **7.4.2**.

Body 10 parameters

- `Teff10`: $T_{\rm eff}$ for star 10 in K, tag ....
- `g10`: Logarithm of the surface gravity of star 10 in c.g.s. units, tag g0.
- `P8ratrad`: Ratio of radius of star 1 to radius of body 10, tag kb.
- `P8Q`: Ratio of $(M_1+M_2)/M_{10}$, tag kz. This parameter can have different meanings depending on the flag **`ilog`** above.
- `omega10`: This parameter specifies the ratio of the rotational frequency of star 10 to the orbital frequency, tag o0. This is needed when **`Nterms>0`** and **`iGR>1`** (e.g. when corrections for tides are computed in the dynamical integrator).
- `axis_I10`: This is the inclination of the rotational axis of body 10 if **`ialign=0`**, tag .... See Hosokawa (1953) for helpful diagrams and equations.
- `axis_beta10`: This is the angle of the rotational axis of body 10 with respect to the orbit if **`ialign=0`**, tag .... See Hosokawa (1953) for helpful diagrams and equations.
- `x10U`, `y10U`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 1, tag ..., ...
- `x10B`, `y10B`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 2, tag ..., ...
- `x10V`, `y10V`: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 3, tag ..., ...

- **x10R**, **y10R**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 4, tag ..., ...

- **x10I**, **y10I**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 5, tag ..., ...

- **x10J**, **y10J**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 6, tag ..., ...

- **x10H**, **y10H**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 7, tag ..., ...

- **x10K**, **y10K**: Need two real numbers for this line, separated by spaces. Limb darkening coefficients for body 10, filter 8, tag ..., ...

- **rk10**: The value of the tidal apsidal constant for star 10, tag .... Needs to have **Nterms>0**, **Nbody>2**, and **iGR>1**, see Section **7.4.4**.

- **flux01**: Flux of body 10, filter 1, tag 01 (the number zero and the number one). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 1 using **flux01**.

- **flux02**: Flux of body 10, filter 2, tag 02 (the number zero and the number two). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 2 using **flux02**.

- **flux03**: Flux of body 10, filter 3, tag 03 (the number zero and the number three). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 3 using **flux03**.

- **flux04**: Flux of body 10, filter 4, tag 04 (the number zero and the number four). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 4 using **flux04**.

- **flux05**: Flux of body 10, filter 5, tag 05 (the number zero and the number five). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 5 using **flux05**.

- **flux06**: Flux of body 10, filter 6, tag 06 (the number zero and the number six). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 6 using **flux06**.

- **flux07**: Flux of body 10, filter 7, tag 07 (the number zero and the number seven). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 7 using **flux07**.

- **flux08**: Flux of body 10, filter 8, tag 08 (the number zero and the number eight). Set **iflux=0** to have the code compute the baseline fluxes of each body from their sizes, effective temperatures and gravities. Set **iflux=1** to specify the baseline flux for filter 8 using **flux08**.

This concludes **ELC.inp**.

7.2.2 ELCgap.inp

When using **itime=2**, the model is computed over a long time span, from **t_start** to **t_end** in steps of **t_step** days. For well-detached systems, like most known circumbinary planet systems, not much happens outside the binary eclipses (apart from the occasional transit of a planet). In cases like these, one really does not need to compute the light curve over these out-of-eclipse intervals. To skip these intervals, set the value of **Ngap** to the number of segments to *skip*. These gaps are specified in the **ELCgap.inp** file. This file should have **Ngap** lines of the form:

```
gap01_start    gap01_end
gap02_start    gap02_end
gap03_start    gap03_end
.
.
.
gapNN_start    gapNN_end
```

where **NN=Ngap**. If you are fitting data, you should avoid having data points with times that fall in a gap. No rigorous checking is done for overlapping gaps, so take care the gaps don't overlap. If there is an error reading this file, then the code defaults to **Ngap=0**. If **Ngap=0** then **ELCgap.inp** does not need to exist.

7.2.3 ELCSC.inp

Light curves from NASA's *Kepler* mission had two sampling cadences: "long cadence" with ≈ 30 minute exposures, and "short cadence" with ≈ 1 minute exposures. When fitting long cadence data it is necessary to rebin the model to match the ≈ 30 minute exposures since eclipse or transit profiles look different in the long cadence data compared to the short cadence data, see Section **7.4.5** for details. One can stitch together a light curve with segments of both long and short cadence. Set **lcbin=29.4744**, and set **NSC** equal to the number of segments of short cadence data the composite light curve has. The file **ELCSC.inp** gives the starting and stopping times of each of these segments:

```
SCstart01    SCend01
SCstart02    SCend02
SCstart03    SCend03
.
.
.
SCstartNN    SCendNN
```

where **NN=NSC**. If there is an error reading this file, then the code defaults to **NSC=0**. If **NSC=0** then **ELCSC.inp** does not need to exist.

7.2.4 `ELCdynwin.inp`

When using the dynamical integrator (**`Nterms > 0`** and **`Nbody > 2`**) the equations of motion are integrated between time **`t_start`** and **`t_end`** days in step sizes of **`hh`** days. Various quantities like positions and velocities of the bodies at each time step are saved in memory so that the positions and velocities of a given body can be found at an arbitrary time. If you have data separated by large gaps in time, you can save some memory and reduce the length of the model files by specifying "windows" where all intermediate quantities from the integrator are saved. The default situation is to have one window in time going from **`t_start`** and **`t_end`** days. If you have two or more time windows then set **`Ndynwin`** equal to the number of these windows. The file **`ELCdynwin.inp`** gives these time windows, one per line:

```
dynwin01start      dynwin01end
dynwin02start      dynwin02end
dynwin02start      dynwin02end
.
.
.
dynwinNNstart      dynwinNNend
```

where **`NN=Ndynwin`**. Note that the first time **`dynwin01start`** will be set equal to **`t_start`**, and the ending time **`dynwinNNend`** will be set to **`t_end`**. Also some checking is done to make sure that gaps where the intermediate output quantities from the integrator are *not* saved don't fall in time intervals where the user intends a model to be computed (i.e. **`ELCgap.inp`** and **`ELCdynwin.inp`** are checked for compatibility).

7.3 Output files from ELC

After you run the code, you will have the following output files:

Name	Contents	Name	Contents
ELC.out	Program messages	**ELC.parm**	Derived parameters
ELC.phases	Phases used (if **Nterms=0**)		
modelU.linear	Light curve, filter 1 (flux units)	**modelB.linear**	Light curve, filter 2 (flux units)
modelV.linear	Light curve, filter 3 (flux units)	**modelR.linear**	Light curve, filter 4 (flux units)
modelI.linear	Light curve, filter 5 (flux units)	**modelJ.linear**	Light curve, filter 6 (flux units)
modelH.linear	Light curve, filter 7 (flux units)	**modelK.linear**	Light curve, filter 8 (flux units)
modelU.mag	Light curve, filter 1 (magnitude units)	**modelB.mag**	Light curve, filter 2 (magnitude units)
modelV.mag	Light curve, filter 3 (magnitude units)	**modelR.mag**	Light curve, filter 4 (magnitude units)
modelI.mag	Light curve, filter 5 (magnitude units)	**modelJ.mag**	Light curve, filter 6 (magnitude units)
modelH.mag	Light curve, filter 7 (magnitude units)	**modelK.mag**	Light curve, filter 8 (magnitude units)
lcstar1.linear	Light curve[1], star 1 (flux units)	**lcstar2.linear**	Light curve[1], star 2 (flux units)
lcstar3.linear	Light curve[1], star 3 (flux units), **Nbody >= 3**	**lcstar4.linear**	Light curve[1], star 4 (flux units), **Nbody >= 4**
lcstar5.linear	Light curve[1], star 5 (flux units), **Nbody >= 5**	**lcdisk.linear**	Light curve[1], disk (flux units), **idint=1**, **Nterms=0**
star1.RV	RV curve, star 1 (km sec^{-1})	**star2.RV**	RV curve, star 2 (km sec^{-1})
star3.RV	RV curve, star 3 (km sec^{-1}), **Nbody >= 3**	**star4.RV**	RV curve, star 4 (km sec^{-1}) **Nbody >= 4**
star5.RV	RV curve, star 5 (km sec^{-1}), **Nbody >= 5**	**star1.delRV**	Corrections to RV curve, star 1 (km sec^{-1})
star2.delRV	Corrections to RV curve, star 1 (km sec^{-1})		

The files with the models (**modelV.linear**, **star1.RV**, etc.) are simple ASCII files that you can plot with any graphics program.

The **ELC.out** file contains a list of the input parameters used (in the format of **ELC.inp**), followed by informational messages such as information regarding the Roche geometry (if used), a breakdown of how **ELC** interpreted various combinations of parameters, and the final adopted masses and radii.

[1] filter set by **iRVfilt** flag

The file **ELC.parm** is particularly useful useful as that file contains various derived parameters:

Index	Meaning	Index	Meaning
3	mass of star 1 ($M_\odot$)	4	mass of star 2 ($M_\odot$)
5	radius of star 1 ($R_\odot$)	6	radius of star 2 ($R_\odot$)
7	$\log g_1$ (cgs)	8	$\log g_2$ (cgs)
9	density of star 1 (g/cc)	10	density of star 2 (g/cc)
11	Roche lobe filling factor of star 1	12	Roche lobe filling factor of star 2
13	K-velocity star 1 (km sec^{-1})	14	K-velocity star 2 (km sec^{-1})
15	$V_{\rm rot}\sin i$ of star 1 (km sec^{-1})	16	$V_{\rm rot}\sin i$ of star 2 (km sec^{-1})
17	Scaled $V_{\rm rot}\sin i$ of star 1 (km sec^{-1})	18	Scaled $V_{\rm rot}\sin i$ of star 2 (km sec^{-1})
19	temperature spot 1, star 1 (K)	20	temperature spot 2, star 1 (K)
21	temperature spot 1, star 2 (K)	22	temperature spot 2, star 2 (K)
23	temperature spot 1, disk (K)	24	temperature spot 2, disk (K)
25	disk radius ($R_\odot$)	26	disk thickness ($R_\odot$)
27	duration of X-ray eclipse (deg)	28	impact parameter of primary transit
29	duration of primary transit (days)	30	depth of primary transit
31	duration of primary transit (days), alternate formula	32	mass of star 2 ($M_\oplus$)
33	radius of star 2 ($R_\oplus$)	34	period of binary (days)
35	semimajor axis of binary orbit (AU)	36	eccentricity of binary orbit
37	ω of binary orbit (deg)	38	inclination of binary orbit (deg)
39	nodal angle of binary orbit (deg)	40	$T_{\rm conj}$ of binary orbit
41	$T_{\rm peri}$ of binary orbit	42	mutual i of binary orbit (deg)
43	number of bodies	44	$T_{\rm ref}$
45	mass of body 3 ($M_\odot$)	46	radius of body 3 ($R_\odot$)
47	mass of body 3 ($M_\oplus$)	48	radius of body 3 ($R_\oplus$)
49	$\log g_3$ (cgs)	50	density of body 3 (g/cc)
51	period of body 3 orbit (days)	52	semimajor axis of body 3 orbit (AU)
53	eccentricity of body 3 orbit	54	ω of body 3 orbit (deg)
55	inclination of body 3 orbit (deg)	56	nodal angle of body 3 orbit (deg)
57	$T_{\rm conj}$ of body 3 orbit	58	$T_{\rm peri}$ of body 3 orbit
59	mutual i of body 3 orbit (deg)	60	mass of body 4 ($M_\odot$)
61	radius of body 4 ($R_\odot$)	62	mass of body 4 ($M_\oplus$)
63	radius of body 4 ($R_\oplus$)	64	$\log g_4$ (cgs)
65	density of body 4 (g/cc)	66	period of body 4 orbit (days)
67	semimajor axis of body 4 orbit (AU)	68	eccentricity of body 4 orbit
69	ω of body 4 orbit (deg)	70	inclination of body 4 orbit (deg)
71	nodal angle of body 4 orbit (deg)	72	$T_{\rm conj}$ of body 4 orbit
73	$T_{\rm peri}$ of body 4 orbit	74	mutual i of body 4 orbit (deg)
75	mass of body 5 ($M_\odot$)	76	radius of body 5 ($R_\odot$)
77	mass of body 5 ($M_\oplus$)	78	radius of body 5 ($R_\oplus$)
79	$\log_g 5$ (cgs)	80	density of body 5 (g/cc)
81	period of body 5 (days)	82	semimajor axis of body 5 orbit (AU)
83	eccentricity of body 5 orbit	84	ω of body 5 orbit (deg)
85	inclination of body 5 orbit (deg)	86	nodal angle of body 5 orbit (deg)
87	$T_{\rm conj}$ of body 5 orbit	88	$T_{\rm peri}$ of body 5 orbit
89	mutual i of body 5 orbit (deg)	90	pattern repeats for bodies 6 – 10

7.4 Notes on setting up and running ELC

7.4.1 Specifying the masses and radii

At some point in the modeling process, the mass and radius of each star needs to be defined. In the most basic case using Roche geometry, the equation that gives the surface potentials is a function of the mass ratio Q. The equation is usually written so that the stars have a unit separation. One can scale all of the coordinates by the semimajor axis a (given in physical units) to give the binary a length scale. Once the period P of the binary is given, the total mass can be found using Kepler's Third Law. The individual masses can be found from the total mass and the mass ratio. Thus, at a minimum, one needs to give **`ELC`** values for **`Q`**, **`separ`**, and **`Period`**. Once we have a physical scale for the Roche lobes, the radius of each star can be set using the **`fill1`** and **`fill2`** parameters.

In many cases, those above parameters are not necessarily the most convenient to use when fitting data. For example, it might not be obvious what a sensible range for the semimajor axis a should be. One often has radial velocity data where the K-velocities of one or both stars can be measured, and when one blindly gives a range for the parameter **`separ`** then many of the models generated may have K-velocities well outside the observed range. Also, when using the analytic modes, Roche geometry is avoided so the parameters **`fill1`** and **`fill2`** can't be used to get the radii of the stars.

Thus, more flexibility is needed when specifying the masses and radii of the stars. **`ELC`** lets you set the masses and radii in a number of ways through the use of various input parameters. Since certain parameter combinations might give inconsistent masses and/or radii, there is a priority list of what parameters to use to set the *masses* (this assumes the binary period P is known):

Rank	Condition	Action
0	`Teff2 < 0`, `asini > 0.0`, `primK = 0.0`, `Nterms = 0`	compute `separ`[2]
0	`Teff2 < 0`, `asini > 0.0`, `primK > 0.0`, `Nterms = 0`	compute `Q` and `separ`[2]
1	`primmass > 0.0` and `secmass > 0.0`	compute `Q` and `separ`
2	`masssum > 0.0` and `Q = 0.0`	use `massdiff` to compute `primmass`, `secmass`, `Q`, and `separ`
3	`masssum > 0.0` and `Q > 0.0`	use `Q` to compute `primmass`, `secmass`, and `separ`
4	`primmass > 0.0`, `Q > 0.0`, `primK = 0.0`, `secmass = 0.0`	compute `secmass` and `separ`
5	`secmass > 0.0`, `Q > 0.0`, `primK = 0.0`, `primmass = 0.0`	compute `primmass` and `separ`
6	`primK > 0.0`, `Q > 0.0`, `primmass = 0.0`, `secmass = 0.0`	compute `separ`, `primmass`, `secmass`
7	`primK > 0.0`, `primmass > 0.0`	compute `secmass`, `Q`, `separ`

Once the masses are set, there is a priority list of what parameters to use to set the *radii*:

Rank	Condition	Action
1	`primrad > 0.0`, `secrad > 0.0`	use those values to set R_1 and R_2
2	`radsum > 0.0`, `ratrad = 0.0`	use `raddiff` to compute `primrad`, `secrad`
3	`radsum > 0.0`, `ratrad > 0.0`	compute `primrad`, `secrad`
4	`fracsum > 0`, `ratrad = 0.0`	use `fracdiff` to compute `frac1`, `frac2`, then `primrad` and `secrad`
5	`fracsum > 0`, `ratrad > 0.0`	compute `frac1`, `frac2`, then `primrad` and `secrad`
6	`frac1 > 0.0`, `frac2 > 0.0`	compute `primrad`, `secrad`
7	`frac1 > 0.0`, `ratrad > 0.0`, `frac2 = 0.0`	compute `primrad`, `secrad`
8	`frac1 > 0.0`, `secrad > 0.0`	compute `primrad`
9	`primrad > 0.0`, `frac2 > 0.0`	compute `secrad`
10	`primrad > 0.0`, `ratrad > 0.0`	compute `secrad`
11	`secrad > 0.0`, `ratrad > 0.0`	compute `primrad`
12	`fillsum > 0.0`, `Nterms = 0`	use `filldiff` to compute `fill1`, `fill2`, then `primrad`, `secrad`

When setting the masses, `ELC` works its way down the table of conditions, and when a particular

[2] millisecond pulsar mode

condition is true, the action for setting other parameters is taken and the program flow leaves the subroutine. Likewise, when setting the radii, **ELC** goes down the table of conditions, and when a particular condition is true, the corresponding action for setting other parameters is taken and the program flow leaves the subroutine. The actions **ELC** takes will be recorded in **ELC.out**. For example:

```
M_1 fixed at 0.957959 solar masses and Q fixed at 0.356899.
M_2 set to 0.341894 solar masses and
the separation set to 17.516333 solar radii.

R_1/a fixed at 0.0534594 and R_2/a fixed at 0.0193499,
R_1 set to 0.936412 solar radii, R_2 set to 0.338940 solar radii and
the separation set to 17.516333 solar radii.
```

If you have the the double binary mode selected (**ibinbin=1** and **Nbody=4** see Section **7.4.4** for instructions on how the masses of the third and fourth stars can be set.

7.4.2 Various parameter combination

In many cases, it is convenient to fit combinations of parameters. **ELC** lets you specify parameter combinations involving e and ω (eccentricity and argument of periastron) of each orbit, P and $T_{\rm conj}$ (period and time of conjunction) of each planet orbit, and Ω and i (nodal angle and inclination) of each planet orbit.

eccentricity and argument of periastron: For the binary, you can specify e and ω, $e\cos\omega$ and $e\sin\omega$, or $\sqrt{e}\cos\omega$ and $\sqrt{e}\sin\omega$. Set **iecosflag=0** to specify e and ω (**ecc** and **argper**). Set **iecosflag=1**, **sqcos=0.0**, and **sqsin=0.0** to specify $e\cos\omega$ and $e\sin\omega$ (**ocose** and **osine**). Set **iecosflag=1** and one or both of **sqcos** and **sqsin** to something *not* **0.0** to specify $\sqrt{e}\cos\omega$ and $\sqrt{e}\sin\omega$. For the planet orbits (bodies 3 through 10) you cannot specify e and ω by themselves, you have to either give $e\cos\omega$ and $e\sin\omega$ or you have to give $\sqrt{e}\cos\omega$ and $\sqrt{e}\sin\omega$. If **P1sqecos=0.0** and **P1sqesin=0.0**, then the eccentricity parameters for the body 3 orbit are found from **P1ecos** and **P1esin** (i.e. $e_1\cos\omega_1$ and $e_1\sin\omega_1$). If one or both of **P1sqecos** and **P1sqesin** are *not* **0.0**, then the eccentricity parameters for the body 3 orbit are found from **P1sqecos** and **P1sqesin** (i.e. $\sqrt{e}_1\cos\omega_1$ and $\sqrt{e}_1\sin\omega_1$). Use the same rules for the orbits for bodies 4 through 10.

period and time of conjunction: To give the period and time of conjunction of the body 3 orbit directly, set **P1qTc=-99.0**, **P1pTc=0.0**, and **P1mTc=0.0**, and give values for **P1period** and **P1Tconj**. Follow the same rules for the orbits for bodies 4 through 10. If **P1qTc=-99.0** and either or both of **P1pTc** and **P1mTc** are *not* **0.0**, then the period and time of conjunction for the body 3 orbit will be found from:

```
P1period = 0.5*(P1mTc+P1pTc)
P1Tconj = P1pTc - P1period
```

When fitting for orbital parameters, the range allowed for the time of conjunction of an orbit should not exceed the period for that orbit. In cases where the period is highly uncertain, a rectangular range in the $P-T_{\rm conj}$ plane would let the value of $T_{\rm conj}$ be too large in some places or not large enough in other places. For these situations you can use the "slope parameter" and the "offset" to define a wedge-shaped region in the $P - T_{\rm conj}$ plane. For example (using the body 3 orbit), set the value of **P1qTc** to something between **-1.0** and **1.0**, and give values for **P1period** and **T1off**. Then the time of conjunction is found from:

```
u1 = (1.0 + 0.5*P1qTc)*P1period
u2 = (1.0 - 0.5*P1qTc)*P1period
P1Tconj = 0.5*(u1 - u2) + T1off
```

When optimizing, fit for **P1period** and **P1qTc**. Follow the same rules for the orbits of bodies 4 through 10.

inclination and Nodal angle: If **angsum1=0.0** and **angdiff1=0.0**, then the inclination and nodal of the body 3 orbit are set by **P1incl** and **P1Omega** respectively. The same rules for the orbits of bodies 4 through 10 apply. If either of both of **angsum1** and **angdiff1** are *not* **0.0**, then the inclination and nodal angle of the body 3 orbit are found by:

```
P1incl = 0.5*(angsum1 + angdiff1)
P1Omega = angsum1 - P1incl
```

The same rules for the orbits of bodies 4 through 10 apply.

If you use the above combinations to set the angles or the period and conjunction time, an informative message is recorded in **ELC.out**. For example:

```
Setting body 3 period to 306.673225 days and the conjunction time to 924.839927.

Setting body 3 orbit inclination to 95.350846 degrees and Omega to -0.567823
 degrees.
```

7.4.3 Fast analytic modes

If you have a binary (**Nbody=2**) where the two objects are nearly perfect spheres, you can compute the light and velocity curves using fast analytic expressions (as opposed to tiling an equipotential surface and numerically integrating over the pieces). Set **iatm=0** or **iatm=2**. There are three algorithms to use to compute the light curves:

Gimenez (2006a): Set **Nterms** to a positive integer (**300** is a good starting value), and set **mandel=0** to use the routines given by Gimenez (2006a). In this case you can only use a linear or a quadratic limb darkening law (**ilaw=1**, **ilaw=4**, or **ilaw=5**). The Rossiter effect will be computed for the RV curves. You can adjust the speed and accuracy of the algorithm by increasing the number of terms in the polynomial expansion by making **Nterms** larger (more accuracy) or by making **Nterms** smaller (faster execution time).

Mandel & Agol (2002): Set **Nterms** to any positive integer, and **mandel=1** to use the routines given by Mandel & Agol (2002). In this case you can only use a linear or a quadratic limb darkening law (**ilaw=1**, **ilaw=4**, or **ilaw=5**). The Rossiter effect will *not* be computed for the RV curves.

Don Short method (2018): Set `Nterms` to any positive integer, and `mandel=2` to use these routines. In this case you can use any limb darkening law. The Rossiter effect will be computed for the RV curves if you set `mandel=3`

When using these fast analytic routines, the reflection effect won't work, and ellipsoidal modulations cannot be mimicked. You cannot have spots on the stars. The Doppler beaming corrections will be applied if `beam1 > 0` and/or `beam2 > 0`. You can use with any value of `itime`.

The Mandel & Agol routine is the fastest, but is sometimes prone to glitches. The Short method is nearly as fast as the Mandel & Agol routine to a similar level of accuracy, and is not prone to the glitches that sometimes occur with the Mandel & Agol routine. The Gimenez routine is somewhat slower than the other two routines, and is also not prone to the glitches that sometimes occur with the Mandel & Agol routine.

If you compute a light curve in this mode (fast analytic expressions and `Nbody=2`), and want to switch over to the mode using Roche geometry, set Nterms=0. The resulting light curve should be very similar to the light curve found when `Nterms > 0`, to the extent that the stars are actually spherical and where the reflection effect is unimportant. It is always a good idea to do some comparisons like this.

If you have a binary plus one or more additional bodies (`Nbody > 2`, you will need to have `itime=2` and in most cases you will be required to use one of the three above fast analytic algorithms. The rules given above about the allowed limb darkening laws will apply to models with three or more bodies.

7.4.4 Dynamical integrator and multi-body systems

To compute the light curve through an eclipse or transit, one needs to know the plane-of-the-sky coordinates and radius of each body in question at any given time or orbital phase (in some cases the plane-of-the-sky coordinates can be relative so that one only needs to know the separations of centers). If one has a simple case with two bodies, the positions or the separation of centers can be found analytically if the orbital period, the inclination, mass ratio, etc. are given. If there are three bodies, interactions between the bodies may cause the orbits to be non-Keplerian. Also, GR precession or tidal apsidal motion can make the orbits non-Keplerian. Thus a better way is needed to get the plane-of-the-sky coordinates.

Starting with Version 5, `ELC` has a *dynamical integrator* that can integrate the Newtonian equations of motion (when given suitable initial conditions) to produce coordinates of multiple bodies as a function of time. Thus `ELC` can be a *photodynamical code* similar to the one devised by Carter et al. (2011). At the heart of the photodynamical code is an Ordinary Differential Equation (ODE) solver that uses a symplectic 12th-order Gaussian Runga Kutta devised by Hairer et al. (2006). This ODE solver needs the masses each body (in units of the solar mass), the initial (x, y, z) coordinates of each body, where the $x - y$ plane is the sky plane and where the z-axis is along the line of sight with positive z pointing towards the observer, and the initial velocities of each body in each of the (x, y, z) directions. The units of the distances are AU, and the units of the velocities are AU per day. For convenience, `ELC` converts instantaneous Keplerian orbital elements (the orbital period, the inclination, the eccentricity, etc.) into (x, y, z) positions and velocities using algorithms described by Murray & Dermott (1999). In this way the user does not need to worry about the coordinate transformations.

To start, decide what type of system to integrate:

- **A purely Newtonian system with only the gravitational forces between the bodies, with each one considered a point mass.** If this is the case, set `iGR=0`. The ODE solver uses a second-order iteration scheme with relatively fast convergence.
- **A Newtonian system with additional force terms to model precession in the binary due to General Relativity.** If this is the case, set `iGR=1`. Additional force terms are added to the equations of motion to produce the desired apsidal advance (see Mardling & Lin 2002), and the ODE solver uses a first-order iteration scheme with a slower rate of convergence. If you use `iGR=1` there are no additional parameters needed. The GR corrections are applied between the first four bodies in all combinations. In practice, the main binary and the second binary if you are using the "double" binary mode described below will experience the fastest rates of periastron advance. In most cases the periastron advance of a third body in orbit around the inner binary will have a much slower rate of periastron advance.
- **A Newtonian system with additional force terms to model precession in the binary due to tides on the stars (tidal apsidal motion).** If this is the case, set `iGR=2`. The appropriate force terms are added to the equations of motion (Mardling & Lin 2002) and a first-order iteration scheme is used to solve the equations of motion. When `iGR=2` you need to specify the apsidal k-constant of each star (`rk1`, `rk2`, etc.), the spin parameters for each star (`omega1`, `omega2`, etc.). Currently the tidal corrections are only applied to the two stars in the inner binary and to the two stars in the second binary if the double binary mode is in use.
- **A Newtonian system with both GR precession and tidal apsidal advance modeled.** If this is the case, set `iGR=3`. Supply the same additional constants needed when `iGR=2`. If you have two similar stars in the inner binary and you want to force `rk2=rk1`, then set `iGR=4`. This will give you a Newtonian system with both GR precession and tidal apsidal advance modeled.

Next, arrange the bodies in the proper order. There are two basic types of configurations, a nested or hierarchical system where the inner binary is given first, followed by planets or stars with increasing orbital periods (`ibinbin=0`) or a double binary system with two short-period binaries that orbit each other with a much longer period (`ibinbin=1`).

A common example of the nested system is the circumbinary planet system like Kepler-16. Set `Nbody=3`, `itime=2`, `Nterms=300`, and `ibinbin=0`. Specify the binary parameters in the usual way (for example see Section **7.4.1**). Based on the orbital period of the binary, set the step size used in the dynamical integrator (`hh`). As a general rule of thumb, `hh=Period/400` is a good starting value. I usually round this to two or three decimals. Next, define the reference time at which the instantaneous Keplerian parameters are valid (`Tref`). For best results, (`Tref-t_start`)/`hh` should be an integer, and should also be relatively small since the equations of motion will be integrated from `Tref` to `t_start`, then from `t_start` to `t_end`. The fewest integration steps obviously occur when `Tref=t_start`. **Once you have decided on a value of `Tref`, don't change it.** Next, set up the third body's orbital parameters. Usually, at a minimum, you want the period of the third body's orbit to be at least 6 times longer than the binary period. Finally, set up the parameters of the third body and start the integration. If all goes well, you should get decent light and velocity curves. See Appendix B on page 136 for an example input file.

You are not limited to circumbinary planets, of course. You can also model triple star systems like KOI-126, which consists of a short-period binary containing two M-stars that orbit an F-star with a period of around 34 days (Carter et al. 2011). Note that in this case, one of the M-stars needs to be star 1, the other M-star is star 2, and the F-star is body 3 (see Appendix C on page 142 for an input file). This labeling is reversed relative to the labeling in Carter et al. (2011).

To add a 4th body, set **`Nbody=4`**, and set up the 4th body's orbit (see page 29 for the list of appropriate parameters) and its physical parameters (see page 30 for the parameter list). Generally, you want the orbital period of the 4th body to be at least three or four times longer than the orbital period of the third body. More bodies can be added in a similar way, just adjust the value of **`Nbody`** and give sensible orbital and physical parameters for additional bodies. Appendix D on page 149 gives an input file for Kepler-47, which has 5 bodies. Appendix E on page 157 gives an input file for Kepler-11, which has 7 bodies.

To model a "double binary" such as EPIC 220204960 (Rappaport et al. 2017), set **`ibinbin=1`**. The orbital properties of the first binary go in the usual spot. The orbital properties of the second binary go into the *Body 4 orbital parameters block* in **`ELC.inp`**. The orbital parameters for the orbit between the two binaries go into the *Body 3 orbital parameters block* in **`ELC.inp`**. If desired, there is a special block in **`ELC.inp`** to give the binary+binary stellar parameters, namely the masses and radii of the third and fourth stars. The following gives the priority order for setting the masses:

Rank	Condition	Action
1	**`bin2masssum > 0.0`** and **`bin2Q = 0.0`**	compute **`bin2M3`** and **`bin2M4`** from **`bin2masssum`** and **`bin2massdiff`**
2	**`bin2masssum > 0.0`** and **`bin2Q > 0.0`**	compute **`bin2M3`** and **`bin2M4`** from **`bin2masssum`** and **`bin2Q`**
3	**`bin2M3 > 0.0`** and **`bin2Q > 0.0`** and **`bin2M4 = 0`**	compute **`bin2M4`** from **`bin2M3`** and **`bin2Q`**
4	**`bin2M4 > 0.0`** and **`bin2Q > 0.0`** and **`bin2M3 = 0`**	compute **`bin2M3`** from **`bin2M4`** and **`bin2Q`**
5	**`bin2M3 > 0.0`** and **`bin2M4 > 0`** and **`bin2Q = 0.0`**	set **`bin2M3`** and **`bin2M4`**
6	**`bin2M3 > 0.0`** and **`bin2M4 = 0`** and **`bin2Q = 0.0`**	set **`bin2M3`** and use **`P2Q`** to set **`bin2M4`**
7	**`bin2M4 > 0.0`** and **`bin2M3 = 0`** and **`bin2Q = 0.0`**	set **`bin2M4`** and use **`P1Q`** to set **`bin2M3`**
8	**`bin2M4 = 0.0`** and **`bin2M3 = 0`** and **`bin2Q = 0.0`**	use **`P1Q`** to compute **`bin2M3`** and use **`P2Q`** to compute **`bin2M4`**

Here is the priority order to set the radii of the third and fourth stars:

Rank	Condition	Action
1	`bin2radsum > 0.0` and `bin2ratrad = 0.0`	compute **bin2R3** and **bin2R4** from **bin2radsum** and **bin2raddiff**
2	`bin2radsum > 0.0` and `bin2ratrad > 0.0`	compute **bin2R3** and **bin2R4** from **bin2radsum** and **bin2ratrad**
3	`bin2R3 > 0.0` and `bin2ratrad > 0.0` and `bin2R4 = 0`	compute **bin2R4** from **bin2R3** and **bin2ratrad**
4	`bin2R4 > 0.0` and `bin2ratrad > 0.0` and `bin2R3 = 0`	compute **bin2R3** from **bin2R4** and **bin2ratrad**
5	`bin2R3 > 0.0` and `bin2R4 > 0` and `bin2ratrad = 0.0`	set **bin2R3** and **bin2R4**
6	`bin2R3 > 0.0` and `bin2R4 = 0` and `bin2ratrad = 0.0`	set **bin2R3** and use **P2ratrad** to set **bin2R4**
7	`bin2R4 > 0.0` and `bin2R3 = 0` and `bin2ratrad = 0.0`	set **bin2R4** and use **P1ratrad** to set **bin2R3**
8	`bin2R4 = 0.0` and `bin2R3 = 0` and `bin2ratrad = 0.0`	use **P1ratrad** to compute **bin2R3** and use **P2ratrad** to compute **bin2R4**

The limb darkening coefficients and the fluxes of the third and fourth stars are always specified via the *Body 3 parameters* and the *Body 4 parameters* blocks. An example input file is given in Appendix F on page 166.

When setting up the dynamical integrator, there is a useful flag called **inform**. If **inform=1** (and **ELC** is used), several extra output files are generated:

File name	contents
ELCdynamics.out	This gives a detailed summary about what the initial conditions that went into the integrator were, the positions and velocities at the end of the integration, etc., see below for an example.
ELCdynamics.pos	This gives the time in column 1, followed by the x, y, and z coordinates of all of the bodies at each time step, starting with star 1 (x,y,z), star 2 (x,y,z), the third body (x,y,z), and so on.
ELCdynamics.vel	This gives the time in column 1, followed by the x, y, and z velocities of all of the bodies at each time step, starting with star 1 (x,y,z), star 2 (x,y,z), the third body (x,y,z), and so on.
ELCdynamics.tex	This file has two tables formatted in LaTeX (using the AASTex deluxetable macros) that summarize the initial Keplerian conditions and the initial barycentric coordinates used for the integration. For this option to work properly, you must compile with an option which makes the compiler ignore the '\' (backslash): for example **-fno-backslash** in gfortran.

The file **ELCdynamics.out** file contains useful information. Below is an example for KOI-126 (see Appendix C on page 142).

```
#  First-order system, GR and tidal apsidal
#  included, integration step =  0.0050000 days
#  Number of bodies = 3
#
#  Apsidal parameters:
#
#  w_rot/w_kep(1) = 1.00000000, w_rot/w_kep(2) = 1.00000000
#  v_rot*sin(i)_1 = 7.404732 km/sec, v_rot*sin(i)_2 = 6.724423 km/sec
#
 9.03765599999999947E-02          k2 for star 1
 3.62166783836871176E+00          Omega1, spin rate in rad/day, star 1
 1.04574660000000000E-01          k2 for star 2
 3.62166783836871176E+00          Omega2, spin rate in rad/day, star 2
#
#  Masses and radii:
#
 2.35742998447302510E-01          mass #1 (solar)
 2.08066561695863517E-01          mass #2 (solar)
 1.27914842107595561E+00          mass #3 (solar)
 1.18139200585777680E-03          radius #1 (AU)
 1.07285178198001014E-03          radius #2 (AU)
 9.32493773754227394E-03          radius #3 (AU)
 2.54037269711549996E-01          radius #1 (solar)
 2.30697631394150005E-01          radius #2 (solar)
 2.00516146319735489E+00          radius #3 (solar)
#
#  Limb darkening parameters for Kepler bandpass:
#
 0.60000000000000                 q1 primary, Kipping quad law
 0.30000000000000                 q2 primary, Kipping quad law
#
 0.46475800154489                 u1 primary, quad law
 0.30983866769659                 u2 primary, quad law
#
#
 0.60000000000000                 q1 secondary, Kipping quad law
 0.30000000000000                 q2 secondary, Kipping quad law
#
 0.46475800154489                 u1 secondary, quad law
 0.30983866769659                 u2 secondary, quad law
#
#
 0.34659633000000                 q1 tertiary, Kipping quad law
 0.33023082000000                 q2 tertiary, Kipping quad law
#
 0.38882983272245                 u1 tertiary, quad law
 0.19989449155844                 u2 tertiary, quad law
#
#  Contamination parameters:
#
 0.00950862650610                 Q1, Q5, Q9, Q13, Q17
 0.02042305229100                 Q2, Q6, Q10, Q14
 0.03390528613330                 Q3, Q7, Q11, Q15
 0.00906901280700                 Q4, Q8, Q12, Q16
#
#  Fluxes and temperatures:
#
 0.00030324539655                 flux primary
 0.00020093374123                 flux secondary
```

```
 0.99949582086222                flux tertiary
 3089.4376                       Teff1 (K)
 3004.7143                       Teff2 (K)
 5875.0000                       Teff3 (K)
#
#  Keplerian elements at day     0.000000000
#
 1.73645727360704294E+00          period 1 (days)
-1.15220365891006912E-02          e*cos(omega) 1
-4.25101642440998701E-03          e*sin(omega) 1
 1.54007370676202004E+00          inclination 1 (rad)
 0.00000000000000000E+00          Omega 1 (rad)
-9.90000000000000000E+01          Tconj 1
 2.15663178157688325E-02          semimajor axis 1 (AU)
 1.22812241979875086E-02          eccentricity 1
 2.00251368721720752E+02          arg. per. 1 (deg)
 2.54262837874196180E+02          true anomaly 1 (deg)
 2.55620770294337348E+02          mean anomaly 1 (deg)
 4.54514206595916903E+02          mean longitude 1 (deg)
 8.82397235365320967E+01          inclination 1 (deg)
 0.00000000000000000E+00          Omega 1 (deg)
#
 3.39889361107455912E+01          period 2 (days)
-1.98739295039254998E-01          e*cos(omega) 2
-2.41019764015160648E-01          e*sin(omega) 2
 1.61606441004765022E+00          inclination 2 (rad)
-1.43744886139746231E-01          Omega 2 (rad)
-9.90000000000000000E+01          Tconj 2
 2.46178993321365697E-01          semimajor axis 2 (AU)
 3.12390515282752246E-01          eccentricity 2
 2.30491800954084681E+02          arg. per. 2 (deg)
 1.99994427019308006E+02          true anomaly 2 (deg)
 2.15603545070672709E+02          mean anomaly 2 (deg)
 4.22250252670996645E+02          mean longitude 2 (deg)
 9.25936701170296317E+01          inclination 2 (deg)
-8.23597530239602271E+00          Omega 2 (deg)
#
#  Cartesian coordinates at day         0.00000
#
-7.49570503369230284E-02          x-coordinate, body 1 (AU)
 2.07164510471452949E-02          y-coordinate, body 1 (AU)
-2.29939405802613267E-01          z-coordinate, body 1 (AU)
#
-7.66598710354479962E-02          x-coordinate, body 2 (AU)
 2.13789727598873609E-02          y-coordinate, body 2 (AU)
-2.08381568220596058E-01          z-coordinate, body 2 (AU)
#
 2.62838580993637397E-02          x-coordinate, body 3 (AU)
-7.29548462713221560E-03          y-coordinate, body 3 (AU)
 7.62724948864688668E-02          z-coordinate, body 3 (AU)
#
 6.10432506627227589E-02          x-velocity, body 1 (AU/day)
-3.25734942930762470E-03          y-velocity, body 1 (AU/day)
-1.50731331483835887E-03          z-velocity, body 1 (AU/day)
#
-1.64242033823244968E-02          x-velocity, body 2 (AU/day)
-3.47364982206813545E-03          y-velocity, body 2 (AU/day)
-8.54552601998637140E-03          z-velocity, body 2 (AU/day)
#
```

```
-8.57851304746686145E-03          x-velocity, body 3 (AU/day)
 1.16534381148271750E-03          y-velocity, body 3 (AU/day)
 1.66781019477485601E-03          z-velocity, body 3 (AU/day)
#
-4.02731464118320657E-18          x-coordinate of C.O.M.
 0.00000000000000000E+00          y-coordinate of C.O.M.
-8.05462928236641313E-18          z-coordinate of C.O.M.
#
 0.00000000000000000E+00          x-velocity of C.O.M.
 0.00000000000000000E+00          y-velocity of C.O.M.
 0.00000000000000000E+00          z-velocity of C.O.M.
#
#  Keplerian elements at day     0.000000000
#
 1.73645727360704116E+00          period 1 (days)
-1.15220365891026046E-02          e*cos(omega) 1
-4.25101642441130974E-03          e*sin(omega) 1
 1.54007370676202004E+00          inclination 1 (rad)
-5.53940307328589998E-18          Omega 1 (rad)
-2.19354043985695440E-02          Tconj 1
 2.15663178157688325E-02          semimajor axis 1 (AU)
 1.22812241979875086E-02          eccentricity 1
 2.00251368721720752E+02          arg. per. 1 (deg)
 2.54262837874196180E+02          true anomaly 1 (deg)
 2.55620770294337348E+02          mean anomaly 1 (deg)
 4.54514206595916903E+02          mean longitude 1 (deg)
 8.82397235365320967E+01          inclination 1 (deg)
-3.17384417121079525E-16          Omega 1 (deg)
#
 3.39889361107455699E+01          period 2 (days)
-1.98739295039254582E-01          e*cos(omega) 2
-2.41019764015160343E-01          e*sin(omega) 2
 1.61606441004765022E+00          inclination 2 (rad)
-1.43744886139746231E-01          Omega 2 (rad)
 2.96680833223680951E+00          Tconj 2
 2.46178993321365697E-01          semimajor axis 2 (AU)
 3.12390515282752246E-01          eccentricity 2
 2.30491800954084681E+02          arg. per. 2 (deg)
 1.99994427019308006E+02          true anomaly 2 (deg)
 2.15603545070672709E+02          mean anomaly 2 (deg)
 4.22250252670996645E+02          mean longitude 2 (deg)
 9.25936701170296317E+01          inclination 2 (deg)
-8.23597530239602271E+00          Omega 2 (deg)
#
#  Cartesian coordinates at day         0.00000
#
-7.49570503369230284E-02          x-coordinate, body 1 (AU)
 2.07164510471452949E-02          y-coordinate, body 1 (AU)
-2.29939405802613267E-01          z-coordinate, body 1 (AU)
#
-7.66598710354479962E-02          x-coordinate, body 2 (AU)
 2.13789727598873609E-02          y-coordinate, body 2 (AU)
-2.08381568220596058E-01          z-coordinate, body 2 (AU)
#
 2.62838580993637397E-02          x-coordinate, body 3 (AU)
-7.29548462713221560E-03          y-coordinate, body 3 (AU)
 7.62724948864688668E-02          z-coordinate, body 3 (AU)
#
 6.10432506627227589E-02          x-velocity, body 1 (AU/day)
```

```
-3.25734942930762470E-03          y-velocity, body 1 (AU/day)
-1.50731331483835887E-03          z-velocity, body 1 (AU/day)
#
-1.64242033823244968E-02          x-velocity, body 2 (AU/day)
-3.47364982206813545E-03          y-velocity, body 2 (AU/day)
-8.54552601998637140E-03          z-velocity, body 2 (AU/day)
#
-8.57851304746686145E-03          x-velocity, body 3 (AU/day)
 1.16534381148271750E-03          y-velocity, body 3 (AU/day)
 1.66781019477485601E-03          z-velocity, body 3 (AU/day)
#
-4.02731464118320657E-18          x-coordinate of C.O.M.
 0.00000000000000000E+00          y-coordinate of C.O.M.
-8.05462928236641313E-18          z-coordinate of C.O.M.
#
 0.00000000000000000E+00          x-velocity of C.O.M.
 0.00000000000000000E+00          y-velocity of C.O.M.
 0.00000000000000000E+00          z-velocity of C.O.M.
#
#  Keplerian elements at day      2999.99500
#
 1.72524562117931723E+00          period 1 (days)
-5.16328366729052954E-03          e*cos(omega) 1
 1.69568932019070230E-02          e*sin(omega) 1
 1.52450050523523672E+00          inclination 1 (rad)
-1.26497141623557810E-02          Omega 1 (rad)
 7.92072760382335983E-01          Tconj 1
 2.14733872234143114E-02          semimajor axis 1 (AU)
 1.77255669948746565E-02          eccentricity 1
 1.06935213236213428E+02          arg. per. 1 (deg)
 1.78426915189132927E+02          true anomaly 1 (deg)
 1.78370404653763728E+02          mean anomaly 1 (deg)
 2.84637353191796478E+02          mean longitude 1 (deg)
 8.73474448155407259E+01          inclination 1 (deg)
-7.24775233549851627E-01          Omega 1 (deg)
#
 3.40953543238980359E+01          period 2 (days)
 5.42346286174554063E-02          e*cos(omega) 2
-3.10410373320956223E-01          e*sin(omega) 2
 1.61686935203577353E+00          inclination 2 (rad)
-1.43066311907048105E-01          Omega 2 (rad)
 1.79784093804007590E+01          Tconj 2
 2.46692577637611432E-01          semimajor axis 2 (AU)
 3.15112670019040531E-01          eccentricity 2
 2.79910629976112716E+02          arg. per. 2 (deg)
 3.07445945269392723E+02          true anomaly 2 (deg)
 3.32194226318973335E+02          mean anomaly 2 (deg)
 5.79159479382729387E+02          mean longitude 2 (deg)
 9.26397898957019663E+01          inclination 2 (deg)
-8.19709586277609326E+00          Omega 2 (deg)
#
#  Cartesian coordinates at day      2999.99500
#
 8.98066359851882740E-02          x-coordinate, body 1 (AU)
-1.75436359235388133E-02          y-coordinate, body 1 (AU)
 1.11529107392594182E-01          z-coordinate, body 1 (AU)
#
 9.52119194881170389E-02          x-coordinate, body 2 (AU)
-1.85918825379803865E-02          y-coordinate, body 2 (AU)
```

```
 9.03804789391351449E-02          z-coordinate, body 2 (AU)
#
-3.20382699093694975E-02          x-coordinate, body 3 (AU)
 6.25739615469949207E-03          y-coordinate, body 3 (AU)
-3.52557693428250940E-02          z-coordinate, body 3 (AU)
#
-7.18366379547069334E-02          x-velocity, body 1 (AU/day)
 4.31660534762271560E-03          y-velocity, body 1 (AU/day)
 1.32858520498151163E-02          z-velocity, body 1 (AU/day)
#
 2.62498878880757002E-03          x-velocity, body 2 (AU/day)
 4.25193948508966781E-03          y-velocity, body 2 (AU/day)
 3.22549495633887850E-02          z-velocity, body 2 (AU/day)
#
 1.28122833660325765E-02          x-velocity, body 3 (AU/day)
-1.48715808550471326E-03          y-velocity, body 3 (AU/day)
-7.69513755424775566E-03          z-velocity, body 3 (AU/day)
#
-3.51544295028882104E-13          x-coordinate of C.O.M.
-4.82794479185042775E-14          y-coordinate of C.O.M.
-3.01911669390940248E-13          z-coordinate of C.O.M.
#
-1.32901383159045804E-16          x-velocity of C.O.M.
 1.86263302154723288E-17          y-velocity of C.O.M.
 8.55804361251431365E-17          z-velocity of C.O.M.
#
######################################################################
#
#      2999.995000042272750       Time span of integration (days)
#
#         1.736457273607041       Initial binary period (days)
#         1.725245621179317       Final binary period (days)
#        -0.011211652427724       Difference (days)
#
#        -0.021935404398570       Initial binary Tconj
#         0.792072760382336       Final binary Tconj
#         0.814008164780906       Difference
#
#         0.012281224197988       Initial binary eccentricity
#         0.017725566994875       Final binary eccentricity
#         0.005444342796887       Difference
#
#       200.251368721720752       Initial binary arg. per. (deg)
#       106.935213236213428       Final binary arg. per. (deg)
#       -93.316155485507323       Difference (deg)
#
#        88.239723536532097       Initial binary inclination (deg)
#        87.347444815540726       Final binary inclination (deg)
#        -0.892278720991374       Difference (deg)
#
#        -0.000000000000000       Initial binary Omega (deg)
#        -0.724775233549852       Final binary Omega (deg)
#        -0.724775233549851       Difference (deg)
#
#         0.021566317815769       Initial binary separation (AU)
#         0.021473387223414       Final binary separation (AU)
#        -0.000092930592355       Difference
#
#       255.620770294337348       Initial binary mean anomaly (deg)
```

```
#       178.370404653763728       Final binary mean anomaly (deg)
#       -77.250365640573634       Difference (deg)
#
#       254.262837874196180       Initial binary true anomaly (deg)
#       178.426915189132927       Final binary true anomaly (deg)
#       -75.835922685063238       Difference (deg)
#
#        -0.054013262333961       Delta_omega (deg/cycle)
#         0.000219188737289       GR Delta_omega (deg/cycle)
#         0.000462595075290       Tidal Delta_omega (deg/cycle)
#         0.000681783812579       Total analytic Delta_omega (deg/cycle)
#
######################################################################
#
#        33.988936110745570       Initial body 3 period (days)
#        34.095354323898036       Final body 3 period (days)
#         0.106418213152466       Difference (days)
#
#         2.966808332236810       Initial body 3 Tconj
#        17.978409380400759       Final body 3 Tconj
#        15.011601048163950       Difference
#
#         0.312390515282752       Initial body 3 eccentricity
#         0.315112670019041       Final body 3 eccentricity
#         0.002722154736288       Difference
#
#       230.491800954084681       Initial body 3 arg. per. (deg)
#       279.910629976112716       Final body 3 arg. per. (deg)
#        49.418829022028021       Difference (deg)
#
#        92.593670117029632       Initial body 3 inclination (deg)
#        92.639789895701966       Final body 3 inclination (deg)
#         0.046119778672335       Difference (deg)
#
#        -8.235975302396023       Initial body 3 Omega (deg)
#        -8.197095862776093       Final body 3 Omega (deg)
#         0.038879439619931       Difference (deg)
#
#         0.246178993321366       Initial body 3 separation (AU)
#         0.246692577637611       Final body 3 separation (AU)
#         0.000513584316246       Difference
#
#       215.603545070672709       Initial body 3 mean anomaly (deg)
#       332.194226318973335       Final body 3 mean anomaly (deg)
#       116.590681248300612       Difference (deg)
#
#       199.994427019308006       Initial body 3 true anomaly (deg)
#       307.445945269392723       Final body 3 true anomaly (deg)
#       107.451518250084703       Difference (deg)
#
```

The information in **ELCdynamics.out** should be sufficient for someone with their own photodynamical code to compute the light curve. We are told which type of ODE was solved. In this case, we included GR corrections and tidal apsidal motion. The parameters needed for the tidal apsidal corrections are given. Then, we are given the masses and radii in units of $M_{\odot}$ and AU, respectively. Next, there are the limb darkening parameters, the *Kepler* contamination parameters, and the fluxes. Next, we have the Keplerian elements at the reference time, including the mean anomaly, the true anomaly, and the mean longitude (these quantities are often used in dynamical studies), followed by the Cartesian coordinates and velocities. In particular, note the coordinates and velocity of the Center of Mass (`C.O.M.`) are given. These should be very close to zero. Next, we see the Keplerian elements and Cartesian coordinates and velocity at the start of the integration. Next, we see the Cartesian coordinates at the end of the integration, as well as the `C.O.M.` coordinates. In this case, round-off errors are starting to creep in a little bit, as the differences from zero have grown. Then, we are given some information about the initial and final orbital orbital elements of the binary and information about the apsidal motion. In this case, the analytic expressions for the GR precession and for the apsidal advance due to tides don't come close to accounting for the actual apsidal advance. The difference is caused by the third body. Finally, we are given some information about the initial and final orbital elements for the third body.

If you have **`iwriteeclipse=1`**, you will see output files that give the times of eclipse and transits:

File name	contents
`ELCMocculNtime.dat`	Here `M` refers to body 1 or 2, and `N` refers to body 3 or higher. These files give the times when body 3 is occulted by body 1, etc. Needs **`iwriteeclipse=1`**.
`ELCNtranMtime.dat`	Here `M` refers to body 1 or 2, and `N` refers to body 3 or higher. These files give the times when body 3 transits body 1, etc. Needs **`iwriteeclipse=1`**.
`ELCprimtime.dat`	Gives the times of the primary eclipses, needs **`iwriteeclipse=1`**.
`ELCsectime.dat`	Gives the times of the secondary eclipses, needs **`iwriteeclipse=1`**.

The files have 6 columns:

1. Cycle count, won't be correct for transits or occultations of bodies 3 through 10.
2. Time of eclipse or transit in days.
3. Fake error bar, set to 1.0.
4. Impact parameter. For primary and secondary eclipses these will not exceed 1.0. For transits or occultations of bodies 3 through 10 these can exceed 1 since the code keeps track of each conjunction.
5. Time of ingress. Note this column will be 0.0 if the file was written by an optimizer (eclipse and transit durations are not computed to save time).
6. Time of egress. Note this column will be 0.0 if the file was written by an optimizer (eclipse and transit durations are not computed to save time).

7.4.5 Working with Kepler data

There are a few issues that one encounters when modeling light curves from NASA's *Kepler* mission, and the two main ones are the contamination of the light curve of an eclipsing binary (or transiting exoplanet) and the smoothing of the observed light curve due to the 30 minute sampling in the long cadence mode.

In many of the *Kepler* eclipsing binaries and transiting planets, there is light from other stars in the photometric aperture. The extra light causes the relative depths of the eclipses (or the transits if you have a planet) to go down. **ELC** can account for the contamination in two ways. The first way is to simply apply a correction to the entire light curve using the parameter **contam**. If this parameter is greater than zero and **Iseason=0** (see below), an offset is applied to the model *Kepler* curve (in the U filter position if you have the special ELC.atm file):

$$y_{\rm off} = \frac{k y_{\rm med}}{1-k}$$

where k is the value of the **contam** parameter and $y_{\rm med}$ is median value of the model light curve. The value of **contam** should not exceed 1. During its nominal mission, the *Kepler* spacecraft rotated every 90 days to align its solar panels with the Sun. In some cases, the optimal photometric aperture of a given star would depend on the spacecraft orientation, and when the photometric aperture changes the contamination of the target could change as well. Thus, the second way **ELC** can account for contamination is by using *seasonal* contamination values. Set the control flag **Iseason=1**, and adjust the values of **contamS0**, **contamS1**, **contamS2**, and **contamS3** accordingly. For this to work, set the flag **itime=2**, and use time units of BJD - 2,455,000. The time ranges for the seasonal contamination parameters are:

start time	end time	parameter	start time	end time	parameter
−46.0	0.0	contamS0	0.0	92.0	contamS1
92.0	183.0	contamS2	183.0	276.0	contamS3
276.0	372.0	contamS0	372.0	463.0	contamS1
463.0	553.0	contamS2	553.0	636.0	contamS3
636.0	739.0	contamS0	739.0	834.0	contamS1
834.0	934.0	contamS2	932.0	1015.0	contamS3
1015.0	1106.0	contamS0	1106.0	1205.0	contamS1
1205.0	1304.0	contamS2	1304.0	1391.0	contamS3
1391.0	1481.0	contamS0			

One also needs to account for the relatively long exposure times in typical *Kepler* long cadence data. The parameter **lcbin** is the exposure time of the typical photometric observation you have, in minutes. For example, if you are working with *Kepler* data, the exposure time is close to 29.5 minutes. For this flag to work properly when **itime=0** or **itime=1**, you need to set the **dphase** parameter such that there are 20 to 30 steps per time bin (e.g. one to three minutes for Kepler long cadence data). For example, if you have a binary with a period of 30 days, a step size of one minute in time is a step size in terms of the normalized orbital phase of 0.00002315, which works out to be $360 \times 0.00002315 = 0.008333$ degrees. In this case, you can round up a bit, and set **dphase=0.01**. When using **itime=2** simply put the bin size in minutes, e.g. **lcbin=29.4744**.

Some targets were observed by *Kepler* in short cadence mode, with exposure times of about 1 minute. In this case, no binning is required. You can construct a light curve with long cadence parts and short cadence parts. The flag **NSC** tells **ELC** how many short cadence segments there are, and the input file **ELCSC.inp** (see Section **7.2.3**) lists the start and ending times of these segments.

For radial velocity data, we have an analogous parameter called **rvbin**. It is similar to **lcbin**, but for the typical exposure time (in minutes) of the spectroscopic observations that were used to find the radial velocities. In practice, setting this parameter hardly makes any difference in the cases I have tried so far.

7.4.6 Third light for binaries

Most often when one sees the term “third light” mentioned in the literature, that term refers to the contaminating light of a distant star that alters the observed depths of the primary and secondary eclipses, but otherwise has no effect on the binary itself. If you have such a situation, you can use the parameters **T3**, **g3**, and **SA3** to model the contaminating light when **iatm=1**. If any of **T3**, **g3**, and **SA3** are negative, the third light computation is skipped. If all of **T3**, **g3**, and **SA3** are positive, the code will select an intensity for each of the 8 filters from **ELC.atm** based on the third light temperature (**T3**) and gravity (**g3**). It will scale that intensity based on **SA3**, which is the ratio of the areas of the third star to the first star. If the third light star is a different temperature than the two stars in the binary, then the amount of contaminating flux will depend on the bandpass, and in this case **ELC** will compute self-consistent contamination values for all 8 filters.

If you have an actual third star in the system (**Nbody > 2**), then don’t use the **SA3** parameter. Use **P1ratrad** instead.

7.4.7 Output files for drawing

If **idraw=1**, and if you are using Roche geometry (**Nterms=0**), then ELC will output several files that separate codes can use to plot images of the stars. You will have files of the following form:

star1coo.000.00000—sky coordinates for star 1’s grid points
star1horiz.000.00000—sky coordinates for star 1’s horizon
star1inty.000.00000—contains instructions for making the intensity map for star 1
star1temp.000.00000—contains instructions for making the temperature map for star 1
star1rotkern.000.00000—The rotational broadening kernel for star 1
star1tempgrav.dat—contains the temperatures and gravities of all of the surface elements for star 1
star2tempgrav.dat—same as above, except for star 2
star2coo.000.00000—same as above, except for star 2
star2horiz.000.00000—same as above, except for star 2
star2inty.000.00000—same as above, except for star 2
star2temp.000.00000—same as above, except for star 2
star2rotkern.000.00000—The rotational broadening kernel for star 2
star2tophor.000.00000—The horizon of star 2 above the disk (when present)
diskcoo.000.00000—the sky coordinates of the disk face
diskedge.000.00000—the sky coordinates of the disk edge
diskinty.000.00000—contains instructions for making the intensity map of the disk
diskhoriz.000.00000—the sky coordinates of the disk horizon
disktophor.000.00000—the sky coordinates of the upper rim of the disk

The last 8 numbers specify the orbital phase in degrees. Hence the file **star1inty.010.00000** contains the intensity map of star 1 at the phase of 10 degrees. Note that all of the files may not be written. For example, if one of the stars or the disk is eclipsed at a particular phase, the "coo" files or the "tophor" files may be missing.

You can use any plotting package to plot the sky coordinates. Only the visible points are output. Drawing "grids" or making grayscale or color intensity maps is a bit more involved, and I do this in the code **intplot.f90**, available by request. However, note that it is necessary to have a complex "clipping" routine to account for partially eclipsed grid elements. This clipping work for the case when there are two stars, but still needs work for the case of two stars and a disk. I will try to improve this routine as time permits.

Use the draw option if you want to compute the broadening kernels, or if you want to look at the limb darkening of a particular phase.

7.4.8 Rotational Broadening Kernels

To compute a rotational broadening kernel for star 1 or star 2, follow these steps. First, adjust **ELC.inp** as needed, in particular use Roche geometry with **Nterms=0**. For the purposes of computing broadening kernels, I suggest you use large numbers for **Nalph1**, **Nbet1**, **Nalph2**, and **Nbet2**. Set **idraw=1** and run **ELC**. The rotational broadening kernel(s) will be in files called **star?rotkern.???.?????**, where the first **?** is either **1** or **2**, and where the suffix represents the orbital phase in degrees.

7.5 Physical constants used in ELC

ELC.f90 uses many physical constants. There is a subroutine called **constants.f90** which all other subroutines call when a physical constant is needed. The following table lists all of the physical constants used and the sources cited:

Constant	Value	Notes
`speedlightkm`	2.99792458d5	Speed of light, km $\sec^{-1}$, exact
`speedlightm`	2.99792458d8	Speed of light, m $\sec^{-1}$, exact
`AU`	149597870700.0d0	Astronomical unit in meters, exact, from IAU 2010
`secinday`	86400.0d0	Number of seconds in a day, exact.
`AUperdaytompersec`	`AU/secinday`	Converts AU per day to meters per second
`Gnewton`	6.67408d-11	Newton's gravitational constant, MKS units, http://physics.nist.gov /cuu/index.html
`Gnewtoncgs`	6.67408d-8	Newton's gravitational constant, cgs units, http://physics.nist.gov/cuu/index.html
`G`	(0.01720209895d0)**2	Gravitational Constant in $\mathrm{AU}^2/M_\odot/\mathrm{day}^2$ units, used in the dynamical integrator. From Clemence, G. M. (1965), "The System of Astronomical Constants", Annual Review of Astronomy & Astrophysics, 3, 93.
`stefan`	5.670367d-5	Stefan-Boltzmann constant in cgs units, http://physics.nist.gov/cuu/index.html
`solarrad`	6.9570d8	Nominal solar radius in meters, from IAU 2015 Resolution B3, Mamajek, E. E, Prsa, A, Torres, G., et al., arXiv:1510.07674
`earthmeter`	6.3781d6	The equatorial radius of Earth in meters, from Mamajek, et al.
`gmsun`	`AU**3*G/secinday**2`	$GM_\odot$ in $\mathrm{m}^3/\mathrm{s}^2$
`solarmass`	`gmsun/Gnewton`	Solar mass in kg.
`gmearth`	3.986004418d14	The geocentric gravitational constant, from Lumin, B., Capitaine, N., Fienga, A., et al. 2011, Celest. Mech. Dyn. Ast., 110:293-304
`earthkg`	`gmearth/Gnewton`	Earth mass in kg.
`earthmassinsolar`	`gmsun/gmearth`	Sun mass to Earth mass ratio.
`AUinsolarrad`	`solarrad/AU`	The number of AU in a nominal solar radius.
`gsun`	`100.0d0*Gnewton*` `solarmass/solarrad**2`	Nominal solar gravity in cgs units.
`daysinyear`	365.25d0	The number of days in a year (Julian year).

8. The Optimizers

8.1 Overview

I assume you have at least one light curve to fit. There are three basic options: (i) Fit input data in phase units. Set the parameter **`itime=0`**. You must first "fold" the light curve on the correct phase: if it is an X-ray binary fold the light curve so that the visible star (star 1) is in front at phase 0.0 (obviously you must have the velocity curve or a published value of T_0 and the orbital period so that this can be done). (ii) Fit for the period and/or the phase zero-point. Set the parameter **`itime=1`**. This assumes each orbital cycle of the observed light curve is the same so that the data can be folded on the ephemeris. (iii) Fit in time units where something is changing from orbit to orbit. Set the parameter **`itime=2`** in this case. The most common use of **`itime=2`** is when using three or more bodies and the dynamical integrator (this also needs **`Nterms>0`**). When using Roche geometry (**`Nterms=0`**), you can mimic moving spots by setting **`omega1`** and/or **`omega2`** to values other than 1. The locations of the spots stay fixed, but the positions of the spots with respect to the observers line-of-sight moves in orbital phase. Currently, there is no capability to have other parameters change with time when using Roche geometry.

8.2 Input files needed by the optimizer codes

8.2.1 `gridloop.opt`

All of the optimizer codes need the **`ELC.inp`** file, and possibly the optional input files described in Sections **7.2.2** through **7.2.4**. In addition, there is another required input file called **`gridloop.opt`**. The first 8 lines of this file contain the names of the files with the light curves, separated by the filter. The default filters are the Johnson $UBVRIJHK$ filters. If you don't have the light curves for certain filters, list the name as "none". The next five names are the file names with the radial velocity curves of star 1, star 2, body 3, body 4, and body 5 respectively. If there are no velocity curves, put "none" as the file names. **Note the previous versions of `gridloop.opt` had only two lines for radial velocity curves. The code should be able to read files in either format.**

For example:

```
none
Bfold.bin
Vfold.bin
none
Ifold.bin
none
none
Kfold.bin
RV1.fold
none
none
none
none
```

Note that you cannot name any of your light curve or velocity curve files "none"!

The next line is an integer called **Nvar**, which is the number of parameters (1 to 80) that you can adjust. The next **Nvar** lines should contain strings. For example

```
4
tag01
tag02
tag03
tag04
```

where the strings **tag01**, **tag02**, etc. give the two letter tags of the variables you want to adjust. These tags were given previously in Section **7.2.1** using blue text. There are over 400 possible tags, and I list on the following pages the strings you should use, organized in order of their appearance in **ELC.inp**. These tags are not case sensitive, so that tag **in** is the same as **In**, **iN**, or **IN**. Additional characters past the first two are ignored.

Control flags for filters and limb darkening

tag	Parameter	Note	tag	Parameter	Note
Lx	**Lx/Lopt**	$\log 10(L_x)$ in erg sec^{-1}			

Third light parameters

tag	Parameter	Note	tag	Parameter	Note
t3	**T3**		g3	**g3**	($\log(g_3)$ in cgs)
SA	**SA3**	Surface area ratio			

Control flags for orbital elements

tag	Parameter	Note	tag	Parameter	Note
ps	**pshift**	Phase shift			

Features for Kepler data

tag	Parameter	Note	tag	Parameter	Note
co	`contam`	Kepler contamination	s0	`contamS0`	Season 0 contamination
s1	`contamS1`	Season 1 contamination	s2	`contamS2`	Season 2 contamination
s3	`contamS3`	Season 3 contamination			

Binary orbit parameters

tag	Parameter	Note	tag	Parameter	Note
pe	`Period`	Binary period (days)	T0	`T0`	Binary periastron passage
Tc	`Tconj`	Primary eclipse	bp	`PbpTc`	Binary $P + T_{\rm conj}$
bm	`PbmTc`	Binary $P - T_{\rm conj}$	bq	`binqTc`	$P - T_{\rm conj}$ slope
bt	`Tbinoff`	Binary $P - T_{\rm conj}$ offset	ec	`ecc2`	Binary eccentricity
ar	`argper`	Binary ω (deg)	oc	`ocose`	$e \cos \omega$ binary
os	`osine`	$e \sin \omega$ binary	bc	`sqecos`	$\sqrt{e} \cos \omega$ binary
bs	`sqesin`	$\sqrt{e} \sin \omega$ binary	in	`finc`	Binary inclination (deg)
ob	`Omega_bin`	Binary nodal angle (deg)	pk	`primK`	K_1 (km sec^{-1})
se	`separ`	Binary separation (AU)	dp	`ecosw`	Phase between eclipses

Body 1 parameters

tag	Parameter	Note	tag	Parameter	Note
T1	**Teff1**	Star 1 $T_{\rm eff}$ (K)	g1	**Tgrav1**	Star 1 gravity darkening
l1	**alb1**	Star 1 albedo	o1	**omega1**	Star 1 spin parameter
f1	**fill1**	Star 1 filling factor	pr	**primrad**	Star 1 radius ($R_\odot$)
q1	**frac1**	R_1/a	pm	**primmass**	Star 1 mass ($M_\odot$)
ai	**axis_I1**	Spin axis tilt, star 1	ab	**axis_beta1**	Rotation axis angle, star 1
x1	**x1U**	Star 1 l.d. x-coef. filter 1	y1	**y1U**	Star 1 l.d. y-coef. filter 1
x2	**x1B**	Star 1 l.d. x-coef. filter 2	y2	**y1B**	Star 1 l.d. y-coef. filter 2
x3	**x1V**	Star 1 l.d. x-coef. filter 3	y3	**y1V**	Star 1 l.d. y-coef. filter 3
x4	**x1R**	Star 1 l.d. x-coef. filter 4	y4	**y1R**	Star 1 l.d. y-coef. filter 4
x5	**x1I**	Star 1 l.d. x-coef. filter 5	y5	**y1I**	Star 1 l.d. y-coef. filter 5
x6	**x1J**	Star 1 l.d. x-coef. filter 6	y6	**y1J**	Star 1 l.d. y-coef. filter 6
x7	**x1H**	Star 1 l.d. x-coef. filter 7	y7	**y1H**	Star 1 l.d. y-coef. filter 7
x8	**x1K**	Star 1 l.d. x-coef. filter 8	y8	**y1K**	Star 1 l.d. y-coef. filter 8
b1	**Tspot11**	Temp. fac. spot 1, star 1	b2	**latspot11**	Lat. spot 1, star 1 (deg)
b3	**longspot11**	Long. spot 1, star 1	b4	**radspot11**	radius spot 1, star 1 (deg)
b5	**Tspot21**	Temp. fac. spot 2, star 1	b6	**latspot21**	Lat. spot 2, star 1 (deg)
b7	**longspot21**	Long. spot 2, star 1	b8	**radspot11**	Rad. spot 2, star 1 (deg)
e1	**beam1**	Doppler boosting, star 1	a1	**rk1**	Apsidal constant, star 1
11	**flux11**	Star 1 flux filter 1	12	**flux12**	Star 1 flux filter 2
13	**flux13**	Star 1 flux filter 3	14	**flux14**	Star 1 flux filter 4
15	**flux15**	Star 1 flux filter 5	16	**flux16**	Star 1 flux filter 6
17	**flux17**	Star 1 flux filter 7	18	**flux18**	Star 1 flux filter 8

Body 2 parameters

tag	Parameter	Note	tag	Parameter	Note
T2	`Teff2`	Star 2 $T_{\rm eff}$ (K)	te	`temprat`	$T_{\rm eff,2}/T_{\rm eff,1}$
g2	`Tgrav2`	Star 2 gravity darkening	l2	`alb2`	Star 2 albedo
o2	`omega2`	Star 2 spin parameter	f2	`fill2`	Star 2 filling factor
sf	`fillsum`	`fill1+fill2`	sd	`filldiff`	`fill1-fill2`
sr	`secrad`	Star 2 radius ($R_\odot$)	q2	`frac2`	R_2/a
ra	`ratrad`	Radius ratio (R_1/R_2)	rs	`radsum`	R_1+R_2 ($R_\odot$)
rd	`raddiff`	R_1-R_2 ($R_\odot$)	fs	`fracsum`	$(R_1+R_2)/a$
fd	`fracdiff`	$(R_1-R_2)/a$	ma	`Q`	Mass ratio $Q \equiv M_2/M_1$
md	`masssum`	M_1+M_2 ($M_\odot$)	md	`massdiff`	M_1-M_2 ($M_\odot$)
bi	`axis_I2`	Spin axis tilt, star 2	bb	`axis_beta2`	Rotation axis angle, star 1
z1	`x2U`	Star 2 l.d. x-coef. filter 1	w1	`y2U`	Star 2 l.d. y-coef. filter 1
z2	`x2B`	Star 2 l.d. x-coef. filter 2	w2	`y2B`	Star 2 l.d. y-coef. filter 2
z3	`x2V`	Star 2 l.d. x-coef. filter 3	w3	`y2V`	Star 2 l.d. y-coef. filter 3
z4	`x2R`	Star 2 l.d. x-coef. filter 4	w4	`y2R`	Star 2 l.d. y-coef. filter 4
z5	`x2I`	Star 2 l.d. x-coef. filter 5	w5	`y2I`	Star 2 l.d. y-coef. filter 5
z6	`x2J`	Star 2 l.d. x-coef. filter 6	w6	`y2J`	Star 2 l.d. y-coef. filter 6
z7	`x2H`	Star 2 l.d. x-coef. filter 7	w7	`y2H`	Star 2 l.d. y-coef. filter 7
z8	`x2K`	Star 2 l.d. x-coef. filter 8	w8	`y2K`	Star 2 l.d. y-coef. filter 8
c1	`Tspot12`	Temp. fac. spot 1, star 2	c2	`latspot12`	Lat. spot 1, star 2 (deg)
c3	`longspot12`	Long. spot 1, star 2	c4	`radspot12`	Rad. spot 1, star 2 (deg)
c5	`Tspot22`	Temp. fac. spot 2, star 2	c6	`latspot22`	Lat. spot 2, star 2 (deg)
c7	`longspot22`	Long. spot 2, star 2	c8	`radspot12`	radius spot 2, star 2 (deg)
e2	`beam2`	Doppler boosting, star 2	a2	`rk2`	Apsidal constant, star 2
21	`flux21`	Star 2 flux filter 1	22	`flux22`	Star 2 flux filter 2
23	`flux23`	Star 2 flux filter 3	24	`flux24`	Star 2 flux filter 4
25	`flux25`	Star 2 flux filter 5	26	`flux26`	Star 2 flux filter 6
27	`flux27`	Star 2 flux filter 7	28	`flux28`	Star 2 flux filter 8

Accretion disk parameters

tag	Parameter	Note	tag	Parameter	Note
be	`beta_rim`	Disk opening angle (deg)	ri	`rinner`	Inner disk radius
ro	`router`	Outer disk radius	td	`Tdisk`	Inner disk temperature
xi	`xi`	$T(r)$ ower-law exponent	d1	`Tspotdisk1`	Temp. fac., disk spot 1
d2	`diskspotaz1`	Az. disk spot 1 (deg)	d3	`rspot_cut1`	$r_{\rm cut}$ of disk spot 1
d4	`angsizespot1`	Ang. size spot 1 (deg)	d5	`Tspotdisk2`	Temp. fac., disk spot 2
d6	`diskspotaz2`	Az. disk spot 2 (deg)	d7	`rspot_cut2`	$r_{\rm cut}$ of disk spot 2
d8	`angsizespot2`	Ang. size of spot 2 (deg)			

Body 3 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
tj	`P1Tconj`	Body 3 bary. conj.	tt	`P1period`	Body 3 period (days)
1p	`P1pTc`	Body 3 $P + T_{\rm conj}$ (days)	1m	`P1mTc`	Body 3 $P - T_{\rm conj}$ (days)
1q	`P1qTc`	Body 3 slope in $P + T_{\rm conj}$	1t	`T1off`	Body 3 offset in $P - T_{\rm conj}$
tu	`P1T0`	Body 3 periastron passage	tv	`P1ecos`	Body 3 $e\cos\omega$
tw	`P1esin`	Body 3 $e\sin\omega$	1c	`P1sqecos`	Body 3 $\sqrt{e}\cos\omega$
1s	`P1sqesin`	Body 3 $\sqrt{e}\sin\omega$	tx	`P1incl`	Body 3 inclination (deg)
ty	`P1Omega`	Body 3 nodal angle (deg)	1a	`angsum1`	Body 3 $i + \Omega$ (deg)
1d	`angdiff1`	Body 3 $i - \Omega$ (deg)			

Body 3 parameters

tag	Parameter	Note	tag	Parameter	Note
T3	`Teff3`	Body 3 $T_{\rm eff}$ (K)	g3	`g3`	Body 3 $\log g$ (cgs)
tb	`P1ratrad`	R_1/R_3	tz	`P1Q`	$(M_1 + M_2)/M_3$
o3	`omega3`	Body 3 spin parameter	ci	`axis_I3`	Spin axis tilt body 3
cb	`axis_beta3`	Rotation axis angle body 3	m1	`x3U`	Body 3 l.d. x-coef. filter 1
n1	`y3U`	Body 3 l.d. y-coef. filter 1	m2	`x3B`	Body 3 l.d. x-coef. filter 2
n2	`y3B`	Body 3 l.d. y-coef. filter 2	m3	`x3V`	Body 3 l.d. x-coef. filter 3
n3	`y3V`	Body 3 l.d. y-coef. filter 3	m4	`x3R`	Body 3 l.d. x-coef. filter 4
n4	`y3R`	Body 3 l.d. y-coef. filter 4	m5	`x3I`	Body 3 l.d. x-coef. filter 5
n5	`y3I`	Body 3 l.d. y-coef. filter 5	m6	`x3J`	Body 3 l.d. x-coef. filter 6
n6	`y3J`	Body 3 l.d. y-coef. filter 6	m7	`x3H`	Body 3 l.d. x-coef. filter 7
n7	`y3H`	Body 3 l.d. y-coef. filter 7	m8	`x3K`	Body 3 l.d. x-coef. filter 8
n8	`y3K`	Body 3 l.d. y-coef. filter 8	a3	`rk3`	Apsidal constant, body 3
31	`flux31`	Body 3 flux filter 1	32	`flux32`	Body 3 flux filter 2
33	`flux33`	Body 3 flux filter 3	34	`flux34`	Body 3 flux filter 4
35	`flux35`	Body 3 flux filter 5	36	`flux36`	Body 3 flux filter 6
37	`flux37`	Body 3 flux filter 7	38	`flux38`	Body 3 flux filter 8

Body 4 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
uj	`P2Tconj`	Body 4 bary. conj.	ut	`P2period`	Body 4 period (days)
2p	`P2pTc`	Body 4 $P+T_{\rm conj}$ (days)	2m	`P2mTc`	Body 4 $P-T_{\rm conj}$ (days)
2q	`P2qTc`	Body 4 slope in $P+T_{\rm conj}$	2t	`T2off`	Body 4 offset in $P-T_{\rm conj}$
uu	`P2T0`	Body 4 periastron passage	uv	`P2ecos`	Body 4 $e\cos\omega$
uw	`P2esin`	Body 4 $e\sin\omega$	2c	`P2sqecos`	Body 4 $\sqrt{e}\cos\omega$
2s	`P2sqesin`	Body 4 $\sqrt{e}\sin\omega$	ux	`P2incl`	Body 4 inclination (deg)
uy	`P2Omega`	Body 4 nodal angle (deg)	2a	`angsum2`	Body 4 $i+\Omega$ (deg)
2d	`angdiff2`	Body 4 $i-\Omega$ (deg)			

Body 4 parameters

tag	Parameter	Note	tag	Parameter	Note
T4	`Teff4`	Body 4 $T_{\rm eff}$ (K)	g4	`g4`	Body 4 $\log g$ (cgs)
ub	`P2ratrad`	R_1/R_4	uz	`P2Q`	$(M_1+M_2)/M_4$
o4	`omega4`	Body 4 spin parameter	di	`axis_I4`	Spin axis tilt body 4
db	`axis_beta4`	Rotation axis angle body 4	i1	`x4U`	Body 4 l.d. x-coef. filter 1
j1	`y4U`	Body 4 l.d. y-coef. filter 1	i2	`x4B`	Body 4 l.d. x-coef. filter 2
j2	`y4B`	Body 4 l.d. y-coef. filter 2	i3	`x4V`	Body 4 l.d. x-coef. filter 3
j3	`y4V`	Body 4 l.d. y-coef. filter 3	i4	`x4R`	Body 3 l.d. x-coef. filter 4
j4	`y4R`	Body 4 l.d. y-coef. filter 4	i5	`x4I`	Body 3 l.d. x-coef. filter 5
j5	`y4I`	Body 4 l.d. y-coef. filter 5	i6	`x4J`	Body 3 l.d. x-coef. filter 6
j6	`y4J`	Body 4 l.d. y-coef. filter 6	i7	`x4H`	Body 3 l.d. x-coef. filter 7
j7	`y4H`	Body 4 l.d. y-coef. filter 7	i8	`x4K`	Body 3 l.d. x-coef. filter 8
j8	`y4K`	Body 4 l.d. y-coef. filter 8	a4	`rk4`	Apsidal constant, body 4
41	`flux41`	Body 4 flux filter 1	42	`flux42`	Body 4 flux filter 2
43	`flux43`	Body 4 flux filter 3	44	`flux44`	Body 4 flux filter 4
45	`flux45`	Body 4 flux filter 5	46	`flux46`	Body 4 flux filter 6
47	`flux47`	Body 4 flux filter 7	48	`flux48`	Body 4 flux filter 8

Binary+binary stellar parameters

tag	Parameter	Note	tag	Parameter	Note
91	`bin2masssum`	M_3+M_4 ($M_\odot$)	92	`bin2massdiff`	M_3-M_4 ($M_\odot$)
93	`bin2Q`	M_4/M_3	94	`bin2radsum`	R_3+R_4 ($R_\odot$)
95	`binraddiff`	R_3-R_4 ($R_\odot$)	96	`bin2ratrad`	R_3/R_4
97	`bin2M3`	M_3 ($M_\odot$)	98	`bin2M4`	M_4 ($M_\odot$)
99	`bin2R3`	R_3 ($R_\odot$)	90	`bin2R4`	R_4 ($M_\odot$)

Body 5 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
vj	**P3Tconj**	Body 5 bary. conj.	vt	**P3period**	Body 5 period (days)
3p	**P3pTc**	Body 5 $P + T_{\rm conj}$ (days)	3m	**P3mTc**	Body 5 $P - T_{\rm conj}$ (days)
3q	**P3qTc**	Body 5 slope in $P + T_{\rm conj}$	3t	**T3off**	Body 5 offset in $P - T_{\rm conj}$
vu	**P3T0**	Body 5 periastron passage	vv	**P3ecos**	Body 5 $e\cos\omega$
vw	**P3esin**	Body 5 $e\sin\omega$	3c	**P3sqecos**	Body 5 $\sqrt{e}\cos\omega$
3s	**P3sqesin**	Body 5 $\sqrt{e}\sin\omega$	vx	**P3incl**	Body 5 inclination (deg)
vy	**P3Omega**	Body 5 nodal angle (deg)	3a	**angsum3**	Body 5 $i + \Omega$ (deg)
3d	**angdiff3**	Body 5 $i - \Omega$ (deg)			

Body 5 parameters

tag	Parameter	Note	tag	Parameter	Note
T5	**Teff5**	Body 5 $T_{\rm eff}$ (K)	g5	**g5**	Body 5 $\log g$ (cgs)
vb	**P3ratrad**	R_1/R_5	vz	**P3Q**	$(M_1 + M_2)/M_5$
o5	**omega5**	Body 5 spin parameter	...	**axis_I5**	Spin axis tilt body 5
...	**axis_beta5**	Rotation axis angle body 5	k1	**x5U**	Body 5 l.d. x-coef. filter 1
p1	**y5U**	Body 5 l.d. y-coef. filter 1	k2	**x5B**	Body 5 l.d. x-coef. filter 2
p2	**y5B**	Body 5 l.d. y-coef. filter 2	k3	**x5V**	Body 5 l.d. x-coef. filter 3
p3	**y5V**	Body 5 l.d. y-coef. filter 3	k4	**x5R**	Body 5 l.d. x-coef. filter 4
p4	**y5R**	Body 5 l.d. y-coef. filter 4	k5	**x5I**	Body 5 l.d. x-coef. filter 5
p5	**y5I**	Body 5 l.d. y-coef. filter 5	k6	**x5J**	Body 5 l.d. x-coef. filter 6
p6	**y5J**	Body 5 l.d. y-coef. filter 6	k7	**x5H**	Body 5 l.d. x-coef. filter 7
p7	**y5H**	Body 5 l.d. y-coef. filter 7	k8	**x5K**	Body 5 l.d. x-coef. filter 8
p8	**y5K**	Body 5 l.d. y-coef. filter 8	a5	**rk5**	Apsidal constant, body 5
51	**flux51**	Body 5 flux filter 1	52	**flux52**	Body 5 flux filter 2
53	**flux53**	Body 5 flux filter 3	54	**flux54**	Body 5 flux filter 4
55	**flux55**	Body 5 flux filter 5	56	**flux56**	Body 5 flux filter 6
57	**flux57**	Body 5 flux filter 7	58	**flux58**	Body 5 flux filter 8

Body 6 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
wj	P4Tconj	Body 6 bary. conj.	wt	P4period	Body 6 period (days)
4p	P4pTc	Body 6 $P+T_{\rm conj}$ (days)	4m	P4mTc	Body 6 $P-T_{\rm conj}$ (days)
4q	P4qTc	Body 6 slope in $P+T_{\rm conj}$	4t	T4off	Body 6 offset in $P-T_{\rm conj}$
wu	P4T0	Body 6 periastron passage	wv	P4ecos	Body 6 $e\cos\omega$
ww	P4esin	Body 6 $e\sin\omega$	4c	P4sqecos	Body 6 $\sqrt{e}\cos\omega$
4s	P4sqesin	Body 6 $\sqrt{e}\sin\omega$	wx	P4incl	Body 6 inclination (deg)
wy	P4Omega	Body 6 nodal angle (deg)	4a	angsum4	Body 6 $i+\Omega$ (deg)
4d	angdiff4	Body 6 $i-\Omega$ (deg)			

Body 6 parameters

tag	Parameter	Note	tag	Parameter	Note
T6	Teff6	Body 6 $T_{\rm eff}$ (K)	g6	g6	Body 6 $\log g$ (cgs)
wb	P4ratrad	R_1/R_6	wz	P4Q	$(M_1+M_2)/M_6$
o6	omega6	Body 6 spin parameter	...	axis_I6	Spin axis tilt body 6
...	axis_beta6	Rotation axis angle body 6	...	x6U	Body 6 l.d. x-coef. filter 1
...	y6U	Body 6 l.d. y-coef. filter 1	...	x6B	Body 6 l.d. x-coef. filter 2
...	y6B	Body 6 l.d. y-coef. filter 2	...	x6V	Body 6 l.d. x-coef. filter 3
...	y6V	Body 6 l.d. y-coef. filter 3	...	x6R	Body 6 l.d. x-coef. filter 4
...	y6R	Body 6 l.d. y-coef. filter 4	...	x6I	Body 6 l.d. x-coef. filter 5
...	y6I	Body 6 l.d. y-coef. filter 5	...	x6J	Body 6 l.d. x-coef. filter 6
...	y6J	Body 6 l.d. y-coef. filter 6	...	x6H	Body 6 l.d. x-coef. filter 7
...	y6H	Body 6 l.d. y-coef. filter 7	...	x6K	Body 6 l.d. x-coef. filter 8
...	y6K	Body 6 l.d. y-coef. filter 8	a6	rk6	Apsidal constant, body 6
61	flux61	Body 6 flux filter 1	62	flux62	Body 6 flux filter 2
63	flux63	Body 6 flux filter 3	64	flux64	Body 6 flux filter 4
65	flux65	Body 6 flux filter 5	66	flux66	Body 6 flux filter 6
67	flux67	Body 6 flux filter 7	68	flux68	Body 6 flux filter 8

Body 7 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
xj	**P5Tconj**	Body 7 bary. conj.	xt	**P5period**	Body 7 period (days)
5p	**P5pTc**	Body 7 $P+T_{\mathrm{conj}}$ (days)	5m	**P5mTc**	Body 7 $P-T_{\mathrm{conj}}$ (days)
5q	**P5qTc**	Body 7 slope in $P+T_{\mathrm{conj}}$	5t	**T5off**	Body 7 offset in $P-T_{\mathrm{conj}}$
xu	**P5T0**	Body 7 periastron passage	xv	**P5ecos**	Body 7 $e\cos\omega$
xw	**P5esin**	Body 7 $e\sin\omega$	5c	**P5sqecos**	Body 7 $\sqrt{e}\cos\omega$
5s	**P5sqesin**	Body 7 $\sqrt{e}\sin\omega$	xx	**P5incl**	Body 7 inclination (deg)
xy	**P5Omega**	Body 7 nodal angle (deg)	5a	**angsum5**	Body 7 $i+\Omega$ (deg)
5d	**angdiff5**	Body 7 $i-\Omega$ (deg)			

Body 7 parameters

tag	Parameter	Note	tag	Parameter	Note
T7	**Teff7**	Body 7 T_{eff} (K)	g7	**g7**	Body 7 $\log g$ (cgs)
xb	**P5ratrad**	R_1/R_7	xz	**P5Q**	$(M_1+M_2)/M_7$
o7	**omega7**	Body 7 spin parameter	...	**axis_I7**	Spin axis tilt body 7
...	**axis_beta7**	Rotation axis angle body 7	...	**x7U**	Body 7 l.d. x-coef. filter 1
...	**y7U**	Body 7 l.d. y-coef. filter 1	...	**x7B**	Body 7 l.d. x-coef. filter 2
...	**y7B**	Body 7 l.d. y-coef. filter 2	...	**x7V**	Body 7 l.d. x-coef. filter 3
...	**y7V**	Body 7 l.d. y-coef. filter 3	...	**x7R**	Body 7 l.d. x-coef. filter 4
...	**y7R**	Body 7 l.d. y-coef. filter 4	...	**x7I**	Body 7 l.d. x-coef. filter 5
...	**y7I**	Body 7 l.d. y-coef. filter 5	...	**x7J**	Body 7 l.d. x-coef. filter 6
...	**y7J**	Body 7 l.d. y-coef. filter 6	...	**x7H**	Body 7 l.d. x-coef. filter 7
...	**y7H**	Body 7 l.d. y-coef. filter 7	...	**x7K**	Body 7 l.d. x-coef. filter 8
...	**y7K**	Body 7 l.d. y-coef. filter 8	a7	**rk7**	Apsidal constant, body 7
71	**flux71**	Body 7 flux filter 1	72	**flux72**	Body 7 flux filter 2
73	**flux73**	Body 7 flux filter 3	74	**flux74**	Body 7 flux filter 4
75	**flux75**	Body 7 flux filter 5	76	**flux76**	Body 7 flux filter 6
77	**flux77**	Body 7 flux filter 7	78	**flux78**	Body 7 flux filter 8

Body 8 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
sj	**P6Tconj**	Body 8 bary. conj.	st	**P6period**	Body 8 period (days)
6p	**P6pTc**	Body 8 $P + T_{\rm conj}$ (days)	6m	**P6mTc**	Body 8 $P - T_{\rm conj}$ (days)
6q	**P6qTc**	Body 8 slope in $P + T_{\rm conj}$	6t	**T6off**	Body 8 offset in $P - T_{\rm conj}$
su	**P6T0**	Body 8 periastron passage	sv	**P6ecos**	Body 8 $e \cos \omega$
sw	**P6esin**	Body 8 $e \sin \omega$	6c	**P6sqecos**	Body 8 $\sqrt{e} \cos \omega$
6s	**P6sqesin**	Body 8 $\sqrt{e} \sin \omega$	sx	**P6incl**	Body 8 inclination (deg)
sy	**P6Omega**	Body 8 nodal angle (deg)	6a	**angsum6**	Body 8 $i + \Omega$ (deg)
6d	**angdiff6**	Body 8 $i - \Omega$ (deg)			

Body 8 parameters

tag	Parameter	Note	tag	Parameter	Note
T8	**Teff8**	Body 8 $T_{\rm eff}$ (K)	g8	**g8**	Body 8 $\log g$ (cgs)
sb	**P6ratrad**	R_1/R_8	sz	**P6Q**	$(M_1 + M_2)/M_8$
o8	**omega8**	Body 8 spin parameter	...	**axis_I8**	Spin axis tilt body 8
...	**axis_beta8**	Rotation axis angle body 8	...	**x8U**	Body 8 l.d. x-coef. filter 1
...	**y8U**	Body 8 l.d. y-coef. filter 1	...	**x8B**	Body 8 l.d. x-coef. filter 2
...	**y8B**	Body 8 l.d. y-coef. filter 2	...	**x8V**	Body 8 l.d. x-coef. filter 3
...	**y8V**	Body 8 l.d. y-coef. filter 3	...	**x8R**	Body 8 l.d. x-coef. filter 4
...	**y8R**	Body 8 l.d. y-coef. filter 4	...	**x8I**	Body 8 l.d. x-coef. filter 5
...	**y8I**	Body 8 l.d. y-coef. filter 5	...	**x8J**	Body 8 l.d. x-coef. filter 6
...	**y8J**	Body 8 l.d. y-coef. filter 6	...	**x8H**	Body 8 l.d. x-coef. filter 7
...	**y8H**	Body 8 l.d. y-coef. filter 7	...	**x8K**	Body 8 l.d. x-coef. filter 8
...	**y8K**	Body 8 l.d. y-coef. filter 8	a8	**rk8**	Apsidal constant, body 8
81	**flux81**	Body 8 flux filter 1	82	**flux82**	Body 8 flux filter 2
83	**flux83**	Body 8 flux filter 3	84	**flux84**	Body 8 flux filter 4
85	**flux85**	Body 8 flux filter 5	86	**flux86**	Body 8 flux filter 6
87	**flux87**	Body 8 flux filter 7	88	**flux88**	Body 8 flux filter 8

Body 9 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
hj	**P7Tconj**	Body 9 bary. conj.	ht	**P7period**	Body 9 period (days)
7p	**P7pTc**	Body 9 $P+T_{\rm conj}$ (days)	7m	**P7mTc**	Body 9 $P-T_{\rm conj}$ (days)
7q	**P7qTc**	Body 9 slope in $P+T_{\rm conj}$	7t	**T7off**	Body 9 offset in $P-T_{\rm conj}$
hu	**P7T0**	Body 9 periastron passage	hv	**P7ecos**	Body 9 $e\cos\omega$
hw	**P7esin**	Body 9 $e\sin\omega$	7c	**P7sqecos**	Body 9 $\sqrt{e}\cos\omega$
7s	**P7sqesin**	Body 9 $\sqrt{e}\sin\omega$	hx	**P7incl**	Body 9 inclination (deg)
hy	**P7Omega**	Body 9 nodal angle (deg)	7a	**angsum7**	Body 9 $i+\Omega$ (deg)
7d	**angdiff7**	Body 9 $i-\Omega$ (deg)			

Body 9 parameters

tag	Parameter	Note	tag	Parameter	Note
T9	**Teff9**	Body 9 $T_{\rm eff}$ (K)	g9	**g9**	Body 9 $\log g$ (cgs)
hb	**P7ratrad**	R_1/R_9	hz	**P7Q**	$(M_1+M_2)/M_9$
o9	**omega9**	Body 9 spin parameter	...	**axis_I9**	Spin axis tilt body 9
...	**axis_beta9**	Rotation axis angle body 9	...	**x9U**	Body 9 l.d. x-coef. filter 1
...	**y9U**	Body 9 l.d. y-coef. filter 1	...	**x9B**	Body 9 l.d. x-coef. filter 2
...	**y9B**	Body 9 l.d. y-coef. filter 2	...	**x9V**	Body 9 l.d. x-coef. filter 3
...	**y9V**	Body 9 l.d. y-coef. filter 3	...	**x9R**	Body 9 l.d. x-coef. filter 4
...	**y9R**	Body 9 l.d. y-coef. filter 4	...	**x9I**	Body 9 l.d. x-coef. filter 5
...	**y9I**	Body 9 l.d. y-coef. filter 5	...	**x9J**	Body 9 l.d. x-coef. filter 6
...	**y9J**	Body 9 l.d. y-coef. filter 6	...	**x9H**	Body 9 l.d. x-coef. filter 7
...	**y9H**	Body 9 l.d. y-coef. filter 7	...	**x9K**	Body 9 l.d. x-coef. filter 8
...	**y9K**	Body 9 l.d. y-coef. filter 8	a9	**rk9**	Apsidal constant, body 9
91	**flux91**	Body 9 flux filter 1	92	**flux92**	Body 9 flux filter 2
93	**flux93**	Body 9 flux filter 3	94	**flux94**	Body 9 flux filter 4
95	**flux95**	Body 9 flux filter 5	96	**flux96**	Body 9 flux filter 6
97	**flux97**	Body 9 flux filter 7	98	**flux98**	Body 9 flux filter 8

Body 10 orbital parameters

tag	Parameter	Note	tag	Parameter	Note
kj	**P8Tconj**	Body 10 bary. conj.	kt	**P8period**	Body 10 period (days)
8p	**P8pTc**	Body 10 $P + T_{\rm conj}$ (days)	8m	**P8mTc**	Body 10 $P - T_{\rm conj}$ (days)
8q	**P8qTc**	Body 10 slope in $P + T_{\rm conj}$	8t	**T8off**	Body 10 offset in $P - T_{\rm conj}$
ku	**P8T0**	Body 10 periastron passage	kv	**P8ecos**	Body 10 $e\cos\omega$
kw	**P8esin**	Body 10 $e\sin\omega$	8c	**P8sqecos**	Body 10 $\sqrt{e}\cos\omega$
8s	**P8sqesin**	Body 10 $\sqrt{e}\sin\omega$	kx	**P8incl**	Body 10 inclination (deg)
ky	**P8Omega**	Body 10 nodal angle (deg)	8a	**angsum8**	Body 10 $i + \Omega$ (deg)
8d	**angdiff7**	Body 10 $i - \Omega$ (deg)			

Body 10 parameters

tag	Parameter	Note	tag	Parameter	Note
...	**Teff10**	Body 10 $T_{\rm eff}$ (K)	g0	**g10**	Body 10 $\log g$ (cgs)
kb	**P8ratrad**	R_1/R_{10}	kz	**P8Q**	$(M_1 + M_2)/M_{10}$
o0	**omega10**	Body 10 spin parameter	...	**axis_I10**	Spin axis tilt body 10
...	**axis_beta10**	Rot. axis angle body 10	...	**x10U**	Body 10 l.d. x-coef. filt. 1
...	**y10U**	Body 10 l.d. y-coef. filt. 1	...	**x10B**	Body 10 l.d. x-coef. filt. 2
...	**y10B**	Body 10 l.d. y-coef. filt. 2	...	**x10V**	Body 10 l.d. x-coef. filt. 3
...	**y10V**	Body 10 l.d. y-coef. filt. 3	...	**x10R**	Body 10 l.d. x-coef. filt. 4
...	**y10R**	Body 10 l.d. y-coef. filt. 4	...	**x10I**	Body 10 l.d. x-coef. filt. 5
...	**y10I**	Body 10 l.d. y-coef. filt. 5	...	**x10J**	Body 10 l.d. x-coef. filt. 6
...	**y10J**	Body 10 l.d. y-coef. filt. 6	...	**x10H**	Body 10 l.d. x-coef. filt. 7
...	**y10H**	Body 10 l.d. y-coef. filt. 7	...	**x10K**	Body 10 l.d. x-coef. filt. 8
...	**y10K**	Body 10 l.d. y-coef. filt. 8	...	**rk10**	Apsidal constant, body 10
01	**flux01**	Body 10 flux filter 1	02	**flux02**	Body 10 flux filter 2
03	**flux03**	Body 10 flux filter 3	04	**flux04**	Body 10 flux filter 4
05	**flux05**	Body 10 flux filter 5	06	**flux06**	Body 10 flux filter 6
07	**flux07**	Body 10 flux filter 7	08	**flux08**	Body 10 flux filter 8

Recall that the **gridloop.opt** file has 8 lines giving the file names of light curve data, then 5 lines giving the names of the files with with radial velocity curves. Then, the next line is an integer called **Nvar**, which is the number of parameters (1 to 80) that you can adjust. The next **Nvar** lines should contain strings. For example:

```
4
tag01
tag02
tag03
tag04
```

where the strings **tag01**, **tag02**, etc. give the two letter tags of the variables you want to adjust.

I have just given tables of all of the available tags, and an example of that part of **gridloop.opt** might look like this:

```
5
in
ma
oc
os
pe
```

There are 5 parameters to adjust, the binary inclination (**finc**), the binary mass ratio (**Q**), $e\cos\omega$ for the binary (**ocose**), $e\sin\omega$ for the binary (**osine**), and the orbital period of the binary (**Period**). The values of any parameter not specified in the **gridloop.opt** file by one of these tags are taken from the **ELC.inp** file.

The next section of **gridloop.opt** consists of **Nvar** lines with three numbers each (two real numbers and an integer) of the form

```
real_col01_row01    real_col02_row01    int_col03_row01
real_col01_row02    real_col02_row02    int_col03_row02
real_col01_row03    real_col02_row03    int_col03_row03
.
.
.
real_col01_rowNN    real_col02_rowNN    int_col03_rowNN
```

where **NN=Nvar**. The real numbers in **row01** will correspond to the tag given by **tag01**, and the real numbers in **row02** will correspond to the tag given by **tag02**, and so on. The precise purpose of **real_col01_row01**, **real_col02_row01**, etc. will depend on the optimizer being used as described below.

An example of a complete **gridloop.opt** file is given below:

```
none
B.fold
V.fold
none
I.fold
none
none
none
RV1.fold
RV2.fold
none
none
none
5
f1
f2
```

```
ms
md
in
0.2      0.8     6
0.5      1.0     6
1.0      2.5     6
0.3      1.3     6
60.0     90.0     6
```

We have B, V, and I-band light curves. We have radial velocity curves for star 1 and star 2. We will adjust the Roche lobe filling factors of both stars (**fill1** and **fill2**), the sum and the difference of the masses of star 1 and star 2 (**masssum** and **massdiff**), and the binary inclination (**finc**). The above example **gridloop.opt** file is sufficient to use.

When optimizing, the basic procedure is the **ELC.inp** file is read and the parameters are stored. Values for the fitting parameters given in **gridloop.opt** are generated according the algorithm of the particular optimizer, and these values replace the corresponding values taken from **ELC.inp**. Light and velocity curves are then computed and compared to the given data. Be aware that parameters from **gridloop.opt** might lead to unexpected behaviors. For example, if you fit for **Teff1** and **Teff2**, but **temprat** in **ELC.inp** is greater than zero, then the value of **Teff2** generated by the optimizer would be ignored. Extensive checking is done to help the user avoid such situations, and the optimizer codes will stop and print an error message where unexpected behavior could occur. There is also a check for repeated tags and unknown tags.

Finally, there is an *optional* additional block where you can specify "observed" parameters. In this context an "observed" is a measurable quantity that is not a light curve or a velocity curve such as the surface gravity of one of the stars, the rotational velocity of one of the stars, etc. This quantity has a measured value and an uncertainty, and could be included in the total χ^2. To include one or more observed parameters, first give the value **Nobsvar**, which should be between 0 and 19, or not there at all. The next `Nobsvar` lines should be of the tags of the observed variables, which can be from the following:

tag	Meaning	tag	Meaning
`m1`	Mass of star 1 ($M_\odot$)	`m2`	Mass of star 2 ($M_\odot$)
`r1`	Radius of star 1 ($R_\odot$)	`r2`	Radius of star 2 ($R_\odot$)
`g1`	$\log g$ of star 1 (c.g.s.)	`g2`	$\log g$ of star 2 (c.g.s.)
`k1`	K-velocity of star 1 (km s^{-1})	`k2`	K-velocity of star 2 (km s^{-1})
`in`	i (deg)	`ma`	$Q \equiv M_2/M_1$
`ec`	e of binary	`ar`	ω of binary (deg)
`t1`	$T_{\rm eff}$ for star 1 (K)	`t2`	$T_{\rm eff}$ for star 2 (K)
`t3`	$T_{\rm eff}$ for star 3 (K)	`su`	**`frac1`+`frac2`**
`v1`	$V_{\rm rot}\sin i$ of star 1 (km s^{-1})	`v2`	$V_{\rm rot}\sin i$ of star 2 (km s^{-1})
`xe`	Duration of the X-ray eclipse (deg)	`di`	Disk fraction in band **`iRVfilt`**
`d1`	Disk fraction, filter 1	`d2`	Disk fraction, filter 2
`d3`	Disk fraction, filter 3	`d4`	Disk fraction, filter 4
`d5`	Disk fraction, filter 5	`d6`	Disk fraction, filter 6
`d7`	Disk fraction, filter 7	`d8`	Disk fraction, filter 8
`12`	Flux ratio, star 2 to star 1, filter 1	`22`	Flux ratio, star 2 to star 1, filter 2
`32`	Flux ratio, star 2 to star 1, filter 3	`42`	Flux ratio, star 2 to star 1, filter 4
`52`	Flux ratio, star 2 to star 1, filter 5	`62`	Flux ratio, star 2 to star 1, filter 6
`72`	Flux ratio, star 2 to star 1, filter 7	`82`	Flux ratio, star 2 to star 1, filter 8
`z1`	Disk fraction upper limit, filter 1	`z2`	Disk fraction upper limit, filter 2
`z3`	Disk fraction upper limit, filter 3	`z4`	Disk fraction upper limit, filter 4
`z5`	Disk fraction upper limit, filter 5	`z6`	Disk fraction upper limit, filter 6
`z7`	Disk fraction upper limit, filter 7	`z8`	Disk fraction upper limit, filter 8
`1z`	Flux ratio upper limit, filter 1	`2z`	Flux ratio upper limit, filter 2
`3z`	Flux ratio upper limit, filter 3	`4z`	Flux ratio upper limit, filter 4
`5z`	Flux ratio upper limit, filter 5	`6z`	Flux ratio upper limit, filter 6
`7z`	Flux ratio upper limit, filter 7	`8z`	Flux ratio upper limit, filter 8
`lr`	Flux ratio, star 2 to star 1, filter **`iRVfilt`**	`l3`	Flux ratio, star 3 to star 1, filter **`iRVfilt`**
`l4`	Flux ratio, star 4 to star 1, filter **`iRVfilt`**		

After you list the tags of the observed variables, the next `Nobsvar` lines should have two entries each, namely the observed value and its error. In the case of the X-ray eclipse duration, you have the option of specifying an upper limit. If you have only an upper limit, set the value of `xe` to a negative value. For example if `xe = -10`, this means the eclipse duration is less than 10 degrees. If the model value of the eclipse duration is less than the upper limit, the χ^2 contribution is zero.

8.2.2 `ELCeclipsetimes.opt`

If you have a list of measured eclipse times for a binary (primary and/or secondary eclipses), or transits and/or occultations, you can fit for these times when using the dynamical integrator (`Nterms > 0` and `Nbody > 2`). You will need an additional input file called **`ELCeclipsetimes.opt`** that gives the names of the data files with the eclipse or transit times. There are 40 types of "events", and the following table gives the line number in the **ELCeclipsetimes.opt** file and the meaning of each event. If you don't have observed times corresponding to a given event (e.g. the transit of the primary by body 5), put "none" on the appropriate line in **ELCeclipsetimes.opt**.

Index	Type	Index	Type
1	Primary eclipse	2	Secondary eclipse
3	Transit of body 3 across primary	4	Occultation of body 3 by primary
5	Transit of body 3 across secondary	6	Occultation of body 3 by secondary
7	Transit of body 4 across primary	8	Occultation of body 4 by primary
9	Transit of body 4 across secondary	10	Occultation of body 4 by secondary
11	Transit of body 5 across primary	12	Occultation of body 5 by primary
13	Transit of body 5 across secondary	14	Occultation of body 5 by secondary
15	Transit of body 6 across primary	16	Occultation of body 6 by primary
17	Transit of body 6 across secondary	18	Occultation of body 6 by secondary
19	Transit of body 7 across primary	20	Occultation of body 7 by primary
21	Transit of body 7 across secondary	22	Occultation of body 7 by secondary
23	Transit of body 8 across primary	24	Occultation of body 8 by primary
25	Transit of body 8 across secondary	26	Occultation of body 8 by secondary
27	Transit of body 9 across primary	28	Occultation of body 9 by primary
29	Transit of body 9 across secondary	30	Occultation of body 9 by secondary
31	Transit of body 10 across primary	32	Occultation of body 10 by primary
33	Transit of body 10 across secondary	34	Occultation of body 10 by secondary
35	Eclipse of body 3 by body 4	36	Eclipse of body 4 by body 3
37	Eclipse of body 3 by body 5	38	Eclipse of body 5 by body 3
39	Eclipse of body 4 by body 5	40	Eclipse of body 5 by body 4

Each file with the measured times should have *three* columns:

1. Cycle count (integer), will be ignored.
2. Eclipse or transit time in days (real number).
3. Uncertainty in days (real number).

Note that in the "double binary" model (**ibinbin=1**, **Nbody=4**, and **Nterms>0**), the "primary" eclipses of the second binary go on line 35 of **ELCeclipsetimes.opt**, and the "secondary" eclipses of the second binary go on line 36 of **ELCeclipsetimes.opt**.

For situations where you don't immediately know what the extra transit/occultation events in the light curve are caused by (e.g. you can't tell if the transit was across the primary or the secondary or if the occultation was caused by the primary or the secondary), you can try using the **itran_occul** flag. Set **itran_occul=1** to place the transits and occultations involving the third body into one list, and put the name of the file containing the times of these events on line 3 of **ELCeclipsetimes.opt**. If there is a possibility of a 4th body, set **itran_occul=1** to place the transits and occultations involving the fourth body into one list, and put the name of the file containing the times of these events on line 7 of **ELCeclipsetimes.opt**. Set **itran_occul=2** to put events involving both body 3 and body 4 into one list, and put the name of the file containing the times of these events on line 3 of **ELCeclipsetimes.opt**.

8.2.3 ELCRVdata.opt

One often has the situation where radial velocity data for a given system come from different sources. Some observers attempt to place their radial velocity measurements on some kind of standard system, usually by making use of observations of radial velocity standard stars. Some other observers simply give relative radial velocities and make no attempt to place the velocities on a standard scale. Even when radial velocity standard stars are observed, placing the velocities on a standard system is not a trivial matter and systematic errors on the order of a few to a few dozen meters per second are common. An illustrative example is provided by the circumbinary planetary system Kepler-16. The radial velocities for the primary star presented in the discovery paper (Doyle et al. 2011) came from the Tillinghast 1.5 meter telescope, and are placed on the standard system. Bender et al. (2012) provided additional velocities for the primary star, and also provided velocities for the secondary star. The Bender et al. velocities are offset from the Doyle et al. velocities by about 10 meters per second. Finally, Winn et al. (2011) presented extremely precise radial velocities for the primary star (the typical uncertainties are two meters per second). The velocities are presented as "relative", where relative to what is not specified (most likely it is the "laboratory frame").

When the radial velocity zero-points of the various data sets are not the same, we simply cannot put all of the radial velocity measurements into one file. The best we can do would be to fit all of the data simultaneously while adjusting the zero-points (the γ-velocities) separately for each set, and **ELC** allows you to do this. Set the parameter **ifixgamma** equal to 11 or 12 in **ELC.inp**, where **ifixgamma=11** has the same effect as **ifixgamma=1** and **ifixgamma=12** has the same effect as **ifixgamma=2**. Then, create an additional file called **ELCRVdata.opt**. This file has the following format:

```
Nset
RVdatafile_star1_01
RVdatafile_star2_01
RVdatafile_star3_01
RVdatafile_star4_01
RVdatafile_star5_01
RVdatafile_star1_02
RVdatafile_star2_02
RVdatafile_star3_02
RVdatafile_star4_02
RVdatafile_star5_02
.
.
.
RVdatafile_star1_Nset
RVdatafile_star2_Nset
RVdatafile_star3_Nset
RVdatafile_star4_Nset
```

```
RVdatafile_star5_Nset
```

Here, **`Nset`** is the number of radial velocity data sets you have (limited to 5). For Kepler-16, the **`ELCRVdata.opt`** file might look like this:

```
3
TRES_A.dat
none
none
none
none
Bender_A.dat
Bender_B.dat
none
none
none
Winn_A.dat
none
none
none
none
```

If you have **`ifixgamma=11`** or **`ifixgamma=12`**, the radial velocity data files specified in the **`gridloop.opt`** file won't be used.

8.2.4 Data files for light and velocity curves

You need simple ASCII files that contain light curve data and radial velocity data. These file(s) should have *three* columns:

1. Normalized orbital phase *or* time in days, real numbers.
2. Source magnitude or flux for light curves (lines 1 through 8 in **`gridloop.opt`**), *or* radial velocity in km $\sec^{-1}$ for radial velocity curves (lines 9 through 13 in **`gridloop.opt`**), real numbers.
3. Uncertainty in magnitudes or flux units for light curves (lines 1 through 8 in **`gridloop.opt`**), *or* uncertainties in km $\sec^{-1}$ for radial velocity curves (lines 9 through 13 in **`gridloop.opt`**), real numbers.

8.3 Output from the optimizers

All of the optimizer codes will have screen output giving details of the model inputs, details about the various data sets' contribution to the χ^2, and details about the progress of the particular algorithm. For example, here is part of the output generated by **`amoebaELC`**:

```
** primK=25.668841 Tconj=57180.141278 i=89.5714 frac1=0.0122549
 * ratrad=1.470302 P=69.728078156 ecos(omega)=-0.03614862
 * esin(omega)=0.03939523 contam=0.04289 e=0.0534669
 * argper=132.539 gam1=33.8202 gam2=33.7443 T_peri=57187.6016
 * Tconj1=57213.3997 Tconj2=57180.1413
chi19 = 1701.765444
  --> chiU=263.7703144 chiB=230.3815504 chiV=741.0455497
  --> chiR=128.6698279 chiI=236.0465033 chiRV1=57.4108276
  --> chiRV2=44.1387322
  --> chi(t1)=0.302138
chi_small = 1622.230345
model number=28
iter = 18

rtol=2.000000,  chi^2(hi)=1758.5262908,  chi^2(lo)=0.000000
iter = 20
```

At a minimum, you will always see the individual fitting parameters and their values given. There often will be "extra" parameters given, for example if you are fitting for $e\cos\omega$ and $e\sin\omega$ (tags **oc** and **os**), the derived values of the eccentricity e and the argument of periastron ω will be given. If you have radial velocity curves, the values of the γ-velocities will be listed. If you are fitting for the time of conjunction and have an eccentric orbit, the time of periastron passage is computed and displayed. Set **isuppressfile=1** to suppress this output.

There will be other screen output that varies depending on which optimizer is used.

All optimizer codes will generate output files. The output files in common to all of the optimizers are given in the tables below:

File name	contents
ELC.optimizer	Used by the helper codes, leave alone.
ELC.parm	Contains the derived parameters of the current best model, see Section **7.3**.
gridELC.inp	A file with the same format as **ELC.inp** that gives the parameters of the best current model.
gridELC.opt	A file with the same format as **gridloop.opt** with the parameters of the best current model in the right column (e.g. the numbers **real_col01_row01**, **real_col01_row02**, etc.).

File name	contents
`ELCparm.SUFFIX`	Contains the derived parameters and χ^2 values for each model computed. The length will depend on the number of bodies in the model. The `SUFFIX` string varies depending on which optimizer was used.
`getELCparmvals.com`	A UNIX shell script to make files containing the derived parameter values and the χ^2 for each derived parameter.
`ELCratio.SUFFIX`	File containing the derived flux ratios and the χ^2 values for each model computed, if `ifracswitch=1`. There are 24 flux ratios: star 2 to star 1 for 8 filters, accretion disk to star 1 for 8 filters, and star 3 to star 1 for 8 filters. The `SUFFIX` string varies depending on which optimizer was used.
`chi.SUFFIX`	Contains the total χ^2 and the χ^2 for the individual data sets and observed parameters for each model computed. The meaning of each column follows the order that is given in the screen output. The `SUFFIX` string varies depending on which optimizer was used.
`generation.SUFFIX`	Contains the values of the fitted and "extra" parameters and χ^2 values for each model computed. The length will depend on the number of free parameters used. The `SUFFIX` string varies depending on which optimizer was used.
`getgenerationvals.com`	A UNIX shell script to make files containing the fitted parameter values and the χ^2 for each fitted and extra parameter.
`ELCdynparm.SUFFIX`	Contains the initial conditions (masses, x,y,z coordinates and velocities) and χ^2 values for each model computed, if `Nterms > 0`, `itime=2`, and `Nbody > 2`. The `SUFFIX` string varies depending on which optimizer was used.

File name	contents
ELCdataRV1.fold	The phase-folded radial velocity data for star 1, if **itime=1**. If **itime=2** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **ELCdataRV1A.fold**, **ELCdataRV1B.fold**, etc., depending on what data sets you have.
ELCdataRV2.fold	The phase-folded radial velocity data for star 2, if **itime=1**. If **itime=2** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **ELCdataRV2A.fold**, **ELCdataRV2B.fold**, etc., depending on what data sets you have.
ELCdataRV3.fold	The radial velocity data for star 3, if **itime=2** and **Nbody > 2**, using the user-supplied input times. This file will only appear if you supply input data for that item. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **ELCdataRV3A.fold**, **ELCdataRV3B.fold**, etc., depending on what data sets you have.
ELCdataRV4.fold	The radial velocity data for star 4, if **itime=2** and **Nbody > 3**, using the user-supplied input times. This file will only appear if you supply input data for that item. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **ELCdataRV4A.fold**, **ELCdataRV4B.fold**, etc., depending on what data sets you have.

File name	contents
`ELCdataU.fold`	The phase-folded light curve for filter 1, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.
`ELCdataB.fold`	The phase-folded light curve for filter 2, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.
`ELCdataV.fold`	The phase-folded light curve for filter 3, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.
`ELCdataR.fold`	The phase-folded light curve for filter 4, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.
`ELCdataI.fold`	The phase-folded light curve for filter 5, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.
`ELCdataJ.fold`	The phase-folded light curve for filter 6, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.
`ELCdataH.fold`	The phase-folded light curve for filter 7, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.
`ELCdataK.fold`	The phase-folded light curve for filter 8, if **`itime=1`**. If **`itime=2`** this file will still appear, but with the original input times. This file will only appear if you supply input data for that item.

File name	contents
`ELCresidualsRV1.fold`	Contains the residuals (model − data) for the radial velocity data for star 1, if the input data are supplied. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **`ELCresidualsRV1A.fold`**, **`ELCresidualsRV1B.fold`**, etc., depending on what data sets you have.
`ELCresidualsRV2.fold`	Contains the residuals (model − data) for the radial velocity data for star 2, if the input data are supplied. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **`ELCresidualsRV2A.fold`**, **`ELCresidualsRV2B.fold`**, etc., depending on what data sets you have.
`ELCresidualsRV3.fold`	Contains the residuals (model − data) for the radial velocity data for star 3, if the input data are supplied and if **`Nbody > 2`**. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **`ELCresidualsRV3B.fold`**, **`ELCresidualsRV3B.fold`**, etc., depending on what data sets you have.
`ELCresidualsRV4.fold`	Contains the residuals (model − data) for the radial velocity data for star 4, if the input data are supplied and if **`Nbody > 3`**. If you are fitting more than one radial velocity data set (see Section **8.2.3**), you will see files like **`ELCresidualsRV4B.fold`**, **`ELCresidualsRV4B.fold`**, etc., depending on what data sets you have.

File name	contents
ELCresidualsU.fold	Contains the residuals (model − data) for the light curve data for filter 1, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).
ELCresidualsB.fold	Contains the residuals (model − data) for the light curve data for filter 2, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).
ELCresidualsV.fold	Contains the residuals (model − data) for the light curve data for filter 3, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).
ELCresidualsR.fold	Contains the residuals (model − data) for the light curve data for filter 4, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).
ELCresidualsI.fold	Contains the residuals (model − data) for the light curve data for filter 5, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).
ELCresidualsJ.fold	Contains the residuals (model − data) for the light curve data for filter 6, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).
ELCresidualsH.fold	Contains the residuals (model − data) for the light curve data for filter 7, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).
ELCresidualsK.fold	Contains the residuals (model − data) for the light curve data for filter 8, if the input data are supplied. The residuals will be either in flux units (**imag=1**) or magnitude units (**imag=0**).

File name	contents
`modelU.linear`	Contains the model in flux units for filter 1. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.
`modelB.linear`	Contains the model in flux units for filter 2. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.
`modelV.linear`	Contains the model in flux units for filter 3. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.
`modelR.linear`	Contains the model in flux units for filter 4. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.
`modelI.linear`	Contains the model in flux units for filter 5. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.
`modelJ.linear`	Contains the model in flux units for filter 6. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.
`modelH.linear`	Contains the model in flux units for filter 7. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.
`modelK.linear`	Contains the model in flux units for filter 8. If **`imag=1`** the fluxes will be scaled to match the input data for this filter, if any. If **`itime=0`** or **`itime=1`** the x-values will be given in phase units, otherwise time units will be given.

File name	contents
modelU.mag	Contains the model in magnitude units for filter 1. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.
modelB.mag	Contains the model in magnitude units for filter 2. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.
modelV.mag	Contains the model in magnitude units for filter 3. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.
modelR.mag	Contains the model in magnitude units for filter 4. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.
modelI.mag	Contains the model in magnitude units for filter 5. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.
modelJ.mag	Contains the model in magnitude units for filter 6. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.
modelH.mag	Contains the model in magnitude units for filter 7. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.
modelK.mag	Contains the model in magnitude units for filter 8. If **imag=0** the magnitudes will be shifted to match the input data for this filter, if any. If **itime=0** or **itime=1** the x-values will be given in phase units, otherwise time units will be given.

8.4 The optimizer codes

The following sections describe the various optimizer codes and give specific details about the set-up of the **gridloop.opt** file. Some of the optimizer codes have optional input files, and the contents of those files are given.

8.4.1 amoebaELC

The program **amoebaELC.f90** is an optimizer program based on the downhill simplex method (Press et al. 1992). This algorithm needs a starting model, and the parameters of the starting model are given by the real numbers in **real_col01_row01**, **real_col01_row02**, ..., **real_col01_rowNN**, where **NN=Nvar**. Recall that the number given in **real_col01_row01** should correspond to the tag **tag01**. Likewise, the number given in **real_col01_row02** should correspond to the tag **tag02**, and so on. This algorithm also needs step sizes for all of the free parameters, and these step sizes are specified by the real numbers in **real_col02_row01**, **real_col02_row02**, ..., **real_col02_rowNN**, where **NN=Nvar**. Finally, the integer given by **int_col03_row01** gives the number of iterations, and **int_col03_row02** gives restarts of the algorithm. All of the other integers given in the right column in rows **3** to **Nvar** are required to be there, but will be ignored.

The example **gridloop.opt** file given previously can be adapted to work with **amoebaELC** as follows:

```
none
B.fold
V.fold
none
I.fold
none
none
none
RV1.fold
RV2.fold
none
none
none
5
f1
f2
ms
md
in
0.66      0.01     3
0.95      0.01     2
2.03      0.05     6
0.97      0.05     6
82.4    0.1    6
1
lr
0.67    0.02
```

As before, we have B, V, and I-band light curves. We have radial velocity curves for star 1 and star 2. We will adjust the Roche lobe filling factors of both stars (**fill1** and **fill2**), the sum and the difference of the masses of star 1 and star 2 (**masssum** and **massdiff**), and the binary inclination (**finc**). The model has initial parameter values of **fill1=0.66**, **fill2=0.95**, **masssum=2.03**, **massdiff=0.97**, and **finc=82.4**. The step sizes are Δ**fill1=0.01**, Δ**fill2=0.01**, Δ**masssum=0.05**, Δ**massdiff=0.97**, and Δ**finc=82.4**. There will be **3** iterations and **2** restarts. In addition, we have one observed constraint, namely the flux ratio between star 2 and star 1 (**tag lr**, the letter L). The observed value is 0.67 ± 0.02.

I really don't have any guidelines for choosing the step sizes. Usually 1% of the parameter value is a good place to start. This is a somewhat local optimizer, but one advantage is that derivatives of the χ^2 hyper-surface are not needed. Note that the parameters are *unbounded*, and can potentially move off into undesirable parts of parameter space.

The **SUFFIX** string noted in Section **8.3** is **1000000**. Thus you will have files called **chi.1000000**, `generation.1000000`, etc.

See Section **8.9** for guidelines on how to get uncertainties on the fitted and derived parameters.

8.4.2 demcmcELC.f90

The program `demcmcELC.f90` is an optimizer program based on a differential evolution Markov Chain (Ter Braak 2006). To use this code, you need to specify the ranges of the free parameters (the priors), and the code will attempt to generate a posterior sample using a number of model "chains" that evolve together. In the `gridloop.opt` file, the value of `real_col01_row01` will represent the lower allowed bound of the parameter given by `tag01`, the value of `real_col01_row02` will represent the lower allowed bound of the parameter given by `tag02`, and so on. The value of `real_col02_row01` will represent the upper allowed bound of the parameter given by `tag01`, The value of `real_col02_row02` will represent the upper allowed bound of the parameter given by `tag02`, and so on. The value of `int_col03_row01` gives the number of chains, and the value of `int_col03_row02` gives the number of generations over which the chains evolve. All of the other integers given in the right column in rows **3** to `Nvar` are required to be there, but will be ignored.

The example `gridloop.opt` file given previously can be adapted to work with `demcmcELC` as follows:

```
none
B.fold
V.fold
none
I.fold
none
none
none
RV1.fold
RV2.fold
none
none
none
5
f1
f2
ms
md
in
0.30     0.80    15
0.50     1.00    10000
2.00     3.00    6
1.00     2.00    6
60.0    90.0    6
```

As before, we have B, V, and I-band light curves. We have radial velocity curves for star 1 and star 2. We will adjust the Roche lobe filling factors of both stars (**fill1** and **fill2**), the sum and the difference of the masses of star 1 and star 2 (**masssum** and **massdiff**), and the binary inclination (**finc**). The model has parameter ranges (e.g. priors) of `0.30` to `0.80` for **fill1**, `0.50` to `1.00` for **fill2**, `2.00` to `3.00` for **masssum**, `1.00` to `2.00` for **massdiff**, and `60.0` to `90.0` for **finc**. There will be a population of `15` ($N_{\mathrm{pop}} \approx 3N_{\mathrm{var}}$ is a good starting place for the number of chains), and the chains will evolve for `10000` generations. The parameter values for all models will always be inside the ranges given in **gridloop.opt**.

The **SUFFIX** string noted in Section **8.3** starts at `1000000` and will increment by `1` after each generation. Thus you will have files called **chi.1000000**, **chi.1000001**, **chi.1000002**, and so on. The **gridELC.inp** file noted in Section **8.3** will be renamed **ELC.SUFFIX** where **SUFFIX** is the counter for the generation where the current best model was computed. You will always have a file called **ELC.1000000** after the first generation, and you may end up with others, for example **ELC.1000009**, **ELC.1000032**, **ELC.1000511**, etc. Likewise, the **gridELC.opt** file noted in Section **8.3** will be renamed **gridELC.SUFFIX** where **SUFFIX** has the same meaning. You will always end up with a file called **gridELC.1000000**, and like before there could be others. The parameter **istart** described below can be used to change the starting value of **SUFFIX**.

The **gridELC.1000000** and similar files have a format slightly different than that of **gridloop.opt**. Instead of three columns of numbers, you will see 5 columns of numbers, a column of tags, and an integer counter in the **gridELC.SUFFIX** files. The first column of numbers give the parameters for the current best model. Columns 2 and 3 are the same the corresponding ones in **gridloop.opt**. The numbers in column 4 give how far the best-fitting value is from the lower boundary in normalized units and the numbers in column 5 give how far the best-fitting value is from the upper boundary in normalized units. If you see numbers either very close to 0.0 or 1.0 in columns 4 and 5, the best-fitting model has run up against a "wall", and those particular ranges in the parameters should be adjusted accordingly. If there is something very close to 1.0 in column 4, make the upper bound in **gridloop.opt** larger, and if there is something very close to 0.0 in column 4, make the lower bound in **gridloop.opt** smaller.

There are three main ways to specify the initial population of chains: (i) you can have completely random parameters in the chains if **ielite=0**; (ii) you can have 1 or more "elite" chains with pre-determined parameters and have the rest of the chains initially set randomly (**ielite > 0**, see section **8.5** for information on how to include elite models into the initial chains); or (iii) you can have 1 or more "elite" chains with pre-determined parameters and have the rest of the chains be tweaked copies of the elite ones (**ielite > 0**). To help you specify the non-elite chains, there is an optional file called **demcmcELC.inp**. This file has 14 lines with the following parameters:

gamma (real number)
gsig (real number)
rfracmod (real number)
numparm (integer)
ipurgestart (integer)
ipurgeint (integer)
purgefactor (real number)
istuck (integer)
target (real number)
itemp (integer)

tempstart (real number)
istart (integer)
iskip (integer)
ioldseed (integer)

gamma is a scale length used to determine the "jumps" that the chains use to advance. A good starting value is **gamma = 2.38/sqrt(2*Nvar)**. This scale length is periodically adjusted by the code to achieve a set number of accepted jumps.

gsig is the standard deviation of a random Gaussian deviate (in normalized units) that is used to "tweak" parameter values.

When initializing the chains, there will be **ielite** chains that have user defined parameter values (see Section **8.5**). The parameter **rfracmod** lets you decide how the rest of the chains are set. Specifically, **rfracmod** is the fraction of the remaining chains that will be "tweaks" of randomly selected elite chains. Tweaked chains are made by adding random Gaussian deviates (scaled by **gsig**) to each parameter in the elite model. The code to do this looks something like

```
  DO i=ielite+1,Npop
    {set i1 to be the index of a randomly selected ielite chain}
    prob=ran1(idum)
    IF(prob.lt.rfracmod)THEN
      DO j=1,Nvar                   ! loop over the free parameters
!
!  get a random Gaussian deviate and scale it to the units of the parameter
!
        urand=gsig*gasdev(idum)
        urand=urand*(vhigh(j)-vlow(j))
!
!  add the scaled deviate to the parameter value from the elite chain
!  and make sure the new parameter value is inside the defined
!  parameter range
!
        parmarray(j,i)=parmarray(j,i1)+urand
        IF(parmarray(j,i).lt.vlow(j))parmarray(j,i)=vlow(j)
        IF(parmarray(j,i).gt.vhigh(j))parmarray(j,i)=vhigh(j)
      END DO
    END IF
  END DO
```

If **rfracmod=1.0** and **gsig** is something very small (say `0.0001`) then *all* of the remaining chains will be very close copies of randomly selected elite chains. In the limit of **gsig=0.0** the remaining chains will be exact copies of randomly selected elite chains, which is not desirable. Conversely, if **rfracmod=0.0**, then none of the remaining chains will be tweaked copies of randomly selected elite chains.

What happens to those chains that were not randomly set to be tweaked copies of the elite chains? For those, we select, for each remaining chain, **numparm** parameters at random and add a random offset to those parameter values, making sure the offset parameter is still in the allowed range. The code to do this looks something like

```
  DO i=ielite+1,Npop
```

```
    {set i1 to be the index of a randomly selected ielite chain}
    prob=ran1(idum)
    IF(prob.ge.rfracmod)THEN
!
!  make the model at index i match the elite model with index i1
!
      DO j=1,Nvar
        parmarray(j,i)=parmarray(j,i1)
      END DO
!
      DO j=1,numparm
!
!  randomly select a parameter index between 1 and Nvar
!
        i2=int(ran1(idum)*dble(Nvar))+1
        IF(i2.lt.1)i2=1
        IF(i2.gt.Nvar)i2=Nvar
!
!  find a random value for that parameter, scaled to the parameter
!  units
!
        urand=ran1(idum)*(vhigh(i2)-vlow(i2))
!
!  add that offset to the parameter, and make sure the offset
!  value is still within the allowed range
!
        parmarray(i2,i)=vlow(i2)+urand
        IF(parmarray(i2,i).lt.vlow(i2))parmarray(i2,i)=vlow(i2)
        IF(parmarray(i2,i).gt.vhigh(i2))parmarray(i2,i)=vhigh(i2)
      END DO
    END IF
  END DO
```

If you have **numparm=1**, then the remaining chains will be copies of randomly select elite chains, with the exception of one random parameter (for example the inclination **finc** might be different). As **numparm** grows, the remaining chains become more and more random.

The next four parameters in **demcmcELC.inp** are related to *purging*. Occasionally chains will become trapped in local minima and can't move over to regions of higher likelihood. Also, chains can go many generations without jumping (i.e. they get "stuck"). Generally speaking, you want to let the chains "burn" in before purging trapped and stuck chains. The parameter **ipurgestart** gives generation number when purging can start. I typically set **ipurgestart=2000**. After **ipurgestart** generations, the model with the highest χ^2 will be replaced by the best overall model if $\chi^2_{\rm worst} > \chi^2_{\rm thresh} + \chi^2_{\rm small}$, where $\chi^2_{\rm thresh}$ is set by the **purgefactor** parameter. After the first purging, wait **ipurgeint** generations before another purge. If $\chi^2_{\rm worst} < \chi^2_{\rm thresh} + \chi^2_{\rm small}$, then nothing happens. The code also keeps track of the intervals between jumps for each chain. After **ipurgestart** generations, if a chain has a jump interval larger than **istuck**, then that chain is replaced by the best overall model.

The value of **purgefactor** depends on the situation. You want to avoid having a value of **purgefactor** that is too small. Likewise, avoid small values of **ipurgeint**, otherwise the purged models won't have time to move and you will get several models with the same parameters. I usually use **ipurgeint=25**.

`target` is the desired fraction of jumps that are accepted. You usually want an acceptance fraction between about 0.2 and 0.4.

The next two parameters in `demcmcELC.inp` are related to *tempering*. As the chains evolve, new models are tried (these are "proposals") and the χ^2 of the new models are found. If χ^2 of the proposal is lower, then that proposed jump is always accepted. If the χ^2 of the proposal is higher, then the jump is accepted with a probability of $\exp(-\Delta\chi^2/2)$. During an optional "tempering" phase, the value of $\Delta\chi^2$ is scaled by a number larger than 1.0 (the scale factor is called the "temperature"). This makes the χ^2 landscape appear "flatter", and chains have an easier time taking "uphill" jumps (i.e. jumps after an increase in χ^2). This helps the chains "disperse" by a small amount. If the temperature is set properly, the chains can move a modest amount while the typical χ^2 values are larger than the minimum value, but not so large that the chains can move back to the optimal region of parameter space after the tempering phase ends.

If you want to try tempering, try the following. Set the initial temperature (`tempstart`) to something more than 1.0 (try `tempstart=10.0`). The length of the tempering phase is given by the parameter `itemp`, try something like `itemp=100` to start with. After the first generation, the temperature decreases in a linear manner until `itemp` generations have passed. After this, the temperature remains at 1.0. If you set `tempstart=1.0` or `itemp=1` then there will be no tempering phase.

`istart` is an offset that is added to `SUFFIX`. For example, if `istart=10` then the value of `SUFFIX` will be `1000010` after the initial generation, `1000011` after the next generation, and so on. This feature is useful if you stopped a run and want to restart it, see Section **8.6**.

`iskip` is a parameter you can use to reduce the number of output files written by the code. If `iskip > 0` then after the initial generation, the code will wait `iskip` generations before writing the output files with the `SUFFIX` ending that are normally written after each generation (e.g. `chi.SUFFIX`, `generation.SUFFIX`, etc.). For example, if `iskip=3`, the values of `SUFFIX` will be `1000000`, `1000004`, `1000008`, `1000012`, `1000016`, etc. This feature also works in combination with the `istart` parameter described above. For example, if `istart = 10` and `iskip = 3` the values of `SUFFIX` will be `1000010`, `1000014`, `1000018`, `1000022`, `1000026`, etc. Note that if a model that is better than the current best model is found, the output files associated with that model (e.g. `modelU.linear`, `ELC.SUFFIX`, `gridELC.SUFFIX`, etc.) will always be written at the end of that generation.

`ioldseed` is an integer number used to restart `demcmcELC.f90`. If you are not restarting, set `ioldseed = -1`, otherwise set this parameter equal to the number in the fifth column of `demcmcELC.out` of the generation you intend to restart. See Section **8.6** for further details.

In addition to the standard output files from the optimizers (Section **8.3**), `demcmcELC` will generation additional output files:

File name	contents
demcmcELC.jumps	Contains jump statistics in three columns: generation number, jump interval of chain k, the index k.
demcmcELC.out	Contains 5 columns: the generation number, the fraction of accepted jumps, the current value of **gamma**, the current value of the temperature, and an integer that can be used for a restart.
demcmcELC_chi.SUFFIX	Similar to the file **chi.SUFFIX**, but for the models in the chains.
demcmc_fitparm.SUFFIX	Similar to the file **generation.SUFFIX**, but for the models in the chains, and with the first two columns omitted.
getdemcmc_fitparmvals.com	A Unix shell script to make files containing the fitted parameter values and the χ^2 from the **demcmc_fitparm.SUFFIX** files.
demcmc_starparm.SUFFIX	Similar to the file **ELCparm.SUFFIX**, but for the models in the chains, and with the first two columns omitted.
getdemcmc_starparmvals.com	A Unix shell script to make files containing the derived parameter values and the χ^2 from the **demcmc_starparm.SUFFIX** files.
demcmc_dynparm.SUFFIX	Similar to the file **ELCdynparm.SUFFIX**, but for the models in the chains, and with the first two columns omitted. Needs **Nterms > 0** and **Nbody > 2**.
demcmc_ratio.SUFFIX	Similar to the file **ELCratio.SUFFIX**, but for the models in the chains, and with the first two columns omitted. Needs **ifracswitch=1**.

demcmcELC can be run in parallel, see Section **8.7**. See Section **8.9** for guidelines on how to get uncertainties on the fitted and derived parameters.

8.4.3 geneticELC.f90

The program **geneticELC.f90** is an optimizer program based on a genetic algorithm (Charbonneau 1995). The format of **gridloop.opt** is same format as that for **demcmcELC.f90**, where the lower bounds of the input parameters appear in the first column of numbers, and the upper bounds of the input parameters appear in the middle column of numbers. The value of **int_col03_row01** is $N_{\rm pop}$, which is the number of models in the *population*. As a guideline, make $N_{\rm pop} \approx N_{\rm var}$. The value of **int_col03_row02** is the number of generations over which the models evolve. All of the other integers given in the right column in rows **3** to **Nvar** are required to be there, but will be ignored.

The **SUFFIX** string noted in Section **8.3** starts at **1000000** and will increment by **1** after each generation, as described in Section **8.4.2**. The **gridELC.inp** file noted in Section **8.3** will be renamed **ELC.SUFFIX** where **SUFFIX** is the counter for the generation where the current best model was computed, as described in Section **8.4.2**. Likewise, the **gridELC.opt** file noted in Section **8.3** will be renamed **gridELC.SUFFIX** where **SUFFIX** has the same meaning. You can use these files to see if any parameters are getting close to their respective lower or upper limits.

You can insert one or more elite models into the initial population, see Section **8.5**. `geneticELC` can be run in parallel, see Section **8.7**. See Section **8.9** for guidelines on how to get uncertainties on the fitted and derived parameters.

8.4.4 gridELC

The program `gridELC.f90` is an optimizer program based on the "grid search" routine given in Bevington (1969). This algorithm needs an initial model, a list of free parameters, and a list of step sizes for each free parameter. Each parameter is offset by the step size, and the stepping is continued, either in the positive direction or the negative direction, until a minimum in χ^2 is found. The format of `gridloop.opt` is the same format as that for `amoebaELC.f90`, where the starting values of the input parameters appear in the first column of numbers, and where the parameter step sizes appear in the middle column of numbers. The number of iterations (e.g. loops over all of the parameters) is given by the value of `int_col03_row01`.

The `SUFFIX` string noted in Section **8.3** is `0000`, the same as for `amoebaELC.f90`, see Section **8.4.1**.

Like `amoebaELC.f90`, the ranges of the free parameters are not bound. Also, the final place in parameter space that `gridELC` ends up at usually depends on the order of the parameters in `gridloop.opt`.

See Section **8.9** for guidelines on estimating the uncertainties on the fitted and derived parameters.

8.4.5 hammerELC.f90

The program `hammerELC.f90` is an implementation of the affine-invariant code EMCEE Hammer (Foreman-Macky et al. 2013). You specify the ranges of the free parameters (the priors), and a large number of "walkers" explore the parameter space. The format of the `gridloop.opt` file is nearly identical to format used by `demcmcELC.f90`. The only difference is the number of walkers is `2*int_col03_row01`. Similar to the situation with `demcmcELC.f90`, $N_{\rm walker} \approx 3N_{\rm var}$ is a good starting place for the number of walkers to use. You can insert one or more "elite" models into the initial population of walkers, see Section **8.5**.

The `SUFFIX` string noted in Section **8.3** starts at `1000000` and increments by `1` after each generation, exactly the same way as for `demcmcELC.f90`, see Section **8.4.2**. The `gridELC.inp` file noted in Section **8.3** will be renamed `ELC.SUFFIX` where `SUFFIX` is the counter for the generation where the current best model was computed, as described in Section **8.4.2**. Likewise, the `gridELC.opt` file noted in Section **8.3** will be renamed `gridELC.SUFFIX` where `SUFFIX` has the same meaning. You can use these files to see if any parameters are getting close to their respective lower or upper limits.

You can specify the initial population of walkers in exactly the same way as for `demcmcELC.f90`, see Section **8.4.2**. There is an optional input file called `hammerELC.inp` that is nearly the same as `demcmcELC.inp` described in Section **8.4.2**:

`aa` (real number)

gsig (real number)
rfracmod (real number)
numparm (integer)
ipurgestart (integer)
ipurgeint (integer)
purgefactor (real number)
istuck (integer)
target (real number)
itemp (integer)
tempstart (real number)
istart (integer)
iskip (integer)
ioldseed (integer)

aa gives the scale length parameter used to generate jumps for the walkers (this is the a parameter discussed in Foreman Mackey et al. 2013). Use **aa=2.0** in most cases. See Section **8.4.2** for detailed explanations of the last 12 parameters. The target acceptance fraction is usually around **target=0.3**. Tempering, if desired, works exactly the same way it does for **demcmcELC.f90**, see Section **8.4.2**.

In addition to the standard output files from the optimizers (Section **8.4.2**), **hammerELC.f90** will generation additional output files:

File name	contents
hammerELC.jumps	Contains jump statistics in three columns: generation number, jump interval of chain k, the index k.
hammerELC.out	Contains 5 columns: the generation number, the fraction of accepted jumps, the current value of **gamma**, the current value of the temperature, and an integer seed used to restart.
hammerELC_chi.SUFFIX	Similar to the file **chi.SUFFIX**, but for the models in the chains.
hammer_fitparm.SUFFIX	Similar to the file **generation.SUFFIX**, but for the models in the chains, and with the first two columns omitted.
gethammer_fitparmvals.com	A Unix shell script to make files containing the fitted parameter values and the χ^2 from the **hammer_fitparm.SUFFIX** files.
hammer_starparm.SUFFIX	Similar to the file **ELCparm.SUFFIX**, but for the models in the chains, and with the first two columns omitted.
gethammer_starparmvals.com	A Unix shell script to make files containing the derived parameter values and the χ^2 from the **hammer_starparm.SUFFIX** files.
hammer_dynparm.SUFFIX	Similar to the file **ELCdynparm.SUFFIX**, but for the models in the chains, and with the first two columns omitted. Needs **Nterms > 0** and **Nbody > 2**.
hammer_ratio.SUFFIX	Similar to the file **ELCratio.SUFFIX**, but for the models in the chains, and with the first two columns omitted. Needs **ifracswitch=1**.

`hammerELC` can be run in parallel, see Section **8.7**. See Section **8.9** for guidelines on how to get uncertainties on the fitted and derived parameters.

8.4.6 loopELC.f90

The program `loopELC.f90` will simply loop over the indicated free parameters in `gridloop.opt` and compute the χ^2 at each step. No optimization is done in this case. The values of `int_col03_row01`, `int_col03_row02`, etc. give the number of iterations in each loop. The `gridloop.opt` file might look like this:

```
none
Bdata
Vdata
none
Idata
none
none
none
RV1.dat
RV2.dat
none
none
none
4
mass ratio
inclination
t1
separ
2.0 0.5 4
55.0 0.25 5
6500.  200.  5
128.  10.  2
```

You will compute $4 \times 5 \times 5 \times 2 = 200$ light curves in four nested loops of 4, 5, 5, and 2 steps. The mass ratio will start at 2.0 and take on the values 2.0, 2.5, 3.0, and 3.5, and so on. The code will print the χ^2 value for each parameter set.

The `SUFFIX` string discussed in Section **8.3** is `0000`.

8.4.7 markovELC.f90

The program `markovELC.f90` is an optimizer program based on a simple Monte Carlo Markov Chain algorithm (Tegmark et al. 2004). For this algorithm you need an initial model—to specify this model put its parameters in `ELC.inp` and set `ielite=1`. You also need to specify the ranges of the free parameters, and you do this in the same way those ranges are specified in `demcmcELC.f90`, `geneticELC.f90`, or `hammerELC.f90`. Finally, you need to specify jump sizes for each parameter. The integers `int_col03_row01`, `int_col03_row02`, etc. give the initial number of steps for the jump:

$$s = (max - min)/N_{\rm step}.$$

Starting from the initial model, a parameter is selected at random and its value is offset by a "jump" and the χ^2 of the new model is computed. If the chi^2 at the proposed position is lower than the χ^2 at the starting place, the jump to the new model is taken and that parameter is added to the "chain". Otherwise the jump is accepted with a probability of $\exp(-\Delta\chi^2/2)$, after which the new parameter is added to the chain. This process repeats until a specified chain length is achieved.

There is an optional input file called **`markovELC.inp`** that contains three lines:

`lengthchain`
`Nchain`
`ireport`

`lengthchain` is the length a chain needs to be before it is considered "finished." Try something like **`lengthchain=30000`**.

`Nchain` is the total number of chains for each parameter. I usually use **`Nchain=30`**.

`ireport` is the "reporting" interval. As the program progresses, the lowest χ^2 value is kept. Every time a model with a lower χ^2 value is found, an internal counter is incremented by 1. All of the model files are output when the internal counter is equal to the value of **`ireport`**. After the "reporting" is finished, the internal counter is reset.

The **`SUFFIX`** string noted in Section **8.3** starts at **`1000000`** and increments by **`1`** after each chain is completed. The **`gridELC.inp`** file noted in Section **8.3** will be renamed **`ELC.SUFFIX`** where **`SUFFIX`** is the counter for the number of completed chains. Likewise, the **`gridELC.opt`** file noted in Section **8.3** will be renamed **`gridELC.SUFFIX`** where **`SUFFIX`** has the same meaning as above. You can use these files to see if any parameters are getting close to their respective lower or upper limits.

The internal step sizes of all of the parameters are adjusted so that the fraction of accepted jumps is $\approx 35\%$. The integers in the third column of numbers of **`gridELC.SUFFIX`** will be updated based on the current step size for each parameter.

After each chain is completed, you have output files of the form

File name	contents
`markovchain01.SUFFIX`	Contains the chain for the first free parameter for chain number `SUFFIX`.
`markovchain02.SUFFIX`	Contains the chain for the second free parameter for chain number `SUFFIX`.
.	
.	
.	
`markovchainNN.SUFFIX`	Contains the chain for the `NN`th free parameter for chain number `SUFFIX`, where `NN=Nvar`.
`markovchainparm.SUFFIX`	Contains the chains for the derived parameters for chain number `SUFFIX`. The order that the parameters are given in is the same as the order given in the `ELC.parm`, see Section **7.3**.

An attempt is made at trimming off the starting parts of the chains to allow for a "burn-in" phase. This attempt is sometimes not as successful as one wants on the initial chain. Also keep in mind that step sizes need to be tuned for each parameter, and if you have a lot of parameters, this process might take some time. Thus it is often desirable to omit the first chain when computing the posteriors. Given the limitations of `markovELC.f90`, if you want to use the poster samples to get uncertainties, it is often more efficient use either `demcmcELC.f90` or `hammerELC.f90` if you can get the parallel loops going (see Section **8.7**). On the other hand, `markovELC.f90` is often reasonably fast at finding decent models, so I often use `markovELC.f90` before using the other two codes.

8.4.8 mqELC.f90

The program `mqELC.f90` is an optimizer program based on the Levenberg-Marquardt (see Marquardt 1963) routine given in Bevington (1969). The format of `gridloop.opt` is similar to that of `amoebaELC.f90` and `gridELC.f90`.

The `SUFFIX` string noted in Section **8.3** is `0000`.

The code frankly is not that good, and it can be inefficient when the number of data points being fitted is large. A better version is under development.

8.4.9 nestedELC.f90

The program `nestedELC.f90` is an optimizer based on the nested sampling technique outlined by Skilling (2006). This algorithm is a method to compute the Bayesian evidence

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$$

where $\mathcal{L}$ is the likelihood and $\pi(\theta)$ is the prior. We can also compute samples from the posterior distribution

$$\mathcal{P} = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}}.$$

To use this code, you need to define the lower and upper bounds of all of the free parameters. Set up the **gridloop.opt** file in the same way as for of **demcmcELC.f90**, **geneticELC.f90**, **hammerELC.f90**, or **markovELC.f90**. The value of **int_col03_row01** gives the length of the main sampling loop ($N_{\rm sample}$). This loop can be run in parallel (see Section **8.7**), so set **int_col03_row01** to be some multiple of the number of cores or threads available. Next, you need a certain number of "live" points, and this is set by the value of **int_col03_row02**. At a minimum, you should have $N_{\rm live} \geq 2N_{\rm var}$. Finally, set **int_col03_row03** to some very large integer $N_{\rm big}$ (100000 is usually sufficient).

There is an optional input file called **nestedELC.inp**. This file has four lines:

tolthresh (real number)
selthresh (real number)
iskipvol (integer)
iskiphammer (integer)

The meanings and uses of these parameters will be explained below.

At the start of the program, the parameter ranges are normalized onto the unit hyper-cube. Then the set of live points is initially filled with random models, and the χ^2 values are found for each live point. The smallest hyper-ellipsoid that encloses the live points is found, and its volume is computed. We then look for a random point selected from the hyper-ellipsoid with a χ^2 value that is better than the live point with the largest χ^2. Once we find such a point, we replace the worst live point with the newly found point, and the replaced live point goes into the "dead points" list. We then recompute the bounding hyper-ellipsoid. The volume of this ellipsoid is scaled by a factor that is related to the number of live points and the number of replacements found to that point (see Skilling 2006). Over time, the volume of the bounding hyper-ellipsoid shrinks by a set rate, and the χ^2 values of the live points get smaller and smaller. The sampling is continued until some stopping condition is met. A posterior sample is constructed using the live points and the dead points.

One can imagine $\mathcal{Z}$ being the volume of a "hill" in the parameter space. Initially, the random live points are near the base of the hill where the likelihood is low (χ^2 is large). As live points are replaced, the likelihood of the remaining points increases, and the volume of the hill is being numerically built up. Eventually it gets harder and harder to add to the integral $\mathcal{Z}$ since the remaining volume gets smaller and smaller. The code has a tolerance parameter that is an estimate of the remaining volume to cover.

Once the process is finished, we have a set of dead points (arranged by increasing likelihood) and a set of live points. These points can be assigned weights based in the likelihood of that point and the statistical volume of the hyper-ellipsoid from which it was drawn. To get a posterior sample, the weights can be converted into a probability of being selected. At the beginning, the probability is essentially zero since the likelihood is very small (χ^2 is very large). The probability of being selected into the posterior sample increases and reaches a maximum where there is a combination of a high likelihood and a decent-sized volume from which the point came. After the maximum selection probability, subsequent points have higher likelihoods, but their overall weights get smaller owing to the ever shrinking volume of the prior space from which they are drawn. Thus the probability to be selected for the posterior sample starts to get less and less.

When should this process stop? One might think that the value of $\mathcal{Z}$ should be made as accurate as possible, which in turn means the tolerance parameter should be small (say less than 0.001, which means 999/1000 of the "hill" has been explored). On the other hand, as process converges (the likelihood values of the live points gets higher and higher), it becomes harder and harder to find replacement points. In addition, the probably of selecting each new dead point for the posterior sample becomes smaller and smaller. One reaches a point where one has to work harder and harder to get another point in the posterior sample.

The code uses the parameters **`tolthresh`** and **`selthresh`** to determine when to stop. The default values of **`tolthresh=0.01`** and **`selthresh=0.05`** seem like good values to start with. There is an continuously updated output file called **`nestedELC.out`** with some useful information. This file has 11 columns:

1. iteration number
2. model number
3. dead point number
4. "hammer" iteration count (see below)
5. the statistically scaled volume of the hyper-ellipsoid
6. the actual volume of the ellipsoid (the volume of the unit hyper-cube is 1)
7. the smallest χ^2 found so far
8. the largest χ^2 among the current live points
9. $\log \mathcal{Z}$
10. the tolerance
11. the selection probability

The selection probability depends on the maximum weight of all of the dead points, and this value won't be known until the code has been running for a while. As a result, the selection probabilities listed in **`nestedELC.out`** for points near the start will be around 1.0. Eventually it becomes obvious when the peak probability has been passed since the selection probability (along with the tolerance) will start to drop rapidly.

You will notice that the statistically scaled volume of the hyper-ellipsoid is usually several orders of magnitude larger than the actual volume. This statistical scaling is necessary if one wants to compute $\mathcal{Z}$ and the posterior sample properly. If one wants to simply find some good models rather quickly, the statistical scaling of the hyper-ellipsoid volume can be turned off. To do this, set the parameter **`iskipvol=1`**. When the hyper-ellipsoid has a smaller volume, replacement points are easier to find. Thus, better models are found more rapidly. As noted above, you can't get a proper posterior sample from the dead points.

In the simplest implementation of the algorithm, one starts with random live points, and their likelihoods are found. Then one chooses a random point and computes its likelihood. If the likelihood is better than the worst live point, that live point is replaced. If not, another random point is chosen and the process is repeated. We can speed up the process by running a parallel loop of length $N_{\rm sample}$ (for maximum efficiency, make $N_{\rm sample}$ a multiple of the number of threads available). Instead of computing a single model and comparing its χ^2 to the worst live point, we compute $N_{\rm sample}$ points and compare the χ^2 of the *best* one to the *worst* live point. If a replacement live point is not found, the parallel loop is run again until a replacement point is found.

We can increase the speed even more. If the initial parallel loop of $N_{\rm sample}$ points does not have a replacement point, we use the EMCEE Hammer algorithm to find the next set of $N_{\rm sample}$ points (as opposed to choosing them randomly). The Hammer algorithm turns out to be about three times faster at finding replacement points than randomly choosing them. If you want to disable the Hammer algorithm, set **`iskiphammer=1`**. The default value is **`iskiphammer=0`**, which causes the code to switch over to the hammer algorithm after the first initial random loop.

The code works well when $N_{\rm var} \lesssim 10$, but the efficiency drops dramatically as the number of free parameters grows past this point. Here are some suggestions to speed things up:

1. More live points means a longer execution time. On the other hand, fewer live points results in a smaller posterior sample. You need $N_{\rm live} = 2N_{\rm var}$ at a bare minimum. Set the number of live points to something close to or just above the minimum and run the code several times in different subdirectories. Be sure to change the value of **`jdum`** for each run (this changes the initial random number seed).

2. You want the size of the parallel sampling loop to be on the order of a few times to several times the number of fitting parameters. Recall that the Hammer algorithm is two or three times faster at finding replacement live points than randomly choosing points. If you have a really large value of $N_{\rm sample}$, then you are waiting longer to switch over to the Hammer algorithm. On the other hand, if you have a computer with a decent number of threads, it won't hurt to make the sampling loop that size.

3. Larger priors mean larger execution times. Try to set sensible ranges.

4. If you have a multibody system and/or light curves in several bandpasses, the number of fitting parameters can become quite large (e.g. Kepler-47 needs over 50 free parameters). In many cases, the limb darkening parameters are not that well constrained and frankly are not that interesting. On the other hand, you just can't choose them randomly and expect to get a decent model fit. You can run **`markovELC`**, **`demcmcELC`**, **`hammerELC`**, etc. to get some initial fits, and these can be refined using **`amoebaELC`**. Once you get a good fit, fix the limb darkening parameters and the contamination parameters (if working with *Kepler* data) at their best values and remove those parameters from the list of fitting parameters. Based on these initial runs, make the priors smaller. With the reduced number of free parameters, set the number of live points to near the minimum value and run **`nestedELC`** several times in separate directories.

The "helper" code to obtain a posterior sample once the code has converged is described below.

8.4.10 onestepELC.f90

Code under development.

8.4.11 randomELC.f90

The program **`randomELC.f90`** will simply create random parameter sets over the range indicated parameters and compute the χ^2 for each model. No optimization is done in this case. Use the **`gridloop.opt`** file to set the lower and upper ranges of the free parameters in a similar manner to that of **demcmcELC.f90**, **geneticELC.f90**, **hammerELC.f90**, or **markovELC.f90**. Then set the value of **int_col03_row01** to be the total number of models you want to compute, for example **int_col03_row01=100000**. Set **int_col03_row02** to be the number of top-ranking models found that will be saved. For example, if **int_col03_row02=2**, then the models with the lowest overall χ^2 and the second lowest χ^2 will be saved.

The **SUFFIX** string noted in Section **8.3** is `0000`.

8.4.12 stepELC.f90

This is a version of **gridELC.f90** where the allowed ranges of the free parameters are bounded. In addition, the code can run in parallel (see Section **8.7**). To use this code, you need a starting model. Put the parameters of this model into **ELC.inp**. Next, you need to define the lower and upper bounds of all of the free parameters. Set up the **gridloop.opt** file in the same way as for **demcmcELC.f90**, **geneticELC.f90**, **hammerELC.f90**, or **markovELC.f90**. Then set the value of **int_col03_row01** to be the number of iterations needed for the first free parameter, the value of **int_col03_row02** to be the number of iterations needed for the second free parameter, and so on. Usually 3 to 6 iterations works well. Next, create a file called **stepELC.inp** with these two lines:

```
Niter
Nloop
```

The value of **Niter** should be even. For each parameter $a(i)$, divide up the range into two intervals from $a_{\rm min}(i)$ to $a(i)$, and from $a(i)$ to $a_{\rm high}(i)$. Divide each of those intervals into $N_{\rm iter}/2$ steps, set $a(i)$ to the value at each of those steps, and compute the χ^2 for each step. Then find the step with the lowest χ^2, and the points on either side. Divide those two intervals into $N_{\rm iter}/2$ steps, and repeat the process. Continue for the number of iterations given by **int_col03_rowii**, where **ii** is the index i. Figure 2 illustrates how the parameter intervals are divided.

This process of dividing the parameter range is repeated for each parameter, and after the **Nvar** parameters have been iterated on, the best over model is put in place of the original one. The loop over the parameters is repeated until there are a total of **Nloop** runs through all **Nvar** free parameters.

The **Niter** steps at the subdividing can run in parallel (section **8.7**), so set **Niter** based on the number of CPU cores or threads available. As one can see from Figure 2, there are a fair number of points with χ^2 values far from the minimum. You want enough divisions to sample the curve properly, but on the other hand, too many steps usually results in extra time spent sampling regions of high χ^2. It is usually more efficient (in terms of total number of models needed) to have a smaller value of **Niter** and slightly more repeats of the subdividing loop. **Niter=20** seems to be a good place to start, as that gives you 10 divisions on either side of the initial model. After 4 iterations of the subdividing loop, you have a resolution of $\approx 10^{-4}$.

8.5 Inserting good models into the initial population

When using **markovELC.f90**, you have the option of starting at a random point in parameter space (usually not recommended), or starting at a point with a reasonably good χ^2. To start

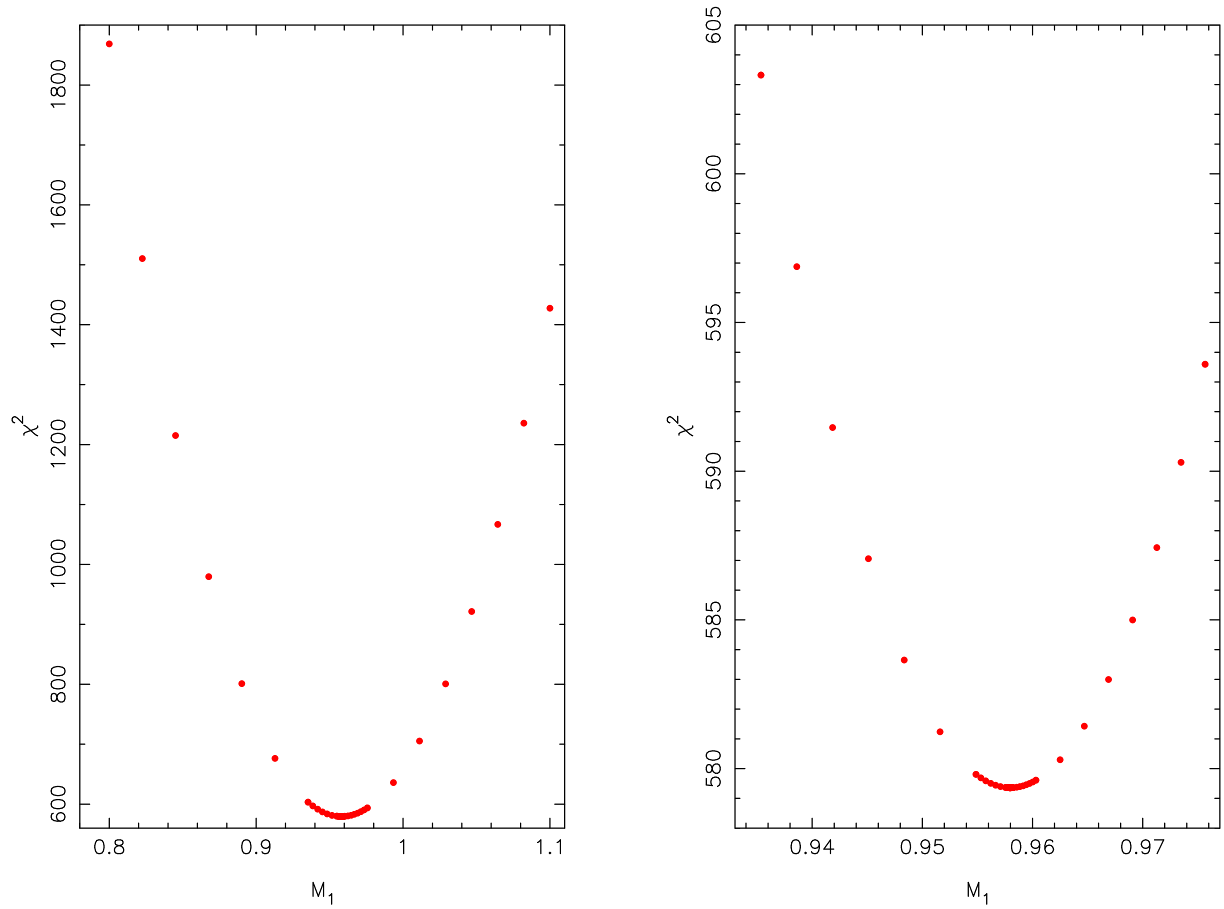

Figure 2: An illustration of how the parameter interval is subdivided in **stepELC**. The x-axis is the parameter (**primmass** in this case, with a range of $0.8\,M_\odot$ to $1.1\,M_\odot$), and the y-axis is χ^2. The number of iterations in the parallel loop was **Niter=16**. The plot on the left shows the entire parameter range, and the plot on the right zooms on the region near the χ^2 minimum. In this case the plot is pretty self-similar.

with a user-specified model, set **ielite=1** in **ELC.inp**. **markovELC.f90** will start its exploration of parameter space at the model given by the parameters in **ELC.inp**.

When using **demcmcELC.f90**, **geneticELC.f90**, or **hammerELC.f90**, you have the option of using more than one "elite" model. When **ielite=1**, the parameters for the first elite model are taken from **ELC.inp**. When **ielite > 1**, we need additional input files, and these additional input file are called

ELC.1001 (when **ielite=2**)
ELC.1002 (when **ielite=3**)
ELC.1003 (when **ielite=4**)
.
.
.

`ELC.1NNN` (when `ielite=NNN+1`)

Each of these extra files have the same exact format as **ELC.inp**. To help you make these extra files, there is a code called **makeELCelite.f90**. The usual procedure is to run an optimizer for a while in a given subdirectory. Gather all of the **generation.SUFFIX** files and select the better models (you can sort on the second column, which is the χ^2 for the model with the indicated parameters). Put the model parameters into a file called **generation.all**, with one model per line. Then run **makeELCelite.f90**. You will have N files with the appropriate names, where N is the number of lines in the file you input to **makeELCelite.f90**. This helper code will read **ELC.inp**, **gridloop.opt**, and **ELC.optimizer**, so don't alter these files.

If you ran either **demcmcELC.f90** or **hammerELC.f90**, you can use **makeELCelite.f90** to read the **demcmc_fitparm.SUFFIX** or the **hammer_fitparm.SUFFIX** files. Thus you can evolve some number of walkers using **hammerELC.f90** and use those models as the initial population for **demcmcELC.f90**, or the other way around.

8.6 Restarting certain optimizers

If a run of **demcmcELC.f90** was stopped, you can continue that run by doing the following. Find the **demcmc_fitparm.SUFFIX** file with the largest value for the **SUFFIX**. Set the **istart** parameter in **demcmcELC.inp** file to **SUFFIX - 1000000 - 1**. For example, if the largest **SUFFIX** was **1000192**, set **istart = 191**. Run **makeELCelite.f90** and enter the *second to the last* **demcmc_fitparm.SUFFIX** file (**demcmc_fitparm.1000191** in the previous example) at the prompt. Edit the **ELC.inp** and set **ielite** to $N_{\rm pop}$ (the number of chains). Then, set the value of **gamma** in **demcmcELC.inp** to the number given in the third column of the last line of **demcmcELC.out**. Finally, set the value of **ioldseed** to the value in the fifth column of the last line of **demcmcELC.out**. In the example above, you would use the fifth column of index 192 in the **demcmcELC.out** file. When you restart **demcmcELC.f90**, it will (more or less) pick up where it left off. That is, the **demcmc_fitparm.SUFFIX** and the **demcmc_starparm.SUFFIX** files will be (nearly) the same. There may be some small round-off error that creeps in at some point, but the results should be very close in a statistical sense. Note the **generation.SUFFIX** file and the **chi.SUFFIX** file at the restart point (e.g. **SUFFIX - 1**) won't be the same as it would have been in an uninterrupted run.

You can also restart a run of **hammerELC.f90** by following the steps given above (you will obviously need to use the latest **hammer_fitparm.SUFFIX** file rather than the **demcmc_fitparm.SUFFIX** file, and also look at the third and fifth columns of the last line of the **hammerELC.out** file rather than the **demcmcELC.out** file). Finally, set **ielite** to the number of walkers, which is **2*int_col03_row01**.

You can restart a run of **geneticELC.f90** by putting the generation number of the last completed generation into a file called **geneticELC.inp**.

8.7 Running certain optimizers in parallel

The codes **demcmcELC.f90**, **geneticELC.f90**, **hammerELC.f90**, **nestedELC.f90**, **randomELC.f90**, and **stepELC.f90** can be run in on computers with multiple CPU cores or threads and shared memory. To use the parallel capabilities, add the compiler flag **-mp** to the list when using the

PGI compilers on a Linux system:

```
pgf90 -mcmodel=medium -O2 -mp -o hammerELC_par hammerELC.f90
```

If you are using gfortran, use

```
gfortran -mcmodel=medium -fopenmp -O2 -o demcmcELC_par demcmcELC.f90
```

You need to set some environment variables run the codes in parallel on a Linux system. I have put these commands in my `.cshrc` file, where my shell is the tcsh:

```
limit stacksize unlimited
setenv OMP_NUM_THREADS 40
setenv OMP_THREAD_LIMIT 88
setenv OMP_SCHEDULE dynamic
setenv OMP_STACKSIZE 12000000
```

These commands will vary slightly with the number of CPU or threads your computer has. For maximum efficiency, it is helpful to have the parallel loop in a given code be an integer multiple of the number of threads used (given by `OMP_NUM_THREADS`). The lengths of the parallel loops in each code are given in the following table:

Code	Loop specifier
`demcmcELC.f90`	The value of `int_col03_row01` in `gridloop.opt`, which is the number of chains.
`geneticELC.f90`	The value of `int_col03_row01` in `gridloop.opt`, which is the number of "organisms" in the population.
`hammerELC.f90`	The value of `int_col03_row01` in `gridloop.opt`, which is *half* the number of walkers.
`nestedELC.f90`	The value of `int_col03_row01` in `gridloop.opt`, which is the length of the sampling loop.
`randomELC.f90`	The value of `int_col03_row01` in `gridloop.opt`, which is the total number of random models to compute.
`stepELC.f90`	The value of `Niter`, which is the first line of the file `stepELC.inp`.

If all goes well, the codes will run in parallel. The screen output will come in a scrambled order. Also, the output files `chi.SUFFIX`, `generation.SUFFIX`, `ELCparm.SUFFIX`, `ELCdynparm.SUFFIX` (if present), and `ELCratio.SUFFIX` (if present) will likewise be in a scrambled order. The scrambling is the same for all files with the same `SUFFIX` and is usually not a problem.

If you get a "segmentation fault" error, try increasing the value of `OMP_STACKSIZE`, or reducing the number of threads (`OMP_NUM_THREADS`), or both.

I have not been able to get any of these codes to run in the parallel mode on my MacBook Pro laptop, owing to issues with the Xcode and the availability of the `mcmodel=medium` option.

I have a prototype code that can run on several different computers at once using MPI libraries. This is still under development.

8.8 Helper codes

The following "helper" codes can be used with the various optimizer codes. You will need the `PGPLOT`[3] libraries, the **`convert`** command from Imagemagick[4], and the **`gs`** command. The latter two codes seem to be pretty standard on Linux systems.

You can use gfortran to compile these codes:

```
gfortran -fno-backslash -o ELCplottrend ELCplottrend.f90 -L /usr/X11R6/lib64/ -lX11
-L /usr/local/pgplot -lpgplot -Xlinker -R -Xlinker /usr/local/pgplot -lpgplot
```

(this command is all on one line). The locations of the various libraries might be slightly different on your system, so adjust accordingly. **Note the code `ELCchiplotter.f90` may require the `-mcmodel=medium` compiler flag.** Since the `PGPLOT` libraries are in single precision, some of the plots may look strange when the variations in the parameters are not resolved in single precision. This usually happens when plotting the orbital period or the times of conjunction for the binary.

The details of the codes are presented in the following sections. Most of the codes have optional command-line arguments, these will be given in curly braces {}.

8.8.1 ELCchiplotter.f90

This code will help you make plots of various parameters vs. the χ^2 values. It should work with any of the optimizers. To use, issue this command:

```
ELCchiplotter {thresh}
```

where **`thresh`** is largest χ^2 value above the minimum you want to show (the default is **`thresh=50.0`**). Thus a typical command might look like

```
ELCchiplotter 30.0
```

or

```
ELCchiplotter
```

to use the default χ^2 threshold. By default, this code will make plots for all of the fitting parameters and any extra parameters, such as the binary eccentricity and argument of periastron. An example plot is shown in Figure 3. If you also want plots for the derived parameters, create a file called **`ELCcolumns.inp`**. Copy the lines from the **`ELC.parm`** file that have the parameters you are interested in. For example, here is my **`ELCcolumns.inp`** file for Kepler-47:

```
3 0.95816336 mass of star 1 in solar masses
4 0.34191998 mass of star 2 in solar masses
5 0.93663960 radius of star 1 in solar radii
6 0.33900607 radius of star 2 in solar radii
7 4.47636 log(g1) cgs
```

[3] http://www.astro.caltech.edu/~tjp/pgplot/

[4] https://imagemagick.org/script/index.php

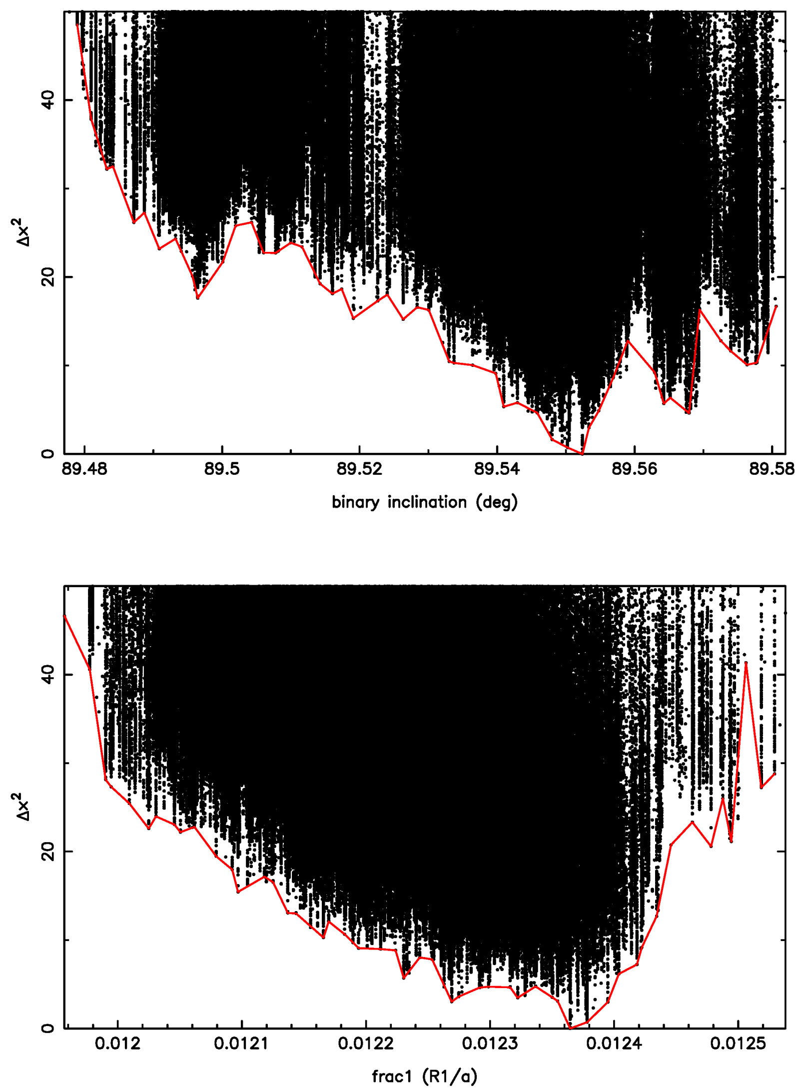

Figure 3: An example page from **ELCchiplotter.f90**. The entire plot may have several pages, and you will see two panels per page.

```
8 4.91158 log(g2) cgs
13 31.30501 K-velocity star 1 in km/sec
14 87.72611 K-velocity star 2 in km/sec
35 0.081464 semimajor axis of binary orbit in AU
47 2.13754 mass of body 3 in Earth masses
48 3.04734 radius of body 3 in Earth radii
50 0.4151 density of body 3 in g/cc
52 0.287817 semimajor axis of body 3 orbit in AU
53 0.0210328 eccentricity of body 3 orbit
54 48.40127 argument of periastron of body 3 orbit in degrees
59 0.1653710 mutual inclination of body 3 orbit in degrees
62 13.40875 mass of body 4 in Earth masses
63 7.04857 radius of body 4 in Earth radii
65 0.2104 density of body 4 in g/cc
67 0.699406 semimajor axis of body 4 orbit in AU
```

```
68 0.0236274 eccentricity of body 4 orbit
69 351.14949 argument of periastron of body 4 orbit in degrees
74 1.1642520 mutual inclination of body 4 orbit in degrees
77 2.78418 mass of body 5 in Earth masses
78 4.65963 radius of body 5 in Earth radii
80 0.1512 density of body 5 in g/cc
82 0.964036 semimajor axis of body 5 orbit in AU
83 0.0526027 eccentricity of body 5 orbit
84 300.51382 argument of periastron of body 5 orbit in degrees
89 1.3834716 mutual inclination of body 5 orbit in degrees
```

Note that only the integers in the left column are actually used, so you can omit the middle column of numbers and the verbose descriptions on the right.

This code uses **ELC.inp**, **gridloop.opt**, and **ELC.optimizer**, so don't alter these files. The tags in **gridloop.opt** will be converted into axis labels for the plots. You will have a lot of files of the form **ELCjunk.TAG_STRING**, where **TAG_STRING** is a suffix generated from a tag in **gridloop.opt** or a suffix associated with a derived parameter. For example, here are the files I have for Kepler-47 when using the **ELCcolumns.inp** given above:

```
ELCjunk            ELCjunk.P2esin     ELCjunk.Tperi2       ELCjunk.ecc5
ELCjunk.K1         ELCjunk.P2incl     ELCjunk.Tsupconj2    ELCjunk.ecos
ELCjunk.K2         ELCjunk.P2period   ELCjunk.a2AU         ELCjunk.esin
ELCjunk.M1solar    ELCjunk.P2ratrad   ELCjunk.a3AU         ELCjunk.frac1
ELCjunk.M2solar    ELCjunk.P2tconj    ELCjunk.a4AU         ELCjunk.frac2
ELCjunk.M3earth    ELCjunk.P3Omega    ELCjunk.a5AU         ELCjunk.gam1
ELCjunk.M4earth    ELCjunk.P3ecos     ELCjunk.argper2deg   ELCjunk.i
ELCjunk.M5earth    ELCjunk.P3esin     ELCjunk.argper3deg   ELCjunk.log_g1
ELCjunk.P          ELCjunk.P3incl     ELCjunk.argper4deg   ELCjunk.log_g2
ELCjunk.P1Omega    ELCjunk.P3period   ELCjunk.argper5deg   ELCjunk.mutual3deg
ELCjunk.P1Q        ELCjunk.P3ratrad   ELCjunk.contamS0     ELCjunk.mutual4deg
ELCjunk.P1ecos     ELCjunk.P3tconj    ELCjunk.contamS1     ELCjunk.mutual5deg
ELCjunk.P1esin     ELCjunk.Q          ELCjunk.contamS2     ELCjunk.out
ELCjunk.P1incl     ELCjunk.R1solar    ELCjunk.contamS3     ELCjunk.primmass
ELCjunk.P1period   ELCjunk.R2solar    ELCjunk.den3         ELCjunk.temprat
ELCjunk.P1ratrad   ELCjunk.R3earth    ELCjunk.den4         ELCjunk.x1_U
ELCjunk.P1tconj    ELCjunk.R4earth    ELCjunk.den5         ELCjunk.x2_U
ELCjunk.P2Omega    ELCjunk.R5earth    ELCjunk.ecc2         ELCjunk.y1_U
ELCjunk.P2Q        ELCjunk.Tconj      ELCjunk.ecc3         ELCjunk.y2_U
ELCjunk.P2ecos     ELCjunk.Tinfconj2  ELCjunk.ecc4
```

These files contain two columns of numbers: the χ^2 and the value of the parameter given by the suffix (most, if not all of these suffixes should be self-explanatory). You can use any plotting program to make your own plots if desired.

8.8.2 ELCcorplotter.f90

This code will plot the correlations between the various parameters. This mainly for use with **demcmcELC.f90**, **hammerELC.f90**, although it should work with **geneticELC.f90** and **nestedELC.f90**. To use, issue this command:

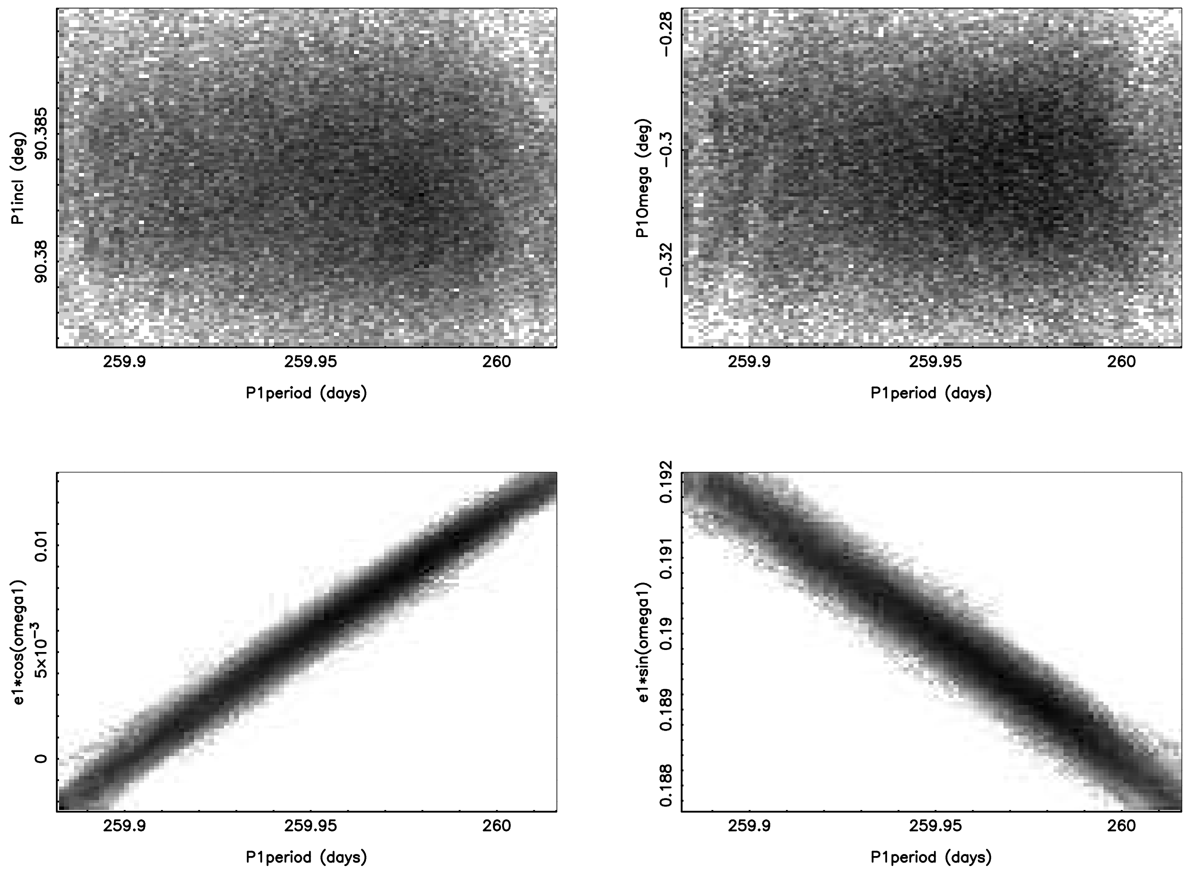

Figure 4: An example page from **ELCcorplotter.f90**. The entire plot may have several pages, and you will see four panels per page.

ELCcorplotter {**istart**} {**Nskip**} {**istop**} {**Nbin**} {**clip**}

istart is the starting generation (default is **istart=500**), make sure this allows for a sufficient "burn in" phase.

Nskip is the number of skips between samples of the chains (default is **Nskip=30**). Since the chains do not jump every generation, you want to skip some generations between samples.

istop is the stopping generation (default is **istop=99999**).

Nbin is the number of bins in the x- and y-directions in the plots (default is **Nbin=200**).

clip will attempt to clip out outliers. This feature is not yet implement so use **clip=0.0** or leave it out of the command line. Thus a typical command might look like

ELCcorplotter 5000 100 100000

This will use the default binning.

After this command is run, you will see plots of everything vs. everything else, with four gray-scale plots per page, see Figure 4 for an example page. In that example, we see the inclination of the third body orbit vs. the third body period, the nodal angle of the third body orbit vs. the third body period, $e\cos\omega$ for the third body orbit vs. the third body period, and $e\sin\omega$ for the third body orbit vs. the third body period. One can see that the latter two quantities are strongly correlated with the period of the third body orbit.

The code will create several files of the form **ELCjunk.TAG_STRING**, with each **TAG_STRING** associated with each tag in **gridloop.opt** or each derived parameter. For example, here are the files I have from KIC 10753734:

```
ELCjunk             ELCjunk.Tperi2      ELCjunk.omega1     ELCjunk.x1_U  ELCjunk.y1_K
ELCjunk.M1solar     ELCjunk.Tsupconj2   ELCjunk.omega2     ELCjunk.x1_V  ELCjunk.y1_R
ELCjunk.P           ELCjunk.argper2deg  ELCjunk.primmass   ELCjunk.x2_B  ELCjunk.y1_U
ELCjunk.P1Omega     ELCjunk.chi         ELCjunk.primrad    ELCjunk.x2_H  ELCjunk.y1_V
ELCjunk.P1Q         ELCjunk.contamS0    ELCjunk.ratrad     ELCjunk.x2_I  ELCjunk.y2_B
ELCjunk.P1ecos      ELCjunk.contamS1    ELCjunk.rk1        ELCjunk.x2_J  ELCjunk.y2_H
ELCjunk.P1esin      ELCjunk.contamS2    ELCjunk.rk2        ELCjunk.x2_K  ELCjunk.y2_I
ELCjunk.P1incl      ELCjunk.contamS3    ELCjunk.temprat    ELCjunk.x2_R  ELCjunk.y2_J
ELCjunk.P1period    ELCjunk.ecc2        ELCjunk.x1_B       ELCjunk.x2_U  ELCjunk.y2_K
ELCjunk.P1ratrad    ELCjunk.ecos        ELCjunk.x1_H       ELCjunk.x2_V  ELCjunk.y2_R
ELCjunk.P1tconj     ELCjunk.esin        ELCjunk.x1_I       ELCjunk.y1_B  ELCjunk.y2_U
ELCjunk.Q           ELCjunk.gam1        ELCjunk.x1_J       ELCjunk.y1_H  ELCjunk.y2_V
ELCjunk.Tconj       ELCjunk.gam2        ELCjunk.x1_K       ELCjunk.y1_I  ELCjunkchi
ELCjunk.Tinfconj2   ELCjunk.i           ELCjunk.x1_R       ELCjunk.y1_J
```

These files each contain a single column that can be taken to be a posterior sample of the parameter in question.

8.8.3 ELCgelmanrubin.f90

This code will compute the Gelman-Rubin (1992) statistic for the chains from **demcmcELC.f90** and **hammerELC.f90**. To use, issue this command:

ELCgelmanrubin {istart} {Nskip} {istop}

istart is the starting generation (default is **istart=500**), make sure this allows for a sufficient "burn in phase.

Nskip is the number of skips between samples of the chains (default is **Nskip=30**). Since the chains do not jump every generation, you want to skip some generations between samples.

istop is the stopping generation (default is **istop=99999**).

The results will be in a file called **ELCgelmanrubin.out**. There is no hard rule on what this statistic should be, but the closer to 1.0, the better.

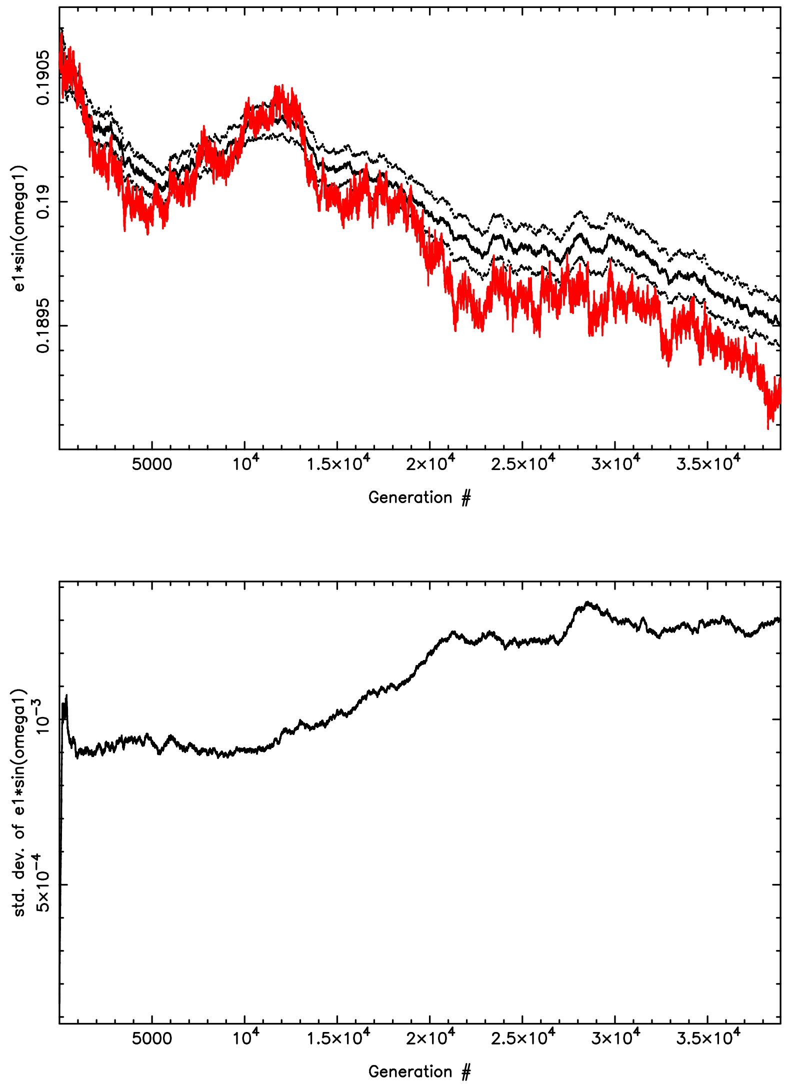

Figure 5: An example page from **ELCtrendplotter.f90**. The entire plot may have several pages, and you will see two panels per page, with the top plot showing the mean of the parameter and its 1σ errors (black lines) and the median of the parameter (red line) vs. the parameter number, and the bottom plot showing the standard deviation of the parameter vs. the generation number.

8.8.4 ELCplottrend.f90

This code is mainly for use with **demcmcELC.f90** and **hammerELC.f90**. It will plot the average and median values of all of the fitting parameters at each generation. As the chains converge and reach a steady state, the mean and median values should not change much as one adds additional generations. To use, issue this command:

ELCplottrend {**istart**} {**istop**} {**Nskip**}

istart is the starting generation (default is **istart=1**), make sure this allows for a sufficient "burn in" phase.

istop is the stopping generation (default is **istop=99999**).

Nskip is the number of skips between samples of the chains (default is **Nskip=1**).

Figure 5 shows an example page. The top plot will be the mean of the parameter and its 1σ errors (black lines) and the median of the parameter (red line). The bottom plot will show the standard deviation of the parameter. In this case we see the trend in $e \sin \omega$ for the third body orbit for a long run of **demcmcELC.f90**. The trend is not flat yet, and the median value seems to be somewhat different than the mean value.

8.8.5 ELCposterior.f90

This program will extract posterior samples for all of the fitting and (optionally) derived parameters of choice from runs of **demcmcELC.f90** and **hammerELC.f90**. To use, issue this command:

ELCposterior {**istart**} {**Nskip**} {**istop**} {**Nbin**} {**clip**}

istart is the starting generation (default is **istart=500**), make sure this allows for a sufficient "burn in" phase.

Nskip is the number of skips between samples of the chains (default is **Nskip=30**). Since the chains do not jump every generation, you want to skip some generations between samples.

istop is the stopping generation (default is **istop=99999**).

Nbin is the number of bins in the x-direction in the plots (default is **Nbin=200**).

clip will attempt to clip out outliers (default is **clip=0.0**). Suggested values, when needed, range from 4.0 to 6.0

```
ELCposterior 5000 100 100000 50
```

This will use 50 bins, and will omit clipping.

The plot will have several pages, and an example of a page is shown in Figure 6. The results will be shown in the file **ELCposterior.out**. This file will have 7 columns showing the parameter mode, the parameter median, the $+1\sigma$ error bar, the -1σ error bar, the 1σ lower bound of the parameter, the 1σ upper bound of the parameter, and the description of the parameter.

The posterior samples for all of the fitted and derived parameters selected in the **ELCcolumns.inp** file will appear in files called **ELCjunk.TAG_STRING**. The computed frequency distributions (a.k.a. these histograms) will appear in files called **hist.TAG_STRING**.

8.8.6 ELCnestedposterior.f90

This program will extract posterior samples for all of the fitting and (optionally) derived parameters of choice from runs of **nestedELC.f90**. To use, issue this command:

ELCnestedposterior {**Nbin**} {**Nsample**}

Nbin is the number of bins in the x-direction in the plots (default is **Nbin=100**).

`Nsample` is the number of times that the list of dead points is sampled (default is `Nsample=6`).

The histograms will be plotted in a file similar to that made by `ELCposterior.f90`. Likewise the posterior samples for all of the fitted and derived parameters selected in the `ELCcolumns.inp` file will appear in files called `ELCjunk.TAG_STRING`. The computed frequency distributions (a.k.a. these histograms) will appear in files called `hist.TAG_STRING`.

8.8.7 ELCprogplotter.f90

This codes lets you visually see the "progress" of `demcmcELC.f90`, `geneticELC.f90`, `hammerELC.f90`, and `nestedELC.f90`. To use, issue this command:

`ELCprogplotter {istart}`

`istart` is the starting generation (default is `istart=1`).

The output plot will have several pages, and each page features two panels, see Figure 7 for an example page. The top panel shows the parameter value on the x-axis and the generation value on the y-axis. The bottom panel shows the frequency distribution of the parameter shown in the top panel.

8.8.8 ELCspreadchiplotter.f90

This program plots the χ^2 values vs. the generation number for `demcmcELC.f90` and `hammerELC.f90`. To use, issue this command:

`ELCspreadchiplotter {istart} {istop} {ichain01,ichain02,ichain03,...}`

`istart` is the starting generation (default is `istart=1`).

`istop` is the stopping generation (default is `istop=99999`).

The last argument is a list of integers separated by commas that list the numbers of the chains you wish to highlight. For example

```
ELCspreadchiplotter 1 400 3,4
ELCspreadchiplotter 400 1400 3,4
ELCspreadchiplotter 1 2000 27,44,2
```

Example plots are shown in Figures 8,9, and 10. Eventually the chains seems to reach some kind of steady state where the χ^2 values overall don't get any better.

8.8.9 ELCspreadparmplotter.f90

This program plots parameters values for a given parameter vs. the generation number for `demcmcELC.f90` and `hammerELC.f90`. To use, issue this command:

```
ELCspreadparmplotter {type} {tag} {istart} {istop} {ichain01,ichain02,ichain03,...}
```

type is either **fit** for a fitted parameter, **parm** for a derived parameter, or **flux** for a flux ratio (need **ifracswitch=1**).

tag is either a two letter tag from **gridloop.opt** if the **type** was **fit**; an integer giving the index of a derived parameter given in **ELC.parm** (see page 48) if the **type** was **parm**; or an integer giving the index of the flux ratio if the **type** was **flux** (1 through 8 for star 2 to star 1 flux ratios; 9 through 16 for disk fractions, and 17 through 24 for star 3 to star 1 flux ratios).

istart is the starting generation (default is **istart=1**).

istop is the stopping generation (default is **istop=99999**).

The last argument is a list of integers separated by commas that list the numbers of the chains you wish to highlight. For example

`ELCspreadparmplotter fit in 1 400 3,4`	(plots inclination)
`ELCspreadparmplotter fit oc 400 1400 66,14`	(plots $e\cos\omega$)
`ELCspreadparmplotter parm 3 1 2000 27,44,2`	(plots M_1 in $M_\odot$)
`ELCspreadparmplotter parm 77 150 550 75`	(plots M_5 in $M_\oplus$)
`ELCspreadparmplotter fit 3 150 550 17,33`	(plots flux ratio in filter 3)

Figure 11 shows an example plot. The colors for the highlighted chains are red for the first chain, green for the second chain, blue for the third chain, cyan for the fourth chain, magenta for the fifth chain, and yellow for the sixth chain.

8.9 Finding the uncertainties on the fitted and derived parameters

With all of the optimizers available, finding something close to the "best" model is a relatively straightforward (but not necessarily quick) process. Finding realistic uncertainties on the fitted and derived parameters is another thing entirely. We have two main ways of finding the uncertainties: (i) marginalize the χ^2 hypersurface over various parameters of interest and declare the 1σ confidence interval to be a χ^2 level of some threshold above χ^2min. If one has normal data with perfectly defined normally-distributed error bars, no systematic effects, etc., then the usual χ^2 threshold is $\chi^2_{\rm min}+1$. The UNIX shell scripts **getgenerationvals.com** and **getELCparmvals.com** that all of the optimizer codes make can help you marginalize the χ^2 hypersurface by generating separate files with columns of parameter value and χ^2. Alternatively, use the code **ELCchiplotter.f90**, and use the **ELCjunk.TAG_STRING** files. You can run various optimizers on the same data in different subdirectories and combine the corresponding **ELCjunk.TAG_STRING** files.

On the other hand, it is often not obvious if one can use the usual $\chi^2_{\rm min}+1$ threshold for the 1σ errors (i.e. one might need to add more than 1 to $\chi^2_{\rm min}$). In cases like this, use the posterior samples from **demcmcELC.f90** or **hammerELC.f90** for situations with many free parameters ($N_{\rm var}>20$), or the posterior samples from **markovELC.f90** or **nestedELC.f90** when the number of free parameters is small.

Use the helper codes to look at the progress of the chains. For example, Figure 12 shows the chains of the **ocose** parameter from a run of **demcmcELC.f90** that had 15 free parameters and 150 chains. There was a burn-in period of perhaps 600 generations. A few of the chains were stuck in regions of relatively poor likelihood, and all but one of them were purged by generation 3200. The two highlighted chains cross the posterior space every few hundred generations. To get a poster sample, start at generation 3200, and skip 200 models between samples:

```
ELCposterior 3200 100 100000 50 4.
```

This will use a clipping of 4σ and 50 bins. The posterior samples have 1650 points, so one might want to let this run continue for a while longer. You can also experiment with the **Nskip** parameter. For example, compare the histograms using **Nskip=150** and **Nskip=200**.

Figure 13 shows a case where the chain mixing was disappointing. This plot was generated from a run of **hammerELC.f90** that had 58 free parameter (many of them were limb darkening coefficients) and 800 walkers. Tempering was used in this case, and one can see the spread in the parameter (**ocose**) early on. After the tempering phase ended at generation 500, the spread of the chains remains fairly constant. However, when a few individual chains are highlighted, one can see that those chains have not crossed the entire space. Since the chains were initially over-dispersed, one still might get a good posterior sample if one uses a large **Nskip** (say 2000). Since there are 800 walkers, the posterior sample will still have a decent number of points.

In cases like that shown in Figure 13, it might be better to use fewer walkers (try something like three times the number of free parameters, or $N_{\rm walker} \approx 150$ in this case), and experiment with the tempering and with **ielite** by putting more than one good model into the initial population.

9. Appendix A: Old ELC.inp

Here is an excerpt from a previous version of this manual showing the contents of **ELC.inp** using the old format. Many of the newer features won't work using the old format input, so moving to the new format is encouraged. To save space, I will use a smaller font.

`Nalph1, Nbet1, Nalph2, Nbet2`: These specify the number of grid elements used to integrate the intensity. The "`alpha`" direction is the latitude on the star, running from the "north pole" (`ialf=1`) to the "south pole" of the star (`ialf=Nalf`) and represents equal steps in the angle. At each `alpha`, you loop over the "`beta`" direction which is the longitude on the star. The `beta` steps are equal in longitude, and there are a total of `4*Nbet` points. Minimal values for these parameters are `Nalph=10` and `Nbet=6`. In the subroutine `lcsubs.f90` you will find these lines near the top

```
parameter(ialphmax1=1000,ibetmax1=1000)

parameter(ialphmax2=200,ibetmax2=800)
```

You can adjust these numbers as needed: smaller numbers will save a bit of memory, larger numbers will allow you to include more points in the integration.

`fill1, fill2`: These are the Roche-lobe filling factors for star 1 and star 2, respectively. `fill=1.0` means the star exactly fills its Roche lobe. The filling factor must be larger than 0.0 and less than or equal to 1.0. The values of `fill1` and/or `fill2` may be overridden if one or more of the parameters `primrad`, `ratrad`, `frac1`, and `frac2` are set below.

`omega1, omega2`: These parameters specify the ratio of the rotational frequency of the star to the orbital frequency of the binary. If the star is tidally locked, then `omega=1.0`.

`dphase`: The step size (in degrees!) used when "turning the binary in space". The flux at phase 0.0 is always computed, and fluxes will be computed for `dphase`, `2*dphase`, ... `360-dphase`. Note, however, that the phase units of the output files (see below) are normalized to 1.

`Q`: This is the mass ratio of the binary. It is the ratio of the mass of star 2 to that of star 1. For an X-ray binary like A0620-00, `Q` would be on the order of 15. The value of `Q` may be overridden if one or more of the parameters `asini`, `primmass`, and `primK` are set below.

`finc`: This is the inclination of the orbital plane, specified in degrees. The orbital plane is viewed edge-on for `finc=90`.

`Teff1, Teff2`: These are the *mean* temperatures of star 1 and star 2, respectively. `Teff1` must be larger than 0. If you want to make star 2 invisible, set `Teff2` to a negative value. Note that in the previous (OB97) code, the *polar* temperature of the star was specified. As discussed by Wilson (1979), the intensity-weighted mean temperature is a more representative of what is actually observed from the spectral type of the star. The value of `Teff2` may be overridden if `temprat>0` (see below).

`Tgrav1, Tgrav2`: These are the "gravity darkening" exponents for star 1 and star 2, respectively. The value of `Tgrav` should be 0.25 for stars with radiative envelopes (von Zeipel 1924) and roughly 0.08 for stars with convective envelopes (Lucy 1967). If the flag `igrav` is larger than 0 (see below), then these input values might be ignored.

`betarim`: This is the opening angle (in degrees) of the disk rim above the plane.

`rinner`: This is the inner radius of the disk, in the same units as `fill2`. Hence `rinner` should be equal to or larger than `fill2`. Currently the optimizer codes assume `fill2=rinner` for cases when star 2 is visible and there is a disk.

`router`: This is the outer radius of the disk, expressed as a fraction of star 2's effective Roche lobe radius. This parameter should be less than 1. If the inner disk radius is larger than the outer disk radius in the internal program units, the program will stop and complain.

`tdisk`: This is the temperature of the *inner* disk. Note that on the old code (OB97) the temperature of the outer rim was specified. If `tdisk` is negative, the disk will be dark. It will still cause eclipses for high inclinations, but it will not contribute to the total light.

`xi`: This is the power-law exponent on the temperature profile of the disk:

$$T(r) \propto T_d (r/r_{\rm inner})^{\xi}$$

For a "steady-state" disk, `xi` should be -0.75. For disks that are irradiated by the central source, `xi` might be larger, perhaps on the order of -0.425 (i.e. the temperature profile on the disk is flatter and the outer parts of the disk are hotter than they would be otherwise).

`Ntheta, Nradius`: The number of grid points on the disk in the azmithual and the radial direction, respectively. You need a relatively large number of points in the radial direction to get the flux integration to converge reasonably well.

`alb1, alb2`: These are the bolometric albedos used in the computation of the reflection effect. The albedo should be 1.0 for stars with a radiative envelope, and 0.5 for stars with a convective envelope.

`Nref`: The number of iterations for the reflection effect (see Wilson 1990). To skip the reflection effect routines entirely, set `NREF=-1`. If `NREF=0` a "simplified" computation is used. This approximation is exact for spherical stars and becomes worse as the stars deviate from spherical symmetry. If `NREF > 0`, then the "detailed" reflection is called `NREF` times. `NREF=1` is adequate for most purposes. In most situations in the black body mode, the computation of the reflection effect takes most of the time.

If you have an X-ray binary and want to compute the effects of X-ray heating, set `Nref=0` and set the parameter `lx/Lopt` accordingly. By default, the X-ray source is assumed to be a thin disk. If you want to make the X-ray source a point source, set `iXheat=1` (see below).

`Lx/Lopt`: This is the logarithm (base 10) of the X-ray luminosity of star 2 in ergs s^{-1} in the cases where it is invisible—i.e. `Teff2 < 0`. This number is used in the computation of X-ray heating.

`Period`: This is the orbital period of the binary in days.

`fm`: This is the mass function of star 2 in solar masses, measured from the radial velocity curve of star 1. The mass function is used to compute the orbital separation in cases where the orbital separation is not expressly given. This feature is mostly obsolete.

`separ`: This is the orbital separation in solar radii. If this number is negative, then the orbital separation is *initially* computed from the mass ratio, period, inclination, and the mass function `fm`. If `separ` is positive, its value may be overridden by one or more of the parameters `asini`, `primmass`, and `primK` below. Once the value of `separ` is set, then the component masses are computed from the period (using Kepler's third law) and the mass ratio. The orbital separation is used to compute the surface gravity of each element when model atmosphere intensities are used, so it is helpful if the scale of the binary is known (the light curve amplitudes in the black body mode are independent of the orbital separation). Note: for eccentric orbits you must input a positive separation.

`gamma velocity`: This is the velocity zero point for the radial velocity curves. This number does not necessarily have to be specified correctly since the fitting programs adjust this number internally.

`t3`: The temperature of the "third light" star. If there is a third star associated with the binary (i.e. a distant star which forms a triple system), then set this number to some positive value. To set the third light to zero, enter a negative value for `t3`.

`SA3`: The surface area of the "third light" star, expressed as a fraction of the surface area of star 1. To set the third light to zero, enter a negative value for `SA3`.

`g3`: The surface gravity of the "third light" star, expressed in the usual convention: the common logarithm of the gravity in cgs units. This number is needed for completeness so the code can pick out a flux from the model atmosphere table. As before, to set the third light to zero, enter a negative value for `g3`.

`density`: The surface gravity of star 1 in cgs units. Owing to the magic of Roche geometry, the surface gravity of a star in a close binary is almost uniquely set by the scale of the binary (e.g. the period and separation). Conversely, if the density of star 1 is known independently, then the range of the parameters `fill1` and `Q` are restricted. Use the `density` parameter to force the density of star 1 to be the specified value. For this to work, the parameters `frac1`, `primmass`, and `primrad` below all must be set to 0. I am not sure how well fitting for this parameter works in practice.

`onephase`: If the value of the integer flag `ionephase` is 1, then the program will compute the flux for only one phase whose value is set by this parameter. This feature is useful if you want to make a picture of the binary at a specific phase.

`usepot1, usepot2`: These are the "Ω-potentials" which are the Wilson Devinney equivalent of the `fill1, fill2` variables above. If `iusepot=1` (see below), then the filling factors of the two stars are computed using these potentials. If the entered value of a potential is smaller than the critical potential, then the filling factor is set to 1.

`T0`: The epoch of the phase zero-point when fitting light curves with time units. The value of `T0` must be in the same units as the input data, and the flag `itime` (see below) must be set to 1. `T0` is the time of inferior conjunction of star 1 for circular orbits, or `T0` is the time of periastron passage for eccentric orbits. The time of primary eclipse can be set with the Tconj parameter below.

`idraw`: The control integer for writing output files. Use `idraw=1` to write output files that the separate graphics programs use. See Section **7.4.7**.

`iecheck`: If the disk is axisymmetric (i.e. there are no spots), the integrated flux does not depend on the phase, except for possible eclipses. Thus you only need to compute the disk flux during the eclipse phases, and only once thereafter. Set `iecheck=0` to compute the disk flux at each phase, or `iecheck=1` to skip the computation at the phases outside of eclipse. If you set `iecheck=-1` the code will not check for eclipses. Thus you can see how much difference the eclipses make. The increase in the execution speed is quite small when the eclipse checking is turned off. Note: if you have `iecheck=1`, then the intensity maps for the disk (see the `idraw` option above) will be *incorrect* for most phases (the light curve will of course be correct).

When you have a normal binary star with no disk, the flag `iecheck` can be used to cut down the execution time. Set `iecheck=5` and the code will compute the light curve at 2 degree intervals outside of eclipse. Set `iecheck=9` and the code will compute the light curves only during eclipses. A constant "reference" flux will be used for the out-of-eclipse light curves. This is useful for the computation of transits of extrasolar planets where the ratio of radii is often extreme.

`idint`: This integer is used to switch on the disk. Set `idint=0` to leave out the disk, and `idint=1` to include the disk in the light curve computation.

`iatm`: Set `iatm=0` to compute the local intensities using the Planck blackbody formula and the limb darkening law, and `iatm=1` to compute the local intensities using the model atmosphere table. If you set `iatm=1`, you must specify the scale of the system correctly since the model atmosphere flux depends on the local gravity in physical units. Thus you should give the code the correct orbital period and mass function (or separation). Caution: The code presently does very little checking to see if the temperatures required by the input values of `Teff1, Teff2, tdisk` are within the range of the table. Ditto for the gravities. The atmosphere table will be expanded in the near future, and as time permits I will add more safeguards.

You can read the first few "header" lines of the table to see which models are there and for which filters.

If you set `iatm=2`, then the code will use the ELC.atm file to set the intensity at the surface normals. For other angles, the limb darkening law is used. Thus, this is like the blackbody mode, except that the model atmospheres are used to set the luminosity scaling of the two stars.

`ism1`: Set `ism1=0` to force the code to compute the entire light curve and `ism1=1` to only compute "half" of the light curve. If the orbit is circular, only the phases 0 to 180 degrees are computed if `ism1=1`, and the rest of the phases are filled in by symmetry. Note that the time savings is far less than a factor of two on average since the "setup" time dominates (the initial geometry and reflection effect computations). If the orbit is eccentric, the program makes use of a symmetry of the *geometry* around periastron if `ism1=1`. For example, the geometry (instantaneous orbital separation, filling factors, etc.) of the binary 10 degrees before periastron is the same as the geometry 10 degrees after periastron. Since the time to compute a light curve is dominated by the "setup", setting `ism1=1` for eccentric cases almost always saves a factor of two in computing. Note if you have spots, then you must set `ism1=0` since you no longer have symmetry.

If you get a strange looking radial velocity curve for an eccentric orbit, set the `ism1` flag back to zero.

`icnU, icnB, icnV, ...`: Switches for the "filter wheel". To save overhead computing time, the code computes the light curves for 8 different wavelengths (or filters) at each phase. In the blackbody mode the wavelengths are given at the end of the input file with the limb darkening parameters (see below). In the model atmosphere mode, the intensities are filter-integrated (i.e. not monochromatic). The current table contains intensities for the Johnson $UBVRIJHK$ filters. Other filters can used without much effort—contact me in case you need a different filter. In the model atmosphere mode, set `icnU` to 1 to compute the light curve for the U band, or 0 to skip; set `icnB` to 1 to compute the B-band light curve, 0 to skip, etc.

`iRVfilt`: This is the index of the wavelength (or filter) used to compute the radial velocity curves. Note that the form of the radial velocity curves is practically independent of the input wavelength. This switch also controls which light curve is output in the `lcstar?.linear` and `lcdisk.linear` files. Thus set `iRVfilt` to 3 to compute radial velocity curves for the V band and to output individual V-band light curves, etc.

`ionephase`: Set this flag to 1 if you want to compute the flux for only a single phase. The specific phase you want is set by the variable `onephase` above. Note that if the orbit is eccentric, then the output phase given in the files listed above may not be the same as `onephase`.

`isquare`: This flag is used to control the number of longitude points per latitude row. `isquare=0` gives you the "elliptical" arrays that are used in the Wilson & Devinney code (fewer longitude points near the poles), and `isquare=1` gives you "square" arrays (the same number of longitude points at all latitude rows). The code might choke if `isquare=0` and if the number of latitude and longitude points (`Nalph, Nbet`) points is small. In this case, set `isquare=1`. If you want to draw the binary, then set `isquare=1`.

`iusepot`: Controls the input variables for the filling factors of the two stars. Set `iusepot=0` to specify the filling factors using `fill1, fill2` (the usual mode). Set `iusepot=1` to use the Ω potentials `usepot1,usepot2`. The use of the Ω potentials makes it much easier to compare the ELC light curves with Wilson Devinney light curves.

`ifixgamma`: Set to `ifixgamma=0` to have the input value of `gamma` used in the fitting one or both velocity curves. Set to 1 to have the `gridELC` code adjust the `gamma` velocities when fitting the radial velocity curves. Set `ifixgamma=2` to have a common value of `gamma` fit to *both* velocity curves.

`ilaw`: This controls the form of the limb darkening law. Set `ilaw=1` for the linear law:

$$I(\mu) = I_0(1 - x + x\mu)$$

(where μ is the cosine of the foreshortening angle of the grid element). Set `ilaw=2` for the logarithmic law:

$$I(\mu) = I_0(1 - x + x\mu - y\mu \ln \mu).$$

Set `ilaw=3` for the square root law:

$$I(\mu) = I_0(1 - x + x\mu - y(1 - \sqrt{\mu})).$$

Finally, set `ilaw=4` for the quadratic law:

$$I(\mu) = I_0(1 - x + x\mu - y(1 - \mu)^2).$$

You can find tables of the coefficients in Van Hamme (1993). Note that the same law is used in the computation of the reflection effect and in the wavelength dependent intensity computations.

If you have a binary with nearly identical stars, you can add 10 to the `ilaw` flag of one of the above values to force the limb darkening laws for both stars to be the same. Thus, if `ilaw=12`, then both stars will have the same logarithmic law with the same coefficients.

The next 8 lines of the input file contain the wavelengths and the limb darkening coefficients for the 8 filters. (The code has a "filter wheel" with 8 positions. To save some computing time, light curves for different wavelengths are computed at the same time. To compute a light curve, you have to define the geometry. This needs to be done only once at phase 0. Once this is done, turning the binary in space takes very little time. This method of computing 8 light curves at once thus takes less time than running the code 8 different times.) Each line should contain the following type of numbers:

```
waveU xbol1(U) ybol1(U) xbol2(U) ybol2(U) x1(U) y1(U) x2(U) y2(U)
```

The coefficients `xbol1, ybol1, xbol2, ybol2` are the coefficients for the limb darkening law used in the reflection effect (Wilson 1990). Van Hamme (1993) has tabulated these coefficients for most temperatures and gravities. These bolometric limb darkening coefficients do not depend on the wavelength, and only the ones for the first line are used by the program. Note that you still have to specify coefficients for the other filters as well—you just don't have to specify them *correctly*! The coefficients `x1, y2, x2, y2` are wavelength dependent. These you should try to specify correctly.

Here is something to note: Some of the `x` coefficients can be larger than 1 for certain situations (for example cool stars in the bluer filters combined with the square root law). If you change the limb darkening law to linear and use the same coefficients, you might get numerical errors.

For the quadratic law, the sum of the coefficients is not allowed to exceed 1 when using the optimizer codes.

The default wavelengths are for the Johnson $UBVRIJHK$ system.

`ecc`: The eccentricity of the orbit. The code has problems with lobe-filling stars and eccentricities greater than about 0.75. The Wilson-Devinney code also has problems with these binaries, and apparently the difficulties are related to the physical model rather than to program coding. These binaries probably do not exist in Nature anyway, so this problem presumably causes few troubles in practice. The value of `eccen` will be overwritten if you are fitting for $e\cos\omega$ and $e\sin\omega$ (see below).

`argper`: The argument of periastron, in degrees! This parameter will be overwritten if you are fitting for $e\cos\omega$ and $e\sin\omega$ (see below).

`pshift`: The phase shift to be applied to the output light and velocity curves. The shift must be between -1.0 and 1.0. This parameter can be used to effectively switch stars 1 and 2 (e.g. put `pshift=0.5` to have the same phase convention as the Wilson-Devinney code).

`asini`: For use with pulsar binaries. This parameter is the projected semi-major axis of the pulsar's orbit *in light seconds*. If `Teff2 < 0` and `asini > 0`, then the orbital separation `separ` is computed from the values of the mass ratio `Q`, the inclination `finc`, and `asini`:

$$a = \frac{\texttt{asini}(1+Q)c}{\sin i}.$$

`rmed`: If `rmed > 1`, then the genetic fitting code will use the absolute deviation as the measure of fitness, rather than the normal χ^2.

`sw7, sw8`: These two parameters can be used to specify a restricted range of phases to compute. Set `sw7 > 0`, `sw8 > 0`, and `sw7 < sw8`. The units are degrees.

`time_step`: This is the time step (in days) used if one is computing the light curves in time units (`itime=2`, see below).

`ikeep`: Set to 1 to place the eclipse of star 2 at phase 0.0, and `ikeep=2` to place the eclipse of star 1 at phase 180.0. If `ikeep=0`, then the phase of the eclipse depends on the eccentricity and argument of periastron, as in the Wilson & Devinney code.

`isynch`: Controls whether the stars rotate synchronously at periastron for eccentric orbits. Set `isynch=0` to have the code use the input values of `omega1, omega2` and `isynch=1` to have the code compute the values of `omega1,` `omega2` for you:

$$\Omega = \sqrt{\frac{1+e}{(1-e)^3}}.$$

`ispotprof`: Controls the type of star spot used. `ispotprof=0` is the default (constant temperature scaling). `ispotprof=1` uses a linear change in the spot profile, so that the temperature scaling is greatest near the center of the spot and falls off near the spot edges. `ispotprof=2` uses a Gaussian profile for the temperature scaling.

`igrav`: Set `igrav=0` to use the input values of `Tgrav1, Tgrav2`, set `igrav=1` to have the code choose `Tgrav1` for you based on the input value of `Teff1`, set `igrav=2` to have the code choose `Tgrav2` for you based on the input value of `Teff2`, and set `igrav=3` to have the code choose *both* `Tgrav1` and `Tgrav2` for you based on the input values of `Teff1` and `Teff2` (Claret 2000, see Figure 1).

`itime`: If `itime=0`, the light curves are computed in phase units, and the input data being fitted with the optimizer codes are given in phase units. If `itime=1`, then the fitting routines will assume the input light and velocity curves are in time units rather than phase units. The light curves are still computed internally in phase units. If `itime=1`, then the values of `period` and `T0` are used to fold the light curves in phase. You may fit for `period` and/or `T0`. If `itime=2`, the light curves are fitted in time units. The value of `time_step` is the time step (in days), and the values of `t_start` and `t_end` give the starting and ending times. **When using the optimizers with `itime=2`, make sure the time range of the model is larger than the time range of the data.** That is, the first time of the model should be smaller than the smallest time in the input data, and the last time of the model should be larger than the largest time in the input data.

`MonteCarlo`: This flag is mainly for the computation of accurate transit light curves for extrasolar planets. If `MonteCarlo=0`, then a simple interpolation scheme is used to correct for partially eclipsed pixels. When `MonteCarlo>10`, a Monte Carlo integration scheme is used to correct for fractionally eclipsed pixels. In practice, `MonteCarlo` should be set to something between 200 and 1000.

`ielete`: Used with the genetic and Markov fitting codes. If `ielete=1`, a light curve with the parameters given in `ELC.inp` is inserted into the initial population (genetic), or is used as the initial starting point (Markov). If `ielete=0`, the initial population is random, or the initial starting point is random.

The next 24 lines are parameters for spots. Each spot has 4 parameters, and each body (star 1, star2, and the disk) can have two spots each. For each spot you need to specify:

`Tfac`: The "temperature" factor of the spot. Make this number larger than 1 for a hot spot and a positive number less than 1 for a cool (or dark) spot. The temperature of a surface element within a spot is the underlying temperature times the temperature factor of the spot, subject to the underlying scaling function determined by the `ispotprof` flag above.

`Lat`: The latitude of the spot for the stars *or* the azimuth of the spot if the body in question is the disk. The units are *degrees*. For stars, the latitude is 0 degrees at the "north pole" (positive z-value), 90 degrees at the equator (orbital plane), and 180 degrees at the "south pole." For the disk, a spot at azimuth 90 degrees is seen directly ($\mu = 1$) at orbital phase 90 degrees (circular orbit), a spot at azimuth 200 degrees is seen directly at orbital phase 200 degrees, etc.

`long`: The longitude of the spot for the stars *or* the radial cutoff of the spot if the body in question is the disk. The units are *degrees*. For stars, the longitude is zero at the inner Lagrangian point and 180 degrees at the back end. A latitude of 90 degrees corresponds to the side of star 1 that is seen directly at orbital phase 90 degrees (circular orbit), or the side of star 2 that is seen directly at orbital phase 270 degrees (circular orbit). For the disk, the spot is always on the rim. How far down the disk is controlled by the cutoff radius. Set the cutoff radius to 0.9 to have the spot fill the outer 10% of the disk sector, etc.

`rad`: The radius of the spot (in degrees) for the star spots *or* the angular size of the disk spot.

If you want to turn on a spot, set the temperature factor to a positive number. If you don't want any spots at all, set all of the temperature factors to negative numbers.

The next 6 parameters let you specify the input geometry in various ways. How these work will be described below.

`primmass`: The mass of star 1 in $M_\odot$.

`primK`: The K-velocity of star 1 in km $\sec^{-1}$.

`primrad`: The radius of star 1 in $R_\odot$.

`ratrad`: The ratio of star 1 radius and star 2 radius.

`frac1`: The fractional radius of star 1: R_1/a, where a is the orbital separation.

`frac2`: The fractional radius of star 2: R_2/a, where a is the orbital separation.

Depending on which of the above 6 parameters are set, the values of `Q`, `separ`, `fill1`, and `fill2` may be changed:

- if `primrad=0` and `frac1>0` then `fill1` is computed and set. If the fractional radius is too large, then `fill1` is set to 1.
- if `frac2>0`, `fill2` is computed and set. If the fractional radius is too large, then `fill2` is set to 1.
- if `primrad>0` and `frac1=0`, the value of `fill1` is computed and set, if possible. If the specified radius turns out to be large than the Roche lobe radius, then `fill1` is set to 1.
- if `primrad=0`, `frac1=0`, and `ratrad>0`, the value of `fill1` is adjusted to get the radius of star 1, based on the value of the radius of star 2 (which either comes from the value of `fill2` directly, or from the value of `fill2` computed from `frac2`). If the computed radius of star 1 turns out to be large than the Roche lobe radius, then `fill1` is set to 1.
- if `frac1=0`, `frac2 =0`, and `ratrad>0`, the value of `fill2` is set to get the radius of star 2, if possible. If the radius of star 2 turns out to be larger than the Roche lobe radius, then `fill2` is set to 1.
- if `primmass>0`, and `primK=0`, the value of `separ` is set.
- if `primK>0` and `primmass=0`, then the value of `separ` is set.
- if `asini>0`, `Teff2 < 0`, and `primK>0`, the values of `Q` and `separ` are set.
- if `primmass>0` and `primK>0`, the values of `Q` and `separ` are set.

The values of `fill1`, `fill2`, `Q`, and `separ` will be reported in the file ELC.out (see below).

`ecosw`: The phase difference between secondary and primary eclipses for an eccentric orbit. The units are normalized phase units such that 2π`ecosw` is the phase difference in radians. If `ecosw>0` and `ecc>0`, the value of `argper` is computed and set. If you have been fitting for this parameter to get initial guesses for `ecc` and `argper`, don't forget to set `ecosw=0` if you wish to fit for `ecc` and `argper`.

`temprat`: The ratios of the effective temperatures of star 2 and star 1: `temprat`=`Teff2`/`Teff1`. If this parameter is greater than zero, then the value of `teff2` is overwritten.

`idark1`: If `idark1=1`, star 1 will not contribute to the light.

`idark2`: If `idark2=1`, star 2 will not contribute to the light. This might be the case for a transiting extrasolar planet, where star 2 is usually the planet.

`Npoly`: For cases when both stars are nearly perfect spheres (e.g. transiting extrasolar planets), set `Npoly=0` for the normal numerical integration (which may be relatively slow), and `Npoly > 0` for to use the analytic expressions given in Gimenez (2006a, 2006b). When the analytic expressions are used, `Npoly` represents the number of terms in the series expansions. Values of `Npoly` between 10 and 200 are recommended. **Note that the limb darkening law must be linear (`ilaw=1`) or quadratic (`ilaw=4`) when the analytic expressions are used.** The square root law (`ilaw=3`) will be added soon.

This analytic mode is mostly obsolete. Use `Nterms` as discussed below.

`ifasttrans`: This flag was part of an effort to speed up the computation of transit light curves of extrasolar planets. If the star has symmetry (e.g. no spots or the like), the flux from the parts of the star that are not eclipsed will be the same always. Setting `ifasttrans=1` will have the code use fewer pixels in the parts of the star that are not eclipsed. However, it turns out that this flag does not really help, so using `ifasttrans=0` is recommended.

`ialign`: Set `ialign=0` to have the rotational axis of star 1 aligned with the angular momentum vector of the orbit. Set `ialign=1` to have the rotational axis misaligned. Adjust the parameters `axis_I` and `axis_beta` below accordingly. This parameter will mainly make subtle changes to the radial velocity curve of star 1 during transit (e.g. the Rossiter-McLaughlin effect).

`ifastgen`: Set `ifastgen=1` to enter the "fast genetic mode." This can be useful if it takes a fair amount of time to compute a single model. In the early phases of the genetic code, most models do not come close to fitting the data. When `ifastgen=1`, a model with a coarse grid in `Nalph` and `Nbet` is computed, and the χ^2 is computed. If the χ^2 value is less than $3\chi^2_{\rm min}(N_{\rm data} - N_{\rm terms})$ then the curve is recomputed using the full grid. This flag helps in the beginning, but becomes less and less useful as the genetic code converges.

`isw23`: Currently inactive.

`ifracswitch`: Set `ifracswitch=1` to enable geneticELC and markovELC to write the "luminosity ratios" and "disk fractions" to the ELCratio.??????? files. See below for more information on the optimizers.

Next in the input file comes the "power law coefficients", which are the limb darkening coefficients for the analytic mode. **These are not yet fully implemented, so put 8 lines of 9 columns of zeros.**

`axis_I`: This is the inclination of the rotational axis of star 1 if `ialign=1`. See Hosokawa (1953) for helpful diagrams and equations.

`axis_beta`: This is the angle of the rotational axis of star 1 with respect to the orbit if `ialign=1`. See Hosokawa (1953) for helpful diagrams and equations.

`t_start`: When `itime=2`, the light curves and velocity curves are internally computed in time units. `t_start` is the starting time.

`t_end`: When `itime=2`, the light curves and velocity curves are internally computed in time units. `t_end` is the ending time.

`asini_error`: In the "millisecond pulsar mode", the parameter `asini` is the projected semimajor axis of the pulsar in light seconds. `asini_error` is the uncertainty in `asini`, and if nonzero will be used in the genetic and Markov codes. Values of `asini` will be drawn from a Gaussian distribution with a standard deviation of `asini_error`.

`disk_frac_ref`: If this parameter is positive, it gives the reference phase for the disk fraction when using the optimizer codes (see below). Otherwise, the median of the disk fraction over the whole orbit is used.

`radfill1`: If this parameter is greater than zero, it sets the Roche lobe filling factor of star 1 in terms of $R_{\rm eff}$, rather than in terms of the distance to the L_1 point.

`radfill2`: If this parameter is greater than zero, it sets the Roche lobe filling factor of star 2 in terms of $R_{\rm eff}$, rather than in terms of the distance to the L_1 point.

`lcbin`: This parameter is the exposure time of the typical photometric observation you have, in minutes. For example, if you are working with *Kepler* data, use `lcbin=29.4744`. For this flag to work properly, you need to set the `dphase` parameter such that there are 20 to 30 steps per time bin (e.g. one to three minutes for Kepler long cadence data). For example, if you have a binary with a period of 30 days, a step size of one minute in time is a step size in terms of the normalized orbital phase of 0.00002315, which works out to be $360 \times 0.00002315 = 0.008333$ degrees. In this case, you can round up a bit, and set `dphase=0.01`.

`rvbin`: Similar to above, but for the typical exposure time (in minutes) of the spectroscopic observations that were used to find the radial velocities. In practice, setting this parameter hardly makes any difference in the cases I have tried so far.

contam: This is the contamination to be applied to models of *Kepler* data. In many of the *Kepler* eclipsing binaries and transiting planets, there is light from other stars in the photometric aperture. The extra light causes the relative depths of the eclipses (or the transits if you have a planet) to go down. If this parameter is greater than zero and Iseason=0 (see below), an offset is applied to the model *Kepler* curve (in the U filter position if you have the special ELC.atm file):

$$y_{\rm off} = \frac{k y_{\rm med}}{1-k}$$

where k is the value of *contam* parameter and $y_{\rm med}$ is median value of the model light curve. The value of contam should not exceed 1.

Tconj: The time of a primary eclipse. To use this parameter, set the Tconj flag below to 1.

beam1: The Doppler boosting factor for star 1 (see van Kerkwijk et al. 2010). If this parameter is zero, no correction to the light curve is applied. A typical value is 2.5 for an A-star.

beam2: The Doppler boosting factor for star 2 (see van Kerkwijk et al. 2010). If this parameter is zero, no correction to the light curve is applied. A typical value is 2.5 for an A-star.

isw25: If isw25=1, the X-ray source is assumed to be a point source when computing X-ray heating. If isw25=0, then a foreshortening correction for the X-ray heating is applied, assuming the X-ray emitting part of the accretion is a thin disk in the orbital plane.

iGR: Flag for adding GR corrections and/or tidal corrections to the dynamical integrator. Set to zero for no corrections, 1 for GR only, 2 for tidal only, and 3 for both.

Nterms: This flag activates the fast analytic mode. See the description of Npoly above. Useful values are between 100 and 500. For various technical reasons, light curves are computed much faster when Nterms=300 compared to when Npoly=300.

iuseTconj: Set this flag to 1 to use a time of primary eclipse as the reference time rather than a time of periastron passage.

iecosflag: Set this flag to 1 to specify the eccentricity and argument of periastron by $e\cos\omega$ and $e\sin\omega$.

Nbody: Flag for the dynamical integrator. Set Nbody=3 for three bodies, Nbody=4 for 4 bodies, etc. The maximum value is 10. When using the dynamical integrator, there is a separate input file called ELCbody3.inp that is used, as explained below.

Ngap: When itime=2, this switch allows the user to skip various segments in the light curve. If this switch is greater than zero, the code looks for a file called ELCgap.inp. This file should contain Ngap lines giving the starting and ending times of the gaps that the code should skip. If there is a problem reading this file, the code defaults to Ngap=0, which computes the entire light curve between t_start and t_end.

jdum: Random number seed for demcmcELC, geneticELC, markovELC, and randomELC. If you wish to use a starting seed other than the default value, set this to a negative integer.

mandel: When Nterms> 0, this switch controls which algorithm to use for the analytic expressions. Set to 0 for the Gimenez routines, and 1 for the Mandel & Agol routines, which are faster. Note, however, that the computation of the Rossiter effect is implemented in the Gimenez routine and not in the Mandel & Agol routine.

Iseason: When using itime=2, set this switch to 1 to compute seasonal contamination values for the different *Kepler* Quarters. For this option to work, the time units must be in the form BJD - 2,455,000. The values of the seasonal contamination parameters are given below.

e*cos(omega): If iecosflag=1, compute the eccentricity using $e\cos\omega$ and $e\sin\omega$. This flag gives $e\cos\omega$.

e*sin(omega): If iecosflag=1, compute the eccentricity using $e\cos\omega$ and $e\sin\omega$. This flag gives $e\sin\omega$.

omega_dot: If itime=2 and Nterms=0, this parameter will attempt to mimic apsidal motion. The units are degrees per year.

`contamS0, contamS1, contamS2, contamS3`: The values of the seasonal contamination parameters when `itime=2` and `Iseason`>0. The time ranges for the parameters are:

start time	end time	parameter
-46.0	0.0	`contamS0`
0.0	92.0	`contamS1`
92.0	183.0	`contamS2`
183.0	276.0	`contamS3`
276.0	372.0	`contamS0`
372.0	463.0	`contamS1`
463.0	553.0	`contamS2`
553.0	636.0	`contamS3`
636.0	739.0	`contamS0`
739.0	834.0	`contamS1`
834.0	934.0	`contamS2`
932.0	1015.0	`contamS3`
1015.0	1106.0	`contamS0`
1106.0	1205.0	`contamS1`
1205.0	1304.0	`contamS2`
1304.0	1391.0	`contamS3`
1391.0	1481.0	`contamS0`

`Tref`: The reference time for the dynamical integrator (`itime=2` and `Nbody` ≥ 3).

`chi_thresh`: If this parameter is more than zero, these optimizers won't update the generation.*, ELCparm.*, etc. files when $\chi^2 > \chi_{\rm small} + \chi_{\rm thresh}$: demcmcELC, geneticELC, markovELC, randomELC.

`Omega_bin`: When `Nbody` ≥ 3, this parameter gives the nodal angle of the binary in degrees.

`Teff4`: The effective temperature of body 4 when `Nbody`≥ 4 and `itime=2`.

`Teff5`: The effective temperature of body 5 when `Nbody`≥ 5 and `itime=2`.

`g4`: The logarithm of the surface gravity of body 4 when `Nbody`≥ 4 and `itime=2`.

`g5`: The logarithm of the surface gravity of body 5 when `Nbody`≥ 5 and `itime=2`.

`sw84, sw85, sw86, sw87, sw88, sw89`: These are currently inactive real*8 input parameters. Use 0.0, one per line.

`isw81, isw82, isw83, isw84, isw85, isw86, isw87, isw88, isw89`: These are currently inactive integer input parameters. Use 0, one per line.

The final entries comes in the form of a matrix of numbers with 8 columns and 5 rows. These give the fluxes for star 1 in 8 filters, star 2 in 8 filters, body 3 in 8 filters, body 4 in 8 filters, and body 5 in 8 filters,

That is it for the input parameters. You should have a total of 180 lines.

If you are using the dynamical integrator `Nbody` ≥ 3 and `itime=2`, you will need a file called ELCbody3.inp. There are 106 lines:

`Nalph3, Nbet3`: These specify the number of grid elements for the third star when `itime=2`, `Nbody=1`, and `Nterms=0`. This mode is currently undergoing some revisions.

`itconj`: This controls what the time the parameter `tertT0` (see below) refers to: `itconj=0` for reference to the time of periastron passage of the third body, `itconj=1` for reference to the time of the third body transit of the *barycenter of the binary*, and `itconj=2` for reference to the time when the third body goes directly behind the *barycenter of the binary*.

`ilog`: When `ilog=1`, the values of mass ratios for the extra bodies are assumed to be actually logarithms (base 10) of the actual mass ratios.

`i_inform`: When using the dynamical integrator (`itime=2` and `Nbody`≥ 3), setting this parameter to 1 will enable the generation of some helpful summary files:

ELCdynamics.out: This gives a detailed summary about what the initial conditions that went into the integrator were, the positions and velocities at the end of the integration, etc.

ELCdynamics.pos: This gives the x, y, and z coordinates of all of the bodies at each time step, starting with star 1 (x,y,z), star 2 (x,y,z), the third body (x,y,z), and so on.

ELCdynamics.vel: This gives the x, y, and z velocities of all of the bodies at each time step, starting with star 1 (v_x,v_y,v_z), star 2 (v_x,v_y,v_z), the third body (v_x,v_y,$v)z$), and so on.

ELCdynamics.tex: This file has two tables formatted in LaTeX (using the AASTex deluxetable macros) that summarize the initial Keplerian conditions and the initial barycentric coordinates used for the integration. For this option to work properly, you must compile with an option which makes the compiler ignore the ” (backslash):

gfortran -ffixed-line-length-132 -fno-backslash -mcmodel=medium -o ELC ELC.f90

`itran`: An experimental flag that is of use only to me.

`isupress`: Set this to 1 to suppress some of the output files generated by demcmcELC (demcmc_dynparm.1* and demcmc_chi.1*).

`tertconj`: The time of the third body transit or occultation of the *binary barycenter* when `itconj`> 0.

`tertperiod`: The period of the third body in days.

`tertT0`: The time of the third body periastron passage when `itconj=0`.

`terte*cos(omega)`: This is $e_3 \cos \omega_3$.

`terte*sin(omega)`: This is $e_3 \sin \omega_3$.

`tertincl`: The inclination of the third body's orbit in degrees.

`tertOmega`: The nodal angle of the third body's orbit in degrees.

`tertQ`: The mass ratio for the third body, defined as

$$Q_3 = \frac{M_1 + M_2}{M_3}.$$

The next 8 lines should consist of pairs of numbers giving the limb darkening coefficients of the third body in the 8 filters:

```
x3(U), y3(U)
x3(B), y3(B)
x3(V), y3(V)
```

and so on.

`tertratrad`: Not yet implemented.

`hh`: The stepsize for the dynamical integrator in days. This stepsize should be roughly 400 times smaller than the binary period.

`rk1`: The value of the tidal apsidal constant for star 1 (needs to have `iGR`≥ 2).

`rk2`: The value of the tidal apsidal constant for star 2 (needs to have `iGR`≥ 2).

Next, there are 7 blocks of 9 numbers giving the orbital parameters and size parameters for bodies 4 through 10:

`P2tconj`: The time when the body in question transits or is occulted by the *barycenter* of the previous bodies in the heiarchy (if `itconj`≥ 1).

`P2period`: The period of the body in question (body 4 through 10) in days.

`P2T0`: The time of periastron passage of the body in question (body 4 through 10) when `itconj=0`.

`P2ecos`: This is $e_N \cos\omega_N$ for body N (4 through 10).

`P2esin`: This is $e_N \sin\omega_N$ for body N (4 through 10).

`P2incl`: The inclination of the orbit (in degrees) of the body in question (body 4 through 10).

`P2Omega`: The nodal angle of the orbit (in degrees) of the body in question (body 4 through 10).

`P2Q`: The mass ratio of the body in question (body 4 through 10) given by

$$Q_N = \frac{M_1 + M_2}{M_N}.$$

`P2ratrad`: The radius ratio R_N/R_1 for N (4 through 10).

After the 7 blocks given the parameters for bodies 4 through 10, you need 16 pairs of numbers giving the limb darkening coefficients of body 4 and body 5:

```
x4(U), y4(U)
x4(B), y4(B)
x4(V), y4(V)
.
.
x5(U), y5(U)
x5(B), y5(B)
x5(V), y5(V)
.
.
```

10. Appendix B: ELC.inp for Kepler-16

In this appendix and in the following ones, I give some **ELC.inp** files for various systems I have worked with. These input files should produce light and velocity curves that roughly match the observed ones. The values of **Ngap** and **NSC** have been set to zero in these inputs.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
2                        itime
   -35.00000000          t_start (if itime=2)
  3000.00000000          t_end   (if itime=2)
 0.00395693              time step in days (if itime=2)
  0.017527               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
2                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
1                        iRVfilt
  5                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  6100.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6100.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
  6700.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
 0                       Nref
   3.30010               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
2                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
    700.0000000000       T3 (Kelvin), tag t3
  4.50000000000          g3 (log(g) in c.g.s), tag g3
    0.0147338523466036   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
   300                   Nterms for fast analytic
2                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 5                       iecheck
```

```
0                        ism1
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
1                        isquare
0                        iusepot
    0.000000             usepot1
    0.000000             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
0                        ionephase
  176.700000             onephase
!
!  Control flags for orbital elements
!
 0.000000000             pshift, tag ps
1                        ikeep (1 to put primary eclipse at phase 0.0)
0                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (1 for rotation aligned with orbit)
1                        set to 1 to fit for Tconj of binary
1                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                       imag (0 for input data in mag, 1 for input data in flux)
   6                     ielite
2                        ifixgamma (0, 1, or 2)
0                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                        frac switch (>1 to enable ELCratio.* files)
0                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
        0.00000          chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
             0           jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                       Ndynwin (number of segments to integrate)
     -35.000000          Tref for dynamical integrator (Nbody >=3)
 0.10000                 hh (step size in days)
0                        iGR (1 for GR, 2 for tidal, 3 for both)
0                        set to 1 for binary+binary model (use Nbody=4)
1                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
0                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
         0.0000          chi^2 penalty for transit or secondary eclipse
0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
```

```
 0.0108574030000         Kepler contamination, tag co
1                        Iseason (1 for seasonal Kepler contamination)
 0.0226991005265         contamS0 (season 0 contamination, tag s0)
 0.0142275413185         contamS1 (season 1 contamination, tag s1)
 0.0162472750096         contamS2 (season 2 contamination, tag s2)
 0.0159845012330         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
1                        set to 1 for fast Kepler binning
  29.46600               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 0                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 3                       Nbody
!
!  Binary orbit parameters
!
    41.079297354650791   Period (days), tag pe
   -34.883873508147062   T0 (time of periastron passage), tag T0
   -34.340642894553717   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
 -99.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.1576395062876          eccentricity, tag ec
  263.3787153840582      argument of periaston in degrees, tag ar
-0.01817680727764700     e*cos(omega), tag oc
-0.15658805069282389     e*sin(omega), tag os
 0.00000000000000000     sqrt(e)*cos(omega), tag bc
 0.00000000000000000     sqrt(e)*sin(omega), tag bs
   90.3387011123486161   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
    0.0000000000000      primK (K-velocity of star 1 in km/sec), tag pk
   48.251791             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  70   40                Nalph1, Nbet1
 4465.171470000          Teff1 (K), tag T1
 0.1908300               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
0.2787320830000          fill1, tag f1
     0.0000000000000     primrad (star 1 radius in solar radii), tag pr
0.0134366048804633       frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.69168004922323600   primmass (star 1 mass in solar masses), tag pm
  90.3387011             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 0.41707335  0.48581585  limb darkening coefficients, filter 1, tag x1, y1
 0.78600000  0.09500000  limb darkening coefficients, filter 2, tag x2, y2
 0.65100000  0.29000000  limb darkening coefficients, filter 3, tag x3, y3
 0.40950000  0.17920000  limb darkening coefficients, filter 4, tag x4, y4
 0.72600000  0.14440000  limb darkening coefficients, filter 5, tag x5, y5
 0.60000000  0.00000000  limb darkening coefficients, filter 6, tag x6, y6
 0.60000000  0.00000000  limb darkening coefficients, filter 7, tag x7, y7
```

```
 0.60000000  0.00000000  limb darkening coefficients, filter 8, tag x8, y8
 0.64400  0.22400        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 1, tag b1
 -1.0000000              Latitude of spot 1, star 1 (degrees), tag b2
 -1.0000000              Longitude of spot 1, star 1 (degrees), tag b3
 -1.0000000              Angular radius of spot 1, star 1 (degrees), tag b4
-1.0000000               Temperature factor spot 2, star 1, tag b5
 -1.0000000              Latitude of spot 2, star 1 (degrees), tag b6
 -1.0000000              Longitude of spot 2, star 1 (degrees), tag b7
 -1.0000000              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.00000000               rk1 (apsidal constant, star 1), tag a1
      0.00000000000      flux star 1, filter 1 (need Nterms > 0), tag 11
      0.00000000000      flux star 1, filter 2 (need Nterms > 0), tag 12
      0.00000000000      flux star 1, filter 3 (need Nterms > 0), tag 13
      0.00000000000      flux star 1, filter 4 (need Nterms > 0), tag 14
      0.00000000000      flux star 1, filter 5 (need Nterms > 0), tag 15
      0.00000000000      flux star 1, filter 6 (need Nterms > 0), tag 16
      0.00000000000      flux star 1, filter 7 (need Nterms > 0), tag 17
      0.00000000000      flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  70   40                Nalph2, Nbet2
 3318.228810000          Teff2 (K), tag T2
  0.7431358099135200     temprat (T_2/T_1), tag te
 0.1084000               Tgrav2, tag g2
 0.5000000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.1986352280000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.0047179857336649       frac2 (fractional radius star 2:  R_2/a), tag q2
     2.8479536901918     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.0000000000000     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     0.29300424784877938 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.000000000000000     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 0.40859011  0.53835425  limb darkening coefficients, filter 1, tag z1, w1
 0.60000000  0.00000000  limb darkening coefficients, filter 2, tag z2, w2
 0.60000000  0.10000000  limb darkening coefficients, filter 3, tag z3, w3
 0.65000000  0.26900000  limb darkening coefficients, filter 4, tag z4, w4
 0.60000000  0.00000000  limb darkening coefficients, filter 5, tag z5, w5
 0.60000000  0.00000000  limb darkening coefficients, filter 6, tag z6, w6
 0.60000000  0.00000000  limb darkening coefficients, filter 7, tag z7, w7
 0.60000000  0.00000000  limb darkening coefficients, filter 8, tag z8, w8
 0.63500  0.24200        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 2, tag c1
 -1.0000000              Latitude of spot 1, star 2 (degrees), tag c2
 -1.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 -1.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
```

```
-1.0000000              Temperature factor spot 2, star 2, tag c5
 -1.0000000             Latitude of spot 2, star 2 (degrees), tag c6
 -1.0000000             Longitude of spot 2, star 2 (degrees), tag c7
 -1.0000000             Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000           beam2 (Doppler boosting factor, star 2), tag e2
0.00000000              rk2 (apsidal constant, star 2), tag a2
       0.00000000000    flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000    flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000    flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000    flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000    flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000    flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000    flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000    flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                       idint (1 for accretion disk)
 90                     Ntheta
 60                     Nradius
 2.00000000             beta_rim (opening angle in degrees), tag be
0.010000000             rinner (radius of inner hole), tag ri
0.750000000             router (radius of outer disk), tag ro
 30000.000000           Tdisk (temperature of inner edge in K) tag td
-0.750000000            xi (power-law exponent on temperature profile), tag xi
-1.0000000000           Temperature factor spot 1, disk, tag d1
 -1.000000000           Azimuth of spot 1, disk (degrees), tag d2
 -1.000000000           Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
 -1.000000000           Angular size of spot 1, disk (degrees), tag d4
-1.0000000000           Temperature factor spot 2, disk, tag d5
 -1.000000000           Azimuth of spot 2, disk (degrees), tag d6
 -1.000000000           Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 -1.000000000           Angular size of spot 2, disk (degrees), tag d8
    0.000000000         reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000       asini (projected semimajor axis in seconds)
 0.0000000000           asini error
!
!  Body 3 orbital parameters
!
    -24.2583126171200   P1Tconj, tag tj
    228.33059953114525  P1period (days), tag tt
      0.0000000000000   P1pTc (period + Tconj for body 3), tag 1p
      0.0000000000000   P1mTc (period - Tconj for body 3), tag 1m
 -99.000000000000       P1qTc (slope in P-Tc plane for body 3), tag 1q
       0.000000000000   T1off (offset in P-Tc plane for body 3), tag 1t
    -39.6825237941377   P1T0, tag tu
-0.00349605424537690    P1e*cos(omega), tag tv
-0.00759499455028933    P1e*sin(omega), tag tw
 0.0000000000000000     P1sqrt(e)*cos(omega), tag 1c
 0.0000000000000000     P1sqrt(e)*sin(omega), tag 1s
  90.037480784288675    P1incl (degrees), tag tx
  -0.040626432971065    P1Omega (degrees), tag ty
    0.00000000000000    angsum1 (P1incl + P1Omega), tag 1a
    0.00000000000000    angdiff1 (P1incl - P1Omega), tag 1d
!
!  Body 3 parameters
```

```
!
  60   20                Nalph3, Nbet3
  700.000000000          Teff3 (K), tag T3
 4.50000                 g3, tag g3
   8.23838043968         P1ratrad (radius of star 1 to body 3), tag tb
    3052.47487124650     P1Q (mass of EB to body 3 mass), tag tz
   0.0000000000000       omega3, tag o3
   0.0000000             axis_I3 (inclination of rotation axis if ialign=0), tag ci
   0.0000000             axis_beta3 (angle of rotation axis wrt orbit if ialign=0), tag cb
 0.63500000  0.13000000  limb darkening coefficients, filter 1, tag m1, n1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag m2, n2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag m3, n3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag m4, n4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag m5, n5
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag m6, n6
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag m7, n7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag m8, n8
0.00000000               rk3 (apsidal constant, body 3), tag a3
       0.00000000000     flux body 3, filter 1 (need Nterms > 0), tag 31
       0.00000000000     flux body 3, filter 2 (need Nterms > 0), tag 32
       0.00000000000     flux body 3, filter 3 (need Nterms > 0), tag 33
       0.00000000000     flux body 3, filter 4 (need Nterms > 0), tag 34
       0.00000000000     flux body 3, filter 5 (need Nterms > 0), tag 35
       0.00000000000     flux body 3, filter 6 (need Nterms > 0), tag 36
       0.00000000000     flux body 3, filter 7 (need Nterms > 0), tag 37
       0.00000000000     flux body 3, filter 8 (need Nterms > 0), tag 38
```

11. Appendix C: ELC.inp for KIO-126

KOI-126 consists of a short period binary with two M-stars that orbit an F-star with a period of around 34 days. The **ELC.inp** file below makes use of **ELCgap.inp**, **ELCSC.inp**, and **ELCdynwin.inp**, which are given after **ELC.inp**.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
2                        itime
     0.00000000          t_start (if itime=2)
  3000.00000000          t_end   (if itime=2)
 0.00200000              time step in days (if itime=2)
  0.414637               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
2                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
1                        iRVfilt
  5                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  6100.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6100.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
  6700.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
-1                       Nref
   3.30010               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
0                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
   5875.0000000000       T3 (Kelvin), tag t3
  4.50000000000          g3 (log(g) in c.g.s), tag g3
   62.3103788007894579   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
   300                   Nterms for fast analytic
2                        mandel (0 for Gimenez, 1 for Mandel & Agol)
  43                     Ngap
 1                       iecheck
```

```
0                        ism1
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
1                        isquare
0                        iusepot
    0.000000             usepot1
    0.000000             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
1                        ionephase
    0.000000             onephase
!
!  Control flags for orbital elements
!
 0.000000000             pshift, tag ps
1                        ikeep (1 to put primary eclipse at phase 0.0)
0                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (0 for rotation aligned with orbit)
1                        set to 1 to fit for Tconj of binary
1                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                       imag (0 for input data in mag, 1 for input data in flux)
   1                     ielite
1                        ifixgamma (0, 1, or 2)
0                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                        frac switch (>1 to enable ELCratio.* files)
0                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
        0.00000          chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
             0           jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 3                       Ndynwin (number of segments to integrate)
       0.000000          Tref for dynamical integrator (Nbody >=3)
 0.00500                 hh (step size in days)
3                        iGR (1 for GR, 2 for tidal, 3 for both)
0                        set to 1 for binary+binary model (use Nbody=4)
1                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
0                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
         0.0000          chi^2 penalty for transit or secondary eclipse
1 0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
```

```
 0.0084135471561         Kepler contamination, tag co
1                        Iseason (1 for seasonal Kepler contamination)
 0.0095086265061         contamS0 (season 0 contamination, tag s0)
 0.0204230522910         contamS1 (season 1 contamination, tag s1)
 0.0339052861333         contamS2 (season 2 contamination, tag s2)
 0.0090690128070         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
0                        set to 1 for fast Kepler binning
  29.46600               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 2                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 3                       Nbody
!
!  Binary orbit parameters
!
     1.736457273607043   Period (days), tag pe
    35.145207598969002   T0 (time of periastron passage), tag T0
    35.475288419805956   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
   0.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
     -99.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0122812241980          eccentricity, tag ec
  200.2513687217181      argument of periaston in degrees, tag ar
-0.01152203658910069     e*cos(omega), tag oc
-0.00425101642440999     e*sin(omega), tag os
 0.00000000000000000     sqrt(e)*cos(omega), tag bc
 0.00000000000000000     sqrt(e)*sin(omega), tag bs
   88.2397235365320967   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
    0.0000000000000      primK (K-velocity of star 1 in km/sec), tag pk
    4.637452             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  60   30                Nalph1, Nbet1
 3089.437640190          Teff1 (K), tag T1
 0.0100000               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
1.0000000000000          fill1, tag f1
     0.2539709208885     primrad (star 1 radius in solar radii), tag pr
0.0000000000000000       frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.23558816329699411   primmass (star 1 mass in solar masses), tag pm
  88.2397235             axis_I1 (inclination of rotation axis if ialign=1), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=1), tag ab
 0.60000000  0.30000000  limb darkening coefficients, filter 1, tag x1, y1
 0.78600000  0.78600000  limb darkening coefficients, filter 2, tag x2, y2
 0.65100000  0.65100000  limb darkening coefficients, filter 3, tag x3, y3
 0.40950000  0.40950000  limb darkening coefficients, filter 4, tag x4, y4
 0.72600000  0.72600000  limb darkening coefficients, filter 5, tag x5, y5
 0.60000000  0.60000000  limb darkening coefficients, filter 6, tag x6, y6
 0.60000000  0.60000000  limb darkening coefficients, filter 7, tag x7, y7
```

```
 0.60000000  0.60000000  limb darkening coefficients, filter 8, tag x8, y8
 0.60000  0.30000        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 1, tag b1
 -1.0000000              Latitude of spot 1, star 1 (degrees), tag b2
 -1.0000000              Longitude of spot 1, star 1 (degrees), tag b3
 -1.0000000              Angular radius of spot 1, star 1 (degrees), tag b4
-1.0000000               Temperature factor spot 2, star 1, tag b5
 -1.0000000              Latitude of spot 2, star 1 (degrees), tag b6
 -1.0000000              Longitude of spot 2, star 1 (degrees), tag b7
 -1.0000000              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.09037656               rk1 (apsidal constant, star 1), tag a1
       0.00000000000     flux star 1, filter 1 (need Nterms > 0), tag 11
       0.00000000000     flux star 1, filter 2 (need Nterms > 0), tag 12
       0.00000000000     flux star 1, filter 3 (need Nterms > 0), tag 13
       0.00000000000     flux star 1, filter 4 (need Nterms > 0), tag 14
       0.00000000000     flux star 1, filter 5 (need Nterms > 0), tag 15
       0.00000000000     flux star 1, filter 6 (need Nterms > 0), tag 16
       0.00000000000     flux star 1, filter 7 (need Nterms > 0), tag 17
       0.00000000000     flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  70   40                Nalph2, Nbet2
 3004.714274576          Teff2 (K), tag T2
  0.9725764441685379     temprat (T_2/T_1), tag te
 0.2500000               Tgrav2, tag g2
 0.5000000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.1986352280000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.000000000000000        frac2 (fractional radius star 2:  R_2/a), tag q2
     0.0000000000000     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.4847349011057     radsum (sum of radii in solar), tag rs
     0.0233396383174     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     0.00000000000000000 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.443809560143166     masssum (sum of masses in solar),  tag ms
   0.027676436751439     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=1), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=1), tag bb
 0.60000000  0.30000000  limb darkening coefficients, filter 1, tag z1, w1
 0.60000000  0.78600000  limb darkening coefficients, filter 2, tag z2, w2
 0.60000000  0.65100000  limb darkening coefficients, filter 3, tag z3, w3
 0.65000000  0.40950000  limb darkening coefficients, filter 4, tag z4, w4
 0.60000000  0.72600000  limb darkening coefficients, filter 5, tag z5, w5
 0.60000000  0.60000000  limb darkening coefficients, filter 6, tag z6, w6
 0.60000000  0.60000000  limb darkening coefficients, filter 7, tag z7, w7
 0.60000000  0.60000000  limb darkening coefficients, filter 8, tag z8, w8
 0.60000  0.30000        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 2, tag c1
 -1.0000000              Latitude of spot 1, star 2 (degrees), tag c2
 -1.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 -1.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
```

```
-1.0000000               Temperature factor spot 2, star 2, tag c5
 -1.0000000              Latitude of spot 2, star 2 (degrees), tag c6
 -1.0000000              Longitude of spot 2, star 2 (degrees), tag c7
 -1.0000000              Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000            beam2 (Doppler boosting factor, star 2), tag e2
0.10457466               rk2 (apsidal constant, star 2), tag a2
       0.00000000000     flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000     flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000     flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000     flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000     flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000     flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000     flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000     flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                        idint (1 for accretion disk)
 90                      Ntheta
 60                      Nradius
 2.00000000              beta_rim (opening angle in degrees), tag be
0.010000000              rinner (radius of inner hole), tag ri
0.750000000              router (radius of outer disk), tag ro
 30000.000000            Tdisk (temperature of inner edge in K) tag td
-0.750000000             xi (power-law exponent on temperature profile), tag xi
-1.0000000000            Temperature factor spot 1, disk, tag d1
 -1.000000000            Azimuth of spot 1, disk (degrees), tag d2
 -1.000000000            Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
 -1.000000000            Angular size of spot 1, disk (degrees), tag d4
-1.0000000000            Temperature factor spot 2, disk, tag d5
 -1.000000000            Azimuth of spot 2, disk (degrees), tag d6
 -1.000000000            Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 -1.000000000            Angular size of spot 2, disk (degrees), tag d8
    0.000000000          reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000        asini (projected semimajor axis in seconds)
 0.0000000000            asini error
!
!  Body 3 orbital parameters
!
     22.8416215973114    P1Tconj, tag tj
     33.98893611074559   P1period (days), tag tt
      0.0000000000000    P1pTc (period + Tconj for body 3), tag 1p
      0.0000000000000    P1mTc (period - Tconj for body 3), tag 1m
   0.000000000000        P1qTc (slope in P-Tc plane for body 3), tag 1q
     -99.000000000000    T1off (offset in P-Tc plane for body 3), tag 1t
     20.9141244042819    P1T0, tag tu
-0.19873929503925500     P1e*cos(omega), tag tv
-0.24101976401516065     P1e*sin(omega), tag tw
 0.0000000000000000      P1sqrt(e)*cos(omega), tag 1c
 0.0000000000000000      P1sqrt(e)*sin(omega), tag 1s
  92.593670117029632     P1incl (degrees), tag tx
  -8.235975302396023     P1Omega (degrees), tag ty
    0.00000000000000     angsum1 (P1incl + P1Omega), tag 1a
    0.00000000000000     angdiff1 (P1incl - P1Omega), tag 1d
!
!  Body 3 parameters
```

```
!
  60   20                 Nalph3, Nbet3
 5875.000000000           Teff3 (K), tag T3
 4.50000                  g3, tag g3
   0.12669167764          P1ratrad (radius of star 1 to body 3), tag tb
       0.34695704801      P1Q (mass of EB to body 3 mass), tag tz
   0.0000000000000        omega3, tag o3
   0.0000000              axis_I3 (inclination of rotation axis if ialign=1), tag ci
   0.0000000              axis_beta3 (angle of rotation axis wrt orbit if ialign=1), tag cb
 0.34659633  0.33023082  limb darkening coefficients, filter 1, tag m1, n1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag m2, n2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag m3, n3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag m4, n4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag m5, n5
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag m6, n6
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag m7, n7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag m8, n8
0.00000000                rk3 (apsidal constant, body 3), tag a3
       0.00000000000      flux body 3, filter 1 (need Nterms > 0), tag 31
       0.00000000000      flux body 3, filter 2 (need Nterms > 0), tag 32
       0.00000000000      flux body 3, filter 3 (need Nterms > 0), tag 33
       0.00000000000      flux body 3, filter 4 (need Nterms > 0), tag 34
       0.00000000000      flux body 3, filter 5 (need Nterms > 0), tag 35
       0.00000000000      flux body 3, filter 6 (need Nterms > 0), tag 36
       0.00000000000      flux body 3, filter 7 (need Nterms > 0), tag 37
       0.00000000000      flux body 3, filter 8 (need Nterms > 0), tag 38
```

ELCgap:

```
   -40.0000000000000        1.0038528579667
   103.2535364413333      136.3332423005667
   137.3373840602333      157.6365722278667
   158.1732388945333      169.5703745956667
   171.3264452825333      191.0007451581667
   192.4029118248334      203.8228924092666
   204.7293602014333      237.2365151872667
   239.0113325456333      257.8315400979667
   260.2402067646333      271.0384556612667
   272.6296412167333      300.
   650.                   677.1441449405667
   677.9216284218334      710.5951170835666
   711.9570580143334      744.6402925181666
   745.5029783834334      778.6198300757666
   779.4045266824334      812.1889534404667
   813.0388011782334      846.1449227955666
   846.9908751087333      879.7353496371667
   881.0902879246333      913.7102380581666
   914.5175200853333      947.6265720875666
   948.5156005962334      981.0557073882667
   982.5238486696334     1015.1691577683666
  1016.0064134190334     1048.6203525695666
  1050.2087085577332     1082.3561344577668
  1083.9049590292332     1116.6036354469668
  1117.4992664394333     1149.9556135031667
  1151.6907516543333     1169.5227931391667
  1171.1139598058332     1184.0377400489667
  1184.9685695482333     1218.0432883622668
```

```
  1219.0107327179333    1236.7022477342668
  1238.8169144009332    1251.3114133540666
  1253.0922566505333    1285.4981058258668
  1286.4349730196332    1319.0400517667667
  1320.6926361444332    1338.8987621301667
  1338.9404287968332    1352.7308836113668
  1354.4405128664332    1373.1189812419668
  1373.1606479086333    1386.9632720071668
  1387.8890685106333    1405.0516743481667
  1407.3508410148333    1420.3316612608667
  1422.2027945427333    1454.1800276905667
  1455.8309640931332    1472.6046320848668
  1474.3277987515332    1488.4389511200668
  1489.3355452873332    1758.8895181281669
  1776.    2660.
```

ELCSC.inp:

```
102.  500.
625.  1450.
```

ELCdynwin.inp:

```
0. 1500.
1750. 1800.
2650. 2720.
```

12. Appendix D: ELC.inp for Kepler-47

Here is an input file for Kepler-47, which is a 5-body system with an inner binary with a period near 8 days surrounded by three planets with periods of around 50 days, 187 days, and 303 days. As before, I have set **Ngap=0** and **NSC=0**.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
2                        itime
   -40.00000000          t_start (if itime=2)
  1430.00000000          t_end   (if itime=2)
 0.00395693              time step in days (if itime=2)
  0.100692               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
2                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
1                        iRVfilt
  5                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  3600.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6700.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
 12000.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
 1                       Nref
   0.00100               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
0                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
    500.0000000000       T3 (Kelvin), tag t3
  4.50000000000          g3 (log(g) in c.g.s), tag g3
    0.0008849510000000   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
   300                   Nterms for fast analytic
1                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 5                       iecheck
```

```
0                        ism1
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
0                        isquare
0                        iusepot
    0.000000             usepot1
    0.000000             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
0                        ionephase
    0.000000             onephase
!
!  Control flags for orbital elements
!
 0.000000000             pshift, tag ps
1                        ikeep (1 to put primary eclipse at phase 0.0)
0                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (0 for rotation aligned with orbit)
1                        set to 1 to fit for Tconj of binary
1                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                       imag (0 for input data in mag, 1 for input data in flux)
 788                     ielite
1                        ifixgamma (0, 1, or 2)
1                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                        frac switch (>1 to enable ELCratio.* files)
1                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
        0.00000          chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
             0           jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                       Ndynwin (number of segments to integrate)
     -35.000000          Tref for dynamical integrator (Nbody >=3)
 0.02000                 hh (step size in days)
1                        iGR (1 for GR, 2 for tidal, 3 for both)
0                        set to 1 for binary+binary model (use Nbody=4)
1                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
1                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
         0.0000          chi^2 penalty for transit or secondary eclipse
0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
```

```
 0.0022183000000         Kepler contamination, tag co
1                        Iseason (1 for seasonal Kepler contamination)
 0.0010108622119         contamS0 (season 0 contamination, tag s0)
 0.0228679817823         contamS1 (season 1 contamination, tag s1)
 0.0054755488070         contamS2 (season 2 contamination, tag s2)
 0.0241585488070         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
1                        set to 1 for fast Kepler binning
  29.47440               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 0                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 5                       Nbody
!
!  Binary orbit parameters
!
     7.448365101859633   Period (days), tag pe
   -30.181838538148028   T0 (time of periastron passage), tag T0
   -29.305247402989259   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
 -99.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0283264212443          eccentricity, tag ec
  225.3863366213110      argument of periaston in degrees, tag ar
-0.01989429234775372     e*cos(omega), tag oc
-0.02016440607839527     e*sin(omega), tag os
 0.00000000000000000     sqrt(e)*cos(omega), tag bc
 0.00000000000000000     sqrt(e)*sin(omega), tag bs
   89.6088153274976520   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
    0.0000000000000      primK (K-velocity of star 1 in km/sec), tag pk
   17.515982             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  40   14                Nalph1, Nbet1
 5636.000000000          Teff1 (K), tag T1
 0.2500000               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
1.0000000000000          fill1, tag f1
     0.9000000000000     primrad (star 1 radius in solar radii), tag pr
0.0534678942535153       frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.95791081136403899   primmass (star 1 mass in solar masses), tag pm
  89.6088153             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 0.49146616  0.19111933  limb darkening coefficients, filter 1, tag x1, y1
 0.83000000  0.24000000  limb darkening coefficients, filter 2, tag x2, y2
 0.60000000  0.10000000  limb darkening coefficients, filter 3, tag x3, y3
 0.60000000  0.10000000  limb darkening coefficients, filter 4, tag x4, y4
 0.60000000  0.10000000  limb darkening coefficients, filter 5, tag x5, y5
 0.60000000  0.10000000  limb darkening coefficients, filter 6, tag x6, y6
 0.60000000  0.10000000  limb darkening coefficients, filter 7, tag x7, y7
```

```
 0.60000000  0.10000000  limb darkening coefficients, filter 8, tag x8, y8
 0.63500  0.24200        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 1, tag b1
 -1.0000000              Latitude of spot 1, star 1 (degrees), tag b2
 -1.0000000              Longitude of spot 1, star 1 (degrees), tag b3
 -1.0000000              Angular radius of spot 1, star 1 (degrees), tag b4
-1.0000000               Temperature factor spot 2, star 1, tag b5
 -1.0000000              Latitude of spot 2, star 1 (degrees), tag b6
 -1.0000000              Longitude of spot 2, star 1 (degrees), tag b7
 -1.0000000              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.00000000               rk1 (apsidal constant, star 1), tag a1
      0.00000000000      flux star 1, filter 1 (need Nterms > 0), tag 11
      0.00000000000      flux star 1, filter 2 (need Nterms > 0), tag 12
      0.00000000000      flux star 1, filter 3 (need Nterms > 0), tag 13
      0.00000000000      flux star 1, filter 4 (need Nterms > 0), tag 14
      0.00000000000      flux star 1, filter 5 (need Nterms > 0), tag 15
      0.00000000000      flux star 1, filter 6 (need Nterms > 0), tag 16
      0.00000000000      flux star 1, filter 7 (need Nterms > 0), tag 17
      0.00000000000      flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  40   14                Nalph2, Nbet2
 3434.259807122          Teff2 (K), tag T2
  0.6093434718102292     temprat (T_2/T_1), tag te
 0.2500000               Tgrav2, tag g2
 1.0000000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.0050000000000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.0193531776366326       frac2 (fractional radius star 2:  R_2/a), tag q2
     2.7627449743606     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.0000000000000     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     0.35688529638570382 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.000000000000000     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 1.00000000  0.27669507  limb darkening coefficients, filter 1, tag z1, w1
 0.60000000  0.10000000  limb darkening coefficients, filter 2, tag z2, w2
 0.60000000  0.10000000  limb darkening coefficients, filter 3, tag z3, w3
 0.60000000  0.10000000  limb darkening coefficients, filter 4, tag z4, w4
 0.60000000  0.10000000  limb darkening coefficients, filter 5, tag z5, w5
 0.60000000  0.10000000  limb darkening coefficients, filter 6, tag z6, w6
 0.60000000  0.10000000  limb darkening coefficients, filter 7, tag z7, w7
 0.60000000  0.20000000  limb darkening coefficients, filter 8, tag z8, w8
 0.63500  0.24200        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 2, tag c1
 -1.0000000              Latitude of spot 1, star 2 (degrees), tag c2
 -1.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 -1.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
```

```
-1.0000000               Temperature factor spot 2, star 2, tag c5
 -1.0000000              Latitude of spot 2, star 2 (degrees), tag c6
 -1.0000000              Longitude of spot 2, star 2 (degrees), tag c7
 -1.0000000              Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000            beam2 (Doppler boosting factor, star 2), tag e2
0.00000000               rk2 (apsidal constant, star 2), tag a2
       0.00000000000     flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000     flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000     flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000     flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000     flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000     flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000     flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000     flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                        idint (1 for accretion disk)
 90                      Ntheta
 60                      Nradius
 2.00000000              beta_rim (opening angle in degrees), tag be
0.005000000              rinner (radius of inner hole), tag ri
0.750000000              router (radius of outer disk), tag ro
 30000.000000            Tdisk (temperature of inner edge in K) tag td
-0.750000000             xi (power-law exponent on temperature profile), tag xi
-1.0000000000            Temperature factor spot 1, disk, tag d1
 -1.000000000            Azimuth of spot 1, disk (degrees), tag d2
 -1.000000000            Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
 -1.000000000            Angular size of spot 1, disk (degrees), tag d4
-1.0000000000            Temperature factor spot 2, disk, tag d5
 -1.000000000            Azimuth of spot 2, disk (degrees), tag d6
 -1.000000000            Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 -1.000000000            Angular size of spot 2, disk (degrees), tag d8
    0.000000000          reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000        asini (projected semimajor axis in seconds)
 0.0000000000            asini error
!
!  Body 3 orbital parameters
!
    -31.3568557310618    P1Tconj, tag tj
     49.46275093167047   P1period (days), tag tt
      0.0000000000000    P1pTc (period + Tconj for body 3), tag 1p
      0.0000000000000    P1mTc (period - Tconj for body 3), tag 1m
 -99.000000000000        P1qTc (slope in P-Tc plane for body 3), tag 1q
       0.000000000000    T1off (offset in P-Tc plane for body 3), tag 1t
    -12.6283976665384    P1T0, tag tu
 0.01390047465352584     P1e*cos(omega), tag tv
 0.01572044536578410     P1e*sin(omega), tag tw
 0.0000000000000000      P1sqrt(e)*cos(omega), tag 1c
 0.0000000000000000      P1sqrt(e)*sin(omega), tag 1s
  89.751195736283677     P1incl (degrees), tag tx
  -0.081896648871350     P1Omega (degrees), tag ty
    0.00000000000000     angsum1 (P1incl + P1Omega), tag 1a
    0.00000000000000     angdiff1 (P1incl - P1Omega), tag 1d
!
!  Body 3 parameters
```

```
!
  60   20                Nalph3, Nbet3
  500.000000000          Teff3 (K), tag T3
 4.50000                 g3, tag g3
  33.50000000000         P1ratrad (radius of star 1 to body 3), tag tb
  202398.78025775391     P1Q (mass of EB to body 3 mass), tag tz
   0.0000000000000       omega3, tag o3
   0.0000000             axis_I3 (inclination of rotation axis if ialign=0), tag ci
   0.0000000             axis_beta3 (angle of rotation axis wrt orbit if ialign=0), tag cb
 0.63500000  0.13000000  limb darkening coefficients, filter 1, tag m1, n1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag m2, n2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag m3, n3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag m4, n4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag m5, n5
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag m6, n6
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag m7, n7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag m8, n8
0.00000000               rk3 (apsidal constant, body 3), tag a3
       0.00000000000     flux body 3, filter 1 (need Nterms > 0), tag 31
       0.00000000000     flux body 3, filter 2 (need Nterms > 0), tag 32
       0.00000000000     flux body 3, filter 3 (need Nterms > 0), tag 33
       0.00000000000     flux body 3, filter 4 (need Nterms > 0), tag 34
       0.00000000000     flux body 3, filter 5 (need Nterms > 0), tag 35
       0.00000000000     flux body 3, filter 6 (need Nterms > 0), tag 36
       0.00000000000     flux body 3, filter 7 (need Nterms > 0), tag 37
       0.00000000000     flux body 3, filter 8 (need Nterms > 0), tag 38
!
!  Body 4 orbital parameters
!
     45.4265947837720    P2Tconj, tag uj
    187.36646089933913   P2period (days), tag ut
      0.0000000000000    P2pTc (period + Tconj for body 4), tag 2p
      0.0000000000000    P2mTc (period - Tconj for body 4), tag 2m
 -99.000000000000        P2qTc (slope in P-Tc plane for body 4), tag 2q
       0.000000000000    T2off (offset in P-Tc plane for body 4), tag 2t
      0.0000000000000    P2T0, tag uu
 0.02335206324250304     P2e*cos(omega), tag uv
-0.00363516555815800     P2e*sin(omega), tag uw
 0.00000000000000000     P2sqrt(e)*cos(omega), tag 2c
 0.00000000000000000     P2sqrt(e)*sin(omega), tag 2s
  90.394932003277390     P2incl (degrees), tag ux
  -0.864310111293524     P2Omega (degrees), tag uy
    0.00000000000000     angsum2 (P2incl + P2Omega), tag 2a
    0.00000000000000     angdiff2 (P2incl - P2Omega), tag 2d
!
!  Body 4 parameters
!
    0.000000000          Teff4 (K), tag T4
 0.00000                 g4, tag g4
  14.52692731622         P2ratrad (radius of star 1 to body 4), tag ub
   32262.66656545246     P2Q (mass of EB to body 4 mass), tag uz
   0.0000000000000       omega4, tag o4
   0.0000000             axis_I4 (inclination of rotation axis if ialign=0), tag di
   0.0000000             axis_beta4 (angle of rotation axis wrt orbit if ialign=0), tag db
 0.63500000  0.13000000  limb darkening coefficients, filter 1, tag i1, j1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag i2, j2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag i3, j3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag i4, j4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag i5, j5
```

```
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag i6, j6
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag i7, j7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag i8, j8
0.00000000               rk4 (apsidal constant, body 4), tag a4
       0.00000000000     flux body 4, filter 1 (need Nterms > 0), tag 41
       0.00000000000     flux body 4, filter 2 (need Nterms > 0), tag 42
       0.00000000000     flux body 4, filter 3 (need Nterms > 0), tag 43
       0.00000000000     flux body 4, filter 4 (need Nterms > 0), tag 44
       0.00000000000     flux body 4, filter 5 (need Nterms > 0), tag 45
       0.00000000000     flux body 4, filter 6 (need Nterms > 0), tag 46
       0.00000000000     flux body 4, filter 7 (need Nterms > 0), tag 47
       0.00000000000     flux body 4, filter 8 (need Nterms > 0), tag 48
!
!  binary+binary stellar parameters
!
     0.00000000000000    bin2masssum (M_3 + M_4 in binary+binary mode), tag 91
     0.00000000000000    bin2massdiff (M_3 - M_4 in binary+binary mode), tag 92
     0.00000000000000    bin2Q (M_4/M_3 in binary+binary mode), tag 93
     0.00000000000000    bin2radsum (R_3 + R_4 in binary+binary mode), tag 94
     0.00000000000000    bin2raddiff (R_3 - R_4 in binary+binary mode), tag 95
     0.00000000000000    bin2ratrad (R_3/R_4 in binary+binary mode), tag 96
    0.000000000000000    bin2M3 (M3 in solar masses), tag 97
    0.000000000000000    bin2M4 (M4 in solar masses), tag 98
    0.000000000000000    bin2R3 (R3 in solar masses), tag 99
    0.000000000000000    bin2R4 (R4 in solar masses), tag 90
!
!  Body 5 orbital parameters
!
    -55.6117754312929    P3Tconj, tag vj
    303.20952237965838   P3period (days), tag vt
      0.0000000000000    P3pTc (period + Tconj for body 5), tag 3p
      0.0000000000000    P3mTc (period - Tconj for body 5), tag 3m
 -99.000000000000        P3qTc (slope in P-Tc plane for body 5), tag 3q
       0.000000000000    T3off (offset in P-Tc plane for body 5), tag 3t
      0.0000000000000    P3T0, tag vu
 0.02673288803833726     P3e*cos(omega), tag vv
-0.04524426356528918     P3e*sin(omega), tag vw
 0.0000000000000000      P3sqrt(e)*cos(omega), tag 3c
 0.0000000000000000      P3sqrt(e)*sin(omega), tag 3s
  90.191817454511025     P3incl (degrees), tag vx
  -1.257881446807056     P3Omega (degrees), tag vy
    0.00000000000000     angsum3 (P3incl + P3Omega), tag 3a
    0.00000000000000     angdiff3 (P3incl - P3Omega), tag 3d
!
!  Body 5 parameters
!
    0.000000000          Teff5 (K), tag T5
 0.00000                 g5, tag g5
  21.92625590754         P3ratrad (radius of star 1 to body 5), tag vb
  154777.42058209371     P3Q (mass of EB to body 5 mass), tag vz
  0.00000000             omega5, tag o5
   0.0000000             axis_I5 (inclination of rotation axis if ialign=0), tag ...
   0.0000000             axis_beta5 (angle of rotation axis wrt orbit if ialign=0), tag ...
 0.63500000  0.13000000        limb darkening coefficients, filter 1, tag k1, p1
 0.63500000  0.13000000        limb darkening coefficients, filter 2, tag k2, p2
 0.63500000  0.13000000        limb darkening coefficients, filter 3, tag k3, p3
 0.63500000  0.13000000        limb darkening coefficients, filter 4, tag k4, p4
 0.63500000  0.13000000        limb darkening coefficients, filter 5, tag k5, p5
 0.63500000  0.13000000        limb darkening coefficients, filter 6, tag k6, p6
```

```
 0.63500000  0.13000000        limb darkening coefficients, filter 7, tag k7, p7
 0.63500000  0.13000000        limb darkening coefficients, filter 8, tag k8, p8
0.00000000             rk5 (apsidal constant, body 5), tag a5
      0.00000000000     flux body 5, filter 1 (need Nterms > 0), tag 51
      0.00000000000     flux body 5, filter 2 (need Nterms > 0), tag 52
      0.00000000000     flux body 5, filter 3 (need Nterms > 0), tag 53
      0.00000000000     flux body 5, filter 4 (need Nterms > 0), tag 54
      0.00000000000     flux body 5, filter 5 (need Nterms > 0), tag 55
      0.00000000000     flux body 5, filter 6 (need Nterms > 0), tag 56
      0.00000000000     flux body 5, filter 7 (need Nterms > 0), tag 57
      0.00000000000     flux body 5, filter 8 (need Nterms > 0), tag 58
```

13. Appendix E: ELC.inp for Kepler-11

Here is an input file for Kepler-11, which is a 7-body system with a Sun-like star with 6 transiting planets. As before, I have set **Ngap=0** and **NSC=0**.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
2                        itime
   -40.00000000          t_start (if itime=2)
  1425.00000000          t_end   (if itime=2)
 0.00395693              time step in days (if itime=2)
  0.138253               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
0                        iatm (0 for black body)
1 0 0 0 0 0 0 0          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
1                        iRVfilt
  5                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  3600.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6700.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
 12000.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
 1                       Nref
   0.00100               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
0                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
    100.0000000000       T3 (Kelvin), tag t3
  4.50000000000          g3 (log(g) in c.g.s), tag g3
    0.0008849510000000   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
   300                   Nterms for fast analytic
2                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 5                       iecheck
0                        ism1
```

```
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
0                        isquare
0                        iusepot
    0.000000             usepot1
    0.000000             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
0                        ionephase
    0.000000             onephase
!
!  Control flags for orbital elements
!
 0.000000000             pshift, tag ps
1                        ikeep (1 to put primary eclipse at phase 0.0)
0                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (0 for rotation aligned with orbit)
1                        set to 1 to fit for Tconj of binary
1                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 1                       imag (0 for input data in mag, 1 for input data in flux)
 159                     ielite
1                        ifixgamma (0, 1, or 2)
0                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                        frac switch (>1 to enable ELCratio.* files)
0                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
        0.00000          chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
           0             jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                       Ndynwin (number of segments to integrate)
     -35.000000          Tref for dynamical integrator (Nbody >=3)
 0.02500                 hh (step size in days)
1                        iGR (1 for GR, 2 for tidal, 3 for both)
0                        set to 1 for binary+binary model (use Nbody=4)
1                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
2                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
         0.0000          chi^2 penalty for transit or secondary eclipse
0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
 0.0022183000000         Kepler contamination, tag co
```

```
1                        Iseason (1 for seasonal Kepler contamination)
 0.0331872369408         contamS0 (season 0 contamination, tag s0)
 0.0112255161055         contamS1 (season 1 contamination, tag s1)
 0.0323221374888         contamS2 (season 2 contamination, tag s2)
 0.0450021566753         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
1                        set to 1 for fast Kepler binning
  29.47440               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 0                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 7                       Nbody
!
!  Binary orbit parameters
!
    10.303508238587229   Period (days), tag pe
   592.693211323140645   T0 (time of periastron passage), tag T0
   589.726300000000037   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
 -99.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0682879916119          eccentricity, tag ec
  201.0956306246866      argument of periaston in degrees, tag ar
-0.06371139770938632     e*cos(omega), tag oc
-0.02457860045440757     e*sin(omega), tag os
 0.00000000000000000     sqrt(e)*cos(omega), tag bc
 0.00000000000000000     sqrt(e)*sin(omega), tag bs
   90.6649646269668210   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
    0.0000000000000      primK (K-velocity of star 1 in km/sec), tag pk
   19.825277             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  40   14                Nalph1, Nbet1
 5636.000000000          Teff1 (K), tag T1
 0.2500000               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
1.0000000000000          fill1, tag f1
     1.0522847755323     primrad (star 1 radius in solar radii), tag pr
0.0000000000000000       frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.98485492149960885   primmass (star 1 mass in solar masses), tag pm
  90.6649646             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 0.66525931  0.16458562  limb darkening coefficients, filter 1, tag x1, y1
 0.83000000  0.24000000  limb darkening coefficients, filter 2, tag x2, y2
 0.60000000  0.10000000  limb darkening coefficients, filter 3, tag x3, y3
 0.60000000  0.10000000  limb darkening coefficients, filter 4, tag x4, y4
 0.60000000  0.10000000  limb darkening coefficients, filter 5, tag x5, y5
 0.60000000  0.10000000  limb darkening coefficients, filter 6, tag x6, y6
 0.60000000  0.10000000  limb darkening coefficients, filter 7, tag x7, y7
 0.60000000  0.10000000  limb darkening coefficients, filter 8, tag x8, y8
```

```
 0.63500  0.24200        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 1, tag b1
 -1.0000000              Latitude of spot 1, star 1 (degrees), tag b2
 -1.0000000              Longitude of spot 1, star 1 (degrees), tag b3
 -1.0000000              Angular radius of spot 1, star 1 (degrees), tag b4
-1.0000000               Temperature factor spot 2, star 1, tag b5
 -1.0000000              Latitude of spot 2, star 1 (degrees), tag b6
 -1.0000000              Longitude of spot 2, star 1 (degrees), tag b7
 -1.0000000              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.00000000               rk1 (apsidal constant, star 1), tag a1
       1.00000000000     flux star 1, filter 1 (need Nterms > 0), tag 11
       0.00000000000     flux star 1, filter 2 (need Nterms > 0), tag 12
       0.00000000000     flux star 1, filter 3 (need Nterms > 0), tag 13
       0.00000000000     flux star 1, filter 4 (need Nterms > 0), tag 14
       0.00000000000     flux star 1, filter 5 (need Nterms > 0), tag 15
       0.00000000000     flux star 1, filter 6 (need Nterms > 0), tag 16
       0.00000000000     flux star 1, filter 7 (need Nterms > 0), tag 17
       0.00000000000     flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  40   14                Nalph2, Nbet2
 3434.259807122          Teff2 (K), tag T2
  0.6093434718102292     temprat (T_2/T_1), tag te
 0.2500000               Tgrav2, tag g2
 1.0000000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.0050000000000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.000000000000000        frac2 (fractional radius star 2:  R_2/a), tag q2
    63.9248937537798     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.0000000000000     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     0.00000433536807809 Q (M_2/M_1), tag ma
   0.000004269708588     secmass (star 2 mass in solar masses), tag sm
   0.000000000000000     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 1.00000000  0.27669507  limb darkening coefficients, filter 1, tag z1, w1
 0.60000000  0.10000000  limb darkening coefficients, filter 2, tag z2, w2
 0.60000000  0.10000000  limb darkening coefficients, filter 3, tag z3, w3
 0.60000000  0.10000000  limb darkening coefficients, filter 4, tag z4, w4
 0.60000000  0.10000000  limb darkening coefficients, filter 5, tag z5, w5
 0.60000000  0.10000000  limb darkening coefficients, filter 6, tag z6, w6
 0.60000000  0.10000000  limb darkening coefficients, filter 7, tag z7, w7
 0.60000000  0.20000000  limb darkening coefficients, filter 8, tag z8, w8
 0.63500  0.24200        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 2, tag c1
 -1.0000000              Latitude of spot 1, star 2 (degrees), tag c2
 -1.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 -1.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
-1.0000000               Temperature factor spot 2, star 2, tag c5
```

```
 -1.0000000              Latitude of spot 2, star 2 (degrees), tag c6
 -1.0000000              Longitude of spot 2, star 2 (degrees), tag c7
 -1.0000000              Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000            beam2 (Doppler boosting factor, star 2), tag e2
0.00000000               rk2 (apsidal constant, star 2), tag a2
       0.00001000000     flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000     flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000     flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000     flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000     flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000     flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000     flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000     flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                        idint (1 for accretion disk)
 90                      Ntheta
 60                      Nradius
 2.00000000              beta_rim (opening angle in degrees), tag be
0.005000000              rinner (radius of inner hole), tag ri
0.750000000              router (radius of outer disk), tag ro
 30000.000000            Tdisk (temperature of inner edge in K) tag td
-0.750000000             xi (power-law exponent on temperature profile), tag xi
-1.0000000000            Temperature factor spot 1, disk, tag d1
 -1.000000000            Azimuth of spot 1, disk (degrees), tag d2
 -1.000000000            Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
 -1.000000000            Angular size of spot 1, disk (degrees), tag d4
-1.0000000000            Temperature factor spot 2, disk, tag d5
 -1.000000000            Azimuth of spot 2, disk (degrees), tag d6
 -1.000000000            Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 -1.000000000            Angular size of spot 2, disk (degrees), tag d8
    0.000000000          reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000        asini (projected semimajor axis in seconds)
 0.0000000000            asini error
!
!  Body 3 orbital parameters
!
    583.3473864174413    P1Tconj, tag tj
     13.02499017089919   P1period (days), tag tt
      0.0000000000000    P1pTc (period + Tconj for body 3), tag 1p
      0.0000000000000    P1mTc (period - Tconj for body 3), tag 1m
 -99.000000000000        P1qTc (slope in P-Tc plane for body 3), tag 1q
       0.000000000000    T1off (offset in P-Tc plane for body 3), tag 1t
    -12.6283976665384    P1T0, tag tu
-0.06589179705461046     P1e*cos(omega), tag tv
-0.03781181834184066     P1e*sin(omega), tag tw
 0.0000000000000000      P1sqrt(e)*cos(omega), tag 1c
 0.0000000000000000      P1sqrt(e)*sin(omega), tag 1s
  90.709092793977362     P1incl (degrees), tag tx
  -1.389551526460250     P1Omega (degrees), tag ty
    0.00000000000000     angsum1 (P1incl + P1Omega), tag 1a
    0.00000000000000     angdiff1 (P1incl - P1Omega), tag 1d
!
!  Body 3 parameters
!
```

```
  60   20               Nalph3, Nbet3
  100.000000000         Teff3 (K), tag T3
 4.50000                g3, tag g3
  39.64266926130        P1ratrad (radius of star 1 to body 3), tag tb
       3.32355606702    P1Q (mass of EB to body 3 mass), tag tz
   0.0000000000000      omega3, tag o3
   0.0000000            axis_I3 (inclination of rotation axis if ialign=0), tag ci
   0.0000000            axis_beta3 (angle of rotation axis wrt orbit if ialign=0), tag cb
 0.63500000  0.13000000  limb darkening coefficients, filter 1, tag m1, n1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag m2, n2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag m3, n3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag m4, n4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag m5, n5
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag m6, n6
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag m7, n7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag m8, n8
0.00000000              rk3 (apsidal constant, body 3), tag a3
       0.00000000000    flux body 3, filter 1 (need Nterms > 0), tag 31
       0.00000000000    flux body 3, filter 2 (need Nterms > 0), tag 32
       0.00000000000    flux body 3, filter 3 (need Nterms > 0), tag 33
       0.00000000000    flux body 3, filter 4 (need Nterms > 0), tag 34
       0.00000000000    flux body 3, filter 5 (need Nterms > 0), tag 35
       0.00000000000    flux body 3, filter 6 (need Nterms > 0), tag 36
       0.00000000000    flux body 3, filter 7 (need Nterms > 0), tag 37
       0.00000000000    flux body 3, filter 8 (need Nterms > 0), tag 38
!
!  Body 4 orbital parameters
!
    616.7774427903546    P2Tconj, tag uj
     22.68995350858421   P2period (days), tag ut
      0.0000000000000    P2pTc (period + Tconj for body 4), tag 2p
      0.0000000000000    P2mTc (period - Tconj for body 4), tag 2m
 -99.000000000000        P2qTc (slope in P-Tc plane for body 4), tag 2q
       0.000000000000    T2off (offset in P-Tc plane for body 4), tag 2t
      0.0000000000000    P2T0, tag uu
-0.04842247532786274     P2e*cos(omega), tag uv
-0.05006606544154608     P2e*sin(omega), tag uw
 0.0000000000000000      P2sqrt(e)*cos(omega), tag 2c
 0.0000000000000000      P2sqrt(e)*sin(omega), tag 2s
  90.590084492940335     P2incl (degrees), tag ux
  -0.903157745542152     P2Omega (degrees), tag uy
    0.00000000000000     angsum2 (P2incl + P2Omega), tag 2a
    0.00000000000000     angdiff2 (P2incl - P2Omega), tag 2d
!
!  Body 4 parameters
!
  100.000000000         Teff4 (K), tag T4
 0.00000                g4, tag g4
  35.55118861305        P2ratrad (radius of star 1 to body 4), tag ub
       0.56545459259    P2Q (mass of EB to body 4 mass), tag uz
   0.0000000000000      omega4, tag o4
   0.0000000            axis_I4 (inclination of rotation axis if ialign=0), tag di
   0.0000000            axis_beta4 (angle of rotation axis wrt orbit if ialign=0), tag db
 0.63500000  0.13000000  limb darkening coefficients, filter 1, tag i1, j1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag i2, j2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag i3, j3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag i4, j4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag i5, j5
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag i6, j6
```

```
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag i7, j7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag i8, j8
0.00000000               rk4 (apsidal constant, body 4), tag a4
       0.00000000000     flux body 4, filter 1 (need Nterms > 0), tag 41
       0.00000000000     flux body 4, filter 2 (need Nterms > 0), tag 42
       0.00000000000     flux body 4, filter 3 (need Nterms > 0), tag 43
       0.00000000000     flux body 4, filter 4 (need Nterms > 0), tag 44
       0.00000000000     flux body 4, filter 5 (need Nterms > 0), tag 45
       0.00000000000     flux body 4, filter 6 (need Nterms > 0), tag 46
       0.00000000000     flux body 4, filter 7 (need Nterms > 0), tag 47
       0.00000000000     flux body 4, filter 8 (need Nterms > 0), tag 48
!
!  binary+binary stellar parameters
!
     0.00000000000000    bin2masssum (M_3 + M_4 in binary+binary mode), tag 91
     0.00000000000000    bin2massdiff (M_3 - M_4 in binary+binary mode), tag 92
     0.00000000000000    bin2Q (M_4/M_3 in binary+binary mode), tag 93
     0.00000000000000    bin2radsum (R_3 + R_4 in binary+binary mode), tag 94
     0.00000000000000    bin2raddiff (R_3 - R_4 in binary+binary mode), tag 95
     0.00000000000000    bin2ratrad (R_3/R_4 in binary+binary mode), tag 96
    0.000000000000000    bin2M3 (M3 in solar masses), tag 97
    0.000000000000000    bin2M4 (M4 in solar masses), tag 98
    0.000000000000000    bin2R3 (R3 in solar radii), tag 99
    0.000000000000000    bin2R4 (R4 in solar radii), tag 90
!
!  Body 5 orbital parameters
!
    595.0798768945198    P3Tconj, tag vj
     31.99575539651168   P3period (days), tag vt
      0.0000000000000    P3pTc (period + Tconj for body 5), tag 3p
      0.0000000000000    P3mTc (period - Tconj for body 5), tag 3m
 -99.000000000000        P3qTc (slope in P-Tc plane for body 5), tag 3q
       0.000000000000    T3off (offset in P-Tc plane for body 5), tag 3t
      0.0000000000000    P3T0, tag vu
-0.04078511991614495     P3e*cos(omega), tag vv
-0.05225168110570375     P3e*sin(omega), tag vw
 0.00000000000000000     P3sqrt(e)*cos(omega), tag 3c
 0.00000000000000000     P3sqrt(e)*sin(omega), tag 3s
  88.949339733688404     P3incl (degrees), tag vx
  -2.363233329233688     P3Omega (degrees), tag vy
    0.00000000000000     angsum3 (P3incl + P3Omega), tag 3a
    0.00000000000000     angdiff3 (P3incl - P3Omega), tag 3d
!
!  Body 5 parameters
!
  100.000000000          Teff5 (K), tag T5
 0.00000                 g5, tag g5
  26.95521770427         P3ratrad (radius of star 1 to body 5), tag vb
       1.76503282147     P3Q (mass of EB to body 5 mass), tag vz
  0.00000000             omega5, tag o5
   0.0000000             axis_I5 (inclination of rotation axis if ialign=0), tag ...
   0.0000000             axis_beta5 (angle of rotation axis wrt orbit if ialign=0), tag ...
 0.63500000  0.13000000        limb darkening coefficients, filter 1, tag k1, p1
 0.63500000  0.13000000        limb darkening coefficients, filter 2, tag k2, p2
 0.63500000  0.13000000        limb darkening coefficients, filter 3, tag k3, p3
 0.63500000  0.13000000        limb darkening coefficients, filter 4, tag k4, p4
 0.63500000  0.13000000        limb darkening coefficients, filter 5, tag k5, p5
 0.63500000  0.13000000        limb darkening coefficients, filter 6, tag k6, p6
 0.63500000  0.13000000        limb darkening coefficients, filter 7, tag k7, p7
```

```
 0.63500000  0.13000000        limb darkening coefficients, filter 8, tag k8, p8
0.00000000              rk5 (apsidal constant, body 5), tag a5
       0.00000000000     flux body 5, filter 1 (need Nterms > 0), tag 51
       0.00000000000     flux body 5, filter 2 (need Nterms > 0), tag 52
       0.00000000000     flux body 5, filter 3 (need Nterms > 0), tag 53
       0.00000000000     flux body 5, filter 4 (need Nterms > 0), tag 54
       0.00000000000     flux body 5, filter 5 (need Nterms > 0), tag 55
       0.00000000000     flux body 5, filter 6 (need Nterms > 0), tag 56
       0.00000000000     flux body 5, filter 7 (need Nterms > 0), tag 57
       0.00000000000     flux body 5, filter 8 (need Nterms > 0), tag 58
!
!  Body 6 orbital parameters
!
    618.2333213360295    P4Tconj, tag wj
     46.68260943009862   P4period (days), tag wt
      0.0000000000000    P4pTc (period + Tconj for body 6), tag 4p
      0.0000000000000    P4mTc (period - Tconj for body 6), tag 4m
 -99.000000000000        P4qTc (slope in P-Tc plane for body 6), tag 4q
       0.000000000000    T4off (offset in P-Tc plane for body 6), tag 4t
      0.0000000000000    P4T0, tag wu
-0.03734112077419934     P4e*cos(omega), tag wv
-0.01256891339696917     P4e*sin(omega), tag ww
-0.03734112077419934     P4sqrt(e)*cos(omega), tag 4c
-0.01256891339696917     P4sqrt(e)*sin(omega), tag 4s
  89.542366834335482     P4incl (degrees), tag wx
  -5.959567425640310     P4Omega (degrees), tag wy
    0.00000000000000     angsum4 (P4incl + P4Omega), tag 4a
    0.00000000000000     angdiff4 (P4incl - P4Omega), tag 4d
!
!  Body 6 parameters
!
  100.000000000          Teff6 (K), tag T6
 0.00000                 g6, tag g6
  45.01562108785         P4ratrad (radius of star 1 to body 6), tag wb
       0.55673183217     P4Q (mass of EB to body 6 mass), tag wz
  0.00000000             omega6, tag o6
   0.0000000             axis_I6 (inclination of rotation axis if ialign=0), tag ...
   0.0000000             axis_beta6 (angle of rotation axis wrt orbit if ialign=0), tag ...
 0.63500  0.13000        limb darkening coefficients, filter 1, tag ..., ...
 0.63500  0.13000        limb darkening coefficients, filter 2, tag ..., ...
 0.63500  0.13000        limb darkening coefficients, filter 3, tag ..., ...
 0.63500  0.13000        limb darkening coefficients, filter 4, tag ..., ...
 0.63500  0.13000        limb darkening coefficients, filter 5, tag ..., ...
 0.63500  0.13000        limb darkening coefficients, filter 6, tag ..., ...
 0.63500  0.13000        limb darkening coefficients, filter 7, tag ..., ...
 0.63500  0.13000        limb darkening coefficients, filter 8, tag ..., ...
0.00000000               rk6 (apsidal constant, body 6), tag a6
       0.00000000000     flux body 6, filter 1 (need Nterms > 0), tag 61
       0.00000000000     flux body 6, filter 2 (need Nterms > 0), tag 62
       0.00000000000     flux body 6, filter 3 (need Nterms > 0), tag 63
       0.00000000000     flux body 6, filter 4 (need Nterms > 0), tag 64
       0.00000000000     flux body 6, filter 5 (need Nterms > 0), tag 65
       0.00000000000     flux body 6, filter 6 (need Nterms > 0), tag 66
       0.00000000000     flux body 6, filter 7 (need Nterms > 0), tag 67
       0.00000000000     flux body 6, filter 8 (need Nterms > 0), tag 68
!
!  Body 7 orbital parameters
!
    593.8049887701468    P5Tconj, tag xj
```

```
   118.37911366989636   P5period (days), tag xt
     0.0000000000000    P5pTc (period + Tconj for body 7), tag 5p
     0.0000000000000    P5mTc (period - Tconj for body 7), tag 5m
 -99.000000000000       P5qTc (slope in P-Tc plane for body 7), tag 5q
      0.000000000000    T5off (offset in P-Tc plane for body 7), tag 5t
     0.0000000000000    P5T0, tag xu
 0.04102552064562547    P5e*cos(omega), tag xv
-0.00336587870815260    P5e*sin(omega), tag xw
 0.00000000000000000    P5sqrt(e)*cos(omega), tag 5c
 0.00000000000000000    P5sqrt(e)*sin(omega), tag 5s
  89.936899717476010    P5incl (degrees), tag xx
   0.475897843751077    P5Omega (degrees), tag xy
    0.00000000000000    angsum5 (P5incl + P5Omega), tag 5a
    0.00000000000000    angdiff5 (P5incl - P5Omega), tag 5d
!
!  Body 7 parameters
!
  100.000000000         Teff7 (K), tag T7
 0.00000                g7, tag g7
  33.60865656234        P5ratrad (radius of star 1 to body 7), tag xb
      5.67287092927     P5Q (mass of EB to body 7 mass), tag xz
  0.00000000            omega7, tag o7
   0.0000000            axis_I7 (inclination of rotation axis if ialign=0), tag ...
   0.0000000            axis_beta7 (angle of rotation axis wrt orbit if ialign=0), tag ...
 0.63500  0.13000       limb darkening coefficients, filter 1, tag ..., ...
 0.63500  0.13000       limb darkening coefficients, filter 2, tag ..., ...
 0.63500  0.13000       limb darkening coefficients, filter 3, tag ..., ...
 0.63500  0.13000       limb darkening coefficients, filter 4, tag ..., ...
 0.63500  0.13000       limb darkening coefficients, filter 5, tag ..., ...
 0.63500  0.13000       limb darkening coefficients, filter 6, tag ..., ...
 0.63500  0.13000       limb darkening coefficients, filter 7, tag ..., ...
 0.63500  0.13000       limb darkening coefficients, filter 8, tag ..., ...
0.00000000              rk7 (apsidal constant, body 7), tag a7
      0.00000000000     flux body 7, filter 1 (need Nterms > 0), tag 71
      0.00000000000     flux body 7, filter 2 (need Nterms > 0), tag 72
      0.00000000000     flux body 7, filter 3 (need Nterms > 0), tag 73
      0.00000000000     flux body 7, filter 4 (need Nterms > 0), tag 74
      0.00000000000     flux body 7, filter 5 (need Nterms > 0), tag 75
      0.00000000000     flux body 7, filter 6 (need Nterms > 0), tag 76
      0.00000000000     flux body 7, filter 7 (need Nterms > 0), tag 77
      0.00000000000     flux body 7, filter 8 (need Nterms > 0), tag 78
```

14. Appendix F: ELC.inp for EPIC 220204960

Here is an input file for EPIC 220204960, which is a quadruple system with two close binaries that orbit each other with a much longer orbital period.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
2                        itime
  2391.90000000          t_start (if itime=2)
  2730.00000000          t_end   (if itime=2)
 0.00395693              time step in days (if itime=2)
  0.107315               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
2                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
1                        iRVfilt
  5                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  6100.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6100.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
  6700.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
-1                       Nref
   3.30010               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
2                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
   3588.7996077220       T3 (Kelvin), tag t3
  4.50000000000          g3 (log(g) in c.g.s), tag g3
    0.8359564746845815   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
   300                   Nterms for fast analytic
1                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 0                       iecheck
0                        ism1
```

```
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
1                        isquare
0                        iusepot
    0.000000             usepot1
    0.000000             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
0                        ionephase
    0.000000             onephase
!
!  Control flags for orbital elements
!
 0.000000000             pshift, tag ps
0                        ikeep (1 to put primary eclipse at phase 0.0)
0                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (0 for rotation aligned with orbit)
1                        set to 1 to fit for Tconj of binary
1                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                       imag (0 for input data in mag, 1 for input data in flux)
   1                     ielite
2                        ifixgamma (0, 1, or 2)
1                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                        frac switch (>1 to enable ELCratio.* files)
1                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
        0.00000          chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
            0            jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                       Ndynwin (number of segments to integrate)
    2401.900000          Tref for dynamical integrator (Nbody >=3)
 0.02500                 hh (step size in days)
3                        iGR (1 for GR, 2 for tidal, 3 for both)
1                        set to 1 for binary+binary model (use Nbody=4)
1                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
0                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
         0.0000          chi^2 penalty for transit or secondary eclipse
0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
 0.9811103772740         Kepler contamination, tag co
```

```
0                       Iseason (1 for seasonal Kepler contamination)
 0.0225912629319        contamS0 (season 0 contamination, tag s0)
 0.0152760155080        contamS1 (season 1 contamination, tag s1)
 0.0139880417402        contamS2 (season 2 contamination, tag s2)
 0.0167317088482        contamS3 (season 3 contamination, tag s3)
   0                    Nseg (number of segments to fit forcontamination)
1                       set to 1 for fast Kepler binning
  29.46600              bin size for light curves (minutes)
   0.00000              bin size for RV curves (minutes)
 0                      NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 4                      Nbody
!
!  Binary orbit parameters
!
    13.274007176856305   Period (days), tag pe
  2400.649703972568204   T0 (time of periastron passage), tag T0
  2401.859627970774454   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
  -9.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0936535852164          eccentricity, tag ec
  230.7524485423807      argument of periaston in degrees, tag ar
-0.05925202300130523     e*cos(omega), tag oc
-0.07252717969241446     e*sin(omega), tag os
 0.00000000000000000     sqrt(e)*cos(omega), tag bc
 0.00000000000000000     sqrt(e)*sin(omega), tag bs
   90.2181483388661292   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
    0.0000000000000      primK (K-velocity of star 1 in km/sec), tag pk
   22.151868             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  60   30                Nalph1, Nbet1
 3543.416705819          Teff1 (K), tag T1
 0.0100000               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
1.0000000000000          fill1, tag f1
     0.0000000000000     primrad (star 1 radius in solar radii), tag pr
0.000000000000000        frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.00000000000000000   primmass (star 1 mass in solar masses), tag pm
  90.2181483             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 0.45274974  0.49002035  limb darkening coefficients, filter 1, tag x1, y1
 0.78600000  0.37616206  limb darkening coefficients, filter 2, tag x2, y2
 0.65100000  0.65100000  limb darkening coefficients, filter 3, tag x3, y3
 0.40950000  0.40950000  limb darkening coefficients, filter 4, tag x4, y4
 0.72600000  0.72600000  limb darkening coefficients, filter 5, tag x5, y5
 0.60000000  0.60000000  limb darkening coefficients, filter 6, tag x6, y6
 0.60000000  0.60000000  limb darkening coefficients, filter 7, tag x7, y7
 0.60000000  0.60000000  limb darkening coefficients, filter 8, tag x8, y8
```

```
 0.00000  0.00000        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 1, tag b1
 -1.0000000              Latitude of spot 1, star 1 (degrees), tag b2
 -1.0000000              Longitude of spot 1, star 1 (degrees), tag b3
 -1.0000000              Angular radius of spot 1, star 1 (degrees), tag b4
-1.0000000               Temperature factor spot 2, star 1, tag b5
 -1.0000000              Latitude of spot 2, star 1 (degrees), tag b6
 -1.0000000              Longitude of spot 2, star 1 (degrees), tag b7
 -1.0000000              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.17431290               rk1 (apsidal constant, star 1), tag a1
      0.00000000000      flux star 1, filter 1 (need Nterms > 0), tag 11
      0.00000000000      flux star 1, filter 2 (need Nterms > 0), tag 12
      0.00000000000      flux star 1, filter 3 (need Nterms > 0), tag 13
      0.00000000000      flux star 1, filter 4 (need Nterms > 0), tag 14
      0.00000000000      flux star 1, filter 5 (need Nterms > 0), tag 15
      0.00000000000      flux star 1, filter 6 (need Nterms > 0), tag 16
      0.00000000000      flux star 1, filter 7 (need Nterms > 0), tag 17
      0.00000000000      flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  70   40                Nalph2, Nbet2
 3474.798848285          Teff2 (K), tag T2
  0.0000000000000000     temprat (T_2/T_1), tag te
 0.2500000               Tgrav2, tag g2
 0.5000000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.1986352280000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.0000000000000000       frac2 (fractional radius star 2:  R_2/a), tag q2
     1.3569899785911     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.7703361751352     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     0.86515217965788138 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.827777502004654     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 0.65408849  0.35807919  limb darkening coefficients, filter 1, tag z1, w1
 0.60000000  0.78600000  limb darkening coefficients, filter 2, tag z2, w2
 0.60000000  0.65100000  limb darkening coefficients, filter 3, tag z3, w3
 0.65000000  0.40950000  limb darkening coefficients, filter 4, tag z4, w4
 0.60000000  0.72600000  limb darkening coefficients, filter 5, tag z5, w5
 0.60000000  0.60000000  limb darkening coefficients, filter 6, tag z6, w6
 0.60000000  0.60000000  limb darkening coefficients, filter 7, tag z7, w7
 0.60000000  0.60000000  limb darkening coefficients, filter 8, tag z8, w8
 0.63500  0.00000        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 2, tag c1
 -1.0000000              Latitude of spot 1, star 2 (degrees), tag c2
 -1.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 -1.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
-1.0000000               Temperature factor spot 2, star 2, tag c5
```

```
 -1.0000000              Latitude of spot 2, star 2 (degrees), tag c6
 -1.0000000              Longitude of spot 2, star 2 (degrees), tag c7
 -1.0000000              Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000            beam2 (Doppler boosting factor, star 2), tag e2
0.07851336               rk2 (apsidal constant, star 2), tag a2
      0.00000000000      flux star 2, filter 1 (need Nterms > 0), tag 21
      0.00000000000      flux star 2, filter 2 (need Nterms > 0), tag 22
      0.00000000000      flux star 2, filter 3 (need Nterms > 0), tag 23
      0.00000000000      flux star 2, filter 4 (need Nterms > 0), tag 24
      0.00000000000      flux star 2, filter 5 (need Nterms > 0), tag 25
      0.00000000000      flux star 2, filter 6 (need Nterms > 0), tag 26
      0.00000000000      flux star 2, filter 7 (need Nterms > 0), tag 27
      0.00000000000      flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                        idint (1 for accretion disk)
 90                      Ntheta
 60                      Nradius
 2.00000000              beta_rim (opening angle in degrees), tag be
0.010000000              rinner (radius of inner hole), tag ri
0.750000000              router (radius of outer disk), tag ro
 30000.000000            Tdisk (temperature of inner edge in K) tag td
-0.750000000             xi (power-law exponent on temperature profile), tag xi
-1.0000000000            Temperature factor spot 1, disk, tag d1
 -1.000000000            Azimuth of spot 1, disk (degrees), tag d2
 -1.000000000            Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
 -1.000000000            Angular size of spot 1, disk (degrees), tag d4
-1.0000000000            Temperature factor spot 2, disk, tag d5
 -1.000000000            Azimuth of spot 2, disk (degrees), tag d6
 -1.000000000            Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 -1.000000000            Angular size of spot 2, disk (degrees), tag d8
   0.000000000           reference phase for disk fraction
!
!  Pulsar parameters
!
   0.00000000000         asini (projected semimajor axis in seconds)
 0.0000000000            asini error
!
!  Body 3 orbital parameters
!
   949.2038458156798     P1Tconj, tag tj
   300.23305749193048    P1period (days), tag tt
     0.0000000000000     P1pTc (period + Tconj for body 3), tag 1p
     0.0000000000000     P1mTc (period - Tconj for body 3), tag 1m
  0.000046809074         P1qTc (slope in P-Tc plane for body 3), tag 1q
    949.196819000000     T1off (offset in P-Tc plane for body 3), tag 1t
   937.5502253517336     P1T0, tag tu
-0.17135566281901160     P1e*cos(omega), tag tv
-0.29845404590710983     P1e*sin(omega), tag tw
 0.0000000000000000      P1sqrt(e)*cos(omega), tag 1c
 0.0000000000000000      P1sqrt(e)*sin(omega), tag 1s
  97.17729595669695      P1incl (degrees), tag tx
  -3.923400994434687     P1Omega (degrees), tag ty
  93.25389496223501      angsum1 (P1incl + P1Omega), tag 1a
 101.10069695110437      angdiff1 (P1incl - P1Omega), tag 1d
!
!  Body 3 parameters
!
```

```
 60   20               Nalph3, Nbet3
 3588.799607722        Teff3 (K), tag T3
 4.50000               g3, tag g3
   1.09372507025       P1ratrad (radius of star 1 to body 3), tag tb
       2.28653399645   P1Q (mass of EB to body 3 mass), tag tz
   1.0000000000000     omega3, tag o3
   0.0000000           axis_I3 (inclination of rotation axis if ialign=0), tag ci
   0.0000000           axis_beta3 (angle of rotation axis wrt orbit if ialign=0), tag cb
 0.81397975  0.26559821  limb darkening coefficients, filter 1, tag m1, n1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag m2, n2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag m3, n3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag m4, n4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag m5, n5
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag m6, n6
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag m7, n7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag m8, n8
0.07833569               rk3 (apsidal constant, body 3), tag a3
       0.00000000000     flux body 3, filter 1 (need Nterms > 0), tag 31
       0.00000000000     flux body 3, filter 2 (need Nterms > 0), tag 32
       0.00000000000     flux body 3, filter 3 (need Nterms > 0), tag 33
       0.00000000000     flux body 3, filter 4 (need Nterms > 0), tag 34
       0.00000000000     flux body 3, filter 5 (need Nterms > 0), tag 35
       0.00000000000     flux body 3, filter 6 (need Nterms > 0), tag 36
       0.00000000000     flux body 3, filter 7 (need Nterms > 0), tag 37
       0.00000000000     flux body 3, filter 8 (need Nterms > 0), tag 38
!
!  Body 4 orbital parameters
!
   2403.0234615847185    P2Tconj, tag uj
     14.42587292855792   P2period (days), tag ut
      0.0000000000000    P2pTc (period + Tconj for body 4), tag 2p
      0.0000000000000    P2mTc (period - Tconj for body 4), tag 2m
  -9.000000000000        P2qTc (slope in P-Tc plane for body 4), tag 2q
       0.000000000000    T2off (offset in P-Tc plane for body 4), tag 2t
      0.0000000000000    P2T0, tag uu
-0.03373479695732393     P2e*cos(omega), tag uv
-0.05103108380660880     P2e*sin(omega), tag uw
 0.00000000000000000     P2sqrt(e)*cos(omega), tag 2c
 0.00000000000000000     P2sqrt(e)*sin(omega), tag 2s
  89.845108908126250     P2incl (degrees), tag ux
  -2.766994921739571     P2Omega (degrees), tag uy
    0.00000000000000     angsum2 (P2incl + P2Omega), tag 2a
    0.00000000000000     angdiff2 (P2incl - P2Omega), tag 2d
!
!  Body 4 parameters
!
 3468.274152601        Teff4 (K), tag T4
 4.50000               g4, tag g4
   1.15967938991       P2ratrad (radius of star 1 to body 4), tag ub
       2.29136323212   P2Q (mass of EB to body 4 mass), tag uz
   1.0000000000000     omega4, tag o4
   0.0000000           axis_I4 (inclination of rotation axis if ialign=0), tag di
   0.0000000           axis_beta4 (angle of rotation axis wrt orbit if ialign=0), tag db
 0.75432699  0.13796238  limb darkening coefficients, filter 1, tag i1, j1
 0.63500000  0.13000000  limb darkening coefficients, filter 2, tag i2, j2
 0.63500000  0.13000000  limb darkening coefficients, filter 3, tag i3, j3
 0.63500000  0.13000000  limb darkening coefficients, filter 4, tag i4, j4
 0.63500000  0.13000000  limb darkening coefficients, filter 5, tag i5, j5
 0.63500000  0.13000000  limb darkening coefficients, filter 6, tag i6, j6
```

```
 0.63500000  0.13000000  limb darkening coefficients, filter 7, tag i7, j7
 0.63500000  0.13000000  limb darkening coefficients, filter 8, tag i8, j8
0.04603732               rk4 (apsidal constant, body 4), tag a4
       0.00000000000     flux body 4, filter 1 (need Nterms > 0), tag 41
       0.00000000000     flux body 4, filter 2 (need Nterms > 0), tag 42
       0.00000000000     flux body 4, filter 3 (need Nterms > 0), tag 43
       0.00000000000     flux body 4, filter 4 (need Nterms > 0), tag 44
       0.00000000000     flux body 4, filter 5 (need Nterms > 0), tag 45
       0.00000000000     flux body 4, filter 6 (need Nterms > 0), tag 46
       0.00000000000     flux body 4, filter 7 (need Nterms > 0), tag 47
       0.00000000000     flux body 4, filter 8 (need Nterms > 0), tag 48
!
!  binary+binary stellar parameters
!
     0.82751255280055    bin2masssum (M_3 + M_4 in binary+binary mode), tag 91
     0.00000000000000    bin2massdiff (M_3 - M_4 in binary+binary mode), tag 92
     0.99311768665310    bin2Q (M_4/M_3 in binary+binary mode), tag 93
     0.78729959010186    bin2radsum (R_3 + R_4 in binary+binary mode), tag 94
     0.00000000000000    bin2raddiff (R_3 - R_4 in binary+binary mode), tag 95
     0.88795834387734    bin2ratrad (R_3/R_4 in binary+binary mode), tag 96
    0.000000000000000    bin2M3 (M3 in solar masses), tag 97
    0.000000000000000    bin2M4 (M4 in solar masses), tag 98
    0.000000000000000    bin2R3 (R3 in solar masses), tag 99
    0.000000000000000    bin2R4 (R4 in solar masses), tag 90
```

15. Appendix G: ELC.inp for HW Per

HW Per is a close binary in which the two stars both nearly fill their Roche lobes.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
0                        itime
     0.00000000          t_start (if itime=2)
     0.00000000          t_end   (if itime=2)
 0.00000000              time step in days (if itime=2)
  2.000000               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
1                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
3                        iRVfilt
  3                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  3600.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6700.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
 12000.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
 5                       Nref
   0.00100               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
0                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
  -4400.0000000000       T3 (Kelvin), tag t3
 -3.37000000000          g3 (log(g) in c.g.s), tag g3
   -1.0000000000000000   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
     0                   Nterms for fast analytic
0                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 1                       iecheck
1                        ism1
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
```

```
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
1                        isquare
0                        iusepot
    3.172300             usepot1
    3.172200             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
0                        ionephase
   80.000000             onephase
!
!  Control flags for orbital elements
!
 0.000000000             pshift, tag ps
0                        ikeep (1 to put primary eclipse at phase 0.0)
0                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (0 for rotation aligned with orbit)
0                        set to 1 to fit for Tconj of binary
0                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                       imag (0 for input data in mag, 1 for input data in flux)
   0                     ielite
0                        ifixgamma (0, 1, or 2)
0                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                        frac switch (>1 to enable ELCratio.* files)
0                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
        0.00000          chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
            0            jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                       Ndynwin (number of segments to integrate)
       0.000000          Tref for dynamical integrator (Nbody >=3)
 0.00000                 hh (step size in days)
0                        iGR (1 for GR, 2 for tidal, 3 for both)
0                        set to 1 for binary+binary model (use Nbody=4)
0                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
0                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
         0.0000          chi^2 penalty for transit or secondary eclipse
0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
 0.0000000000000         Kepler contamination, tag co
0                        Iseason (1 for seasonal Kepler contamination)
 0.0000000000000         contamS0 (season 0 contamination, tag s0)
```

```
 0.0000000000000         contamS1 (season 1 contamination, tag s1)
 0.0000000000000         contamS2 (season 2 contamination, tag s2)
 0.0000000000000         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
0                        set to 1 for fast Kepler binning
   0.00000               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 0                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 0                       Nbody
!
!  Binary orbit parameters
!
     0.634828000000000   Period (days), tag pe
     0.000000000000000   T0 (time of periastron passage), tag T0
     0.000000000000000   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
 -99.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0000000000000          eccentricity, tag ec
    0.0000000000000      argument of periaston in degrees, tag ar
 0.0000000000000000      e*cos(omega), tag oc
 0.0000000000000000      e*sin(omega), tag os
 0.0000000000000000      sqrt(e)*cos(omega), tag bc
 0.0000000000000000      sqrt(e)*sin(omega), tag bs
   76.1709500000000048   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
    0.0000000000000      primK (K-velocity of star 1 in km/sec), tag pk
    3.000000             separ (semimajor axis in solar radii), tag se
  -30.960000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  30   14                Nalph1, Nbet1
 7400.000000000          Teff1 (K), tag T1
 0.2500000               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   0.9260000000000       omega1, tag o1
0.9249450000000          fill1, tag f1
     0.0000000000000     primrad (star 1 radius in solar radii), tag pr
0.000000000000000        frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.0000000000000000    primmass (star 1 mass in solar masses), tag pm
   0.0000000             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 0.82200000 -0.39800000  limb darkening coefficients, filter 1, tag x1, y1
 0.78300000 -0.18000000  limb darkening coefficients, filter 2, tag x2, y2
 0.81400000  0.03300000  limb darkening coefficients, filter 3, tag x3, y3
 0.74600000  0.14200000  limb darkening coefficients, filter 4, tag x4, y4
 0.65500000  0.19400000  limb darkening coefficients, filter 5, tag x5, y5
 0.58100000  0.24700000  limb darkening coefficients, filter 6, tag x6, y6
 0.46000000  0.29500000  limb darkening coefficients, filter 7, tag x7, y7
 0.46000000  0.29500000  limb darkening coefficients, filter 8, tag x8, y8
 0.63700  0.17400        bolometric L.D. coefficients (used when Nref >= 0)
 0.0000000               Temperature factor spot 1, star 1, tag b1
```

```
  0.0000000              Latitude of spot 1, star 1 (degrees), tag b2
  0.0000000              Longitude of spot 1, star 1 (degrees), tag b3
  0.0000000              Angular radius of spot 1, star 1 (degrees), tag b4
 0.0000000               Temperature factor spot 2, star 1, tag b5
  0.0000000              Latitude of spot 2, star 1 (degrees), tag b6
  0.0000000              Longitude of spot 2, star 1 (degrees), tag b7
  0.0000000              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.00000000               rk1 (apsidal constant, star 1), tag a1
      0.00000000000      flux star 1, filter 1 (need Nterms > 0), tag 11
      0.00000000000      flux star 1, filter 2 (need Nterms > 0), tag 12
      0.00000000000      flux star 1, filter 3 (need Nterms > 0), tag 13
      0.00000000000      flux star 1, filter 4 (need Nterms > 0), tag 14
      0.00000000000      flux star 1, filter 5 (need Nterms > 0), tag 15
      0.00000000000      flux star 1, filter 6 (need Nterms > 0), tag 16
      0.00000000000      flux star 1, filter 7 (need Nterms > 0), tag 17
      0.00000000000      flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  30   14                Nalph2, Nbet2
 4567.360000000          Teff2 (K), tag T2
  0.0000000000000000     temprat (T_2/T_1), tag te
 0.0800000               Tgrav2, tag g2
 0.5000000               alb2 (albedo of star 2), tag l2
   1.0040000000000       omega2, tag o2
0.9900000000000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.0000000000000000       frac2 (fractional radius star 2:  R_2/a), tag q2
     0.0000000000000     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.0000000000000     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     0.78037999999999996 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.000000000000000     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 0.84600000  0.15500000  limb darkening coefficients, filter 1, tag z1, w1
 0.81400000  0.12800000  limb darkening coefficients, filter 2, tag z2, w2
 0.72900000  0.23900000  limb darkening coefficients, filter 3, tag z3, w3
 0.63800000  0.27300000  limb darkening coefficients, filter 4, tag z4, w4
 0.54700000  0.27000000  limb darkening coefficients, filter 5, tag z5, w5
 0.45700000  0.25600000  limb darkening coefficients, filter 6, tag z6, w6
 0.32600000  0.22200000  limb darkening coefficients, filter 7, tag z7, w7
 0.32600000  0.22200000  limb darkening coefficients, filter 8, tag z8, w8
 0.62600  0.22000        bolometric L.D. coefficients (used when Nref >= 0)
 0.0000000               Temperature factor spot 1, star 2, tag c1
  0.0000000              Latitude of spot 1, star 2 (degrees), tag c2
  0.0000000              Longitude of spot 1, star 2 (degrees), tag c3
  0.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
 0.0000000               Temperature factor spot 2, star 2, tag c5
  0.0000000              Latitude of spot 2, star 2 (degrees), tag c6
  0.0000000              Longitude of spot 2, star 2 (degrees), tag c7
```

```
  0.0000000             Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000           beam2 (Doppler boosting factor, star 2), tag e2
0.00000000              rk2 (apsidal constant, star 2), tag a2
       0.00000000000    flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000    flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000    flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000    flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000    flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000    flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000    flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000    flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                       idint (1 for accretion disk)
 90                     Ntheta
 60                     Nradius
 2.01000000             beta_rim (opening angle in degrees), tag be
0.010000000             rinner (radius of inner hole), tag ri
0.750000000             router (radius of outer disk), tag ro
  7000.000000           Tdisk (temperature of inner edge in K) tag td
-0.750000000            xi (power-law exponent on temperature profile), tag xi
 0.0000000000           Temperature factor spot 1, disk, tag d1
  0.000000000           Azimuth of spot 1, disk (degrees), tag d2
  0.000000000           Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
  0.000000000           Angular size of spot 1, disk (degrees), tag d4
 0.0000000000           Temperature factor spot 2, disk, tag d5
  0.000000000           Azimuth of spot 2, disk (degrees), tag d6
  0.000000000           Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
  0.000000000           Angular size of spot 2, disk (degrees), tag d8
    0.000000000         reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000       asini (projected semimajor axis in seconds)
 0.0000000000           asini error
```

16. Appendix H: ELC.inp for SV Cam

SV Cam is a close binary where the primary star has large star spots.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
0                        itime
     0.00000000          t_start (if itime=2)
     0.00000000          t_end   (if itime=2)
 0.00000000              time step in days (if itime=2)
  2.000000               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
1                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
3                        iRVfilt
  1                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  3600.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6700.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
 12000.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
 3                       Nref
   0.00100               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
3                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
   4822.5800000000       T3 (Kelvin), tag t3
  4.50464000000          g3 (log(g) in c.g.s), tag g3
   -1.5606000000000000   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
     0                   Nterms for fast analytic
0                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 1                       iecheck
0                        ism1
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
```

```
    0.000000              phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000              phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
1                         isquare
0                         iusepot
    0.000000              usepot1
    0.000000              usepot2
!
!  Drawing flags
!
0                         idraw (1 to write output drawing files)
0                         ionephase
    0.000000              onephase
!
!  Control flags for orbital elements
!
-0.000052860              pshift, tag ps
1                         ikeep (1 to put primary eclipse at phase 0.0)
0                         isynch (1 to keep rotation synchronous at periastron)
0                         ialign (0 for rotation aligned with orbit)
0                         set to 1 to fit for Tconj of binary
0                         set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                        imag (0 for input data in mag, 1 for input data in flux)
   0                      ielite
2                         ifixgamma (0, 1, or 2)
0                         iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                         frac switch (>1 to enable ELCratio.* files)
0                         set to 1 to supress optimizer screen output
0                         set to 1 to supress demcmcELC output files
        0.00000           chi^2 threshold to write to output files
0.0                       median fit (0 for chi^2, >0 for median)
            0             jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                        Ndynwin (number of segments to integrate)
       0.000000           Tref for dynamical integrator (Nbody >=3)
 0.00000                  hh (step size in days)
0                         iGR (1 for GR, 2 for tidal, 3 for both)
0                         set to 1 for binary+binary model (use Nbody=4)
0                         itconj (0=T_peri, 1=T_tran, 2=T_occul)
0                         set to 1 for logarithmic mass ratios
0                         set to 1 for planet radii in Earth radii
0                         set to 1 to treat transits and occultations together
0                         set to 1 for transit penalty
0                         set to 1 for secondary eclipse penalty
         0.0000           chi^2 penalty for transit or secondary eclipse
0                         set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
 0.0000000000000          Kepler contamination, tag co
0                         Iseason (1 for seasonal Kepler contamination)
 0.0000000000000          contamS0 (season 0 contamination, tag s0)
```

```
 0.0000000000000         contamS1 (season 1 contamination, tag s1)
 0.0000000000000         contamS2 (season 2 contamination, tag s2)
 0.0000000000000         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
0                        set to 1 for fast Kepler binning
   0.00000               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 0                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 0                       Nbody
!
!  Binary orbit parameters
!
     0.593073000000000   Period (days), tag pe
     0.000000000000000   T0 (time of periastron passage), tag T0
     0.000000000000000   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
 -99.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0000000000000          eccentricity, tag ec
  130.0000000000000      argument of periaston in degrees, tag ar
 0.0000000000000000      e*cos(omega), tag oc
 0.0000000000000000      e*sin(omega), tag os
 0.0000000000000000      sqrt(e)*cos(omega), tag bc
 0.0000000000000000      sqrt(e)*sin(omega), tag bs
   74.6372999999999962   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
    0.0000000000000      primK (K-velocity of star 1 in km/sec), tag pk
    3.832270             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  90   45                Nalph1, Nbet1
 6157.630000000          Teff1 (K), tag T1
 0.0842100               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
0.6079410000000          fill1, tag f1
     0.0000000000000     primrad (star 1 radius in solar radii), tag pr
0.000000000000000        frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.0000000000000000    primmass (star 1 mass in solar masses), tag pm
   0.0000000             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 0.16800000 -1.57700000  limb darkening coefficients, filter 1, tag x1, y1
 0.72000000 -1.04200000  limb darkening coefficients, filter 2, tag x2, y2
 0.60000000 -0.24600000  limb darkening coefficients, filter 3, tag x3, y3
 0.74600000  0.14200000  limb darkening coefficients, filter 4, tag x4, y4
 0.65500000  0.19400000  limb darkening coefficients, filter 5, tag x5, y5
 0.58100000  0.24700000  limb darkening coefficients, filter 6, tag x6, y6
 0.46000000  0.29500000  limb darkening coefficients, filter 7, tag x7, y7
 0.46000000  0.29500000  limb darkening coefficients, filter 8, tag x8, y8
 0.46800  0.24200        bolometric L.D. coefficients (used when Nref >= 0)
 0.8699500               Temperature factor spot 1, star 1, tag b1
```

```
149.4828000              Latitude of spot 1, star 1 (degrees), tag b2
 24.3666000              Longitude of spot 1, star 1 (degrees), tag b3
 20.1033300              Angular radius of spot 1, star 1 (degrees), tag b4
 1.4061500               Temperature factor spot 2, star 1, tag b5
172.6668000              Latitude of spot 2, star 1 (degrees), tag b6
 44.2404000              Longitude of spot 2, star 1 (degrees), tag b7
 43.8299700              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.00000000               rk1 (apsidal constant, star 1), tag a1
       0.00000000000     flux star 1, filter 1 (need Nterms > 0), tag 11
       0.00000000000     flux star 1, filter 2 (need Nterms > 0), tag 12
       0.00000000000     flux star 1, filter 3 (need Nterms > 0), tag 13
       0.00000000000     flux star 1, filter 4 (need Nterms > 0), tag 14
       0.00000000000     flux star 1, filter 5 (need Nterms > 0), tag 15
       0.00000000000     flux star 1, filter 6 (need Nterms > 0), tag 16
       0.00000000000     flux star 1, filter 7 (need Nterms > 0), tag 17
       0.00000000000     flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  40   10                Nalph2, Nbet2
 4550.450000000          Teff2 (K), tag T2
  0.0000000000000000     temprat (T_2/T_1), tag te
 0.1007100               Tgrav2, tag g2
 0.1385000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.7777440000000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.0000000000000000       frac2 (fractional radius star 2:  R_2/a), tag q2
     0.0000000000000     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.0000000000000     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     0.63937999999999995 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.000000000000000     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 0.84600000  0.15500000  limb darkening coefficients, filter 1, tag z1, w1
 0.81400000  0.12800000  limb darkening coefficients, filter 2, tag z2, w2
 0.72900000  0.23900000  limb darkening coefficients, filter 3, tag z3, w3
 0.63800000  0.27300000  limb darkening coefficients, filter 4, tag z4, w4
 0.54700000  0.27000000  limb darkening coefficients, filter 5, tag z5, w5
 0.45700000  0.25600000  limb darkening coefficients, filter 6, tag z6, w6
 0.32600000  0.22200000  limb darkening coefficients, filter 7, tag z7, w7
 0.32600000  0.22200000  limb darkening coefficients, filter 8, tag z8, w8
 0.52200  0.27100        bolometric L.D. coefficients (used when Nref >= 0)
-1.0500000               Temperature factor spot 1, star 2, tag c1
120.0000000              Latitude of spot 1, star 2 (degrees), tag c2
110.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 30.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
-1.2000000               Temperature factor spot 2, star 2, tag c5
 90.0000000              Latitude of spot 2, star 2 (degrees), tag c6
190.0000000              Longitude of spot 2, star 2 (degrees), tag c7
```

```
  8.0000000              Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000            beam2 (Doppler boosting factor, star 2), tag e2
0.00000000               rk2 (apsidal constant, star 2), tag a2
       0.00000000000     flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000     flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000     flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000     flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000     flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000     flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000     flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000     flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                        idint (1 for accretion disk)
120                      Ntheta
 60                      Nradius
 2.00000000              beta_rim (opening angle in degrees), tag be
0.010000000              rinner (radius of inner hole), tag ri
0.500000000              router (radius of outer disk), tag ro
  4700.000000            Tdisk (temperature of inner edge in K) tag td
-0.750000000             xi (power-law exponent on temperature profile), tag xi
-1.5000000000            Temperature factor spot 1, disk, tag d1
280.000000000            Azimuth of spot 1, disk (degrees), tag d2
  0.900000000            Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
  4.000000000            Angular size of spot 1, disk (degrees), tag d4
-1.2000000000            Temperature factor spot 2, disk, tag d5
270.000000000            Azimuth of spot 2, disk (degrees), tag d6
  1.000000000            Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 10.000000000            Angular size of spot 2, disk (degrees), tag d8
    0.000000000          reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000        asini (projected semimajor axis in seconds)
 0.0000000000            asini error
```

17. Appendix I: ELC.inp for 4U 1543-47

4U 1543-47 is an X-ray binary with a black hole primary and an A-star secondary. In this case there is no accretion disk.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
0                        itime
   -39.88000000          t_start (if itime=2)
  1450.00000000          t_end   (if itime=2)
 0.00000000              time step in days (if itime=2)
  3.000000               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
1                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
3                        iRVfilt
  1                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  3600.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6700.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
 12000.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
-1                       Nref
   0.00100               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
0                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
0                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
  -4822.5800000000       T3 (Kelvin), tag t3
 -4.50464000000          g3 (log(g) in c.g.s), tag g3
   -1.5606000000000000   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
     0                   Nterms for fast analytic
1                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 1                       iecheck
1                        ism1
```

```
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
1                        isquare
0                        iusepot
    0.000000             usepot1
    0.000000             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
0                        ionephase
    0.000000             onephase
!
!  Control flags for orbital elements
!
-0.003885794             pshift, tag ps
0                        ikeep (1 to put primary eclipse at phase 0.0)
0                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (0 for rotation aligned with orbit)
0                        set to 1 to fit for Tconj of binary
0                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                       imag (0 for input data in mag, 1 for input data in flux)
   1                     ielite
1                        ifixgamma (0, 1, or 2)
0                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
0                        frac switch (>1 to enable ELCratio.* files)
0                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
        0.00000          chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
           0             jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                       Ndynwin (number of segments to integrate)
     -30.800000          Tref for dynamical integrator (Nbody >=3)
 0.00000                 hh (step size in days)
0                        iGR (1 for GR, 2 for tidal, 3 for both)
0                        set to 1 for binary+binary model (use Nbody=4)
0                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
0                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
         0.0000          chi^2 penalty for transit or secondary eclipse
0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
 0.0084135471561         Kepler contamination, tag co
```

```
1                        Iseason (1 for seasonal Kepler contamination)
 0.0225912629319         contamS0 (season 0 contamination, tag s0)
 0.0152760155080         contamS1 (season 1 contamination, tag s1)
 0.0139880417402         contamS2 (season 2 contamination, tag s2)
 0.0167317088482         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
0                        set to 1 for fast Kepler binning
  29.46600               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 0                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 0                       Nbody
!
!  Binary orbit parameters
!
     1.116407200000000   Period (days), tag pe
     0.000000000000000   T0 (time of periastron passage), tag T0
   -20.276983528256199   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
 -99.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0000000000000          eccentricity, tag ec
  150.0000000000000      argument of periaston in degrees, tag ar
 0.16583185021865429     e*cos(omega), tag oc
 0.49371186776473858     e*sin(omega), tag os
 0.00000000000000000     sqrt(e)*cos(omega), tag bc
 0.00000000000000000     sqrt(e)*sin(omega), tag bs
   20.0683502497356763   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
  132.6411179973248      primK (K-velocity of star 1 in km/sec), tag pk
   10.623566             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  44   20                Nalph1, Nbet1
 9400.000000000          Teff1 (K), tag T1
 0.2500000               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
1.0000000000000          fill1, tag f1
     0.0000000000000     primrad (star 1 radius in solar radii), tag pr
0.000000000000000        frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   2.55117437560940452   primmass (star 1 mass in solar masses), tag pm
  89.8628747             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 0.16800000 -1.57700000  limb darkening coefficients, filter 1, tag x1, y1
 0.72000000 -1.04200000  limb darkening coefficients, filter 2, tag x2, y2
 0.60000000 -0.24600000  limb darkening coefficients, filter 3, tag x3, y3
 0.74600000  0.14200000  limb darkening coefficients, filter 4, tag x4, y4
 0.65500000  0.19400000  limb darkening coefficients, filter 5, tag x5, y5
 0.58100000  0.24700000  limb darkening coefficients, filter 6, tag x6, y6
 0.46000000  0.29500000  limb darkening coefficients, filter 7, tag x7, y7
 0.46000000  0.29500000  limb darkening coefficients, filter 8, tag x8, y8
```

```
 0.46800  0.24200        bolometric L.D. coefficients (used when Nref >= 0)
-0.8699500               Temperature factor spot 1, star 1, tag b1
149.4828000              Latitude of spot 1, star 1 (degrees), tag b2
 24.3666000              Longitude of spot 1, star 1 (degrees), tag b3
 20.1033300              Angular radius of spot 1, star 1 (degrees), tag b4
-1.4061500               Temperature factor spot 2, star 1, tag b5
172.6668000              Latitude of spot 2, star 1 (degrees), tag b6
 44.2404000              Longitude of spot 2, star 1 (degrees), tag b7
 43.8299700              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.00000000               rk1 (apsidal constant, star 1), tag a1
       0.00000000000     flux star 1, filter 1 (need Nterms > 0), tag 11
       0.00000000000     flux star 1, filter 2 (need Nterms > 0), tag 12
       0.00000000000     flux star 1, filter 3 (need Nterms > 0), tag 13
       0.00000000000     flux star 1, filter 4 (need Nterms > 0), tag 14
       0.00000000000     flux star 1, filter 5 (need Nterms > 0), tag 15
       0.00000000000     flux star 1, filter 6 (need Nterms > 0), tag 16
       0.00000000000     flux star 1, filter 7 (need Nterms > 0), tag 17
       0.00000000000     flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  44   20                Nalph2, Nbet2
-4000.000000000          Teff2 (K), tag T2
  0.0000000000000000     temprat (T_2/T_1), tag te
 0.2500000               Tgrav2, tag g2
 0.1385000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.0705690000000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.0000000000000000       frac2 (fractional radius star 2:  R_2/a), tag q2
     0.0000000000000     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.0000000000000     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
     4.06566480102542194 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.000000000000000     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 0.84600000  0.15500000  limb darkening coefficients, filter 1, tag z1, w1
 0.81400000  0.12800000  limb darkening coefficients, filter 2, tag z2, w2
 0.72900000  0.23900000  limb darkening coefficients, filter 3, tag z3, w3
 0.63800000  0.27300000  limb darkening coefficients, filter 4, tag z4, w4
 0.54700000  0.27000000  limb darkening coefficients, filter 5, tag z5, w5
 0.45700000  0.25600000  limb darkening coefficients, filter 6, tag z6, w6
 0.32600000  0.22200000  limb darkening coefficients, filter 7, tag z7, w7
 0.32600000  0.22200000  limb darkening coefficients, filter 8, tag z8, w8
 0.52200  0.27100        bolometric L.D. coefficients (used when Nref >= 0)
-1.0500000               Temperature factor spot 1, star 2, tag c1
120.0000000              Latitude of spot 1, star 2 (degrees), tag c2
110.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 30.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
-1.2000000               Temperature factor spot 2, star 2, tag c5
```

```
 90.0000000             Latitude of spot 2, star 2 (degrees), tag c6
190.0000000             Longitude of spot 2, star 2 (degrees), tag c7
  8.0000000             Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000           beam2 (Doppler boosting factor, star 2), tag e2
0.00000000              rk2 (apsidal constant, star 2), tag a2
       0.00000000000    flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000    flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000    flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000    flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000    flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000    flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000    flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000    flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
0                       idint (1 for accretion disk)
120                     Ntheta
 60                     Nradius
 2.00000000             beta_rim (opening angle in degrees), tag be
0.010000000             rinner (radius of inner hole), tag ri
0.500000000             router (radius of outer disk), tag ro
  4700.000000           Tdisk (temperature of inner edge in K) tag td
-0.750000000            xi (power-law exponent on temperature profile), tag xi
-1.5000000000           Temperature factor spot 1, disk, tag d1
280.000000000           Azimuth of spot 1, disk (degrees), tag d2
  0.900000000           Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
  4.000000000           Angular size of spot 1, disk (degrees), tag d4
-1.2000000000           Temperature factor spot 2, disk, tag d5
270.000000000           Azimuth of spot 2, disk (degrees), tag d6
  1.000000000           Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 10.000000000           Angular size of spot 2, disk (degrees), tag d8
    0.000000000         reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000       asini (projected semimajor axis in seconds)
 0.0000000000           asini error
```

18. Appendix J: ELC.inp for GRS 1124-683

GRS 1124-683 (aka X-ray Nova Muscae 1991) is an X-ray binary with a black hole primary and a K-star secondary. In this case there is an accretion disk that has spots.

```
#1 New version 6 input, this line required
!
!  Lines starting with # or ! can be inserted anywhere
!
!  Units
!
0                        iunit
!
!  Parameters for light surve sampling
!
0                        itime
     0.00000000          t_start (if itime=2)
     0.00000000          t_end   (if itime=2)
 0.00000000              time step in days (if itime=2)
  2.000000               dphase in degrees (if itime=0 or 1)
!
!  Control flags for filters and limb darkening
!
1                        iatm (0 for black body)
1 1 1 1 1 1 1 1          icnU,icnB,icnV,icnR,icnI,icnJ,icnH,icnK
1                        iRVfilt
  1                      ilaw (1=lin., 2=log, 3=sqrt, 4=quad, 5=Kipping quad)
  3600.0000              central wavelength for filter 1 when iatm=0
  4500.0000              central wavelength for filter 2 when iatm=0
  5550.0000              central wavelength for filter 3 when iatm=0
  6700.0000              central wavelength for filter 4 when iatm=0
  8700.0000              central wavelength for filter 5 when iatm=0
 12000.0000              central wavelength for filter 6 when iatm=0
 16200.0000              central wavelength for filter 7 when iatm=0
 22000.0000              central wavelength for filter 8 when iatm=0
-1                       Nref
   0.00100               log10(Lx) in erg/sec, tag Lx
0                        X-ray foreshortening switch (1 for point source)
0                        idark1
0                        idark2
1                        ispotprof (0=constant, 1=linear, 2=Gaussian change in temperature profile
0                        igrav
0                        set to 1 to use fluxes
!
!  Third light parameters
!
  -4400.0000000000       T3 (Kelvin), tag t3
 -3.37000000000          g3 (log(g) in c.g.s), tag g3
   -1.0000000000000000   SA3 (ratio of body 1 to body 3 areas), tag SA
!
!  Control flags related to speed and accuracy
!
     0                   Nterms for fast analytic
0                        mandel (0 for Gimenez, 1 for Mandel & Agol)
   0                     Ngap
 0                       iecheck
0                        ism1
```

```
0                        ifasttrans (>0 for fast transit mode)
     0                   MonteCarlo (0 for interpolation, >10 for Monte Carlo integration)
    0.000000             phaselow  (phase range if phaselow>0 and phasehigh>0
    0.000000             phasehigh  and phaselow < phasehigh)
!
!  Control flags for Roche geometry
!
1                        isquare
0                        iusepot
    3.000000             usepot1
   10.000000             usepot2
!
!  Drawing flags
!
0                        idraw (1 to write output drawing files)
0                        ionephase
   -0.554000             onephase
!
!  Control flags for orbital elements
!
-0.008422142             pshift, tag ps
0                        ikeep (1 to put primary eclipse at phase 0.0)
1                        isynch (1 to keep rotation synchronous at periastron)
0                        ialign (0 for rotation aligned with orbit)
0                        set to 1 to fit for Tconj of binary
0                        set to 1 to fit for e*cos(omega), e*sin(omega) of binary
!
!  Control flags for use with optimizers
!
 0                       imag (0 for input data in mag, 1 for input data in flux)
 141                     ielite
0                        ifixgamma (0, 1, or 2)
0                        iwriteeclipse (1 to fit for eclipse times when Nbody >=3 and itime=2)
1                        frac switch (>1 to enable ELCratio.* files)
0                        set to 1 to supress optimizer screen output
0                        set to 1 to supress demcmcELC output files
       0.00000           chi^2 threshold to write to output files
0.0                      median fit (0 for chi^2, >0 for median)
           0             jdum (seed for markovELC, geneticELC, randomELC, demcmcELC, hammerELC)
!
!  Control flags and parameters for use with the dynamical integrator (Nbody >=3)
!
 0                       Ndynwin (number of segments to integrate)
      0.000000           Tref for dynamical integrator (Nbody >=3)
 0.00000                 hh (step size in days)
0                        iGR (1 for GR, 2 for tidal, 3 for both)
0                        set to 1 for binary+binary model (use Nbody=4)
0                        itconj (0=T_peri, 1=T_tran, 2=T_occul)
0                        set to 1 for logarithmic mass ratios
0                        set to 1 for planet radii in Earth radii
0                        set to 1 to treat transits and occultations together
0                        set to 1 for transit penalty
0                        set to 1 for secondary eclipse penalty
        0.0000           chi^2 penalty for transit or secondary eclipse
0                        set to 1 for informational output (ELC only)
!
! Features for Kepler data
!
 0.0000000000000         Kepler contamination, tag co
```

```
0                        Iseason (1 for seasonal Kepler contamination)
 0.0000000000000         contamS0 (season 0 contamination, tag s0)
 0.0000000000000         contamS1 (season 1 contamination, tag s1)
 0.0000000000000         contamS2 (season 2 contamination, tag s2)
 0.0000000000000         contamS3 (season 3 contamination, tag s3)
   0                     Nseg (number of segments to fit forcontamination)
0                        set to 1 for fast Kepler binning
   0.00000               bin size for light curves (minutes)
   0.00000               bin size for RV curves (minutes)
 0                       NSC (number of short cadence intervals)
!
!  Number of bodies (2 to 10)
!
 0                       Nbody
!
!  Binary orbit parameters
!
     0.432606000000000   Period (days), tag pe
     0.000000000000000   T0 (time of periastron passage), tag T0
     0.000000000000000   Tconj (time of primary eclipse), tag Tc
     0.0000000000000     PbpTc (period + Tconj for binary), tag bp
     0.0000000000000     PbmTc (period - Tconj for binary), tag bm
 -99.000000000000        binqTc (slope in P-Tc plane for binary), tag bq
       0.000000000000    Tbinoff (offset in P-Tc plane for binary), tag bt
0.0000000000000          eccentricity, tag ec
  210.0000000000000      argument of periaston in degrees, tag ar
 0.0000000000000000      e*cos(omega), tag oc
 0.0000000000000000      e*sin(omega), tag os
 0.0000000000000000      sqrt(e)*cos(omega), tag bc
 0.0000000000000000      sqrt(e)*sin(omega), tag bs
   41.8317816671879470   finc (inclination in degrees), tag in
   0.0000000000          Omega_bin (nodal angle of binary in degrees), tag Ob
  406.6862076138307      primK (K-velocity of star 1 in km/sec), tag pk
    5.640169             separ (semimajor axis in solar radii), tag se
    0.000000             gamma
 0.0000000               ecosw (phase difference between eclipses) tag dp
!
!  Body 1 parameters
!
  40   20                Nalph1, Nbet1
 4400.000000000          Teff1 (K), tag T1
 0.1023200               Tgrav1, tag g1
 1.0000000               alb1 (albedo of star 1), tag l1
   1.0000000000000       omega1, tag o1
1.0000000000000          fill1, tag f1
     0.0000000000000     primrad (star 1 radius in solar radii), tag pr
0.000000000000000        frac1 (fractional radius star 1:  R_1/a), tag q1
0.0000000000000          radfill1 (set to use fill1 in terms of R_eff)
   0.97790673647000070   primmass (star 1 mass in solar masses), tag pm
  47.2821536             axis_I1 (inclination of rotation axis if ialign=0), tag ai
   0.0000000             axis_beta1 (angle of rotation axis wrt orbit if ialign=0), tag ab
 2.16800000 -1.57700000  limb darkening coefficients, filter 1, tag x1, y1
 1.72000000 -1.04200000  limb darkening coefficients, filter 2, tag x2, y2
 0.00000000 -0.24600000  limb darkening coefficients, filter 3, tag x3, y3
 0.74600000  0.14200000  limb darkening coefficients, filter 4, tag x4, y4
 0.65500000  0.19400000  limb darkening coefficients, filter 5, tag x5, y5
 0.58100000  0.24700000  limb darkening coefficients, filter 6, tag x6, y6
 0.46000000  0.29500000  limb darkening coefficients, filter 7, tag x7, y7
 0.46000000  0.29500000  limb darkening coefficients, filter 8, tag x8, y8
```

```
 0.28200  0.41300        bolometric L.D. coefficients (used when Nref >= 0)
-1.1000000               Temperature factor spot 1, star 1, tag b1
114.7684155              Latitude of spot 1, star 1 (degrees), tag b2
132.0954681              Longitude of spot 1, star 1 (degrees), tag b3
 24.4494145              Angular radius of spot 1, star 1 (degrees), tag b4
-1.0000000               Temperature factor spot 2, star 1, tag b5
  1.0000000              Latitude of spot 2, star 1 (degrees), tag b6
  1.0000000              Longitude of spot 2, star 1 (degrees), tag b7
  1.0000000              Angular radius of spot 2, star 1 (degrees), tag b8
 0.0000000000            beam1 (Doppler boosting factor, star 1), tag e1
0.00000000               rk1 (apsidal constant, star 1), tag a1
       0.00000000000     flux star 1, filter 1 (need Nterms > 0), tag 11
       0.00000000000     flux star 1, filter 2 (need Nterms > 0), tag 12
       0.00000000000     flux star 1, filter 3 (need Nterms > 0), tag 13
       0.00000000000     flux star 1, filter 4 (need Nterms > 0), tag 14
       0.00000000000     flux star 1, filter 5 (need Nterms > 0), tag 15
       0.00000000000     flux star 1, filter 6 (need Nterms > 0), tag 16
       0.00000000000     flux star 1, filter 7 (need Nterms > 0), tag 17
       0.00000000000     flux star 1, filter 8 (need Nterms > 0), tag 18
!
!  Body 2 parameters
!
  24    6                Nalph2, Nbet2
-4200.000000000          Teff2 (K), tag T2
  0.0000000000000000     temprat (T_2/T_1), tag te
 0.0800000               Tgrav2, tag g2
 0.5000000               alb2 (albedo of star 2), tag l2
   1.0000000000000       omega2, tag o2
0.0100000000000          fill2, tag f2
0.0000000000000          fillsum, tag sf
 0.0000000000000         filldiff, tag sd
  0.0000000000000        secrad (secondary star radius in solar radii), tag sr
0.0000000000000000       frac2 (fractional radius star 2:  R_2/a), tag q2
     0.0000000000000     ratrad (ratio of star 1 radius to star 2 radius,  tag ra
     0.0000000000000     radsum (sum of radii in solar), tag rs
     0.0000000000000     raddiff (R_1 - R_2  in solar), tag rd
0.000000000000000        fracsum ((R_1 + R_2)/a), tag fs
0.000000000000000        fracdiff ((R_1 - R_2)/a), tag fd
0.0000000000000          radfill2 (set to use fill2 in terms of R_eff)
    12.17057675932010241 Q (M_2/M_1), tag ma
   0.000000000000000     secmass (star 2 mass in solar masses), tag sm
   0.000000000000000     masssum (sum of masses in solar),  tag ms
   0.000000000000000     massdiff (M_1 - M_2 in solar), tag md
   0.0000000             axis_I2 (inclination of rotation axis if ialign=0), tag bi
   0.0000000             axis_beta2 (angle of rotation axis wrt orbit if ialign=0), tag bb
 0.84600000  0.15500000  limb darkening coefficients, filter 1, tag z1, w1
 0.81400000  0.12800000  limb darkening coefficients, filter 2, tag z2, w2
 0.40000000  0.23900000  limb darkening coefficients, filter 3, tag z3, w3
 0.63800000  0.27300000  limb darkening coefficients, filter 4, tag z4, w4
 0.54700000  0.27000000  limb darkening coefficients, filter 5, tag z5, w5
 0.45700000  0.25600000  limb darkening coefficients, filter 6, tag z6, w6
 0.32600000  0.22200000  limb darkening coefficients, filter 7, tag z7, w7
 0.32600000  0.22200000  limb darkening coefficients, filter 8, tag z8, w8
 0.29100  0.39800        bolometric L.D. coefficients (used when Nref >= 0)
-1.0000000               Temperature factor spot 1, star 2, tag c1
 -1.0000000              Latitude of spot 1, star 2 (degrees), tag c2
 -1.0000000              Longitude of spot 1, star 2 (degrees), tag c3
 -1.0000000              Angular radius of spot 1, star 2 (degrees), tag c4
-1.0000000               Temperature factor spot 2, star 2, tag c5
```

```
 -1.0000000              Latitude of spot 2, star 2 (degrees), tag c6
 -1.0000000              Longitude of spot 2, star 2 (degrees), tag c7
 -1.0000000              Angular radius of spot 2, star 2 (degrees), tag c8
 0.0000000000            beam2 (Doppler boosting factor, star 2), tag e2
0.00000000               rk2 (apsidal constant, star 2), tag a2
       0.00000000000     flux star 2, filter 1 (need Nterms > 0), tag 21
       0.00000000000     flux star 2, filter 2 (need Nterms > 0), tag 22
       0.00000000000     flux star 2, filter 3 (need Nterms > 0), tag 23
       0.00000000000     flux star 2, filter 4 (need Nterms > 0), tag 24
       0.00000000000     flux star 2, filter 5 (need Nterms > 0), tag 25
       0.00000000000     flux star 2, filter 6 (need Nterms > 0), tag 26
       0.00000000000     flux star 2, filter 7 (need Nterms > 0), tag 27
       0.00000000000     flux star 2, filter 8 (need Nterms > 0), tag 28
!
!  Accretion disk parameters
!
1                        idint (1 for accretion disk)
120                      Ntheta
 90                      Nradius
13.99820087              beta_rim (opening angle in degrees), tag be
0.001000000              rinner (radius of inner hole), tag ri
0.460846369              router (radius of outer disk), tag ro
 10191.175236            Tdisk (temperature of inner edge in K) tag td
-0.674347366             xi (power-law exponent on temperature profile), tag xi
14.9466100072            Temperature factor spot 1, disk, tag d1
297.337004761            Azimuth of spot 1, disk (degrees), tag d2
  0.269940041            Radial cutoff of spot 1, disk (0 <=  r_cut <=1), tag d3
 22.253217604            Angular size of spot 1, disk (degrees), tag d4
-1.0000000000            Temperature factor spot 2, disk, tag d5
 -1.000000000            Azimuth of spot 2, disk (degrees), tag d6
 -1.000000000            Radial cutoff of spot 2, disk (0 <= r_cut <=1), tag d7
 -1.000000000            Angular size of spot 2, disk (degrees), tag d8
    0.054000000          reference phase for disk fraction
!
!  Pulsar parameters
!
    0.00000000000        asini (projected semimajor axis in seconds)
 0.0000000000            asini error
```

REFERENCES

Avni, Y. 1978, in Physics and Astrophysics of Neutron Stars and Black Holes, ed. R. Giacconi & R. Ruffini (Amsterdam: North Holland), p. 42

Avni, Y., & Bahcall, J. N. 1975, ApJ, 197, 675

Bender, C. F., Mahadevan, S., Deshpande, R., Wright, J. T., Roy, A., et al. 2012, ApJ, 751, L31

Bevington, P. R. 1969, Data Reduction and Error Analysis for the Physical Sciences (New York: McGraw-Hill)

Carter, J. A., Fabrycky, D. C., Ragozzine, D., Holman, M. J., Quinn, S. N., et al. 2011, Science, 331, 562

Charbonneau, P. 1995, ApJS, 101, 309

Claret, A. 2001, A&A, 359, 289

Doyle, L. R., Carter, J. A., Fabrycky, D. C., Slawson, R. W., Howell, S. B., et al. 2011, Science, 333, 1602

Espinoza, N., & Jordán 2016, MNRAS, 457, 357

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306

Gelman, A., & Rubin, D. B. 1992, Statistical Science, 7, 457

Giménez, A. 2006a, A&A, 450, 1231

Giménez, A. 2006b, ApJ, 650, 408

Harier, E., Lubich, C., & Wanner, G. 2006, Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations, 2nd Ed. Springer Series in Computational Mathematics, 31 (Springer-Verlag, Berlin)

Hosokawa, Y. 1953, PASJ, 5, 88.

Kipping, D. 2013, MNRAS, 435, 2152

Lanz, T, & Hubeny, I. 2003, ApJS, 146, 417

Lanz, T, & Hubeny, I. 2007, ApJS, 169, 83

Lucy, L. B. 1967, Zs. Ap., 65, 89

Mandel, K., & Agol, E., 2002, ApJ, 580, L171

Mardling, R. A., & Lin, D. N. C. 2002, ApJ, 573, 829

McClintock, J. E., & Remillard, R. A. 1990, ApJ, 350, 386

Murray, C. D., & Dermott, S. F. 1999, Solar System Dynamics. (Cambridge University Press, Cambridge)

Press, W. H., et al. 1992, "Numerical Recipes, Second Edition", Cambridge University Press

Orosz, J. A., & Bailyn, C. D. 1997, ApJ, 477, 876 (Erratum: ApJ, 482, 1086)

Orosz, J. A., & Hauschildt, P. H. 2000, A&A, 364, 265

Rappaport, S., Vanderburg, A., Borkovits, T., Kalomeni, B., Halpern, J. P., et al. 2017, MNRAS, 467, 2160

Skilling, J. 2006, Baysian Analysis, 1, 833

Tegmark, M., Strauss, M. A., Blanton, M. R., et al. 2004, PhRvD, 69, 103501

Ter Braak, C. J. F. 2006, Stat. Comput., 16, 239

Van Hamme, W. 1993, AJ, 106, 2096

van Kerkwijk, M. H., Rappaport, S. A., Breton, R. P., et al. 2010, ApJ, 715

von Zeipel, H. 1924, MNRAS, 84, 665

Wilson, R. E., & Devinney, E. J. 1971, ApJ, 166, 605

Wilson, R. E. 1979, ApJ, 234, 1054

Wilson, R. E. 1990, ApJ, 356, 613

Winn, J. N., Albrecht, S., Johnson, J. A., Torres, G., Cochran, W. D., et al. 2011, ApJ, 741, L1

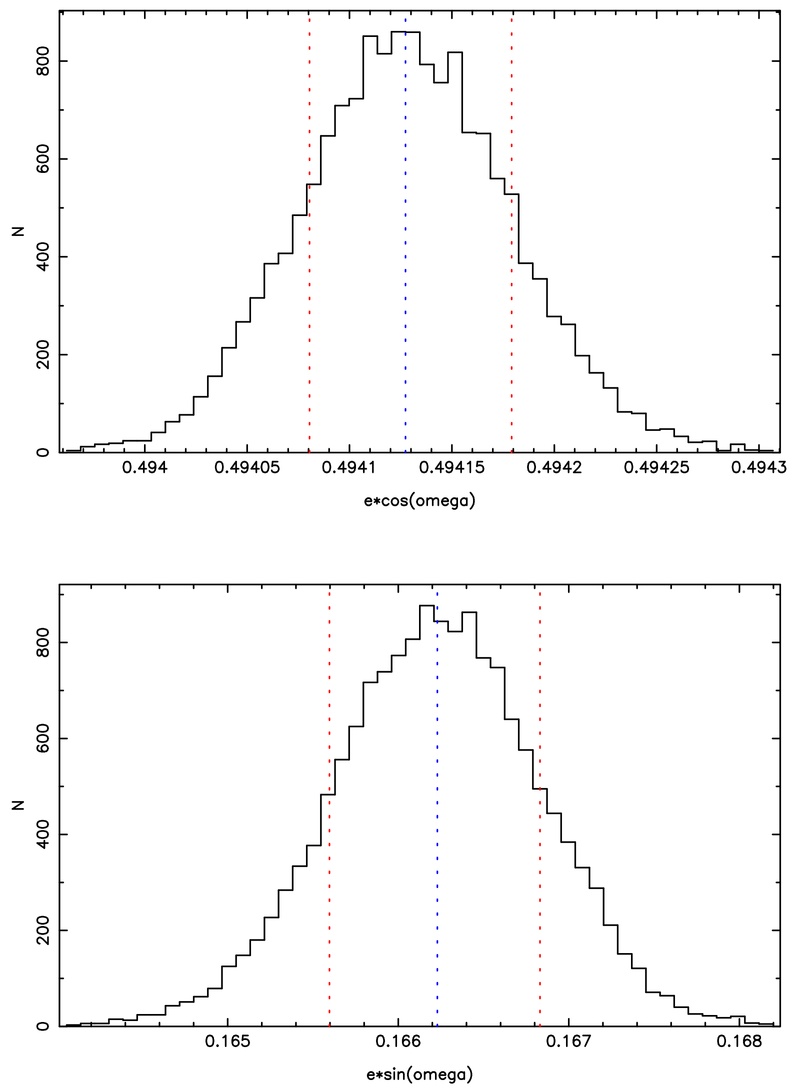

Figure 6: An example page from **`ELCposterior.f90`**. The entire plot may have several pages, and you will see two panels per page, with each plot showing the frequency distribution of a fitted or derived parameter. The blue line is the median value, and the red lines are the 1σ limits determined by integrating under the histogram, and the green lines are lower 15.85% and upper 84.15% limits, respectively.

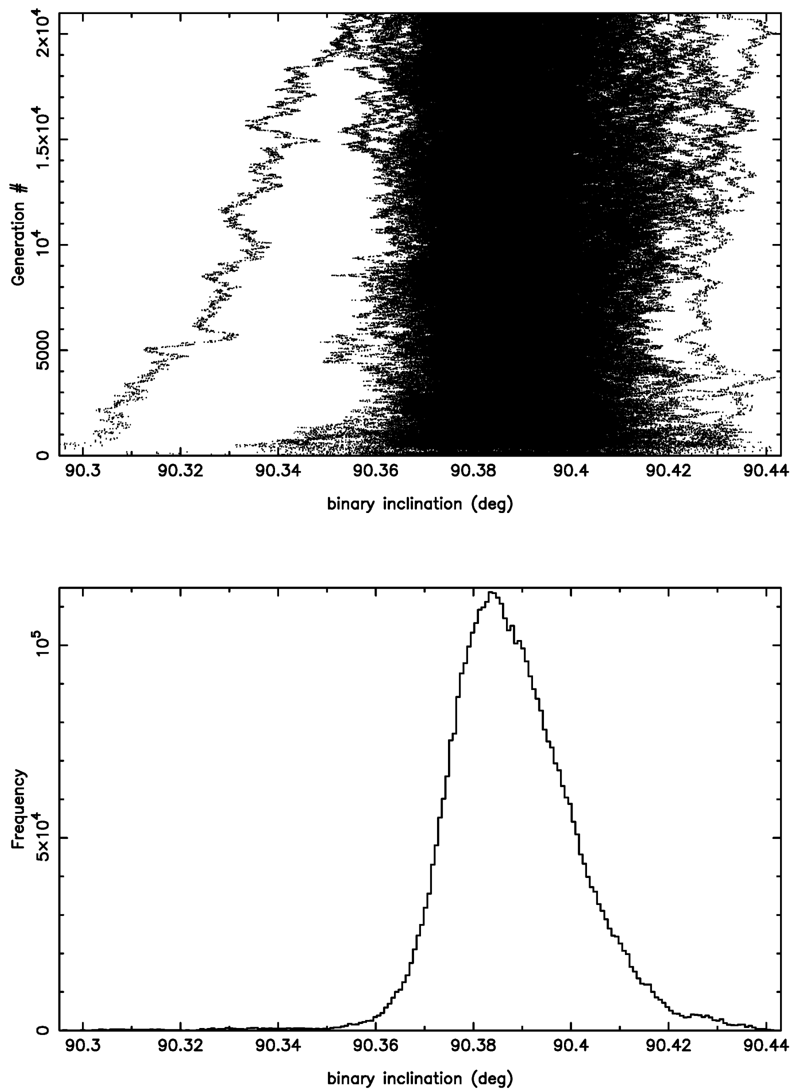

Figure 7: An example page from **ELCprogplotter.f90**. The entire plot may have several pages, and you will see two panels per page. The top panel shows the parameter value of the x-axis and the generation number on the y-axis. The bottom panel shows the frequency distribution of the parameter shown in the top panel.

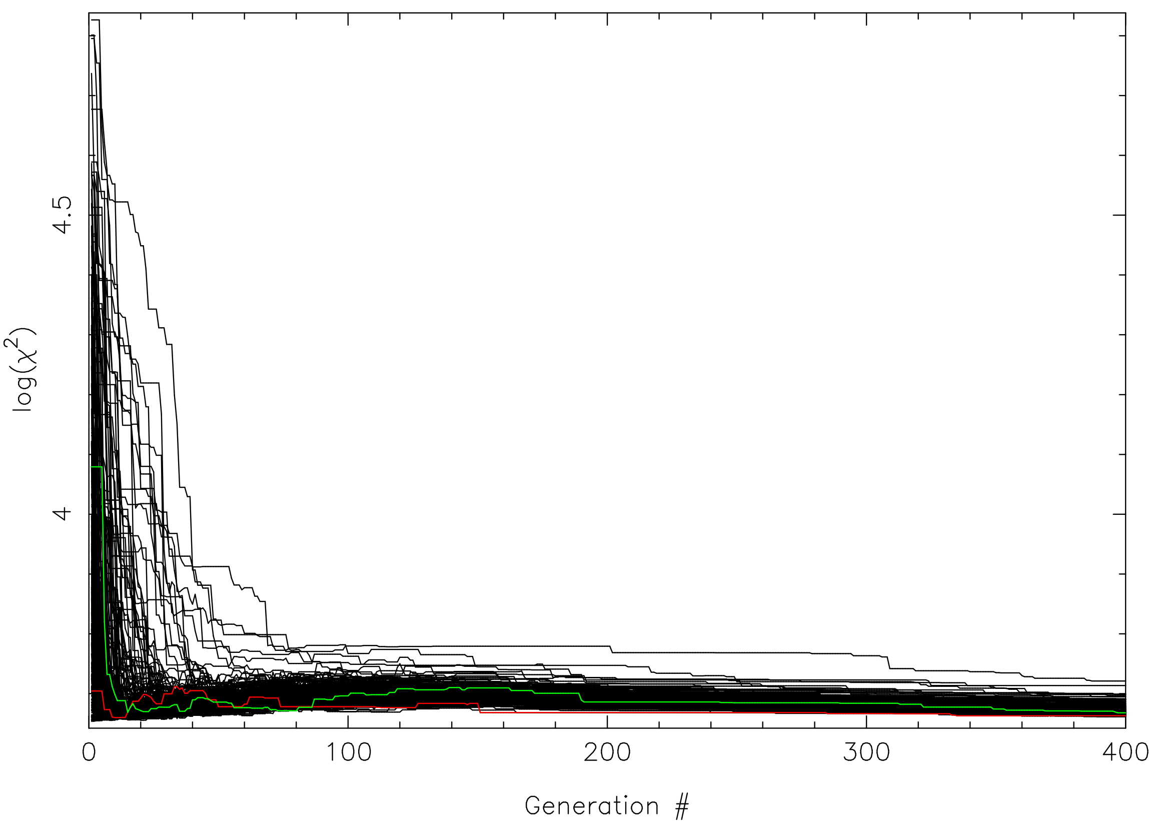

Figure 8: An example plot from **`ELCspreadchiplotter.f90`**. This shows the generation number on the x-axis and $\log \chi^2$ on the y-axis. The chains that are in color are given on the command line.

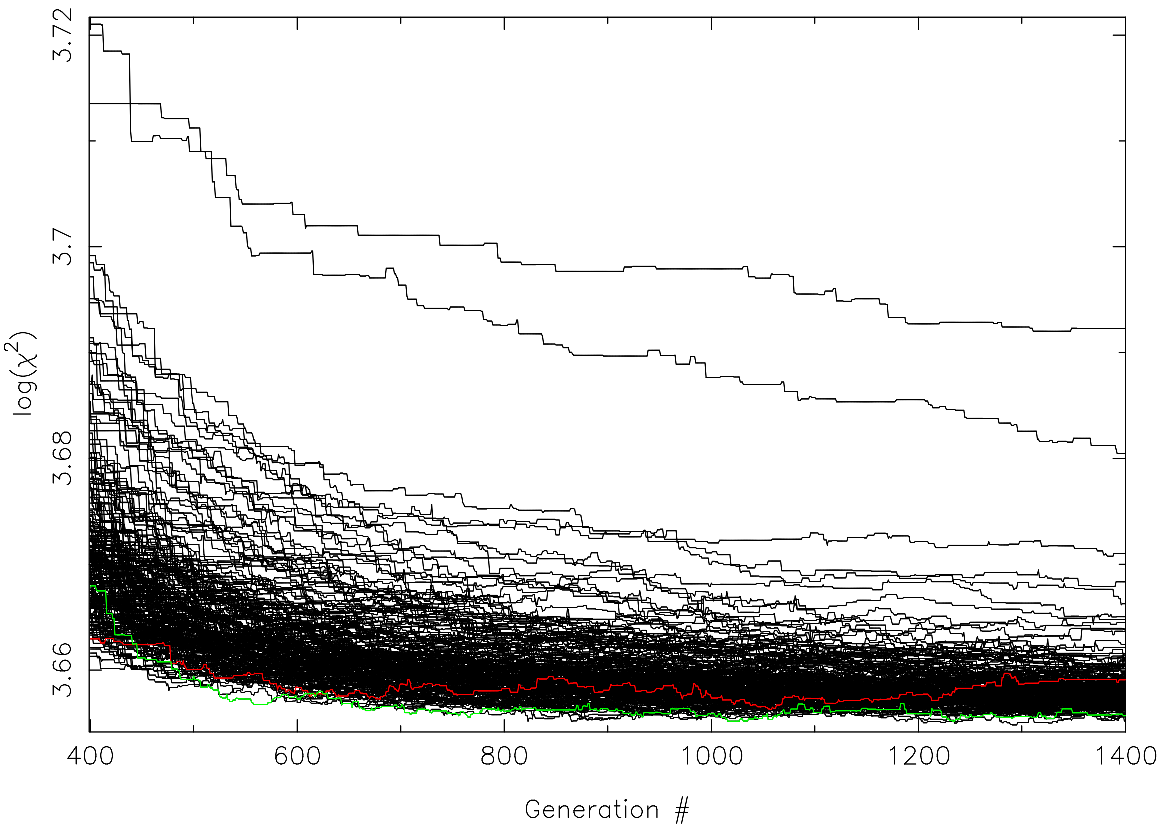

Figure 9: An example plot from **`ELCspreadchiplotter.f90`**. This shows the generation number on the x-axis and $\log \chi^2$ on the y-axis. The chains that are in color are given on the command line.

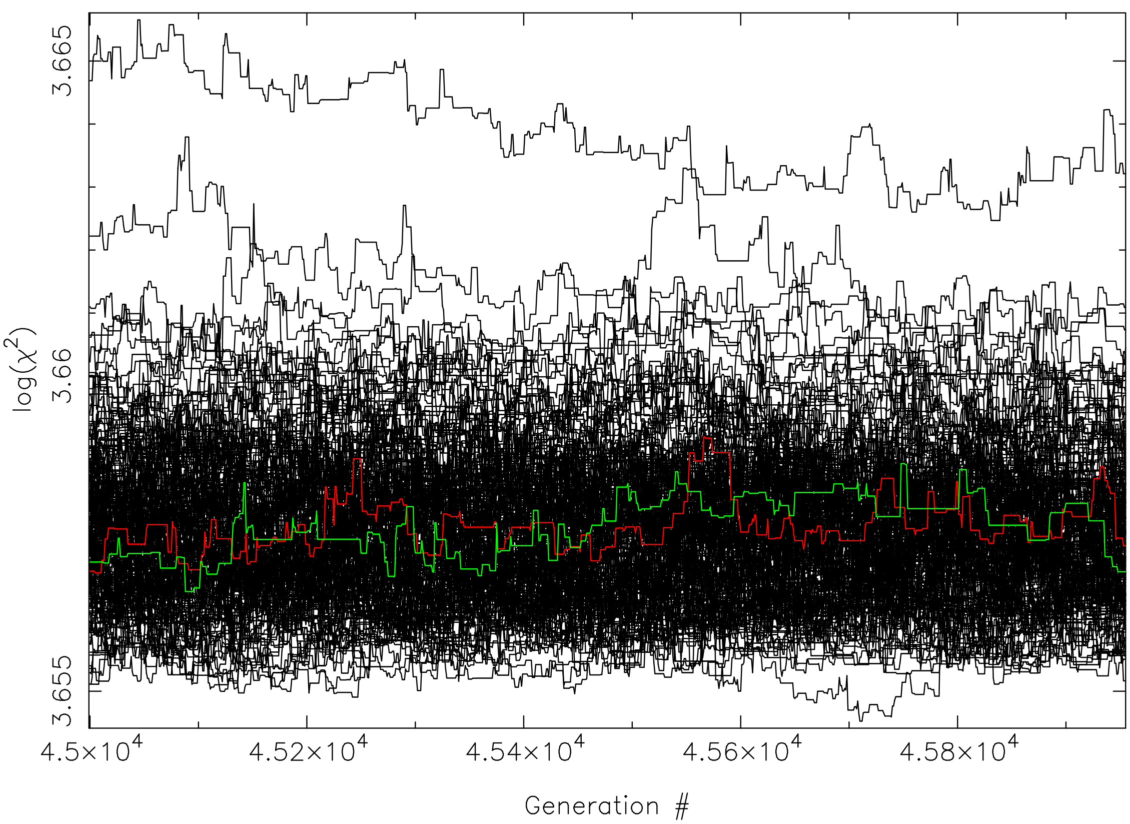

Figure 10: An example plot from **`ELCspreadchiplotter.f90`**. This shows the generation number on the x-axis and $\log \chi^2$ on the y-axis. The chains that are in color are given on the command line.

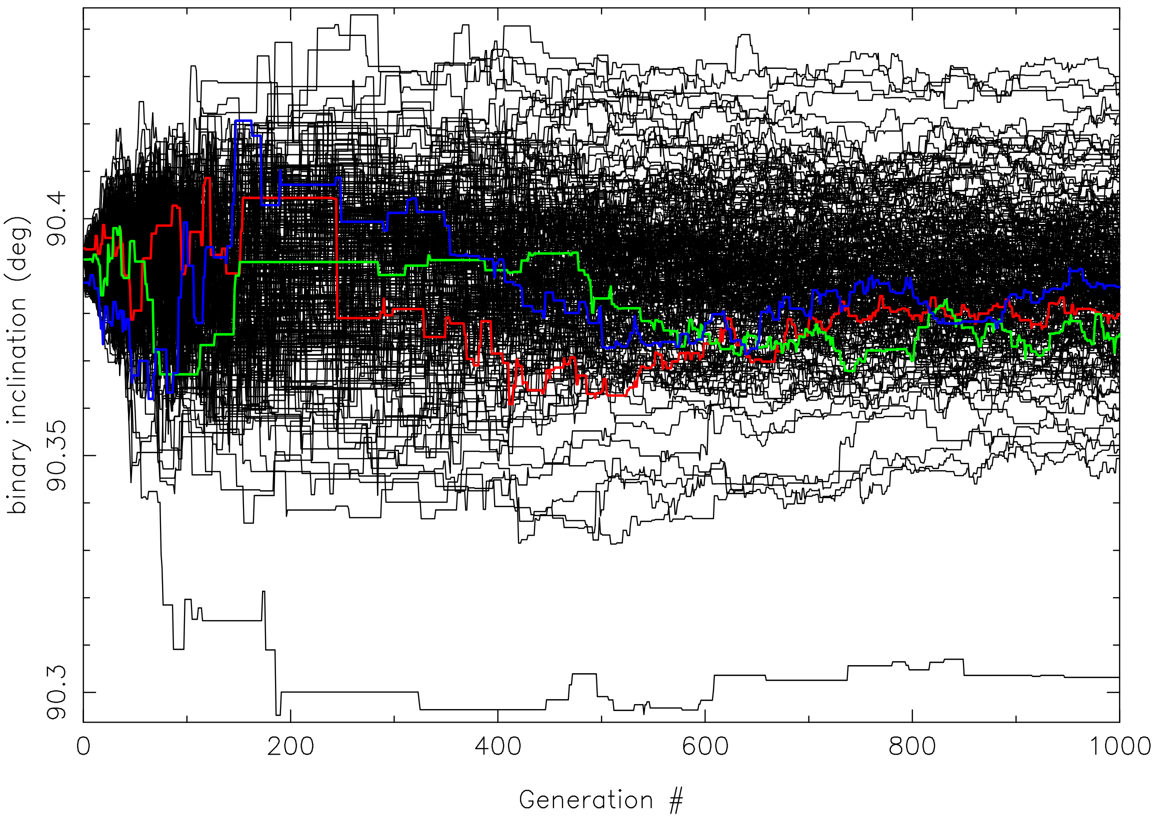

Figure 11: An example plot from **`ELCspreadparmplotter.f90`**. This shows the generation number on the x-axis and the parameter value on the y-axis. The chains that are in color are given on the command line.

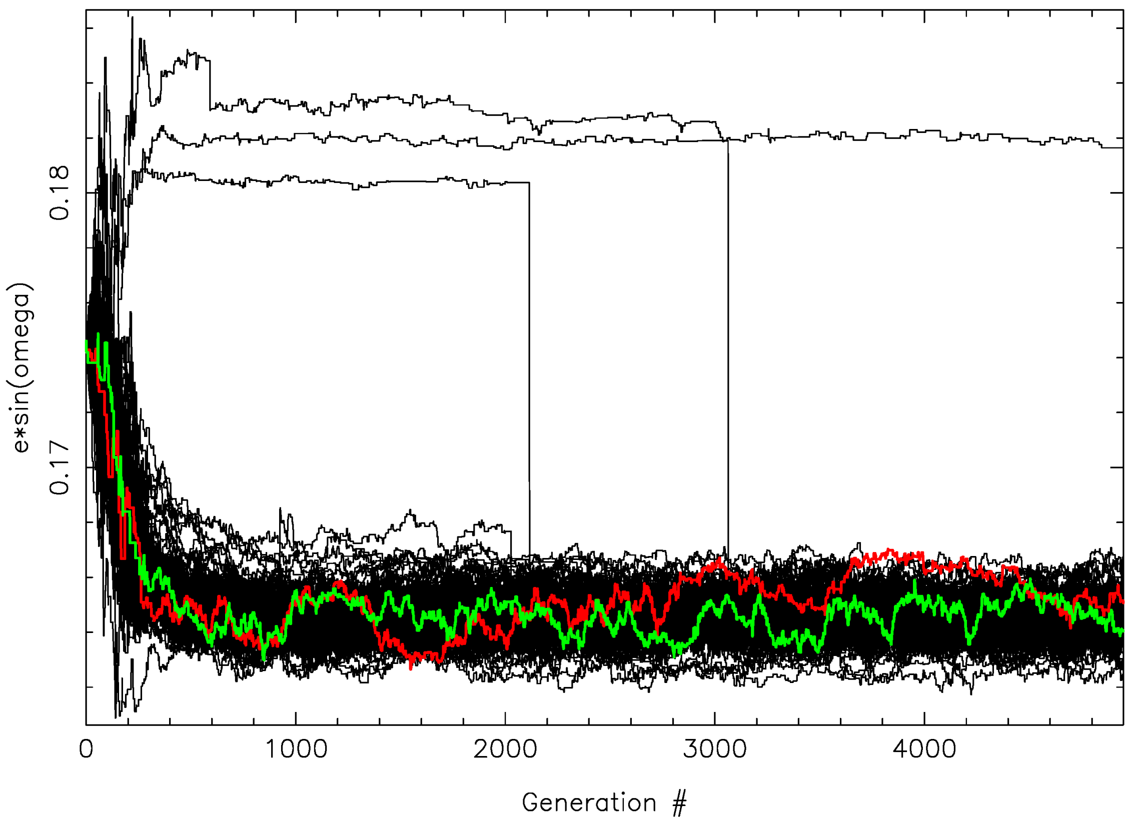

Figure 12: An example plot from **`ELCspreadparmplotter.f90`**. This shows the generation number on the x-axis and the parameter value on the y-axis (in this case $e \sin \omega$). This particular example was made using **`demcmcELC.f90`** with 15 free parameters and 150 chains.

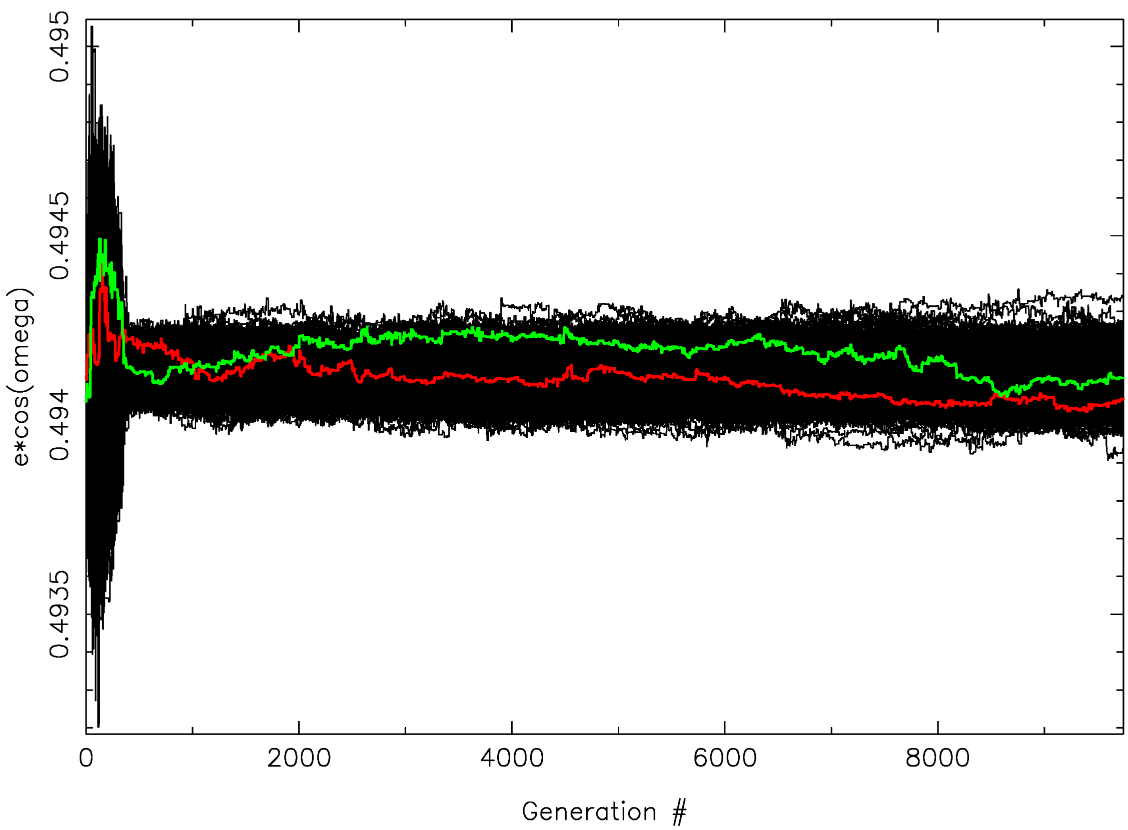

Figure 13: An example plot from **ELCspreadparmplotter.f90**. This shows the generation number on the x-axis and the parameter value on the y-axis (in this case $e \sin \omega$). This particular example was made using **hammerELC.f90** with 58 free parameters and 800 walkers.