KRC Version 3.5 with eclipses and planetary fluxes

Hugh H. Kieffer File=-/krc/Doc/v35/eclipse.tex 2017mar12:Apr07

April 7, 2017

Contents

Abstract

KRC has been expanded to handle two kinds of eclipses. Rather than wait for Version 4 with full longitude support, Version 3.5 has been generated with one-longitude-at-a-time support for two types of eclipses

1) Daily: as for Jovian satellites.

2) Rare, in that the lead-up days did not have eclipse, as for Earth-lunar or Phobos shadow on Mars.

The insolation profile through an eclipse has been modeled in consideable detail. However, PORB has not been changed, so the user will have to do some work to calculate the eclipse "bias" from perfect alignment. For Jovian (and similar) satellites, reflected and thermal radiation from the planet can be significant, especially during eclipse; a sinusoidal approximation for these in the form $F = c_1 + c_2 \cos(\psi - c_3)$ has been included for each. Added the capability to write binary files of surface temperatures at every computed time-step on the last day of the last season for any set of cases. Version 3.5 is backward compatible with earlier versions 3.x, so that non-eclipse use is unchanged.

1 Introduction

Terminology:
Occulting body: OB The body casting the shadow. For daily eclipses, this is typically a planet. For rare eclipses, this is commonly a satellite.
Eclipsed body: EB The body in shadow. For daily eclipses, this is typically a satellite. For rare eclipses, this is commonly a planet.
Eclipse body Surface Point: ESP The location on the EB for which calculations are done.
Bias The Sun:ESP line closest approach to the OB center, as a fraction of the OB radius.
Central hour The KRC hour at ESP

Eclipse insolation profile includes the full geometry for round body occulting a round Sun. Simplifying assumptions for eclipse insolation profile.

1. Assume circular, uniform irradiance source (Sun)
2. Effect of planet atmosphere treats Sun as a point source. ? [atmosphere not implimented]
   Convolve this with the geometric extinction.

NOTE: The atmosphere effects became too messy, and are currently omitted!

KRC 3.5 assumes synchronous rotating satellites, so longitudes are not all the same as in earlier versions of KRC. For simplicity, specify surface longitudes as Hours from the sub-solar point at inferior conjunction (from the Sun) and increasing eastward (right-hand about the North pole).

The finite size of the EB is included in computing distances; in the Solar System, this is important only for Phobos shadow on Mars.

The symbols <>> and <<> are used here to bound direct quotes from articles or prior documents.

2 Users Guide

This guide is a supplement to prior KRC User Guides; it repeats virtually nothing.

Version 3.5 is backward compatible with earlier versions 3.x, so that non-eclipse use is unchanged.

Suggest starting with an input file from the distribution, e.g.,*eurD.inp*, and modifying as you wish.

1. Generate the geometry matrix for the planet:satellite of interest, and cut-and-paste it into the input file. A matrix for Europa is in *PORBCM.mat* and in the suggested input file.
2. To invoke an eclipse, insert a change line 14 ; see §4.4. This eclipse will be in following cases until a change line '14 0 /' is used. If a Rare eclipse is specified, a binary file named *tfinexx.bin5* will appear in the running directory, where *xx* is the case number.
3. To invoke planetary fluxes, insert a change line 15; see §6.1. This will apply to following cases until a change line '15 0 /' is used.
   It may be helpful to look at the discussion in §B calculating the flux values.
4. To output a binary file containing the detailed surface temperature versus hour for one latitude, insert a change line 15; see §7. This will apply to following cases until a change line '16 0 /' is used.

If both eclipse and planetary heating are invoked in a case; the longitudes (expressed in Hours) should be the same. KRC does not check for this consistency.

2.1 New routines

There are two major new routines:

ECLIPSE eclipse.f Calculates the detailed insolation history of a circular body occulting a uniform round source.

TFINE tfine8.f Increases the depth (layer) and time resolution beyond that of TDAY to follow the details of a Rare eclipse.

EVMONO3D evmonod.f Evaluation of 3rd-degree polynomial with scaling. This is a modification of EVMON03 that has the scaling coefficients firm-coded, thus two less arguments, and is 9% faster.

Also utility routines **STRUMI** and **STRUMR8**, and two routines uses only in testing that can modify parameters: **GETPI4** and **GETPR8**.

3 Liens

1). Type 52 output for Rare eclipses in version 3.5.1 contains un-eclipsed values until a discontinutity at the end of the eclipse, with the proper details in a seperate binary file; these are merged in a post-run IDL routine. The TFINE algorithm could be moved into an optional (LRARE only) loop entirely within TDAY so that the Type 52 file had the eclipse results.

The layer TMIN and TMAX do not consider temperatures during a Rare eclipse; they do consider the remainder of the eclipse day. However, only the near-surface TMAX would likely be affected by rare eclipse.

Use of a special change line to toggle TOUT binary file is crude. Should be moved to an integer when KRCCOM ID is increased.

4 Eclipse design

Length input paramters are in physical units of km, but within the ECLIPSE routine, all distances are arbitrarily scaled to a characteristic length taken as the semi-major axis (radius, in this simplified case) of the mutual orbit of the occulting and eclipsed bodies.

4.1 Notation

"time-step" means as used in TDAY unless specifically called a fine time-step (or f-time) as used in TFINE.

Define a few angles and variables:
M: mutual orbital radius between the centers-of-mass of the OB and EB.
This is the normalization scale for all distances
R_O, r_O: radius (km) and normalized radius of the OB
R_E, r_E: radius (km) and normalized radius of the EB
ψ: orbital longitude; zero when EB is at inferior conjunction as seen from the Sun
ϕ: orbital angle from the center of the eclipse $\phi = \psi - \pi$
H_c: KRC hour at the center-of-eclipse for the satellite surface point of interest.
K_L: fine layer factor for rare eclipse
$K_T \equiv K_L^2$; fine-time factor for rare eclipse
β: orbital angle from center to edge of eclipse shadow
α: Angular radius of the Sun from OB:EB system, radians
r_S: Normalized radius of the Sun in the working plane.
b: Closest approach of the sun-line to the center of OB, as a fraction of OB radius.
N_2=N2: KRC number of time-steps per sol; = N2
J_7=J7: Last 1-based time step before the start of eclipse phenomona
J_8=J8: First 1-based time step after the end of eclipse phenomona

4.2 Basic eclipse phenomenon equation

Basic assumption is that a round source (the Sun) is being blocked by a round Occulting Body (OB). The formula from the intersection of two circles is taken from http://mathworld.wolfram.com/Circle-CircleIntersection.html

$$A = r^2 \underbrace{\arccos\left(\frac{d^2 + r^2 - R^2}{2dr}\right)}_{ANG2} + R^2 \underbrace{\arccos\left(\frac{d^2 + R^2 - r^2}{2dR}\right)}_{ANG1}$$

$$-\frac{1}{2}\underbrace{\sqrt{(-d+r-R)(-d-r+R)(-d+r+R)(d+r+R)}}_{SQP} \qquad \mathrm{eq:e14} \tag{1}$$

where r and R are the radii of the two circles and d is the separation of their centers.

Implement using B for the radius of the Bigger circle (which might be either r_O or r_S) and R for the other, with many intermediate variables and tests for speed and avoiding faults.

If $d \geq (B + R)$ then $A = 0$; if $d \leq (B - R)$ then $A = \pi R^2$.

The fraction of sun-light reaching the surface of EB is $F = 1 - A/(\pi r_S^2)$.

If B is the Sun, then have an annular eclipse and $F_{min} = 1. - (R/B)^2$. If R is the sun, then have total eclipse with $F = 0$ for some time.

4.2.1 Relations in eclipse

Define the "eclipse surface point" (ESP) as the point of interest on the surface of the EB at local hour H_C, and assummed to be on the EB equator (relax this assumption later?).

For solar eclipses by Phobos on Mars, need to consider the radius of Mars as it is a significant fraction of the radius of Phobos' orbit. So, include these geometric relations in the code, they will be trivial for most objects. However, do not include the small variation through an eclipse of the relative angular size of Phobos and the Sun as seen from the surface of Mars, just use the sizes at the center of an eclipse.

The angular radius of the Sun is $\alpha = R_S/(H_U U)$ where R_s is the radius of the Sun (km), H_U is the heliocentric range in Astronomical Units, and U is the Astronomical Unit in km.

Define a geometry "working" plane through the center of OB normal to the direction to the Sun. +Z is away from the Sun.The X direction is parallel to the EC orbital plane

The effective orbital radius of the ESP is $Q = M - R_E \cos(z)$, where z is the zenith angle of the Sun from ESP at the mid-time of eclipse. Assuming the eclipse is on the equator, $z = \frac{\pi}{2}(H_c - 12)/6 = \pi(H_c/12 - 1)$. In the normalized system, $Q = 1 - r_E \cos(z)$

At any time, the position of the ESP is $X = Q \sin\phi$ and $\phi = 2\pi t/P$ where t is time from mid-eclipse and P is the EB orbital period. Also, $\phi = 2\pi(J - J_C)/N2$ where J is the c-time count and J_C is the c-time of mid-eclipse.

The apparent size of the Sun in this plane, as seen from the surface of the EB, is $r_S = \alpha Q/M$ There will some eclipse effect if the bias $b < (1 + r_S/r_O)$

At first contact, $X_c = \sqrt{(R+B)^2 - (br_O)^2}$, this sets the half-duration of the eclipse t_h in the same units as P.

At any time, the center separation is

$$d = \sqrt{X^2 + \underbrace{(br_O)^2}_{Y2}}$$

In the KRC diurnal system, if surface point of interest is at hour H_c when the middle of eclipse occurs, and there are N2 time steps in a sol, with the last at midnight, then fractional (1-based) indices at the beginning and end of eclipse are $\frac{H_c}{24}N_2 \pm \frac{t_h}{\mathrm{sol}}N_2$.

V 3.5 does not handle a sol different from P; this case may never be of interest.

Eclipse is symmetric about orbital angle of π but the eclipse function must be centered about ESP at the satellite surface hour requested.

To allow any resolution in the satellite surface position, KRC uses fractional orbit angles that are on the time grid, but shifts application in TLATS or TDAY by integral time steps.

For an eclipse with zero bias, the transition to totality, as a fraction of a sol, is α/π. In these units, the full eclipse lasts $(r_o + \alpha)/\pi$. For Europa, the values are roughly 1/3500 and 1/30. Thus, to begin to resolve the penumbral phase, would require N2> 7000.

4.3 Implementation

In general, for any eclipse, set N24 as large as allowed (MAXNH).

To deal with rapid insolation changes, shorten the time-steps by some factor. Do not need to change the layering for stability, but should change it for responsiveness. To keep same stability factor, divide each layer by integral factor f and increase the number of times/day by f^2.

To resolve the penumbra stage, there should be several time steps within it; if this is impractical, there should be many (more than a dozen?) time steps within the entire eclipse.

Ideally f^2 would be roughly length-of-a-sol / 2t, where t is the time for the satellite orbital phase to change by the angular diameter of the Sun θ_S . E.g., for Europa, t is ???

Daily: Handle entirely in Tlats, with consideration for the large N2 required to see shape of insolation through an eclipse.

.

Planet thermal load into new array: PLANH, compute in TLATS
Planet reflected solar load into new array: PLANV, compute in TLATS.
    For daily eclipses, these are incorporated in TDAY, see Eq. 9
    For rare eclipses, TFINE combines the two and does linear interpolation to fine time.

Satellites are assumed synchronus. Yet, both PERIOD and PARC(4) must be specified and should be the same.

4.3.1 Vector geometry

The coordinate system used by TLATS is the "Day" system
    +Z toward body right-hand spin axis (north pole),
    +X in the true solar midnight meridian,
    +Y is Z cross X, and is in the equatorial plane
Sun at declination δ at midnight: $M = [\cos\delta, 0., \sin\delta]$
    Sun diurnal progress is left-hand $\phi = -\frac{2\pi}{24}t$ where t is in hours, rotating around +Z
Local surface normal at latitude α, in the noon meridian: $F = [-\cos\alpha, 0, \sin\alpha]$
To get the normal to a surface with slope (dip) β facing toward azimuth ψ measured east from North:
    rotate F around +Y by β , generates temporary vector Q
    then rotate Q around the original F by $-\phi$ to get tilted surface normal T

For an ESP at hour H_C, $\omega = \pi H_c/12$ a planetary heat source above the equator at noon would have a unit vector $P = [\cos\omega, \sin\omega, 0]$

4.3.2 Time indices for eclipses

Time indices passed between routines are always in TDAY units, in some places called coarse time or ctime. Where they are converted to/from fine-time (or ftime), they are treated as refering to the start of a time interval, before the diffusion calculation.

ECLIPSE and TFINE are the only FORTRAN routines that deal with fine-time.

ECLIPSE calculates the time of first and last contact in floating-point time; first contact is rounded down to JBE(1) and last contact is rounded up to JBE(2); JBE is passed between routines; the rounding ensures that the indices in JBE capture the full optical eclipse. ECLIPSE returns an array for the insolation factor FINSOL; the fraction of insolation that makes it past the occulting body to the Eclipse Surface point (ESP). FINSOL is 1 outside the eclipse.

4.4 Eclipse Specification,

Eclipse specification:
1: Style: 0=none 1=Daily 1.3+=rare, round of value is layer factor
time factor is square of layer factor to retain stability
2: Distance to sun, AU (used to get Sun angular diameter)
3: Occulting body (OB) radius, km
4: Mutual center-of-mass orbit radius, km = M
5: Eclipsed body (EB) surface radius, km
6: [Mutual orbit period, days] Assumed same as diurnal PERIOD
7: Eclipse Bias
8: [J2000 date of Rare eclipse] Assumed to be on the last "season"
KRC 3.5 uses the sign as a flag for base treatment. + is maintain heat-flow - is maintain temperature.
9: Eclipse central hour
10: Debug code. ne.0 prints constants and ¿1 prints one point,
Negative runs a " null eclipse" test mode in which the OB is considered transparent.
x: Extinction scale height of OB's atmosphere, km. NOT implimented

These will be input as a change line 14: first real value being non-positive means turn off. Typical input line:
14 1 5.2026 71492. 0.6711D6 3121.6 3.551 0.01 6000. 12. 2 / Europa

4.5 Eclipses: Daily

For "Daily" (typically long) eclipses, fine-time is never used; the user can set N2 as large as they want to get the eclipse details. FINSOL covers the entire day in ctime steps; it is unity outside ot the JBE range.

Binary output files are the same as earlier versions of KRC.

4.5.1 Details

TLATS: Handles only daily eclipse: Sets LECL flag if PARC(1) $0.8 < x < 1.2$. If set, then
Calls ECLIPSE once per season, which generates FINSOL insolation factor for each time-step
and duplicates as SOLAU the variable for solar flux at current AU
Each time step, multiplies the solar insolation by FINSOL(JJ)

TDAY: No change for daily eclipses.

4.6 Eclipses: Rare

For "Rare" (typically short), the uses specifies a fine-layer factor K_L (rounding the eclipse "style" parameter); the fine-time factor K_T is the square of the layer-factor. FINSOL covers in fine-time steps from the beginning of JBE(1) to the end of JBE(2).

In TLATS and TDAY, the Hour of a time step J(1-based) is in the middle: H= (J-.5)24./N2.
The hour of FINSOL element I is in middle: (J7-1)*24./N2 + (I-.5) 24./(K_T*N2)
where J7 is the index of the coarse time-step when fine-time starts

Because of the rapid changes that can follow the return of sunlight at the end of eclipse, the detailed calculations of TFINE are continued for the number of ctime steps of the optical eclipse (at least one) KRC output interval after last contact.

TFINE always outputs additional binary file *tfinexx.bin5* with ASOL, FINSOL and all layer temperatures at every fine time-step within eclipse for each Rare eclipse case. Header contains N2,J7,J8,J9. Array is [2 + fine layer, fine-time]

4.6.1 Details

To ensure catching all the eclipse effects, expand the fine-time range by one earlier TDAY timestep and later by the duration of the eclipse dJ= JBE(2)-JBE(1). Thus the "trigger" values of the time-steps JJ in TDAY are: J7=JBE(1)-1 and J9=JBE(2)+[dJ> 1] , and TDAY switches to and from TFINE before the diffusion loop. TFINE will run over ctime JJ: J7 through J9-1, with linear interpolation of upper boundary conditions to fine time-steps.

In TFINE, as in TDAY, the surface temperature is stored in layer index 1, but that layer is reconstructed as the virtual layer in each time-step.

Number of fine layers: virtual layer + (number of physical TDAY layers * layerFactor)
    N1F=1+(N1-1)*KFL and must store one more for the base

. LRARE is normally False. and TDAY(1 normally sets J7=-1
- If PARC(1) is $\geq$1.3 then TDAY(1 sets flag LRARE True and will do a RARE eclipse. It calls ECLIPSE to get JBE which contains the normal time steps before and after the eclipse. It calls TFINE(1 to do all that can be done without having layer temperatures.
- The eclipse is entirely within ctime JBE(1) to JBE(2), which could be the same for a very short eclipse.
- In TDAY(2 day loop, if LRARE, then on the last season, at the start of the last day, TDAY(2 sets J7 to JBE(1)-1 .
- In TDAY(2 time loop, when JJ equals J7, if J7 $\leq$ JBE(1), TDAY transfers the layer temperatures to TFINE.
- After TFINE returns, TDAY sets J7 equal to the end of eclipse followon J9+1 and proceeds normally until the (JJ equal J7) test is again satisfied, when (before diffusion) it sets the temperature profile to the final from TFINE, and finishes the last day normally, leaving a discontinuity in temperature at the end of the eclipse.

TFINE is a modified copy of TDAY, with a single "day". It increases the layer and time resolution, interpolating in time and depth as needed, then steps through the eclipse. It calls ECLIPSE to get the detailed insolation profile.

Because solar eclipse must occur before dusk, except for the pathologic case of an eclipse near the summer pole, there would be many output "hours" after the start of eclipse into which the detailed time results could be loaded.
Might not be enough, and is hokey. So, output a separate file

TFINE has a large storage buffer, and stuffs the eclipse factor and temperature profile (up to 99 fine layers) into this for writing to a .bin5 file for each rare eclipse case. This file covers the hours H(I,J)= (I/KFT +J-.5)* 24./N2 from I,J=1,J7 to I,J= KFT,J9 .

The output file is always named "tfinexx.bin5" where xx is the case number. Thus, user should **rename tfinexx.bin5 before any other eclipse run**.

4.6.2 Extra printout

Rare eclipse cases put additional material in the print file, apart from debug options. Below, left-adjusted lines are example printout and inset lines are explanations.

.
TFINE IQ,J4= 1 2
    IQ is the TDAY action requested: 1=setup, 2=do the time and layer loops
    J4 is latitude index; not reliable or relevant for IQ=1
TFINE layers: Num,lowest center[m] 82 0.8995
    TFINE(1: the number of fine layers, including virtual. Depth to the center of the lowest (not base) layer
Min safety: layer,factor= 1 0.000 0.000
    TFINE(1: Minimum convergence safety factor: layer, factor,
TFINE low lay of time doubli: 12 20 27 34 42 49 57 64 72 82
    Deepest layer for each fine-time doubling
-777 3 10 2 4 760 776 792 82
    TFINE(1: tag, NCASE, J5, J4(+1), J3, JBE(1:2), J9, N1F
TFINE exit   notification of exiting TFINE
TFINE IQ,J4= 2 1

TTJ(1)... 260.19471582799264 258.02291589626202 -0.0000000000000000
    Virtual layer T for TDAY and fine layer system. Delta T at fine base
LZONE,LALCON,J5, IK1:4= F T 10 0 0 0 0

In TDAY(2: last 4 are the T-dep. layer specifications.
TFINE: Case= 3 JJJ= 3 83 306 1 0 0 0 5 80 0
JJJ is the set of 10 integers given to BIN5F
TFINE wrote tfine03.bin5 iret= 0
TFINE exit
End eclipse: JJ,KG,delT,delE 793 28 0.63383E-01 8067.1
KG is the lowest TDAY layer represented in the fine layer system
delT is the T discontinuity of the lowest TDAY layer at end of eclipse
delE is the delta thermal energy in the lowest TDAY layer at that time.

5 TFINE

TFINE interacts primarily through the many KRC common's. Subroutine arguments are used to transfer in:
the stage index: with value 1 or 2
the physical properties of each layer.
and transfer out:
the coarse layer temperatures after eclipse
the number of reliable output temperatures, or a negative values indicating an error.

The initial call to ECLIPSE returns the last original time step before the eclipse starts and the first after it ends. When TDAY on the last day of convergence reaches the starting time setp, it calls TFINE which proceeds through all of its time steps and returns the temperature depth profile that it gets, TTF, which is remembered by TDAY. TDAY proceeds with its normal calculations until it reaches the time step after those covered by TFINE, when the temperature profile is reset to TTF, and TDAY runs through the rest of the last day.

TFINE ignores any atmosphere, except for any effect TLATS may have included on collimated insolation. It does handle far-field radiation.

There can be a small non-physical effect if the number of finer layers 1+(N2-1)*PARC(1) would exceed the firm-code size MAXFL. The bottom of the fine system is considered insolating during the eclipse, whereas the normal interface at that depth would be conducting. The non-physical change in system energy is roughly $B_j \rho_j C_j \Delta T_j$ where the last term is the amount that the temperature of the deepest original layer treated by the fine system j is changed at the end of the eclipse.

Note that continuation to another season using the asymptotic predictor would be inappropriate after a "rare" eclipse.

The temperature gradient that existed in the TDAY system at the depth of the bottom of the TFINE layers at the start of the eclipse is held constant at the bottom of TFINE through the eclipse.

$$\nabla T \;=\; \frac{T_{j+1}-T_j}{(B_{j+1}-B_j)/2} = \frac{T_{i+1}-T_i}{(B_{i+1}-B_i)/2} \qquad (2)$$

where the i subscripts are for the TFINE values at it lowest two layers and the j subscripts are for the TDAY layer values at the corresponding depth. Thus

$$T_{i+1} = T_i + \underbrace{(T_{j+1}-T_j)\frac{B_{i+1}-B_i}{B_{j+1}-B_j}}_{delbot} \qquad \text{eq : tbot} \qquad (3)$$

When TFINE starts an eclipse period , it uses cubic spline interpolation with natural boundary conditions (zero 2nd derivative at top and bottom). To avoid interpolation failure, all fine layer centers must be within range of the TDAY layer centers; thus maximum I is $K(N_1-1)+1+K/2$.

5.1 Details

Design with two sections, similar to TDAY. TFINE(1 does everthing that does not require the starting temperature profile of the time-dependent radiation field. TFINE(2 does the timesteps.

Has access to commons
Creates finer layers and has an inner timeloop for the finer time steps.

Needs the original center depths of each layer, and must generate the center depths of the new layers for interpolation and the bottom depths for the diffusion equations.

Anything that is defined in TDAY (vrs defined in commons) is not available in TFINE unless an argument.

Zone table logic is complex, could duplicate them in TFINE but better to pass in as arguments the ultimate products: KTT, DENN, CTT as arguments; TTJ, XCEN and BLAY are in common

Each TDAY layer of thickness B_j is divided into K layers with thickness $B_i \equiv B_{j_k} = f_i B_j$

K fine layers must have a geometric ratio r that yields the same full-layer ratio as $R \equiv$ RLAY.

$$r^K = R \qquad \text{or} \qquad r = \exp \frac{\ln R}{K} \qquad \text{eq : rk} \tag{4}$$

and the sum of f_k must be 1.

The sum of a geometric series of ratio r, first term 1 and n terms is

$$S \equiv \sum_{j=0}^{n-1} r^j = \frac{1-r^n}{1-r} \qquad \text{eq : sumr} \tag{5}$$

$$f_1 = 1/S = \frac{r-1}{r^K-1} \Rightarrow \frac{r-1}{R-1} \qquad \text{eq : f1} \tag{6}$$

and $f_i = r f_{i-1}$

Each time-dependent input is linearly interpolated to the fine-time steps:
- ALBJ Surface albedo, which may vary with incidence angle
- ASOL Collimated flux onto (sloped) surface
- FARAD Far-field radiance
- SOLDIF Diffuse solar flux
- ϵ PLANH + $(1 - A_s)$PLANV absorbed Planetary flux

TFINE always writes to print file
- Number of fine layers and depth[m] to center of deepest layer
- If any T-dependent layers, the first and number of the A and B layers
- Low layers for fine-time doubling
- -777, Indices for: case,season,latitude,converg.day, eclipse hour range ...

Layers temperatures j returned by TFINE, after TSUR as the layer (1) the rest of the layers are from the fine layers i. This could be based on:
- if K odd, $i = jK - 3(K-1)/2$
- if K even, average of layers $i = jK - 3K/2$ and $i+1$

However, simpler (and better for even K) to use spline interpolation.

6 Planetary fluxes

For synchronous satellites, the temperatures depend strongly on longitude but KRC version 3.5 does not treat longitude explicity. For the side facing the planet (the "near-side"), the peak reflected radiation comes at midnight and eclipses come near noon. For the side away from the planet, there is no reflected or thermal planetary heat load and no eclipses.

Since there are two bodies in addition to the Sun, must treat the effect of solar reflection and thermal emission from the "planet". Version 3 will assume these are sinusoidal with orbital phase. E.g., in the form $F = c_1 = c_2 \cos(\psi - c_3)$ and in units of W m^{-2}. Phase = ψ is zero when the satellite is at inferior conjunction as seen from the Sun (is this the convention?).

6.1 Planetary Flux specification

Must specify 7 values:
- 1: Average Solar flux from OB (planet) at the EB (satellite)
- 2: half-amplitude of variation with phase
- 3: Phase lag, in degrees
- 4: Average thermal emission from OB at the EB
- 5: half-amplitude of variation with phase
- 6: Phase lag, in degrees
- 7: The longiitude (in Hours) of the surface point.

These will be input as a change line 15: the first real value being non-positive means turn off.

Example:

```
15  0.62 0. 0.     0.21 0.21  0.    12. / Jupiter heat load on Europa at Sub-J
```

6.2 Planetary load away from zenith

Assume the absorption surface is Lambertian. For a point source, the effect varies with zenith angle as $\cos\phi$. For a modest source of radius R radians, the effect is

$$W(\phi_1) = \frac{\int_y^\pi \cos\phi \cdot 2R\sin\theta \ R\sin\theta \ d\theta}{\pi R^2} = \frac{2}{\pi}\int_y^\pi \cos(\phi_1 + R\cos\theta)\cdot \sin^2\theta \ \ d\theta \tag{7}$$

where y is the horizon limit of $\arccos\left((\frac{\pi}{2} - \phi_1)/R\right)$ if $\phi > (\frac{\pi}{2} - R)$ and 0 otherwise.

As $R \to 0$,

$$W \to \frac{2}{\pi}\cos\phi_1 \int_0^\pi \sin^2\theta \ d\theta = \frac{2}{\pi}\cos\phi_1 \coprod_0^\pi \frac{x}{2} - \frac{\sin 2x}{4}\Bigg] = \frac{2}{\pi}\cos\phi_1 \cdot \frac{\pi}{2} = \cos\phi_1$$

as expected. Here $\coprod_l^u \cdots]$ stands for evaluation at the upper limit minus evaluation at the lower limit

Expanding $\cos(\phi_1 + R\cos\theta) \Rightarrow \cos\phi_1\cos(R\cos\theta) - \sin\phi_1\sin(R\cos\theta)$ The form of the integral becomes

$$c\int \cos(a\cos x)\sin^2 x \ dx \ + \ s\int \sin(a\cos x)\sin^2 x \ dx$$

for which I could not find analytic solution.

Numerical solution coded in IDL planheat.pro; see Fig. 1. For Europa, finite size makes at most 2% difference, and greater than 1% only when within 1.5° of the horizon. Until the lower edge of the planet nears the horizon of the satellite, the normalized factor is virtually constant; the effect is about 0.13% for Europa and barely 1% for a 0.3 radian source

Because the effect of a finite angular size is small, I elected to omit it for version 3.5; the influence follows the cosine of the incidence angle for the OB center onto the [tilted] surface, μ_P.

6.3 Static geometry for synchronous rotation

Only synchronous satellites are considered.

In TLATS, a planet source is assumed to be in the equatorial plane and above "noon" so that it has a fixed relation to the target (tilted) surface with cosine of angle onto tilted surface μ_P=COSP. / IR and visual fluxes are initially set to zero for each latitude. If the logical flag LPH is True, then Planetary fluxes computed for orbital phase at each time-step and multiplied by $\mu_P \geq 0$.

In TDAY, if LPH is True, then both the plantary fluxes are multiplied by their absorption coefficents and added to the surface enery budget.

6.4 How it works

TLATS: Sets the flag LPH True if PARW(1) positive. At each time step, if LPH True,
sets PLANH(JJ)= $w_1 + w_2\cos(\theta - w_3/\mathrm{RADC})$ where RADC is degrees/radian.
add to the diffuse light SOLDIF(JJ) $w_4 + w_5\cos(\theta - w_5/\mathrm{RADC})$

TDAY: Sets the flag LPH in the same way as TLATS. At each time step, if LPH True
adds to the absorbed hemispheric downwelling IR flux ABRAD the amount FAC6*PLANH(JJ) where FAC6 is fraction of the sky visible times surface emissivity.

7 Detailed temperature output

KRC has traditionally output temperatures and other values every Hour (1/24th of the bodies day) or sub-multiple thereof, down to 1/4 Hour, the firm-code limit for N24 being 96. To track surface temperatures through an eclipse requires

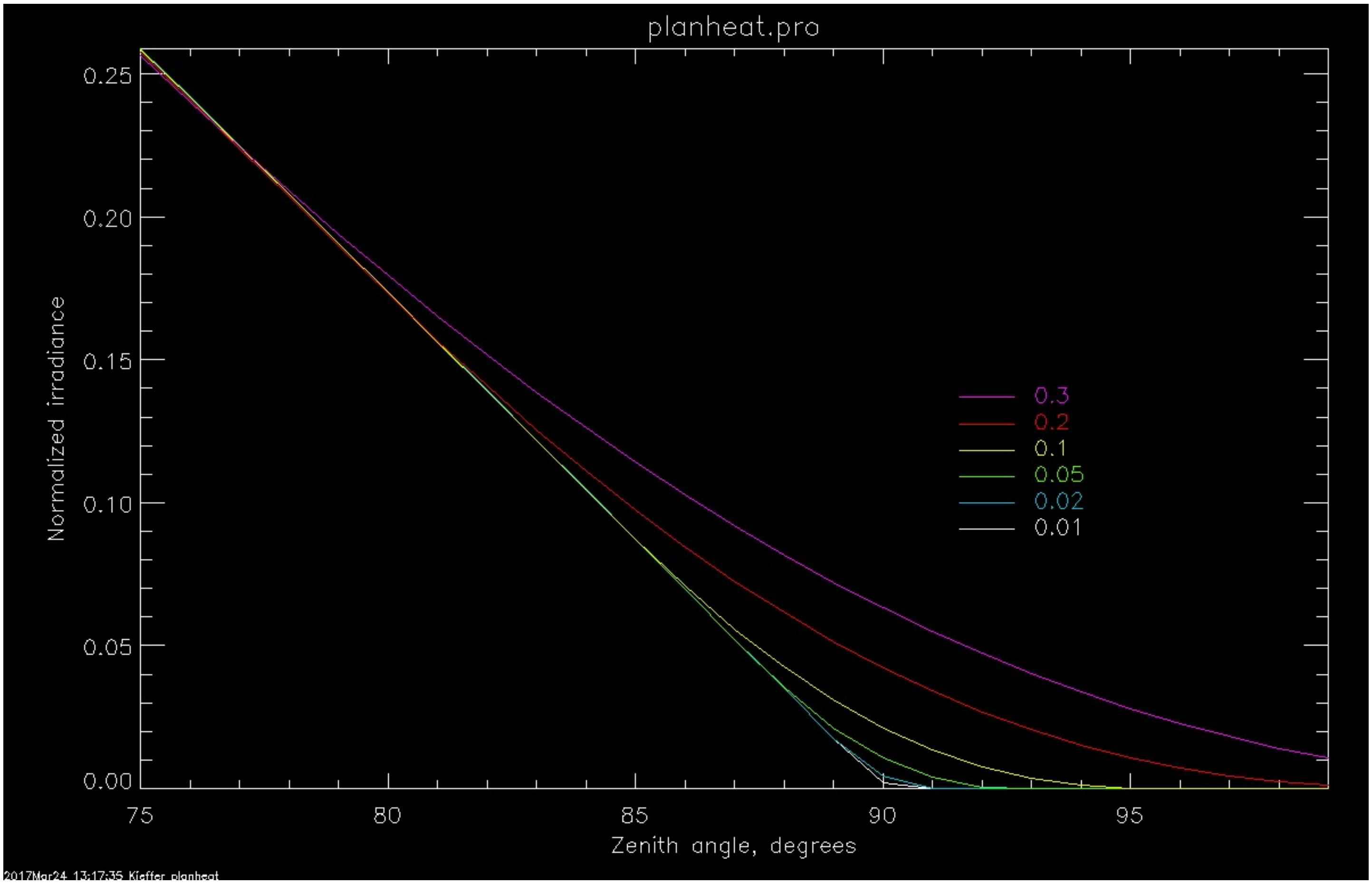

Figure 1: Normalized heat-load for finite-sized round sources as a function of zenith angle. Lambertian surface assumed. Legend shows the radius of the source in radians; for Europa the value is 0.1067. Infinitesimal source follows a cosine relation. Larger source are slightly less than cosine except near the horizon. planheatb.png

higher resolution. While a Rare eclipse will output a file of temperatures at high resolution (every fine-time step) for times around the using the special routine TFINE, the Daily eclipse calculations are done with a small modification of TDAY with no special output.

To address this in part a generic way, the existing array `TOUT` that is already in a COMMON can now be written to a binary file on the last day of the last season for one latitude. This is invoked by a change line 16, containing the 1-based index of the latitude desired and the leading part of the file name. Added to that part will be *cxx.bin5* where *xx* is the case number, generated automatically by KRC. The recommended leading name is

  ¡input file stem¿
  + 'lat'
  + ¡latitude in degrees with a following N or S as appropriate¿

Example:

```
16 1 'eurDlat0'  / output file for Tsurf every time-step
```

This will generate an output file in the running directory for each case untill stopped by a change-line: "16 0 / "

8 Summary of code changes

.
KRC:

In the case Loop, after TCARD and TPRINT, if any eclipse or planetary heat is active, will print: 'Eclipse or PlanHeat on',PARC(1),PARW(1)

Calls TDAY(1

TSEAS: none

TLATS:
  [un]set LPH and LECL=Daily flags
  Before the latitude loop, if Daily and first season, call ECLIPSE to get FINSOL
  In the time loop:
    If LECL, multiply Sun by insolation factor; can affect TEQUIL
    If LPH, calc PLANH(JJ) and PLANV(JJ).
  After time Loop
    If LPH, incorporate the average absorbed planetary heating into TEQUIL
  after TDAY(2 call: If at NLAD latitude and last season. Write TOUT to binary file

TDAY(1:
  [un]Set the LPH flag, [un]Set the LRARE flag
  If LRARE,
    call ECLIPSE to get the time-step range of eclipse, JBE
    Set full eclipse range to start 1 time step earlier and end after 2nd duration
    call TFINE(1 to do what can be done without temperatures

TDAY(2:
  If the last day and LRARE and the last season, set J7 to start of eclipse
  In the time loop, when reach J7, then
    if at start of eclipse call TFINE(2, before layer loop, then set J7 to end of eclipse followon J9 else, transfer TFINE results into layer temperatures, set J7 negative
  After the layer loops: if LPH, add in the planetary heating at each time step
  After last day, exit even if daily convergence tests fail (as they should)

TFINE(1: [called only for LRARE and only at start of eclipse on last day of last season]
  call ECLIPSE to get both time range JBE and FINSOL
  set J7=JBE(1)-1 and J8=JBE(2)
  set J9= J8+ (J8-J1 >1)

TFINE(2: Diffusion calculations cover ctime J7:J9
    Thus has J9-J7 +1 ctime-steps and K^2 times this ftime-steps.
  Interpolates current T/depth profile.
  Uses FINSOL and steps forward in time until end of eclipse J8
  Throughout the follow-on (J8 to J9) treat FINSOL as unity.

Layer relations: 1-based
  fine, first in set = I = (J-1)*KFL +1 where J is TDAY layer. KFL is layer factor
Time relations: 1-based. KFT=KFL^2
  Fine, first in set = I = (JJ-t0)KFT where JJ is TDAY time-step. H(JJ)=JJ*24./N2
    and 1=(JBE(1)-1-t0)KFT, thus: I=(JJ-(JBE(1)-1))KFT +1
  Since H=24*JJ/N2, H(I)= ((I-1)/KFT)+JBE(1)-1)* 24./N2

8.1 Other

TCARD: reads and prints a 14 or 15 line, loads the values into PARC or PARW in HATCOM

Because the first real value is used as a test for activation, either effect can be turned off by a single negative value. e.g.
  14 0. / turn off eclipses

Hour-dependant values computed in TLATS. Constant factors applied in TDAY.
In some cases, rather than logic tests for eclipse or Planetary loads, it is easier to always add them, but ensure they are zero when not invoked.

FINSOL in common used differently for daily and rare eclipses, which cannot be invoked at once. ECLIPSE calculates values only through the eclipse, so FINSOL must be replaced with 1.0 during the follow-on.

In TDAY, the insolation is evaluated at the instant of the middle of each time interval and the upper boundary condition evaluated after the diffusion ΔT is applied. Thus the assessment in ECLIPSE should also be at the middle of a TDAY interval. Strictly, the interpolation in TFINE should use the same instant, which can be done with no extra logic because eclipses cannot occur near the end s of the day (except at the poles)

As with TDAY, the upper boundary condition is applied after the layer loop for each timestep. In TDAY(2, TFINE(2 is called ???

A change 15 lien,

9 Formulation

Starting with Equation wb=27 and some associated text of V34UG: <>>

$$\underbrace{W}_{POWER} = \underbrace{(1. - \overbrace{A_{h(i_2)}}^{ALBJ})}_{FAC3}\overbrace{S_M F_\parallel \cos i_2}^{ASOL} + \underbrace{(1 - \overbrace{A_s}^{SALB})}_{FAC3S} \underbrace{S_M \left(\overbrace{\Omega F_\ominus^\downarrow}^{DIFFUSE} + \overbrace{\alpha A_s (G_1 \cos i F_\parallel + \Omega F_\ominus^\downarrow)}^{BOUNCE} \right)}_{SOLDIF}$$

$$+ \underbrace{\Omega\epsilon}_{FAC6} \underbrace{R^0_\Downarrow}_{ATMRAD} + \underbrace{k \frac{\partial T}{\partial z}_{(z=0)}}_{SHEATF} - \underbrace{\epsilon\sigma}_{FAC5} T^4 + \overbrace{\underbrace{(1-\Omega)\epsilon\sigma\epsilon_x}_{FAC5X} T_x^4}^{FARAD} \qquad \mathrm{eq:wb} \tag{8}$$

where the overbrace items are computed in TLATS and transfered in COMMON. All terms up to and including ATMRAD make up the total absorbed radiation ABRAD. When frost is present, its albedo replaces A_h and A_s on a time-step basis except the A_s in SOLDIF (from TLATS) is on a season basis; however, the A_S term includes the far-ground fraction α which is small except for steep slopes.

Assumes that normal albedo is the same for the sloped and the flat surfaces.

The fraction of solar flux reflected ALBJ≡ A_h =ALB*AHF is composed of two factors, ALB≡ A_0 and AHF= $A_h(i)/A_h(0)$, a hemispherical reflectance function. Likewise, the spherical albedo is A_s = ALB*PUS where the second factor is P_s.

The floor of a "pit" does not see the flat terrain, but rather the same slope at all azimuths, and therefor different temperatures. The most practical assumption is that the average radiation temperature of the pit walls is the same as flat terrain. This will be an under-approximation. In a later version of KRC with more input parameters, a radiation scale factor could be included; if practical, code to include a constant factor, initally unity for v 3.4.

Because `FARAD` is not dependent upon the calculation of T, it can pre-computed for a given day. T_x is interpolated to the proper season in **TSEAS**; **TLATS** selects the proper latitude, multiplies by `FAC5X` for each of its stored hours, and interpolates to each time-step to form $\mathtt{FARAD}_t$ transfered to `TDAY`. However, to then accomodate variable frost emission, need to multiply by ϵ_f/ϵ for the frost case (relatively rare). <<>

Version 3.5, add eclipse attenuation of solar insolation FINSOL= F_X and add visual and IR planetary fluxes, PLANV= P_V and PLANH= P_H. For daily eclipses, TLATS includes F_X into S_M, so that TDAY needs be no different. TLATS does nothing for rare eclipses and F_X in handled entirely within TFINE.

$$\underbrace{W}_{POWER} = \underbrace{(1. - \overbrace{A_{h(i_2)}}^{ALBJ})}_{FAC3}\overbrace{S_M F_\parallel \cos i_2}^{ASOL} + \underbrace{(1 - \overbrace{A_s}^{SALB})}_{FAC3S} \left[\underbrace{S_M \left(\overbrace{\Omega F_\ominus^\downarrow}^{DIFFUSE} + \overbrace{\alpha A_s (G_1 \cos i F_\parallel + \Omega F_\ominus^\downarrow)}^{BOUNCE} \right)}_{SOLDIF} + \mu_P P_V e^{-\tau_v/\mu_P} \right]$$

$$+\epsilon\mu_P P_H e^{-\tau_R/\mu_P} + \underbrace{\Omega\epsilon}_{FAC6} \underbrace{R^0_\Downarrow}_{ATMRAD} + \underbrace{k \frac{\partial T}{\partial z}_{(z=0)}}_{SHEATF} - \underbrace{\epsilon\sigma}_{FAC5} T^4 + \overbrace{\underbrace{(1-\Omega)\epsilon\sigma\epsilon_x}_{FAC5X} T_x^4}^{FARAD} \qquad \mathrm{eq:wbe} \tag{9}$$

However, in version 3.5, the atmosphere effects on planetary fluxes are ignored, and the μ_P term is handled in TLATS.

ABRAD accumulates terms until SHEATF.

9.0.1 Synopsis of TLATS radiation calculations

```
TLATS
      LATM=PTOTAL.GT.1.0        ! atmosphere present flag
      LPH = PARW(1).GT.0.      ! doing planetary heat loads
      LECL= (ABS(PARC(1)-1.).LT. 0.2)       ! doing daily eclipses
      IF (LATM) allow twilight, else TWILFAC = 1. and LTW is False
      IF (LOPN3) setup TFAR8 and set LINT iff will need to interpolate in time
      SOLAU=SOLR=SOLCON/(DAU*DAU)! solar flux at this heliocentric range
      SALB=PUS*ALB              ! spherical albedo, for diffuse irradiance
        CALL ECLIPSE(PARC,PARI JBE, FINSOL) only if DailyEclipse and first season
in Lat. loop
 in time loop
  calc PUH= PhotFunc for horizontal surface using COSI
  calc AVEA=ALB*PUH and ensure 1-A cannot be negative
  If LATM do delta-Eddington, else
             TOPUP=COSI*AVEA         ! upward solar
             BOTDOWN=0.         ! no atm scattering
             ATMHEAT=0.         ! no atm absorbtion
             COLL=1.D0          ! no atm attenuation of beam
             DIRFLAT=COSI ! incident intensity on horizontal unit area
  if day or twilight
           DIFFUSE=SKYFAC*BOTDOWN ! diffuse flux onto surface
             G1=1.0D0
           BOUNCE=(1.D0-SKYFAC)*SALB*(G1*DIRFLAT+DIFFUSE)
  else
           DIFFUSE=0.
           BOUNCE=0.

  if  target is directly illuminated
   calc PUH=PhotFunc for (sloped) surface using COS2, HALB=ALB*PUH
           DIRECT=COS2*COLL
         IF (LECL) SOLR=SOLAU*FINSOL(JJ) ! eclipse factor     Daily only
         QI=DIRECT*SOLR         ! collimated solar onto slope surface

         ASOL(JJ)=QI            ! collimated insolation onto slope surface
         ALBJ(JJ)=MAX(MIN(HALB,1.D0),0.D0) ! current hemispheric albedo
         SOLDIF(JJ)=(DIFFUSE+BOUNCE)*SOLR ! all diffuse, = all but the direct.

         IF (LPH) THEN ! calc planetary heat loads and add to day sum
           PLANH(JJ) and PLANV(JJ)

         ADGR(JJ)=HUV=ATMHEAT*SOLR ! solar flux available for heating of atm. H_v
 end of time loop
        IF (LPH)  add in absorbed planetary heating
        IF (LATM)  set BETA and TEQUIL and other equilbrium temperatures
          else BETA=0. and set TEQUIL
         If first season TATMJ=77.7.  If no atm, no routine changes this
         CALL TDAY8 (2,IRL)      ! execute day loop
         Predict and store results
End of latitude loop
```

9.0.2 Synopsis of TDAY radiation calculations

```
TDAY(2
      FAC9=SIGSB*BETA           ! factor for downwelling hemispheric flux
      if no atm, ATMRAD=0.
Top of day loop
        IF (LDAY) THEN if LRARE and last season then J7=JBE(1)-1
```

```
IN time loop:
       IF (JJ.EQ.J7) and  J7 .LE. JBE(1)  CALL TFINE8  the reset J7
                     else Transfer layer T's and set J7=1
after layer loops: when no frost
            ABRAD=FAC3*ASOL(JJ)+FAC3S*SOLDIF(JJ) ! surface absorbed radiation
            IF (LATM) THEN
              ATMRAD=FAC9*TATMJ**4 ! hemispheric downwelling IR flux
              ABRAD=ABRAD+FAC6*ATMRAD ! add absorbed amount
            ENDIF
            IF (LPH) ABRAD=ABRAD+EMIS*PLANH(JJ)+FAC3S*PLANV(JJ)
            SHEATF= FAC7*(TTJ(2)-TSUR) ! upward heat flow to surface
            POWER = ABRAD + SHEATF - FAC5*TSUR*TS3 ! unbalanced flux
            IF (LOPN3) POWER=POWER+FARAD(JJ) ! fff only

          IF (LATM .AND. LSELF) THEN  !v-v-v-v-v  Adjust atmosphere temperature
            TATM4=TATMJ**4
C  ADGR is solar heating of atm
            HEATA=ADGR(JJ)+FAC9*(EMIS*TSUR4-2.*TATM4) ! net atm. heating flux
            TATMJ=TATMJ+HEATA*DTAFAC ! delta Atm Temp in 1 time step
          ENDIF                 !^-^-^-^-^

  IF (LATM) THEN  DOWNIR(IH,J4)=ATMRAD ! save downward IR flux  ELSE left as was!

            DOWNVIS(IH,J4)=ASOL(JJ)+SOLDIF(JJ) ! downward coll.+diffu. solar flux
```

10 Test results

10.1 Validation

Against 344. Minimal edit of 342/run/342v3t.inp to 344/run/344v3t.inp
difference negligable away from cap edges.

351: edit krc/Eur/351v3t.inp

10.2 New capabililties

During development, many runs were done with all the eclipse debug options enabled, which generates 6 "fort.xx" text files; the results viewed using the IDL routine krv35.pro.

Some test were done with largely realistic conditions. E.g., for the sub-Jovian longitude on Europa, using the nominal values listed in §B with thermal inertia 200 in MKS and 22 layers to a total depth of 11.8 diurnal skin depths. Cases are

0 Insolation only, no eclipse
1 add geothermal heatflow= 100mW/m^2
2 Daily eclipse at local noon, with heatflow
3 Daily eclipse at local noon, no heatflow
4 Daily eclipse at local noon, with Planetary fluxes
5 Rare eclipse at local noon, with base heatflow at start of eclipse maintained
6 Rare eclipse at local noon, with base layer temperature at start of eclipse maintained
7 Daily eclipse at local noon, 30° East slope,with heatflow and planetary fluxes
8 As above, but no planetary fluxes

Results were displayed using the IDL routine krv36.pro and some are shown in Figures 2 and Figure 3. With the values used, at the equator, basal heat-flow of 100 mW/m^2 increases T about 0.5 K, and radiation from Jupiter increases T about 3.2 K. The two methods of handling the lower boundary condition for Rare eclipses differ by < 1 nK.

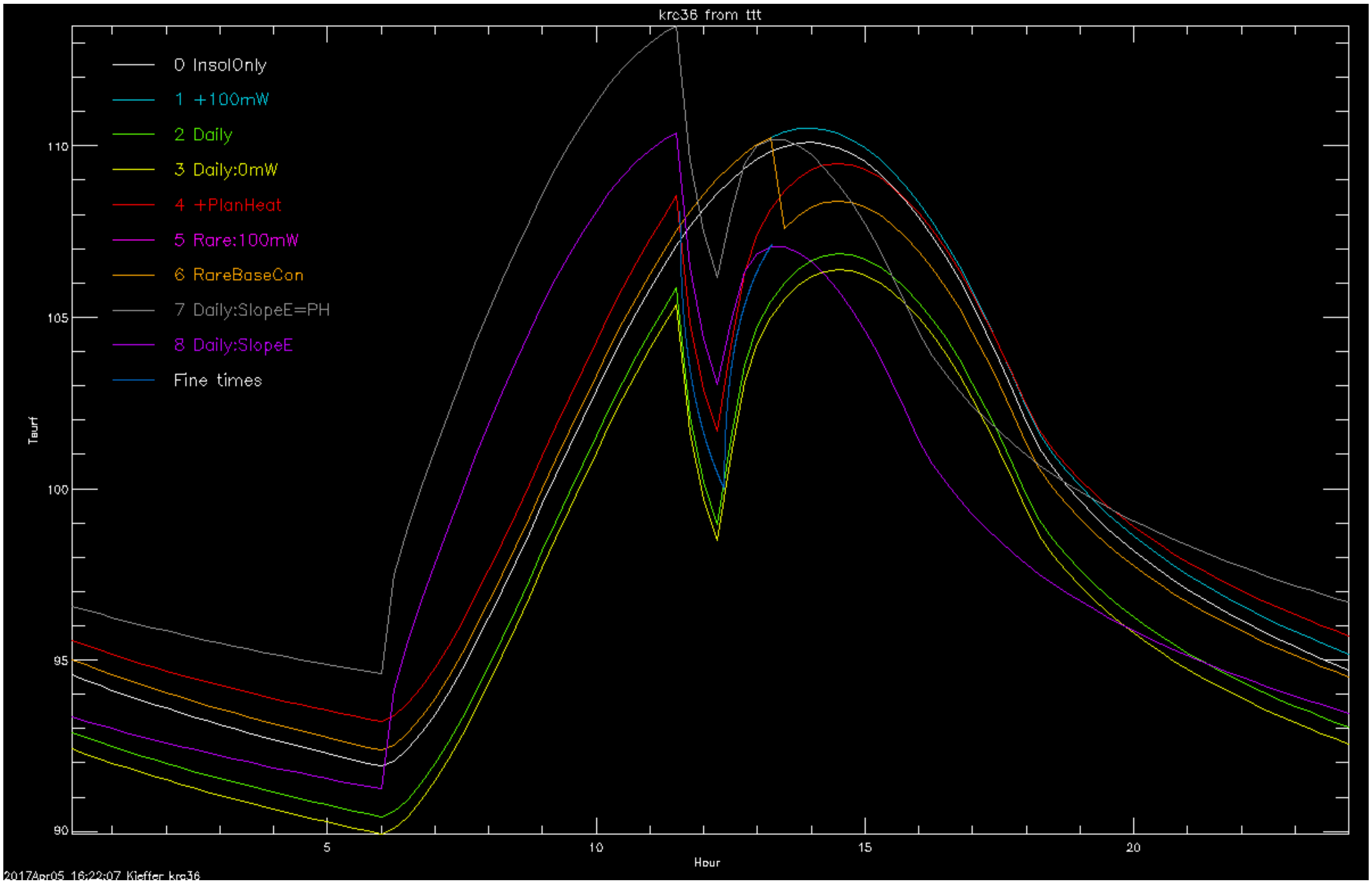

Figure 2: Europa surface temperatures for several conditions for the equator, with eclipse at local noon. Legend has an abbreviation for the conditions; see Table ??? for description eurCTs.png

11 Other version 3.5 changes

Replace EVMONO38 with EVMONO3D, which has the scaling factors firm-coded in the routine, eliminating 2 arguments. Latter routine is 9% faster

Change line: " 16 N 'ffff' " will toggle output of a binary file named ffffxx.bin5 for each case; xx will be the case number. This file will contain the surface temperature for every time step for the last season for the N'th latitude. A non-positive value of N turns this off.

Because all the KRCCOM arrays are full, add storage of N to HATCOM and use FMOON in FILCOM for the file name stem.

A Debug options new with v 3.5

TFINE always outputs *tfinexx.bin5*: ASOL, FINSOL and all layer temperatures at every fine time-step.
The file header contains N2,J7,J8,J9
array is [2+fine layer, fine-time] IDL krc35.pro reads as bbb

Optional files: Each may have more than one case

Table below: columns are:
1: Minimum IDB5 value to trigger output
2: fort.X file. P means it goes to print file. M means to Monitor
3: Routine that writes this. D=TDAY, F=TFINE. And which stage: 1 or 2

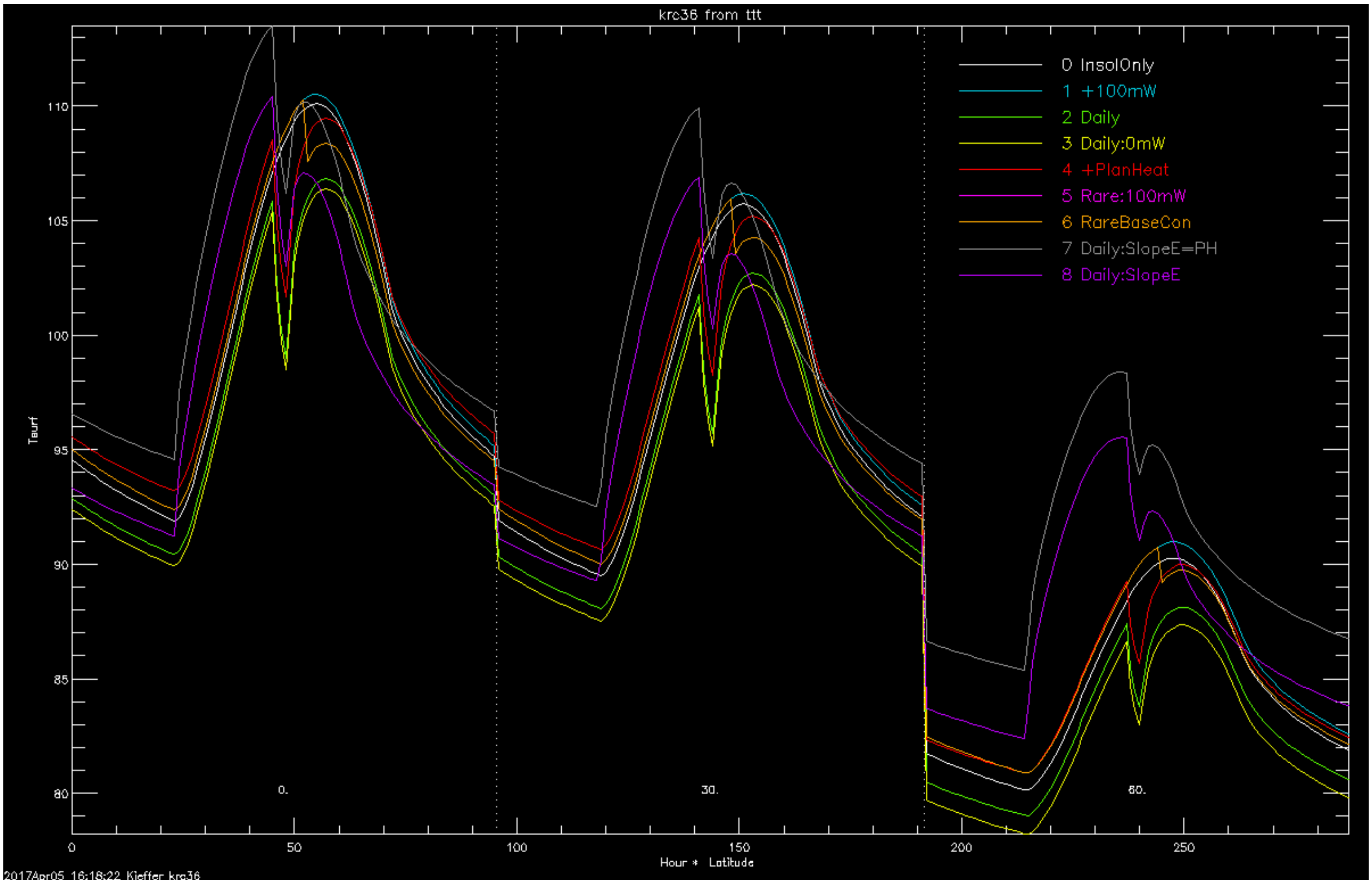

Figure 3: Europa surface temperatures for several conditions for latitudes 0, 30 and 60. See Table ??? eurClats.png

ID B5	out	St-age	Description	IDL code
1	P	F	IQ,JJ upon entry, print exit	
1	P	F1	Least stable layer and T-dep. layer set	
1	M	F1	QB.. key values	
1	P	D2	LZONE... T-dep layer ranges	
2	P	F2	Layer stability table	
2	42	F1	J,BLAF,SCONVF,QA for each fine layer	
3	P	F	Starting Tsurf, delbot	
4	43	F1	for N1 layers at start: TDAY: depth,T,splineY,c-thick,f-thick,f-depth	fff[layer,item]
	"	F1	for fine layers: depth, T, FA1, FA3	uuu[layer,item]
?	44	F2	JFI,FINSJ, TSUR,ABRAD,SHEATF,POWER,FAC7,KN each fine time near edge	ddd[ctime,item]
4	47	F2	T for fine layers and for coarse layers at end of eclipse, followed by :	vvv[item*case, layer]
4	47	D2	T for layers, just before being replace by eclipse results.	?
5	P	F	Index, center depth and initial temperature for each fine layer	
6	M	F	I,J, fine-layer factor for each layer	
7	44	F2	values for each fine time step near eclipse ends	
7	46	D2	JJ ,ATMRAD,TSUR,ABRAD,SHEATF,POWER,FAC7,KN each coarse time. Rare only.	aaa[time,item]

Notes: 1) Radiation fields do not show eclipse because they are normal for Rare eclipse
2) TSUR (and SHEATF) will show discontinuity at end of followon.

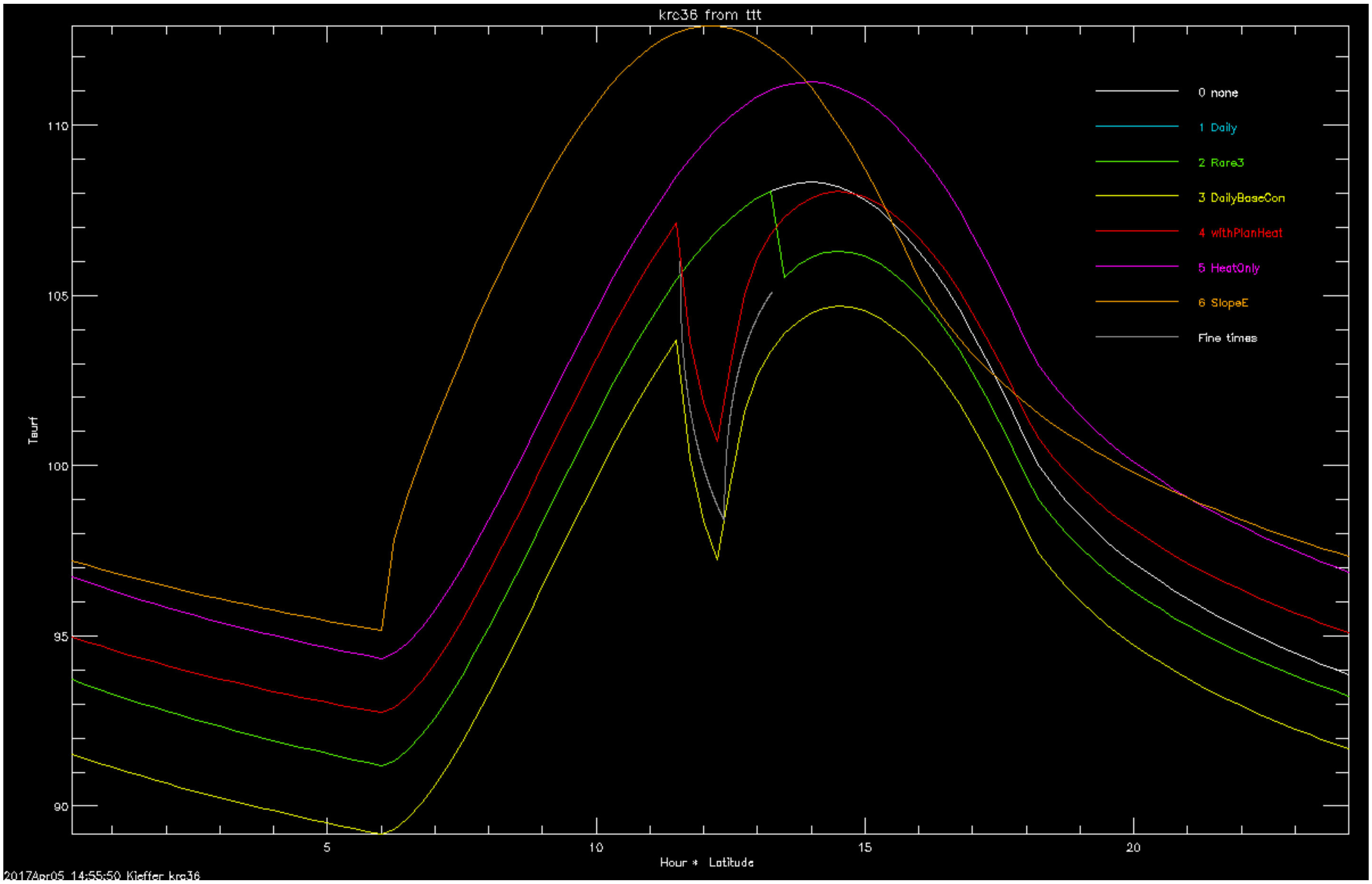

Figure 4: Results from the eurB.inp run. Nominal Europa, albedo=0.67, Inertia=200, obliquity and inclination assumed=0. Results for sub-Jupiter equator. teaser.png

B Some values for Solar system satellites

Earth:
Eclipse card for Earth lunar eclipse might be
14 3 1. 6378.2 384.4d3 1737. 29.53 0.345 6000 12. 7 / Moon
Eclipse card for Earth solar eclipse might be
14 3 1. 1737.4 384.4d3 6315. 29.53 0.345 6000 12. 7 / Earth solar
but need to account for sol not the same as lunar synodic month
a test of the routine is that with bias=0, mid-eclipse would be about 6% short of total.

Mars: eq. radius = 3396.2 km
Orbit SMA= 1.523679 AU
Period 1.8808 yr or 686.971 day
Satellites =['Phobos','Deimos']
Satellite orbit radius = 9376., 23463.2 km
Satellite radius: 11.2667, 6.2 km
Mutual period 0.3189, 1.263 day
For Phobos solar eclipse, the surface radius of Mars is a significant term.

Jupiter: eq. radius =71492. km
Orbit SMA= 5.2026 AU
Period 11.8618 yr or 4332.59 day
Satellites =['Io','Europa','Ganymede','Callisto']
Satellite orbit radius =[.4218,.6711,1.0704, 1.8827]*1.e6
Satellite radius: =[3640.,3121.6,5268,2,4820.6]
Mutual period =[1.77,3.55,7.15,16.69] days
Angular radius of Jupiter from satellite: arcsin(d/R)

0.1703 0.1067 0.0668 0.0380

Europa heat flow: 30 to 130 mW/m2: J. Ruiz, 'The heat flow of Europa', Icarus v. 177, p438:446 (2005)

Jupiter: emission temperature about 134 K, or 18.3 W/m^2/steradian.
At Europa, Jupiter is about 0.0356 steradian.
Thus thermal flux onto Europa about 0.62 W/m2

Jovian bolometric albedo about 0.73(?), so reflected radiance about 0.73 * scon/($5.2026^2 * \pi$) = 11.7 W/ster
or 0.42 W/m2 onto Europa at inferior conjunction.
Thus, planHeatcard for Europa might be
15 0.62 0. 0. 0.21 0.21 0. 12. / Jupiter heat load on Europa, nearside center

These can be compared to the solar irradiance at Jupiter of 50.53 W/m2

Saturn: Eq. radius. 60268 km
Orbit SMA= 9.5549 AU
Period 29.4571 yr or 10759.22 day
Satellites =['Enceledus','Titan','Iapetus']
Satellite orbit radius =[0.237950,1.22193,3.56082]*1.e6
Satellite radius: =[504.2,5149.,1468.6]/2. km
Mutual period =[1.370, 15.945 ,79.3215] days
Titan has atm: Psurf=147 Pa N_2+ 1.4% CH_4
Lakes and varied surface geology
Iapetus has inclination 15.5°

Neptune: r_m=24622. SMA=30.33 AU
Triton, r=1353.4, sma=354759. incl.=157 (to nep)
Psurf=1.4:1.9 Pa $N_2$2 , "geysers"

B.1 Test input files

Chronologic; several run many times. Any run older than 2017 Apr 5 13:15 should be abandoned. Many .inp files deleted.

```
0=no eclipse, D=Daily, R=Rare H=PlanetaryHeating, n=nill
cirMars = circular orbit at Mars distance, zero obliquity

 3874 Dec  9 06:46 thin9.inp Mars  5 lats, 120 days, 9 cases: vary layers
 3448 Mar 20 16:50 V35a.inp Europa 5 lats, 20 days, 4 cases: 0,D,R,0H
 3536 Mar 30 12:49 eur6.inp Europa 1 lat, 10 days, 3 cases: 0, D and R
 3168 Mar 30 14:21 phob.inp Mars, real phobos, 1 lat, 10 days, 3 cases: 0, D and R
 3195 Mar 30 14:25 phon.inp Mars, no atm,  1 lat, 10 days, 3 cases: 0, D and R
 3942 Mar 30 15:58 351v3t.inp  Mars 5-lats, 6 cases for standard V3 validation
 3255 Mar 31 06:04 phoz.inp Mars,   1 lat, 10 days, 4 cases: all 0, vary PTOTAL
 3339 Apr  2 16:22 phoc.inp cirMars 600 km Phobos 1 lat 4 cases: 0,D,R,Rn
 4127 Apr  4 23:26 eurA.inp 1lat, 20 days
 3916 Apr  5 15:38 eurB.inp Europa 1 lat,
 3916 Apr  5 15:39 eurC.inp
```

Analysis of each run using IDL kv3 calling krc35

FORTRAN routines are tested individually using testrou.f, executable testr

B.2 Notes on the need to separate radiation fields

Want to allow planetary loads when have an atmosphere.

Must separate radiation fields:
Solar incident top-of-atm. all and only these influenced by eclipse
abs in atm,
collimated at surface,
diffuse at surface,
[lost]
Atm down-going IR

Assume existing treatment included multiple reflections, messy to rederive
Planetary visible top-of-atm: PLANV
abs in atm,
abs at surface
[lost]
Planetary thermal top-of-atm: PLANH
abs in atm,
abs at surface,
[lost]
Hemispherical albedo: ALBJ

C Integer to:from real conversion

Let a real value x run from 0 to V, e.g., 0 to 2π or 0. to 24.
Let the integral indices I representing this interval run from 1 to N; the 1-based system
For notation convenience, define $R \equiv \frac{V}{N}$

```
x:  0=|++^++|++^++|++ ... ++|++^++|=V   Real representation
I:    |  1  |  2  |   ...   |  N  |     Integral representation, 1-based
M:    |  0  |  1  |   ...   | N-1 |     Integral representation, 0-based
```

Integer to real: $x = (I - \frac{1}{2})\frac{V}{N}$ or $x = (I - 0.5)R$
Real to integer: $I = \text{NINT } (x/R + .5)$
BEWARE, the default real:integer conversion in many languages is to truncate magnitude.
This results in a relationship discontinuity (no change in I) at $x = 0$.
If y is always positive: NINT(y-.5) and INT(y) are identical.

or $I = x/R + 1$ if x is non-negative and the default real:interger conversion is to truncate magnitude.

FORTRAN intrinsics for real to integer (all tested in testrou.f @28)
CEILING - Integer ceiling function
FLOOR - Integer floor function
INT - Convert to integer type identical results to I=x
INT2 - Convert to 16-bit integer type
INT8 - Convert to 64-bit integer type
LONG - Convert to integer type
NINT - Nearest whole number
FLOAT - Convert integer to default real
Only CEILING, FLOOR,and NINT are consistent across 0.
The use of NINT and FLOAT maintains integrity.

C.0.1 version testing

```
 against 344.  minimal edit of 342/run/342v3t.inp to 344/run/344v3t.inp

351: edit krc/Eur/351v3t.inp

kv3.pro

File names
  0 VerA=new DIR    200 = ~/krc/Eur/out/
  1  " case file   202  = 351v3tb
  5 VerB=prior DIR 201  = /work2/KRC/344/run/out/
  6  " case file   202  = 344v3tb


@115 123 116 123
kv3 Enter selection: 99=help 0=stop 123=auto> 550
Num lat*seas*case with NDJ4 same/diff=        1197           3

@116 makes kons=233      56      561      562      563      564      565       61      622       -1       63
```

```
@56: t
@561: 0
.
% ARRSUB: some index error, see above comment
ARRSUB error        2
SOME ERROR CONDITION at kon=     561.  Any key to Go
@12  11=0 12=4 17=0 18=-1
@561
 help,qy
QY              DOUBLE    = Array[48, 5, 40, 6]

Tsurf caseRange=all LatRange=0:4  SeasonRange=all  hour   lat  seas  case
 quilt before any other display
                   Mean       StdDev      Minimum      Maximum
         1    -0.00101340    0.0169005    -0.800859    0.0731013  signed
N=   57600     0.00268893    0.0167159      0.00000     0.800859  absolute

kv3 Enter selection: 99=help 0=stop 123=auto>  562
351v3tb - 344v3tb:  Tsurf. caseRange=all LatRange=0:4  SeasonRange=all
         -60.         -30.           0.          30.          60.
% Compiled module: MEAN_STD2.
Mean= (each case)
    0.0453352      0.00000      0.00000      0.00000      0.00000
    0.0280654      0.00000      0.00000      0.00000      0.00000
   0.00726742      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
StDev=
    0.0616917      0.00000      0.00000      0.00000      0.00000
    0.0299598      0.00000      0.00000      0.00000      0.00000
    0.0100296      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000

kv3 Enter selection: 99=help 0=stop 123=auto> 563
    Item       Mean     StdDev        Min        Max    MeanAbs     MaxAbs  0]=NDJ4
    NDJ4   -0.00167    0.07072   -2.00000    1.00000    0.00333    2.00000
    DTM4   -0.00000    0.00059   -0.00399    0.01548    0.00008    0.01548
    TTA4   -0.00036    0.00710   -0.10882    0.02495    0.00126    0.10882

QUILT3 displayed value range is      -0.10881889     0.024951097
sample is: latitude(5)  * 8 planes of case
line is: season(40)  * 1 groups of case.  Lines increase upward
SOuthern lats show the changes
  FROST4    0.05272    0.34532   -0.23976    3.60657    0.05312    3.60657
QUILT3 displayed value range is      -0.23975942       3.6065696
sample is: latitude(5)  * 8 planes of case
line is: season(40)  * 1 groups of case.  Lines increase upward
Any key to go
   AFRO4    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
  HEATMM   -0.00174    0.02303   -0.20097    0.11920    0.00585    0.20097
QUILT3 displayed value range is      -0.20096924      0.11919618
sample is: latitude(5)  * 8 planes of case
line is: season(40)  * 1 groups of case.  Lines increase upward

kv3 Enter selection: 99=help 0=stop 123=auto> 564
```

```
    Item       Mean     StdDev        Min        Max    MeanAbs     MaxAbs  0]=Lat
    Lat.    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
    elev    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000

kv3 Enter selection: 99=help 0=stop 123=auto> 565
    Item       Mean     StdDev        Min        Max    MeanAbs     MaxAbs  0]=DJU5
    DJU5    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
    SUBS    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
   PZREF    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
    TAUD    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
    SUMF    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
kv3 Enter selection: 99=help 0=stop 123=auto> 61
Maximum difference in Ls is:       0.0000000
kv3 Enter selection: 99=help 0=stop 123=auto> 62

RESULT, negligable differences away from frost edge.
```