2(M1X)

UTIS HC $6.75

X-110-72-473
PREPRINT

NASA TM X- 66155

A COMPUTER PROGRAM FOR MODELING NON-SPHERICAL ECLIPSING BINARY STAR SYSTEMS

D. B. WOOD

DECEMBER 1972

GSFC
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

(NASA-TM-X-66155) A COMPUTER PROGRAM FOR MODELING NON-SPHERICAL ECLIPSING BINARY STAR SYSTEMS (NASA) 94 p HC $6.75 CSCL 03A
N73-16821
Unclas
G3/30 53344

X-110-72-473

A Computer Program

for

Modeling Non-Spherical Eclipsing

Binary Star Systems

D. B. Wood
Advanced Plans Staff

December 1972

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

A COMPUTER PROGRAM FOR MODELING NON-SPHERICAL ECLIPSING BINARY STAR SYSTEMS

D. B. Wood
Advanced Plans Staff

ABSTRACT

The accurate analysis of eclipsing binary light curves is fundamental to obtaining information on the physical properties of stars. Until recently, the analysis of these stars has been performed using the relatively simple "spherical model." This model, however, does not account for a number of complications which we know to exist.

The model described in this document accounts for the important geometric and photometric distortions such as rotational and tidal distortion, gravity brightening, and reflection effect. This permits a more accurate analysis of interacting eclipsing star systems.

The model is designed to be useful to anyone with "moderate" computing resources. The programs, written in FORTRAN IV for the IBM 360 , consume about 80k bytes of core. On the 360/75, about 93 milliseconds (ms) is required to predict a point on the light curve to an accuracy of ± 0.0002 magnitude. Solution for 10 unknowns, using 100 normal points, requires about 65 seconds per iteration.

The FORTRAN program listings are provided, and the computational aspects are described in some detail. Implementation of the programs as they are presently written should present no problems. In addition, it should be fairly straightforward to modify the programs to suit the users' individual requirements.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

A COMPUTER PROGRAM FOR MODELING NON-SPHERICAL ECLIPSING BINARY STAR SYSTEMS

I. INTRODUCTION

Most of our knowledge about the physical properties of stars comes from the analysis of eclipsing binary stars. We observe the nature of the change of light intensity with time, and "solve" this light curve to determine such physical quantities as mass, radius, surface temperature, luminosity, density gradient, variation of intensity across the surface, etc. Until recently, the analysis of light curves has been done via the "spherical model," wherein the stars are represented as non-interacting spheres. Any deviations from this simple picture were "rectified" out of the light curve.

In recent literature, considerable emphasis has been placed on the advantage of analyzing eclipsing binary systems with models which are more realistic than the traditional "spherical model." Some of these new models are quite general (Hill and Hutchings 1970; Wilson and Devinney 1971), whereas others are more specific (e.g. Lucy 1968; Mochnacki and Doughty 1972; Horak 1966; Rucinski 1969). Unfortunately, however, these new models have not been made generally available to observers for the analysis of their data.

The intent of this publication is to provide a relatively simple, inexpensive (in terms of computer use) modern model which anyone can use to analyze an observed light curve and arrive at physical parameters of the stars. Various aspects of this model have already been described (Wood 1969, 1971a, 1971b, 1971c, 1971d). In order to provide a complete description, much of the material in these earlier publications will be repeated here.

The model is programmed in the FORTRAN IV language for the IBM 360. Most calculations are in the single precision. The programs occupy about 80k bytes of core and require an average of about 93 ms on the 360/75 to calculate one point on a light curve to a precision of ± 0.0002 magnitudes (73 ms out of eclipse; 110 ms for partial or annular; 37 ms for total). The deletion of reflection effect from the calculation decreases the compute time by about 40%. Solution for 10 unknowns using 100 normal points requires about 65 seconds per iteration. If convergence does not occur by about 4 iterations, it is not likely to occur at all with that parameter set. Almost one third of the compute time is devoted to the extraction of square roots.

This model takes into account the best understood geometric and photometric distortions, including rotational and tidal distortion, limb darkening, gravity brightening, and reflection. The major simplifications are 1) each star is treated as a triaxial ellipsoid, 2) reflection effect is approximated, and 3) the stars are considered to rotate in their orbital plane with a period equal to the orbital period. The first simplification is the main detail which separates this model from other non-spherical models, and permits the relatively fast compute time. This approximation is probably quite valid except for extremely close systems, since the limiting potential (Roche) surface would define the extent of the stars' outer atmosphere, not the photosphere. Geometrically, the reflection is very good. Astrophysically, it is consistent with gray atmosphere approximation. As we will see later, the constraint on rotation can be relaxed.

In this publication we will first describe the model and its parameters. Then the computational technique will be outlined. A description of the important features of each computer subprogram follows. Finally, the details of running the program are described

II. THE MODEL

A. Physical Description

Basically, the model is defined by the following parameters:

Orbital parameters

Period of revolution	P
Time of conjunction	T_c
Semimajor axis or orbit	R_o
Orbital eccentricity	e
Longitude of periastron	ω
Inclination	i

Geometric parameters

Semiaxes of stars A and B	$a_A, b_A, c_A,$ a_B, b_B, c_B

Photometric parameters

Surface intensity (see text)	$\bar{I}_A, \bar{I}_B$
Limb darkening coefficient	u_A, u_B

Gravity brightening coefficient	v_A, v_B
Reflection coefficient	w_A, w_B

The two stars are not identified in the usual way as "larger" and "smaller" but rather as star "A" and star "B". Star A is eclipsed at the deeper eclipse and is considered to be the central star about which B revolves.

It is convenient to replace the six geometric parameters (the stellar semi-axes) with six dimensionless parameters:

$$a, k_a, \epsilon_A, \epsilon_B, \zeta_A, \zeta_B$$

which are related to the actual axes by the following:

$$\left.\begin{aligned} a_A &= aR_o \\ a_B &= k_a a R_o \\ b_A &= \epsilon_A a R_o \\ b_B &= \epsilon_B k_a a R_o \\ c_A &= (1+\zeta_A)\epsilon^2{}_A a R_o \\ c_B &= (1+\zeta_B)\epsilon^2{}_B k_a a R_o \end{aligned}\right\} \tag{1}$$

Note that primary eclipse is an occultation if $k_a < 1$ and a transit if $k_a > 1$. The semi-major axis of the orbit, R_o, is the unit of length, and is usually set to unity.

From the work of Chandrasekhar (1933) the six axes of the stars can be expressed as functions of the mass ratio and polytropic indices. If we retain terms only to the third order in stellar radius divided by separation, then the stars are triaxial ellipsoids with axes given by:

$$\left.\begin{aligned} a_A &= \nu_A \left[1 + \tfrac{1}{6}(1+7q)\Delta_{2A}\nu_A{}^3\right] \\ b_A &= \nu_A \left[1 + \tfrac{1}{6}(1-2q)\Delta_{2A}\nu_A{}^3\right] \\ c_A &= \nu_A \left[1 - \tfrac{1}{6}(2+5q)\Delta_{2A}\nu_A{}^3\right] \end{aligned}\right\} \tag{2}$$

for star A. Here q is the mass ratio (star B to star A) and ν_A is given by:

$$\nu_A = a_{oA}/R_o \tag{3}$$

where a_{oA} is the "unperturbed radius" of star A; i. e. the radius of a sphere of equivalent volume. The parameter Δ_{2A} is a slowly varying function of the polytropic index, n, and assomptotically approaches 1 as n approaches 5.

For star B we have, similarly,

$$\left.\begin{aligned} a_B &= \nu_B \left[1 + \tfrac{1}{6}(1 + 7/q)\Delta_{2B}\nu_B{}^3\right] \\ b_B &= \nu_B \left[1 + \tfrac{1}{6}(1 - 2/q)\Delta_{2B}\nu_B{}^3\right] \\ c_B &= \nu_B \left[1 - \tfrac{1}{6}(2 + 5/q)\Delta_{2B}\nu_B{}^3\right] \end{aligned}\right\} \tag{4}$$

where:

$$\nu_B = a_{oB}/R_o = a_{oA}k_\nu/R_o \tag{5}$$

The six stellar axes or the six dimensionless geometric parameters may thus be expressed as functions of:

$$a_{oA},\ k_\nu,\ q,\ n$$

where we here assume, for convenience, that one n applies to both stars.

Limb darkening is expressed by the usual linear law

$$I = I_o(I - u + u \cos \gamma) \tag{6}$$

where γ is the foreshortening angle. I_o is the intensity of the "sub-earth" point, where the line of sight from the observer is normal to the stellar surface and $\gamma = 0$. Arbitrarily, the intensity ratio is defined at time $T_Q = T_c + P/4$ as

$$j = \bar{I}_B/\bar{I}_A \tag{7}$$

where $\bar{I}$ is the value of I_o at T_Q.

If the observations are taken over a narrow wavelength region, then we can write:

$$j \simeq \frac{\exp(c/T_A) - 1}{\exp(c/T_B) - 1} \tag{8}$$

where T_A, T_B are the effective surface temperatures at the sub-earth point at T_Q. Here $C = hc/k\lambda = 1.43879/\lambda$ where λ is the wavelength of observation. For a bolometric light curve:

$$j = (T_B/T_A)^4 \tag{9}$$

The gravity brightening coefficient v, is defined to be used analogously to limb darkening so that

$$I_o = \bar{I}\left[1 - v + v\,(r/\bar{r})\right] \tag{10}$$

where r is the local radius and $\bar{r}$ is the radius to the sub-earth point at T_Q, where $\bar{I}$ is defined.

If the local effective temperature, T, varies as local gravity to some power β, then we may write v in terms of β:

$$v = \left(\frac{\Delta_2 - 5}{\Delta_2}\right)\frac{\beta C}{T}\left(\frac{e^{c/T}}{e^{c/T} - 1}\right) \tag{11}$$

where C = 1.43879/ λ.

Reflection is treated by determining the local incident intensity, L*, at any point and reflecting a fraction, w, of this uniformly over the out-going hemisphere. Thus the local emergent intensity, I_o, has added to it the intensity I*, given by:

$$I^* = wL^*/2\pi(1 - u/2). \tag{12}$$

The incident radiation L*, can be calculated in detail by integrating over the source star as described by Chen and Rhein (1969) and by Wood (1971b, 1971d). The temperature distributions produced in this manner agree quite closely between Chen and Rhein, Wood, and Napier (1968). Unfortunately, such an exact treatment is prohibitively time consuming. Over the range of interest (a_A and $a_B < 0.5$) a fairly good approximation is possible which avoids the necessity of integration (Wood 1972). See Appendix II.

B. Perturbations on Basic Model

1. Extended Atmospheres

Normally, the stars are considered to have very small scale heights, so that they can be considered as having sharp edges. If the scale height becomes appreciable (greater than about 10^{-3} of the stellar radius) then the star must be considered to have an extended atmosphere. A light ray passing through this atmosphere will be attenuated according to:

$$j = j_o e^{-\tau} \tag{13}$$

where, for a simple exponential atmosphere, the optical depth, τ, is given by:

$$\tau = \tau_0 e^{-r/h} \tag{14}$$

τ_0 is the optical depth at the "surface," where a ray grazes the limb. The apparent distance of a ray above the limb is given by r, and h is the scale height. The effect of such an extended atmosphere has been described by Wood (1971b, 1971e).

2. Orbital Skew

The basic model assumes that the stars rotate in the orbital plane with the major axes aligned on the apse. Deviations to this geometry can be introduced; however, they must be constant over the entire orbital period.

The star's major axis may be made to lag or lead periastron passage by an amount σ. The star's pole may be tilted from the orbital pole by an amount ι.

C. System Luminosity

The total observed luminosity of a star is given by an integral over the apparent ellipse:

$$L = \iint (I_0 + I^*)(I - u + u \cos \gamma) dA \tag{15}$$

The total system luminosity at any time t is

$$L_{TOT}(t) = L_A(t) + L_B(t) - L_{ECL}(t) \tag{16}$$

The system luminosity is normalized at $t = T_Q$:

$$L_{TOT}(T_Q) = L_A(T_Q) + L_B(T_Q) = 1$$

III. OUTLINE OF COMPUTATIONAL TECHNIQUE

A. General

Calculation of a light curve is just the evaluation of equation (16) for various values of t. This involves, generally, three double integrations of equation (15); one over star A, one over star B, and one over the overlapping area.

To prepare for these integrations, the computer programs must perform two fundamental tasks: 1) determine the outlines of the stars and their overlapped area as projected on the plane of the sky; and 2) determine the light intensity at any point on the apparent stars.

The solution of a light curve, that is going from an observed light curve to orbital elements, uses least squares differential corrections. The light curve intensity is a function of time and some number, n, of parameters:

$$\text{Intensity} = I(t, X_1, X_2, X_3, \ldots X_n).$$

Taking a linearized Taylor expansion around approximate values of X_i, we may write:

$$I_{OBS} - I_{COMP} = \Delta I = \frac{\partial I}{\partial x_1}\,\delta x_1 + \ldots \frac{\partial I}{\partial x_n}\,\delta x_n = \sum_{i=1}^{n} \frac{\partial I}{\partial x_i}\,\delta x_i \tag{17}$$

I_{OBS} is the observed intensity at t and I_{COMP} is the intensity computed by the model at time t_o for the approximate values of x_i. δx_i is then the difference $x_{TRUE} - x_{APPROX}$, and is a differential correction to the approximate x_i. We have a different equation (17) for each observed time, t, and can solve them by least squares to get the best δx_i. Hopefully, the process can be repeated to converge upon the best set of X_i which most nearly represent the light curve.

Convergence, especially in the presence of observational error, is very hard to accomplish. Exactly which parameters to allow as variables, and what initial values to assign, is still a black art. Completely automated solution is not generally possible.

Each partial derivitive in equation (17) is calculated numerically by varying x_i to determine its effect on I. Fortunately some shortcuts are available, for potentially, to solve a light curve of m observations, we could have to evaluate the system intensity m (2n + 1) times per iteration.

B. Coordinate Systems

The fundamental coordinate system used in most calculations is a rectangular system centered on star A. The x-axis points along the line of sight; the plane of the sky is the y-z plane, with z directed northward. This is the $(x, y, z)_A$ system. Some calculations, for star B, are performed in a similar $(x, y, z)_B$ system, centered on star B.

Each star contains a coordinate system imbedded within it with the x', y', z' axes coincident with the principle axes of the stars; the $(x', y', z')_A$ and

$(x', y', z')_B$ systems. Except for reflection, there is octant symmetry, so the relative sense of $(x', y', z')_A$ and $(x', y', z')_B$ does not matter.

The apparent stars have principle axes which are rotated from the (y, z) axes, so each star has its own coordinate system which is coincident with the principle axes of the apparent ellipse: $(\bar{y}, \bar{z})_A$ and $(\bar{y}, \bar{z})_B$.

C. Model Parameters

The model can be considered as having parameters defined either as "model space" or "astrophysics space." These are shown in Table I. All of the parameters have been previously described.

A light curve may be generated by inputting either model or astrophysical parameters. The solution of a given light curve can be done only in astrophysics space.

Table I. Model Parameters

Astrophysics Space	Model Space
Orbital Elements	
$e \sin \omega$, $e \cos \omega$, P, T_C, R_o, i	$e \sin \omega$, $e \cos \omega$, P, T_C, R_o, i
Geometric Elements	
a_{oA}, k_ν, q, n	a, k_a, ϵ_A, ϵ_B, ζ_A, ζ_B
Photometric Elements	
T_A, T_B, β_A, β_B u_A, u_B, w_A, w_B	j, v_A, v_B u_A, u_B, w_A, w_B
Atmospheric Parameters	
h_A, h_B, τ_{oA}, τ_{oB}	h_A, h_B, τ_{oA}, τ_{oB}
Orbital Perturbation Parameters	
σ_A, σ_B, ι_A, ι_B	σ_A, σ_B, ι_A, ι_B

IV. THE COMPUTER PROGRAM

The computer program is subdivided into a number of subroutines and functions. Some parameters are passed through CALL statements, but most are contained in COMMON blocks. Figure 1 shows the subprogram interrelationships. Common variables are defined in Tables II, III and IV. Appendix III contains program listings, written in FORTRAN-IV for the IBM S-360. In the following paragraphs the general functioning of each subprogram will be described.

A. Main Program (WINK)

The main program, WINK, performs all initializations, input, and ordinary output. Coarse logic flow is shown in Table V. As part of the initialization process, WINK uses the following three subroutines:

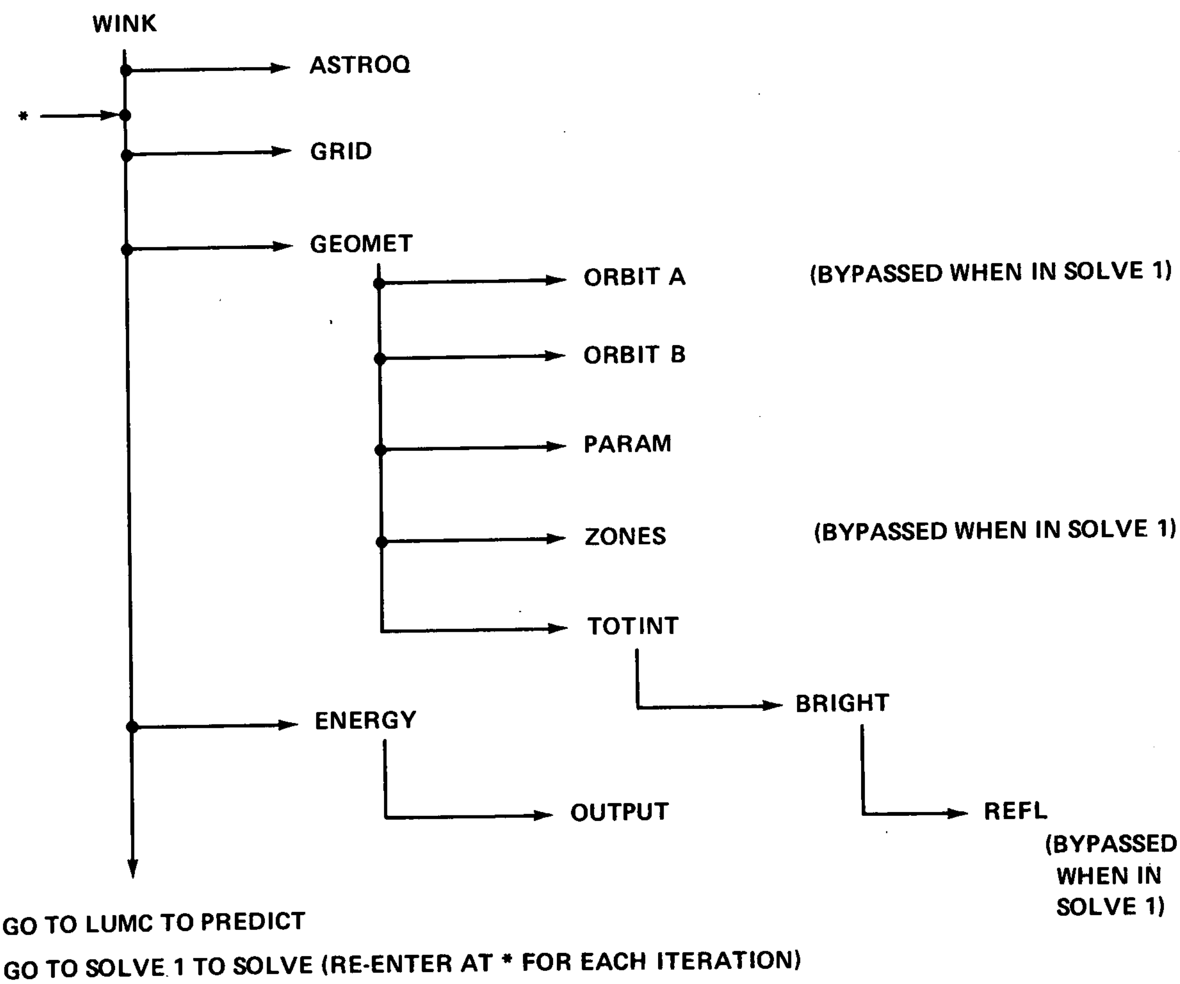


Figure 1a. Sub-program Interrelationships (Initialization/Normalization).

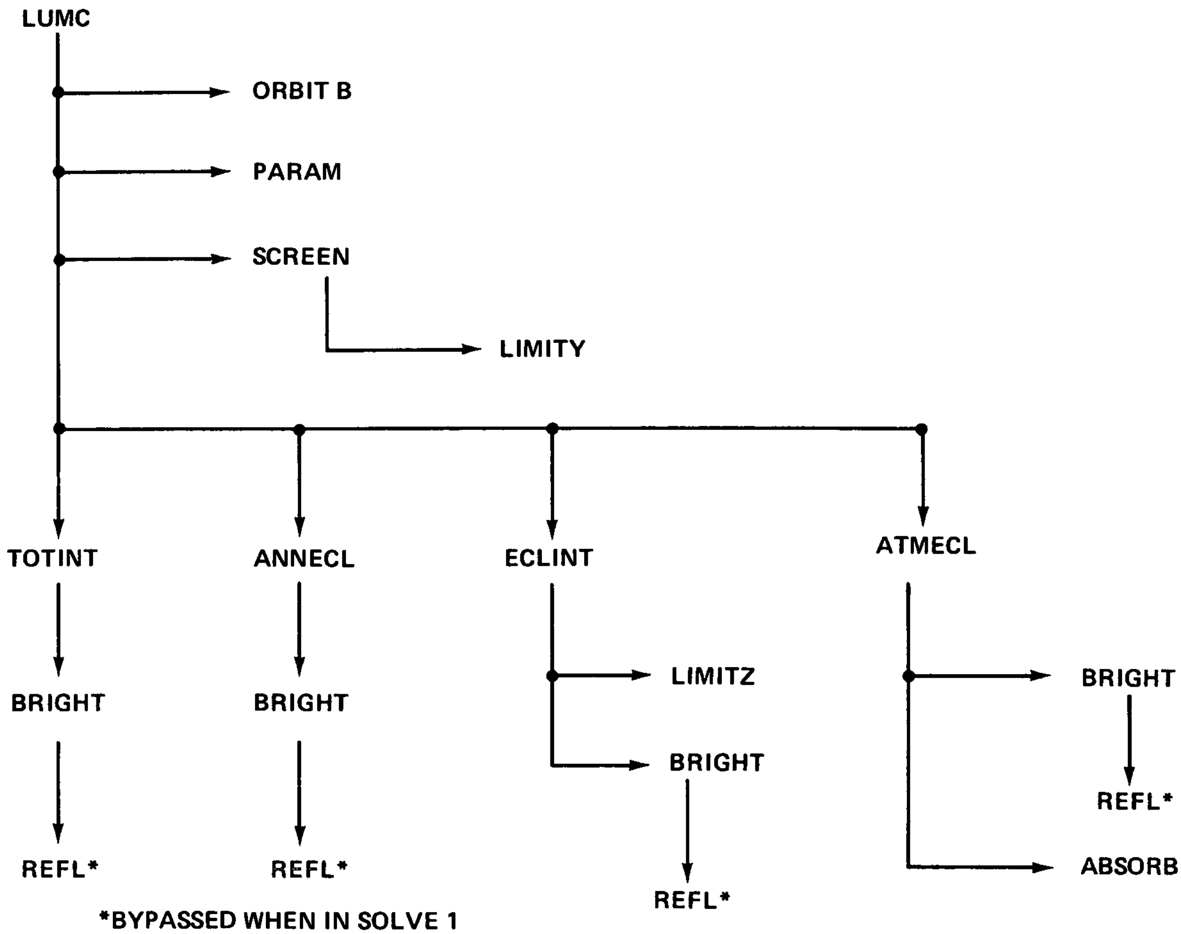


Figure 1b. Sub-program Interrelationships (Light-Prediction Function LUMC).

GRID	-	to set up the Gauss integration coefficients and weights
GEOMET	-	to set all time-independent system variables
ASTROQ	-	to convert astrophysical parameters to model parameters

The system luminosity at any specified time is calculated by the function LUMC. Thus, in LUMC are contained all the computations of equations (16) and (17). The total stellar energy radiated over 4π steradians is calculated by ENERGY. The calculation is approximate, and is for display purposes only. Light curve solution is done through the differential-corrector program SOLVE1.

B. Gauss Quadrature Constants (GRID)

GRID loads COMMON/GAUSS/ with weights, coefficients and $(1\text{-coefficients}^2)^{1/2}$ for the accuracy specified (4-point, 6-point or 16-point integration).

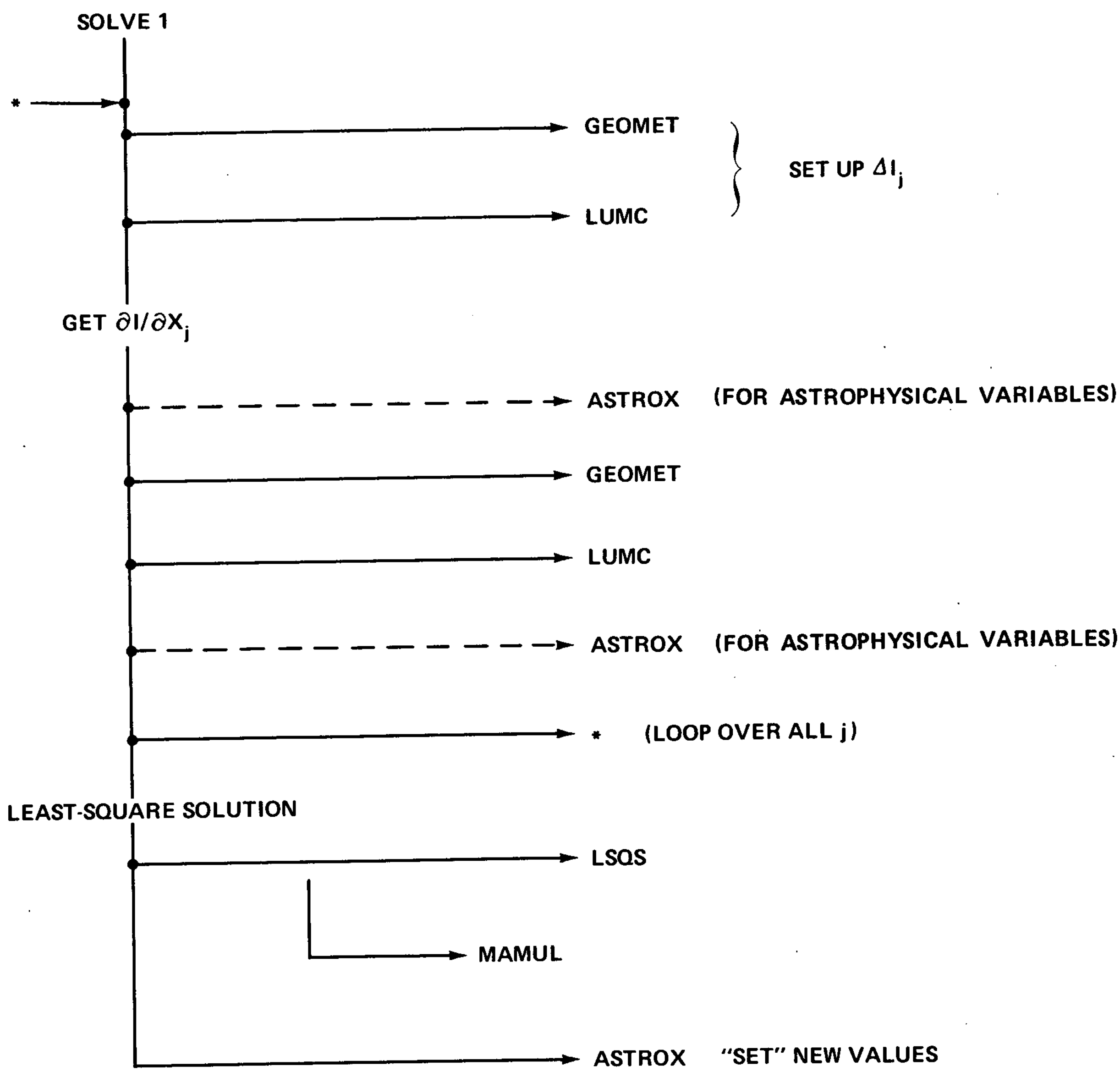


Figure 1c. Sub-program Interrelationships (Differential Corrector SOLVE1).

In addition, COMMON/GAUXX/ is always loaded for 16-point integration to be used by ENERGY and ATMECL. The approximate errors in luminosity are 0.6% for 4-point; 0.19% for 6-point; 0.012% for 16-point. The computing time required goes roughly as 0.6:1:5.1.

C. Generation of Model Parameters from Astrophysical Parameters (ASTROQ and ASTROX)

The subroutine ASTROQ is used to generate model parameters from astrophysical parameters, using equations (2), (4), (8) and (11). It is programmed

Table II. Input Parameter Codes

Code	FORTRAN Variable	Use	Default Code
1	AINCL	inclination in degrees	90.0
2	ESINW	e sin ω	0
3	ECOSW	e cos ω	0
4	UA	limb darkening star A	0.6
5	UB	limb darkening star B	0.6
6	A	semi-major axis star A (model); radius unperturbed sphere of star A (ap.)	0.25
7	RATIOK	ratio of star B to star A; semi-axes (model) or radii (ap.)	1.0
8	ELLIP	ellipticity star A (model); gravity exponent star A (ap.)	0.25
9	ELLIPB	ellipticity star B (model); gravity exponent star B (ap.)	0.25
10	EPSI	differential ellipticity star A (model); surface temperature star A (ap.)	10000.0
11	EPSIB	differential ellipticity star B (model); surface temperature star B (ap.)	10000.0
12	TCONJ	time of conjunction	0
13	VA	gravity bright. coefficient star A (model); mass ratio (ap.)	1.0
14	VB	gravity bright. coefficient star B (model); not used for input (ap.)	0
15	RATIOJ	relative surface brightness (model) not used for input	0
16	WA	reflection albedo star A	0
17	WB	reflection albedo star B	0
18	PERIOD	period	1.0
23	STAR3	third star light (intensity)	0
24	QUAD	quadrature magnitude	0
25	RNOT	orbital semi-major axis length	1.0
26	WAVE	wavelength of observations	5500.0
27	POLYX	polytropic index (if = 0 interpret input as 'model', not 'ap')	5.0
31	SCALE(1)	scale height of atmos. star A	0
32	SCALE(2)	scale height of atmos. star B	0
33	SURF(1)	optical depth at edge star A	0
34	SURF(2)	optical depth at edge star B	0

Table II. Input Parameter Codes (Continued)

Code	FORTRAN Variable	Use	Default Code
35	SLIP(1)	phase shift star A (radians)	0
36	SLIP(2)	phase shift star B (radians)	0
37	TILT(1)	tilt of pole star A (radians)	0
38	TILT(2)	tilt of pole star B (radians)	0

Table III. Other Input Codes

Code	FORTRAN Variable	Use	Default Code
19	NTEG	Integration precision code 1.--4 X 4 integration 2.--6 X 6 integration 3.--16 X 16 integration	1.
20	S	start time for light curve prediction	0
21	TINT	time interval for light curve prediction (0 to read time cards)	0. 1
22	END	end time for light curve prediction	0. 5
84	IFFY	maximum number of iterations	6.
86	ICRD	for punched output of predicted curve 0--no punched output 1.--punched output in mag. 2.--punched output in lum.	0
87	ILUM	interpret input light curve as lum 0--magnitude 1.--luminosity	0
88	JUMP	Flip/Flop for printout of orbital elements (no data field)	print
99	---	Stop	
0	---	begin light curve prediction	
-1	---	begin light curve solution read obs.	
-2	---	begin light curve solution, use old observations	
		(NOTE. For 0, -1, -2, the data field, if non-zero, specifies the quadrature magnitude.)	

Table IV. Other Important Variables

FORTRAN Name	Use
INCL	i; orbital inclination in radians
E	e; orbital eccentricity
T0	T_0; time of periastron passage
Q	q; $(1 - e^2)^{1/2}$
MU	μ; mean daily motion (type REAL)
ALPHA	α; angle, in plane of sky between semi-major axis of one apparent ellipse and the center of the other
THETA	θ; orbital longitude
CHI	χ; angle, in plane of sky, between z-axis of $(x, y, z)_A$ system and center of star B
R	orbital radius
DELTA	δ; apparent separation of centers of stars
PARA	constants which define the ellipsoids
AAXIS	semi-major axis of apparent elipses
BAXIS	semi-minor axis of apparent ellipses
PHASE	orbital phase
PHI	ϕ; angle, in plane of sky, between axes of θ;apparent ellipse and (y, z) axes
THETAP	θ'; orbital longitude of each star
QINT	surface intensity at T_Q
AA	a-axes of ellipsoids
BB	b-axes of ellipsoids
CC	c-axes of ellipsoids
RBAR	$\bar{r}$; radius of ellipsoid in line of sight direction at T_Q
UMA	1 - u
VMA	1 - v
WMA	$w/2\pi(1 - u/2)$
BOLOJ	bolometric intensity ratio
TOTAL	system luminosity at T_Q
STAR 1	luminosity of star A at T_Q
STAR2	luminosity of star B at T_Q
IFSPH	set to 1 if star A spherical, 2 if star B spherical, 3 if both spherical, 0 if none spherical
JTYPE	set to 1 if annular eclipse, 2 if total, 3 if partial, 4 if atmospheric
KSTAR	set to 1 if star A eclipsed, 2 if star B eclipsed, 3 if no eclipse

Table IV. Other Important Variables (Continued)

FORTRAN Name	Use
TEST	logical variable identifying which parameters are "variable"
NOBS	number of observational data cards read
NZONE	for detailed reflection, contains number of zones per octant - set to 0 for shortcut reflection
MREF	set to 1 when integrating total luminosity of star A, 2 for star B, and 3 when integrating partial eclipse
NREF	counter which keeps track of point in integration grid - used for reflection
IREF	set to 2 when calculating partial derivitives, otherwise set to 1
NINC	set to 3 when calculating normalization, otherwise zero
LST	indicates variable for which partial derivitive is currently being calculated
SINI	$\sin i$
COSI	$\cos i$
SISQ	$\sin^2 i$
CISQ	$\cos^2 i$
SINJ	$\sin (i + \iota)$
COSJ	$\cos (i + \iota)$
SINT	sin orbital longitude of each star (θ')
COST	cos orbital longitude of each star (θ')
COTH	cos (true anomaly - mean anomaly) for each star (Θ)
SINP	$\sin \phi$
COSP	$\cos \phi$
DCHI	$\delta \sin \chi$
ECHI	$\delta \cos \chi$
WT	Gauss weights, W, in COMMON/GAUSS/ also observational weights, in COMMON/OBS/
X	Gauss coordinates, X
XC	$(1 - X^2)^{1/2}$
L	the number of Gauss points
TIME	times - observed or internally-generated
LUM	observed luminosity (type REAL)
CLUM	computed luminosities (also used internally in LSQS)

Table V. Main Program Logic Flow

Statement Number	Sub Program	Computational Milestone
		Initializations
66		Input parameters
6	ASTROQ	Convert astrophysical input to model parameters. Decide if prediction or solution. Prediction: generate times (calculate or read) and go to 52
200		Solution: read observations
203		Read card to see which parameters are variables
204		Generate normalized intensities and go to 52
52	GRID	Set up Gaussian integration constants. Set IREF flag (1 means not calculating partial derivatives)
	GEOMET	Compute time-independent quantities; including normalization
	ENERGY	Approximate total stellar luminosity. Set up output quantities.
105		Output model parameters. If solution mode, go to 150.
	LUMC	If prediction, generate light curve. Punch output if requested. Go to 66.
150	SOLVE1	Solution: perform one iteration. Go to 52 for next iteration, or to 66 if through iterating.

to accept integral polytropic indices from 1 to 5. A polytropic index of zero is used as a flag to indicate that astrophysical variables are not being used. A special entry, ASTROX, is used for re-entry during the evaluation of partial derivatives.

D. Time-Independent Ellipsoid Calculations (GEOMET)

GEOMET computes the star axes from the geometric parameters. See Table VI. The axes are screened for fatal errors (semi-axes $\leqslant 0$). A call to subroutine ORBITA establishes the orbital eccentricity, time of periastron and mean daily motion.

Table VI. GEOMET Logic Flow

Statement Number	Sub Program	Computational Milestone
		If calculating partial derivitives (IREF = 2) for photometric variables (LST < 11), bypass much of calculation--go to 5.
52		Calculate ellipsoid axes and abort if $\leqslant 0$. Also abort if inclination or period $\leqslant 0$. Calculate sine and cosine functions of inclination (orbital and individual stars). Set flag for possible short cuts if one or both stars spherical.
	ORBITA	Set up basic orbital parameters Define luminosity of star A as 1 Define luminosity of star B as j
5		Set up functions of u, v, and w
	ORBITB	Orbital mechanical solution for quadrature time Calculate $\bar{r}$
6	PARAM	Calculate ellipsoid parameters at quadrature. Set flag NINC for "normalization" (this is used in BRIGHT in conjunction with reflection)
	ZONES	Initialize reflection only if not calculating partial derivitives.
	TOTINT	Get total light of stars at quadrature. Set system normalization value, TOTAL. Set flag NINC for normal operation.

To prepare for intensity normalization, the system luminosity at time = T_Q must be calculated. The two subroutines used are:

ORBITB calculates time-dependent orbital quantities
PARAM calculates time-dependent ellipsoid quantities

Reflection is initialized through a call to ZONES, but only when partial derivations are not being calculated. The system luminosity for normalization is then calculated by obtaining the total light of each star, through two calls to TOTINT.

Note that in the case where partial derivatives are being calculated for variables which do not alter systems geometry, a large portion of GOEMET is bypassed.

E. Calculation of Luminosity (LUMC)

LUMC serves primarily as a dispatcher, to arrange all necessary time-dependent calculations, to determine the nature of the eclipse, if any, and to call the appropriate integration routines. See Table VII. Thus, LUMC calls for the following subroutines:

ORBITB	calculates time-dependent orbital quantities
PARAM	calculates time-dependent ellipsoid quantities
SCREEN	determines nature of eclipse (partial, total, annular, atmospheric; primary, secondary)

The integrations are performed by one of the following functions:

TOTINT	for total uneclipsed light of one star
ANNECL	for annular eclipse light loss
ECLINT	for partial eclipse light loss
ATMECL	for atmospheric eclipses (replaces TOTINT for total light).

F. Time-Independent Orbital Parameters (ORBITA)

ORBITA loads COMMON/ORBIT/ with the basic fixed quantities eccentricity e; time of periastron T_0; mean daily motion μ; and $q = (1 - e^2)^{1/2}$.

G. Time-Dependent Orbital Parameters (ORBITB)

ORBITB performs the orbital mechanics and sets up a number of necessary angles. The following important calculations are performed using the equations shown in Appendix I:

1. Solution of Kepler's equation
2. Set radius vector, scaled by R_0 (R)*

*Normally R_0, the system unit of length, is unity. However, if a stellar axis is extremely small, R_0 may have to be greater than one to preserve computational accuracy.

Table VII. LUMC Logic Flow

Statement Number	Sub Program	Computational Milestone
	ORBIT B	Calculate orbital mechanical quantities at time t
	PARAM	Calculate ellipsoid quantities at time t
	SCREEN	Determine nature of eclipse, if any
10	TOTINT	If no eclipse, get total light of each star; go to 50
25	TOTINT	If total eclipse, total light is just star in front; go to 50
35	ANNECL	If annular eclipse, light loss given by ANNECL; go to 36
45	ECLINT	If partial eclipse, light loss given by ECLINT; go to 36
36		Test scale height of atmosphere
39	TOTINT	If no atmosphere, star behind calculated by TOTINT; go to 38
37	ATMECL	If atmosphere, star behind calculated by ATMECL; go to 38
38	TOTINT	Total light is sum of each star less eclipse; go to 50
50		For final light, add 3rd star and normalize

3. Set orbital longitude (θ)
4. Set apparent separation of centers (δ)
5. Set the (y, z) projections of apparent separation ($\delta \sin \chi$; $\delta \cos \chi$)
6. Set sin and cos of rotation angle of stars (θ') (orbital longitude plus any phase shift).

H. Time-Dependent Ellipsoid Calculations (PARAM)

PARAM determines all the necessary ellipsoid parameters and puts them into the matrix PARA in COMMON/TVARS/. In addition, the following important operations are performed:

1. Establish semi-axes of apparent ellipses
2. Set sin and cos of rotation of ($\bar{y}, \bar{z}$) axes from (y, z) axes (ϕ)
3. Set angle from y axis of star A to center of star B(α).

The equations used are shown in Appendix I.

I. Identification of Eclipse (SCREEN)

SCREEN sets two important flags in COMMON/FLAGS/. KTYPE tells if there is an eclipse and of which star; JTYPE tells the type of eclipse (annular, total, partial, atmospheric).

The subroutine LIMITY is used for fine screening once it is determined that the stars are potentially close to eclipse (when the apparent separation of centers is less than the sum of radius vectors). If the eclipse is partial, LIMITY returns the y-limits of the overlapping area, which SCREEN in turn returns to LUMC.

J. Determination of Geometry of Overlapping Stars (LIMITY)

LIMITY is logically the most complex subroutine. It is called upon when there is reason to believe there is an eclipse.

In general, two ellipses may intersect in as many as four points. This subroutine uses a vertical "scan wire" (a line y = constant) to sample the nature of the intersections of the two apparent stars. Refer to Figures 2 and 3. Only if the "scan wire" finds two roots for each ellipse (four total roots) is it possible to have an eclipse. For some y = constant, let the four roots be Z_{A1}, Z_{A2} (for star A) and Z_{B1}, Z_{B2} (for star B). The way in which these roots interleave determines if there is an eclipse, and if it is partial. If Z_{A1} and Z_{A2} are both always greater (or less) than Z_{B1} and Z_{B2}, then there is no eclipse (Figure 2, case III). For a partial eclipse, at least part of the time the four roots must alternate; e.g. $Z_{A1}>Z_{B1}>Z_{A2}>Z_{B2}$. The scan wire is stepped from right to left to determine a change in the interleaving of the four roots or a change in the number of roots. When a change occurs, the scan is reversed at 1/10 of the step size. This is repeated to determine the y-coordinate of the limits of integration to about 10^{-8} of the semi-major axis.

Certain special situations are sought for:

1. If the eclipse is potentially shallow, a smaller step size and fewer scan reversals are used.

2. If the potentially shallow eclipse is to the left hand end of the star (-y axis), the negative limit is sought first.

3. If partial derivatives of photometric parameters are being calculated, the previously-determined limits are used.

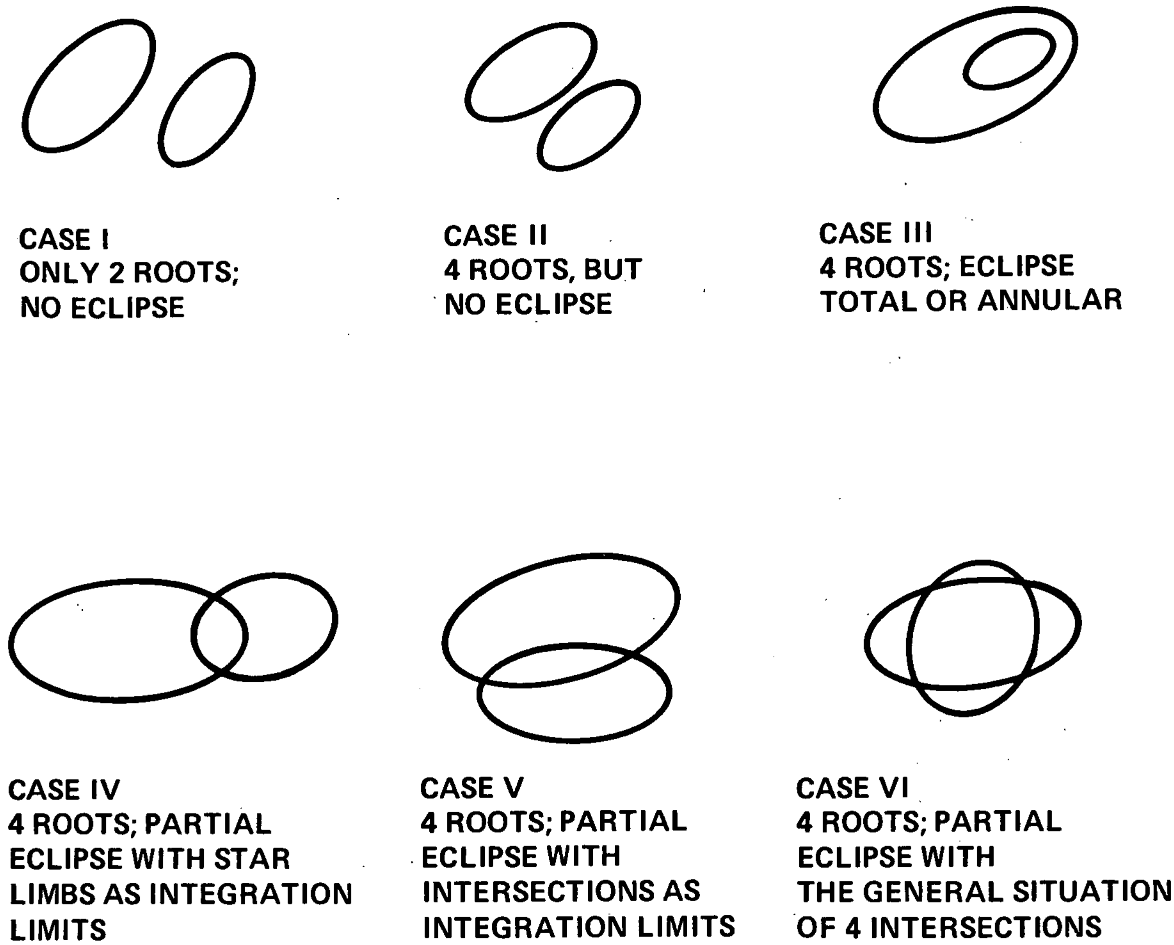


Figure 2. Cases Encountered by Subroutine Limity in Determining Partial Eclipse Integration Limits.

4. If partial derivatives of geometric parameters are being calculated, the previously-determined limits are used to set initial values.

K. Determination of Z Limits of Partial Eclipse (LIMITZ)

LIMITZ is another entry in LIMITY, used by ECLINT to determine the limits of integration in the z direction. Referring to Figure 2, cases IV, V and VI, notice that of the four roots for any given y within the region of overlap, the two "inner" roots are the z limits.

L. Integrations (TOTINT, ANNECL, ECLINT, ATMECL)

Integration by Gaussian quadrature over an ellipse of semi-axes a, b is given by:

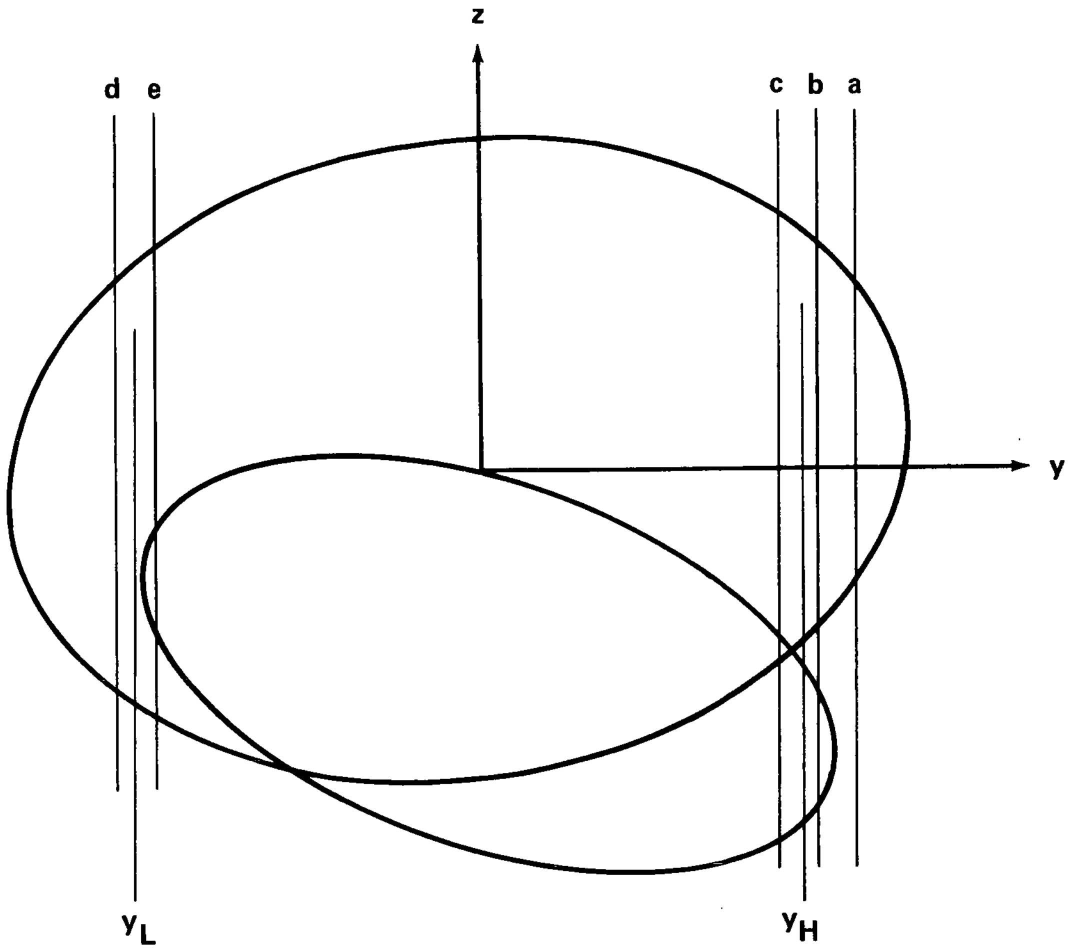


Figure 3. Scan Line Search for Integration Limits, y_H and y_L. Line a Finds Only Two Roots; Line b Finds Four Roots but They do not Intermesh; Line c Finds Four Roots Which Intermesh. Thus y_H is Found by Reversing the Direction of Scan (and Reducing the Step Size) Whenever the Scan Line Changes From Situation b to c or Vice Versa. Line d Finds Only Two Roots; Line e Finds Four Roots, and One Pair is Contained Within Another. Thus y_L is found by going Back and Forth Between Situation d and e.

$$I = ab \sum_{j=1}^{n} W_j \sqrt{1 - X_j^2} \sum_{i=1}^{n} W_i I_p \left(aX_j, X_i b \sqrt{1 - X_j^2}\right) \tag{18}$$

where $I_p(y, z)$ is the intensity, along the line of sight, at point y, z. This is the argument $(I_o + I^*)(1 - u + u \cos \gamma)$ of equation (15). This argument is calculated by the function BRIGHT. The W and X are the Gaussian weights and coordinates, set by GRID.

Integration over an eclipsed area, which, in this case is bounded by elliptical arcs, is given by:

$$I = Y_D \sum_{j=1}^{n} W_j Z_D \sum_{i-1}^{n} W_i I_p (Y_D X_j + Y_S, Z_D X_i + Z_S) \tag{19}$$

where:

$$\left.\begin{aligned} Y_D &= \tfrac{1}{2}(Y_H - Y_L) \\ Y_S &= \tfrac{1}{2}(Y_H + Y_L) \\ Z_D &= \tfrac{1}{2}(Z_H - Z_L) \\ Z_S &= \tfrac{1}{2}(Z_H + Z_L) \end{aligned}\right\} \tag{20}$$

Y_H and Y_L are the y limits of integration. $(Y_H > Y_L)$ from LIMITY; Z_H and Z_L are the Z limits, from LIMITZ, and are functions of:

$$Y_j = Y_D X_j + Y_S.$$

The function BRIGHT operates in the plane of the sky (y, z) coordinate system. Except for ECLINT, integration is performed in the $(\bar{y}, \bar{z})$ coordinate system, aligned with the ellipse axes. Thus, in most cases, a coordinate rotation is necessary. Additionally, coordinate translations must be performed in the case of ANNECL, because the integration is performed over the area of one star, but with the intensity of the points on the other star. Note that in ECLINT, if star B is eclipsed, this translation must also be performed, since the integration limits are in the $(x, y, z)_A$ system, centered on star A.

Whenever an atmospheric eclipse can occur, ATMECL is used in lieu of TOTINT for the eclipsed star. Every point which is behind the atmosphere of the eclipsing star is calculated by BRIGHT but then attenuated by ABSORB. Note that the portion of the eclipsed star which will be physically eclipsed is not attenuated, since that light will be removed by ECLINT or ANNECL. Integration precision in ATMECL is relatively poor.

Schematic grid points (for 4 X 4 integration) are shown in Figure 4.

M. Intensity at a Point (BRIGHT)

BRIGHT is used by the integration routines TOTINT, ANNECL, ECLINT and ATMECL to evaluate the argument of the integration; that is the apparent intensity at a point on a star. Normally, BRIGHT calls the reflection function REFL. However, when partial derivatives are being calculated, certain short-cuts are desirable. In the matrix RINTS are stored five sets of the results of the function REFL; for star A at t, star B at t, eclipsed area at t, star A at T_Q, and star B at T_Q. When a numerical derivative is calculated, these "old" stored values of incident energy are used to calculate reflection. That is, it is assumed that the effect of a change in incident energy is second order when a parameter is perturbed. See Table VIII.

Table VIII. BRIGHT

Statement Number	Sub Program	Computational Milestones
		Step MREF, which counts the calls to BRIGHT for this particular integration
		Set NN. NREF is 1 if total integration of star 1; 2 if total integration of star 2. 3 if eclipse integration.
		NINC is 3 for normalization integrations; 0 for all others. NN will thus always be 1, 2, 3, 4 or 5
		Calculate cos γ
		Set up RAD for call to REFL. The parameters $x, y, z, \sqrt{x^2 + y^2 + z^2}$ are passed via COMMON/RINT
	REFL	If this is not a partial derivative calculation, load RINTS with new reflection values. Otherwise, the pre-stored values are used.
16		Calculate intensity
22		If the y, z coordinates in the call to BRIGHT fall off the star, then x is imaginary. In this case x is set to zero and a message printed. This may occur occasionally due to rounding error, in which case x = 0 is a good approximation.

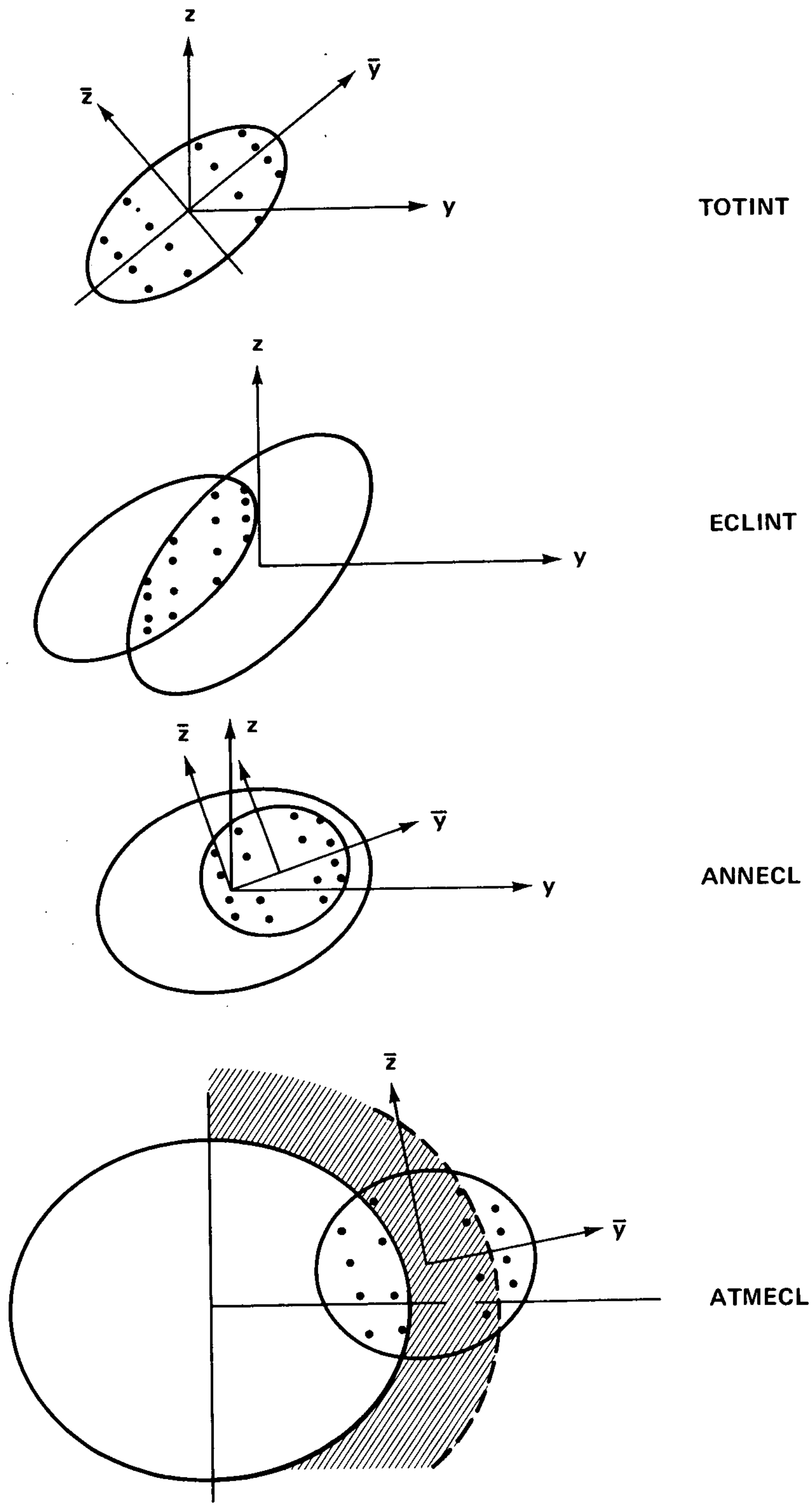


Figure 4. Schematic Representation of Integration Grid as Used in Integration Functions

N. Incident Energy for Reflection (REFL)

The energy incident upon one star is assumed to be a function of the following properties of the other (source) star:

1. Apparent angular extent as seen from the first star,
2. Intensity at the end of the a-axis,
3. Limb darkening,
4. Apparent zenith distance as seen from the first star.

This approximation has been empirically determined from more exact integrations over the source star. The more rigorous treatment increases computing time by about an order of magnitude.

O. Initialization of Reflection (ZONES)

ZONES remains because historically it was necessary when detailed reflection was used. It is preserved in case the user wants to incorporate a more exact reflection model. The present ZONES calculates the emergent intensity at the end of the a-axis.

P. Calculation of Total Stellar Radiation (OUTPUT and ENERGY)

These routines are used only for display purposes. The total 4π steradian stellar energy output is calculated to more nearly represent the relative stellar brightnesses than the surface intensity at time T_Q. The calculation has an accuracy of 2 to 3%.

Q. Atmospheric Absorption (ABSORB)

ABSORB attenuates any intensity (as calculated by BRIGHT) according to equations (13) and (14).

R. Differential Corrector (SOLVE1)

The differential corrector logic flow is shown in Table IX. The program has the following traits:

1. The outer loop is over the individual observations; so that for a particular time of observation, t_i, all necessary partial derivatives are calculated.
2. The logical vector TEST determines which system parameters are to be considered as variables.

Table IX. Differential Corrector Logic Flow

Statement Number	Sub Program	Computational Milestones
		Start outer loop over all observations. When all observations are done, go to 250
		Set IREF flag to 1 for calculation of ΔI
	GEOMET	Compute time-independent quantities; including normalization
	LUMC	Obtain calculated intensity
		Form ΔI
		Set IREF flag to 2 for calculation of partials
200		Check each parameter to see if designated as a variable; if so, continue. When no more variables, go to next observation
401-415		Special coding for each variable to set up high and low values for partials. From here control goes to 440, 450, 480, 4400, or 4800
4400, 4800	ASTROX GEOMET	If this is astrophysical variable, convert to model parameters, and calculate new normalization
4530	LUMC	Calculate partial derivative; go to 200
440, 450 480	GEOMET	Calculate new normalization for non-astrophysical variable
453	LUMC	Calculate partial derivative; go to 200
250		Output time vs. ΔI
500	LSQS	Solve normal equations
520		Screen size of corrections; reduce if necessary. If not converged, step INDIC counter. If extremely small correction, delete from further consideration by changing logical vector
533		Screen values of new parameters
710	ASTROX	Insure that all new parameters are installed
		Set QUAD with new value
1720, 1730		If, upon entry, there were no "TRUE" variables; or if convergence has failed, output time and delta magnitude

3. These variables are arranged (by integer vector VV) so that "photometric" variables precede "geometric" variables in order of treatment.
4. Since internally all subprograms operate in "model space," whereas SOLVE1 operates in "astrophysical space," all astrophysical variable perturbations must be "translated" by a call to ASTROX. The two types of variables are differentiated by the integer vector V.
5. After being calculated by the least squares subroutine LSQS, the differential corrections are subjected to a variety of filters, any of which the user may feel he wants to alter. These are:
   a. Restrict all differential corrections to 25% of the current value of the variable (or to 0.25 if the variable is zero).
   b. Consider a variable as "over-converged," and delete it as a variable, if the differential correction is less than 0.01% of the current value (or less than 0.001% if the current value is less than 0.001).
   c. Flag a variable as not converged if the differential correction is greater than 1% (or greater than 0.001 if the variable is zero). Convergence is accepted only if no variable is so flagged.
   d. Examine the new values of the variables (old value plus differential correction) to see if they lie in an acceptable range. The upper limits are in vector SA and the lower limits in vector SB.
6. Note that system quadrature magnitude (i.e., the normalization) is available as an adjustable parameter, but it is handled in a different manner than the other variables.

S. Least Squares (LSQS and MAMUL)

LSQS and the matrix multiplication routine MAMUL, have been taken from an old IBM 704 SHARE routine.

T. Common Blocks

The common blocks are organized in the following manner:

1. ORBE main orbital elements.
2. AUXE auxilliary or secondarily derived orbital elements.
3. TVARS quantities which are time-dependent.
4. FLAGS a variety of flags and counters.
5. CONST π, 2π, and $\pi/2$.
6. OBS vectors set aside primarily to hold observational data (up to 101 observations).
7. VARIB variables for "astrophysics space."
8. ORBIT fixed celestial mechanical parameters.
9. ROTAT coordinate rotation and translation constants.

10. GAUSS Gauss quadrature constants.
11. GAUXX Guass 16 X 16 quadrature constants.
12. TRIG sin and cos functions of important angles.
13. RINT communication between BRIGHT and REFL.

V. Operation of the Computer Program

The computer program can be used for two distinct calculations; predicting a light curve for given parameters or solving an observed light curve for the parameters which produce the best fit.

A. Input of Parameters

Orbital parameters, whether for prediction or as initial approximations for solution, are input one per input record. The nature of the parameter is specified by a two-digit code, and the value in a 10-digit field, which must have the decimal point specified. Table II indicates the code, the parameter meant, and the default values. Note that the interpretation of several of the input parameters (astrophysical vs. model) is determined by the value specified for the polytropic index: an index of zero specifies model parameters. Input is terminated by a code of 1, -1, or -2. Table III lists a number of control parameters and their interpretations.

B. Light Curve Prediction

A light curve may be predicted either in astrophysics or model space. The use of model space may be useful for "unusual" distortions, for example, simulating a rapidly-rotating star by an oblate spheroid.

The prediction mode is specified by terminating input with a code of zero. Prediction may be either for equally spaced times over any interval, or at individually specified times.

1. Prediction at equally spaced times—specify start time, interval, and end time (see Table III). The end time may equal the start time, but it may not be earlier than the start time. The interval must not be zero.
2. Prediction at specified times—specify a time interval of zero. Then, following the card which ends parameter input, specify times, one per card, in a 10 wide field. The decimal must be specified. End of input is specified by a negative time.

C. Light Curve Solution

Light curve solution is possible only in astrophysics space. Solution mode is specified by terminating parameter input with an input code of -1. This card is then followed by the observations, one per card. An observation card consists of three fields of ten (specify the decimal in all fields):

col 1-10: time
col 11-20: observed magnitude or intensity (See input code 87 in Table III)
col 21-30: weight of observation

Note that the least squares subroutine will give all observations unit weight if the first observation is specified to have a weight of zero. In spite of possibly intrinsically lower accuracy, it is not advisable to give in-eclipse observations lower weight, since their low accuracy is compensated by their high information content.

Observational data input is ended by a data card with a negative time. This card is ten followed by a "T/F" card which specifies which parameters are "variable." This card consists of seventeen 2-character fields. A "T" indicates the parameter is to be variable. An "F" (or blank) means it is to be held at its specified value. The seventeen fields, which mostly correspond to input data codes 1 through 17, are described in Table X. Note that another solution of the same observational data can be made by terminating parameter input with input code -2. No data cards are read, so the next card must be the T/F card.

If the T/F card specifies all parameters false (a blank card) then observed minus computed magnitudes are printed for each observed time.

D. Data Card Arrangement

Data input decks are shown schematically in Figure 5.

1. Card 1--title card. Must have first four letters "WINK". Any other four characters (including blanks) terminates the program.
2. Any number of parameter cards, the last card containing a code of 1, -1, or -2. The requirement for any other cards is determined by this number.
3. Observation cards
   a. If the last parameter card had code zero, and the interval is not zero, no more cards are read. The next card should be a new title card ('WINK' to go on, anything else, e.g., 'STOP', to stop).

Table X. T/F Card Codes

column; A "T" in column makes this a variable	
2	Inclination
4	$e \sin \omega$
6	$e \cos \omega$
8	u_A
10	u_B
12	A_{oA}
14	k_ν
16	β_A
18	β_B
20	T_A*
22	T_B*
24	T_{CONJ}
26	mass ratio
28	quadrature magnitude
30	(not used)
32	w_A
34	w_B

b. If the last parameter card had code zero, and the interval is zero, observation cards specifying times for prediction are needed. The last card must have a negative time. The next card would be a new title card.

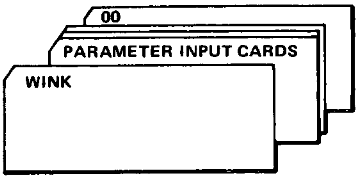
c. If the last parameter card had code -1, observation cards are necessary; the last card must have a negative time. This card must be followed by a T/F card, specifying which parameters are variable. The next card would be a new title card.

d. If the last parameter card was -2, the next card must be the T/F card; followed by a new title card.

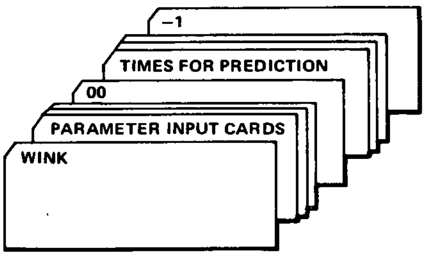
E. Sample Data Run

To assist in checking out the program on another computer, Appendix IV contains sample input/output for different runs.

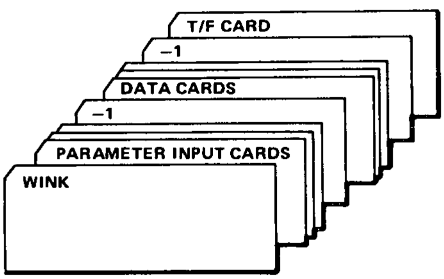
*It is never advisable to allow both temperatures as variables: There is not that much information in one light curve. Fix the temperature best determined by spectroscopy or multicolor photometry.



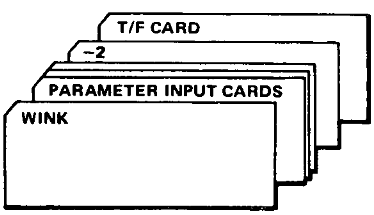
PREDICT LIGHT CURVE FOR EQUALLY-SPACED POINTS (PARAMETER 21 $\neq$ 0)



PREDICT LIGHT CURVE FOR INDIVIDUALLY SPECIFIED TIMES (PARAMETER 21=0)



SOLVE LIGHT CURVE



SOLVE LIGHT CURVE WHICH IS ALREADY READ IN

Figure 5. Input Card Decks.

REFERENCES

Chandrasekhar, S., 1933, M.N.R.A.S. 93, 462.

Chen, K. Y. and W. J. Rhein, 1969, Publ. A.S.P., 81, 387.

Hill, G. and J. B. Hutchings, 1970, Ap. J., 162, 265.

Horak, T. B., 1970, A.J., 75, 1116.

Lucy, L. B., 1968, Ap. J., 151, 1123.

Mochnacki, S. W. and N. A. Doughty, 1972, M.N.R.A.S., 156, 51.

Napier, W. McD., 1968, Ap. & Sp. Sci., 2, 61.

Rucinski, S. M., 1969, Acta. Astr., 19, 125.

Wilson, R. C. and E. J. Devinney, 1971, Ap. J., 166, 605.

Wood, D. B., 1969, Bull A.A.S., I, 267 (abstract).

Wood, D. B., 1971a, Publ. A.S.P., 83, 286.

Wood, D. B., 1971b, A.J., 76, 701.

Wood, D. B., 1971c, Bellcomm Publication TM-71-1011-4 (April 16, 1971).

Wood, D. B., 1971d, Bellcomm Publication B71-09018 (September 22, 1971).

Wood, D. B., 1971e, invited paper, IAU Colloquium No. 16, to be published.

Wood, D. B., 1972, to be published.

APPENDIX I

EQUATIONS

A. Orbital Mechanical Equations Used in ORBITB

1. Time of periastron, T_o, is found from:

$$\cos^{-1}\left[\frac{e \sin \omega + e^2}{e(1 + e \sin \omega)}\right] - q \frac{e \cos \omega}{1 + e \sin \omega} = \mu (T_c - T_o)$$

where:

$$q = \sqrt{1 - e^2}$$
$$\mu = 2 \pi / P$$

2. The mean anomaly, M, is given by:

$$M = \mu (t - T_o)$$

3. Kepler's equation, $E - e \sin E = M$, is conveniently solved by iteration when expressed in the form:

$$\Delta E = \frac{e \sin E - E + M}{1 - e \cos E}$$

where the first approximation, for $e \lesssim 0.75$, is $E \simeq M$.

4. The true anomaly, v is found from:

$$\cos v = \frac{\cos E - e}{1 - e \cos E}$$
$$\sin v = \frac{q \sin E}{1 - e \cos E}$$

5. The orbital longitude θ follows from:

$$e \sin \theta = e \sin \omega \sin v - e \cos \omega \cos v$$
$$e \cos \theta = e \cos \omega \cos v + e \cos \omega \sin v$$

6. The radius vector, R, is given by

$$R = R_o (1 - e \cos E)$$

Preceding page blank

7. For an eccentric orbit, the star, rotating in uniform motion, will deviate from θ by a small angle Θ where:

$$\sin \Theta = \sin v \cos M - \cos v \sin M$$

The stars orbital longitude, θ; is given by:

$$\theta' = \theta - \Theta$$

8. The apparent separation of centers is given by:

$$\delta = R\delta'$$

where:

$$\delta' = (\sin^2\theta + \cos^2\theta \cos^2 i)^{1/2}$$

9. The angle, χ, from the Z-axis to the center of the other star is given by:

$$\cot \chi = \cos i \cot \theta'$$

B. Ellipsoidal Star Equations Used in PARAM

1. The triaxial ellipsoid in 3-dimensions is generally written as:

$$Ax^2 + By^2 + Cz^2 + 2Dxy + 2Eyz + 2Fxz = a^2b^2c^2$$

where:

$$A = b^2c^2\cos^2\theta'\sin^2 i + a^2c^2\sin^2\theta'\sin^2 i + a^2b^2\cos^2 i$$
$$B = b^2c^2\sin^2\theta' + a^2c^2\cos^2\theta'$$
$$C = b^2c^2\cos^2\theta'\cos^2 i + a^2c^2\sin^2\theta'\cos^2 i + a^2b^2\sin^2 i$$
$$D = (b^2c^2 - a^2c^2)\sin\theta'\cos\theta'\sin i$$
$$E = (b^2c^2 - a^2c^2)\sin\theta'\cos\theta \cos i$$
$$F = (b^2c^2\cos^2\theta' + a^2c^2\sin^2\theta - a^2b^2)\sin i \cos i$$

2. The cosine of the projection angle for limb darkening is:

$$\cos \gamma = (Ax + Dy + Fz)/T$$

where $T = [x^2(A^2 + D^2 + F^2) + y^2(B^2 + D^2 + E^2) + z^2(C^2 + E^2 + F^2)$
$+ 2xy(AD + BD + EF) + 2xz(AF + CF + DE)$
$+ 2yz(BE + CE + DF)]^{1/2}$

3. To project onto the plane of the sky, x must be eliminated by the expression:

$$x = -\frac{Dy + Fz}{A} + \frac{1}{A}\left[Ny^2 + Pz^2 + 2Ryz + S\right]^{1/2}$$

where:

$N = D^2 - AB$
$P = F^2 - AC$
$R = DF - AE$
$S = Aa^2b^2c^2$

4. The axes of the apparent ellipse—the outline of the ellipsoid on the plane of the sky—are:

$$a' = \left[\frac{2S}{-[(P - N)^2 + (2R)^2]^{1/2} - (P + N)}\right]^{1/2}$$

$$b' = \left[\frac{2S}{[(P - N)^2 + (2R)^2]^{1/2} - (P + N)}\right]^{1/2}$$

5. The major axis of this ellipse is rotated from the y-axis by angle ϕ given by:

$$\cot 2\phi = \frac{N - P}{2R}$$

6. The angle, α between the major axis of one apparent ellipse and the center of the other is given by:

$$\cos\alpha = (\sin\theta' \cos\phi - \cos\theta' \cos i \sin\phi)/\delta'$$
$$\sin\alpha = -(\sin\theta' \sin\phi + \cos\theta' \cos i \cos\phi)/\delta'$$

C. Other Important Equations

1. Mean stellar radius (to sub-earth point at quadrature)

$$\bar{r} = \left[\sin^2 i\left(\frac{\sin^2\Theta}{a^2} + \frac{\cos^2\Theta}{b^2}\right) + \frac{\cos^2 i}{c^2}\right]^{-1/2}$$

2. Partial derivative $\partial I/\partial Q$ where Q is the normalization factor:

$$\partial I/\partial Q \simeq 0.92061\ I_{COMP}$$

3. Approximation to total bolometric energy radiated over 4π steradians:

$$E_{bolo} = 8\left(\frac{a}{2}\right)\left(\frac{b}{2}\right)\sum_{j=1}^{n} W_j \frac{1}{2}\sqrt{4 - (1 + X_j)^2}\sum_{i=1}^{n} W_i I_N (x_N, y_N)$$

where:

$$x_N = \left(\frac{a}{2}\right)(1 + X_j)$$

$$y_N = 1/2\left(\frac{b}{2}\right)(1 + X_i)\ \sqrt{4 - (1 + X_j)^2}$$

and the normal emergent intensity, I, at (x_N, y_N) is given by:

$$I_N = \bar{I}\left(1 - v + v\frac{r_N}{\bar{r}}\right)G$$

The radius, r_N to point (x_N, y_N) is given by:

$$r_N = \left[c^2 - X_N^2\left(\frac{c^2}{a^2} - 1\right) - y_N^2\left(\frac{c^2}{b^2} - 1\right)\right]^{1/2}$$

The quantity G is the area element:

$$G = \left[\frac{1 + \frac{x_N^2}{a^2}\left(\frac{c^2}{a^2} - 1\right) + \frac{y_N^2}{b^2}\left(\frac{c^2}{b^2} - 1\right)}{1 - \frac{x_N^2}{a^2} - \frac{y_N^2}{b^2}}\right]^{1/2}$$

APPENDIX II

REFLECTION APPROXIMATION

The amount of radiation, L^*, which is incident upon the reflecting star is assumed to be a function of:

1) the intensity of the substellar point on the source star, I_s.
2) the apparent area of the source star, A_s
3) the limb darkening of the source star, u_s
4) the cosine of the zenith distance of the source star, $\cos \lambda'$.

Thus, we seek a function of the form:

$$L^* = I_s A_s f(u_s) g(\cos \lambda')$$

If the source star were at an infinite distance, we would expect the limb darkening dependence to be $(1 - u/3)$. However, since the star is quite close when reflection is important, its radiation is more strongly dependent upon limb darkening. Empirically, the dependence is found to be very well approximated by:

$$f(u_s) = 1 - u_s/2,$$

which is reasonable since:

$$\int_0^{\pi/2} (1 - u + u \cos\theta) \sin\theta d\theta = 1 - u/2$$

The function g is more complex. For a given geometry, g can be approximated by two straight lines, with the change of slope occurring when the source star starts to set. The slopes and intercepts of these lines are functions of the sizes of the source star, a_s, and the reflecting star, a_r. The approximation used is:

$$g(\cos \lambda') \simeq -0.065354 + a_r g_1 + (\cos \lambda')(2.044 + a_r g_2) + C$$

where:

$$g_1 = 0.224935 - 0.761696 a_s + 3.81425 a_s^2$$
$$g_2 = -.170831 + 1.231707 a_s - 9.955083 a_s^2$$

C is the larger of 0 or:

$$0.38736 + a_r(-.82442 + 1.43431a_s) + (\cos\lambda')\left[-1.22172 + a_r(-.43316 + 4.9378a_s)\right]$$

The zenith angle is given by:

$$\cos\lambda' = (\lambda\Delta x + \mu\Delta y + \nu\Delta z)/d$$

where:

(λ, μ, ν) are the direction cosines of the local normal and $(\Delta x, \Delta y, \Delta z)$ are the direction numbers of the center of the source star. The distance between the point on the reflecting star and the center of the source star, d, is given by:

$$d = [(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2]^{1/2}$$

The apparent area of the source star is given by calculating:

$\lambda = r_s/d$ and

$\lambda_2 = \lambda - \frac{\pi}{2} + \lambda'$.

Then:

1) If $0 < \lambda_2 < \lambda$

$$A = \lambda^2 \cos^{-1}\left(\frac{\lambda - \lambda_2}{\lambda}\right) - (\lambda - \lambda_2)(2\lambda\lambda_2 - \lambda_2^2)^{1/2}$$

2) If $\lambda \leqslant \lambda_2 < 2\lambda$

$$A = \pi\lambda^2 - \lambda^2 \cos^{-1}\left(\frac{\lambda - \lambda_1}{\lambda}\right) + (\lambda - \lambda_1)(2\lambda\lambda_1 - \lambda_1^2)^{1/2}$$

3) If $\lambda_2 \geqslant 2\lambda$

$$A = \pi\lambda^2$$

4) If $\lambda_2 \leqslant 0$, $A = 0$

Note that $\lambda_1 = \lambda + \frac{\pi}{2} - \lambda'$.

APPENDIX III

COMPUTER LISTINGS

PRECEDING PAGE BLANK NOT FILMED

```
****************************************************************************

C   MAIN PROGRAM FOR ECLIPSING BINARY CURVE PREDICTION OR SOLUTION.
C   THIS VERSION CONSTRUCTED FOR LEAST SQUARE SOLUTION WITH
C        ASTROPHYSICAL VARIABLES
C             WRITTEN BY D. B. WOOD
C             REVISED FOR MINIMUM IBM S360 CONFIG ON 30 SEPT 1972
      COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1  EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
      COMMON/CONST/ PI,TWOPI,HALPI
      COMMON/ORBIT/ E,TO,Q,MU
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/OBS/ TIME(101),LUM(101),WT(101),CLUM(101)
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
      COMMON/VARIB/BLQ(18)
      DIMENSION BLOAD(18),BL(18),SKAL(2),TP(2)
      DIMENSION AUX(8),WORD(15),TITLE(15),LABL(3)
      EQUIVALENCE (AUX(1),SCALE(1)),(BL(1),INCL)
      DATA LABL/'A','B',' '/
      DATA SKAL/0,0/
      DATA STOP/'WINK'/
      REAL*8 IMX(3)/'4X4     ','6X6     ','16X16   '/
      DATA BLOAD/90.,2*0,2*.6,.25,1.,2*.25,2*1.E4,0,1.,4*0,1./
      DATA WORD/'    ','    ','    ',' ATM','    ',' ANN','   T',
     1   ' PAR','OSPH','    ','ULAR','OTAL','TIAL','ERIC','    '/
      REAL  INCL, MU, LUM, LUMC
      LOGICAL TEST
C  ONE-TIME INITIALIZATIONS
      DEGRAD = 1.745329E-2
      BLOAD(1) = BLOAD(1)*DEGRAD
      DO 10  I=1,18
      BL(I) = BLOAD(I)
   10 BLQ(I) = BL(I)
      DO 11  I=1,8
   11 AUX(I) = 0.
      RNOT = 1.
      NTEG = 1
```

Preceding page blank

```
      S = 0.
      TINT = 0.1
      END = 0.5
      QUAD = 0.
      ONE = 1.
      WAVE = 5500.
      POLYX = 5.
      JUMP = 1
      ILUM = 0
      ICRD = 0
      PI = 3.141592
     TWOPI = PI+PI
      HALPI = PI/2.
 50   NFLAG = 1
C   RE-ENTRY POINT
  66  READ (5,965) (TITLE(I),I=1,15)
      WRITE (6,966) (TITLE(I),I=1,15)
C  STOP IF FIRST WORD OF TITLE IS NOT 'WINK'
      IF (TITLE(1) .NE. STOP)  STOP
  47  DO 48  I=1,17
      BL(I) = BLQ(I)
  48  TEST(I) = .FALSE.
      IFFY = 6
 51     READ (5,805) I,DATA
      WRITE (6,951) I,DATA
COL 1-2, INT. CODE -- COL 3-12, PARAMETER     ***REMARK***
C  N<0                 QUAD *                 INITIATE SOLVE MODE
C   (USE N=-1 TO REQUEST OBS. DATA INPUT. USE N=-2 IF OBS. IN CORE)
C    =0                QUAD *                 INITIATE PREDICT MODE
C    =1                I (DEGREES)            CODE 1-18 FOR ORB. PARAM.
C    =2                E SIN OMEGA
C    =3                E COS OMEGA
C    =4                UA
C    =5                UB
C    =6                A                      (UNPERTURBED SPHERE)
C    =7                K                      (UNPERTURBED SPHERE)
C    =8                ELLIP A                (BETA:  GRAV. EXP.)
C    =9                ELLIP B                (BETA:  GRAV. EXP.)
C    =10               EPSI A                 (EQUATORIAL TEMP)
C    =11               EPSI B                 (EQUATORIAL TEMP)
C    =12               T CONJ
C    =13               VA                     (MASS RATIO)
C    =14               VB                     (USED INTERNALLY FOR QUAD MAG)
C    =15               J                      (NOT USED FOR AP SOLN)
C    =16               WA
```

```
C    =17                  WB
C    =18                  PERIOD
C    =19                  NTEG               INTEGRATION PRECISION CODE
C    =20                  START TIME         FOR LIGHT CURVE PREDICTION
C    =21                  DELTA TIME         FOR LIGHT CURVE PREDICTION
C    =22                  END TIME           FOR LIGHT CURVE PREDICTION
C    =23                  INTENSITY          'THIRD' STAR LIGHT
C    =24                  QUAD *             SET QUADRITURE MAGNITUDE
C    =25                  RNOT               SET ORBITAL RADIUS SCALE FACTOR
C    =26                  WAVELENGTH         WAVELENGTH OF OBS (ANGSTROMS)
C    =27                  POLYTROP I         POLYTROPIC INDEX OF STARS
C                                            IF INDEX=0, BYPASS ASTROPHYSICS
C    =31                  SCALE HT A         STAR ATMOSPHERE
C    =32                  SCALE HT B         STAR ATMOSPHERE
C    =33                  TAU A              OPTICAL DEPTH OF ATMO AT SURFACE
C    =34                  TAU B              OPTICAL DEPTH OF ATMO AT SURFACE
C    =35                  SLIP A             ORBITAL PHASE SHIFT (RADIANS)
C    =36                  SLIP B             ORBITAL PHASE SHIFT (RADIANS)
C    =37                  TILT A             STAR EQUATOR TILT (RADIANS)
C    =38                  TILT B             STAR EQUATOR TILT (RADIANS)
C    =84                  IFFY               MAX. ITERATIONS
C    =86                  0,1,2              1 FOR PUNCHED OUTPUT IN MAG (2:LUM)
C    =87                  0,1                1 TO READ INPUT AS LUMINOSITY
C    =88                    --               FLIP/FLOP FOR ORB. ELEMENT PRINTOUT
C    =99                    --               LEAVE PROGRAM - RETURN TO EXEC SYSTEM
C    (ANY UNIDENTIFIABLE CODE IS IGNORED)
C  * NOTE THAT 'QUAD' IS LIGHT AT QUADRITURE - MAY BE INPUT WITH '24'
C         OR WITH COMPUTATION INITIATION CODE (N .LE. 0).  IN THIS LATTER
C         CASE, QUAD=0 IS IGNORED AND PREVIOUS VALUE USED.
C    NOTE THAT CODES .LE. 0  TERMINATE INPUT AND START COMPUTATION
          IF (I .LE. 0)  GO TO 55
          IF (I .EQ. 20)  S = DATA
          IF (I .EQ. 21)  TINT = DATA
          IF (I .EQ. 22)   END = DATA
          IF (I .EQ. 23)  ONE = 1. - DATA
          IF (I .EQ. 24)  QUAD = DATA
          IF (I .EQ. 25)  RNOT = DATA
        IF (I .EQ. 26) WAVE=DATA
        IF (I .EQ. 27) POLYX=DATA
         IF (I .GT. 30 .AND. I .LT. 39) GO TO 70
        IF (I .EQ. 19)  NTEG = DATA
        IF (I .EQ. 84)  IFFY = DATA
        IF (I .EQ. 86) ICRD = DATA
        IF (I .EQ. 87) ILUM = DATA
        IF (I .EQ. 88)  JUMP = 3 - JUMP
```

```
         IF (I .EQ. 99)  STOP
          IF (I .GT. 18)  GO TO 51
          BL(I) = DATA
        IF (I .EQ. 1)  INCL = INCL*DEGRAD
          GO TO 51
  55    IF (DATA)  5,6,5
  5     QUAD = DATA
   6    CALL ASTROQ(WAVE,POLYX)
        STAR3 = 1. - ONE
        SCALE(1) = SKAL(1)*RNOT*A
        SCALE(2) = SKAL(2)*RNOT*A*RATIOK
        IF (I .LT. 0)  GO TO 200
        II = 0
        IF (END .LT. S) GO TO 60
C   IF TINT=0, GO READ TIMES AS INPUT DATA, USING  LOGIC WHICH READS
C       LIGHT CURVE FOR SOLUTION
        IF (TINT)  60,200,53
  53    NOBS = 1
        TIME(1) = S
  7     NOBS = NOBS + 1
        IF (NOBS .EQ. 102)  GO TO 160
        TIME(NOBS) = TIME(NOBS-1) + TINT
        IF (TIME(NOBS) .LE. END) GO TO 7
  100 CONTINUE
      NOBS = NOBS - 1
C   LOOP REENTRY POINT FOR SOLUTION OR NEW PARAMETERS
  52    CONTINUE
        CALL GRID(NTEG)
        AINCL = INCL/DEGRAD
        IREF = 1
        CALL GEOMET
        RATIO = STAR2/STAR1
        SLB = ONE*RATIO/(1. + RATIO)
        SLA = ONE - SLB
        GO TO (103,102), JUMP
 103    CONTINUE
        EA = ENERGY(1)
        EB = ENERGY(2)
        ER = EB/EA
        ELB = ER/(1. + ER)
        ELA = 1. - ELB
        OMEGA = 0
        IF (E .NE. 0.)  OMEGA = ARSIN(ESINW/E)/DEGRAD
        GNU2 = BLQ(6)*BLQ(7)
        AZ1 = RNOT*BLQ(6)
```

```
      AZ2 = RNOT*GNU2
      IF (POLYX .EQ. 0)  GO TO 105
CALCULATE POLAR TEMPERATURE (FOR DISPLAY ONLY)
      C = 1.43879E8/WAVE
      DO 104  I=1,2
      TP(I) = EXP(C/BLQ(9+I)) - 1.
 104  TP(I)=C/(ALOG(1.+1./(VMA(I)+BL(12+I)*CC(I)/BB(I))*TP(I)))
 105  CONTINUE
      WRITE (6,901) PERIOD,TCONJ,AINCL,ESINW,ECOSW,OMEGA,E,RNOT
      WRITE (6,902) LABL(1),BLQ(6),AZ1,BLQ(8),BLQ(10),TP(1)
     1             , LABL(2), GNU2 ,AZ2,BLQ(9),BLQ(11),TP(2)
      WRITE (6,903) WAVE,A,RATIOK,BLQ(7),RATIOJ,BOLOJ,BLQ(13),QUAD
     1   ,STAR3,POLYX
      WRITE (6,904) LABL(1),ELLIP,EPSI,UA,VA,WA,AA(1),BB(1),CC(1)
     1  ,LABL(2),ELLIPB,EPSIB,UB,VB,WB,AA(2),BB(2),CC(2)
      WRITE (6,905) LABL(1),STAR1,SLA,EA,ELA,LABL(2),STAR2,SLB,EB,ELB
      WRITE (6,906) RATIO,ER
      WRITE (6,907) (LABL(I),SCALE(I),SURF(I),SLIP(I),TILT(I),I=1,2)
      WRITE (6,908) IMX(NTEG)
  102 IF (II .LT. 0)  GO TO 150
COMPUTE THEORETICAL LIGHT CURVE
      WRITE (6,909)
      DO 110  I=1,NOBS
      CLUM(I) = LUMC(TIME(I))
      CINT = -2.5*ALOG10(CLUM(I)) + QUAD
      IF (KSTAR .EQ. 3)  JTYPE = 5
      WRITE (6,910) WORD(JTYPE),WORD(JTYPE+5),WORD(JTYPE+10),
     1  LABL(KSTAR),TIME(I),PHASE,CLUM(I),CINT
      IF (ICRD .EQ. 0)  GO TO 110
      IF (ICRD .EQ. 2)  GO TO 107
C   PUNCH EITHER IN LUMINOSITY OR MAGNITUDE
      WRITE (7,980) TIME(I),CINT
      GO TO 110
  107 WRITE (7,980) TIME(I),CLUM(I)
  110 CONTINUE
      GO TO 66
C     LIGHT CURVE SOLUTION
  150 CONTINUE
      CALL SOLVE1(IFFY)
  155 IF (IFFY) 59,58,52
  58  WRITE (6,925)
      GO TO 66
  59  IF (IFFY+2) 66,66,46
  46  WRITE (6,926)
      DO 49  I=1,17
```

```
      IF (.NOT. TEST(I))  GO TO 49
      WRITE (6,949) I,BLQ(I)
 49   CONTINUE
      GO TO 66
 70   J = I - 30
      IF (J .LE. 2)  GO TO 75
      AUX(J) = DATA
      GO TO 51
 75   SKAL(J) = DATA
      GO TO 51
C SOLUTION DESIRED, READ OBSERVATIONAL DATA
 200   IF (I .LT. -1)  GO TO 203
      NOBS = 1
      WRITE (6,921)
 201  READ (5,800) TIME(NOBS),LUM(NOBS),WT(NOBS)
     WRITE (6,900) TIME(NOBS),LUM(NOBS),WT(NOBS)
      IF (TIME(NOBS) .LT. 0)  GO TO 202
      NOBS = NOBS + 1
      IF (NOBS .EQ. 102)  GO TO 260
      GO TO 201
 202  NOBS = NOBS - 1
      IF (I .EQ. 0)  GO TO 52
C     'T' MAKES THAT A VARIABLE TO ITERATE (PARAM. ARE IN ORDER 1-18)
C     IF BLANK LINE RETURNED, PROGRAM OUTPUTS TIME VS DEL MAG
 203  READ (5,812) (TEST(J),J=1,17)
      WRITE (6,922) NOBS, (TEST(J), J=1,17)
      IF (I .LT. -1)  GO TO 206
      F = EXP(-.921034*QUAD)
      WRITE (6,915)
      DO 207  J=1,NOBS
      IF (ILUM)  204,204,210
 210  CINT = LUM(J)
      GO TO 205
 204  CINT = EXP(-.921034*LUM(J))
 205  LUM(J) = CINT/F
 207  WRITE (6,916)  TIME(J),LUM(J)
 206  II = I
      GO TO 52
 60   WRITE (6,960)
      STOP
 160  WRITE (6,961)
      GO TO 100
 260  WRITE (6,961)
      GO TO 202
 805    FORMAT (I2,F10.3)
```

```
800      FORMAT (3F10.7)
812      FORMAT (17L2)
900   FORMAT (5X,2F12.7,F12.1)
901    FORMAT('1   ORBITAL PARAMETERS'//5X,'P',13X,'TC',11X,'I',7X,
     1    'ESINW      ECOSW       NODE        E       SEMI-AXIS'/
     2    F11.6,F16.6,F9.3,F11.5,F10.5,F10.3,F8.3,F10.3)
902    FORMAT('0   ASTROPHYSICAL PARAMETERS'//10X,'NU',9X,'A0',8X,
     1    'BETA    TEMP(EQ)     (POLAR)'/(' STAR ',1A1,F9.5,F10.4,F10.3
     2    ,2F11.2))
903    FORMAT('0   MODEL PARAMETERS'//6X,'A',9X,'K(A)',7X,'K(NU)',5X,
     1    'J(',F5.0,')   J(BOLO)   MASS RAT   QUAD MAG   3RD LT',5X,
     2    'POLYTROP'/5F11.5,F10.4,F12.4,F11.5,F8.0)
904    FORMAT('0',9X,'ELLIP    ZETA',7X,'U',8X,'V',7X,'W',9X,'A-AXIS',
     1    4X,'B-AXIS    C-AXIS'/(' STAR ',1A1,2F9.4,2F9.3,F8.3,2X,
     2    3F10.4))
905    FORMAT('0   LUMINOSITIES'//10X,'APPARENT   NORMALIZED',5X,
     1    'TOT(4PI)  NORMALIZED'/(' STAR ',1A1,2F11.5,3X,2F11.5))
906    FORMAT('0RATIOS ',  F10.5,15X,F10.5)
907    FORMAT('0',10X,'ATMOSPHERE',10X,'ORBITAL SKEW'//10X,'SCALE',4X,
     1    'TAU',9X,'PHASE    TILT'/(' STAR ',1A1,2F9.4,3X,2F9.4))
908    FORMAT(/'0   INTEGRATION ',1A5/1H1)
909    FORMAT (1H0,6X,'ECL  STAR    TIME',7X,'PHASE',5X,'INT',8X,'MAG'/)
910    FORMAT (3A4,2X,1A1,F12.5,2F10.5,F10.4)
915    FORMAT ('0NORMALIZED INTENSITIES'/'0      TIME',7X,'INT'/)
916    FORMAT (1X,F12.5,F10.5)
921    FORMAT (' ENTER OBS')
922    FORMAT (I6,' OBS'//(17L2))
925    FORMAT (' CONVERGED')
926    FORMAT (' FAILED TO CONVERGE')
949    FORMAT (I3,G14.7)
951    FORMAT (I3,G13.6)
960   FORMAT (' TIME ERROR ABORT')
961   FORMAT (' PROCESSING LIMITED TO 101 POINTS')
965    FORMAT (15A4)
966    FORMAT (1H1,15X,15A4//' INPUT DATA')
980    FORMAT (2F10.6)
      END

*******************************************************************************

      FUNCTION ABSORB (Y,Z,K)
CALCULATES ABSORBTION FOR GIVEN Y,Z IN ATMOSPHERE OF STAR N(=1,2)
C  ABSORBTION EXPRESSED AS FRACTION TRANSMITTED (UNITY FOR NO ABSORB..
```

```
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      N = K
      ABSORB = 1.
C  IF HT NEGATIVE, THIS PART WILL BE PHYSICALLY ECLIPSED. NO ABSORB.
      IF (SCALE(N))  6,6,3
  3   CONTINUE
      DSQ = Y**2 + Z**2
      COSQ = Y**2/DSQ
      RD = 1./SQRT(COSQ/AAXIS(N)**2 + (1.-COSQ)/BAXIS(N)**2)
      HT = (SQRT(DSQ) - RD)/SCALE(N)
      IF (HT)  6,6,5
C  IF HT POSITIVE, (X,Y) IS ABOVE THE SURFACE
  5   TAU = EXP(-HT)*SURF(N)
      IF (TAU .LT. 1.E-4)  GO TO 8
      ABSORB = EXP(-TAU)
  6   RETURN
C  FOR SMALL TAU, APPROXIMATE EXPONENTIAL
  8   ABSORB = 1. - TAU
      GO TO 6
      END
```

```
***************************************************************************
```

```
      FUNCTION ANNECL (N)
C  GAUSSIAN INTEGRATION FOR ANNULAR ECLIPSE OF STAR N (1 OR 2)
C INTEGRATION IS OF STAR N OVER BOUNDARY OF OTHER STAR (K)
      COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
      COMMON /GAUSS/ WT(16),X(16),L,XC(16)
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
      MREF = 0
      NREF = 3
      DCHIP = SIGN(1.,ECHI)*DCHI
      M = N
      K = 3 - M
      SUM2 = 0.
      DO 100  J=1,L
```

```
      SUM1 = 0.
    6  Y = AAXIS(K)*X(J) + DCHIP
    8  DO 50  I=1,L
       Z = BAXIS(K)*X(I)*XC(J) + ABS(ECHI)
       YP = Y*COSP(M) - Z*SINP(M)
       ZP = Y*SINP(M) + Z*COSP(M)
   50  SUM1 = SUM1 + WT(I)*BRIGHT(YP,ZP,M)
  100  SUM2 = SUM2 + WT(J)*XC(J)*SUM1
  110 ANNECL =AAXIS(K)*BAXIS(K)*SUM2
      RETURN
      END
```

```
****************************************************************************
```

```
       SUBROUTINE ASTROQ(WAVE,POLY)
C   SUBROUTINE TO CONVERT PHYSICAL VARIABLES TO MODEL PARAMETERS
       COMMON/VARIB/BLQ(17)
       COMMON/ORBE/BLK(18)
       COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1    ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2    ,STAR1,STAR2,STAR3
       DIMENSION D(5),T(2),NU(2),EP(2),Z(2),V(2),AO(2),QU(2),E(2)
     1 ,BETA(2),AP(2)
       EQUIVALENCE(BLQ(6),GNU),(BLQ(7),RATIOK),(BLQ(13),Q)
       EQUIVALENCE (BLK(8),EP(1)),(BLK(10),Z(1)),(BLK(13),V(1))
     1  ,(BLK(15),RATIOJ)
       DATA D/1.51985,1.1482,1.0289,1.00267,1.0/
C    D IS SET FOR INTEGRAL POLYTROPES ONLY
       REAL NU
       FCNA(A,B)= 1.+(1.+7.*A)*D(IX)*B**3/6.
       FCNB(A,B)= 1.+(1.-2.*A)*D(IX)*B**3/6.
       FCNC(A,B)= 1.-(1.+2.5*A)*D(IX)*B**3/3.
C    INITIALIZATIONS
       IX = INT(POLY)
       IF (IX)  2,2,1
  1    PK = (D(IX)-5.)/D(IX)
       C  = 1.43879E8/WAVE
C    SET UP COMMON/VARIB/ FROM INPUT IN COMMON/ORBE/
  2    DO 3  I=1,17
   3  BLQ(I) = BLK(I)
      BLQ(14) = QUAD
C    CONVERT TO MODEL PARAMETERS
  5    IF (IX)  12,12,6
```

```
 6     NU(1) = GNU
       NU(2) = GNU*RATIOK
       QU(1) = Q
       QU(2) = 1./Q
       DO 10 I=1,2
       BETA(I) = BLQ(7+I)
       T(I) = BLQ(9+I)
       AP(I) = FCNA(QU(I),NU(I))
       EP(I) = FCNB(QU(I),NU(I))/AP(I)
       Z(I) = FCNC(QU(I),NU(I))/(AP(I)*EP(I)**2) - 1.0
       CEX = C/T(I)
       E(I) = EXP(CEX)
       V(I) = PK*BETA(I)*CEX*E(I)/(E(I) - 1.)
 10    CONTINUE
       RATIOJ = (E(1)-1.)/(E(2)-1.)
       BOLOJ  = (T(2)/T(1))**4
      BLK(6) = AP(1)*GNU
       BLK(7) = RATIOK*AP(2)/AP(1)
 12    CONTINUE
       RETURN
       ENTRY ASTROX
       GO TO 5
       END

************************************************************************

       FUNCTION ATMECL (N)
C GAUSSIAN INTEGRATION OVER ENTIRE AREA OF STAR N (1 OR 2)
C  REPLACES TOTINT TO CALCULATE 'TOTAL' INTENSITY IF ATMOS. ECL.
C  PORTION OF STAR N WHICH IS BEHIND ATMOSPHERE OF OTHER STAR
C     (STAR K) IS GIVEN ATTENUATION BY 'ABSORB' FUNCTION.
C     'ABSORB' ATTENUATES OUT-OF-GEOMETRIC-ECLIPSE LIGHT ONLY.
       COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
       COMMON /GAUXX/ WT(16),X(16),L,XC(16)
       COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
       COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
       MREF = 0
       NREF = N
       SUM2 = 0.
       K = 3 - NREF
 6     DO 100  J=1,L
```

```
      SUM1 = 0.
      Y = AAXIS(NREF)*X(J)
      YPP = Y - DCHI
      DO 10  I=1,L
      Z = BAXIS(NREF)*X(I)*XC(J)
      YP = Y*COSP(NREF) - Z*SINP(NREF)
      ZP = Y*SINP(NREF) + Z*COSP(NREF)
      ZPP = Z - ECHI
 10   SUM1 = SUM1 + WT(I)*BRIGHT(YP,ZP,NREF)*ABSORB(YPP,ZPP,K)
100   SUM2 = SUM2 + WT(J)*XC(J)*SUM1
      ATMECL = AAXIS(NREF)*BAXIS(NREF)*SUM2
      RETURN
      END
```

```
************************************************************************

      FUNCTION BRIGHT(YQ,ZQ,N)
CALCULATES BRIGHTNESS IN L.O.S. FOR GIVEN Y,Z ON STAR N (=1,2)
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
     COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1 EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
      COMMON/RINT/X,Y,Z,RAD
C    RINTS STORES REFLECTION VALUES TO ABBREVIATE CALC FOR SOLUTION
      DIMENSION RINTS (5,256)
     DIMENSION U(2),V(2),W(2)
      EQUIVALENCE (U(1),UA),(V(1),VA),(W(1),WA)
     REAL INCL
     REAL LOCINT
      MREF = MREF + 1
      NN = NREF+NINC
      Y = YQ
      Z = ZQ
      II = N
    3 Q1 = PARA(4,II)*Y + PARA(6,II)*Z
     YSQ = Y*Y
     ZSQ = Z*Z
     YZ = Y*Z
```

```
 5      Q2 = PARA(13,II)*YSQ + PARA(14,II)*ZSQ + PARA(16,II)
        Q2 = Q2 + 2.*PARA(15,II)*YZ
        IF (Q2 .LE. 0)  GO TO 22
        X = (SQRT(Q2) - Q1)/PARA(1,II)
   10 T = SQRT(PARA(7,II)*X*X + PARA(8,II)*YSQ + PARA(9,II)*ZSQ + 2.*
     1  (PARA(10,II)*X*Y + PARA(11,II)*X*Z + PARA(12,II)*YZ))
       COSGAM = (PARA(1,II)*X + Q1)/T
COSGAM IS LIMB DARKENING PROJECTION ANGLE
        RAD = SQRT(X**2 + YSQ + ZSQ)
CALL FOR REFLECTED LIGHT IF IREF=1, OTHERWISE USE STORED VALUES
        IF (IREF .EQ. 1)  RINTS(NN,MREF) = REFL(II)
   16  RRATIO = RAD/RBAR(II)
       LOCINT = QINT(II)*(VMA(II) + V(II)*RRATIO) + WMA(II)*RINTS
     1   (NN,MREF)
       BRIGHT = LOCINT*(UMA(II) + U(II) *COSGAM)
   20 RETURN
C      IF DESCRIMINATE NEGATIVE (X IMAGINARY) WRITE MSG AND RETURN ZERO
  22   BRIGHT = 0.
       WRITE (6,900) Q2,Y,Z,II,PHASE
       GO TO 20
  900  FORMAT(' DESC.=',E17.8,' AT',2E17.8,' ON STAR',I3,' AT PHASE'
     1   ,F8.5)
       END
```

****

```
       FUNCTION ECLINT (N,YH,YL)
C GAUSSIAN INTEGRATION OVER ECLIPSED AREA OF STAR N (1 OR 2)
C ECLIPSED AREA IS THAT COMMON TO TWO INTERSECTING ELLIPSES
C YH AND YL ARE THE Y LIMITS OF THE INTERSECTION
       COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
       COMMON /GAUSS/ WT(16),X(16),L,XC(16)
       COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
       MREF = 0
       NREF = 3
       M = N
       DIF = 0.5*(YH - YL)
       SUM = 0.5*(YH + YL)
       SUM2 = 0.
       DO 100  J=1,L
       SUM1 = 0.
       Y = DIF*X(J) + SUM
```

```
C 'LIMITZ' OBTAINS THE Z LIMITS (ZH,ZL) FOR GIVEN Y COORDINATE.
      CALL LIMITZ(M,Y,ZH,ZL)
      YY = Y
       IF (M .EQ. 2)  YY = Y-DCHI
    6 CONTINUE
      DIFZ = 0.5*(ZH - ZL)
      SUMZ = 0.5*(ZH + ZL)
       DO 10  I=1,L
       Z = DIFZ*X(I) + SUMZ
C 'BRIGHT' CALCULATES BRIGHTNESS OF STAR N AT POINT (YY,Z)
   10  SUM1 = SUM1 + WT(I)*BRIGHT(YY,Z,M)
  100  SUM2 = SUM2 + WT(J)*DIFZ*SUM1
      ECLINT = DIF*SUM2
      RETURN
      END
```

```
*****************************************************************************
```

```
      FUNCTION ENERGY(N)
C   CALCULATES TOTAL STELLAR RADIATION OF STAR N (1 OR 2)
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      COMMON /GAUXX/ WT(16),X(16),L,XC(16)
      M = N
      AX = 0.5*AA(M)
      BX = 0.5*BB(M)
      SUM2 = 0.
      DO 100  J=1,L
      SUM1 = 0.
      XJ = X(J) + 1.
      XJS = 0.5*SQRT(4. - XJ**2)
      XX = AX*XJ
      DO 10  I=1,L
      YY = BX*XJS*(X(I) + 1.)
   10 SUM1 = SUM1 + WT(I)*OUTPUT(XX,YY,M)
  100 SUM2 = SUM2 + WT(J)*XJS*SUM1
      ENERGY = 8.*AX*BX*SUM2
      RETURN
      END
```

```
*****************************************************************************
```

```
      SUBROUTINE GEOMET
C     CALCULATE GEOMETRICAL FACTORS BASED ON ORBITAL ELEMENTS
      COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1 EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1  ,IREF,NINC,LST
      COMMON/TRIG/ SINI,COSI,SISQ,CISQ,SINJ(2),COSJ(2),SINT(2),COST(2)
     1     ,COTH(2)
      COMMON/CONST/ PI,TWOPI,HALPI
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1  ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2  ,STAR1,STAR2,STAR3
      EQUIVALENCE (ISUM,IFSPH)
      REAL INCL
      IF ((IREF .EQ. 2) .AND. (LST .LT. 11))  GO TO 5
   52 AA(1) = A*RNOT
      AA(2) = AA(1)*RATIOK
      BB(1) = AA(1)*ELLIP
      BB(2) = AA(2)*ELLIPB
      CC(1) = BB(1)*ELLIP*(1.+EPSI)
      CC(2) = BB(2)*ELLIPB*(1.+EPSIB)
C  CHECK FOR BAD INPUT DATA
      IF ((AA(1) .LE. 0) .OR. (AA(2) .LE. 0))   GO TO 777
      IF ((BB(1) .LE. 0) .OR. (BB(2) .LE. 0))   GO TO 777
      IF ((CC(1) .LE. 0) .OR. (CC(2) .LE. 0))   GO TO 777
      IF ((INCL .LE. 0) .OR. (PERIOD .LE. 0))  GO TO 777
      SINI = SIN(INCL)
      SISQ = SINI**2
      CISQ = 1. - SISQ
      COSI = SQRT(CISQ)
      ANG = TILT(1) + INCL
      SINJ(1) = SIN(ANG)
      COSJ(1) = COS(ANG)
      ANG = TILT(2) + INCL
      SINJ(2) = SIN(ANG)
      COSJ(2) = COS(ANG)
      ISUM = 0
C     CHECK FOR SPHERICAL STARS AND SET ISUM ACCORDINGLY
      IF (((1. - ELLIP) .LE. 1.E-6) .AND. (ABS(EPSI) .LE. 1.E-6))
     1 ISUM = 1
      IF (((1. - ELLIPB) .LE. 1.E-6) .AND. (ABS(EPSIB) .LE. 1.E-6))
     1 ISUM = ISUM + 2
      CALL ORBITA
```

```
      QINT(1) = 1.
  5    CONTINUE
      QINT(2) = RATIOJ
       UMA(1) = 1. - UA
       UMA(2) = 1. - UB
       VMA(1) = 1. - VA
       VMA(2) = 1. - VB
       WMA(1) = WA/(TWOPI*(1. - .5*UA))
       WMA(2) = WB/(TWOPI*(1. - .5*UB))
      T = TCONJ + PERIOD/4.
      CALL ORBITB(T)
       DO 6  I=1,2
       CSQ2 = COTH(I)**2
       SSQ2 = 1. - CSQ2
       SJSQ = SINJ(I)**2
       CJSQ = COSJ(I)**2
  6    RBAR(I) = 1./SQRT(SJSQ*(SSQ2/AA(I)**2 + CSQ2/BB(I)**2) +
     1      CJSQ/CC(I)**2)
       CALL PARAM
       NINC = 3
       IF (IREF .EQ. 1)  CALL ZONES
       STAR1 = TOTINT(1)
       STAR2 = TOTINT(2)
       TOTAL = STAR1 + STAR2 + STAR3
       NINC = 0
       RETURN
 777  WRITE (6,977) INCL,PERIOD,ELLIP,(AA(I),BB(I),CC(I),I=1,2)
       STOP
 977   FORMAT ('0*** ABORT ***    ***    ***'/(3F15.6))
       END

*******************************************************************************

        SUBROUTINE GRID(N)
C  SETS INTEGRATION GRID SIZE
C   N=1 FOR VERY COARSE GRID (INITIAL CONVERGENCE)
C   N=2 FOR INTERMEDIATE GRID (PREDICTION AND FINAL CONVERGENCE)
C   N=3 FOR FINE GRID (DETAILED PREDICTION ONLY)
C     THE FINE GRID IS ALWAYS SET UP FOR USE FOR ATMOSPHERIC ECLIPSES
        COMMON /GAUSS/ WT(16),X(16),L,XC(16)
        COMMON /GAUXX/ WTX(16),XX(16),LX,XCX(16)
C  A,C,E  ARE THE GAUSS WEIGHTS
C  B,D,F  ARE  THE GAUSS COORD.
```

```
      REAL*8 C(6)       /.17132449,.36076157,.46791393,.46791393
     1 ,.36076157,.17132449/,     D(6)    /-.93246951,-.66120939
     2 ,-.23861919,.23861919,.66120939,.93246951/
      REAL*8   A(16)     /.0271524594,.0622535239,.0951585117,.124628971
     1 ,.149595989,.169156519,.182603415,2*.18945061,.182603415
     2 ,.169156519
     3 ,.149595989,.124628971,.0951585117,.0622535239,.0271524594/
     4     ,B(16)     /-.989400935,-.944575023,-.865631202,-.755404408
     5 ,-.617876244,-.458016778,-.281603551,-.0950125098,.0950125098
     6 ,.281603551,.458016778,.617876244,.755404408,.865631202
     7 ,.944575023,.989400935/
      REAL*8 E(4)/.347854845,.652145155,.652145155,.347854845/
     1  ,F(4)/ -.86113631,-.33998104,.33998104,.86113631/
      DATA M4/4/, M6/6/, M16/16/
      GO TO (4,6,16), N
C   COARSE GRID
    4 DO 5  I=1,M4
      WT(I) = E(I)
      X(I) =  F(I)
    5 XC(I) = DSQRT(1. - F(I)**2)
      L = M4
      GO TO 40
C    INTERMEDIATE GRID
    6 DO 10  I=1,M6
      WT(I) = C(I)
      X(I)  = D(I)
   10 XC(I) =DSQRT(1. - D(I)**2)
      L = M6
      GO TO 40
C   FINE GRID
   16 DO 20  I=1,M16
      WT(I) = A(I)
      X(I)  = B(I)
   20 XC(I) =DSQRT(1. - B(I)**2)
      L = M16
   40 DO 50  I=1,M16
      WTX(I) = A(I)
      XX(I) = B(I)
   50 XCX(I) =DSQRT(1. - B(I)**2)
      LX = M16
      RETURN
      END
```

****

```
      SUBROUTINE LIMITY(JTAG,YH,YL)
C  DETERMINES INTEGRATION LIMITS FOR INTERSECTING ELLIPSES
C MAIN (INITIAL) ENTRY FOR Y LIMITS
C LIMITZ ENTRY FOR Z LIMITS
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
      COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
      DIMENSION Z(2,2),TOP(2),BOT(2),Q1(2),Q2(2),
     1   Q3(2),Q4(2),Q5(2),Q6(2),Q7(2),ZZ(4),YSQ(2),LTAG(2)
      EQUIVALENCE (Z(1,1),ZZ(1)),(YSQ(1),HOLD)
     LOGICAL QB
      MM = 1
      JS = 0
      AXE = AAXIS(1)
C   IF IREF=2, THEN ARE CALCULATING PARTIAL DERIVITIVES, AND THE SEARCH
C     CAN BE SHORTENED (ELIMINATED IF IT IS A NON-GEOMETRIC PARAMETER)
      IF (IREF .EQ. 1)  GO TO 200
      IF (LST .LT. 11)  GO TO 55
      AXE = YHIGH + 0.03*AAXIS(1)
 200  JS = JTAG
C   UPON ENTRY NORMALLY JTAG=0.  BUT SET TO 1 (OR -1) IF POTENTIALLY
C     SHALLOW ECLIPSE (SIGN TELLS WHICH END TO EXPECT IT ON)
      IF (JS)  201,202,203
 201  F = -.001
      MM = 2
      N = 5
      GO TO 205
 203  F = .001
      N = 5
      GO TO 205
 202  F = 0.1
      N = 7
 205  DEL = -F*AAXIS(1)
     L = 1
     M = 1
     JSET = 0
C   JSET SET TO 1 IF 4 ROOTS FOUND AT ANY POINT
C   JSET SET TO -1 IF TOTAL/ANNULAR ECLIPSE IS POSSIBLE
C  IF YL IS ANTICIPATED TO BE NEAR EDGE, START SEEKING LOW ROOT FIRST
      Y = SIGN(AXE,F)
     DO 5  I=1,2
```

```
      Q1(I) = 0.5/PARA(14,I)
      Q2(I) = -PARA(15,I)*Q1(I)
      Q3(I) = PARA(15,I)**2 - 4.*PARA(14,I)*PARA(13,I)
    5 Q4(I) = -4.*PARA(14,I)*PARA(16,I)
C   N MUST BE AN ODD INTEGER TO END SCAN INSIDE STAR
    4 DO 30  K=1,N
      KK = 0
    6 Q8 = Y - DCHI
      Q7(2) = Q8*Q2(2) + ECHI
      Q5(2) = Q3(2)*Q8*Q8 + Q4(2)
      IF (Q5(2))  8,7,7
    7 Q7(1) = Y*Q2(1)
      Q5(1) = Q3(1)*Y*Y + Q4(1)
      IF (Q5(1))  8,9,9
    8 IF (L-1)  24,24,25
C   IF L=2, THEN HAVE JUST PASSED OFF STAR EDGE
    9 Q6(1) = Q1(1)*SQRT(Q5(1))
      Q6(2) = Q1(2)*SQRT(Q5(2))
      Z(1,1) = Q7(1) + Q6(1)
      Z(1,2) = Q7(1) - Q6(1)
      Z(2,1) = Q7(2) + Q6(2)
      Z(2,2) = Q7(2) - Q6(2)
      JSET = 1
C   ALL ROOTS HAVE BEEN CALCULATED, NOW SIZE THEM
      IF (Z(1,1) - Z(1,2))  11,12,12
   11 TOP(1) = Z(1,2)
      BOT(1) = Z(1,1)
      GO TO 13
   12 TOP(1) = Z(1,1)
      BOT(1) = Z(1,2)
   13 IF (Z(2,1) - Z(2,2))  14,15,15
   14 TOP(2) = Z(2,2)
      BOT(2) = Z(2,1)
      GO TO 16
   15 TOP(2) = Z(2,1)
      BOT(2) = Z(2,2)
C   UPPER/LOWER ESTABLISHED FOR EACH FUNCTION, SET UP AND TEST TRUTH TABL
   16 IF (TOP(1) - TOP(2))  17,18,18
   17 I=2
      J=1
      GO TO 20
   18 I=1
      J=2
C   INDEX I FOR UPPER FUNCTION (HAS LARGEST ROOT), J FOR OTHER
   20 QB = .TRUE.
```

```
      LTAG(M) = J
      IF (BOT(I) .GT. TOP(J))    QB = .FALSE.
       IF (L .EQ. 2)  GO TO 23
C  L IS FORWARD/BACKWARD SWITCH (1 TO SEE INTERMESH, 2 TO SEEK NON MESH)
   22 IF (QB)    GO TO 25
      GO TO 24
   23 IF (.NOT. QB)  GO TO 25
C STEP SCAN LINE BY ONE INCREMENT
  24    IF (ABS(Y) .GT. 1.1*AAXIS(1))   GO TO 65
      Y = Y + DEL
       IF (K .GT. 1)  KK = KK+1
       IF (KK .LE. 10)  GO TO 6
C   KK EXCEEDS 10 ONLY IF NO LONGER SENSITIVE TO DEL
C  INTERSECTION OR EDGE PASSED, REVERSE SCAN AT 0.1 INTERVAL
   25 DEL = -0.1*DEL
      Y = Y + DEL
       L = 3 - L
C  IF L=1 SET IT TO 2, IF L=2 SET IT TO 1
   30 CONTINUE
      IF(BOT(I) .GE. BOT(J))  LTAG(M) = 3
       IF (MM .EQ. 2)  GO TO 50
C  WHEN MM=1, HAVE HIGH Y ROOT.  WHEN MM=2, HAVE LOW Y ROOT.
   32 YHIGH = Y
       IF (M .EQ. 2)  GO TO 53
 33    L = 1
      M = 2
       MM = 3-MM
       DEL = F*AAXIS(1)
       IF (JS)  40,34,40
  34   Y = -AXE
      GO TO 4
C  WHEN ECLIPSE IS SHALLOW, START SEARCH FOR 2ND LIMIT NEAR 1ST
  40   Y = Y - 0.2*SIGN(AAXIS(1),F)
       GO TO 4
   50 YLOW = Y
       IF (M .EQ. 1)  GO TO 33
  53   CONTINUE
      IF ((LTAG(1) .EQ. 3) .OR. (LTAG(2) .EQ. 3))  GO TO 55
      IF (LTAG(1) .EQ. LTAG(2))  GO TO 60
   55 CONTINUE
       YH = YHIGH
       YL = YLOW
       JTAG = JSET
      RETURN
C  APPARENTLY NO INTERSECTION SO ECLIPSE MAY BE TOTAL OR ANNULAR
```

```
   60    JSET = -1
         GO TO 55
C    NO INTERSECTION - FALSE ALARM - HAVE RUN OFF STAR EDGE
 65      JSET = 0
         GO TO 55
C
        ENTRY LIMITZ(NX,YY,ZHIGH,ZLOW)
        Y = YY
        YSQ(1) = Y*Y
        Q8 = Y - DCHI
        YSQ(2) = Q8*Q8
        Q7(1) = Y*Q2(1)
        Q7(2) = Q8*Q2(2) + ECHI
        Q9 = 0.
         IF (NX .EQ. 2)    Q9 = -ECHI
        DO 100  I=1,2
        Q6(I) = Q1(I)*SQRT(YSQ(I)*Q3(I) + Q4(I))
        Z(I,1) = Q7(I) + Q6(I)
  100 Z(I,2) = Q7(I) - Q6(I)
C   Z LIMITS ARE INNER TWO Z VALUES. DISCARD HIGHEST AND LOWEST BY SORT
        DO 150  J=1,4
         JJ = 4-J
        JEXIT = 0
        DO 110  I=1,JJ
        IF (ZZ(I) - ZZ(I+1))  110,110,105
  105    HOLD = ZZ(I)
        ZZ(I) = ZZ(I+1)
        ZZ(I+1) = HOLD
        JEXIT = 1
  110 CONTINUE
        IF (JEXIT)  160,160,150
  150 CONTINUE
  160 ZHIGH = ZZ(3) + Q9
        ZLOW = ZZ(2) + Q9
        RETURN
        END

**********************************************************************

         SUBROUTINE LSQS(A,C,D,W,M,N,ERR,DE,IX)
C        USES MATRIX MULTIPLICATION SUBROUTINE 'MAMUL'
C        A = OBSERVATION EQUATION MATRIX
C        D = OBSERVATION EQUATION VECTOR (RESIDUALS UPON EXIT)
```

```
C       C = ANSWERS VECTOR (EXIT)
C       W = WEIGHT VECTOR. 1ST IS ZERO FOR UNIT WEIGHTING
C       M = NUMBER OF UNKNOWNS
C       N = NUMBER OF OBSERVATION EQUATIONS
C       ERR = R.M.S. ERROR + INDIVIDUAL ERRORS (EXIT)
C       DE = DETERMINANT OF NORMAL EQUATIONS (EXIT)
C       IX = 0 USUALLY. NOT 0 IF SAME OBSERVATION EQUATION MATRIX
C       DIMENSIONS SHOULD BE ......
C       A(N,M), AT(M,N), B(M,M)
C       C(M), D(N), DC(N), W(N), WT(N), ERR(M+1)
C       IPIVOT(M),INDEX(M,2),PIVOT(M)
        DIMENSION A(100,17),AT(17,100),B(17,17)
        DIMENSION DC(100),ERR(18),IPIVOT(17),INDEX(17,2),PIVOT(17)
        DIMENSION C(17),D(100),W(100)
        COMMON/OBS/ DUMMY(303),WT(101)
        EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
 1000   IF (IX) 1200,1010,1200
CCCC            HERE ONLY IF IX IS ZERO
CCCC            APPLY WEIGHTS IF FIRST ONE IS GREATER THAN ZERO
 1010   IF (W(1)) 1040,1040,1020
 1020   DO 1030 J=1,N
        WT(J) = SQRT(W(J))
        D(J) = D(J)*WT(J)
        DO 1030 I=1,M
 1030   A(J,I) = A(J,I)*WT(J)
CCCC            GET TRANSPOSE OF OBSERVATION EQUATION MATRIX
 1040   DO 1050 I=1,M
        DO 1050 J=1,N
 1050   AT(I,J) = A(J,I)
CCCC            FORM NORMAL EQUATION MATRIX
        CALL MAMUL(AT,A,B,N,M,M)
        CALL MAMUL(AT,D,C,N,M,1)
CCCC            SOLVE NORMAL EQUATIONS
C       MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
C       INITIALIZATION
C
   10   DE = 1.0
   15   DO 20 J=1,M
   20   IPIVOT(J)=0
   30   DO 550 I=1,M
C
C       SEARCH FOR PIVOT ELEMENT
C
   40   AMAX=0.0
```

```
   45   DO 105 J=1,M
   50   IF (IPIVOT(J)-1) 60, 105, 60
   60   DO 100 K=1,M
   70   IF (IPIVOT(K)-1) 80, 100, 740
   80   IF (ABS(AMAX) - ABS(B(J,K))) 85,100,100
   85   IROW=J
   90   ICOLUM=K
   95   AMAX = B(J,K)
  100   CONTINUE
  105   CONTINUE
  110   IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C       INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130   IF(IROW-ICOLUM) 140, 260, 140
  140   DE = -DE
  150   DO 200 L=1,M
  160   SWAP = B(IROW,L)
  170   B(IROW,L) = B(ICOLUM,L)
  200   B(ICOLUM,L) = SWAP
  220   SWAP = C(IROW)
  230   C(IROW) = C(ICOLUM)
  250   C(ICOLUM) = SWAP
  260   INDEX(I,1)=IROW
  270   INDEX(I,2)=ICOLUM
  310   PIVOT(I) = B(ICOLUM,ICOLUM)
  320   DE = DE*PIVOT(I)
C
C       DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330   B(ICOLUM,ICOLUM) = 1.0
  340   DO 350 L=1,M
  350   B(ICOLUM,L) = B(ICOLUM,L)/PIVOT(I)
  370   C(ICOLUM) = C(ICOLUM)/PIVOT(I)
C
C       REDUCE NON-PIVOT ROWS
C
  380   DO 550 L1=1,M
  390   IF(L1-ICOLUM) 400, 550, 400
  400   T = B(L1,ICOLUM)
  420   B(L1,ICOLUM) = 0.0
  430   DO 450 L=1,M
  450   B(L1,L) = B(L1,L) - B(ICOLUM,L)*T
  500   C(L1) = C(L1) -C(ICOLUM)*T
  550   CONTINUE
```

```
C
C       INTERCHANGE COLUMNS
C
  600   DO 710 I=1,M
  610   L = M+1-I
  620   IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630   JROW=INDEX(L,1)
  640   JCOLUM=INDEX(L,2)
  650   DO 705 K=1,M
  660   SWAP = B(K,JROW)
  670   B(K,JROW) = B(K,JCOLUM)
  700   B(K,JCOLUM) = SWAP
  705   CONTINUE
  710   CONTINUE
CCCC            FORM RESIDUALS
  740   DO 1070 J=1,N
        SWAP = 0.0
        DO 1060 K=1,M
 1060   SWAP = SWAP + A(J,K)*C(K)
 1070   DC(J) = SWAP
        SWAP = 0.0
        DO 1080 J=1,N
        D(J) =D(J) - DC(J)
 1080   SWAP = SWAP + D(J)**2
        FNM = N-M
        IF (FNM) 1090,1090,1100
 1090   ERR(1) = 0.0
        GO TO 1110
 1100   ERR(1) = SQRT(SWAP/FNM)
CCCC            CALCULATE INDIVIDUAL ERRORS
 1110   DO 1120 I=1,M
 1120   ERR(I+1) = ERR(1)*SQRT(ABS(B(I,I)))
 1150   RETURN
CCCC    HERE IF IX NOT 0. FOR ITERATIVE LSQ, NORM EQN MTRX IS SAME
 1200   IF (W(I)) 1230,1230,1210
 1210   DO 1220 J=1,N
 1220   D(J) = D(J)*WT(J)
 1230   CALL MAMUL(AT,D,DC,N,M,1)
        DO 1250 J=1,M
        SWAP = 0.0
        DO 1240 K=1,M
 1240   SWAP = SWAP + B(J,K)*DC(K)
 1250   C(J) = SWAP
        GO TO 740
        END
```

```
*************************************************************************

        FUNCTION LUMC(T)
CALCULATE SYSTEM LUMINOSITY AT TIME T
        COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
      1    ,IREF,NINC,LST
        COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
      1    ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
      2    ,STAR1,STAR2,STAR3
        EQUIVALENCE (JJ,JTYPE),(KK,KSTAR)
       REAL LUMC
       CALL ORBITB(T)
CALC ORBITAL PARAMETERS
       CALL PARAM
CALC ELLIPSE PARAMETERS
C
CALC LUMINOSITY
       CALL SCREEN(YH,YL)
        LL = 3 - KK
C  DETERMINE TYPE OF ECLIPSE (JJ,KK SET BY SCREEN)
C     JJ: 1=ANNULAR, 2=TOTAL, 3=PARTIAL, 4=ATMOSPHERIC
C     KK: 1=STAR 1, 2=STAR 2, 3=NO ECLIPSE
       GO TO (20,20,10), KK
C  NO ECLIPSE
   10  ALUM = TOTINT(1) + TOTINT(2)
       GO TO 50
   20 GO TO (35,25,45,55), JJ
C  TOTAL ECLIPSE
   25  ALUM = TOTINT(LL)
       GO TO 50
C  ANNULAR ECLIPSE
  35    ECLIPS = ANNECL(KK)
C  IF THERE IS NO ATMOSPHERE, USE TOTINT - OTHERWISE ATMECL
   36   IF (SCALE(LL)) 39,39,37
   39   TOTKK = TOTINT(KK)
        GO TO 38
  37    TOTKK = ATMECL(KK)
  38    ALUM = TOTINT(LL) + TOTKK - ECLIPS
       GO TO 50
C  PARTIAL ECLIPSE
  45    ECLIPS = ECLINT(KK,YH,YL)
        GO TO 36
```

```
C  ATMOSPHERIC ECLIPSE ONLY
   55   ALUM = TOTINT(LL) + ATMECL(KK)
  50    LUMC = (ALUM + STAR3)/TOTAL
        RETURN
       END
```

```
*************************************************************************
```

```
        SUBROUTINE MAMUL(A,B,C,NCA,NRA,NCB)
        DIMENSION A(17,100),B(100,17),C(17,17)
        DO 300 I=1,NCB
        DO 300 J=1,NRA
        SUMC = 0.0
        DO 200 K=1,NCA
  200   SUMC = SUMC + A(J,K)*B(K,I)
  300   C(J,I) = SUMC
        RETURN
        END
```

```
*************************************************************************
```

```
       SUBROUTINE ORBITA
CALCULATES BASIC QUANTITIES FOR ORBITAL MECHANICS CALCULATIONS
       COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1  EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
        COMMON/CONST/ PI,TWOPI,HALPI
        COMMON/ORBIT/ E,TO,Q,MU
       REAL INCL
       REAL MU
       MU = TWOPI/PERIOD
       ESQ = ESINW**2 + ECOSW**2
       IF (ESQ)  20,20,5
     5 E = SQRT(ESQ)
       Q = SQRT(1. - ESQ)
        IF (ECOSW) 6,25,6
    6   CONTINUE
       Q1 = 1. + ESINW
        Q2 = ARCOS((ESINW + ESQ)/(E*Q1))
       Q3 = ECOSW*Q/Q1
       IF (Q3)  10,10,12
    10 Q4 = Q3 + Q2 + TWOPI
```

```
      GO TO 15
   12 Q4 = Q3 - Q2
   15 TO = TCONJ + Q4/MU
   18 RETURN
CIRCULAR ORBIT
   20 Q = 1.
      E = 0
      TO = TCONJ
      GO TO 18
  25   Q4 = TWOPI
       GO TO 15
      END
```

****

```
      SUBROUTINE ORBITB(T)
CALCULATES ORBITAL MECHANICS PORTION - R AND LONGITUDE FOR GIVEN TIME
      COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1  EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
      COMMON/CONST/ PI,TWOPI,HALPI
      COMMON/ORBIT/ E,TO,Q,MU
      COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
      COMMON/TRIG/ SINI,COSI,SISQ,CISQ,SINJ(2),COSJ(2),SINT(2),COST(2)
     1      ,COTH(2)
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1    ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1    ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2    ,STAR1,STAR2,STAR3
      DIMENSION CHIFCN(2)
      REAL MU,INCL,M,MP
      M = MU*(T - TO)
C    FORCE MEAN ANOMALLY (M) TO LIE BETWEEN + OR - 2*PI
    5 IF (ABS(M) .LT. TWOPI)  GOTO 10
      M = SIGN((ABS(M) - TWOPI),M)
      GO TO 5
C    GET FIRST GUESS FOR SOLUTION OF KEPLER EQUATION
   10 IF(E .GT. 0.75)  GOTO 15
      ECC = M
C    IF CIRCULAR (E=0) SOLUTION IS TRIVIAL
      IF (E)  40,40,16
   15 ECC = 0.5*(M + SIGN(PI,M))
C    PROCEDURE FOR SOLUTION OF KEPLER EQUATION
```

```
   16 COSE = COS(ECC)
      Q1 = 1. - E*COSE
      SINE = SIN(ECC)
      DELE = (E*SINE - ECC + M)/Q1
C   EQUATION SOLVED WHEN CORRECTION TO ECCENTRIC ANOMALLY LESS THAN
C           10**-6
       IF (ABS(DELE) .LT. 1.0E-6)  GO TO 20
      ECC = ECC + DELE
      GO TO 16
CONVERGED
C   R=RADIUS VECTOR, V=TRUE ANOMALLY, W=OMEGA=LONGITUDE OF PERIASTRON
C   THETA = ORBITAL LONGITUDE
   20 R = Q1
      COSV = (COSE - E)/Q1
      SINV = (SINE*Q)/Q1
       SINTH = (ESINW*SINV - ECOSW*COSV)/E
       COSTH = (ESINW*COSV + ECOSW*SINV)/E
       THETA = ARCOS(COSTH)
       IF (SINTH)  25,28,28
   25 THETA = -(THETA - TWOPI)
  28   V = ARCOS(COSV)
      IF (SINV)  30,35,35
   30 V = -(V - TWOPI)
  35   CONTINUE
       W = ARCOS(ECOSW/E)
      IF (ESINW)  36,37,37
   36 W = -(W - TWOPI)
 37    R = R*RNOT
       Q6 = SQRT(SISQ*SINTH**2 + CISQ)
       IF (ABS(SINTH) .GE. 1.E-6)  GO TO 102
       CHIFCN(1) = SINTH
       CHIFCN(2) = COSTH
       CHI = HALPI - THETA
       GO TO 104
 102   CHIFCN(1) = SINTH/Q6
       CHIFCN(2) = COSTH*COSI/Q6
       CHI = ARSIN(CHIFCN(2))
       IF (CHIFCN(1))  103,104,104
 103   CHI = -(CHI - TWOPI)
C   DELTA = APPARENT SEPARATION OF CENTERS
 104   DELTA = Q6*R
       DCHI = R*SINTH
       ECHI = R*COSTH*COSI
       DO 110  J=1,2
       MP = M + SLIP(J)
```

```
      COTH(J) = COS(V - MP)
      THETAP(J) = W - HALPI + MP
      SINT(J) = SIN(THETAP(J))
      COST(J) = COS(THETAP(J))
 110  CONTINUE
      PHASE = (T - TCONJ)/PERIOD
     RETURN
  40 THETA = M
CIRCULAR ORBIT
     R = 1.
      W = HALPI
      SINTH = SIN(THETA)
      COSTH = COS(THETA)
      V = M
     GOTO 37
     END
```

****

```
      FUNCTION OUTPUT(X,Y,N)
C   CALCULATES ENERGY OUTPUT OF STAR N AT COORD (X,Y) ON SURFACE
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1  EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
      DIMENSION V(2)
      EQUIVALENCE (V(1),VA)
      M = N
      Q1 =  (CC(M)/AA(M))**2 - 1.
      Q2 =  (CC(M)/BB(M))**2 - 1.
      XSQ = X**2
      YSQ = Y**2
   5  R = SQRT(CC(M)**2 - XSQ*Q1 - YSQ*Q2)
      XX = XSQ/AA(M)**2
      YY = YSQ/BB(M)**2
      G = SQRT((1.+XX*Q1+YY*Q2)/(1.-XX-YY))
  10  OUTPUT = QINT(M)*(VMA(M)+V(M)*R/RBAR(M))*G
      RETURN
      END
```

****

```
      SUBROUTINE PARAM
CALCULATES PARAMETERS FOR ELLIPSOIDAL STARS
C   II=1(2) IF STAR 1(2) SPHERICAL
C   II=0 IF NO STAR SPHERICAL     =3 IF BOTH STARS SPHERICAL
      COMMON/CONST/ PI,TWOPI,HALPI
      COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
      COMMON/TRIG/ SINI,COSI,SISQ,CISQ,SINJ(2),COSJ(2),SINT(2),COST(2)
     1     ,COTH(2)
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
      DIMENSION X(6,6),ASQ(2),BSQ(2),CSQ(2),ABSQ(2),ACSQ(2),BCSQ(2)
      EQUIVALENCE (ASQ(1),X(2,1)),(BSQ(1),X(4,1)),(CSQ(1),X(3,2)),
     1   (ABSQ(1),X(5,2)),(ACSQ(1),X(4,3)),(BCSQ(1),X(5,4))
    4 IX = IFSPH + 1
      GO TO (41,42,43,100), IX
    5 DO 10  I=IL,IU
      ASQ(I) = AA(I)**2
      BSQ(I) = BB(I)**2
      CSQ(I) = CC(I)**2
      ABSQ(I) = ASQ(I)*BSQ(I)
      ACSQ(I) = ASQ(I)*CSQ(I)
   10 BCSQ(I) = BSQ(I)*CSQ(I)
   15 DO 30  I=IL,IU
       SIS = SINJ(I)**2
       CIS = COSJ(I)**2
       SCI = SINJ(I)*COSJ(I)
       STSQ = SINT(I)**2
       CTSQ = COST(I)**2
       SCT = SINT(I)*COST(I)
      Q1 = ACSQ(I)*STSQ
      Q2 = BCSQ(I)*CTSQ + Q1
       PARA(1,I) = Q2*SIS + CIS*ABSQ(I)
      PARA(2,I) = BCSQ(I)*STSQ + ACSQ(I)*CTSQ
       PARA(3,I) = Q2*CIS + SIS*ABSQ(I)
      Q3 = (BCSQ(I) - ACSQ(I))*SCT
       PARA(4,I) = SINJ(I)*Q3
       PARA(5,I) = COSJ(I)*Q3
      PARA(6,I) = (Q2 - ABSQ(I))*SCI
```

```
CROSS PRODUCTS
      DO 20  J=1,6
      DO 18  K=J,6
   18 X(J,K) = PARA(J,I)*PARA(K,I)
   20 CONTINUE
      PARA(7,I) = X(1,1) + X(4,4) + X(6,6)
      PARA(8,I) = X(2,2) + X(4,4) + X(5,5)
      PARA(9,I) = X(3,3) + X(5,5) + X(6,6)
      PARA(10,I) = X(1,4) + X(2,4) + X(5,6)
      PARA(11,I) = X(1,6) + X(3,6) + X(4,5)
      PARA(12,I) = X(2,5) + X(3,5) + X(4,6)
      PARA(13,I) = X(4,4) - X(1,2)
      PARA(14,I) = X(6,6) - X(1,3)
      PARA(15,I) = X(4,6) - X(1,5)
      PARA(16,I) = PARA(1,I)*ASQ(I)*BCSQ(I)
      PARA(17,I) = -PARA(13,I)/PARA(16,I)
      PARA(18,I) = -PARA(14,I)/PARA(16,I)
      PARA(19,I) = -PARA(15,I)/PARA(16,I)
      Q4 = SQRT((PARA(17,I) - PARA(18,I))**2 + (2.*PARA(19,I))**2)
      Q5 = PARA(17,I) + PARA(18,I)
      AAXIS(I) = SQRT(2./(Q5 - Q4))
      BAXIS(I) = SQRT(2./(Q5 + Q4))
      Q7 = PARA(13,I) - PARA(14,I)
       TAN2P = 2.*PARA(15,I)/Q7
       IF (ABS(TAN2P) .GT. 1.0E-6)  GO TO 26
C   SPECIAL CASE FOR COT2P NEAR OR EQUAL INFINITY
C   PHI IS NEARLY ZERO UNLESS STAR ON END (Q7 NEGATIVE)
       IF (Q7 .LT. 0.)  GO TO 25
       PHI(I) =  0.5*TAN2P
      SINP(I) = PHI(I)
      COSP(I) = 1. - 0.5*PHI(I)**2
      GO TO 27
 25    COSP(I) = 0.5*TAN2P
       SINP(I) = 1. - 0.5*COSP(I)**2
       PHI(I) = 0.5*PI - COSP(I)
       GO TO 27
 26    COT2P = 1./TAN2P
      Q8 = SQRT(1. + COT2P**2)
       TANP = -COT2P + SIGN(Q8,PARA(15,I))
      PHI(I) = ATAN(TANP)
      COSP(I) = COS(PHI(I))
      SINP(I) = COSP(I)*TANP
   27 CONTINUE
       ALPHA(I) =   CHI - PHI(I)
   30 CONTINUE
```

```
      IF (IL-IU)  35,102,35
  35   RETURN
C   SET UP FOR SPHERICAL/NON-SPHERICAL STARS
   41 IL = 1
      IU = 2
      GO TO 5
   42 IL = 2
      IU = 2
      GO TO 5
   43 IL = 1
      IU = 1
      GO TO 5
  100 IL = 1
      IU = 2
      GO TO 110
 102   IL = IFSPH
       IU = IFSPH
C
C  SPECIAL CALCULATION FOR SPHERICAL STARS
  110 DO 120  I=IL,IU
      Q2 = AA(I)**2
      Q1 = AA(I)**4
      PARA(1,I) = Q1
      PARA(2,I) = Q1
      PARA(3,I) = Q1
      PARA(7,I) = Q1**2
      PARA(8,I) = Q1**2
      PARA(9,I) = Q1**2
      PARA(13,I) = -Q1**2
      PARA(14,I) = -Q1**2
      PARA(16,I) = Q2*Q1**2
      PARA(17,I) = -1./Q2
      PARA(18,I) = -1./Q2
      PARA(4,I) = 0.
      PARA(5,I) = 0.
      PARA(6,I) = 0.
      PARA(10,I) = 0.
      PARA(11,I) = 0.
      PARA(12,I) = 0.
      PARA(15,I) = 0.
      PARA(19,I) = 0.
      PHI(I) = 0.
      SINP(I) = 0.
      COSP(I) = 1.
       ALPHA(I) =    CHI
```

```
      AAXIS(I) = AA(I)
      BAXIS(I) = AA(I)
  120 CONTINUE
      GO TO 35
       END

************************************************************************

       FUNCTION REFL(N)
CALCULATES REFLECTED LIGHT  ONTO STAR N AT COORD X,Y,Z
      COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1  EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
       COMMON/TRIG/ SINI,COSI,SISQ,CISQ,SINJ(2),COSJ(2),SINT(2),COST(2)
     1      ,COTH(2)
       COMMON/RINT/X,Y,Z,RAD
       COMMON/CONST/ PI,TWOPI,HALPI
       COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1    ,IREF,NINC,LST
       COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1    ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2    ,STAR1,STAR2,STAR3
       COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1    ,BAXIS(2),PHASE,PHI(2),THETAP(2)
       COMMON/TABLE/ENDINT(2)
       DIMENSION U(2),V(2),W(2)
       EQUIVALENCE (U(1),UA),(V(1),VA),(W(1),WA)
      REAL INCL,LAMB,LAMP
    3   II = N
        IF (W(II)) 5,36,5
    5   STP = SINT(II)
        CTP = COST(II)
       IF (II .NE. 2)  GO TO 4
        STP = -STP
        CTP = -CTP
     4 CONTINUE
       XX = X*SINI*CTP + Y*STP + Z*COSI*CTP
       YY = -X*SINI*STP + Y*CTP - Z*COSI*STP
       ZZ = -X*COSI + Z*SINI
        RINV = 1./RAD
        COSL = XX*RINV
        COSM = YY*RINV
        COSN = ZZ*RINV
COSL,COSM,COSN = DIRCOS NORMAL TO STAR AT X,Y,Z
```

```
      L = 3 - II
  100  CONTINUE
C   SHORTCUT VERSION - APPROXIMATES LIGHT REFLECTED ONTO N AT X,Y,Z
       XDEL = R*COTH(L) - XX
       YDEL = -YY
       ZDEL = -ZZ
       DINV = 1./SQRT(XDEL**2 + YDEL**2 + ZDEL**2)
       LAMB = ARSIN(RBAR(L)*DINV)
       CLP = DINV*(XDEL*COSL + YDEL*COSM + ZDEL*COSN)
       LAMP = HALPI + LAMB -ARCOS(CLP)
       IF (LAMP .LE. 0)  GO TO 36
       IF (LAMP .LT. LAMB)  GO TO 110
       IF (LAMP .LT. 2.*LAMB)  GO TO 120
       AREA = PI*LAMB**2
       GO TO 150
  110  AREA = (LAMB**2)*ARCOS((LAMB-LAMP)/LAMB)-(LAMB-LAMP)*
     1      SQRT(2.*LAMB*LAMP-LAMP**2)
       GO TO 150
 120  LAMP = 2.*LAMB - LAMP
       AREA = (LAMB**2)*(PI-ARCOS((LAMB-LAMP)/LAMB))+(LAMB-LAMP)*
     1      SQRT(2.*LAMB*LAMP-LAMP**2)
 150   C = .38736 + AA(II)*(1.43431*AA(L)-.82442) + CLP*(AA(II)*
     1  (4.9378*AA(L)-.43316)-1.22172)
       IF (C .LT. 0)  C=0
       FM =2.044+AA(II)*(-.170831+AA(L)*(1.231707-AA(L)*9.955083))
       FB=-.065354+AA(II)*(.224935+AA(L)*(-.761696+AA(L)*3.81425))
       REFL = ENDINT(L)*AREA*(1.-.5*U(L))*(C+FB+CLP*FM)
       RETURN
  36   REFL = 0.
       RETURN
       END

*******************************************************************

      SUBROUTINE SCREEN(YH,YL)
C   FOR EACH ORBITAL POSITION, SCREEN STARS TO FIND IF ECLIPSE
C   SET  KK                          SET  JJ
C      =1 STAR 1 ECLIPSED              =1 ANNULAR ECLIPSE
C      =2 STAR 2 ECLIPSED              =2 TOTAL ECLIPSE
C      =3 NO ECLIPSE                   =3 PARTIAL ECLIPSE
C                                      =4 ATMOSPHERIC ECLIPSE
C   YH,YL = RIGHT AND LEFT HAND LIMITS FOR PARTIAL ECLIPSE
       COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
```

```
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
      EQUIVALENCE (JJ,JTYPE),(KK,KSTAR)
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      DIMENSION RV(2)
      II = 0
      JJ = 3
      KK = 1
   20 IF (ABS(THETA) .GT. 1.570796 .AND. ABS(THETA) .LT. 4.712289)
     1   KK = 2
C  KK=1 WHEN STAR 2 ECLIPSES STAR 1.  KK=2 WHEN STAR 1 ECLIPSES STAR 2
      DO 5  I=1,2
      COSA = COS(ALPHA(I))**2
    5 RV(I) = 1./SQRT(COSA/AAXIS(I)**2 + (1.-COSA)/BAXIS(I)**2)
      SEP = RV(1) + RV(2) - DELTA
C  NO ECLIPSE CAN OCCUR IF SEPARATION ECCEEDS SUM OF RADIUS VECTORS
      LL = 3 - KK
      IF (SEP)  6,6,12
C  TEST TO SEE IF INSIDE 10 SCALE HEIGHTS OF ATMOSPHERE
    6 IF (SEP + 10.*RV(LL)*SCALE(LL))  10,10,7
C  SET JJ=4 FOR ATMOSPHERIC ECLIPSE AND NO PHYSICAL ECLIPSE
    7 JJ = 4
      GO TO 50
   10 KK = 3
      RETURN
   12 IF (SEP .LT. 0.1*AAXIS(1))  II = IFIX(SIGN(1.,DCHI))
C  FOR ENTERING LIMITY, SET II=1 IF YH IS POTENTIALLY NEAR LIMB
C                              -1 IF YL IS POTENTIALLY NEAR LIMB
      CALL LIMITY(II,YH,YL)
C  II = 0 WHEN NO SET OF 4 ROOTS FOUND
C  II = 1 FOR INTERSECTION (PARTIAL ECLIPSE)
C  II = -1 FOR POSSIBLE TOTAL/ANNULAR ECLIPSE
      IF (II)  15,10,50
C  HERE TO EXAMINE TOTAL/ANNULAR ECLIPSE CASE
   15 IF (DELTA - ABS(RV(1) - RV(2)))  17,50,50
   17 JJ = 1
      IF (AAXIS(1) - AAXIS(2))  22,23,24
   22 IF (KK-1)  23,23,50
   23 JJ=2
C  SET JJ TO 2 FOR TOTAL ECLIPSE
      GO TO 50
```

```
24    IF (KK-1)  23,50,23
 50 CONTINUE
120  RETURN
     END
```

****

```
      SUBROUTINE SOLVE1(INDEX)
C  DIFFERENTIAL CORRECTOR TO OPERATE WITH ASTROPHYSICAL VARIABLES
      COMMON/ORBE/BLKX(18)
      COMMON/CONST/ PI,TWOPI,HALPI
      COMMON/OBS/ TIME(101),LUM(101),WT(101),CLUM(101)
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      COMMON/VARIB/BLKV(17)
      DIMENSION V(8),VV(17),SA(17),SB(17),SC(17)
      DATA SA     /1.570796,2*.9,2*1.,.9,100.,2*1.,2*1.E5,0,50.,10.
     1   ,0,2*2./
      DATA SB     /.2,-.9,-.9,6*0,2*1.E3,2*0,-5.,3*0/
      DATA SC     /.2,2*-.9,2*0,2*.001,2*0,2*1.E3,0,.01,-5.,3*0/
      DIMENSION BLKS(17),BLKA(17),ERR(18),D(100,17),DELTI(100)
      REAL LUM,LOW,LUMC
      LOGICAL TEST
      INTEGER V,VV
      DATA V    /1,2,3,4,5,12,16,17/
C    V CONTAINS THE INDICES OF THE NON-ASTROPHYSICAL VARIABLES
      DATA VV     /4,5,8,9,10,11,14,15,16,17,1,2,3,6,7,12,13/
C    VV CONTAINS ORDER IN WHICH PARTIALS ARE TO BE CALCULATED.
C      THEY ARE ARRANGED SO THAT THE FIRST 10 ARE ALL NON-GEOMETRIC.
      INDEX = INDEX-1
      IF (INDEX .LE. 0)  GO TO 1720
      INDIC = 0
      DO 250 I=1,NOBS
      IREF = 1
      CALL GEOMET
      CINT = LUMC(TIME(I))
      DELTI(I) = LUM(I) - CINT
      IREF = 2
      NVAR = 0
      DO 200  LST=1,17
```

```
      L = VV(LST)
      BLKS(L) = BLKV(L)
      IF (TEST(L)) GO TO 210
  200 CONTINUE
      IF (NVAR) 1730,1730,205
  205 CONTINUE
  250 CONTINUE
      WRITE (6,732)
      WRITE (6,730) (TIME(J),DELTI(J),J=1,NOBS)
      SUM=0
      DO 260  J=1,NOBS
  260 SUM = SUM + DELTI(J)**2
      GO TO 500
  210 NVAR = NVAR + 1
      CLUMH = CINT
      CLUML = CINT
      NN = 1
      GO TO (401,402,402,404,404,406,407,408,408,410,410,412,413,
     1   414,415,404,404), L
C SUB-BLOCKS FOR EACH VARIABLE
 401  HIGH = BLKS(L)
      LOW  = BLKS(L) - .01
      GO TO 440
 402  HIGH = BLKS(L) + .00125
      LOW  = BLKS(L) - .00125
      GO TO 450
 404  IF (BLKS(L)-.1) 4042,4042,4041
 4041 HIGH = BLKS(L)
      LOW  = BLKS(L) - .1
      GO TO 440
 4042 HIGH = .1
      LOW  = 0.
      GO TO 450
  406 HIGH = 1.025*BLKS(L)
      LOW  = BLKS(L)
      GO TO 4800
  407 HIGH = 1.025*BLKS(L)
      LOW  = BLKS(L)
      GO TO 4800
  408 HIGH = BLKS(L)
      LOW  = BLKS(L) - .025
      GO TO 4400
  410 HIGH = BLKS(L) + 100.
      LOW  = BLKS(L)
      GO TO 4800
```

```
  412   TD = .001*BLKX(18)
        HIGH = BLKS(L) + TD
        LOW  = BLKS(L) - TD
        GO TO 450
  413   HIGH = BLKS(L) + .1
        LOW  = BLKS(L)
        GO TO 4800
  414   D(I,NVAR) = 0.92061*CINT
C  SPECIAL CALC OF PARTIAL DERIV FOR QUADRITURE MAG
        GO TO 200
C THERE IS NO VARIABLE FOR L=15
  415   NVAR = NVAR-1
        GO TO 200
C FORM PARTIAL DERIVITIVES
  440   BLKX(L) = LOW
        CALL GEOMET
        CLUML = LUMC(TIME(I))
        GO TO (453,480), NN
  450   NN = 2
        GO TO 440
  480   BLKX(L) = HIGH
        CALL GEOMET
        CLUMH = LUMC(TIME(I))
  453   D(I,NVAR) = (CLUMH - CLUML)/(HIGH - LOW)
        BLKX(L) = BLKS(L)
        GO TO 200
C SPECIAL TREATMENT FOR ASTROPHYSICAL VARIABLES
 4400   BLKV(L) = LOW
        CALL ASTROX
        CALL GEOMET
        CLUML = LUMC(TIME(I))
        GO TO 4530
 4800   BLKV(L) = HIGH
        CALL ASTROX
        CALL GEOMET
        CLUMH =LUMC(TIME(I))
 4530   D(I,NVAR) = (CLUMH - CLUML)/(HIGH - LOW)
        BLKV(L) = BLKS(L)
        CALL ASTROX
        GO TO 200
C SOLVE OBS EQUATIONS
  500   CALL LSQS (D,BLKA,DELTI,WT,NVAR,NOBS,ERR,DET,0)
        WRITE (6,911) ERR(1),SUM,DET
        J = 0
        DO 511  LST=1,17
```

```
      L = VV(LST)
      IF (TEST(L)) GO TO 520
  510 CONTINUE
  511 CONTINUE
      GO TO 700
  520 J = J+1
      IF (BLKV(L)) 524,523,524
  523 IF (ABS(BLKA(J)) .GT. .25) BLKA(J) = SIGN(.25,BLKA(J))
      GO TO 522
  524 IF (ABS(BLKA(J)/BLKV(L)) .GT. .25)
     1          BLKA(J) = SIGN(0.25*BLKV(L),BLKA(J))
  522 BLKV(L) = BLKV(L) + BLKA(J)
      WRITE (6,910) L,BLKA(J),ERR(J+1),BLKV(L)
      IF (ABS(BLKV(L)) .LT. .001)  GO TO 530
      IF (ABS(BLKA(J)/BLKV(L)) .GT.  .01) INDIC=INDIC+1
C  DROP VARIABLES THAT HAVE VERY SMALL CORRECTIONS
      IF (ABS(BLKA(J)/BLKV(L)) .LT. .0001) TEST(L) = .FALSE.
      GO TO 533
  530 IF (ABS(BLKA(J)) .GT. .0001)  INDIC=INDIC+1
      IF (ABS(BLKA(J)) .LT. .00001)  TEST(L) = .FALSE.
  533 CONTINUE
      IF (L .EQ. 12)  GO TO 612
      IF (BLKV(L) .GT. SA(L))  BLKV(L) = SA(L)
      IF (BLKV(L) .LE. SB(L))  BLKV(L) = SC(L)
      GO TO 510
  612 IF (ABS(BLKA(J)/BLKX(18)) .LT. .1) GO TO 510
      BLKV(L)=BLKV(L)-BLKA(J)+SIGN(.1*BLKX(18),BLKA(J))
      GO TO 510
  700 IF (INDIC .EQ. 0) INDEX=0
      DO 710 I=1,8
  710 BLKX(V(I)) = BLKV(V(I))
      CALL ASTROX
      QUAD = BLKV(14)
 1710 RETURN
 1720 INDEX = -1
      GO TO 1721
 1730 INDEX = -2
 1721 IREF = 1
      DO 1731 J=1,NOBS
      CINT = LUMC(TIME(J))
 1731 DELTI(J) = -2.5*ALOG10(LUM(J)/CINT)
      WRITE (6,731)
      WRITE (6,730) (TIME(J),DELTI(J),J=1,NOBS)
      GO TO 1710
  910 FORMAT (3X,I3,F11.5,F10.5,F14.5)
```

```
911   FORMAT ('0-------- RMS ERR',E13.5,5X,'SUM RESID SQ',E13.5,5X,
    1   'DET',E13.5/'0    VAR    DELTA',6X,'ERR',9X,'NEW VAL')
 730  FORMAT (4(F12.5,F9.5))
 731  FORMAT (1X,4(4X,'TIME',5X,'DEL.MAG.'))
 732  FORMAT (1H0,4(4X,'TIME',6X,'DELTA I'))
      END
```

```
***********************************************************************
```

```
      FUNCTION TOTINT (N)
C GAUSSIAN INTEGRATION OVER ENTIRE AREA OF STAR N (1 OR 2)
C STAR IS ELLIPSE WITH AXES (AAXIS,BAXIS) ROTATED FROM
C     (YP,ZP) AXES BY ANGLE SPECIFIED BY SINP,COSP.
      COMMON/ROTAT/ SINP(2),COSP(2),DCHI,ECHI
      COMMON /GAUSS/ WT(16),X(16),L,XC(16)
      COMMON/TVARS/ALPHA(2),THETA,CHI,R,DELTA,PARA(19,2),AAXIS(2)
     1   ,BAXIS(2),PHASE,PHI(2),THETAP(2)
      COMMON/FLAGS/IFSPH,JTYPE,KSTAR,TEST(17),NOBS,NZONE,MREF,NREF
     1   ,IREF,NINC,LST
      MREF = 0
      NREF = N
      SUM2 = 0.
      DO 100  J=1,L
      SUM1 = 0.
      Y = AAXIS(NREF)*X(J)
      DO 10  I=1,L
      Z = BAXIS(NREF)*X(I)*XC(J)
      YP = Y*COSP(NREF) - Z*SINP(NREF)
      ZP = Y*SINP(NREF) + Z*COSP(NREF)
   10 SUM1 = SUM1 + WT(I)*BRIGHT(YP,ZP,NREF)
  100 SUM2 = SUM2 + WT(J)*XC(J)*SUM1
      TOTINT = AAXIS(NREF)*BAXIS(NREF)*SUM2
      RETURN
      END
```

```
***********************************************************************
```

```
      SUBROUTINE ZONES
C   ESTABLISHES TABLE OF COORD AND LOCAL INTENSITY FOR REFLECTION CALC
      COMMON/ORBE/INCL, ESINW, ECOSW, UA, UB, A, RATIOK, ELLIP, ELLIPB,
     1 EPSI, EPSIB, TCONJ, VA, VB, RATIOJ, WA, WB, PERIOD
```

```
      COMMON/TABLE/ENDINT(2)
      COMMON/AUXE/SCALE(2),SURF(2),SLIP(2),TILT(2),QINT(2),AA(2),BB(2)
     1   ,CC(2),RBAR(2),UMA(2),VMA(2),WMA(2),RNOT,BOLOJ,QUAD,TOTAL
     2   ,STAR1,STAR2,STAR3
      DIMENSION V(2),AREA(2)
      EQUIVALENCE (V(1),VA)
      QLITE(VV) = QINT(J)*(VMA(J) + V(J)*VV/RBAR(J))
  100 DO 110  J=1,2
  110 ENDINT(J) = QLITE(AA(J))
      RETURN
     END
```

****

APPENDIX IV

SAMPLE COMPUTER OUTPUT

WINK- SAMPLE RUN

```
INPUT DATA
 1   88.0000
 6  0.383700
 7  0.739500
10   10000.0
11   8660.00
13  0.500000
27   5.00000
19   2.00000
21  0.250000E-01
 0  0.0
```

ORBITAL PARAMETERS

P	TC	I	ESINW	ECOSW	NODE	E	SEMI-AXIS
1.000000	0.0	88.000	0.0	0.0	0.0	0.0	1.000

ASTROPHYSICAL PARAMETERS

	NU	A0	BETA	TEMP(EQ)	(POLAR)
STAR A	0.38370	0.3837	0.250	10000.00	10415.04
STAR B	0.28375	0.2837	0.250	8660.00	8953.68

MODEL PARAMETERS

A	K(A)	K(NU)	J(5500.)	J(BOLO)	MASS RAT	QUAD MAG	3RD LT	POLYTROP
0.39996	0.74996	0.73950	0.65006	0.56243	0.5000	0.0	0.0	5.

	ELLIP	ZETA	U	V	W	A-AXIS	B-AXIS	C-AXIS
STAR A	0.9594	-0.0018	0.600	-2.822	0.0	0.4000	0.3837	0.3574
STAR B	0.9352	0.0323	0.600	-3.176	0.0	0.3000	0.2305	0.2708

LUMINOSITIES

	APPARENT	NORMALIZED	TOT(4PI)	NORMALIZED
STAR A	0.36949	0.74010	1.80842	0.74491
STAR B	0.12976	0.25990	0.61929	0.25509
RATIOS	0.35117		0.34245	

	ATMOSPHERE		ORBITAL SKEW	
	SCALE	TAU	PHASE	TILT
STAR A	0.0	0.0	0.0	0.0
STAR B	0.0	0.0	0.0	0.0

INTEGRATION 6X6

ECL	STAR	TIME	PHASE	INT	MAG
ANNULAR	A	0.0	0.0	0.49632	0.7606
PARTIAL	A	0.02500	0.02500	0.55260	0.6440
PARTIAL	A	0.05000	0.05000	0.69613	0.3933
PARTIAL	A	0.07500	0.07500	0.82261	0.2120
PARTIAL	A	0.10000	0.10000	0.90990	0.1025
		0.12500	0.12500	0.94764	0.0584
		0.15000	0.15000	0.96390	0.0399
		0.17500	0.17500	0.97851	0.0236
		0.20000	0.20000	0.99006	0.0108
		0.22500	0.22500	0.99745	0.0028
		0.25000	0.25000	1.00000	0.0000
		0.27500	0.27500	0.99745	0.0028
		0.30000	0.30000	0.99006	0.0108
		0.32500	0.32500	0.97851	0.0236
		0.35000	0.35000	0.96390	0.0399
		0.37500	0.37500	0.94764	0.0584
PARTIAL	B	0.40000	0.40000	0.91794	0.0930
PARTIAL	B	0.42500	0.42500	0.85764	0.1667
PARTIAL	B	0.45000	0.45000	0.77519	0.2765
PARTIAL	B	0.47500	0.47500	0.69224	0.3994
TOTAL	B	0.50000	0.50000	0.67387	0.4286

WINK- INCLUDE REFLECTION EFFECT

```
INPUT DATA
 1   88.0000
 6 0.383700
 7 0.739500
10   10000.0
11   8660.00
13 0.500000
27   5.00000
16   1.00000
17   1.00000
19   2.00000
21 0.250000E-01
 0 0.0
```

ORBITAL PARAMETERS

P	TC	I	ESINW	ECOSW	NODE	E	SEMI-AXIS
1.000000	0.0	88.000	0.0	0.0	0.0	0.0	1.000

ASTROPHYSICAL PARAMETERS

	NU	A0	BETA	TEMP(EQ)	(POLAR)
STAR A	0.38370	0.3837	0.250	10000.00	10415.04
STAR B	0.28375	0.2837	0.250	8660.00	8953.68

MODEL PARAMETERS

A	K(A)	K(NU)	J(5500.)	J(BOLO)	MASS RAT	QUAD MAG	3RD LT	POLYTROP
0.39996	0.74996	0.73950	0.65006	0.56243	0.5000	0.0	0.0	5.

	ELLIP	ZETA	U	V	W	A-AXIS	B-AXIS	C-AXIS
STAR A	0.9594	-0.0018	0.600	-2.822	1.000	0.4000	0.3837	0.3674
STAR B	0.9352	0.0323	0.600	-3.176	1.000	0.3000	0.2805	0.2708

LUMINOSITIES

	APPARENT	NORMALIZED	TOT(4PI)	NORMALIZED
STAR A	0.37190	0.73381	1.80842	0.74491
STAR B	0.13491	0.26619	0.61929	0.25509
RATIOS	0.36275		0.34245	

	ATMOSPHERE		ORBITAL SKEW	
	SCALE	TAU	PHASE	TILT
STAR A	0.0	0.0	0.0	0.0
STAR B	0.0	0.0	0.0	0.0

INTEGRATION 6X6

ECL	STAR	TIME	PHASE	INT	MAG
ANNULAR	A	0.0	0.0	0.49214	0.7698
PARTIAL	A	0.02500	0.02500	0.55095	0.6472
PARTIAL	A	0.05000	0.05000	0.69879	0.3891
PARTIAL	A	0.07500	0.07500	0.82930	0.2032
PARTIAL	A	0.10000	0.10000	0.91813	0.0927
		0.12500	0.12500	0.95400	0.0511
		0.15000	0.15000	0.96736	0.0360
		0.17500	0.17500	0.97959	0.0224
		0.20000	0.20000	0.98963	0.0113
		0.22500	0.22500	0.99656	0.0037
		0.25000	0.25000	1.00000	0.0000
		0.27500	0.27500	0.99955	0.0005
		0.30000	0.30000	0.99536	0.0050
		0.32500	0.32500	0.98785	0.0133
		0.35000	0.35000	0.97804	0.0241
		0.37500	0.37500	0.96681	0.0366
PARTIAL	B	0.40000	0.40000	0.93745	0.0701
PARTIAL	B	0.42500	0.42500	0.86700	0.1550
PARTIAL	B	0.45000	0.45000	0.77167	0.2814
PARTIAL	B	0.47500	0.47500	0.68227	0.4151
TOTAL	B	0.50000	0.50000	0.66381	0.4449

☆ U.S. GOVERNMENT PRINTING OFFICE: 1972—735-965/410