

Computing Binary Star Observables

R. E. Wilson

Astronomy Department, University of Florida, Gainesville, FL 32611

W. Van Hamme

Department of Physics, Florida International University, Miami, FL 33199

Revision of May 2016

Table of Contents

1	Introduction	1
2	Running the Programs	2
3	Output Options for the LC Program	3
4	How To Use the LC and DC Programs	5
4.1	Getting Started	5
4.2	LC Program Controls and Fixed Quantities	6
4.3	LC Program: Internally Computed Output Quantities	8
4.4	LC Program: Binary System Parameters	8
4.5	Input Temperatures from Spectra	10
4.6	Early Rough Curve Fitting Experiments	10
4.7	Sigmas and Del's for the DC Program	11
4.8	Three Kinds of Weights	11
4.9	Standard Deviations Generated by DC for Curve-dependent Weight Estimation . .	12
4.10	Automated Curve-dependent Weight Computation	12
4.11	Computational Spot Algorithms, Old and New	13
4.12	LC Input from DC output	13
4.13	Do Non-absolute Light Curve Computations Respond to Absolute Dimensions? . . .	14
5	Operation Modes (Solution Constraints)	14
6	Luminosity vs. Light (Flux)	17
7	Circumstellar Light-attenuating Regions	18
8	Spectral Line Profiles	19
9	Time/Phase Smearing	20

10 Model Parameters	21
10.1 Curve-independent parameters that cannot be adjusted by DC	21
10.2 Curve-dependent parameters that cannot be adjusted by DC	22
10.3 Curve-independent parameters that can be adjusted by DC	22
10.4 Curve-dependent parameters that can be adjusted by DC	25
10.5 Ephemerides from Whole Curves, along with Apsidal Motion and Third Body Pa- rameters	26
10.6 Unified Ephemeris Solutions	27
11 Parameter Order	28
12 Control Integers, Units, Scaling Factors, Special Quantities	30
12.1 Those Common to LC and DC	30
12.2 Those for LC only	32
12.3 Those for DC only	33
13 Solution Convergence and Related Issues	37
13.1 Method of Multiple Subsets (MMS)	39
13.2 Levenberg-Marquardt (L-M)	39
13.3 Vector Length Reduction (VLR)	40
13.4 MMS, VLR, or L-M?	40
13.5 Error Distributions and Standard Error Estimates	41
14 Interactive Branching and Iteration of Solutions	42
15 Photometric Bands and Radiative Physics	43
15.1 Band IDs, Chemical Composition, Temperature and Gravity Ranges	43
15.2 Atmosphere to Blackbody Transitions	49
16 Special Features	49

17 Simultaneous Solutions	50
17.1 Absolute Flux Computations and Direct Distance Estimation (DDE)	50
17.2 $T-d$ and $T-d-A$ Theorems	51
17.3 Absolute Flux Solutions: Strategy and Procedures	52
18 Input Lines, Including Observations and Weights	55
19 The Scaling of Run Time	56
20 Common Difficulties	57
21 Summary of Program Version Differences	61
21.1 Differences from Pre-1992 Versions	61
21.2 Differences Between 1998 and 1992 Versions	61
21.3 Differences Between 2003 and 1998 Versions	63
21.4 Differences Between 2007 and 2003 Versions	63
21.5 Differences Between 2010 and 2007 Versions	64
21.6 Differences Between 2013 and 2010 Versions	65
21.7 Differences Between 2015 and 2013 Versions	67
22 Summary of Control Integers	68
22.1 [0,1] Control Integers	68
22.2 [0,1,2] and [1,2] Control Integers	68
22.3 [1,2,3] and [1,2,3,4,5,6] Control Integers	68
23 Acknowledgments	69
A Sample LC Input File	73
B Sample DC Input File	75

List of Tables

1	Temperature Limits	44
2	Bandpass List	45
2	Bandpass List	46
2	Bandpass List	47
3	Absolute Flux Calibrations	53
3	Absolute Flux Calibrations	54
4	Dimensioning vs. Grid Fineness	59
5	Physical and Astronomical Constants	66

1. Introduction

The main theory, organization, concepts and much of the mathematics of the model have been described and quantified in papers by Wilson & Devinney (1971), Wilson (1979, 1990, 1993, 2008, 2012a,b), Van Hamme & Wilson (2003, 2007), Wilson, Van Hamme, & Terrell (2010), and Wilson & Van Hamme (2014). Some of the model’s lesser known capabilities are briefly discussed in Wilson (2012a). The reader is referred to those papers for background. Since it would be impractical to cover the programming ideas within a reasonable length, only the 1993 paper goes significantly into programming, and there only in abbreviated form. This booklet also is not an explanation of programming ideas—its purpose is to tell how to use the model.

The overall program consists of a main FORTRAN¹ program (**LC**) for generating light and radial velocity curves, spectral line profiles, images, conjunction times, and timing residuals, plus a differential corrections main program (**DC**) for parameter adjustment of light curves, velocity curves, and eclipse timings by the Least Squares criterion. About three dozen subroutines are used by both main programs.² The present program has grown over the years from much simpler beginnings. Growth has consisted of occasional improvements in regard to generality, speed, and elimination of bugs, perhaps about every six months, punctuated by eight revisions. The revision of 1982 introduced eccentric orbits, asynchronous rotation, capacities to do several new kinds of constrained solutions, computation of velocity curves (with proximity and eclipse effects), simultaneous light and radial velocity solutions, and a simple star spot capability. Most features of the 1982 version are described in Wilson (1979). The revision of 1992 had options for detailed reflection and non-linear (logarithmic law) limb darkening, adjustment of spot parameters, an optional provision for spots to drift over the surface, capability for following light curve development over large numbers of orbits, and greater speed. The third revision (1998) included semi-transparent circumstellar clouds, a simple spectral line profile capability for fast-rotating stars, inclusion of the Marquardt λ factor in differential corrections solutions, generation of sky coordinates for images, an option to work with either observed times or phases, additional solution parameters (T_0 , P_0 , dP/dt , and $d\omega/dt$), conversion of the entire program to double precision, and some other improvements that are listed in §21.2 on pages 61–63. The fourth revision (2003) was mainly a conversion from the previous effective wavelength characterization of bandpasses to one based on integrations over actual bands of standard photometric systems. The radiative functions depend on effective temperature, $\log g$, and chemical composition and are applied locally. A few changes also were made in the constraints applied to some overcontact binaries (mode 3 operation). The fifth revision of 2007 included light-time and velocity shifts due to third bodies (hereafter 3b’s), adjustment of corresponding 3b

¹Apparently the f77 FORTRAN compiler used by most persons over recent decades is becoming less available. However, **LC** and **DC** should run just as well under the newer f90 compilers.

²In this monograph, program and subprogram names are in boldface type (e.g. **LC**), FORTRAN variable names are in sans serif style (e.g. XINCL), and FORTRAN statements and file names are in typewriter style (e.g. open(unit=22,...)).

parameters (see §10.5 on page 26), computation of x, y limb darkening coefficients as functions of local T_{eff} , $\log g$, and $[M/H]$ for any of three laws (see §10.1 on page 21 and §10.4 on page 26), option of magnitude or flux input to **DC** (see explanation of control integer **MAGLITE** in §12.3 on page 36), simplified re-dimensioning (provided by A. Prša), and several additions and improvements in output quantities and formats. Instructions for re-dimensioning are in the **LC** and **DC** main programs. The primary purpose of the sixth (2010) revision was to introduce absolute flux solutions (described in §17.1 on page 50) that allow Direct Distance Estimation (**DDE**). In these absolute solutions, distance (actually $\log_{10} d$) is an ordinary solution parameter with a standard error, and with no spherical star assumption in any part of the process. Accordingly one need not hesitate to derive **DDE** distances for semi-detached or over-contact binaries (i.e. there is no reason to favor well-detached binaries for distance estimation with **DDE**). Also possible are improved solutions for ellipsoidal variables, and for EBs with very shallow eclipses, via independently known distance (inverse distance estimation, or **IDE**). Absolute flux solutions also can estimate temperatures of both EB components under suitable circumstances. Input integer **IFCGS** tells **LC** and **DC** whether to operate in the program flux unit or in centimeter-gram-second (cgs) units. Set **IFCGS** to 0 for arbitrary flux and luminosity units, or set **IFCGS** to 1 for standard units (respectively $\text{erg s}^{-1} \text{cm}^{-3}$ and $\text{erg s}^{-1} \text{cm}^{-1}$). The seventh (2013) revision added a high-precision star spot algorithm (see §4.11 on page 13) that allows for developing spots characterized by times of onset, maximum size and disappearance. Spot drift is independent of star rotation. The radiative treatment is now based on cubic spline fits to stellar atmospheres (formerly Legendre polynomials), and the original list of 25 photometric bands introduced in 2003 is expanded to 93 bands (see §15 on page 43). Time/phase smearing is a light curve option (see §9 on page 20). **DC** can now (optionally) find its own curve-dependent weights (see §4.9 on page 12). The eighth revision (2015) has added eclipse timings as a data input option. For details of this capability, see §10.6 on page 27. A 94th band (Lunar UV Telescope) is added in this minor update of 2016.

2. Running the Programs

To make things simple, the light and velocity curve program (**LC**) and the differential corrections program (**DC**) are supplied complete with input data sets, so that all one needs to do to get started is to run the programs with the sample input data. There should be few, if any, machine-dependent problems, as those have been eliminated via feedback from users. If such a problem is found, please communicate it to R.E.W. However, there may be a trivial problem with the arcsine and arccosine routines, which are **DASIN** and **DACOS** on some systems and **DARSIN** and **DARCOS** on others. That problem can easily be fixed. Then just change the input numbers to run a particular binary star problem. Be sure to keep at least one copy of the supplied sample data in case the input formatting gets scrambled or shifted. Of course, one always can re-construct the correct format by comparing the program's **READ** and **FORMAT** statements with the input lines, but save the original data anyway in the interest of keeping things simple. It is not recommended to change the programs, but if you must, be sure to keep a copy of the original version for comparison. Tinkering can introduce bugs

that may not show up in particular cases but jump out at you later on.

Compilation is quite direct as **LC** is supplied in one composite file and so is **DC**, including all required subroutines. Most subroutines are in both modules, with memory cost of the redundancy being of little significance. Some auxiliary data files (for limb darkening, etc.) need to be in the sub-directory where the program is run, or links to them must be provided.

3. Output Options for the LC Program

The kind of output produced by **LC** is determined by control integer **MPAGE** and can be light curves (**MPAGE**=1), radial velocity curves (**MPAGE**=2), spectral line profiles (**MPAGE**=3), relative star dimensions vs. phase (**MPAGE**=4), sky coordinates for producing images (**MPAGE**=5), conjunction times and eclipse timing residuals (**MPAGE**=6 and **KSTEP**=0), or computed conjunction times (**MPAGE**=6 and **KSTEP**=1, 2, 3, ...).

For **MPAGE**=1 (light curves), the first five output columns in the block after the header information contain time (col. 1), phase (col. 2), separate light for stars 1 and 2 (cols. 3, 4), and the combined system light from stars 1 and 2 plus third light (col. 5). **Star 1 is, by definition, the one at superior conjunction near phase zero when parameter PSHIFT is entered as zero.** It will be the one eclipsed near phase zero if there are eclipses. Light is in the program unit, which is explained in Wilson (1993). The program unit is, in a sense, an absolute unit of observable flux because it can be converted to standard physical units if absolute star luminosities and a definite observer distance are specified. Light (observable flux) and luminosities are discussed in §6 on page 17. Column 6 contains the light of column 5 re-scaled (normalized) to a specified input value (labeled **FACTOR**) at a specified input phase (labeled **PHN**). For example, one can require that the normalized light be 1.2000 at phase 0.1500. This re-scaling provision is only for convenience in working with graphs, and has no relevance to differential corrections solutions. That is, the **DC** program works only with direct flux and does not even know about normalized light. Column 7 is the star separation, with the unit being the relative orbital semi-major axis ($a = a_1 + a_2$). Column 8, called “set-level magnitude,” is system brightness expressed in magnitudes, pinned so as to pass through a definite point in a graph of magnitude vs. phase. The point is specified by input quantities **MZERO** and **PHN**. For example, if **MZERO** is +7.300, column 8 will read 7.300 at the phase of normalization (**PHN**), which is the same phase of normalization used for column 6. Column 9, called “direct magnitude,” is $-2.5 \log_{10} \ell$, which is the exact inverse of flux used by **DC** for its solutions, as generated in **DC** by $\ell = 10^{-0.4m}$ in the case of magnitude input. Here $\ell = \ell_1 + \ell_2 + \ell_3$ (column 5). Accordingly, **LC**’s direct magnitudes will give theoretical light curves that correspond to **DC** solutions and can be used to make graphs to compare theory with observation. Column 10 is the light-time shift (Δt , in days) caused by the third body.

For **MPAGE**=2 (radial velocities), the first two output columns are again time and phase. Columns 3 and 4 are dimensionless radial velocities for the two stars (in circular relative orbit circumference,

$2\pi a$, per orbit cycle). Columns 5 and 6 are eclipse-proximity corrections that are included in columns 3 and 4, respectively, and in the same dimensionless unit. Columns 7 and 8 are velocities in kilometers per second if velocity unit **VUNIT** was entered as 1.00, or, in general, in unit **VUNIT** km/s. Column 9 is the light-time shift.

For **MPAGE**=3 (spectral line profiles), the output is by blocks according to phase, with results for stars 1 and 2 given first and second, respectively, within each phase block. Column 1 is the equivalent velocity difference between the reference wavelength and the wavelength of the profile point, in unit **VUNIT**. Column 2 is the corresponding wavelength difference, in microns. The reference wavelength is entered with the main binary star parameters (labeled “**wv lth**” in the output). Column 3 has the actual wavelengths of the profile points, in microns. Column 4 is the profile in terms of a flat continuum at flux 1.00000. Column 5 is the profile in terms of a continuum that can be shifted vertically and can have a slope. Guard against generating excessive output with **MPAGE**=3 (usually one does only one phase at a time). Input for **MPAGE**=3 operation is described in §8 on page 19. Extra data lines beyond those for other **MPAGE** values are required. The program will crash without those lines if **MPAGE**=3.

For **MPAGE**=4 (star figures), **LC** lists the pole, point, side, and back relative radii (R/a) of each star vs. time and phase. This provision can be useful for eccentric orbit cases, as it shows the variation of figure with phase.

For **MPAGE**=5 (images), **LC** produces only two output columns for each time/phase, beyond the usual header information. Those columns are y_{sky} and z_{sky} rectangular plane of sky coordinates of the projected surface elements of the two stars. A picture of the binary at a given phase can be made by sending just those two columns to a plot program such as **GNUPLOT**, **MONGO**, etc. Any spots or parts of spots that may be in view will show in the pictures, although there is no distinction in the pictures between bright and dark spots. The images show only spot location, not spot surface brightness (there is no gray scale). The origin (0,0) of the image coordinates is at the binary system center of mass.

For **MPAGE**=6, **LC** computes conjunction times or their differences from observed eclipse times (i.e. residuals), both with and without a dP/dt term in the latter case. Only for **MPAGE**=6 does **LC** input include observations (for **MPAGE**=1 to 5, input consists of integers that specify program operation and parameter values but no observations). Of course the reason is that *residuals* are the needed output and residuals require observations.

Control integer **KTSTEP** is entered only with **MPAGE**=6 and allows computed conjunction times to be spaced by an arbitrary number of whole orbit cycles rather than only by one cycle. The times include dP/dt and third body light-time effects, which may be useful for planning observations over long intervals. The case **KTSTEP**=0 is not for conjunction times but for observed minus computed timing residuals, which are needed to make eclipse timing diagrams. Of course the observed times must be included in the **LC** input file. Residuals for both a linear ($dP/dt=0$) and a non-linear ephemeris are written, and light-times due to a third body are included.

4. How To Use the LC and DC Programs

Adjustment of parameters while fitting light and velocity curves normally involves both subjective (**LC**) and objective (**DC**) iteration. It is assumed that the user has reasonable background knowledge of binary stars and can show good judgment in deciding which parameters to fix from theory, which to fix from other kinds of observations, and which to adjust. Solution constraints, if there are to be any, must also be decided upon. In most situations, some of the decisions are obvious while others are debatable, and seldom will two persons make the same set of choices, even if given exactly the same circumstances. Therefore, this section goes only into the “mechanics” of solutions, which means how to use the **LC** and **DC** programs. Historical background is in Wilson (1994a) and Wilson (2001) while intuitive ideas are discussed in Wilson (1994b, 2006).

4.1. Getting Started

Getting started should only be a matter of running the programs with the supplied sample input data and changing the numbers to those of particular binaries. Most persons will want to begin with a quick graphical fit, so as to be somewhere near to a proper solution. The following discussion specifically concerns non-absolute light curves in **LC**, although absolute light curves and radial velocities would be fitted in much the same way. It is best to be ready to feed either column 5 (light in program units) or column 6 (normalized light) to a plot program such as MONGO or GNUPLOT. Remember that the normalized light column has no counterpart in **DC**, so after a few first cuts with normalized light (to reach rough agreement with the observations) switch from use of column 6 to use of column 5. At this point one should effect a correct transfer such that the column 5 numbers will be approximately the same as the column 6 (normalized) numbers that may have been made to fit the light curve(s). This little problem involves the estimation of a scaling constant, or sometimes more than one. Please consult §6, **Luminosity vs. Light (Flux)**, on page 17 if you are not conversant with the distinction between “light” or “flux” on the one hand (an output quantity), and “luminosity” on the other (an input quantity). This will sound trickier than it is, but do not be concerned with making a perfect transfer from column 6 to column 5 output because you are iterating anyway in **DC** and only need to get close. Remember that the light of each star scales with its luminosity, while third light (ℓ_3) is just a direct add-on. Also remember that in all modes of program operation except modes 0 and -1 , the luminosity of star 2 (L_2) is computed by the program and scales with that of star 1 (L_1), assuming fixed values for all other parameters. Therefore the main, or perhaps only, scaling parameter in most situations is L_1 . The simplest case is in mode 1, 2, 3, 4, 5, or 6 with no third light. Then, for example, doubling L_1 will double column 5 output light. If there is third light, that also should be doubled if column 5 light ($\ell_1 + \ell_2 + \ell_3$) is supposed to double. In mode 0, L_2 does not scale with L_1 , so L_2 also would then need to be doubled. In mode -1 , L_1 has only minor influence (through reflection heating), so L_2 and ℓ_3 mostly control the scaling, although L_1 also needs to be scaled in the general case. So to make column 5 agree with column 6, bump L_1 and ℓ_3 (most modes) or L_1 , L_2 , and ℓ_3 (modes 0 and -1) up or

down by a fixed factor.

4.2. LC Program Controls and Fixed Quantities

These are brief statements about essential purpose. The first entry is the FORTRAN name and the entry in parentheses is the output name written by **LC**. The quantities are defined more properly elsewhere in this document, usually in connection with their use in the **DC** program. See the index of FORTRAN names for page numbers.

FORTTRAN names of most quantities are the same in the **LC** and **DC** programs. Slight exceptions are for quantities that are single numbers in **LC** and arrays in **DC**, for example names **X1** in **LC** and **X1A** in **DC**, where the added “A” stands for “array.”

MPAGE (mpage): The kind of output produced by **LC** (light curve, RV curve, etc.).

KTSTEP (ktstep): Positive integer that steps *computed* superior and inferior conjunction times by **KTSTEP** whole orbit cycles. Entered only with **MPAGE**=6. If **KTSTEP**=0, output is timing residuals instead of conjunction times.

NREF (nref): The number of reflections in detailed reflection computation.

MREF (mref): Specifies whether simplified or detailed reflection is to be applied.

IFSMV1, **IFSMV2** (ifsmv1, ifsmv2): These tell whether spots are allowed to move in longitude or are to be stationary.

ICOR1, **ICOR2** (icor1, icor2): Tell whether computed RV’s are to include proximity and eclipse corrections or not.

IF3B (if3b): Tells whether there is a light time effect due to a kinematic third body.

LD1, **LD2** (LD1, LD2): Tells the limb darkening law to be applied (linear, log, or square root) for stars 1 and 2.

KSPEV (kspev): Tells whether spots are to grow and decay or not.

KSPOT (kspot): Tells whether the old simple spot model or a more accurate one, based on position vectors, applies.

NOMAX (nomax): Tells whether the spot growth and decay timewise profile is trapezoidal or triangular.

IFCGS (ifcgs): Tells whether light curves are to be computed in arbitrary units or absolute physical units.

JDPHS (JDPHS): Tells whether time or phase is the independent variable in light and RV curves.

PSHIFT (Ph. shift): A constant shift that can be applied to move a light or RV curve in phase.

DELPH (del phs): Phase spacing of abscissas in Gaussian quadrature for phase or time smearing simulation.

NGA (NGA): Number of Gaussian quadrature abscissas in phase or time smearing simulation.

STDEV (fract. sd.): Fractional standard deviation of simulated scatter in output light or RV. Enter zero for zero output noise.

NOISE (noise): Specifies how output scatter (simulated noise) is to vary with light level.

SEED (seed): A seed number that determines individual noise excursions of output light or RV points.

HJDST (JD start): Start time for output points. Applied only if JDPHS=1. The time system does not have to be Julian Date.

HJDSP (JD stop): Stop time for output points. Applied only if JDPHS=1. The time system does not have to be Julian Date.

HJDIN (JD incr): Increment for times of output points. Applied only if JDPHS=1.

PHSTRT (Ph start): Starting phase for output points. Applied only if JDPHS=2.

PHSTOP (Ph stop): Stop phase for output points. Applied only if JDPHS=2.

PHIN (Ph incr): Increment for phases of output points. Applied only if JDPHS=2.

PHN (Ph norm): Phase of normalization for columns of normalized light and set magnitude.

PHOBS (Ph Obs): Phase at which a spectroscopic temperature was estimated (for use in a possible side computation to convert directly estimated temperature to flux-weighted mean surface temperature).

TOBS (Tobs): Spectroscopic temperature at phase PHOBS.

TAVSP (Tavesp): Computed flux-weighted mean surface temperature.

LSP (LSP): Tells whether TOBS and TAVSP are for star 1 or star 2.

MODE (MODE): Integer to determine which model constraints, if any, are to be applied (for example semi-detached, with star 2 exactly filling its lobe).

IPB (IPB): Control integer that can sever the temperature-luminosity connection.

IFAT1, IFAT2 (IFAT1, IFAT2): Tells whether blackbody or stellar atmosphere radiative physics applies to stars 1 and 2.

N1, N2 (N1, N2): Grid fineness integers for stars 1 and 2 (number of latitude rows per hemisphere).

VUNIT (V UNIT(km/s)): Radial velocity input unit in kilometers per second.

IBAND (band): Photometric band identifier. For example IBAND is 6 for Johnson B and 7 for Johnson V.

4.3. LC Program: Internally Computed Output Quantities

VFAC (V FAC): Conversion factor from dimensionless velocity units (circumference of the relative orbit per orbit cycle) to kilometers per second.

NSPOT1, NSPOT2 (Nspot1, Nspot 2): Numbers of spots on stars 1 and 2.

TPOLE1, TPOLE2 (Tpole 1, Tpole 2): Polar effective temperatures on stars 1 and 2.

4.4. LC Program: Binary System Parameters

HJD0 (J.D. zero): Ephemeris reference time – usually heliocentric Julian Date, although any consistent time system can be adopted.

PZERO (Pzero): Orbit period in mean solar days at time HJD0.

DPDT (dPdt): Rate of period change, assumed constant (dimensionless).

ARGPER (Arg. Per): Argument of periastron in radians.

DPERDT (dPerdt): Rate of change of argument of periastron in radians per mean solar day.

THE (Th e): Semi-duration in phase (phase range 0 to 1) of the eclipse of star 1 (ordinarily only for X-ray binaries – applied only in mode -1).

ECC (ecc): Binary system orbital eccentricity.

SMAxis (s-m axis): Semi-major axis of binary system relative orbit ($a = a_1 + a_2$) in solar radii.

F1, F2 (F1, F2): Ratios of axial rotation rate to mean orbital rate for stars 1 and 2.

VGAM (Vgam): Radial component of the systemic velocity in unit VUNIT.

XINCL (Incl): Binary system orbital inclination (zero in plane of sky, 90 degrees edge-on).

G1, G2 (g1, g2): Exponents in bolometric gravity brightening (a.k.a. darkening) law for stars 1 and 2.

ABUNIN ([M/H]): System metallicity as usually defined logarithmically, relative to the Sun.

FSPOT1, FSPOT2 (Fspot1, Fspot2): Spot angular drift rates in longitude for stars 1 and 2. Rate 1.0000 means that drift just matches the mean orbital angular rate.

T1, T2 (T1, T2): Flux-weighted mean surface temperatures for stars 1 and 2 in unit 10000 K.

ALB1, ALB2 (Alb 1, Alb 2): Bolometric albedos for stars 1 and 2.

POT1, POT2 (Pot 1, Pot 2): Modified surface 'potentials' for stars 1 and 2.

RM (M2/M1): Mass ratio of binary components, m_2/m_1 .

X1BOLO, X2BOLO (x1(bolo), x2(bolo)): Bolometric limb darkening coefficients for stars 1 and 2 in linear cosine law, and in the linear terms of the logarithmic and square root laws. See §10.1 on page 21 for forms of the laws.

Y1BOLO, Y2BOLO (y1(bolo), y2(bolo)): Bolometric limb darkening coefficients for stars 1 and 2 in logarithmic or square root terms

A3B (a3b): Semi-major axis of the relative 3b orbit in solar radii.

P3B (P 3b): Period of the 3b orbit in mean solar days.

XINC3B (incl 3b): Inclination of the 3b orbit to the plane of the sky, in degrees.

E3B (e 3b): Eccentricity of the 3b orbit.

PER3B (arg. perr. 3b): Argument of periastron of the 3b orbit in radians.

TC3B (T conj 3b): Time of superior conjunction of the EB center of mass in its reflex motion about the triple system barycenter, in the adopted time system (usually HJD or HJED).

RM3B ($m_3/(m_1+m_2)$): Ratio of 3b mass to EB mass.

L1, L2 (L1, L2): Component bandpass luminosities (input), integrated over 4π steradians at each star. Only their ratios matter, for example L_2/L_1 or $L_1/(L_1+L_2)$, since non-absolute EB solutions of uncalibrated light curves cannot directly supply luminosities in physical units. Usually $L_1/(L_1+L_2)$ for each photometric band is reported in publications. Luminosities in physical units (erg/s/cm) and in solar units are in the output of program LC, based on the physical and geometric input parameters.

X1, X2 (x1, x2): Limb darkening coefficients in the linear cosine law.

Y1, Y2 (y1, y2): Coefficients in the non-linear limb darkening terms.

EL3 (el3): Third light at a user-specified phase in the same unit as input light (flux).

OPSF (opsf): Opacity in a specific band, κ_{sb} , explained in §7, and related to attenuation by circumstellar clouds.

MZERO (m zero): A reference magnitude that can shift a magnitude light curve (LC output) vertically. See §3.

FACTOR (factor): A scaling factor for LC output curves in flux (not magnitude). See §12.2 and

§4.6.

WL (wv lth): Wavelength of light, in microns, now used only for circumstellar matter and spectral line profiles.

CALIB (calibration): Flux in $\text{erg s}^{-1}\text{cm}^{-3}$ for a zero magnitude star in a given photometric band. One of the cm factors represents effective bandwidth.

4.5. Input Temperatures from Spectra

A small and easily resolved problem concerns the distinction between directly observed temperature (T_{obs}) and mean surface temperature (T_{mean}), given that T_{mean} (the actual model parameter) is a global quantity while T_{obs} depends on phase and inclination due to phenomena such as gravity brightening and the reflection effect. Rigorous conversion from T_{obs} to T_{mean} is covered in the appendix to Wilson (2008), with full explanation and mathematics. The **LC** program asks for input of an observed (presumably spectroscopic) temperature (TOBS) and its phase (PHOBS) and computes a mean surface temperature (labeled Tavesp). **LC** knows the orbital inclination since that is a system parameter. Input integer LSP is 1 or 2 depending on whether those temperatures are for star 1 or 2.

4.6. Early Rough Curve Fitting Experiments

In subjective (trial & error) fitting with normalized output the idea is to select some phase where the synthesized curve is made to match the observed curve so that one can concentrate on form rather than the scale. Usually one chooses some innocuous phase outside eclipse where the brightness is not changing very fast. Vertical shifts of the whole normalized curve are made by changing input quantity FACTOR, if working in direct light, or MZERO if working in magnitudes. After you switch to column 5 output (getting ready for **DC**), control of vertical scaling passes to L_1 , L_2 , and ℓ_3 , as explained above. After gaining some experience with **LC**, you will be using the normalized output only for a few preliminary runs and staying with program-unit light thereafter.

To start running **DC**, begin with the sample data, which contain control integers, initial parameter estimates, radial velocity curves for stars 1 and 2, light curves in four bands, and eclipse times. Run **DC** and inspect the output to be sure you understand everything. You may want to change a few things in the input to verify that the output responds as expected.

4.7. Sigmas and Del's for the DC Program

To begin a solution with your own data, change the sample input numbers to those appropriate to the observed binary, being careful to keep the original format. By this time you should have made a reasonable number of fitting experiments with **LC**, decided on the mode of program operation, decided on the parameter set and subsets to be adjusted, and made (perhaps rough) estimates of the **SIGMA**'s (standard deviations) of the observed curves (see next sub-section). You should also give a reasonable amount of thought to the sizes of the increments (**DEL**'s) used in forming the numerical derivatives. It is easy to fall into the error of trusting the **DEL**'s in the sample data to be appropriate for your binary, and sometimes they will be, but you cannot rely on that assumption. For example, suppose the mass ratio **DEL** is 0.01 and you just leave that value in for two binaries with mass ratios of 5.00 and 0.10. The increment for the first binary is one part in 500 (probably too small), while that for the second is one part in 10 (grossly too large, of course).

There is no input **DEL** for the spot timing parameters because their derivatives are hybrids, part analytic and part numerical. The analytic parts need no increments and the numerical parts use the spot radius **DEL**.

4.8. Three Kinds of Weights

Weighting of observations is discussed in Wilson (1979, pages 1064-1065), Wilson (1988) and Kallrath & Milone (2009). Briefly, **DC** applies three kinds of weights, which are “intrinsic” weights (assigned by the user to the individual observations), “curve-dependent” weights (based on estimated standard deviations [**SIGMA**'s] at a reference phase), and “level-dependent” weights (computed by **DC** according to the input parameter **NOISE**, which tells how observational scatter scales with light level). **NOISE** should be set to 1 for scatter that scales with the square root of the light level, such as counting statistics, and to 2 for scatter that scales with the light level, such as scintillation noise or fluctuations in sky transparency. If **NOISE** is set to 0, no level-dependent weighting is applied. Level-dependent weighting does not apply to velocity curves. **SIGMA**'s pertain to the directly entered velocity and light values. An exception is where the **SIGMA**'s were actually measured from individual observations, but normal points (i.e. averages) are entered. In that case, the number of individual points in each normal point should be incorporated in the assigned weight of each point. In some papers the **SIGMA**'s have been used to manipulate relative curve weights (usually RV vs. light curve weights). However, the **SIGMA**'s need to be *CALCULATED*, not set arbitrarily. One can start with approximately correct **SIGMA**'s but they cannot be off by large factors in a relative sense. Of course, only ratios of weights matter. See §4.10 for a recent provision that allows **DC** to calculate its own curve-dependent **SIGMA**'s and weights.

4.9. Standard Deviations Generated by DC for Curve-dependent Weight Estimation

DC writes a table of standard deviations (σ) for the 1 or 2 velocity curves, NLC light curves, and the dataset of eclipse times, immediately after writing the input data, and then after each iteration of each subset solution (so for that iteration). The table heading is “*Standard Deviations for Computation of Curve-dependent Weights.*” The σ ’s are listed in the order their curves occur in the input. Originally they were intended for use in subsequent **DC** runs to calculate curve-dependent weights so as to specify the relative influences of the various curves, and that option remains. That is to say, one can enter rough guesses for the σ ’s for the first few iterations, then enter the calculated ones for the middle and fine-tuning runs. The rough guesses can be made by putting a straight line through a representative section of the data. However, the automated weighting of §4.10 will usually make such personal intervention unnecessary.

DC estimates σ ’s for the whole curves and also estimates combined σ ’s within two restricted phase ranges. The idea here is that some phase ranges may have anomalous excursions or increased scatter, often due to unknown phenomena, so as to give false estimates of observational precision. The start/stop phases for the restricted ranges are labeled “phase 1, phase 2, phase 3 and phase 4” in the **DC** output, and their FORTRAN names are SPHAS1, SPHAS2, SPHAS3 and SPHAS4. With respective values of, say, 0.07000, 0.43000, 0.57000 and 0.93000, they would define two phase ranges, one from 0.07000 to 0.43000, and one from 0.57000 to 0.93000. Usually it is best to choose a modest range for the “representative phases,” or perhaps an average of two such ranges. Often one would adopt intervals around the maxima. The table also contains the reference flux and reference magnitude (for each band) that are averages over the two chosen phase ranges. To have only one range count, enter a null range for the other one (such as 0.75 to 0.75). The reference flux can be used to compute σ at a general light level, given σ_{ref} at the reference level. The relation is $\sigma = \sigma_{\text{ref}}(\mathcal{F}/\mathcal{F}_{\text{ref}})^{n/2}$, where \mathcal{F} is flux and n is the NOISE exponent described in §12.3 (page 36) and in §18 (page 11). In the unlikely and easily prevented event that *no* points of a given light curve fall within either of the specified phase ranges, **DC** adopts the average for the entire phase range of 0.0000 to 1.0000 and writes a message to that effect as a warning about the problem.

4.10. Automated Curve-dependent Weight Computation

Curve-dependent weighting (where a common weight is applied to each light or RV curve, or to a collection of eclipse timings) can be tedious and often neglected. Such weights might be separately computed from observational scatter as a preliminary step in the analysis or might be supplied by the observers. An alternative to separately computed or observer-specified curve-dependent weights is to let **DC** compute curve-dependent weights from **DC**’s σ estimates. This **DC** facility eliminates a step in the solution process, thereby decreasing the likelihood of weighting being bypassed or handled with insufficient care. The procedural decision can be decided on a curve by curve basis (see explanation for control integer KSD on page 36). For example, with two velocity curves and

three light curves, one could have **DC** adopt observer-specified input σ 's for the first velocity curve and the middle light curve and apply **DC**'s own σ estimates for the other three curves. That could be a good strategy if the best information on precision is from the observers for some curves and from observational scatter for others. The option for automated curve-dependent weighting came in with the 2013 version, for light curves and RV curves, and now extends to eclipse timings. The importance of consistent weighting must be emphasized, as the three data types typically differ in order of magnitude and in units. Without proper weighting, astrophysical results would depend strongly on the arbitrary scale factors associated with units. Proper weights assure that the several kinds of data contribute to results quantitatively according to their precision. The “curve by curve” description means that whether weights are to be computed by the machine or just assigned is specified individually for each curve or timing dataset.

4.11. Computational Spot Algorithms, Old and New

The old spot algorithm that dates from about 1985 has now been superseded by a far more accurate one (Wilson 2012b) whose spots have fractional areas based on position vectors. The old logic remains in the program to enable comparisons with results from the new logic, but usually the new scheme will be preferable. Set **KSPOT**=1 for the old logic or **KSPOT**=2 for the new logic. Remember not to try star spots that are unduly small in angular radius if **KSPOT**=1 because there are then no fractional areas to the spots—they have staircase-like boundaries where their idealized circular outlines encounter the finite grid elements. Of course, the roughness of the simple **KSPOT**=1 treatment becomes particularly troublesome for small spots, and very small spots might not show up at all. The new (**KSPOT**=2) logic also allows spots to age (i.e. grow and decay), parameterized by times of onset, maximum size and disappearance that can be adjusted by **DC**. Spot motions in longitude are now governed by drift parameters **Fspot1** and **Fspot2** for stars 1 and 2, respectively, rather than by star rotation parameters **F1** and **F2**, as before, so spot motion is now decoupled from star rotation. To have spot motions and star rotations be the same, set **Fspot1** equal to **F1** and **Fspot2** equal to **F2**.

4.12. LC Input from DC output

If input integer **IFLCIN** is set to 1, **DC** writes an input file for the **LC** program (called `lcin.input_from_DC`) from the numbers that enter its very last iteration of a given submission. Of course the purposes are to avoid copying mistakes and to save typing. If **DC** does the base set and then, for example, four subsets, the **LC** input file will correspond to the last iteration of the fourth subset (so the order of subset computation and the number of iterations for the last subset both matter for the **LC** input file).

4.13. Do Non-absolute Light Curve Computations Respond to Absolute Dimensions?

Persons familiar with very old program versions will notice a (usually subtle) response of light curve output to changes in absolute masses and dimensional scaling in the **LC** and **DC** versions of 2003 and later. In versions of 1998 and earlier, one could enter any period or orbit size without affecting light curves, as the scaling of observable light (output) from luminosities (input) involved temperature but not $\log g$ or chemical composition. That was because the old stellar atmosphere routine applied only to main sequence stars of normal composition, but now $\log g$ and composition do affect light curve output, and the programs essentially compute $\log g$ from GM/R^2 (really from local conditions, including rotation and the other star's gravity), with M and R dependent on period and absolute size. In those old program versions one could even enter grossly wrong P 's and a 's without adverse light curve effects, and some users may have become accustomed to that circumstance. However, correct P , a , and $[M/H]$'s should be used with the newer radiative treatment. Actually, the attendant effects are small, barring radically wrong values, but one should be consistent. Where orbit size is unknown, make a best guess rather than entering unrealistic numbers. In a non-simultaneous light-velocity solution, be sure that the final semi-major axis (a) from the velocities is also used for the light curves. That condition will be satisfied automatically in a simultaneous solution. **LC** and **DC** are made to be mutually consistent but will have different L_2 's and light if absolute dimensions and masses differ between the two programs. Naturally, the foregoing warnings do not apply for blackbody computations, where the programs' light curves are unaffected by absolute masses and dimensions, although absolute flux computations (IFCGS=1) will of course show first-order responses to dimensional changes.

5. Operation Modes (Solution Constraints)

Imagine that you know or are convinced of something about possible parameter values, and your “fact” is not in the form of a definite parameter value, but rather is a functional relationship among parameters. You do not claim to know the value of parameter c , but if someone tells you the values of parameters a and b , you then can compute c . The most common example is connected with one of the stars accurately filling a limiting lobe, as in a semi-detached case. It is impossible (or anyway, inadvisable) to say in less than a small monograph why we believe in the phenomenon of lobe filling and our ability to recognize semi-detached binaries as such. Of course, there are binaries for which we cannot be sure—the lobe may or may not be filled—but there are many for which we would “bet the ranch” that they are. In fact, for more than a few of the classical Algol type binaries, conflicts between mass ratios determined from radial velocities and mass ratios found from a lobe filling condition (through light curves) were eventually resolved in favor of lobe filling and light curves. S Cancri is a good example. For the simple case of a circular orbit and synchronous rotation, the mass ratio, m_2/m_1 , of a semi-detached binary determines the relative size of the lobe filling star, R_{lobe}/a . Since the model expresses star size via the Ω potential energy function, $\Omega = \Omega_{\text{lobe}}$

is determined by the mass ratio and cannot be adjusted under a semi-detached or double contact constraint (modes 4, 5, 6). If we have lobe filling with non-synchronous rotation there is a functional relation among three parameters (now also F to express the rotation rate), and we add parameter e if the orbit is eccentric. Application of the lobe filling condition, or another condition, reduces the number of free parameters by one. Because every parameter enlarges and complicates the parameter correlation matrix and thus weakens the solution, one should take advantage of every opportunity to eliminate free parameters. Essentially we thereby introduce information from sources external to the light and velocity curves, and we tell the Least Squares program “here is something about the binary that you cannot figure out from the data, but we are letting you know so that you can eliminate a whole dimension of incorrect solutions.” Each such condition can be represented as a constraint among possible parameter values, and a constraint or set of constraints is identified with a particular **operation mode** of **LC** and **DC**. Parameters that are set by constraints are no longer free parameters. The program computes them from the constraint relations and ignores their input values. Of course, they cannot be adjusted. Unfortunately, a few authors of other programs have failed to understand the idea of constrained solutions and have made comments something like “our program allows each star to have the radius that gives the best fit to the light curve, so that application of a lobe filling constraint is unnecessary.” This remark is akin to saying “we amputate the legs so as to make walking unnecessary.”

Mode –1: This mode is for X-ray binaries for which the eclipse duration of a compact object is known from X-ray observations. The compact star must be star 1. The applied constraint is that the surface potential of the ordinary star (star 2) must be such as to produce the observed X-ray eclipse duration. Therefore, the program computes the Ω_2 potential from the input q , F_2 , e , ω , i , and X-ray eclipse semi-duration, ϕ_e (FORTRAN name is **THE**). Note that solutions are not limited to circular orbits or synchronous rotation—any e between 0 and 1 is allowed, as is any F_2 . Note also that lobe filling is not implied. It is recommended to use equal temperatures for the two stars in most situations where one would apply mode –1 logic. With component 1 a neutron star, for example, it will be unrealistic to deal with the actual surface temperature (which usually will be in a bizarre surface distribution—perhaps 50,000,000 K in a small hot spot or two and far lower over the rest of the star. The tiny ultra-hot spot will contribute negligible optical flux anyway. Usually it will be best to assign star 1 a temperature equal to that of star 2, so as to be sure of having zero bolometric correction in the reflection effect, thereby keeping things relatively simple.

Mode 0: No constraints are applied in mode 0, and the component luminosity ratio is not even required to be consistent with the surface temperatures. That is, the program uses L_1 and L_2 as supplied, and does not recompute L_2 based on the input temperatures.

Mode 1: This is a mode for overcontact binaries, such as W UMa stars. Six or seven constraints are applied, depending on whether fixed or computed limb darkening is specified (viz. comments on LD1, LD2 on page 32). The first is that the surface potential of star 2 is equal to that of star 1 ($\Omega_2 = \Omega_1$). Another is that the polar temperature of star 2 is set by the gravity brightening law of the entire common envelope, which is done via Eqn. 8 of Wilson (1979). Other constraints are

that the gravity brightening, bolometric albedo, and perhaps limb darkening parameters, of stars 1 and 2 are the same.³ Finally, the luminosity of star 2 is computed from the other parameters via blackbody or stellar atmosphere radiation formulas. Therefore parameters Ω_2 , g_2 , T_2 , A_2 , L_2 , x_2 , and y_2 are not free in mode 1. A consequence of mode 1 operation is smooth variation of surface brightness over the entire common envelope with no discontinuity where the stars join together, and thus one continuous gravity brightening law. In mode 1 light curve fitting experiments, there will be essentially no freedom to change the relative primary and secondary eclipse depths, which will be set almost entirely by geometry (although also a little by gravity brightening, limb darkening, etc.).

Mode 2: This mode is for detached binaries with no constraints on the potentials. The only constraint is that the secondary luminosity, L_2 , is computed from the other parameters via the specified radiation prescription (blackbody or stellar atmosphere). That is, L_2 is coupled to the temperatures. The temperature-luminosity coupling can be severed by setting control integer IPB to 1 (normal value = 0). Mode 2 is the same as mode 0 except for the one constraint, and is exactly like mode 0 if IPB is set to 1.

Mode 3: This mode is for overcontact binaries and is like mode 1 except that the constraints on g_2 , T_2 , A_2 , x_2 , and y_2 are not applied. The other two mode 1 constraints (on Ω_1 and L_2) are applied in mode 3, but T_2 , A_2 , g_2 , x_2 , and y_2 are free. *This treatment is changed from the 1998 version* (see §21.3 on page 63). The stars can be overcontact in mode 3, yet have much different surface brightnesses. In structural language, they can be in geometrical contact without being in thermal contact.

Mode 4: This mode is for semi-detached binaries with star 1 accurately filling its limiting lobe, which is the classical Roche lobe for synchronous rotation and a circular orbit, but differs from the classical Roche lobe for non-synchronous and eccentric cases. The applied constraints are that Ω_1 has the lobe filling value and that L_2 is coupled to the temperatures (unless IPB=1).

Mode 5: The same as mode 4, except that it is star 2 that fills its limiting lobe. This is the usual mode for Algol-type binaries.

Mode 6: This mode is for double contact binaries, in which both stars accurately fill their limiting lobes (Wilson 1979). Astrophysically this makes sense only if at least one of the stars rotates non-synchronously. Mode 6 applies the surface potential constraints of modes 4 and 5 together.

Note on lobe filling: The program operates under a general definition of a limiting lobe that applies for non-synchronous as well as synchronous rotation and for eccentric as well as circular orbits. The classical Roche lobe is a special case that obtains for the synchronous, circular orbit case. In the general definition, *a limiting lobe is an equipotential for which the effective gravity is*

³Naturally the $x_2 = x_1$ and $y_2 = y_1$ conditions are applied only for fixed x, y coefficients, not for actively computed x, y 's.

zero on the line of star centers at periastron (Wilson 1979). In a strict sense, equipotentials do not exist in the eccentric orbit case, but in essence the concept of equipotentials still is useful in most realistic situations (Avni 1976; Wilson 1979).

6. Luminosity vs. Light (Flux)

It is essential to emphasize the very important, yet commonly ignored, distinction between “light” or “flux” on the one hand, and “luminosity” on the other. As simple as this point is, it has led to remarkable confusion in the literature. The problem has its roots in models of past decades that could have made the distinction but did not, and it has carried over into the general binary star literature of today. A luminosity as discussed here is characterized by a bandpass and bandpass properties (sometimes written “band” for brevity). Where it is necessary to refer to a bolometric luminosity, the qualifier “bolometric” can specifically be attached. In some papers, unqualified luminosity means bolometric luminosity, but in this document unqualified luminosity means bandpass luminosity.

A discussion of this point is in the description of subroutine **LUM** in Wilson (1993). The main thing to remember is that luminosity is a global quantity, represented by a single number for each star and band. It does not depend on aspect (neither on inclination nor on phase). Some authors write of “normalizing luminosities by the sum of the luminosities outside eclipse.” However, luminosity has nothing to do with eclipses or their absence, nor with any other aspect-dependent variation. Flux (or, in more relaxed terminology, “light”) is a directly observable (aspect-dependent) quantity, while luminosity is a model parameter that cannot be observed directly but must be inferred as part of a photometric solution. The luminosities, L_1 and L_2 , still can be normalized by their sum (a common and reasonable practice), but two points need to be kept in mind. The first is that the normalization must be made only after completion of a solution, not during its progress. The reason is that the luminosities are the main handles for scaling of (output) flux. As such they need to float freely so that the model is able to match the observed fluxes, which often are in an arbitrary unit, since the measurements are commonly made with respect to the flux from a comparison star. The second point is that L_1 and L_2 are not the same kind of quantity as third light (a flux), are not to be compared directly with third light, and are not in the same units. This is important because many papers have listed values of third light with no indication of what the number means. Not only is the written ℓ_3 thereby rendered useless, but strictly speaking the entire solution is made meaningless because ℓ_3 usually is significantly correlated with other parameters. An estimate of third luminosity, based on third light, involves an assumption about the direction dependence of the third source’s radiation. If that radiation is emitted isotropically, then third luminosity, L_3 , will be 4π times ℓ_3 , which is per $1/4\pi$ of the area of a sphere centered on the binary (“a steradian’s worth of area”). That L_3 value is to be compared with the un-normalized L_1 and L_2 . A meaningful way to specify the unit of a published ℓ_3 is to express it in the light (relative flux) of the multiple star system at a definite phase. For example, suppose the solution output light is 1.0500 at reference

phase 0.2500 for a particular light curve. Then divide ℓ_3 and its error estimate by 1.0500, list those numbers, not the direct ℓ_3 program output, and tell the reference phase in a footnote to the table of results.

7. Circumstellar Light-attenuating Regions

Light curves of some interacting binaries show effects of attenuation of star light by circumstellar matter. Most such matter is gaseous, although dust attenuation may be significant in a few cases. The attenuation may be due to Thomson scattering, Rayleigh scattering, another scattering process, or true absorption. Since circumstellar matter follows dynamical trajectories, one might expect there to be little interest in attenuating regions that are fixed in a coordinate frame that rotates with the binary, but such is not entirely the case. A few binaries (e.g. RZ Sct, AX Mon) have approximately stationary loci of circumstellar gas that distort their light curves. The loci may be stream-stream, stream-disk, or stream-wind interaction regions. Efforts to represent such light curve distortions via bright or dark star spots sometimes can rule out a spot explanation and point to essentially fixed attenuation regions (hereafter “clouds,” for brevity). The model includes n spherical semi-transparent clouds specified by their locations (x, y, z) [in a rectangular frame that co-rotates with the stars], cloud radius ($r_{\text{cloud}} = R_{\text{cloud}}/a$), density (ρ), electron density (n_e), and mean molecular weight per free electron (μ_e). The x coordinate is zero at the center of star 1 and increases toward star 2. The x, y, z system is right handed and serves for the entire binary system. The part of the line of sight that passes through the various clouds is computed individually for the lines of sight to all surface points and individually for all clouds. Regions of variable density can be made by nesting individual clouds. Regions of non-spherical shape can be approximated by overlapping spherical clouds. Each cloud is allowed its own attenuation law, whose general form is

$$\frac{d\tau}{ds} = \sigma_e n_e + (\kappa_\lambda + \kappa_{sb}) \rho,$$

where τ is optical thickness, σ_e is the Thomson scattering cross section per electron, s is distance along the line of sight (in cm), κ_λ is a wavelength-dependent opacity, and κ_{sb} is an additional opacity for a specific band (κ 's are in cm^2/g). The κ_{sb} term might represent, for example, opacity due to absorption lines averaged over a particular band. The κ_λ term is

$$\kappa_\lambda = \kappa_0 \lambda^\alpha,$$

where κ_0 and α are input quantities. Each cloud has its individual κ_0 , α , and κ_{sb} . However, to make it easy to change the κ_{sb} 's of all clouds together, the κ_{sb} 's are not entered directly as individual numbers. Instead one enters an overall κ_{sb} and the fraction of it that applies for each cloud. Thus

$$\kappa_{sb} = f_i \kappa_{sb0},$$

where all the f_i can be unity if κ_{sb} is to be the same for all clouds, or non-unity if κ_{sb} is to differ from cloud to cloud. The program figures absolute lengths from the system geometry, including the orbital semi-major axis.

At present the clouds only attenuate starlight that passes through them, but they may be made to scatter starlight toward the observer in a future program version.

8. Spectral Line Profiles

Profiles of absorption and emission lines are generated for **MPAGE=3**. The profiles are for rotation only, although other broadening mechanisms may be added later. Blending is incorporated, including blending of mixed absorption and emission lines. Lines can originate either from an entire star or from designated sub-areas of the surface, as explained below. Only **LC** (not **DC**) computes line profiles. Spectra are formed by binning, with the spectra of the two stars formed separately. The user can add them (weighted by observable flux) if spectra of the binary are needed. Computations for each star are characterized by four quantities related to spectral and accuracy characteristics (**BINWM1**, **SC1**, **SL1**, **NF1** for star 1 and **BINWM2**, **SC2**, **SL2**, **NF2** for star 2) that have the following meanings:

BINWM1(2): The bin width in microns. Too small a bin width gives noisy profiles. Too large a bin width gives insufficient spectral resolution.

SC1(2): The continuum scale (continuum flux at the reference wavelength). The unit is decided by the user.

SL1(2): The continuum slope in flux units per micron.

NF1(2): Grid fineness for micro-integration on each surface element. **NF1(2)=1** means that there is no micro-integration. **NF1(2)= n** breaks each surface element into n^2 pieces, each with its own radial velocity, thus improving integration accuracy.

With **MPAGE=3**, an **LC** input line with the above quantities is required for each star, even if one of the stars is assigned no spectral lines (see sample input file **lcin.dat3**). **DC** program input does not have the analogous data lines.

Following the star-dependent input are data for individual spectral lines in two sets (for star 1, then star 2). The quantities are (FORTRAN names):

WLL1(2): The line rest wavelength in microns for a line of star 1 (or 2).

EWID1(2): The line equivalent width for a line of star 1 (or 2), in microns—the traditional measure of line strength. Absorption and emission lines both have positive equivalent width by program convention. Whether a line is in absorption or emission is controlled by parameters **DEPTH1** and **DEPTH2** (see next).

DEPTH1(2): Rectangular line depth for a line of star 1 (or 2). Line profiles are formed by binning of Doppler shifted elements that have rectangular profiles, each with a depth and a width. The user supplies the depth and the program then calculates the rectangular line width needed to reproduce

the specified equivalent width. The depth is relative to a unit continuum, so 0.80000 means that 80 percent of the continuum flux is missing within the rectangular profile element, or that the residual flux is 20 percent of the continuum. Negative depths correspond to emission lines, so -0.50000 means 50 percent above the continuum. Depths must be less than 1.0000 (i.e. an absorption line cannot go to zero flux or below), but can be less than -1.0000 (an emission line can go arbitrarily high).

KKS: This integer specifies a surface region associated with a given spectral line. If **KKS**=0, the line is not specific to a location but applies to the entire star. If **KKS**=1, then the line applies only to the first spot on that star; if **KKS**=2 it applies only to the second spot, and so on. Naturally the star must have spots for this scheme to work, but the spots need not be hot or cool spots—they can have temperature factors of unity. Negative **KKS** specifies avoidance of regions. Thus **KKS**= -4 means that the spectral line applies everywhere on the star except within spot 4. If you find this confusing, just set **KKS**=0 and the line applies in the old simple way—everywhere on the star.

If computed absorption lines have flat bottoms, you probably used too small a depth. What happens is that the computed line is a superposition of many elemental bar-like lines, so if the bars are not deep, then they have to be wide in order to have the specified equivalent width. So essentially the problem is one of **DEPTH** being too small for a given equivalent width. The “flat-bottom” problem may be especially troublesome for slowly rotating stars. The converse problem is where the equivalent width is very small and **DEPTH** is large (like 0.99). The elemental bars are then very narrow (needle-like). This situation can produce raggedy lines, especially for fast rotating stars. The remedy is to use a smaller **DEPTH**.

9. Time/Phase Smearing

Some datasets (notably the Kepler mission’s 30-minute ‘long cadence’ sets) are integrations over intervals that may not be sufficiently short to fully resolve variations of astrophysical interest. The variations may be resolved but smoothed so as not to correspond to the un-smoothed behavior. Sharp features such as eclipses can thereby be widened and reduced in amplitude, perhaps even enough to lose their resemblance to the original features. **LC** and **DC** deal with this problem by integrating the model output over a phase window specified by parameter **DELPH** (Wilson 2012b). The program uses the binary period to convert to the corresponding time window. The integration is by Gaussian quadrature with accuracy specified by the number of abscissae (**NGA**), which can be as few as 2 or as many as 10 in the present program. To bypass smearing, enter **NGA**=1. Although Gaussian quadrature is very accurate for a given number of abscissae, there is a cost in computing time, essentially by a factor of **NGA**. For most applications, **NGA**=3 should easily suffice, and **NGA**=2 may be OK.

10. Model Parameters

These are the same for the **LC** and **DC** programs, although not all of the parameters are among the 50 that can be adjusted by the present version of **DC**. The actual number of parameters subject to adjustment is greater than 50 for two reasons. First, although **DC** can adjust the parameters of at most two star spots in any one run, successive runs can adjust the parameters of other spots (viz. definitions of KSPA, KSPB, NSPA and NSPB in §12.3 on page 34). Second, curve-dependent parameters (limb darkening coefficients, relative bandpass luminosities, third light) have n values for n simultaneously fitted light curves. For example, specifying the adjustment of ℓ_3 in a simultaneous solution of four light curves will produce four ℓ_3 corrections, one for each curve. The word “curve” is used here in the sense of “light curve” or “radial velocity curve,” etc. Parameters are identified below according to their FORTRAN names.

10.1. Curve-independent parameters that cannot be adjusted by DC

THE (ϕ_e): The semi-duration of primary eclipse in phase units (i.e. range 0 to 1 for a cycle). This parameter is applied only in mode -1 and usually relates to binaries in which star 1 has negligible size compared to star 2. The idea is to fix ϕ_e according to X-ray eclipses of a neutron star, black hole, or white dwarf, and require the overall solution to be consistent with that value. Although mode -1 logic was originally intended for X-ray eclipses of compact objects, it also will work for optical eclipses of any star for which R_1/R_2 is extremely small (if the eclipse duration is somehow known independently of the light curve), such as for a main sequence star with a red giant companion. The orbit can be eccentric and rotation can be non-synchronous. Parameter ϕ_e is ignored in all other operation modes. See the explanation of mode -1 in §5 on page 15. The mathematics for the general case with any eccentricity and any rotation is in Wilson (1979).

XBOL1, XBOL2 (x_{bol1}, x_{bol2}): The coefficients of $\cos \gamma$ in the bolometric limb darkening law. They are used only in computation of “detailed” reflection (Wilson 1990), with MREF=2. See y_{bol1}, y_{bol2} below for the complete law and explanation.

YBOL1, YBOL2 (y_{bol1}, y_{bol2}): If control integer LD1(2)=+2 or -2 , these are the coefficients of the $\cos \gamma \ln(\cos \gamma)$ term in the bolometric logarithmic limb darkening law. The complete logarithmic law is

$$\frac{I}{I_0} = 1 - x + x \cos \gamma - y \cos \gamma \ln(\cos \gamma),$$

which was advocated by KlingleSmith & Sobieski (1970). If LD1(2)=+3 or -3 , they are the coefficients of the bolometric square root law. The complete square root law (Diaz-Cordovés & Giménez 1992) is

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

Coefficients for all these darkening laws have been tabulated by Van Hamme (1993).

XCL, YCL, ZCL: Rectangular coordinates of the centers of spherical circumstellar clouds (see §7 on page 18).

RCL: Radii of individual circumstellar spherical attenuating regions in unit of a .

EDENS (n_e): Electron density in cm^{-3} for individual attenuating clouds. For a given cloud, n_e is constant, although clouds can be nested or overlapped.

XMUE (μ_e): Mean molecular weight per free electron in atomic mass units for individual attenuating clouds, and constant throughout a given cloud.

ENCL (α): Exponent in the wavelength-dependent term of the attenuation law for individual attenuating clouds (see §7 on pages 18–19). The program internally computes density, ρ , from n_e and μ_e .

10.2. Curve-dependent parameters that cannot be adjusted by DC

WLA: Array of observational wavelengths in microns of light/velocity curves. Wavelengths need to be entered for velocity curves, although they have little effect on the output, and any wavelength somewhere near the spectral region of interest should be adequate. Wavelengths are no longer used for light curves, which are based on integrated bandpass radiation since the 2003 revision (see §15 on pages 43–49). Wavelengths are still entered for use as reference wavelengths for line profiles and for opacity computations in circumstellar attenuation.

Y1A, Y2A (y_1, y_2): These are the bandpass-specific limb darkening coefficients *in the non-linear terms*. The laws have the same forms as for bolometric limb darkening (see page 21). In the case of definite (i.e. fixed, not locally computed) limb darkening, there is one value for each of these coefficients for each light or velocity curve or set of line profiles. **LC** uses both wavelength and bandpass input for line profile computation (wavelength for basic profiles, bandpass for weighting by flux). Integer parameter LD1(2)=+1 or −1 applies for the linear cosine law (in which case y_1 and y_2 are assumed to be zero), LD1(2)=+2 or −2 for the logarithmic law, or LD1(2)=+3 and −3 for the square root law. Negative LD1(2) is for internally computed x, y ’s and positive LD1(2) for fixed x, y ’s. LD1(2) should not be negative for star temperatures outside the range of Kurucz atmosphere temperatures (listed in Table 1 on page 44).

10.3. Curve-independent parameters that can be adjusted by DC

XLAT: The “latitude” of a star spot center, measured from 0 at the “north” (+ z) pole to π radians at the “south” pole. XLAT’s are subscripted by star (1 or 2) and by spot number on a star (viz. explanation of KSPA, NSPA, KSPB and NSPB in §12.3 on page 34).

XLONG: The longitude of a star spot center at the ephemeris reference time (HJD0), measured

counter-clockwise (as viewed from above the $+z$ axis) from the line of star centers (0 to 2π radians). XLONG is subscripted in the same way as XLAT.

RADSP: The angular radius of a star spot, in radians. The angle is subtended by the spot radius at the center of the star. RADSP is subscripted in the same way as XLAT. Note that spot radius and spot temperature are usually strongly correlated so typical solutions for both parameters may be unlikely to succeed. A reasonable strategy might be to step one parameter and solve for the other, then publish a small table of mutually consistent radius/temperature combinations. Possibly the radius-temperature correlation might be acceptably small for very well separated bands such as B and K .

TEMSP: The “temperature factor” of a spot that specifies the ratio of local spot temperature to local temperature that would obtain without the spot. A TEMSP larger or smaller than unity corresponds to the spot being hotter or cooler than the un-spotted surface, respectively. TEMSP is constant for a given spot, but local temperature will vary over a spot if the underlying un-spotted temperature varies. TEMSP is subscripted in the same way as XLAT. See the comments on correlation between spot radius and spot temperature just above.

E (e): Binary orbital eccentricity.

PERR0 (ω_0): Initial argument of periastron **for star 1** in radians. The argument of periastron for star 2 differs by π radians. PERR0 is ω at the ephemeris reference time, t_0 . The program is written so that, as the argument of periastron changes, both conjunctions migrate in phase in the same way as in a real binary (neither conjunction is locked at zero phase).

DPERDT ($\frac{d\omega}{dt}$): The first time derivative of ω . DPERDT can be adjusted only if JDPHS=1 and observation times (rather than phases) are entered. The instantaneous argument of periastron is $\omega = \omega_0 + \frac{d\omega}{dt}(t - t_0)$. The program does not consider any more complicated variations of ω . The unit is radians per adopted time unit (usually mean solar day).

A (a): The length of the semi-major axis in solar radii (6.96000×10^5 km) of the relative orbital ellipse. It is the sum of the two absolute semi-major axes, so $a = a_1 + a_2$.

F1, F2 (F_1, F_2): The ratio of the (constant) axial rotation rate to the mean orbital rate for stars 1 and 2, respectively. The angular rotation is assumed to be uniform (not latitude dependent). Value unity represents synchronous rotation in a circular orbit. In eccentric cases it is expected that rotation will tend to synchronize to a value near the periastron angular rate because of the strong dependence of the tide raising force on distance. Periastron-synchronized F is given by

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

The F ’s affect star figures, surface brightnesses (through the gravity effect), limiting lobe sizes, and thus both light curves and radial velocities.

VGA (V_γ): The radial velocity of the (possibly multiple-system) barycenter in the unit VUNIT km/s.

V_γ is assumed constant, and the EB center of mass radial velocity follows from motion about the barycenter if there is a third body.

PSHIFT (ϕ_0): A constant shift applied to computed phases. Usually one enters 0.0000 for ϕ_0 , but it may be convenient to shift the phases. For example, a phase shift of about half a cycle can effectively interchange the star labels (1 vs. 2) without altering the observational data. The main purpose of ϕ_0 is to allow the **DC** program to adjust for a zero point error in the ephemeris used to compute the phases. The unit is the orbital period. One should not adjust both **PSHIFT** and **HJD0** because they will be perfectly correlated.

XINCL (i): The binary orbital inclination to the plane of the sky, in degrees. If the inclination is in the range 0 to 90° , the binary orbits counter-clockwise as projected onto the plane of the sky, while above 90° it orbits clockwise, according to the program's coordinate conventions. Those conventions were different prior to the revision of 1992.

GR1, GR2 (g_1, g_2): The exponents in the bolometric gravity brightening (a.k.a. darkening) law for stars 1 and 2, respectively. A value of 1.000 means that bolometric flux is proportional to local effective gravity, while 0.000 means that it is constant over the surface (ignoring other effects such as spots, reflection heating, etc.). The g 's are expected to be unity for radiative envelopes, while they should be smaller for convective envelopes, perhaps about 0.3. Some other programs use a gravity brightening exponent expressed in terms of effective temperature, for which the usual symbol is β . The quantities are related by $g = 4\beta$.

TAVH, TAVC (T_1, T_2): The mean surface effective temperatures of stars 1 and 2, respectively, not including re-radiation (reflection) or spots. The mean is weighted by the local bolometric flux. The program accepts T_1 and T_2 as model parameters and converts to local surface temperatures for internal computations. The conversion between mean and polar temperatures is made via Eqn. 8 of Wilson (1979), and the local surface temperatures are then computed from the polar temperatures and the gravity brightening law. The unit is 10,000 K.

ALB1, ALB2 (A_1, A_2): The bolometric albedos for reflection heating and re-radiation on stars 1 and 2, respectively. The bolometric albedo is the local ratio of re-radiated bolometric energy to received bolometric energy. It is assumed to be constant for each star. The expected value for radiative envelopes is 1.0, while for convective envelopes it should be perhaps 0.5, although observations sometimes indicate values between 0.5 and 1.0.

PHSV, PCSV (Ω_1, Ω_2): These are the "potentials" for stars 1 and 2, respectively, that originally were defined by Kopal (1954) for the synchronous, circular orbit case. They would be actual potentials, except that a term was deleted in Kopal's convention. The deleted term depends on mass ratio but not on position, so Ω gradients are equivalent to potential gradients. Fixed Ω specifies a constant potential energy (gravitational plus rotational) over the surface of each star. A generalized defining equation (Eqn. 1 of Wilson 1979), based on contributions by Plavec (1958), Limber (1963), and Avni (1976), allows the Ω 's to serve also for non-synchronous rotation and eccentric orbits. Together with the mass ratio, rotation rate, orbital eccentricity, argument of periastron, and phase, the Ω 's

specify the size, figure, surface gravity, and certain other geometric properties of the stars. Special values of the Ω 's correspond to exact filling of limiting lobes.

RM (q): The mass ratio of stars 2 and 1, m_2/m_1 .

HJD0 (t_0): This is the zero point of the orbital ephemeris. Usually one uses Heliocentric Julian Date, although that is only a convention and any consistent system of time can be used.

PZERO (P_0): The binary orbit period at the reference time t_0 , ordinarily in mean solar days. P_0 affects computed radial velocity amplitudes and is used to compute phase from time (if JDPHS=1) and time from phase (if JDPHS=2). P_0 can be adjusted only if JDPHS=1 and observation times (rather than phases) are entered.

DPDT (dP/dt): The first time derivative of the orbital period. Second and higher derivatives are not used in the present program. DPDT can be adjusted only if JDPHS=1 and observation times (rather than phases) are entered. This quantity is dimensionless.

A3B (a_{3b}): The length of the semi-major axis of the relative third body orbit in solar radii.

P3B (P_{3b}): The third body orbit period in mean solar days.

XINC3B (i_{3b}): The third body orbital inclination to the plane of the sky, in degrees. This parameter should *not* be adjusted if a_{3b} is adjusted since a_{3b} and i_{3b} are perfectly correlated.

E3B (e_{3b}): Third body orbital eccentricity.

PERR3B (ω_{3b}): Argument of periastron of third body orbit, in radians.

TC3B (t_{c3b}): Time of superior conjunction of the EB center of mass with the third body, in HJD or whatever time system has been adopted.

DPCLOG: Logarithm (base 10) of distance (d) in parsecs.

AEXTINC: Interstellar extinction (A_{band} in magnitude) in the designated photometric band. AEXTINC refers to a definite photometric band (the band designated by control integer LINKEXT), so it has only one value and it is not band-dependent.

10.4. Curve-dependent parameters that can be adjusted by DC

HLA, CLA (L_1, L_2): Bandpass luminosities for stars 1 and 2, respectively. There has been some confusion about the units of luminosity and of light, which is partly due to a tradition in the binary star field of normalizing luminosity and light separately, and failing to recognize that they are fundamentally distinct (although connected) quantities (viz. §6 on page 17). The program luminosity unit is effectively user-supplied, and determines the unit of output light (i.e. ℓ_1 or ℓ_2 bandpass flux). The output flux will integrate to the luminosity over a sphere at any large distance, centered on the binary system. The computed fluxes will be in the unit $1/4\pi$ luminosity

units/(steradian’s worth of area). Now this may not sound like correct flux units—with “steradian” in there it sounds like intensity. However, that last item, “steradian’s worth of area” is indeed an area, not a solid angle. So to be formally correct, replace “steradian’s worth of area” with “ $d^2 \text{ cm}^2$,” where d is the assumed binary–observer distance in cm. Pictorially, imagine the observer’s detecting instrument covering 1 steradian (a little on the big side, but one can always re-scale to more practical units). To keep things simple, imagine that the star radiates isotropically and that we enter a luminosity of 4π , in our chosen unit, for one of the stars. The program will produce an output flux of 1.000 for that star for all phases and inclinations (outside eclipse), and since that flux refers to 1 steradian, the 4π steradians surrounding the star would have 4π units, which is just what we entered as the luminosity. If we specify the luminosity unit to be, say, $1 \times 10^{33} \text{ erg/s/micron}$, then the luminosity is $4\pi \times 10^{33} \text{ erg/s/micron}$, and the flux (light) is $1 \times 10^{33} \text{ erg/s/micron}/d^2 \text{ cm}^2$ (constant in this special case).

X1A, X2A (x_1, x_2): These are the wavelength-specific limb darkening coefficients in the linear terms. The laws have the same forms as for bolometric limb darkening (see page 21). Parameters x_1 and x_2 are adjustable only if the input darkening coefficients are specified as being definite (i.e. not to be re-computed), which is decided by input quantities **LD1** and **LD2** being positive. There is one x_1 and one x_2 value for each light or velocity curve or set of line profiles. For negative **LD1** or **LD2**, local surface computation of limb darkening coefficients as functions of T_{eff} , $\log g$, and $[M/H]$ is done by subroutine **LIMDARK**, which interpolates tabulated band-integrated coefficients calculated with the method of Van Hamme (1993). In this case, limb darkening tables need to be accessible to the program, see further §15 on page 43. Flux-weighted mean x, y ’s for both stars are written by **LC** if **MPAGE**=1.

EL3A (ℓ_3): Third light. There is one value for each light curve, but of course no value for a radial velocity curve. The unit should be the total system light at a specified phase. For example, suppose ℓ_3 (program input-output value) for some particular light curve is 0.0500, the specified phase is chosen to be 0.2500, and the total system light produced by **LC** at phase 0.2500 is 1.0400. Then the number to be published for the ℓ_3 of that curve would be 0.0500 divided by 1.0400, or 0.0481. The standard error written by **DC** for ℓ_3 also should be divided by 1.0400 for publication. A footnote can be placed in the publication to tell the reference phase.

10.5. Ephemerides from Whole Curves, along with Apsidal Motion and Third Body Parameters

Ephemeris parameters T_0 , P_0 , and dP/dt can be found from whole light and velocity curves by **DC**, as an alternative to ephemerides from eclipse timings. % The simple but necessary mathematics for rigorous conversion from time to phase and phase to time is in §4 of Wilson (2005) along with a history of its use in **LC** and **DC**, and with citations to a number of application papers. **DC** also can extract apsidal motion and third body (3b) orbit parameters (2007 version and later). The 3b parameters are numbered 30 to 35 in §11 on page 29. Observational phenomena that affect 3b

parameter results are light-time and radial velocity shifts (not third light, ℓ_3). Analysis of light and velocity shifts and third light effects may be done together, with the two kinds of results being essentially independent. Relevant history, strategies, and logic of the 3b development and a similar one by Hadrava (2004) are in Van Hamme & Wilson (2007), as are the forms of 3b derivatives required by **DC**. Rather than repeat here the experience described in Van Hamme & Wilson (2007), we recommend that potential users of **DC**'s 3b capability read that paper. A solution that includes 3b parameters is basically the same as other solutions, except that (1) the increased number of parameters may make application of the MMS imperative (see §13.1 on page 39), and (2), a satisfactory starting estimate for P_{3b} may require separate use of power spectral analysis. Simultaneous processing of light and velocity curve data is an important advantage of **DC**'s 3b facility, as the two types taken together reduce the severity of the well known data gap problem for period searches. However, even with that advantage, initial application of power spectral analysis often will be needed to sift the correct period from apparent periods that are only artifacts of the timewise observing gaps. Note the example of VV Orionis, where recent papers find no evidence for a supposedly well established 3b period (Terrell, Munari, & Siviero 2007; Van Hamme & Wilson 2007).

10.6. Unified Ephemeris Solutions

A thorough account of unified ephemeris solutions is in Wilson & Van Hamme (2014). The remainder of this subsection is excerpted from that paper.

Ephemeris solutions for time-related parameters directly from light and RV curves lead naturally into use of those information sources *together* with eclipse times. If the original light curve data from which timings were measured have survived, then they can simply be entered into **DC** and the published timings ignored. However that will seldom be the case.

Time-related binary system characteristics such as orbital period, period rate of change, apsidal motion, and variable light-time delay due to a third body, have historically been measured in two ways that can be mutually complementary. The older way is via eclipse timings, while ephemerides by simultaneous whole light and velocity curve analysis have appeared recently. Each has its advantages, for example eclipse timings typically cover relatively long time spans while whole curves often have densely packed data within specific intervals and allow access to systemic properties that carry additional timing information. Synthesis of the two information sources can be realized in a one-step process that combines several data types, with automated weighting based on their standard deviations. *Simultaneous* light-velocity-timing solutions treat parameters of apsidal motion and the light-time effect coherently with those of period and period change, allow the phenomena to interact iteratively, and produce parameter standard errors based on quantity and precision of the curves and timings. Logic and mathematics of unified ephemeris solutions that coherently utilize light curves, velocity curves, and eclipse timings are in Wilson & Van Hamme (2014). All three data types together, each separately, or any subset (radial velocities with light curves; radial velocities

with timings; light curves with timings) can be processed, with automated precision-based weights. **DC** can determine which star is eclipsed, thus detecting any incorrect initial eclipse type assignments. A reasonably correct starting ephemeris (perhaps including dP/dt and a light-time term) is needed, otherwise an anomaly will become obvious in the run of residuals as time extends well beyond the reference time. A trend to more positive or negative residuals will appear until there is a large abrupt jump, which is the signature of an ephemeris with insufficient validity range. Some timing residuals, especially ones far from the reference time, can then be off by orders of magnitude, and perhaps even correspond to the wrong eclipse (type I when star 1 is at superior conjunction vs. type II when star 1 is at inferior conjunction). A solution could get into permanent trouble if there were no escape process. The remedy is to have **DC** ignore minima when the type computed from the ephemeris disagrees with the designated type. The revised ephemeris from the next iteration should have a greater applicable range, perhaps including all observed minima. This “clip and solve” process is repeated until all observed timings have small residuals, and has been automated. At succeeding iterations the remaining timings improve the initial ephemeris enough so that fewer timings need be ignored in the following iteration, then still fewer in the next, until all the timings are utilized. Occasional disagreements due to wrong I vs. II *assignments* (as opposed to an inaccurate ephemeris) will be obvious from residuals that continue to be off by much of a cycle.

11. Parameter Order

The **DC** program now has 60 parameter channels, 50 of which are assigned to specific parameters while the other 10 are for future expansion. The 60 **DC** parameter channels are assigned as follows:

- (1) - Spot A latitude in radians
- (2) - Spot A longitude in radians
- (3) - Spot A angular radius in radians
- (4) - Spot A dimensionless temperature factor
- (5) - Spot B latitude in radians
- (6) - Spot B longitude in radians
- (7) - Spot B angular radius in radians
- (8) - Spot B dimensionless temperature factor
- (9) - Orbital semi-major axis in solar radii, $a = a_1 + a_2$
- (10) - Orbital eccentricity, e
- (11) - Argument of periastron in radians, ω_0 , at the reference epoch

- (12) - Dimensionless rotation parameter for star 1, F_1
- (13) - Dimensionless rotation parameter for star 2, F_2
- (14) - Phase shift = phase of primary conjunction (for $\omega = \pi/2$), ϕ_0
- (15) - System center of mass radial velocity (systemic velocity), V_γ
- (16) - Orbital inclination in degrees, i
- (17) - Gravity law exponent for star 1, g_1
- (18) - Gravity law exponent for star 2, g_2
- (19) - Mean surface temperature of star 1 in 10,000 K, T_1
- (20) - Mean surface temperature of star 2 in 10,000 K, T_2
- (21) - Bolometric albedo of star 1 (dimensionless), A_1
- (22) - Bolometric albedo of star 2 (dimensionless), A_2
- (23) - Modified dimensionless surface potential of star 1, Ω_1
- (24) - Modified dimensionless surface potential of star 2, Ω_2
- (25) - Mass ratio, $q = m_2/m_1$
- (26) - Reference epoch in ephemeris (conventionally in Heliocentric JD), t_0
- (27) - Orbital period at reference epoch, P_0
- (28) - First time derivative of the orbital period, dP/dt
- (29) - First time derivative of the argument of periastron, $d\omega/dt$
- (30) - Semi-major axis length of third body orbit in solar radii, a_{3b}
- (31) - Period of third body orbit in days, P_{3b}
- (32) - Inclination of third body orbit to the plane of the sky in degrees, i_{3b}
- (33) - Eccentricity of third body orbit, e_{3b}
- (34) - Argument of periastron of third body orbit in radians, ω_{3b}
- (35) - Time (conventionally Heliocentric Julian Date) of superior conjunction of the EB center of mass with respect to the third body, $T_{c_{3b}}$
- (36 to 40) - Not used, reserved for future expansion
- (41) - log (base 10) of distance in parsecs
- (42) - Interstellar extinction in designated photometric band, in magnitude

- (43) - Onset time of spot A
- (44) - Start time of constant maximum size for spot A
- (45) - End time of constant maximum size for spot A
- (46) - Disappearance time of spot A
- (47) - Onset time of spot B
- (48) - Start time of constant maximum size for spot B
- (49) - End time of constant maximum size for spot B
- (50) - Disappearance time of spot B
- (51 to 55) - Not used, reserved for future expansion
- (56) - Bandpass luminosities of star 1, L_1
- (57) - Bandpass luminosities of star 2, L_2
- (58) - Bandpass linear limb darkening coefficient for star 1, x_1
- (59) - Bandpass linear limb darkening coefficient for star 2, x_2
- (60) - Bandpass third light, ℓ_3

12. Control Integers, Units, Scaling Factors, Special Quantities

These numbers control program operation and are set according to the aims of the user. They are given here by their FORTRAN names.

12.1. Those Common to LC and DC

JDPHS: This is 1 if the independent variable is time and 2 if it is phase. Setting JDPHS=1 in **LC** causes time (ordinarily Heliocentric Julian Date) to be stepped from HJDST to HJDSP in uniform intervals of length HJDIN (see further §12.2). Phases for those times are computed according to the supplied ephemeris, including the effect of dP/dt . Setting JDPHS=2 in **LC** causes phase to be stepped from PHSTRT to PHSTOP in uniform intervals of length PHIN (described in §12.2). Times for those phases are computed according to the supplied ephemeris, again including the effect of dP/dt . Regardless of whether JDPHS is 1 or 2, mutually consistent time and phase are listed in the output (columns 1 and 2, respectively). At present, the ephemeris includes only the initial epoch (t_0), the period at that epoch (P_0), and the first time derivative of the period (dP/dt). Third body light time also affects the connection between time and phase.

MODE: This integer can be $-1, 0, 1, 2, 3, 4, 5$, or 6 according to the constraints or lack of constraints to be applied. The operation modes are described in Wilson (1993) and in §5 on pages 14–17.

IPB: Assign $IPB=0$ for normal **MODE** $1, 2, 3, 4, 5$, or 6 operation in which star 2’s luminosity (L_2) is to be computed from temperatures T_1 and T_2 , the luminosity of star 1, and the radiation laws (as well as other information known by the program about system geometry, etc.). If you want to set L_2 independently (perhaps because you have no trust in the radiation laws in a particular situation), set $IPB=1$ and the program will use the input L_2 value. Modes 0 and -1 always accept the input L_2 , so they operate as if $IPB=1$. See Van Hamme & Wilson (1986) for ideas on the use of **IPB** in practical situations.

IFAT1, IFAT2: These control whether a blackbody or a stellar atmosphere formulation is used for local emission on stars 1 and 2, respectively. Set $IFAT1(2)=0$ or 1 for blackbody or atmosphere, respectively.

N1, N2: These are the grid size integers for stars 1 and 2, respectively. Each is the number of latitude rows per hemisphere. The number of surface elements in longitude scales with $N1(2)$ and scales approximately with the sine of the “latitude” coordinate, which runs from 0 at the “North” ($+z$) pole to π radians at the “South” ($-z$) pole.

VUNIT: This is the unit for radial velocity input and output, in km/sec. Usually it is a round number, such as 100 km/sec, of the order of the input velocities for **DC**.

MREF: The reflection effect can be handled either in detail (viz. Wilson 1990) or by the inverse square law, with corrections for penumbral and ellipsoidal effects. The latter method is much faster and is adequate for many realistic situations. Set $MREF=1$ for the simple treatment and $MREF=2$ for the detailed treatment. It is not advisable to use the detailed treatment for eccentric orbit cases because the required computing time will be almost prohibitively long. Cases for which the detailed treatment is especially recommended include super-synchronous rotators and overcontact binaries. However, computing time for detailed reflection scales with grid integer to the 4th power (actually with $N1^2 N2^2$), so you might keep machine time within reason, even with $MREF=2$ in eccentric systems, by use of coarse grids.

NREF: If detailed reflection is selected ($MREF=2$), then **NREF** specifies the number of reflections in a multiple reflection effect. Set $NREF=1$ for 1 reflection from each star, $NREF=2$ for 2 reflections, etc. More reflections use more computing time. If $MREF=1$, the value of **NREF** is irrelevant.

IFSMV1, IFSMV2: These integers tell whether spots on stars 1 and 2, respectively, are to drift in longitude. If $IFSMV1=0$, star 1’s spots remain at fixed longitudes, referenced to the line of star centers. This behavior is expected for hot spots due to an accretion stream unless stream dynamics vary. If $IFSMV1=1$, then star 1’s spots drift in longitude as time progresses at a rate set by parameter $Fspot1$, and similarly for star 2 and $Fspot2$. This behavior might be expected for magnetic spots. There is no motion in latitude in either case. Solutions for moving spots needs good starting estimates and attention to specifics.

ICOR1, ICOR2: These integers refer to proximity and eclipse effects on radial velocities for stars 1 and 2, respectively. Value 0 turns the effects off, 1 turns them on.

IF3B: This control integer tells the program if a kinematic third body is assumed in the model (IF3B=1) or not (IF3B=0).

LD1, LD2: These integers set the limb darkening laws for stars 1 and 2, respectively. LD1(2)= ± 1 for the linear cosine law, LD1(2)= ± 2 for a logarithmic law, and LD1(2)= ± 3 for a square root law. For positive LD1 or LD2, the x, y coefficients are fixed and equal to the input values. For negative LD's, subroutine **LIMDARK** ignores the input $[x, y]$ and computes x, y coefficients locally in terms of T_{eff} , $\log g$ and $[M/H]$. Mathematical expressions for the logarithmic and square-root laws are on page 21.

KSPEV: Controls whether spots age (grow and decay) in radius. Currently there is no aging in spot temperature. Set KSPEV=0 for no aging, KSPEV=1 for aging. Solutions for spot aging need good starting estimates for spot parameters and careful monitoring of solution progress.

KSPOT: Controls whether the old simple spot algorithm (KSPOT=1) or the much more precise Vector Fractional Area algorithm (Wilson 2012b) for KSPOT=2 is applied.

NOMAX: Setting NOMAX=1 eliminates the interval of constant size that otherwise exists at spot maximum (spot aging profiles then become triangular). Spot aging profiles are trapezoidal for NOMAX=0.

IFCGS: Control integer IFCGS=0 for a conventional solution in arbitrarily scaled flux and IFCGS=1 for an absolute solution with flux unit $\text{erg s}^{-1} \text{cm}^{-3}$.

NGA: Number of Gaussian abscissas in the Gaussian quadrature of time/phase smearing. With NGA=1 there is no phase smearing. **LC** and **DC** allow NGA to be as small as 2 and as large as 10. There is no radial velocity smearing as of now, although that may soon be added.

DELPH: The duration of a light curve observation (assumed to be the same for all observations) for phase smearing computations as a decimal fraction the orbit period.

CALIB: Flux calibration constant in $\text{erg s}^{-1} \text{cm}^{-3}$ for a star of magnitude 0.00. It is an array in **DC** and a single quantity in **LC**. Table 3 on page 53 lists bands that have CALIB values determined and published.

IBAND: Photometric band identification integer.

12.2. Those for LC only

HJDST: The time at which program **LC** is to start computing output points. HJDST and the next two quantities, HJDSP and HJDIN, are utilized only if JDPHS=1. They are ignored if JDPHS=2.

HJDSP: The time at which program **LC** is to stop computing output points.

HJDIN: The time increment for output points. HJDIN=0.001 will produce output points spaced by 0.001 days.

PHN: The phase of normalization, which is the phase at which the column of normalized light is normalized to the input value **FACTOR** and the magnitude column is caused to equal the magnitude zero point, whose name is **ZERO**.

PHSTRT, PHSTOP: The first and last phases at which output points are to be produced by program **LC**. PHSTOP should be larger than PHSTRT, but neither has to be in the range 0 to 1. For example, PHSTRT=−3.2000, PHSTOP=27.4422 is a valid phase range. PHSTRT, PHSTOP, and the next quantity, PHIN, are utilized only if JDPHS=2.

PHIN: The phase increment for output points. PHIN=0.020 will produce output points every 0.020 in phase, within the range PHSTRT to PHSTOP.

MPAGE: This integer is 1, 2, 3, 4, 5, or 6 according to whether the output is a light curve, radial velocity curves, spectral line profile(s), star radii, sky image coordinates, or conjunction times, respectively.

KTSTEP: This integer is 1, 2, 3, ... and is to step computed conjunction times by 1, 2, 3, ... whole orbit cycles. With KTSTEP=0, timing residuals (in the sense ‘observed minus computed’) rather than conjunction times are generated.

ZERO: This is the reference level for output magnitudes (the magnitude at phase PHN).

FACTOR: This is the scaling factor for the normalized light column. The number in that column will equal **FACTOR** at phase PHN.

12.3. Those for **DC** only

DEL’s: Most of the partial derivatives needed by **DC** must be computed numerically. The DEL’s are the parameter increments applied when approximating those derivatives by finite differences. Some derivatives can be computed analytically and require no DEL’s. That is why the number of DEL’s is smaller than the number of parameters. However, the numerical labels of the DEL’s match those of the parameters (e.g. DEL number 25 goes with parameter 25). The order in which the DEL’s appear in the input lines can be read from the output of **DC**, where they are labeled by parameter name, or from the **DC** input in Appendix B, where they are labeled by order of listing. Give some thought to reasonable values for the DEL’s. DEL’s that are too large will cause systematic errors in the derivatives, and DEL’s that are too small will cause excessive numerical noise.

KEP or KEEP: These mean the same thing generically. The difference in meaning in a programming sense is not something the user needs to be concerned with. The KEEP’s determine which parameters

are to be adjusted, and they have only two possible values, 0 to allow adjustment and 1 to keep a fixed value. There are 60 parameter channels, with numbers 1 to 35 and 41 to 50 assigned to band-independent parameters and 56 to 60 assigned to band-dependent parameters. Channels 36 to 40 and 51 to 55 are reserved for future expansion. Accordingly, there are 60 **KEEP**'s but only numbers 1 to 35, 41 to 50, and 56 to 60 are now meaningful. All **KEEP**'s are entered together on an input line, separated into blocks so as to be easy to count off. There is one such input line for the base set, early in the data stream, and a set of n such input lines at the end of the data stream (for the n subset solutions). The subsets must indeed be subsets, not supersets, since a given parameter cannot be adjusted in a subset unless it is adjusted in the base set. The order of appearance of the **KEEP**'s within the string of 1's and 0's is the same as that of the 60 parameter channels, which are listed in §11 (on pages 28–30) and also written with every run of **DC**.

IFDER: Write control integer **IFDER** is set to 1 for “yes, write” or to 0 for “do not write.” It decides whether the matrix of parameter derivatives and residuals, or observational equations (both the unweighted and the weighted versions), are written. Since the observational equations for the first iteration pertain to the *input* parameters (and they change only a little for subsequent iterations), they are written only for the base set.

IFLCIN: Set **IFLCIN**=1 for **DC** to write an input file for the **LC** program or **IFLCIN**=0 not to write one. The generated file is based on all RV curves, light curves, and eclipse timings in the **DC** solution. It is named `lcin.input_from_DC` and it overwrites any previously generated file of the same name. So copy `lcin.input_from_DC` to a file with another name if you want to save it. The file represents input to the last **DC** iteration of the last **DC** subset (or the base set if there are no subsets).

IFOC: Stands for ‘IF Observed-Computed.’ Causes output of two extra columns in **DC**'s unweighted observational equations – one column for the observations and the other for computed values at the same times or phases. This is just a convenience for making quick plots without need to run **LC**.

NITERS: The number of iterations for the base set or a subset (see §14 on page 42 for details).

XLAMDA (λ): The Marquardt multiplier (see §13.2 on page 39). **XLAMDA** is the next to last entry on the “**KEEP**” lines and is in D format.

VLR: The fraction of **DC**'s output corrections to be applied to the input parameters to form the output parameters. **VLR** values smaller than 1.000 may improve convergence. **VLR** should not be greater than 1.000 or negative. For more on **VLR**, see §13.3 on page 40.

KSPA, **NSPA**, **KSPB**, **NSPB**: Each of the two stars may have many spots, but **DC** can adjust the parameters of at most two spots in any one run. This remark does not imply a limit to the number of spots adjusted overall, since other spots can be adjusted in other runs. Think in terms of spot A and spot B, which may be on the same or different stars. **KSPA**, **NSPA**, **KSPB**, and **NSPB** are best understood as a set. Given that some spot parameters are to be adjusted (which is determined by

KEEP's 1 to 4 for spot A and KEEP's 5 to 8 for spot B), the four integers (KSPA, etc.) determine for which of the (perhaps) many spots on the two stars those parameters will be adjusted. KSPA=1 means that spot A is on star 1, and KSPA=2 means that it is on star 2. NSPA determines which spot on the thus determined star is spot A. If KSPA=2 and NSPA=3, then star 2's third spot is spot A. Star 2 might have some large number of spots, say 10, and spot A will be the third one. Of course, the program will do something crazy if you ask it to adjust the third spot on a star that has only two spots—we have not bothered to find out what. The order assigned to the spots is just the order in which their parameters are read in the input lines. Naturally, KSPB and NSPB apply the same way in identifying spot B. If you want to adjust, say four spots, you can use the MMS (see §13.1) and adjust two of them in one **DC** run, the other two in the next run, and so on.

IFVC1, IFVC2: These integers tell **DC** whether or not to expect a radial velocity curve in the input stream for star 1 and star 2, respectively (0 for no, 1 for yes). Velocity curves precede light curves in the input. So with IFVC1=0 and IFVC2=1, **DC** expects the first curve it sees to be a velocity curve of star 2.

NLC: This integer is the number of light curves in the input stream. With IFVC1=1, IFVC2=1, and NLC=3, **DC** expects to encounter two velocity curves (the first for star 1, the second for star 2) and then 3 light curves. It will do a simultaneous adjustment of these five curves, producing one corrected value for each parameter that is the same for all curves (e.g. mass ratio) and 3 corrected values for each parameter that differs for the 3 light curves (e.g. limb darkening). The 3 curve-dependent parameter results for a given stellar component are written in direct succession. Input and output parameters, their differences (corrections from input values), corrected values, and estimated standard errors are listed by parameter number (see list in §11 on pages 28–30). Curve-dependent parameters are listed also by curve number, according to the input order of the curves. “Curve 0” means “curve-independent.”

IFTIME: An integer that indicates the presence (IFTIME=1) or absence (IFTIME=0) of eclipse timings in the data input stream.

KO: **DC** can process multiple parameter subsets in a given run, as explained under the method of multiple subsets (MMS, see §13.1 on page 39). Control integer KO provides three options to the user who has established a “scratch pad” data file on the local computer. Such a scratch file should be designated as input-output unit 9. The purpose of this provision is to save computing time by writing all of the derivatives and residuals on the scratch file so that they can be read back for further subset solutions in a later submission (in case you think of some subsets that you first thought were not needed). Weighting can be changed for the re-submission. The later run(s) then will take extremely little computing time, since all the hard computation has been done. This feature is much less important than it once was because of improvements in machine speed and intrinsic program speed, but it is still available. Multiple subsets can be processed within a given run without using the scratch pad (set KO=2, KDISK=0). KO can be 0, 1, or 2, as follows:

KO=0: Do the base set and stop. Actually KO=0 serves little purpose, as simple omission of subsets

has the same result.

KO=1: Read in the derivatives and residuals that were written to the scratch pad in a previous run. Process these according to the weighting schemes, subsets, Marquardt λ 's, and VLR factors entered for this run. Do not re-compute any derivatives. A run with this option should take only a few seconds or less of computing time.

KO=2: Write the derivatives and residuals generated in this run on the scratch pad for future use. **KO=2** can be used with **KDISK=0** if the user does not wish to change the present contents of the scratch pad (or if there is no scratch pad). In this case the program still can process multiple subsets of the parameter list but, of course, cannot come back later to process this run's data further.

KDISK: Set **KDISK=1** to use the scratch pad, or **KDISK=0** not to use it. Caution: the program will crash if no scratch pad has been set up and you set **KDISK=1**.

ISYM: **DC** can carry out its solutions with either asymmetrical (**ISYM=0**) or symmetrical (**ISYM=1**) derivatives. Solution convergence is somewhat better with symmetrical derivatives, but they take almost twice as long to compute as asymmetrical derivatives. Because of the improved convergence, fewer iterations may be needed with **ISYM=1**, but the individual iterations will take more machine time. Recommendations are to use **ISYM=0** for strong (well determined) solution situations, and **ISYM=1** when convergence is a problem or you are in the final fine-tuning stages. Setting **ISYM=1** is particularly advisable when large increments (**DEL**'s) are used to compute the numerical derivatives. See Wilson & Biermann (1976) or Wilson (1979), Eqns. 20-21.

NPPL: (number of points per line) The number of data triplets (time or phase, velocity or light, weight) on each line of the input data stream. The number of points per output line is the same. **NPPL** can be as small as 1 or as large as 5.

MAGLITE: Control integer that is 0 if **DC** input is flux (light) or 1 if **DC** input is magnitude.

N1L, N2L: These are the coarse (low) grid integers. As explained in §16 (Special Features) on page 49, they apply to certain derivatives for which computing time can be saved by using coarse grids. Otherwise they are just like **N1** and **N2**.

NOISE: This integer is on the same input lines as the curve-dependent parameters for light curves, so there is one value for each light curve (no values for velocity curves or eclipse timings). **NOISE** specifies how level-dependent weights are to be applied (see discussion of weighting in §18 on page 11).

KSD: An integer array that is 0, 1, or 2 for each input sub-dataset (velocity, light, or eclipse timings). The **KSDs** tell **DC** whether to apply the input standard deviations (σ 's) to compute curve-dependent weights (**KSD=0**), to apply **DC**'s internally computed σ 's for the weights (**KSD=1**), or to apply σ 's based on one or two restricted phase ranges for the weights (**KSD=2**).

SIGMA: The estimated standard deviations of the various observed light and velocity curves. **SIGMA** specifies the relative weights of the curves in the solution (weight is inversely proportional to **SIGMA**

squared). See the discussion of weighting in §18 on page 11.

SPHAS1, SPHAS2, SPHAS3, SPHAS4: The start/stop phases of two phase ranges for calculating sums of squares of residuals (see §4.9 on page 12).

LINKEXT: Integer to set the designated band for interstellar extinction. **LINKEXT** should be 0 if there is no designated band and should be the same as the band number (**IBAND**) otherwise. There must be a designated band if multi-band curves are to be solved for interstellar extinction. Extinction in bands other than the designated band are computed via ratios determined by Cardelli, Clayton, & Mathis (1989). The ratios are expressed via fitted functions from that paper that are based on effective wavelength.

DESEXTINC: Interstellar extinction (in magnitude) in the designated band.

XUNIT: This quantity is intended only for flux (as opposed to magnitude) input, and allows rescaling of observed fluxes so that the input flux numbers can be in a convenient range. If input fluxes are simply inverted magnitudes (e.g. $10^{-0.4V}$ for V magnitudes), then **XUNIT** should be unity. However, it may be convenient to enter rescaled (i.e. wrongly scaled) fluxes in cases where they would otherwise be of order, say, 10^{-10} or 10^3 . Then **XUNIT** can be the multiplier that restores the flux numbers to directly inverted magnitudes. If working in (presumably standard) magnitudes with control integer **MAGLITE**=1, of course, then don't worry about **XUNIT**—just set it to 1.0000.

13. Solution Convergence and Related Issues

The most essential guideline relevant to solution convergence is to compare the mean weighted residual [input] and mean weighted residual [predicted] that are written with each **DC** iteration. Here is an example:

Mean residual for input values	Mean residual predicted	Determinant
0.9837237712112231D-03	0.9837202893928959D-03	0.979816D+00

Convergence is shown when the input number becomes smaller as the iterations go by. The mean input residual will approach the mean predicted residual when the solution is reasonably linear and free from serious correlation problems. Correlations can often be large (near ± 1.0000) without seriously degrading a solution if there are not “too many” large ones and the solution is approximately linear (meaning that 2nd and higher derivatives in the **DC** equation of condition are not important). Even in the best of circumstances, the input mean residuals will eventually (given enough iterations) jiggle around within a small range. The range is not zero for several reasons such as solution non-linearities and finite precision of the numerical model. Often the jiggling range can be reduced by going to finer grids or by reducing the list of adjusted parameters. Whether such tactics have astrophysical significance is best judged for individual cases.

There are three main reasons for poor convergence or convergence failure in differential corrections solutions, and it is important to understand all three. The problems can occur in combination, but let us try to understand them separately.

1. One problem is a broad, shallow minimum in parameter space. This problem can result in slow convergence, even when only a few parameters are being adjusted, if the minimum is very shallow indeed. It is a problem not only for **DC**, but for any minimization scheme, since it is intrinsic to parameter space itself. Nothing can be done except to compute with precision (use a fine grid) and do many iterations.

2. A second problem, also easy to understand, is lack of precision in computing residuals and derivatives. Since many derivatives in **DC** are computed numerically, we have two obvious sources of error connected with finite differencing, which approximates the true derivative of a synthesized observable, f , by the slope near to a given operating point, either asymmetrically,

$$\frac{\partial f}{\partial p_n} = \frac{f(p_n + \delta p_n) - f(p_n)}{\delta p_n},$$

or symmetrically,

$$\frac{\partial f}{\partial p_n} = \frac{f(p_n + \frac{1}{2}\delta p_n) - f(p_n - \frac{1}{2}\delta p_n)}{\delta p_n}.$$

Here, p_n is a parameter and δp_n is its increment. Pseudo-random errors arise from limited precision in the computed f while systematic errors come from approximation of $\partial f/\partial p_n$ by $\delta f/\delta p_n$. A symmetrical derivative is a better representative of the true derivative, but takes longer to compute. Of course, the systematic error can be reduced by taking smaller δp_n , but the increment should not be too small or errors in computation of f will become important. Inaccuracy in computing f can be reduced by use of a finer numerical grid, but at the cost of increased machine time.

3. The third problem is more subtle, particularly because it is due to a combination of *two* conditions, neither of which causes convergence difficulty when acting alone. It is important to realize that this third problem is quite distinct from the other two and can degrade convergence even if the other problems are completely absent. The two interacting conditions that cause the problem are *non-linearity in the appropriate equation of condition* and *parameter correlation*. The first of these conditions can be expressed equivalently by saying that second or even higher order derivatives are required in the differential corrections equation of condition (which normally is written with only first derivatives). One way of seeing into the nature of this problem is to notice that a correct solution algorithm needs to know about parameter correlations, but a linear Least Squares algorithm cannot compute them correctly because it has no knowledge of the higher order derivatives. As a result there will be virtual tradeoffs among the invisible higher order terms that result in incorrect predictions of parameter corrections. It usually is not practical to add second order terms to the equation of condition for several reasons, so three procedures that improve convergence greatly while keeping only linear terms will now be discussed.

13.1. Method of Multiple Subsets (MMS)

One realistic remedy for poor convergence is to break the full parameter set into subsets and thus reduce the complexity of the correlations (Wilson & Biermann 1976). This procedure, in which subsets A and B, or A, B and C, etc., are solved iteratively in a closed loop has come to be known as the Method of Multiple Subsets (MMS), and it works very well in practice. It seems that a **DC** solution has only minor difficulty in dealing with one or two fairly strong correlations among parameters. The real difficulty comes when there are many large or even moderately large correlations, and this circumstance accounts for the success of the MMS. Two important points need to be emphasized. First, the published standard errors should come from a final run that includes all adjusted parameters together—not from the subset solutions (which will give unrealistically low error estimates because they see only part of the correlation matrix). It must be said that even the standard errors for the full set will be of questionable accuracy when there is a complicated correlation matrix, but they will have to do in the absence of more rigorously computed standard errors. The second point is that one must resist the temptation to apply the corrections written in this final run (which is for error estimates only). It is the nature of the problem that, when the MMS is needed, it is needed all the way to the end. Even if you start from the exactly correct Least Squares minimum, the correlation problem discussed here will produce relatively large, erroneous corrections for the full parameter set. This outcome has nothing to do with the accuracy of the derivatives computed by **DC**, but only with the form of parameter space. Some persons have misunderstood this point and assumed that correlation problems occur only away from the correct solution and that they will disappear at the correct solution. Not only do the problems not then disappear, but they do not even lessen in severity.

13.2. Levenberg-Marquardt (L-M)

A very good trick for effecting convergence is the Levenberg-Marquardt procedure (Levenberg 1944; Marquardt 1963), which has been utilized with excellent success in several binary star solution programs such as those by Hill & Rucinski (1993), by Kallrath et al. (1998), and by Djurasevic (1992). The L-M algorithm operates on the matrix of normal equations so as to effect a compromise between the ordinary **DC** corrections (which usually give fast convergence but sometimes lead to convergence failure) and those of the method of steepest descent (convergent but sometimes slow). The **DC** normal equations are re-scaled as prescribed by Marquardt, so as to have diagonal elements of unity, although it would be possible to apply the procedure without re-scaling. At the heart of the scheme is a quantity λ that is added to each diagonal element of the re-scaled normal equations. A decision faced by all those who adopt the scheme is that of how to fix the value of λ , and various kinds of iterations have been used in the literature to optimize λ , including one in the original Marquardt (1963) reference. The **DC** program operates on a somewhat different idea, following experience that—at least in binary star problems—a broad range of λ 's typically gives nearly the same corrections. Usually such λ 's are very small, and one finds results for say $\lambda = 10^{-4}$, 10^{-5} , or

10^{-6} that differ by so little that the choice is inconsequential. Because the **DC** program already has been set up to make solutions for many subsets of the main parameter set, it is easy to do solutions for many λ values, so that no iteration on λ is needed. One simply includes a line of data for each such solution. For example, if not sure of the λ value to use, just do solutions (same **DC** submission) for as many as deemed potentially interesting, all for the same parameter set or subset. The lines will look like this:

```
1111 1111 0111110 01110 11010 11111 11111 11111 01111 11111 11111 11110 06 1.000d-04 1.000
1111 1111 0111110 01110 11010 11111 11111 11111 01111 11111 11111 11110 06 1.000d-05 1.000
1111 1111 0111110 01110 11010 11111 11111 11111 01111 11111 11111 11110 06 1.000d-06 1.000
```

where the first 60 integer 1's and 0's on each line are the **KEEP**'s. The two digit integers (06, 06, 06 in the example lines) are the numbers of iterations (**NITERS**) for the subset solutions. The $1.000d-nn$ is the Marquardt λ and the 1.000 is the vector length reduction parameter **VLR** (see next subsection) for each solution. The λ value makes little difference within a wide range, although 10^{-11} or 10^0 may give much different answers. Execution time for the *first* iteration of each subset solution is negligible because **DC** remembers the residuals and parameter derivatives from the first base set solution. Subsequent iterations require usual execution times.

13.3. Vector Length Reduction (VLR)

A correction vector in parameter space can be inaccurate due to non-linearities and correlations that affect its length and direction. A common result is that an attempted improvement overshoots the proper solution. However, the vector usually points at least approximately in the right direction, so an easily implemented fix for convergence difficulties is to apply some fraction of the computed corrections so as to shorten the vector without changing its direction. Of course all corrections are to be reduced by the same factor since correct individual changes are unknown—the idea is simply to make the vector shorter. **DC**'s input now includes a multiplier that makes Vector Length Reduction (FORTRAN name **VLR**) easy. Ordinarily **VLR** is set to unity, but can be 0.5 to apply half the corrections, 0.25 to apply a quarter of the corrections, etc. Many iterations may be needed if **VLR** needs to be very small (say 0.10), but at least the scheme has always worked, in our experience. No decisions need be made about subset groupings, as in the **MMS**, which is a convenience for automatically iterated solutions. See §6.1 of Wilson (2006) for further remarks on **VLR**.

13.4. **MMS**, **VLR**, or **L-M**?

Either the **MMS** or Levenberg-Marquardt algorithm will nicely solve most convergence problems of the correlation – non-linearity type. The **L-M** algorithm is the more convenient because it does not extend the level of iteration. However, **L-M** alone cannot handle the most severe convergence problems, where a combination of **L-M** and **MMS** and/or **VLR** may be needed. So **DC** has several

options for dealing with convergence problems.

13.5. Error Distributions and Standard Error Estimates

Estimated standard errors are written along with the parameter corrections for the base set and subset solutions. Published comments occasionally appear to the effect that the **DC** program produces unrealistic error estimates, although the errors are computed from the covariance matrix by the standard method. For example, a 1997 paper comments about **DC**: “... *the estimates of errors of the adjustable parameters are unrealistically small. The reason is partly the strong correlation between the relatively many parameters, and partly the non-normal distribution of measurement errors.*” Such remarks have some formal validity, but are misleading for the following reasons:

1. *Strong correlation between the relatively many parameters*

In significantly non-linear situations the standard errors will be only approximately right, but that is also the case with all other binary star fitting programs that now exist. The shortcoming is not peculiar to **DC**. However, an impression that the problem *is* peculiar to **DC** obviously has arisen, and one can speculate as to why that is so. An obvious point is that some solution programs avoid such complaints by not computing error estimates at all, or by doing so with invented rules that produce comfortably large estimates. However, some other solution programs do compute standard error estimates, *but usually not for adjusted mass ratios*. Experience shows that mass ratios cause most of the problems with fitting and with error computation. This is so because the mass ratio often has major correlations with other parameters, and it also shows particularly non-linear behavior. Programs that cannot adjust the mass ratio (most programs) will not encounter the problem.

2. *Non-normal distribution of measurement errors*

The cited phrase probably does not literally refer to *measurement errors*, whose distributions for light curves are as close to Gaussian as one is likely to find for any astrophysical measurements. (Observers measure light with respect to nearby comparison stars whose brightness and color are close to that of the variable star. The observers typically expend considerable thought and effort to eradicate possible systematic errors.) The phrase more likely refers to *non-normal distribution of residuals*, which one finds when a real binary does not match the model very well (perhaps the binary has un-recognized spots or circumstellar gas, etc.), or if the solution is at an incorrect local minimum. If interpreted in this way, the statement is correct, although trivial. Under any interpretation, the comment has nothing to say about **DC**.

D. Terrell has carried out experiments by solving synthetic light curves by means of **DC**. The light curves contained simulated Gaussian errors and, of course, their parameters were known. He found that the parameter errors agreed with those expected from the standard output errors.

14. Interactive Branching and Iteration of Solutions

Beginning with the 2010 version, **DC** can do iterative solutions by setting the number of iterations (integer **NITERS**) greater than 1. For previous versions, many persons were surprised at having to re-submit the program at each iteration. The lack of automatic iteration was not an oversight but a way to cause users to look at the progress of a solution. Even without automatic iteration, some persons believe whatever comes out of the machine, although some results may violate common sense. In this highly non-linear problem, the solution process can indeed get into trouble and produce inappropriate corrections. To some extent it is possible to have a program detect inconsistent and strange results. **DC** has a few such kinds of detection and message generation, and further messages about such problems may be worked into future versions. However, until such screening has been made very extensive, users are encouraged to think about every iteration. The **DC** of 2010 and later allows inspection of each iteration without requiring re-submission. Each iteration of the base set and subsets is displayed and each base set and subset solution of however many iterations begins from the input parameter values for that **DC** submission. The new way works the same as the old way except that the machine now chases down the iterated corrections to a Least Squares minimum (within the usual jiggle, of course), separately and independently for the base set and each subset, if enough iterations are done.

There is another important reason to examine each iteration. In real situations an experienced person will decide which parameters to adjust and which to hold fixed as part of an interactive process with the iterative solution. Occasionally the set of adjusted parameters will remain unchanged from start to finish, but usually it will not. Because of this reality, **DC** is set up for interactive branching, by which each iteration produces solutions of as many subsets of the main (base) set as are requested (control integer **KO** must be set to 2). Notice that the sample input data supplied with the program request several subset solutions via extra lines of **KEEP**'s at the end of the data block. There is no limit to the number of such subset solutions—you can ask for a hundred or more if you like. The first iteration of each subset requires almost no machine time since the observational equations have already been computed and their solution takes almost no time compared to their generation. So the idea is to decide, after each submission, which set of adjustable parameters to follow from that point on (to the next such change). If you decide not to change parameters and convergence was satisfactory, the solution is finished. Interactive branching is discussed in Wilson (1988).

The most cogent reason for personal monitoring of **DC** iterations is simply this: *there is more to astrophysics than parameter estimation*. Recognition of strange and unusual behavior points the way to the really major insights and discoveries. Many close binaries are not well represented by even our best generic models, and in some cases the problem is not due just to approximate computation but to quite unsuspected features. To drive this point home, imagine applying one's favorite "standard" light curve model automatically to one of the classic strange objects—perhaps one with an unusual kind of circumstellar disk. Persons familiar with β Lyrae might be amused by imagining the outcome for that binary. One will get answers (meaningless ones, naturally), but

will have at best only a much reduced chance for insights into the real problems. Often one really needs to get in there and watch things develop.

15. Photometric Bands and Radiative Physics

The radiative treatment was re-done for 2003, with the previous effective wavelength prescription replaced by a bandpass prescription (Van Hamme & Wilson 2003) based on model stellar atmospheres by Kurucz (1993). Since the 2003 version, the radiative routine has computed bandpass-integrated normal emergent intensities, as well as Planck intensities, by means of Legendre polynomials whose coefficients were stored in data files `atmcof.dat` and `atmcofplanck.dat`. Beginning with the 2013 version, band-integrated model and Planck intensities have been based on cubic spline fits to model (and Planck) intensities, with the cubic spline coefficients pre-computed and stored in data files `atmcof.dat` and `atmcofplanck.dat`. These files are online at <http://faculty.fiu.edu/~vanhamme/lcdc/>, together with other supporting data files that are needed by **LC** and **DC** (`effwv1.dat`, and 19 files called `limcof_bp_preamble_*.dat`, with `*` for each file to be replaced by a chemical composition label). A path to these files must be set within the computing system or they must be in the same sub-directory as the programs. Besides 94 bands (viz. Table 2 on page 45), there are coefficients for 19 chemical compositions and 11 surface gravities (Table 1 on the next page). Note that if local interpolation in $[T_{eff}, \log g]$ for x and y limb darkening coefficients is requested (with the law-specifying integers LD1 and/or LD2 being negative, see §12.1 on page 32), a path to the bandpass-specific limb darkening tables must be set or the limb darkening files must be in the same sub-directory as the program. The limb darkening tables are in data files named `limcof_bp_*.dat`, with the asterisk to be replaced by the chemical composition indicator. They can be downloaded from the webpage <http://faculty.fiu.edu/~vanhamme/limb-darkening/>.

15.1. Band IDs, Chemical Composition, Temperature and Gravity Ranges

Allowed temperature ranges for stellar atmosphere emission vary according to $\log g$ and are listed in Table 1 on the next page, together with 19 abundances (relative to the Sun). Bands, band identification numbers (IBAND), and references for band response curves are in Table 2 on page 45. Chemical composition needs to be specified via input parameter ABUNIN. If ABUNIN is not one of the 19 Kurucz values (Table 1 on the next page), it is reset automatically to the nearest Kurucz value, as there is no interpolation in $[M/H]$. (**Important note:** program logic requires that data files `atmcof.dat` and `atmcofplanck.dat` remain unchanged. Please make these files “write protected” so they cannot be changed inadvertently.)

Table 1. Temperature Limits

$[M/H]^a$	$\log g$ (cgs)	T_{eff} Range (K)	$[M/H]^a$	$\log g$ (cgs)	T_{eff} Range (K)
-5.0 ... + 0.5	0.0	3500 ... 6000	-5.0	3.5	4000 ... 31000
+1.0	0.0	3500 ... 5750	-4.5 ... - 4.0	3.5	3750 ... 31000
-5.0 ... - 3.5	0.5	3500 ... 7000	-3.5 ... - 2.5	3.5	3500 ... 31000
-3.0	0.5	3500 ... 7250	-2.0	3.5	3500 ... 29000
-2.5 ... + 0.2	0.5	3500 ... 7500	-1.5 ... + 0.1	3.5	3500 ... 31000
+0.3	0.5	3500 ... 7250	+0.2 ... + 0.3	3.5	3500 ... 30000
+0.5	0.5	3500 ... 7000	+0.5	3.5	3500 ... 29000
+1.0	0.5	3500 ... 7250	+1.0	3.5	3500 ... 27000
-5.0 ... - 3.0	1.0	3500 ... 7500	-5.0 ... - 4.5	4.0	4250 ... 35000
-2.5	1.0	3500 ... 7750	-4.0	4.0	4000 ... 35000
-2.0 ... - 1.5	1.0	3500 ... 8000	-3.5	4.0	3750 ... 35000
-1.0	1.0	3500 ... 8250	-3.0 ... - 2.5	4.0	3500 ... 35000
-0.5 ... - 0.1	1.0	3500 ... 8500	-2.0	4.0	3500 ... 32000
0.0	1.0	3500 ... 8250	-1.5 ... - 0.1	4.0	3500 ... 35000
+0.1 ... + 0.3	1.0	3500 ... 8500	0.0	4.0	3500 ... 39000
+0.5 ... + 1.0	1.0	3500 ... 8250	+0.1 ... + 0.2	4.0	3500 ... 35000
-5.0 ... - 3.0	1.5	3500 ... 9000	+0.3	4.0	3500 ... 34000
-2.5 ... - 1.5	1.5	3500 ... 8500	+0.5	4.0	3500 ... 33000
-1.0 ... + 0.5	1.5	3500 ... 9000	+1.0	4.0	3500 ... 31000
+1.0	1.5	3500 ... 8750	-5.0	4.5	4750 ... 40000
-5.0 ... + 0.1	2.0	3500 ... 14000	-4.5	4.5	4250 ... 40000
+0.2 ... + 0.5	2.0	3500 ... 10500	-4.0	4.5	4500 ... 40000
+1.0	2.0	3500 ... 10000	-3.5 ... - 3.0	4.5	4000 ... 40000
-5.0 ... + 0.2	2.5	3500 ... 19000	-2.5	4.5	3750 ... 40000
+0.3	2.5	3500 ... 18000	-2.0 ... - 0.1	4.5	3500 ... 40000
+0.5	2.5	3500 ... 17000	0.0	4.5	3500 ... 49000
+1.0	2.5	3500 ... 11000	+0.1 ... + 0.3	4.5	3500 ... 40000
-5.0	3.0	3500 ... 26000	+0.5	4.5	3500 ... 37500
-4.5	3.0	3750 ... 26000	+1.0	4.5	3500 ... 35000
-4.0 ... - 3.5	3.0	3500 ... 26000	-5.0	5.0	5000 ... 50000
-3.0	3.0	3500 ... 27000	-4.5 ... - 3.5	5.0	4500 ... 50000
-2.5 ... + 0.1	3.0	3500 ... 26000	-3.0 ... - 2.5	5.0	4250 ... 50000
+0.2 ... + 0.3	3.0	3500 ... 25000	-2.0 ... + 0.3	5.0	3500 ... 50000
+0.5	3.0	3500 ... 24000	+0.5	5.0	3500 ... 45000
+1.0	3.0	3500 ... 21000	+1.0	5.0	3500 ... 40000

$${}^a \log\left(\frac{M}{H}\right)_\star - \log\left(\frac{M}{H}\right)_\odot$$

Table 2. Bandpass List

Origin	Symbol	Identifier	Reference
Strömgren	u	1	Crawford & Barnes (1974)
	v	2	
	b	3	
	y	4	
Johnson	U	5	Buser (1978)
	B	6	Ažusienis & Straižys (1969)
	V	7	
	R	8	Johnson (1965)
	I	9	
	J	10	
	K	11	
	L	12	
	M	13	
	N	14	
Cousins	R_C	15	Bessell (1983)
	I_C	16	
Bessell	UX	17	Bessell (1990)
Bessell	BX	18	
Bessell	B	19	
Bessell	V	20	
Bessell	R	21	
Bessell	I	22	
Tycho	B_T	23	Bessell (2000)
	V_T	24	
Hipparchos	H_p	25	
KEPLER		26	Koch et al. (2010); Van Cleve (2005)
COROT	SIS	27	Cautain et al. (2006)
COROT	EXO	28	
Geneva	U	29	Rufener & Nicolet (1988)
	B	30	
	$B1$	31	
	$B2$	32	
	V	33	

Table 2—Continued

Origin	Symbol	Identifier	Reference
Vilnius	<i>V1</i>	34	Straizys & Zdanavičius (1965)
	<i>G</i>	35	
	<i>U</i>	36	
	<i>P</i>	37	
	<i>X</i>	38	
	<i>Y</i>	39	
	<i>Z</i>	40	
	<i>V</i>	41	
Milone	<i>S</i>	42	Milone & Young (2008)
	<i>iz</i>	43	
	<i>iJ</i>	44	
	<i>iH</i>	45	
	<i>iK</i>	46	
YMS94	<i>iz</i>	47	Young, Milone, & Stagg (1994) Milone & Young (2005)
	<i>iJ</i>	48	
	<i>iH</i>	49	
	<i>iK</i>	50	
	<i>iL</i>	51	
	<i>iL'</i>	52	
	<i>iM</i>	53	
	<i>in</i>	54	
	<i>iN</i>	55	
Sloan DSS	<i>u'</i>	56	Fukugita et al. (1996)
	<i>g'</i>	57	
	<i>r'</i>	58	
	<i>i'</i>	59	
	<i>z'</i>	60	
HST STIS	Ly α	61	Baum et al. (1996)
	Fclear	62	
	Fsrf2	63	
	Fqtz	64	
	C III	65	

Table 2—Continued

Origin	Symbol	Identifier	Reference
	Mg II	66	
	Nclear	67	
	Nsrf2	68	
	Nqtz	69	
	cn182	70	
	cn270	71	
	Oclear	72	
	Oclear-lp	73	
	[O II]	74	
	[O III]	75	
2MASS	<i>J</i>	76	Cohen, Wheaton, & Megeath (2003)
2MASS	<i>H</i>	77	
2MASS	<i>K_s</i>	78	
SWASP		79	Pollacco et al. (2006)
MOST		80	Walker et al. (2003)
GAIA (2006) ^a	<i>G</i>	81	Jordi et al. (2006)
GAIA (2010)	<i>G</i>	82	Jordi et al. (2010)
GAIA (2010)	<i>G_{BP}</i>	83	
GAIA (2010)	<i>G_{RP}</i>	84	
GAIA (2010)	<i>G_{RVS}</i>	85	
Milone ^b	230	86	Kallrath et al. (1998)
	250	87	
	270	88	
	290	89	
	310	90	
	330	91	
Ca II triplet ^c	858	92	
WIRE ^d	<i>V+R</i>	93	Bruntt et al. (2006)
Lunar Ultraviolet Telescope	LUT	94	Cao et al. (2011)

^aBand #81 is the original and now obsolete GAIA broad *G* photometric band of 2006. As the GAIA instrumental system has evolved since then, there are now four GAIA passbands defined (82 through 85): a very broad *G* band from 350 to 1000 nm, two broad bands representing the

blue (330-680 nm) and red (640-1000 nm) ends of the *G* band, respectively, and the Radial Velocity System narrow band (in the region 847-874 nm near the Ca II triplet).

^bResponse curves for bands 86 to 91 are rectangular in shape, have widths of 20 nm and are centered on the wavelength (in nm) indicated by the band name number. They are at the request of Gene Milone and useful for certain space-based observations in the UV.

^cThis is a trapezoidal band centered on the Ca II triplet, also requested by Gene Milone. Response increases linearly from 0 to 1 between 847 and 851 nm, is equal to 1 between 851 and 867 nm, and decreases linearly from 1 to 0 between 867 and 869 nm.

^dThe WIRE field tracker response is unknown but estimated to be approximately “*V* + *R*” according to Bruntt et al. (2006). We constructed band No. 93 to consist of the short-wavelength ascending part of the Johnson *V* band (between 475 and 530 nm), the long-wavelength descending part of the Johnson *R* band (from 700 to 960 nm), and a flat part with a fixed value of 1 between 530 and 700 nm.

15.2. Atmosphere to Blackbody Transitions

If a T_{eff} , $\log g$ combination is modestly outside the range of applicability (see Table 1 on page 44), the program smoothly transfers to the Planck intensity curve for the given band over a predetermined range that is coded into the program. We call the procedure ramping. Four ramp parameters are required, two for $\log g$ that set the ramp interval below $\log g = 0.0$ (GLOWTOL) and above $\log g = 5.0$ (GHIGHTOL), and two for T_{eff} . At the lower T_{eff} limit, there is only one temperature ramp interval, TLOWTOL. For each $\log g$, the ramp interval at the upper T_{eff} limit is a fixed percentage of the maximum model atmosphere temperature. This fixed percentage is determined from the ramp interval of the highest model temperature of 50,000 K (THIGHTOL). The program, as supplied, has GLOWTOL = 4.0, GHIGHTOL = 4.0, TLOWTOL = 1500 K, and THIGHTOL = 50,000 K. As an example of what this means, consider the THIGHTOL of 50,000 K coupled with a highest Kurucz T_{eff} of 50,000 K. The **LC** and **DC** programs will then apply Kurucz atmosphere intensities for local surface elements with T_{eff} up to 50,000 K, smoothly transfer between the 50,000 K atmosphere and a 100,000 K blackbody between 50,000 K and 100,000 K, and use blackbody intensities above 100,000 K. GLOWTOL and GHIGHTOL were smaller prior to February, 2004. The current numbers seem to work well but the user may change them. They should be the same in **LC** and **DC**.

16. Special Features

Coarse and fine grids: To save computing time, **DC** uses both coarse and fine computing grids, and applies the fine grids only where accuracy requires them. The fine grids are for the residuals and for derivatives with respect to parameters e , ω , F_1 , F_2 , ϕ_0 , i , Ω_1 , Ω_2 , q , spot latitude, spot longitude, and spot radius. Fine grids are required for parameters that are in some way geometrical. The derivatives of the non-geometrical parameters g_1 , g_2 , T_1 , T_2 , A_1 , A_2 , L_1 , L_2 , x_1 , x_2 , and spot temperature are computed with the coarse grids. The fine grids are specified by **N1**, **N2** and the coarse grids by **N1L**, **N2L**, for stars 1 and 2, respectively.

Synthetic Noise: Light curves with synthetic Gaussian noise can be produced by entering a non-zero value for the **LC** input quantity **STDEV**, which is labeled “fract. sd.” in the output. **STDEV** is in the unit of light at the reference phase (PHN). The way that the noise (scatter) scales with light level is controlled by input integer **NOISE**. Scatter is proportional to light level for **NOISE**=2, proportional to the square root of the light level for **NOISE**=1, and independent of light level for **NOISE**=0. The random number generator needs a seed (FORTRAN name **SEED**), labeled “seed” in the output. **SEED** should be larger than 100000001. and smaller than twice that value. The procedure should work for other **SEED**’s, but one should stay within the mentioned range for best results. Noise generation works the same way for radial velocities, except that there is no variation of the scatter with level. For velocity scatter, enter **STDEV** in the velocity unit (**VUNIT**). Users who have a favorite random number generator can easily replace the program generator with their own.

17. Simultaneous Solutions

A capability of **DC** that should be used in most circumstances is that of making simultaneous multi-band light curve solutions as well as simultaneous (one or two curve) radial velocity and multi-band light curve solutions (Wilson 1979). Beginning with the 2015 version simultaneous velocity, light and eclipse timing solutions are also a capability (Wilson & Van Hamme 2014). The advantages are to avoid inconsistencies among solutions of the separate curves, to reduce the number of free parameters (no need to have both photometric and spectroscopic mass ratios, nor to have a separate inclination for each light curve), and to utilize information that is discarded in separate solutions (the knowledge that there is only one true mass ratio, one true eccentricity, etc.). Weighting is very important in simultaneous solutions and is discussed in Wilson (1979) and in §18 on page 11. A sample **DC** data file supplied with the program is for a simultaneous solution of two radial velocity curves, four light curves, and a collection of eclipse timings. There can be 0, 1, or 2 velocity curves, any number of light curves (the program is dimensioned for up to 50 light curves, which seems more than ample for realistic situations), and one set of eclipse timings. Input integers **IFVC1** and **IFVC2** tell **DC** whether velocity curves for stars 1 and 2 are expected, and **NLC** tells how many light curves are expected. Input integer **IFTIME** indicates whether eclipse timings are entered or not. The **DC** solution will produce one correction for each adjusted curve-independent parameter (most of them) and n corrections for each adjusted curve-dependent parameter (L_1 , L_2 , x_1 , x_2 , ℓ_3). It takes no more (actually somewhat less) machine time for a simultaneous solution of n curves than for n individual curve solutions, so there really is no reason not to take advantage of this feature. If one has misgivings about the simultaneous solution, separate solutions always can be carried out in addition. One particularly illogical practice in a few papers has been to publish averaged parameters from separate solutions and to offer the averages as a substitute for a simultaneous solution. However, the average of separate solutions will not be a correct solution of *any* of the separate curves. Ask “what is the proper way to take such an average?” The answer is that there is no self-consistent way—the only way to have the results of a simultaneous solution is to *do* a simultaneous solution.

17.1. Absolute Flux Computations and Direct Distance Estimation (DDE)

Although most applications of the near future are likely to continue to work with arbitrarily scaled fluxes, **LC** and **DC** now can operate in standard physical flux units—specifically, centimeter-gram-second (cgs) units (Wilson 2004, 2005, 2007, 2008; Wilson & Van Hamme 2010; Wilson, Van Hamme, & Terrell 2010). Theoretical fluxes are based on model stellar atmospheres that are naturally in standard physical units, so the only calibration is on the observational side. Specifics of model flux generation are in §3.3 of Wilson (2008). Absolute solutions for distance (d) measurement are

independent of binary morphology⁴ in regard to accuracy (no need to reject OCs or near-contact binaries for distance measurement). They also generate standard distance errors routinely and reduce the human workload. The process is called Direct Distance Estimation (**DDE**), as $\log_{10} d$ becomes an ordinary solution parameter.⁵ The basic idea of **DDE** can be inverted by adoption of an independently known distance so as to strengthen the solution of a poorly conditioned binary light curve, such as one with shallow partial eclipses or even no eclipses (an ellipsoidal variable), and then becomes Inverse Distance Estimation, or **IDE** (Wilson et al. 2009). Although an arbitrarily scaled EB light curve yields only a one-temperature ($1T$) solution, in principle one can solve for temperatures of both stars ($2T$ solutions) from combined absolute flux curves and RV curves if the calibrative information is accurate. Only a subset of standard bands in Table 2 have published flux calibration estimates, and these are collected in Table 3 starting on page 53. The logical basis for $2T$ solutions is explained in Wilson (2007, 2008). Areas such as third light, overdetermined solutions, and sources of observational and calibrative error are covered in Wilson (2008).

Control integer IFCGS should be set to 1 for absolute operation or to 0 for traditional operation in arbitrary flux units. Parameter $\log_{10} d$ (DPCLOG) replaces the luminosity parameter(s) when IFCGS=1. Attempts to adjust L_1 or L_2 will not work with IFCGS=1, nor will attempts to adjust $\log_{10} d$ when IFCGS=0.

17.2. T - d and T - d - A Theorems

Theorems in §4 of Wilson (2008) tell how many light curves should be solved simultaneously from light-velocity observations for various combinations of T_1 , T_2 , $\log_{10} d$, and interstellar extinction (A). The theorems' basic logic and usefulness also are covered there and checks via simulations are described qualitatively. The T - d theorem may be stated as

EB light curves optimally yield distance along with temperatures of both stars if and only if the light curves are standardized and calibrated, two substantially different photometric bands are fitted, and the absolute length scale is known or derivable (usually from RV's).

The idea is to use two light curve bands to solve for T_1 , T_2 , and $\log_{10} d$, but only one band if T_1 or T_2 or $\log_{10} d$ is dropped from the list. With only one light curve, a $[T_1, T_2, \log_{10} d]$ solution is indeterminate and will not come to definite results. With more than two light curves, the solution is overdetermined (usually leaving misfits for all or most of the bands). The T - d - A theorem is a generalization that adds a designated interstellar extinction, A , as an adjustable parameter. Three bands will be optimal if A is added (so that the list becomes $T_1, T_2, \log_{10} d, A$). Here “designated” means that A pertains to a designated photometric band such as Johnson V . The practicality of extinction solutions from light/velocity curves of real binaries is not yet clear, so work on that

⁴Results for overcontact (OC) and near-contact examples are not inferior to those for well detached binaries.

⁵Convergence is improved with $\log_{10} d$ rather than d as a solution parameter.

problem continues. See Wilson (2007, 2008) for more detailed information on the T - d theorem, and particularly §7 of Wilson (2008) for ideas on the T - d - A theorem that were gained partly from simulations. Central to those ideas is adoption of a reference extinction (in magnitude) in a designated photometric band.

17.3. Absolute Flux Solutions: Strategy and Procedures

Weighting is especially important in simultaneous absolute solutions because RV's and absolute flux curves are likely to differ numerically by many orders of magnitude. The standard deviations that govern curve-to-curve weighting need to be at least roughly right, even in the first iterations. They can be wrong by a factor 10 or so at the beginning, but not by factors of 10^2 or larger. Consider an example of a star with $V \approx 10$ mag and RV's of order 100 km/s. A $V = 0.00$ mag star has a bandpass flux of about $0.36 \text{ ergs}^{-1}\text{cm}^{-3}$, so a $V \approx 10$ mag star's flux will be about 0.36×10^{-4} in the same units. If the RV's and V fluxes have similar fractional noise, the RV's (being of order 3×10^6 times larger numerically) will have noise $\approx 3 \times 10^6$ times larger. Then, since weights are inversely proportional to noise squared, the RV weights should be around 10^{13} times smaller than the light curve weights. Now suppose the RV and light curve sigmas actually entered are about the same (for lack of thought applied to weighting). The light curve weights will be 10^{13} times smaller than they should be and the Least Squares algorithm will basically ignore the light curves and run to an operating point from which the program cannot recover. Such a gross discrepancy is not so likely in traditional non-absolute solutions, where fluxes are usually within a factor 10 of unity.

The present **DC** version is written to operate with input of flux or stellar magnitude. For absolute solutions, entered magnitudes should be actual standard magnitudes (not magnitude differences) with **MAGLITE** set to 1. Entered fluxes (with **MAGLITE** set to 0) can be in the unit of the bandpass flux of a comparison star, in which case the input quantity **XUNIT** should be set to $10^{-0.4m_{\text{comp}}}$. Alternatively, entered fluxes can be already in cgs units (specifically $\text{ergs}^{-1}\text{cm}^{-3}$), in which case **XUNIT** and **CALIB** should be set to unity. Probably most persons will find the magnitude option easier. Note that distance, d , or actually $\log_{10} d$, replaces bandpass luminosity, L_{band} , as a (possible) free parameter. Obviously it is not possible for both d and L_{band} to be free parameters because a definite L_{band} is determined if d (and the other system parameters) are set. The programs compute L_{band} 's as auxiliary parameters. Note that **DC**'s actual distance parameter is $\log_{10} d$ rather than d , as $\log_{10} d$ solutions converge better than d solutions. An adopted (say spectroscopic) temperature is commonly the main source of distance uncertainty in $1T$ solutions, so we recommend that published $1T$ absolute solutions be done for at least two assumed temperatures, separated by something like 100 K or a few hundred K, to allow interpolation or extrapolation of results.

Table 3. Absolute Flux Calibrations

Band Origin	Band Name	IBAND	CALIB	Reference
Strömgren	u	1	1.180	Fabregat & Reig (1996)
			1.1799	Wilson, Van Hamme, & Terrell (2010)
	v	2	0.869	Fabregat & Reig (1996)
			0.8420	Wilson, Van Hamme, & Terrell (2010)
	b	3	0.584	Fabregat & Reig (1996)
			0.58895	Wilson, Van Hamme, & Terrell (2010)
	y	4	0.369	Fabregat & Reig (1996)
			0.37415	Wilson, Van Hamme, & Terrell (2010)
Johnson	U	5	0.435	Johnson (1965, 1966)
			0.419	Bessell (1979)
			0.4175	Bessell, Castelli, & Plez (1998)
			0.4306	Wilson, Van Hamme, & Terrell (2010)
	B	6	0.688	Johnson (1965, 1966)
			0.660	Bessell (1979)
			0.632	Bessell, Castelli, & Plez (1998)
			0.6266	Wilson, Van Hamme, & Terrell (2010)
	V	7	0.378	Johnson (1965, 1966)
			0.361	Bessell (1979)
			0.3631	Bessell, Castelli, & Plez (1998)
			0.36895	Wilson, Van Hamme, & Terrell (2010)
	R	8	0.185	Johnson (1965, 1966)
			0.17813	Wilson, Van Hamme, & Terrell (2010)
	I	9	0.0899	Johnson (1965, 1966)
			0.08643	Wilson, Van Hamme, & Terrell (2010)
	J	10	0.0340	Johnson (1965, 1966)
			0.03147	Bessell, Castelli, & Plez (1998)
	K	11	0.0039	Johnson (1965, 1966)
			0.00401	Bessell (1979)
			0.003961	Bessell, Castelli, & Plez (1998)
	L	12	0.00080	Johnson (1965, 1966)
			0.000708	Bessell, Castelli, & Plez (1998)
	M	13	0.00022	Johnson (1965, 1966)
	N	14	0.0000124	Johnson (1965, 1966)

Table 3—Continued

Band Origin	Band Name	IBAND	CALIB	Reference
Cousins	R_C	15	0.225	Bessell (1979)
			0.2177	Bessell, Castelli, & Plez (1998)
	I_C	16	0.122	Bessell (1979)
			0.1126	Bessell, Castelli, & Plez (1998)
Geneva	U	29	0.03290	Rufener & Nicolet (1988)
	B	30	0.06595	Rufener & Nicolet (1988)
	$B1$	31	0.06748	Rufener & Nicolet (1988)
	$B2$	32	0.06172	Rufener & Nicolet (1988)
	V	33	0.03532	Rufener & Nicolet (1988)
	$V1$	34	0.03692	Rufener & Nicolet (1988)
	G	35	0.02945	Rufener & Nicolet (1988)
2MASS	J	76	0.03129	Cohen, Wheaton, & Megeath (2003)
2MASS	H	77	0.01133	Cohen, Wheaton, & Megeath (2003)
2MASS	K_s	78	0.004283	Cohen, Wheaton, & Megeath (2003)

Note. — CALIB fluxes refer to a star of zero magnitude and are in $\text{erg s}^{-1}\text{cm}^{-3}$. They are listed with the number of digits given in the original publications.

18. Input Lines, Including Observations and Weights

Identification of the various input quantities can be done in several ways, for example via Appendices A or B, by examining the output where the numbers are written with labels, or by comparing the input data lines with the FORTRAN READ statements.

For **LC** (see the sample input file in Appendix A on page 73), there is a “more or less standard” set of input lines with control integers and parameters for each curve, and n such sets of lines can be concatenated to produce n output curves. Output can be a light curve, a radial velocity curve, or a spectral region containing mixed absorption and emission spectral lines (perhaps blended). The “standard set” of input data lines is only “more or less standard” because there can be extra lines to tell about circumstellar attenuating clouds and/or about bright or dark star spots and/or about spectral lines. For light or velocity curves, the number of input lines for a given curve is $(11 + c + s)$, where c is the number of circumstellar clouds and s is the number of star spots for the two stars combined. One of the 11 mandatory lines is the *stop* line for clouds and two others are stop lines for star spots, with one stop line to follow the lines of spot parameters for each star. The cloud stop line should contain a number greater than 100.0 but less than 200.0 in the first field, which corresponds to a cloud x -coordinate. Each of the two stop lines for spots should contain a number greater than 200.0 in the first field, which corresponds to a spot latitude on the normal spot parameter lines. According to the value of MPAGE, **LC** computes a light curve (MPAGE=1), a radial velocity curve (MPAGE=2), a set of spectral regions (MPAGE=3), star dimensions (MPAGE=4), plane of sky coordinates for images (MPAGE=5), or conjunction times and eclipse timing residuals (MPAGE=6). Spectral line profile computation requires extra input lines for the line characteristics. The several sets of input lines for a velocity curve, light curve, etc., need not be of the same binary since each curve computation is an independent operation. A final line stops execution and should contain integer 9 in the field normally occupied by MPAGE (that final line is not counted in the $11 + c + s$ lines of the individual curves).

For **DC** (a sample input file is in Appendix B on page 75), three lines of DEL’s enter first. The next line pertains to the base parameter set and contains 60 KEEP’s (10 of which are not used at present) for the 50 adjustable parameters, the number of iterations (NITERS) to be done, the Marquardt λ (called XLAMDA), and the VLR factor (called VLR, see §12.3, page 34). Note that the ‘KEEP’ line is changed from the 2007 version, in that IFDER has been moved to another line (see below) and also has a changed interpretation. Formerly IFDER controlled whether the matrix of observational equations (partial derivatives and residuals) was written for a given parameter subset (or base set). Now it controls whether the observational equations are written *for the base set*—they are *not* written for the subsets. The old write control integers IFR (for radii, R/a) and IFM (for the normal equations, identity matrix, and correlations) are now gone—those matrices are always written for the base set and each subset. Next comes a line of spot identification integers (KSPA, NSPA, KSPB, NSPB), then six lines of control integers and curve-independent parameters. Note that the first of these six lines now contains IFDER, moved from the KEEP lines and entered only once. Next comes a line of third body parameters, then as many lines of curve-dependent

parameters as there are observed curves, then as many lines of spot parameters as there are spots (and also two spot “stop” lines), then as many lines of circumstellar cloud parameters as there are clouds (and a cloud “stop” line), then the observed radial velocities, then the observed light curves, then the observed eclipse timings, and finally as many lines of **KEEP**’s for subset solutions as the user wants. These final **KEEP** lines are of the same format as the **KEEP** line for the base set and also contain **NITERS**, the Marquardt λ , and **VLR**. Note that solutions that are triggered by these **KEEP** lines can be for varied Marquardt λ and **VLR** factor as well as for parameter subsets. For example, one can run two solutions for exactly the same parameters but with different λ or **VLR** factor.

The observed radial velocity curves and light curves of the **DC** input data stream are entered as triplets (time or phase, velocity or light, weight), with up to 5 triplets (i.e. data points) per line. If eclipse timings are provided, they are entered after the velocity and light curves, with data points including eclipse time, type (1 for type I and 2 for type II) and weight. See §10.6 on page 27 for details on eclipse time type. The number of data points per line is set by input integer **NPPL** (*number of points per line*). A possible formatting problem, caused by velocities not being of the same order of magnitude as light measures, is solved by providing for the separate entry of a convenient sized velocity unit. The (0, 1, or 2) velocity curves, **NLC** light curves, and eclipse timings are separated by “data stop lines” that serve two purposes. They identify the last data line for each velocity or light curve and also tell how many data points are on that last line (1, 2, 3, 4, or 5). The only number needed on a data stop line is in the first field and should be $-(10000 + k)$, where k is the number of data points on the preceding line. Example: if there are 2 data points on the preceding line, the “stop” number should be -10002 . Some computing systems require “something” in the remaining fields of the line and blanks are suitable, but there must be at least blanks (not an absence of all characters). No special stop information is to follow the final curve because the program already knows how many curves to expect, and it knows which is a velocity curve and which is a light curve because it has already read **IFVC1**, **IFVC2**, and **NLC**. After the data stop line are the **KEEP** lines for subsets of the main (i.e. base) set, with one **KEEP** line for each subset. A line with integer 2 in column 2 should follow the final line of subset **KEEP**’s to signal the end of subset processing and of the entire job.

19. The Scaling of Run Time

The main computational activity of **LC** consists of summations over surface grid elements in forming the observable fluxes and velocities. The number of grid elements on each star is essentially proportional to the square of the grid fineness integer for that star (N_1 , N_2 , defined as the number of latitude rows per hemisphere), so run time very nearly scales with $N_1^2 + N_2^2$. Computations of all other kinds take negligible time. Of course, run time in **LC** also scales with the number of output points, or more precisely, with that number plus 1 (because one extra point is done at the phase of normalization). Eccentric orbits take longer than circular orbits because the local physical

computations must be done anew at each phase. Note that a very small eccentricity (say 0.000001) differs in this regard from one of exactly zero because the program skips the extra computations only if $e = 0$. The actual time factor depends on whether the old approximate (MREF=1) or detailed (MREF=2) reflection model is specified. The MREF=2 case takes longer for both circular and eccentric orbits, but it takes *much* longer (usually impractically longer) for eccentric orbits. If MREF=2, then run time also depends on NREF, the number of multiple reflections. There will also be some dependence on the particular binary star configuration. For example, runs for stars with small $r = R/a$ go faster than those with large r . The situation is sufficiently complicated so that actual speed factors are best estimated via a few experiments at low grid fineness.

For **DC**, many of the same considerations apply as for **LC**. However, there are four grid fineness integers because **DC** uses both a high and a low grid for each star (viz. §16 on page 49), so run time scales with $(P + 1)(N_1^2 + N_2^2) + P_L(N_1^2 + N_2^2)_L$, where P is the number of fine grid parameters and P_L is the number of low (coarse) grid parameters under adjustment in the main (base) set of parameters. The first term involves $P + 1$ rather than just P because not only derivatives but also residuals must be computed, and the residuals are done with the fine grid. Run time in **DC** also scales with the number of observations (not with the number of observations plus 1 because there is no phase of normalization, as there is with **LC**). **DC** iterations with ISYM=1 (i.e. symmetrical derivatives) will take nearly twice as long as those with ISYM=0 (i.e. asymmetrical derivatives).

20. Common Difficulties

1. **Minimum dimensioning:** One of the most frequent reasons for failure in program execution is under-dimensioning of arrays. These failures are disconcerting because very strange things can happen and usually no logical interpretation of the machine error messages is apparent. Array dimensions need be changed only in the main programs (**LC** and **DC**), not in the subroutines. This can be done in the “Parameter Wrapper” at the beginning of the main programs (described in §21.4 on page 64). Table 4 (Dimensioning vs. Grid Fineness) on page 59 lists minimum dimensions of the arrays RV, GRX, GRY, GRZ, RVQ, GRXQ, GRYQ, GRZQ, SLUMP1, SLUMP2, SRV, SGRX, SGRY, SGRZ, SRVQ, SGRXQ, SGRYQ, SGRZQ, SRVL, SGRXL, SGRYL, SGRZL, SRVQL, SGRXQL, SGRYQL, SGRZQL, SLMP1, SLMP2, SLMP1L, SLMP2L, FR1, FR2, GLUMP1, GLUMP2, GRV1, GRV2, XX1, XX2, YY1, YY2, ZZ1, ZZ2, GMAG1, GMAG2, CSBT1, CSBT2, RF1, RF2, RFTMP, SXX1, SXX2, SYY1, SYY2, SZZ1, SZZ2, SGMG1, SGMG2, SGRV1, SGRV2, SGLM1, SGLM2, SCSB1, SCSB2, SRF1, SRF2, SGLM1L, SGLM2L, SGRV1L, SGRV2L, SXX1L, SXX2L, SYY1L, SYY2L, SZZ1L, SZZ2L, SGMG1L, SGMG2L, SCSB1L, SCSB2L, SRF1L, SRF2L, ERV, EGRX, EGRY, EGRZ, ELMP1, EGLM1, EGRV1, EXX1, EYY1, EZZ1, EGMG1, ECSB1, ERF1, ERVQ, EGRXQ, EGRYQ, EGRZQ, ELMP2, EGLM2, EGRV2, EXX2, EYY2, EZZ2, EGMG2, ECSB2, ERF2, ERVL, EGRXL, EGRYL, EGRZL, ELMP1L, EGLM1L, EGRV1L, EXX1L, EYY1L, EZZ1L, EGMG1L, ECSB1L, ERF1L, ERVQL, EGRXQL, EGRYQL, EGRZQL, ELMP2L, EGLM2L, EGRV2L, EXX2L, EYY2L, EZZ2L, EGMG2L, ECSB2L, ERF2L, SFR1, SFR1L, ERF1, ERF1L, SFR2, SFR2L, EFR2, and EFR2L. The minimum

dimension of each of these arrays depends on the grid fineness. For example, if the grid fineness integer (N_1 , N_2 , N_{1L} , or N_{2L}) is 30, the minimum dimension is 762. Arrays **SNTHH**, **CSTHH**, **SNTHL** and **CSTHL** are dimensioned to at least N_1+N_2 . Arrays **SNFIH**, **CSFIH**, **SNFIL**, **CSFIL**, **TLDH**, **TLDL**, **STLDH**, **STLDL**, **ETLDH**, and **ETLDL** are dimensioned to the sum of the minimum dimensions for both stars (in above long list of arrays). For N 's of 30, this would then be $762 + 762 = 1524$. The arrays **PHAS**, **FLUX**, **MNTYPE** and **WT** are dimensioned to include all the observations in all curves *plus* the blanks on the last data lines *plus* the blank observations on the data stop lines (see **DC** sample input file in Appendix B). Arrays **OBS** and **HOLD** are dimensioned to $[number\ of\ observations \times (number\ of\ parameters\ in\ Least\ Squares\ solution + 1)]$. A curve-dependent parameter counts n times for n light curves. Thus if you had 98 observations in star 1's velocity curve, 102 in star 2's velocity curve, 470 in one light curve, and 530 in another, the first factor would be $1200 = 98 + 102 + 470 + 530$. Then, if you adjust i , g_2 , L_2 , and x_1 , the second factor is $7 = 2 + 2 \times 2 + 1$ (remember, L_2 and x_1 are curve-dependent and count twice each, since there are two light curves). The last +1 is for the residuals. **OBS** and **HOLD** need then be dimensioned to a minimum of $1200 \times 7 = 8400$.

2. The parameter increments must be neither too large (gives systematic errors) nor too small (gives numerical noise). These increments are called the **DEL**'s (FORTRAN name). For most parameters, **DEL**'s of about 1% of the parameter value are appropriate. However, particular circumstances sometimes affect that guideline, so common sense and experience are the best guides. Finer grids allow smaller **DEL**'s. Use of the **ISYM=1** option allows larger **DEL**'s before curvature effects become important.

3. It is best to have the initial parameter guesses for differential corrections based on experiments with the light-velocity program (**LC**). Column 5 in the main block of output from **LC**, which is $\ell_1 + \ell_2 + \ell_3$, should approximately match the observed light values.

4. The distinction between direct light (column 5 of **LC** output) and normalized light (column 6) can be a source of confusion. Remember that normalized light is only intended for convenience in initial graphical trials and that it has no counterpart in **DC**, which deals with direct light only (see also §4.1 on page 5). If this continues to confuse you, forget that normalized light exists and work always with direct light.

5. Always check Ω_1 and Ω_2 to be sure they are in the permitted range for given q , F , and e . Exact lobe filling or otherwise special Ω 's that are computed by the program in modes -1, 4, 5, and 6 will be correct and need not be checked. In differential corrections, be sure that both the input and incremented values of Ω_1 and Ω_2 are within allowed ranges. Overcontact Ω 's should be between the critical values for inner and outer contact (see the table of critical Ω 's in the file **critout.tab** available from the FTP download site). For detached stars, Ω is greater than the critical Ω for inner contact.

6. Certain parameters cannot be adjusted in certain program modes. The reason is either that the parameters are not free, but functionally determined from the mode logic, or that they have no effect

Table 4. Dimensioning vs. Grid Fineness

N	Minimum Dimension ^a	N	Minimum Dimension ^a	N	Minimum Dimension ^a
1	...	35	1034	69	3979
2	4	36	1092	70	4094
3	9	37	1157	71	4214
4	16	38	1217	72	4330
5	23	39	1281	73	4447
6	33	40	1345	74	4576
7	45	41	1418	75	4696
8	58	42	1482	76	4823
9	72	43	1555	77	4946
10	88	44	1628	78	5082
11	106	45	1703	79	5210
12	125	46	1777	80	5341
13	147	47	1857	81	5477
14	169	48	1935	82	5613
15	195	49	2013	83	5747
16	220	50	2098	84	5890
17	250	51	2180	85	6029
18	278	52	2269	86	6165
19	310	53	2352	87	6314
20	342	54	2445	88	6457
21	378	55	2533	89	6606
22	413	56	2628	90	6750
23	452	57	2722	91	6906
24	491	58	2818	92	7056
25	530	59	2915	93	7210
26	575	60	3011	94	7364
27	617	61	3116	95	7521
28	667	62	3215	96	7681
29	713	63	3320	97	7840
30	762	64	3423	98	8002
31	813	65	3536	99	8163
32	867	66	3642	100	8331
33	919	67	3755	101	8496
34	978	68	3866	102	8670

^aFor arrays RV, ..., EFR2L enumerated on the preceding page.

on the computed light or velocity. Attempts to adjust those parameters are the most common cause of blowups in subroutines **SQUARE** and **DMINV**. L_2 cannot be adjusted in any mode greater than 0 unless **IPB** has been set to 1. In mode -1 , parameters g_1 , T_1 , T_2 , A_1 , Ω_1 , Ω_2 , and x_1 cannot be adjusted. In mode 1, parameters g_2 , T_2 , A_2 , Ω_2 , L_2 , and x_2 cannot be adjusted. In mode 3, Ω_2 , and L_2 cannot be adjusted. In mode 4, Ω_1 and L_2 cannot be adjusted. In mode 5, Ω_2 and L_2 cannot be adjusted. In mode 6, Ω_1 , Ω_2 , and L_2 cannot be adjusted. Limb darkening coefficients x_1 and x_2 cannot be adjusted in any mode if **LD1** or **LD2** is negative (i.e. if limb darkening coefficients are computed as functions of local temperature and other quantities).. Luminosities $L_{1,2}$ should not be adjusted in absolute solutions (i.e. with **IFCGS**=1), nor should $\log_{10} d$ be adjusted in non-absolute solutions (i.e. with **IFCGS**=0).

7. Luminosities (input L_1 and L_2) are approximately 4π times larger than computed light values (output ℓ 's). Therefore, to obtain $\ell_1 + \ell_2$ of about unity outside eclipse, enter input luminosities that add to about 4π . Note that the traditional treatment of third light as if it were third luminosity is incorrect (for a discussion of luminosity vs. light, see §6 on page 17).

8. Critical Ω 's depend on orbital eccentricity, e . Although a table of lobe filling circular orbit potentials for synchronous and non-synchronous rotation is available along with this document (file **critout.tab**), such tables for eccentricity and rotation combined would be too extensive for practicality. Therefore, to find critical eccentric Ω_1 and Ω_2 , run **LC** in mode 6. The program will replace your input Ω 's for both stars with the critical Ω 's in the output listing.

9. Overcontact binaries, such as W UMa stars, require finer grids than do detached and semi-detached binaries because the numerics of the neck region are particularly difficult to treat accurately. N 's about 50% larger than normal are recommended for overcontact systems.

10. The programs can apply the detailed reflection model of Wilson (1990) for eccentric as well as circular orbits, but eccentric cases then use an enormous amount of machine time. If you are doing an eccentric binary and the programs run almost forever, check to see if you set **MREF**=2. In most realistic situations, the old approximate reflection (**MREF**=1) should be entirely adequate for eccentric binaries.

11. A very common cause of failed runs is simply inadvertent shifting of numbers on the input lines out of their proper fields, so that leading digits or signs are clipped away, or the numbers are not read at all, or they are read under the wrong name. Be sure to keep a copy of the sample data in exactly its original form for later comparison.

12. Sometimes users make changes that cause the programs not to work. Perhaps the program will work for the immediate application, but will fail in another situation. Be sure to keep a copy of the entire program in exactly the form supplied, so that you can see whether it runs correctly in the circumstances under question.

13. Be sure not to mix subroutines from program versions.

14. Spot A and Spot B cannot be *the same spot*! So the set **KSPA**, **NSPA**, **KSPB**, **NSPB** cannot be

something like 2, 3, 2, 3, which could cause blowups even when only one spot is adjusted.

15. **DC** input is to contain *two* sets of lines for photometric band-dependent quantities, each having as many lines as there are input light curves (for example, 5 light curves, 5 lines—in the same order). These lines are broken into *two* sets only to keep page width conveniently small. The band identifying numbers are only in the first set but ‘belong’ also to the second set, and of course in the same order. **DC** will certainly crash if either set has the wrong number of lines.

21. Summary of Program Version Differences

21.1. Differences from Pre-1992 Versions

1. Star spot parameters can now be adjusted.
2. One can now use an optional non-linear limb darkening law (pre-1992 versions had only the linear cosine law).
3. The reflection effect now can be computed either with the detailed model of Wilson (1990) or with the approximate reflection model of the old program (which is faster). Multiple reflection is included in the detailed model.
4. The **LC** and **DC** programs are much faster than pre-1992 versions.
5. The specification of phase range in **LC** was made more convenient than before, with allowance for phases outside the range 0 to 1.
6. The stars now orbit counter-clockwise in the sky (for $i < 90^\circ$), rather than clockwise as in the old program. This makes a difference in output only for pictures of the binary (MPAGE=5).
7. Messages about exceeding limiting lobes are now generated if the stars exceed the lobes at all, instead of only when at least one grid point falls in the hole near the inner Lagrangian point, as in the old program.
8. Star spots now can optionally move in longitude, keeping pace with the physical surface of an asynchronously rotating star, rather than being tied to the coordinate grid.
9. For both stars, **LC** now provides absolute mass, bolometric luminosity, equivalent sphere radius, and approximate absolute mean surface gravity.

21.2. Differences Between 1998 and 1992 Versions

1. The output from **LC** now is determined by setting integer MPAGE to 1, 2, 3, 4, or 5. Older versions had light and velocity curve output on the same pages, which made the page format

inconveniently wide. The output should now be easier to read. The input file format now differs for the various values of **MPAGE** (samples are provided at the public FTP site).

2. **DC** solutions can include the Marquardt λ factor.
3. Semi-transparent circumstellar clouds (at fixed locations in the rotating frame) can be included.
4. Rotational spectral line profiles can be computed (other broadening mechanisms no; blending yes).
5. The program now is entirely in double precision.
6. Non-linear limb darkening via a square root law is an added option (1992 version had only the logarithmic and linear laws).
7. One can now use either time or phase as the independent variable in **DC**. Previously only phase could be used. Use of time as the independent variable allows solutions for the ephemeris parameters t_0 , P , and dP/dt , as well as the apsidal motion parameter $d\omega/dt$ and spot motion and aging parameters. *More extensive use of this provision might well be made, as it complements ephemerides from eclipse timings. Baselines in time are substantially different for whole curve and eclipse timing ephemerides of most EB's, so application of both procedures often enhances perspectives.*
8. The stepped independent variable in **LC** now can be either time or phase. Previously only phase could be stepped. If time is the stepped variable, phase is computed and also listed. If phase is the stepped variable, time is computed and also listed.
9. **DC** now can read the observational input in 1, 2, 3, 4, or 5 data triplets per line, according the the value of **NPPL**. The old version read only 5 triplets per line.
10. Input and output formats for some parameters of **LC** and **DC** have been expanded to more digits. Some quantities that are likely to range over many orders of magnitude are now entered and written in D format. See the sample input data sets in Appendices A (page 73) and B (page 75) for examples.
11. The argument of periastron, ω , now is in radians rather than degrees. The new parameter $d\omega/dt$ is in radians per day (since t is Julian Date in days).
12. Spot longitudes, latitudes, and angular radii now are in radians, as are their DEL's and corrections. Previously they were in degrees.
13. The **DC** output format for solution results has been changed and should now be more convenient.
14. **LC** now generates coordinates of plane of sky projected images for use with an external plot program (**MPAGE**=5).

15. Error estimates written by **DC** are now standard errors. Previously they were probable errors. Standard errors seem to be more commonly used in the literature than probable errors.
16. Simulated observational scatter can be applied to the light curves computed by **LC**. Similar provisions for radial velocities and line profiles were not included in 1998.

21.3. Differences Between 2003 and 1998 Versions

1. The bandpass-based radiative prescription that is discussed extensively in §15 (on pages 43–49) replaces the much simpler previous one based on effective wavelength. The (Kurucz) atmospheres are newer, $\log g$ is now an atmosphere parameter (allowing for handling giants, sub-giants, etc., in addition to main sequence stars), and 19 chemical compositions can be specified.
2. Another significant change concerns **MODE=3** operation, where A_2 , g_2 , x_2 , and y_2 now are free parameters, not set equal to A_1 , g_1 , x_1 , and y_1 . Since T_2 may differ considerably from T_1 in mode 3, it seems logical to eliminate those constraints.
3. Input now is via **LC** and **DC** files with standard names that are accessed by **OPEN** statements (FORTRAN). The 1998 and earlier versions utilized the UNIX or LINUX command line syntax, e.g. `lcjob.exe < algolin.d1 > algolout.d1`, but that syntax does not work in some computing environments. The 2003 version of **LC** automatically looks for a file called `lcin.active`, so the idea is to copy `algolin.d1` (or whatever file) into `lcin.active`. The **DC** program looks for a file called `dcin.active`. Of course, persons who like the UNIX command line scheme can just comment out the relevant **LC** and **DC** **OPEN** statements. **LC** and **DC** output goes to files called `lcout.active` and `dcout.active`. Those files are usually copied or renamed to avoid being overwritten in the next **LC** or **DC** run.

21.4. Differences Between 2007 and 2003 Versions

1. Light-time and velocity shifts due to 3b's are now included, with six corresponding adjustable parameters added to **DC**.
2. **LC** and **DC** now interpolate (locally) in $[T_{eff}, \log g]$ for x and y limb darkening coefficients from the Van Hamme (1993) tables for any of 19 compositions ($[M/H]$) if the law-specifying integer (LD1, LD2) is negative (§12.1 on page 32). The programs adopt the input coefficients if the integer is positive, so fixed limb darkening is still an option.
3. **DC** now has 60 parameter channels rather than the previous 35 and has 42 adjustable parameters rather than the previous 34. Channels 1 to 29 have the same parameter assignments as before, while the previous channels 30 to 35 are now assigned to 3b parameters. Channel 41

is assigned to $\log_{10} d$ and channel 42 to designated extinction. The band-dependent parameters ($L_1, L_2, x_1, x_2, \ell_3$) have been moved from channels 31-35 into channels 56-60. Channels 36 to 40 and 43 to 55 can be assigned to other parameters, according to needs, via minor changes in the **DC** main program.

4. **DC** now can have either light or magnitude input. Magnitudes are converted internally to light and solutions are always done in light (flux).
5. **LC**'s radial velocity curves now can have Gaussian scatter with a specified standard deviation. Formerly only light curves could have scatter.
6. The "Parameter Wrapper" contributed by A. Prša facilitates array re-dimensioning. Although re-dimensioning has always required changes only in the **LC** and **DC** main programs, the changes had to be done individually for many quantities. The Parameter Wrapper greatly reduces the number of needed changes and also improves organization by grouping quantities according to array size.
7. **DC** now writes phase along with the un-weighted and weighted observational equations (first column) so as to allow easy graphing of derivatives and residuals vs. phase.
8. The observational equations of partial derivatives and residuals (both unweighted and weighted) are now in D format (formerly they were in F format).
9. Some I/O quantities that previously were in F format are now in D format (to allow for possible very large or small values).
10. **LC** now has an output column labeled "direct magnitude" that is $-2.5 \log_{10} \ell$, where $\ell = \ell_1 + \ell_2 + \ell_3$ from column 5 of **LC** output. No constant is added to the direct magnitudes. The purpose is for **LC** to have the exact analog of **DC**'s magnitude input, mainly for making magnitude graphs that compare observed and computed light curves. **LC**'s preceding column, labeled "set-level magnitude," allows a light curve to be pinned at a specified level at the normalization phase.
11. **DC** versions of 2003 and earlier divided the square root of each point weight by 100 so as to make each weight 10,000 times smaller and allow weighted derivatives and residuals to fit more easily within formatted fields. Those quantities are now written in D (rather than F) format so the factor of 100 has been removed, thus making all sums of squares of residuals 10,000 times larger than before. Of course this change has no effect of parameter corrections or their uncertainties.

21.5. Differences Between 2010 and 2007 Versions

1. **LC** and **DC** now (optionally) can operate in absolute physical flux, so as to measure distances directly and find temperatures of both EB components. See §17.1 on page 50.

2. Following suggestions from Petr Harmanec and Andrej Prša, many of the physical and astronomical constants in the **LC** and **DC** programs have been updated. Table 5 on the next page lists the values of various constants as currently coded in the program.
3. **DC** now can iterate base set and subset solutions within a submission (see §14 on page 42).
4. **DC** now can solve for interstellar extinction in a designated photometric band, while accounting for extinction in all entered bands (see page 37). Whether that facility gives reliable results in practice for real stars is not yet clear, although it works for synthetic data.
5. **LC** now computes a flux-weighted average surface temperature, based on a temperature inferred from spectroscopy or color index made at a specific aspect (i.e. phase and inclination). That is, **LC** outputs a converted temperature that can be regarded as a proper global mean, given an input temperature that pertains to an observer’s estimate from a particular vantage point (see page 10).
6. **DC** now can write an input file for **LC** (see §4.1 on page 5). Naturally that file must be copied into file `lcin.active` in order to be recognized by **LC**.
7. Some quantities and formats of the **LC** and **DC** input/output files are changed (see samples in Appendices A and B on pages 73–76).

21.6. Differences Between 2013 and 2010 Versions

1. The original list of 25 photometric bands introduced in 2003 has now been expanded to 93 bands (listed in Table 2 on page 45).
2. There is a new starspot algorithm with enhanced precision, with the old simple model remaining as an option. Spot aging is now in place, governed by times of onset, maximum size, and disappearance. Spot motion is now decoupled from star rotation and characterized by drift parameters (see §4.11 on page 13).
3. Time smearing is a light curve option (see §9 on page 20).
4. **DC** now can compute curve-dependent weights iteratively (i.e. compute new weights for each parameter iteration). **DC** does this on a curve by curve basis as specified by the `KSD` array (see §4.9 on page 12).
5. **DC** now writes mean residuals instead of sums of squares of residuals.
6. The observational equations of partial derivatives and residuals (both weighted and unweighted) are now in **E** format (formerly they were in **D** format). The same change was made in several of **LC**’s output columns. The reason for this change is that some plot programs do not recognize **D** format.

Table 5. Physical and Astronomical Constants

Name	Value	Reference
Speed of light (c)	$2.99792458 \times 10^{10} \text{ cm s}^{-1}$	(1)
Avogadro constant (N_0)	$6.02214199 \times 10^{23} \text{ mol}^{-1}$	(1)
Boltzmann constant (k)	$1.3806488 \times 10^{-16} \text{ erg K}^{-1}$	(1)
Stefan-Boltzmann constant (σ_{SB})	$5.670373 \times 10^{-5} \text{ erg cm}^{-2} \text{ s}^{-1} \text{ K}^{-4}$	(1)
Planck constant (h)	$6.62606957 \times 10^{-27} \text{ erg}\cdot\text{s}$	(1)
Thomson cross section (σ_e)	$0.6652458734 \times 10^{-24} \text{ cm}^2$	(1)
Astronomical Unit ^a	$1.49597870700 \times 10^{13} \text{ cm}$	(2)
Constant of gravitation (G)	$6.67428 \times 10^{-8} \text{ cm}^3 \text{ g}^{-1} \text{ s}^{-2}$	(1)
Heliocentric gravitational constant (GM_{\odot})	$1.32712442099 \times 10^{26} \text{ cm}^3 \text{ s}^{-2}$	(2)
Mass of the Sun ^b (M_{\odot})	$1.988435 \times 10^{33} \text{ g}$	(3)
Radius of the Sun ^c (R_{\odot})	$6.9566 \times 10^{10} \text{ cm}$	(4)
Luminosity (bolometric) of the Sun ^d (L_{\odot})	$3.846 \times 10^{33} \text{ erg s}^{-1}$	(3)
Absolute bolometric magnitude of the Sun ^d	+4.75	...
Effective temperature of the Sun ^e	5779 K	...

References. — (1) NIST CODATA2010 (Mohr, Taylor, & Newell 2012); (2) the Astronomical Almanac Online (<http://asa.usno.navy.mil/SecK/Constants.html>); (3) Harmanec & Prša (2011); (4) Haberreiter, Schmutz, & Kosovichev (2008)

^aDefined, exact value; adopted by the IAU at its XXVIII General Assembly in 2012 (Resolution B2).

^bValue proposed by Harmanec & Prša (2011) and obtained from the values of G and GM_{\odot} .

^cValue obtained by Haberreiter, Schmutz, & Kosovichev (2008); also adopted by Torres, Andersen, & Giménez (2010).

^dValue proposed by Harmanec & Prša (2011) and consistent with the zero point of the bolometric luminosity scale ($M_{\text{bol}} = 0$ corresponding to $L = 3.055 \times 10^{28} \text{ W}$) adopted by IAU Commissions 25 (Stellar Photometry and Polarimetry) and 36 (Theory of Stellar Atmospheres) in 1999.

^eCalculated from the adopted values for L_{\odot} , R_{\odot} and σ_{SB} .

21.7. Differences Between 2015 and 2013 Versions

1. Eclipse timings can be added as **DC** data input, allowing for unified velocity-light-eclipse timing solutions in a one-step process (see §10.6 on page 27). Utilization decisions among the three datatypes can thereby be avoided.
2. Program **LC** can compute timing residuals (say for plots) with **MPAGE**=6. See §4.1 and §4.2 of Wilson & Van Hamme (2014) for circular and eccentric orbits, respectively.
3. An algorithm explained in §4.3 of Wilson & Van Hamme (2014) allows automatic determination of eclipse type (star 1 or star 2 eclipsed).
4. An algorithm explained in §4.4 of Wilson & Van Hamme (2014) allows **DC** to recover from an inaccurate starting ephemeris that may place one or more timings on a wrong orbit cycle.
5. The option for automated curve-dependent weighting for light curves and RV curves now extends to eclipse timings.
6. **DC**'s observational equations (output) now optionally begin with four columns that are useful for making plots. The headers are 'time,' 'phase,' 'observed,' and 'computed.' The columns are written if print control integer **IFOC**=1 and not written if **IFOC**=0. Residuals (observed minus computed) were already in the output of earlier versions (last column) and continue to be.
7. A 94th band (Lunar Ultraviolet Telescope) was added in 2016 as a small update to the 2015 version of the program.

22. Summary of Control Integers

22.1. [0,1] Control Integers

Integer	0	1
IPB	normal value	decouple L from T
IFAT1, IFAT2	blackbody	atmospheres
IFSMV1, IFSMV2	spots fixed in longitude	spots can move in longitude
IFDER	not write derivative matrix	write derivative matrix
IFCGS	program flux & luminosity units	cgs flux & luminosity units
IF3B	ignore third body parameters	utilize third body parameters
IFLCIN	do not write LC input file	write LC input file
IFOC	do not write extra DC output	write extra DC output
IFTIME	no input of eclipse timings	eclipse timings included in data input
ISYM	asymmetrical partial derivatives	symmetrical partial derivatives
ICOR1, ICOR2	not apply RV proximity corrections	apply RV proximity corrections
KDISK	not use disk scratch pad	use disk scratch pad
KEEP	adjust a given parameter	keep parameter fixed
MAGLITE	DC input is flux (light)	DC input is magnitude
KSPEV	no spot aging	spots grow and decay
NOMAX	spot aging function is trapezoidal	spot aging function is triangular

22.2. [0,1,2] and [1,2] Control Integers

Integer	0	1	2
JDPHS	n.a.	indep. variable is time	indep. variable is phase
NOISE	no level-dependent weights	scatter scales with \sqrt{level}	scatter scales with $level$
KO	DC does base set only	read from scratch pad	write on scratch pad
KSD (array)	curve weights based on input σ 's	based on computed σ 's	based on computed σ 's in restricted phase ranges
MREF	n.a.	approximate reflection	detailed reflection
KSPOT	n.a.	simple on/off spot algorithm	vector fractional area spot algorithm

22.3. [1,2,3] and [1,2,3,4,5,6] Control Integers

Integer	1	2	3	4	5	6
LD1, LD2	linear cosine law	log law	square root law	n.a.	n.a.	n.a.
MPAGE	light curves	velocity curves	line profiles	radii vs. phase	images	timing residuals

23. Acknowledgments

Andrej Prša alerted us to several bugs that are now fixed, and also contributed a block of FORTRAN statements that make re-dimensioning easier than before, in addition to suggestions for several other improvements, most of which have been implemented. In particular Andrej, as well as Petr Harmanec, made general and specific suggestions about updating of physical and astronomical constants. Thanks continue to the many persons who helped with suggestions, identification of bugs, and direct testing of the 2013 and earlier versions, as cited in previous documentation booklets. **LC** and **DC** program development was supported by U.S. National Science Foundation grant 0307561.

REFERENCES

- Avni, Y. 1976, *ApJ*, 209, 574
- Ažusienis, A., & Straizys, V. 1969, *Sov. Astron.*, 13, 316
- Baum, S., et al. 1996, *STIS Instrument Handbook*, Version 1.0, (Baltimore: STScI)
- Bessell, M. S. 1979, *PASP*, 91, 589
- Bessell, M. S. 1983, *PASP*, 95, 480
- Bessell, M. S. 1990, *PASP*, 102, 1181
- Bessell, M. S., Castelli, F., & Plez, B. 1998, *A&A*, 333, 231
- Bessell, M. S. 2000, *PASP*, 112, 961
- Bruntt, H., Southworth, J., Torres, G., Penny, A. J., Clausen, J. V., & Buzasi, D. L. 2006, *A&A*, 456, 651
- Buser, R. 1978, *A&A*, 62, 411
- Cao, L., Ruan, P., Cai, H., et al. 2011, *Science China: Physics, Mechanics & Astronomy*, 54, 558
- Cardelli, J. A., Clayton, G. C., & Mathis, J. S. 1989, *ApJ*, 345, 245
- Cautain, R., et al. 2006, *Bilan de transmission de l'instrument COROT*, COR-30000-NT-870-LAM
- Cohen, M., Wheaton, Wm. A., & Megeath, S. T. 2003, *AJ*, 126, 1090
- Crawford, D. L., & Barnes, J. V. 1974, *AJ*, 75, 978
- Diaz-Cordovés, J., & Giménez, A. 1992, *A&A*, 259, 227
- Djurasevic, G. 1992, *Ap&SS*, 197, 17

- Fabregat, J., & Reig, P. 1996, *PASP*, 108, 90
- Fukugita, M., Ichikawa, T., Gunn, J. E., Doi, M., Shimasaku, K., & Schneider, D. P. 1996, *AJ*, 111, 1748
- Haberreiter, M., Schmutz, W., & Kosovichev, A. G. 2008, *ApJ*, 675, 53
- Hadrava, P. 2004, *Publ. Astr. Inst. Acad. Sci. Czech*, No. 92, 1
- Harmanec, P., & Prša, A. 2011, *PASP*, 123, 976
- Hill, G., & Rucinski S. 1993, *Light Curve Modeling of Eclipsing Binary Stars*, ed. E. F. Milone (New York: Springer), 135
- Johnson, H. L. 1965, *ApJ*, 141, 923
- Johnson, H. L. 1966, *ARA&A*, 4, 193
- Jordi, C., et al. 2006, *MNRAS*, 367, 290
- Jordi, C., et al. 2010, *A&A*, 523, A48
- Kallrath, J., & Milone, E. F. 2009, *Eclipsing Binary Stars: Modeling and Analysis*, second edition (New York: Springer), pp. 179-182
- Kallrath, J., Milone, E. F., Terrell, D., & Young, A. T. 1998, *ApJ*, 508, 308
- Klinglesmith, D. A., & Sobieski, S. 1970, *AJ*, 75, 175
- Koch, D. G., et al. 2010, *ApJ*, 713, L79
- Kopal, Z. 1954, *Jodrell Bank Ann.*, I, 37
- Kurucz, R. L. 1993, *Light Curve Modeling of Eclipsing Binary Stars*, ed. E. F. Milone (New York: Springer), 93
- Levenberg, K. 1944, *Quart. J. Appl. Math.*, 2, 164
- Limber, D. N. 1963, *ApJ*, 138, 1112
- Marquardt, D. W. 1963, *J. Soc. Indust. Appl. Math.*, 11, 431
- Milone, E. F., Stagg, C. R., & Kurucz, R. L. 1992, *ApJS*, 79, 123
- Milone, E. F., & Young, A. T. 2005, *PASP*, 117, 485
- Milone, E. F., & Young, A. T. 2008, *JAAVSO*, 36, 110
- Mohr, P. J., Taylor, B. N., & Newell, D. B. 2012, *CODATA Recommended Values of the Fundamental Physical Constants: 2010*, NIST, 46

- Plavec, M. J. 1958, Mem. Soc. Roy. Liège, 20, 411
- Pollacco, D. L., et al. 2006, PASP, 118, 1407
- Rufener, F., & Nicolet, B. 1988, A&A, 206, 357
- Straižys, V., & Zdanavičius, K. 1965, Bull. Vilnius Obs., 14, 1
- Terrell, D., Munari, U., & Siviero, A. 2007, MNRAS, 374, 530
- Torres, G., Andersen, J., & Giménez, A. 2010, A&A Rev., 18, 67
- Van Cleve, J. 2005, Kepler Instrument Handbook,
<http://keplergo.arc.nasa.gov/Instrumentation.shtml>
- Van Hamme, W. 1993, AJ, 106, 2096
- Van Hamme, W., & Wilson, R. E. 1986, AJ, 92, 1168
- Van Hamme, W., & Wilson, R. E. 2003, ASP Conf. Ser. 298, *GAIA Spectroscopy, Science and Technology*, ed. U. Munari (San Francisco: ASP), 323
- Van Hamme, W., & Wilson, R. E. 2007, ApJ, 661, 1129
- Walker, G., et al. 2003, PASP, 115, 1023
- Wilson, R. E. 1979, ApJ, 234, 1054
- Wilson, R. E. 1988, *Critical Observations vs. Physical Models for Close Binary Systems*, ed. K. C. Leung (New York: Gordon and Breach), 193
- Wilson, R. E. 1990, ApJ, 356, 613
- Wilson, R. E. 1993, ASP Conf. Ser. 38, *New Frontiers in Binary Star Research*, ed. K. C. Leung & I. S. Nha (San Francisco: ASP), 91
- Wilson, R. E. 1994a, PASP, 106, 921
- Wilson, R. E. 1994b, International Amateur-Professional Photoelectric Photometry Communications, 55, 1
- Wilson, R. E. 2001, IBVS, 5076
- Wilson, R. E. 2004, New Astr. Rev., 48, 695
- Wilson, R. E. 2005, Ap&SS, 296, 197
- Wilson, R. E. 2006, ASP Conf. Ser. 349, *Astrophysics of Variable Stars*, ed. C Sterken & C. Aerts (San Francisco: ASP), 71

- Wilson, R. E. 2007, ASP Conf. Ser. 362, *The Seventh Pacific Rim Conference on Stellar Astrophysics*, ed. Y. W. Kang, H. W. Lee, K. S. Cheng, & K. C. Leung (San Francisco: ASP), 3
- Wilson, R. E. 2008, ApJ, 672, 575
- Wilson, R. E. 2012a, J. Astron. Space Sci., 29, 115
- Wilson, R. E. 2012b, AJ, 144, 73
- Wilson, R. E., Chochol, D., Komzik, R., Van Hamme, W., Pribulla, T., & Volkov, I. 2009, ApJ, 702, 403
- Wilson, R. E., & Biermann, P. 1976, A&A, 48, 349
- Wilson, R. E., & Devinney, E. J. 1971, ApJ, 166, 605
- Wilson, R. E., & Van Hamme, W. 2010, ASP Conf. Ser. 435, *Binaries—Key to Comprehension of the Universe*, ed. A. Prša & M. Zejda (San Francisco: ASP), 45
- Wilson, R. E., & Van Hamme, W. 2014, ApJ, 780, 151
- Wilson, R. E., Van Hamme, W., & Terrell, D. 2010, ApJ, 723, 1469
- Young, A. T., Milone, E. F., & Stagg, C. R. 1994, A&AS, 105, 259

A. Sample LC Input File

Below is a sample **LC** input file with six concatenated input blocks. The first and second blocks are for light curves in the V and B bands, respectively, with phase as the independent variable for the V curve and time as the independent variable for the B curve. The third block is for a radial velocity curve with phase as the independent variable. The fourth block is for computing line profiles. The fifth block is for output of star figures (relative radii) versus phase. The sixth block is for output of eclipse timing residuals for listed input timings. Not all possible combinations of input data types are shown.

```

1 1 1 1 1 1 1 0 -3 -2 0 2 1 0      0
2 30979.347628 0.1919411642D+01 0.289188D-10 0.0000 0.01156 1 0.0000D+00 1 138472375.
30979.347628 30980.347628 0.100000 -0.200000 1.000000 0.010000 0.250000 0.7500 1 1.0140
2 0 1 1 30 30 2.608469 0.110053D-03 0.00000 100.00
.21859 0.103034D+02 2.9008 1.5924 -0.0990 89.362 1.000 1.000 0.00 1.0000 1.0000
1.0140 0.8487 1.000 1.000 0.659389D+01 0.636603D+01 0.741712D+00 0.538 0.658 0.174 0.142 2.73240
0.000000D+00 0.3278213D+02 76.00000 0.160038 3.6276558 44655.41483100
7 0.107590D-02 0.465577D-03 -0.029 0.669 0.726 0.285 0.1048D-04 0.0000D+00 8.000 1.0000 0.550000 0.0000 0.36895D+00
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
150.
1 1 1 1 1 1 1 0 -3 -2 0 2 1 0      0
1 30979.347628 0.1919411642D+01 0.289188D-10 0.0000 0.01156 1 0.0000D+00 1 138472375.
30979.347628 30980.347628 0.100000 -0.200000 1.000000 0.010000 0.250000 0.7500 1 1.0140
2 0 1 1 30 30 2.608469 0.110053D-03 0.00000 100.00
.21859 0.103034D+02 2.9008 1.5924 -0.0990 89.362 1.000 1.000 0.00 1.0000 1.0000
1.0140 0.8487 1.000 1.000 0.659389D+01 0.636603D+01 0.741712D+00 0.538 0.658 0.174 0.142 2.73240
0.000000D+00 0.3278213D+02 76.00000 0.160038 3.6276558 44655.41483100
6 0.100962D-02 0.378750D-03 -0.029 0.773 0.841 0.315 0.4973D-05 0.0000D+00 8.000 1.0000 0.440000 0.0000 0.62660D+00
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
150.
2 1 1 1 1 1 1 0 -3 -2 0 2 1 0      0
2 30979.347628 0.1919411642D+01 0.289188D-10 0.0000 0.01156 1 0.0000D+00 0 138472375.
30979.347628 30980.347628 0.100000 -0.200000 1.000000 0.010000 0.250000 0.7500 1 1.0140
2 0 1 1 30 30 2.608469 0.110053D-03 0.00000 100.00
.21859 0.103034D+02 2.9008 1.5924 -0.0990 89.362 1.000 1.000 0.00 1.0000 1.0000
1.0140 0.8487 1.000 1.000 0.659389D+01 0.636603D+01 0.741712D+00 0.538 0.658 0.174 0.142 2.73240
0.000000D+00 0.3278213D+02 76.00000 0.160038 3.6276558 44655.41483100
6 0.103500D-02 0.408948D-03 -0.029 0.773 0.841 0.315 0.0000D+00 0.0000D+00 8.000 1.0000 0.440000 0.0000 0.00000D+00
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
150.
3 1 1 1 1 1 1 0 -3 -2 0 2 1 0      0
2 30979.347628 0.1919411642D+01 0.289188D-10 0.0000 0.01156 1 0.0000D+00 1 138472375.
30979.347628 30980.347628 0.100000 -0.000000 0.300000 0.250000 0.250000 0.7500 1 1.0140
2 0 1 1 30 30 2.608469 0.110053D-03 0.00000 100.00
.21859 0.103034D+02 2.9008 1.5924 -0.0990 89.362 1.000 1.000 0.00 1.0000 1.0000
1.0140 0.8487 1.000 1.000 0.659389D+01 0.636603D+01 0.741712D+00 0.538 0.658 0.174 0.142 2.73240
0.000000D+00 0.3278213D+02 76.00000 0.160038 3.6276558 44655.41483100
6 0.100962D-02 0.378750D-03 -0.029 0.773 0.841 0.315 0.4973D-05 0.0000D+00 8.000 1.0000 0.440000 0.0000 0.62660D+00
0.10000d-03 000.9900 -0005.00 03
00.447148 1.50000d-04 +00.80000 +001
00.446990 1.45000d-04 +00.70000 +002
00.448115 1.40000d-04 +00.50000 -001
-1.
0.10000d-04 001.0000 -0000.00 03
00.447100 1.20000d-05 +00.80000 +000
-1.
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
150.
4 1 1 1 1 1 1 0 -3 -2 0 2 1 0      0
2 30979.347628 0.1919411642D+01 0.289188D-10 0.0000 0.01156 1 0.0000D+00 1 138472375.
30979.347628 30980.347628 0.100000 -0.200000 1.000000 0.010000 0.250000 0.7500 1 1.0140
2 0 1 1 30 30 2.608469 0.110053D-03 0.00000 100.00
.21859 0.103034D+02 2.9008 1.5924 -0.0990 89.362 1.000 1.000 0.00 1.0000 1.0000
1.0140 0.8487 1.000 1.000 0.659389D+01 0.636603D+01 0.741712D+00 0.538 0.658 0.174 0.142 2.73240
0.000000D+00 0.3278213D+02 76.00000 0.160038 3.6276558 44655.41483100
6 0.100962D-02 0.378750D-03 -0.029 0.773 0.841 0.315 0.4973D-05 0.0000D+00 8.000 1.0000 0.440000 0.0000 0.62660D+00

```

```

300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
150.
6 1 1 1 1 1 0 -3 -2 0 2 1 0 0
1 30979.347628 0.1919411642D+01 0.289188D-10 0.0000 0.01156 1 0.0000D+00 0 138472375.
30979.347628 30980.347628 0.100000 -0.200000 1.000000 0.010000 0.250000 0.7500 1 1.0140
2 0 1 1 30 30 2.608469 0.110053D-03 0.00000 100.00
.21859 0.103034D+02 2.9008 1.5924 -0.0990 89.362 1.000 1.000 0.00 1.0000 1.0000
1.0140 0.8487 1.000 1.000 0.659389D+01 0.636603D+01 0.741712D+00 0.538 0.658 0.174 0.142 2.73240
0.000000D+00 0.3278213D+02 76.00000 0.160038 3.6276558 44655.41483100
0 1.000000D+00 0.234250D-01 0.000 0.000 0.000 0.000 0.0000D+00 0.0000D+00 8.000 1.0000 0.000000 0.0000 0.00000D+00
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
300.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
150.
13552.17300 2
14702.66100 1
14799.78600 2
...
37175.34300 1
37525.37600 2
41829.90030 1
-10000.
9

```

```

MPAGE,NREF,MREF,IFSMV1,IFSMV2,ICOR1,ICOR2,IF3B,LD1,LD2,KSPEV,KSPOT,NOMAX,IFCGS,KTSTEP
8(I1,1X),2(I2,1X),4(I1,1X),I6
JDPHS,HJD0,PZERO,DPDT,PSHIFT,DELPH,NGA,STDEV,NOISE,SEED
I1,F15.6,D17.10,D14.6,F10.4,F8.5,I3,D11.4,I2,F11.0
HJDST,HJDSP,HJDIN,PHSTRT,PHSTOP,PHIN,PHN,PHOBS,LSP,TOBS
F14.6,F15.6,F13.6,4F12.6,F10.4,I2,F8.4
MODE,IPB,IFAT1,IFAT2,N1,N2,PERR0,DPERDT,THE,VUNIT
4I2,2I4,F13.6,D14.6,F8.5,F8.2
E,A,F1,F2,VGA,XINCL,GR1,GR2,ABUNIN,Fspot1,Fspot2
F6.5,D13.6,2F10.4,F10.4,F9.3,2F7.3,F7.2,2F10.4
TAVH,TAVC,ALB1,ALB2,POTH,POTC,RM,XBOL1,XBOL2,YBOL1,YBOL2,DPCLOG
F7.4,1X,F7.4,2F7.3,3D13.6,4F7.3,F8.5)
A3B,P3B,XINC3B,E3B,PERR3B,TC3B → values have no effect when IF3B=0
D12.6,D14.7,F11.5,F9.6,F10.7,F17.8
IBAND,HL,CL,XH,XC,YH,YC,EL3,OPSF,ZERO,FACTOR,WL,AEXTINC,CALIB
I3,2D15.7,4F7.3,D12.4,D11.4,F8.3,F8.4,F10.6,F8.4,d12.5
BINWM1,SC1,SL1,NF1 → star 1 line profile parameters, for MPAGE=3 only
D11.5,F9.4,F9.2,I3
WLL1,EWID1,DEPTH1,KKS → star 1 line profile parameters, for MPAGE=3 only
F9.6,D12.5,F10.5,I5
BINWM2,SC2,SL2,NF2 → star 2 line profile parameters, for MPAGE=3 only
D11.5,F9.4,F9.2,I3
WLL2,EWID2,DEPTH2,KKS → star 2 line profile parameters, for MPAGE=3 only
F9.6,D12.5,F10.5,I5
XLAT,XLONG,RADSP,TEMSP,TSTART,TMAX1,TMAX2,TFINAL → spot parameters
4F9.5,4F14.5
XCL,YCL,ZCL,RCL,OP1,FCL,EDENS,XMUE,ENCL → cloud parameters
3F9.4,F7.4,D11.4,F9.4,D11.3,F9.4,F7.3
HJDT,MNTYPE → observed or synthetic eclipse timings, for MPAGE=6 and KTSTEP=0 only
F14.5,I6

```

B. Sample DC Input File

The sample input file below has most observations removed and is shown for format only. There are two radial velocity curves, four light curves (in V , B , V , B), and a set of eclipse timings.

```
+0.2d-1 +0.2d-1 +1.0d-3 +0.2d-1 +0.2d-1 +1.0d-3 +0.1d-3 +0.2d-1
+5.0d-2 +1.0d-3 +1.0d-2 +1.0d-2 +1.0d-2 +2.0d-3 +2.0d-1 +1.0d-2 +1.0d-3 +2.0d-2 +2.0d-2
+5.0d-2 +5.0d-2 +2.0d-2 +2.0d-2 +0.3d-2 +1.0d-2 +1.0d-2 +1.0d-2 +1.0d-2
1111 1111 1111110 11111 11000 10111 11111 11111 11111 11111 01110 08 1.000d-05 1.000
2 1 2 2
1 1 04 1 2 0 0 3 1 1 0
1 1 1 1 1 1 0 -3 -2 0 2 1 0 1 0 0.0000
1 30979.347460 0.1919411773D+01 0.183488D-10 0.0000 0.01156 1
2 0 1 1 20 20 10 10 2.587092 0.11091D-03 0.00000 100.000
.21854 0.102911D+02 2.9008 1.5924 -0.1003 89.228 1.000 1.000 0.00 1.0000 1.0000
1.0140 0.8567 1.000 1.000 0.659287D+01 0.633099D+01 0.739553d+00 0.538 0.658 0.174 0.142 2.73240
0.000000D+00 0.3278213D+02 76.00000 0.160038 3.6276558 44655.41483100
6 0.103500D-02 0.204825D+01 -0.029 0.773 0.841 0.315 0.000D+00 0.58000D-01 0.10000 0.40000 0.55000 0.95000 0.440000 1
6 0.103500D-02 0.204825D+01 -0.029 0.773 0.841 0.315 0.000D+00 0.48500D-01 0.10000 0.40000 0.55000 0.95000 0.440000 1
7 0.103018D-02 0.204825D+01 -0.029 0.669 0.726 0.285 0.1421D-04 0.000D+00 1 0.68500D-06 0.12000 0.44000 0.55000 0.95000 1
6 0.963158D-03 0.204825D+01 -0.029 0.773 0.841 0.315 0.8481D-05 0.000D+00 1 0.51500D-06 0.12000 0.44000 0.55000 0.95000 1
7 0.101564D-02 0.204825D+01 -0.029 0.669 0.726 0.285 0.1452D-04 0.000D+00 1 0.78400D-06 0.12000 0.38000 0.55000 0.90000 1
6 0.948209D-03 0.204825D+01 -0.029 0.773 0.841 0.315 0.7946D-05 0.000D+00 1 0.85000D-06 0.12000 0.38000 0.55000 0.90000 1
0.82000D-02 1
0.550000 0.0000 0.1000D+01 0.36895D+00
0.440000 0.0000 0.1000D+01 0.62660D+00
0.550000 0.0000 0.1000D+01 0.36895D+00
0.440000 0.0000 0.1000D+01 0.62660D+00
300.00000
300.00000
150.
45273.57800 -0.978000 1.000 47778.62490 -1.401000 1.000 49486.93560 -1.323000 1.000
47778.66940 -1.349000 1.000 47651.99100 -1.387000 1.000 45868.88710 -1.372000 1.000
49488.97420 -1.063000 1.000 47780.71630 -1.015000 1.000 45575.64880 0.620000 1.000
49485.95110 0.801000 1.000 47777.71750 0.747000 1.000 46249.87520 0.595000 1.000
-10003.00000 0.000000 0.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
45273.57800 1.127000 1.000 47778.62490 1.611000 1.000 49486.93560 1.524000 1.000
47778.66940 1.607000 1.000 47651.99100 1.559000 1.000 45868.88710 1.640000 1.000
49488.97420 1.257000 1.000 47780.71630 1.189000 1.000 45575.64880 -1.191000 1.000
49485.95110 -1.219000 1.000 47777.71750 -1.133000 1.000 46249.87520 -1.147000 1.000
-10003.00000 0.000000 0.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
47378.67510 9.716000 1.000 47378.68835 9.708000 1.000 47378.69434 9.708000 1.000
47378.69930 9.710000 1.000 47378.70311 9.711000 1.000 47378.70693 9.700000 1.000
...
48070.88290 9.694000 1.000 48070.88546 9.691000 1.000 48070.88809 9.700000 1.000
48070.89065 9.688000 1.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
-10001.00000 0.000000 0.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
47378.67552 9.868000 1.000 47378.68877 9.869000 1.000 47378.69474 9.869000 1.000
47378.69971 9.873000 1.000 47378.70352 9.866000 1.000 47378.70735 9.858000 1.000
...
48070.88330 9.855000 1.000 48070.88584 9.858000 1.000 48070.88849 9.848000 1.000
48070.89104 9.856000 1.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
-10001.00000 0.000000 0.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
37548.22870 9.698000 1.000 37548.22980 9.698000 1.000 37548.23570 9.715000 1.000
37548.23670 9.713000 1.000 37548.24000 9.716000 1.000 37548.24090 9.718000 1.000
...
38297.37710 9.711000 1.000 38297.38950 9.710000 1.000 38297.39110 9.721000 1.000
38297.39570 9.712000 1.000 38297.39760 9.721000 1.000 0.000000 0.000000 0.000
-10002.00000 0.000000 0.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
37548.22870 9.865000 1.000 37548.22980 9.869000 1.000 37548.23570 9.890000 1.000
37548.23670 9.879000 1.000 37548.24000 9.874000 1.000 37548.24090 9.878000 1.000
...
38297.37450 9.882000 1.000 38297.37710 9.892000 1.000 38297.38950 9.883000 1.000
38297.39110 9.891000 1.000 38297.39570 9.887000 1.000 38297.39760 9.882000 1.000
-10003.00000 0.000000 0.000 0.000000 0.000000 0.000 0.000000 0.000000 0.000
13552.17300 2 0.50107 14702.66100 1 0.50107 14799.78600 2 0.50107
15758.33900 1 1.00000 15797.86800 2 1.00000 16641.27800 1 1.00000
...
51963.34900 2 9.92250 52678.25600 1 9.92250 52829.88200 1 9.92250
53121.63340 1 248.06250 53123.55210 1 3969.00000 53124.60060 2 3969.00000
51342.36700 1 1.58760 37175.34300 1 5.87130 37525.37600 2 1.96000
41829.90030 1 992.25000 0. 0 0. 0. 0 0.
-10001.
1111 1111 1111110 11111 11001 10111 11111 11111 11111 11111 01110 08 1.000d-05 1.000
2
```



```

(DEL(I),I=1,8)
11(1X,D7.1)
(DEL(I),I=9,14),(DEL(I),I=16,20)
11(1X,D7.1)
(DEL(I),I=21,25),(DEL(I),I=56,59)
11(1X,D7.1)
(KEP(I),I=1,60),NITERS,XLAMDA,VLR
1X,2(4I1,1X),7I1,1X,9(5I1,1X),I2,D10.3,F6.3
KSPA,NSPA,KSPB,NSPB
4I3
IFVC1,IFVC2,NLC,IFTIME,KO,KDISK,ISYM,NPPL,IFDER,IFLCIN,IFOC
I1,1X,I1,1X,9I2
NREF,MREF,IFSMV1,IFSMV2,ICOR1,ICOR2,IF3B,LD1,LD2,KSPEV,KSPOT,NOMAX,IFCGS,MAGLITE,LINKEXT,DESEXTINC
7(I1,1X),2(I2,1X),6(I1,1X),F7.4
JDPHS,HJD0,PZERO,DPDT,PSHIFT,DELPH,NGA
I1,F15.6,D17.10,D14.6,F10.4,F8.5,I3
MODE,IPB,IFAT1,IFAT2,N1,N2,N1L,N2L,PERR0,DPERDT,THE,VUNIT
4I2,4I4,F13.6,D13.5,F8.5,F9.3
E,A,F1,F2,VGA,XINCL,GR1,GR2,ABUNIN,Fspot1,Fspot2
F6.5,D13.6,2F10.4,F10.4,F9.3,2F7.3,F7.2,2F10.4
TAVH,TAVC,ALB1,ALB2,PHSV,PCSV,RM,XBOL1,XBOL2,YBOL1,YBOL2,DPCLOG
F7.4,F8.4,2F7.3,3D13.6,4F7.3,F9.5
A3B,P3B,XINC3B,E3B,PERR3B,TC3B
D12.6,D14.7,F11.5,F9.6,F10.7,F17.8
IBAND,HLA,CLA,X1A,X2A,Y1A,Y2A,OPSFA,SIGMA,SPHAS1,SPHAS2,SPHAS3,SPHAS4,WLA,KSD
I3,2D15.7,4F7.3,D10.3,D12.5,4F8.5,F10.6,I2
IBAND,HLA,CLA,X1A,X2A,Y1A,Y2A,EL3A,OPSFA,NOISE,SIGMA,SPHAS1,SPHAS2,SPHAS3,SPHAS4,KSD
I3,2D15.7,4F7.3,D12.4,D10.3,I2,D12.5,4F8.5,I2
SIGMA,KSD
86X,D11.5,32X,I2
WLA,AEXTINC,XUNIT,CALIB
F9.6,F8.4,D11.4,D12.5
XLAT,XLONG,RADSP,TEMSP,TSTART,TMAX1,TMAX2,TFINAL
4F9.5,4F14.5
XCL,YCL,ZCL,RCL,OP1,FCL,EDENS,XMUE,ENCL
3F9.4,F7.4,D11.4,F9.4,D11.3,F9.4,F7.3
(PHJD(in),FLUX(in),WT(in),in=ifirst,last)
5(F14.5,F11.6,F8.3)
(PHJD(in),MNTYPE(in),WT(in),in=ifirst,last)
5(F14.5,I6,F13.3)
(KEP(I),I=1,60),NITERS,XLAMDA,VLR
1X,2(4I1,1X),7I1,1X,9(5I1,1X),I2,D10.3,F6.3

```