Guide to KRC version 3.4, and background

Hugh H. Kieffer File=-/krc/Doc/V34UG.tex 2016sep05

September 6, 2016

Contents

List of Figures

1 Preamble

This accompanies an alpha release of KRC version 3.4.2 . ASU latest formal release is version 2.4.1, latest version in the ASU repository is 3.2.1 .

The prior version 3.3 has been abandoned; that alpha release should be ignored.

This document includes both a Users Guide and a detailed description of the development.

Sections 2 to 11 are a Users Guide to changes since the prior formal release.
The Users Guide ends with §12.0.2; most users should not need to read beyond that, where the tone of this document becomes closer to lab notes.

Sections 12 to 14.0.6 have some development details.

Sections 18 to 18.0.8 cover testing.

Sections 16 discusses briefly plans for additional capabilities

The appendix includes a short discussion of tuning KRC for efficiency.

One intention of including the far-flat and photometric function capabilities was to address asteroid models, including “thermal beaming” . This can be done by running a large grid of sloped models and post-processing the output files; see *Beaming.tex*

1.1 Distribution files

Organization into directories is the same as other recent KRC distributions. Files in the V3.4 alpha distribution:

- All source code and a Makefile
- Revised helplist
- Updated KRC evolution
- Several example input files for V3.4
  - Some annotated sample zone tables.
  - One lengthy file for generating a grid of slopes
- Print file and binary .t52 file generated by input files
- Ash.pdf; 2016 January comparison of KRC and Ashwin Vasavada thermal models for Mars
- simple.pdf; 2016 Feb. study of convergence and performance of KRC versus a simple, but slow, thermal model
- V34UG.pdf; this document, and its source files: *V34UG.tex, farg.tex, fard.tex, v34p.tex.*

2 Introduction

This includes a users guide for version 3.4.2 of KRC, which is in alpha release. It covers ONLY THE CHANGES from version 2.4.1, which are:

- May have a condensing gas other than CO_2. §4
- May invoke some Surface Photometric Models when there is no atmosphere. §5
- May specify Tables of material properties versus depth. §6

May include Geothermal heat flow. §7
Optional user-specified "unique" output; two specific sets are coded thus far. §10
Can now write type 52 and a direct-access file at the same time
For sloped surfaces, the far-field solar and thermal radiance can be from a flat surface or the traditional"self-heating".

Some new default values are:
TAURAT=0.25: closer to modern estimates of dust properties.
RLAY=1.12, FLAY=0.1, N2=1536, CONVG=3.
these improve the numerical accuracy at a modest cost in speed
N24=48: binary output every 1/2 hour, rather than once per hour.

With the incorporation of zone tables, the two materials specified in the 'ConUp0 ...' and 'SphUp0 ...' lines are no longer necessarily the "upper" and "lower" materials; they are here referred to as material A (normally Above, was 'upper') and material B (normally Below, was 'lower')

2.1 Notation use here

The following fonts styles have been partially implemented:
File names are shown as *filename*.
Program and routine names are shown as **PROGRM [,N]** , or simple UPPERCASE
where **N** indicates a major control index.
The last character"8", representing the double precision (8-byte) version , is often omitted.
Code variable names are shown as `variab` or `VARIAB` and within equations as `variab`.
Input parameters are shown as INPUT and within equations as INPUT

2.2 New routines

FORTRAN routines:
TFAR Open and read KRC type -n direct-access file
CUBUTERP Cubic interpolation of uniformly spaced points to higher density
TDISK Revised extensively to store/write type 52 and type -n files simultaneously
FILLMV a set of 12 routines to fill an array or any contiguous part thereof with a constant or MoVe (copy) a part of one array to another. Arrays are dimensioned with the size of the move, so the compiler can check out-of-bounds. Individual routines FILLx and MVx for each data type where x is: B=byte, I=integer*2, L=integer*4, R=real*4, D=real*8. Also:
MVDF multiplies each value by a constant factor
MVDA multiplies each value by a constant factor and adds it to the current destination
these last two are handy for linear interpolation, such as in season.

IDL routines:
KRCLAYER Compute and print KRC layer depth table from KRCCOM values
NUMGEOMLAY Compute number of geometrically thick layers for total depth
FLAYER Replicate Fortran KRC TDAY statement function
KV3 Check consistency of KRC within and between versions
this has grown almost unmanageable, 2650 lines
@535 can generate zone table for lunar-like compaction profiles.
HEMIALB Evaluate various photometric functions, especially for hemispheric albedo

3 User changes needed even if none of the new capabilities are used

1. Specification of the saturation point. Real parameters 11 and 32 were unused; they now specify the saturation temperature relation for the condensing gas. Values for CO_2 are 3182.48 and 27.9546. These values are in the latest KRC master input file.

2. The 22'nd real parameter, ARC3, was unused. It is now the minimum allowable numerical convergence factor, which in theory is 1. The default value is 0.801 as KRC has been found to be generally stable to this level. If you have any hint of instability, set this to 1.0 (and let me know).

3. The 7'th integer parameter IIB (was IB). This controls the lower boundary condition. Positive values are heat-flow, see §7. The flag values are now the negative of the earlier meanings. e.g.:
0 = insulating (as before)
-1 = constant temperature
-2 = start all layers =TDEEP and keep boundary at a constant temperature

4. The 14'th logical value, LZONE, is now a flag for use of a zone table. It should initially be 'F'; it will be set/reset automatically by specification of a zone table file name.

3.1 ALERT: Layer depths may vary with albedo or emissivity

In V342, if Temperature-dependent properties are invoked, the scaling to layer thicknesses in meters is done using the diffusivity computed from the input parameter DENS and the specific heat and conductivity computed for the global equilibrium temperature, unless zones defined the layers. I.e., if LKOFT is True and there is not a zone table, the layer thicknesses in meters will depend upon Albedo. In V321 and earlier, the scaling is always based on the input parameters DENS, SPHT and conductivity derived from INERTIA; and corresponding parameters for the lower material if used.

3.2 OnePoint mode

All the new capabilities have not been thoroughly tested in onePoint mode. The master onePoint file *Mone.inp* works. The geothermal-flow, non-CO_2 gas and photometric functions are expected to be compatible with onePoint mode, but use of optional files for zone tables or far-field temperatures is questionable and has not been tested; see the notes in §15

4 Any condensible gas

The coefficients for the saturation temperature relation for the condensing gas are now input parameters. KRC uses the Clausius-Clapeyron relation $\ln P = a - b/T$ where P is pressure in Pascal and T is temperature in Kelvin. This should be useful for Titan, Pluto, ...

Real parameter 11, ABRPHA, was unused; it is now SatPrA, the Clausius-Clapeyron coefficient 'a'. Real parameter 32, fd32, was unused; it is now SatPrB, the Clausius-Clapeyron coefficient 'b' . Values for CO_2 are a=3182.48 and b=27.9546.

The proper molecular weight should be input as real parameter 10, AMW. The Mass-fraction of mean atmosphere that is non-condensing, real parameter 13, FANON, should also be specified.

5 Surface Photometric Models

This is only active if there is not an atmosphere; the Delta-Eddington model used to handle atmospheric opacity includes the assumption that the surface is Lambertian. In some later version when more input parameters are available, it is intended to be allowed when there is an atmosphere

The 21'th real parameter, ARC2, otherwise meaningless when no atmosphere, is defined as follows:

$0 =$ Lambertian: $\frac{I}{\pi F} = A \cos i$

$\leq -1 =$ Lommel-Seeliger: $\frac{I}{\pi F} = A \cos i / (\cos i + \cos e)$

This is the result for isotropic single scattering.

$-1 < x < 0 =$ Minnaert exponent, k=-x: $\frac{I}{\pi F} = A \mu_0^k \mu^{(k-1)}, \ \mu \equiv \cos e, \ \mu_0 \equiv \cos i$

$k = 1$ is same as Lambert

$0 < x =$ Lunar-like. : $A_h = A\left(1. + x(\theta/45)^3 + 1.17(\theta/90)^8\right)$

x= +0.25 [and A=0.12] is Keihm84=[7] x=0.375 is Vasavada12=[9]

With incorporation of non-Lambertian albedos; must be thorough in defining just what is meant by the "albedo" input parameter in KRC. As of version 3.4.1, it is the hemispheric albedo (solar weighted) for incidence angle of zero; which simplifies comparisons between KRC runs but places a burden of scaling on the user of non-Lambertian photometric functions. The KRC equilibrium temperature (normal incidence, zero thermal inertia) is thus not affected by the photometric function.

[Another option might be to normalized to the apparent brightness at some standard non-opposition geometry, such as $i = 0°, e = 30°$]

The bolometric hemispherical albedo is computed as a function of incidence angle. See §13 for details.

Curves for various hemispherical albedos are shown in Figure 1

Development details are in §13.

6 Tables of material properties

A zone file name is defined by a change card: 8 25 x 'fileName' / where the 3rd column (x) is ignored. The zone file may have up to 20 lines of comments before a "C_END" line. Thereafter each row defines a zone and must contain 4 columns (additional columns are allowed but not read):

Col 1: thickness, m

Col 2: density, kg/m^3 However

-1 = use DENS [real parameter 8] the material A density

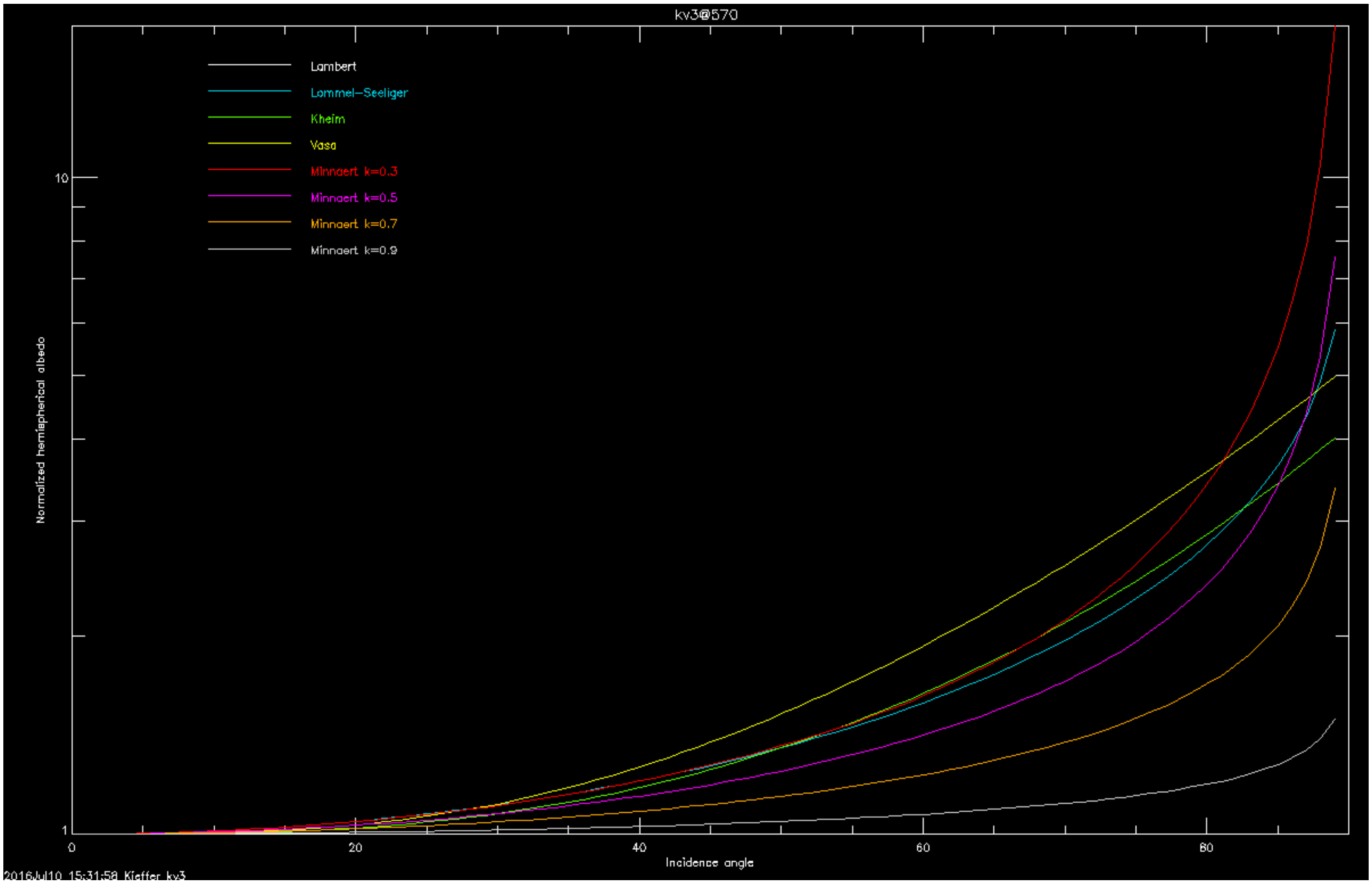

Figure 1: Dependence of hemispherical albedo on incidence angles for various models, indicated in legend. All have been normalized to unity at normal incidence. hem46n.png

-2 = use DENS2 [real parameter 5] the material B density

Col 3: Conductivity, SI Units. $\lceil$ If negative then col 4 is a pointer

Col 4: Specific heat, SI Units $\lfloor$ 1=material A , 2=material B

There must be at least 3 zones [otherwise the two materials within the current input parameters would be adequate]

The thickness of the last zone is ignored; KRC will fill out to the number of geometric layers specified by N1 [integer parameter 1]

A line with column 1 non-positive will stop reading. E.g., " 0 0 0 0 " . This may be followed by any number of comment lines.

KRC will be expand the zones into an appropriate number of layers to approximate the RLAY and FLAY relation [real parameters 32 and 33]. If the table contains zones that are near the implied geometric progression; that exact thickness value will be used. If a zone definition violates the convergence stability requirements or the number of layers specified by N1, the case will be skipped with a warning.

If the second debug parameter is set to 3, a detailed list of layer generation will be printed.

To cancel a zone table, specify a file path-name containing a single character.

Development details are in §12.

7 Geothermal heat flow

The 7'th integer parameter IIB (was IB): positive values are the geothermal heat flow in milli-Watts/m^2. Zero and negative integers set other lower-boundary conditions (see §3)

Development details are in §14.

8 Multiple output files

KRC can now write type 52 files and one of the direct-access file types at the same time. Either or both can be changed before any case.

A type 52 file is initiated by defining its name:

```
8  5 0 '<file>.t52' / Initiate multi-case output file of temperatures and other values
```

A type 52 file can be terminated by setting the file path-name to less than 4 characters; e.g., 'off'. A direct-access file opened for write is automatically closed after one case.

9 Far-field for sloped surfaces

Prior to version 3, KRC with a sloped surface assumes that the far "ground" was at the same temperature as the slope, a model called in the literature"self-heating". This assumption becomes increasingly non-realistic with increasing slope, and with slopes oriented East or West. Version 3.4 allows the far-field ground temperature to come from any prior model for which an appropriate direct-access file was saved, normally a zero-slope model of the same thermal properties; this is termed the "far-field file" or fff.

To create a type -3 model, set K4OUT=-3 and include the input line:

```
8 21 0 '<file>.tm3' / Direct-access output file for far-field
```

This may be the first case of a run. This can be in addition to saving a type 52 file. If there is no atmosphere (PTOTAL less than 1.), may use type -2 or -1, with the corresponding changes to K4OUT and the file extension.

To invoke use of a "far-flat" model, include the input line:

```
8 3 0  '<file>.tm3' / Direct-access file to read for far-field
```

Because the type of file (-1,-2 or -3) is stored in the file (as K4OUT in KRCCOM), the current value of K4OUT is ignored. Thus, it is possible to save a different type of direct-access by setting K4OUT as desired for output.

To revert to "self-heating", include an input line with full path name less than 4 characters; e.g. 'off'

```
8 3 0  'off' / Revert to self-heating for sloped cases
```

Note: For change type 8, the third argument, 0 in the above examples, is ignored.

The latitudes in a fff must be within 0.1°of those of the slope run. The seasons of fff must include the range of the slope run. The number of hours, N24, need not match, TLATS will do cubic interpolation if needed.

9.0.1 How things work: short file names as 'off'

Zone table: TCARD detects name length, sets LZONE true if length is 4 characters or more, else sets it false.

Far-field file: TCARD does not look at the name length. Any write should be closed after each case.

KRC calls TFAR(1 inside the case loop if the name length is > 3

If Far-field invoked, N2 will be limited to the values of MAXFF (firm-code as 6144)

Although TFAR can handle type -1 to -3 files, the far-field algorithm requires type -3 if there is an atmosphere; TSEAS will check for this, and an error will return code 41.

To terminate writing a type 52 file or reading a fff, define a new name of less than 4 characters lengths, such as 'off'; to change to a new file, enter its name (4 characters or more).

fff's are written as type -1,-2, or -3 by TDISK; they are read by TFAR. TDISK can read type -n files, but it is not used for that.

KRC will close any direct-access file open for write at the end of a case and turn off further writes to such files until a new direct-access file name is read.

A fff may be left open for read for multiple cases

All type -n files contain KRCCOM with the values when the file being written was closed. Because such files are closed at the end of each case, the value of K4OUT must be the same as when the file was opened for write, and thus proper for the file.

Development details are in §15.

10 Special output

The "unique" routine TUN8 is designed to address potential special requests for data output. It writes lines of text to 'fort.77', which must be renamed after a run to avoid being overwritten. There are currently two uses coded into the KRC system; more can be added.

TUN8 has two required arguments:

Arg. 1 (integer) is a code that determines what to print

existing: 101 writes temperatures for each layer.

existing: 102 writes a number of radiation items.

Arg. 2 (integer) is a stage:

1 writes case count and expected sizes

Else: writes a line of data

Two addition arguments are available to the user.

Arg. 3, Integer

Arg. 4, Real*8

Sections of code within TUN8 could write anything that is in the COMMONs.

Activity is controlled by integer parameter 15, I15. To avoid conflict/confusion with older uses of this parameter, only values greater than 100 activate TUN8.

Existing calls to TUN8:

For any value I15 of 100 or more: KRC will call: TUN8 (I15,1) for each case. Existing code:
101: TDAY will write call TUN8(I15,2,IH,(ignored)) at each output 'hour' on last day of every season of JDISK or more
102: TLATS will call TUN8(I15,2,I,SOLDIF) after TDAY done for every latitude for every season of JDISK or more.

11 Invisible changes

- Layer stability when using T-dependent materials: need stability from midday equator to winter poles. For low-obliquity or airless bodies this could be large range in temperature.

  Because KRC design has the same layer structure for all latitude and seasons, a practical compromise is to use the global equilibrium temperature: $S_o(1-A)/(2\pi U^2) = \epsilon\sigma T_g^4$.

  Layer thickness prior to V3.3 was based on the T-constant properties even if using T-dependent materials. V 3.4 uses the appropriate materials for each layer at T_g

  Note: Basing layer on T_g properties means that layering will change with ALB and EMISS for T-dependent materials.

  Layer table generation was moved into TDAY to accomplish this even with intermixed T-con and T-dep materials from zone tables.

- A third value of the TDAY input argument is used to print a layer table.

- The logic for page-header lines in the print file changed slightly.

12 Zone table Details

12.0.2 Format

Zone table format:
Up to 19 lines of comments are permitted before a required lines starting exactly as "C_END"
Zones specified from surface down. Columns white separated.
Each line specifies a zone of uniform material properties.

Col 1: zone thickness, m. Value for last row ignored; KRC fills out layers
Col 2: density, kg/m^3. -1=A material DENS, -2=B material DENS2
Col 3: Conductivity, SI Units. ⌈ If negative, then col 4 is a pointer
Col 4: Specific heat, SI Units. | 1,2=Tcon: 1=A material, 2=B material
. ⌊ 3,4=Tdep: 3=A material, 4=B material

3 or more valid lines are required. Comments may follow the 4 numeric columns
The thickness in the last valid line is ignored. KRC will fill out layers up to N1.
A non-positive first column (of 4 numeric columns) stops the input; Required.

Any number of zone lines or comments may follow the stop line

KRC will convert a zone to multiple layers as allowed by stable convergence.

Numerical stability: $(\Delta t/(\Delta Z)^2)\kappa < 1/2$ or $C_s \equiv B/\sqrt{2\kappa\Delta t}$

If definition of a zone has convergence safety factor $C_s <$ ARC3, KRC will print a layer table and quit that case.
C_s is an estimate of how many layers could be used in this zone.

12.0.3 Modifications to TDAY: Logic flow

The potential to have more than two types of material required a major change in the organization of TDAY.

V3.4 logic flow for the TDAY **setup** call:

- Compute global equilibrium temperature T_g.
- Compute properties of T-dep materials at T_g.
- Calculate a table of the sum of normalize layer thicknesses using RLAY and FLAY.
- If zone table active, LZONE true, read a zone table.
- If LZONE is true, process the zone table from top to bottom, expanding each zone into the number of layers which is the closest fit to maintaining the RLAY sequence. Generate layer-arrays for conductivity k, density ρ, and specific heat C_P. Write

each layer to the print file. Generate the virtual layer using the properties of the first physical layer.
    - If number of layers exceeds N1, print message
- If LZONE is false, generate the same 3 layer-arrays as above.
- Set a logical flag LALCON true if and only if all layers are T-con.
- If LALCON false, compute first and number of layers in the T-dep zones.
- If LALCON false, print a table of T-dep material properties versus temperature.
- If using 2 materials, check the convergence factor of the top layer of the lower material; increase its thickness if required for stability.
- Compute the center depth and the convergence safety factor for each layer.
- Compute the minimum safety factor at each layer for itself and all lower layers.
    - If minimum factor less than allowed, Set return flag to skip this case and print message.
- Compute time-doubling allowed and the resulting factors: $F_{1_i},\ F_{2_i},\ F_{3_i},\ F_{B_i},\ F_{C_i}$. See §3.2 of Kieffer12=[8]
- Build the doubling-depth comb.
- If using geothermal heat-flow, set the flag LGHF true and compute the lower-boundary temperature offset.
- Print the layer table.

V3.4 logic flow for the TDAY **season** call:

- Compute constant factors; set atmosphere and frost flags.
- Loop on days
    - Loop on time-steps
        - Set the lower boundary condition
        - Loop on layers
            - If LALCON true, ΔT_i for all layers in the comb
            - If LALCON false, compute k_i and C_{Pi} for any layers in the first and second T-dep zones.
              Then compute $F_{1_i},\ F_{3_i}$ and ΔT_i for all layers in the comb
        - Apply ΔT_i to all layers in the comb
        - Satisfy the upper boundary condition. Adjust atmosphere temperature and frost amounts as needed.
        - If on an output hour, save values.
    - Check if next day could be the last of the season
- If this is the last day; save values and return

Detailed printouts:

to monitor:

As zones are read: e.g., K,IH,LALCON= 1 0 T

K is zone (row in table) count

IH is code set by column 3. 0 means was an actual conductivity

+ 1,2,3,4 are codes for A/B material and Tcon/Tdep

LALCON is current value of "all layers thus far are Tcon"

As layers are created: e.g., Zone,I,II,ZBOT,DBOT= 22 1 2 0.79880 35.60257

Zone is zone (row in table) count

I is is count of new layers within one zone

II is upper index of layers planned in this zone

ZBOT is depth in meters of the bottom of PRIOR layer

DBOT is depth in diurnal scale of the bottom of PRIOR layer

to print file:

```
 Zon I Lay   D___bottom___m thick_m Conductiv Density SphHeat  Diffusive  Inertia
  1  1  2    0.134   0.0029  0.0029 0.1004E-01 1002.8  600.00 0.1669E-07    77.72
Zon: is 1-based index of zone in file
  I: is 1-based layer count within a zone
Lay: is KRC layer index
```

.

J,BLAY,SCONVG,QA 2 0.29042E-02 0.52716E-10 1.0

J is KRC layer

BLAY is layer thickness in m

SCONVG is convergence safety factor before any time-doubling

QA is the time doubling factor for that layer

If Layer depths are increasing too rapidly, must decrease FLAY by appropriate factor.

12.0.4 Generated layer thickness

As layer are generated from zones, maintain cumulative depth in local-scale units. For each new zone, determine the ideal first-layer thickness from the depth/thickness relation generated by FLAY and RLAY; do this by fractional interpolation in a normalized depth/thickness table. Because the summation formula is degenerate for RLAY=1, need some logic branches. Maintain total depth in D units based on N1, FLAY and RLAY. Logic flow:
- get Δz in m from zone table
    - If this is last zone, compute depth-to-go using a sum based on N1, FLAY, RLAY
- Compute desired first layer of zone based on depth thus far in D units
    - If RLAY ≤ 1 then $n = \Delta z/$ first in m
    - else: use formula in normalized units
- Round n to nearest integer
- Recompute first layer based on integral number of layers
- Loop on layers, increasing each by RLAY, keep track of $\sum$ in m and D units

$D = \sqrt{\kappa}\sqrt{P/\pi} = I/(\rho C_P)\sqrt{P/\pi}$.

For typical geologic materials, ρC_P ~1.e6, and for a diurnal period of 1 day, $D \sim I\cdot$ 1.7e-4 m

–§ **Grott 2007 profile** I wrote a small IDL function that generates the lunar-like conductivity and density profiles of Grott07=[5], which were then output at geometric series spacing close to KRC v3.4 normal; making the zone table *Grott07.tab*.

Using last layer as 15 yields a little nicer result than 14, which puts 2 layers in the last zone and has less efficient time doubling.

12.0.5 Ensure stability

If the first layer would be unstable, then can adjust RLAY to generate a stable BLAY and while not changing the total depth or the number of layers. No [easy] analytic solution; must iterate to find RLAY.

Minimum safe first layer: BSAFE= $\sqrt{2\kappa\Delta t}$ where Δt = PeriodDays*86400./N2 If BLAY lt BSAFE, then can use IDL routine NUMGEOMLAY to find RLAY that will allow FLAY to yield the needed BSAFE:

Compute the diffusivity $\kappa = k/(\rho C_P)$

Compute the diurnal scale: $D = \sqrt{\kappa P/\pi}$ where P is the spin period in seconds, = days*86400.

Set the desired total depth in D units, Z =Z(meters)/D

Set the desired number of physical layers N (N1-1)

One way:

Set the desired FLAY in D-units or compute it as B/D where B is the first physical layer thickness in m.

In IDL: yy=NUMGEOMLAY(Z,jint=N,flay=FLAY) ; yy will be the needed RLAY

Z must be real, not integer.

Can add the following to .inp file, then will run very quickly and produce a layer table.

```
2 5 2 'N5' / few seasons THIS AND BELOW USED WHEN TESTING layer structure
2 4 1 'N4' / single latitude (beware if more than one latitude row)
```

13 Photometric Function Details

13.0.6 Notation

Some notation here follows Hapke93=[6], sections therein are indicated as H[x.x] and equations by H(x.x)

i is the incidence angle from the surface normal. $\mu_0 \equiv \cos i$

e is the emergence (viewing) angle from the surface normal. $\mu \equiv \cos e$

g is phase angle

ψ is the azimuthal angle between the planes of incidence and emergence

Other symbols

I radiance, usually in the emergence direction

J irradiance, usually the incident power per unit are normal to the incidence

A_K the traditional KRC input albedo "ALB".

"TOI" means "Table of Integrals" Dwight61=[4]

13.0.7 KRC needs

KRC needs two types of albedos; the fraction of collimated light hitting a surface at local incidence angle i that is reflected, and the fraction of diffuse (presumed isotropic) light relected by a surface. KRC deals with energy, so all photometric terms are assumed to represent the bolometric value, as weighted by the solar spectrum.

Compute the absorbed direct, diffuse and bounce insolation in TLATS where it has been done. For thermal models, need an **absorption** photometric function; equivalent to $1 - A_h$ H[10.D]. KRC does not need the full BRDF.

13.0.8 Thrashing with Hapke

BRDF == Bidirectional reflectance distribution function, H[10.B]. Commonly a function of the incidence angle relative to the surface normal, i, e and g.

A_h == Directional-hemispherical reflectance (or hemispherical reflectance) is the integral of the BRDF over all viewing directions. Equivalent to the ratio of the total power scattered into the upper hemisphere to the collimated power incident on the same area, H[10.D.2]. This can be a function of incidence angle.

A_s == Bi-hemispherical reflectance (or spherical reflectance) is the reflectance of a surface under diffuse illumination, H[10.D.4]. It has no geometric dependence.

Prior to KRC v3.3, all albedoes were considered to be Lambertian.

In general, Hapke uses r for "reflectance", however, there are many of them.

A potential source of confusion is the KRC treats "albedo" as the reflected fraction of power incident on a unit area; this incident power is $J \cos i$ where i is the incident angle relative to the local surface normal. Thus, "albedo" has no units. However, Hapke generally uses reflectance as the fraction of incidence irradiance that is reflected, which is different by a factor of $\cos i$.

He defines r, the bidirectional reflectance, at the bottom of page 261 by: "The incident radiant power [collimated] per unit area of surface is $J\mu_0$, and the scattered radiance is $Jr(i,e,g)$, where $r(i,e,g)$ is the bidirectional reflectance of the surface." [10.B]

The ratio of the reflectance of a surface to that of a perfectly diffuse surface under the same conditions is $\frac{\pi}{\mu_0} r(i,e,g)$ H(10.3)

Also: H[8.E.1] starts with "the scattered radiance $I(i,e,\psi) = Jr(i,e,\psi)$. The scattering of light from a (planar) surface is in general described by its bidirectional-reflectance distribution function(**BRDF**) or $r(i,e,g)/\mu_0$ H(10.1).

Whatever the photometric function, it should obey reciprocity.

The incident power (no atmosphere) on a surface is $\mu_0 J = S_m \cos i$

The reflected fraction is $A_h(i)$ and the absorbed power is $S_m \cos i \, (1 - A_h(i))$

In Hapke terminology of his Table 8.1, the two albedos KRC needs are the:

$r_h \equiv r_{dh}$ or "hemispherical albedo" or hemispherical reflectance, $=\mathbf{A_h}(i)$ H(10.33)

spherical reflectance, his r_s, my $\mathbf{A_S}$.

I found Hapke to have many similiar reflectance terms that are not needed here, and the normalization is obscure. Here, I will simply treat the reflectance $r(i,e,\psi$ or $g)$ as the relative radiance into direction e,ψ for an irradiance from direction i; and heuristically determine the normalization required by KRC.

H[10.D.2] has "The general expression for hemispherical reflectance is $r_h(i) = \frac{1}{\mu_0} \int_{2\pi} r(i,e,g)\mu \; d\Omega_e$ H(10.10) where $d\Omega_e = \sin e \; de \; d\psi$.

H[10.D.4] has "the spherical reflectance is $r_s = \frac{1}{\pi} \int_{2\pi} \int_{2\pi} r(i,e,g)\mu \; d\Omega_e \; d\Omega_i$ H(10.21)

13.0.9 Procedure here

We shall (must) assume that the planar surface has no preferred orientation; that is, its reflectance is invarient under rotation of the surface abouts its normal relative to the scattering plane.

Apparently what Hapke means by $\int_{2\pi} d\Omega_e$ includes a normalization of $1/2\pi$, although I did not find where he stated this. Here the hemipherical and spherical albedo will be used to establish any normalization factors needed by KRC.

If $r(i,e,\psi)$ does not involve ψ (or g), which is true for the simple photometric functions considered here, then the integration over ϕ or ξ simply results in a factor of 2π

Desire that the ALB used in KRC retain its historic meaning of the fraction of energy reflected by the surface. By conservation of energy, this can never exceed unity, so must scale some of the results here for $A_h(0)$ and A_s. See table in §13.0.14,

13.0.10 Lambert

A Lambertian surface by definition has the same radiance when viewed from any direction, and that radiance scales with $\cos i$. Thus A_h must be independent of i. However, here develop the mathmatical formalities to be used for other photometric function.

$r(i,e,\psi) = A_L \frac{\mu_0}{\pi}$ H(8.13)

Hemispherical:

$$A_h(i) = \frac{1}{\mu_0} \int_{\psi=0}^{2\pi} \int_{e=0}^{\pi/2} r(i,e,\psi) \cdot \cos e \; \sin e \; de \; d\psi \qquad \text{eq : Ahr} \tag{1}$$

$$A_h(i) = \frac{2\pi}{\mu_0} \int_{e=0}^{\pi/2} r(i,e,\psi) \cdot \cos e \; \sin e \; de \qquad \text{eq : gah} \tag{2}$$

$$= \frac{2\pi}{\mu_0} \int_{e=0}^{\pi/2} A_L \frac{\mu_0}{\pi} \cdot \cos e \; \sin e \; de$$

$$= \frac{2\pi}{\mu_0} A_L \frac{\mu_0}{\pi} \int_{e=0}^{\pi/2} \cos e \; \sin e \; de$$

$$= 2A_L \left[_{e=0}^{\pi/2} \; \frac{\sin^2 e}{2} \right| = 2A_L \left[1/2 - 0\right| = A_L$$

Spherical:

$$A_s = \frac{1}{\pi} \int_{2\pi} \int_{2\pi} r(i,e,g) \cdot \cos e \; d\Omega_e \; d\Omega_i$$

$$A_s = \frac{1}{\pi} \int_{i=0}^{\pi/2} \int_{\psi=0}^{2\pi} \int_{e=0}^{\pi/2} \int_{\xi=0}^{2\pi} r(i,e,\psi) \cdot \cos e \sin i \sin e \; d\xi \; de \; d\psi \; di \qquad \text{eq : Asr} \tag{3}$$

$$= 4\pi^2 \frac{1}{\pi} \int_{i=0}^{\pi/2} \int_{e=0}^{\pi/2} r(i,e,\psi) \cdot \sin i \; \cos e \sin e \; de \; di \qquad \text{eq : gas} \tag{4}$$

$$= 4\pi \int_{i=0}^{\pi/2} \int_{e=0}^{\pi/2} \frac{A_L}{\pi} \cos i \cdot \sin i \; \cos e \; \sin e \; de \; di$$

$$= 4\pi \frac{A_L}{\pi} \int_{i=0}^{\pi/2} \left[_{e=0}^{\pi/2} \; \frac{sin^2 e}{2} \right] \cos i \sin i \; di$$

$$= 4 \frac{A_L}{2} \left[_{i=0}^{\pi/2} \; \frac{sin^2 i}{2} \right] = A_L \qquad ; \qquad \frac{A_S}{A_h(0)} = 1.$$

Can use the forms of Eq. 2 and Eq. 4 for other photometric functions.
Comparing Eq. 2 and Eq. 4, find

$$A_s = 4\pi \int_{i=0}^{\pi/2} \frac{\mu_0}{2\pi} Ah(i) \cdot \sin i \; di = 2 \int_{i=0}^{\pi/2} Ah(i) \cdot \cos i \; \sin i \; di \qquad \text{eq : Ahs} \tag{5}$$

Confirm by substitution for Lambert, Minnaert, Lommel-Seliger.

13.0.11 Minnaert

$r(i,e,\psi) = A_M \mu_0^\nu \mu^{(\nu-1)}$ where: $0 < \nu \leq 1$ H(8.14)

When $\nu = 1$, $r = A_M \mu_0$ is Lambert behaviour and $A_M = A_L/\pi$

Minnaert breaks down at limb, where $\mu = 0$

$$A_h(i) = \frac{2\pi}{\mu_0} \int_{e=0}^{\pi/2} A_M \cos^\nu i \cos^{\nu-1} e \cdot \cos e \; \sin e \; de$$

$$A_h(i) = A_M \frac{2\pi}{\mu_0} \left[_{e=0}^{\pi/2} \; - \frac{\cos^{\nu+1} e}{\nu+1} \right| \cos^\nu i$$

$$A_h(i) = A_M \frac{2\pi}{\mu_0} [\frac{1}{\nu+1}] \mu_0^\nu = A_M \frac{2\pi}{(\nu+1)} \mu_0^{\nu-1}$$

Spherical:

$$A_s = 4\pi \int_{i=0}^{\pi/2} \int_{e=0}^{\pi/2} A_M \cos^\nu i \cos^{\nu-1} e \cdot \sin i \; \cos e \; \sin e \; de \; di$$

$$= 4\pi A_M \int_{i=0}^{\pi/2} \int_{e=0}^{\pi/2} \cos^\nu i \sin i \; di \quad \cos^\nu e \sin e \; de$$

$$= 4\pi A_M \int_{i=0}^{\pi/2} \left[_{e=0}^{\pi/2} \; - \frac{\cos^{\nu+1} e}{\nu+1} \right| \; \cos^\nu i \sin i \; di$$

$$= 4\pi A_M \left[\frac{1}{\nu+1}\right] \left[{}^{\pi/2}_{i=0} \ -\frac{\cos^{\nu+1} i}{\nu+1}\right|$$

$$= 4\pi A_M \left[\frac{1}{\nu+1}\right] [\frac{1}{\nu+1}] = A_M \frac{4\pi}{(\nu+1)^2}$$

$$P_f \equiv A_h(0)/A_M = \frac{2\pi}{(\nu+1)} \qquad ; \qquad P_s \equiv \frac{A_s}{A_h(0)} = \frac{2}{\nu+1} \qquad \text{eq : aam} \tag{6}$$

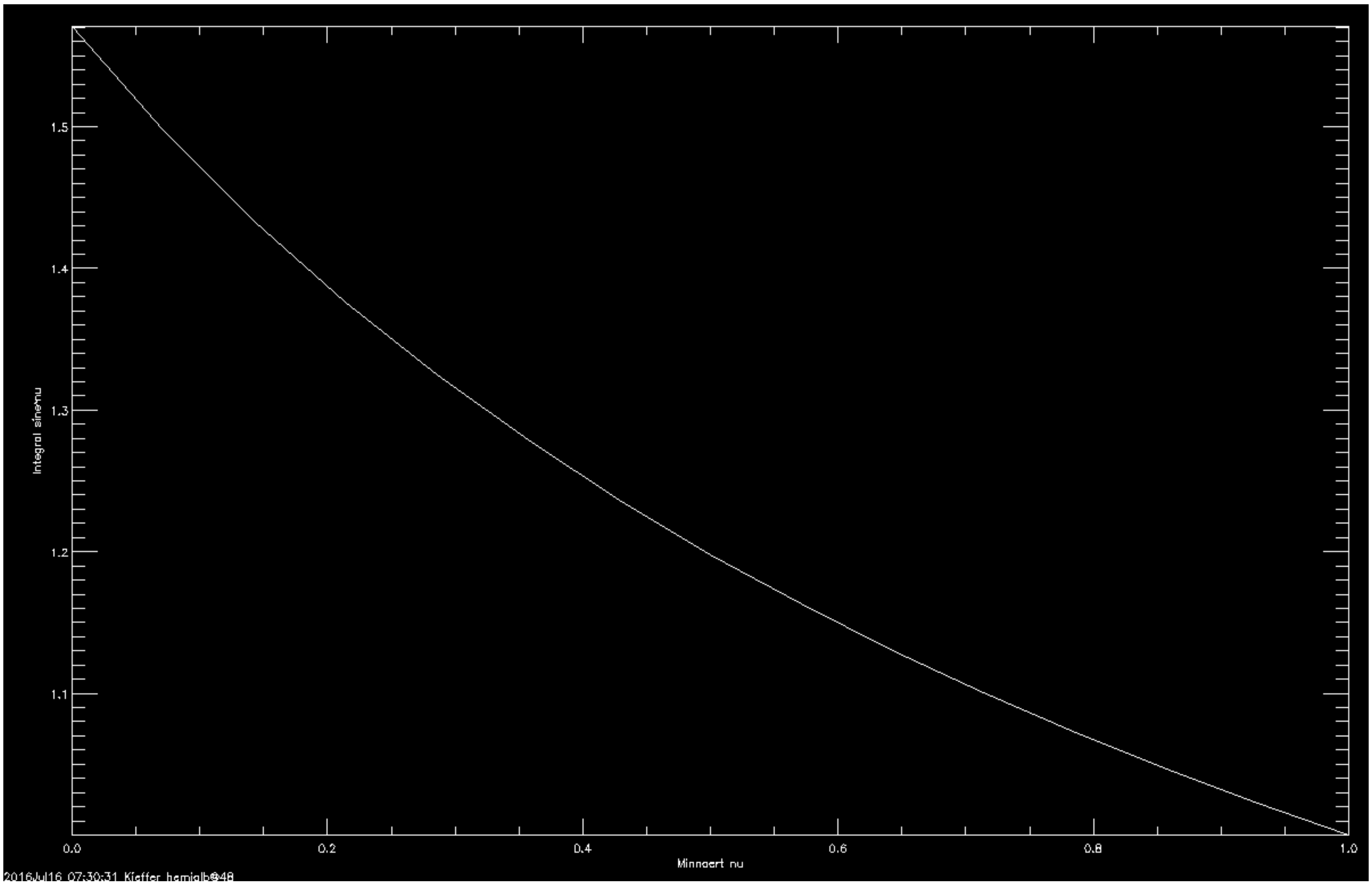

Figure 2: Spherical albedo for Minnaert reflectance. minnHemi.png

13.0.12 Lommel-Seeliger

$r =$ Lambert $/4(\cos i + \cos e)$, H(6.11) and H(6.12), thus $r = \frac{A}{4\pi}\frac{\cos i}{\cos i + \cos e}$
Hemispherical:

$$A_h(i) = \frac{2\pi}{\mu_0} \int_{e=0}^{\pi/2} \frac{A}{4\pi} \frac{\cos i}{\cos i + \cos e} \cdot \cos e \ \sin e \ de$$

$$= \frac{2\pi}{\mu_0} \frac{A}{4\pi} \cos i \int_{e=0}^{\pi/2} \frac{\cos e}{\cos i + \cos e} \ \sin e \ de$$

$$= \frac{A}{2\mu_0} \cos i \int_{e=0}^{\pi/2} \frac{\cos e \ \sin e}{\cos i + \cos e} \ de \qquad \text{eq : LSAha} \tag{7}$$

$$= \frac{A}{2} \left[{}^{\pi/2}_{e=0} \ \cos i \ln(\cos i + \cos e) - \cos e\right|$$

$$A_h(i) = \frac{A}{2}\left(\cos i \ln\frac{\cos i}{\cos i + 1} + 1)\right) \;= \frac{A}{2}\left(\mu_0 \;\ln\frac{\mu_0}{1+\mu_0} + 1\right) \qquad \text{eq : LSAh} \tag{8}$$

Spherical:

$$A_S = 4\pi \int_{i=0}^{\pi/2}\int_{e=0}^{\pi/2} \frac{A}{4\pi}\frac{\cos i}{\cos i + \cos e}\cdot \sin i\;\cos e\;\sin e\;de\;di$$

$$= 4\pi\frac{A}{4\pi}\int_{i=0}^{\pi/2}\cos i \sin i\;\int_{e=0}^{\pi/2}\frac{\cos e\;\sin e}{\cos i + \cos e}\;de\;di$$

$$= A\int_{i=0}^{\pi/2}\cos i \sin i \left[_{e=0}^{\pi/2}\;\cos i \ln(\cos i + \cos e) - \cos e\right|\;di$$

$$= A\int_{i=0}^{\pi/2}\cos i \sin i\left[\underbrace{1}_{1} + \underbrace{\cos i \ln \cos i}_{2} - \underbrace{\cos i \ln(1+\cos i)}_{3}\right]\;di \qquad \text{eq : LSAs} \tag{9}$$

Numerical integration yields A 0.20483 . Using Wolfram integral solver for each part, many opportunities for a blunder:

$$A_s = A\left[_{i=0}^{\pi/2}\;\underbrace{\frac{\sin^2 i}{2}}_{1} - \underbrace{\frac{1}{9}\cos^3 i\;(3\ln(\cos i) - 1)}_{2}\right.$$

$$\left.-\underbrace{\frac{1}{36}\left(-3\cos 2i + \cos 3i - 6\ln\cos\frac{i}{2} + \cos i(15 - 9\ln(1+\cos i)) - 3\cos 3i \ln(1+\cos i) - 9\ln(1+\cos i)\right)}_{3}\right|$$

$$A_s = 0.20457A \qquad \text{eq : LSAa} \tag{10}$$

$$A_s = A\left(\underbrace{\frac{1}{2} - 0}_{1} - \underbrace{\frac{1}{9}[0 - (-1(3\cdot 0 - 1)]}_{2}\right.$$

$$\left.-\underbrace{\frac{1}{36}[-3(-1-1) + (0-1) - 6(\ln\sqrt{2} - 0) + (0 - 1(15 - 9\ln 2)) - 3(0 - \ln 2) - 9(0 - \ln 2)}_{3}\right)$$

Yields 0.542315 A, some blunder in part 3 above.

$$P_f \equiv A_h(0)/A = \frac{\ln(1/2)+1}{2} = 0.153426 \qquad ; \qquad P_s \equiv \frac{A_s}{A_h(0)} = 1.3333333 = 4/3 \qquad \text{eq : aal} \tag{11}$$

13.0.13 Lunar-like

Keihm84=[7] does not list a BRDF, but has hemispherical albedo (his equation A5):

$$A(\theta) = 0.12 + 0.03\,(\theta/45)^3 + 0.14\,(\theta/90)^8$$

Vasavada12=[9] measurements with Diviner led to the form:

$$A(\theta) = A_0 + 0.045\,(\theta/45)^3 + 0.14\,(\theta/90)^8$$

In both cases θ in degrees is equivalent i in radians. See Appendix §A for background.
Implimentation in KRC 34 is a slightly more generalized form:

$$A_h(i) = A\left(1. + x\,(\theta/45)^3 + (0.14/0.12)\,(\theta/90)^8\right) = A\left(1. + x\,\underbrace{(\frac{4}{\pi})^3}_{f3}\,i^3 + \underbrace{\frac{0.14}{0.12}(\frac{2}{\pi})^8}_{f8}\,i^8\right) \tag{12}$$

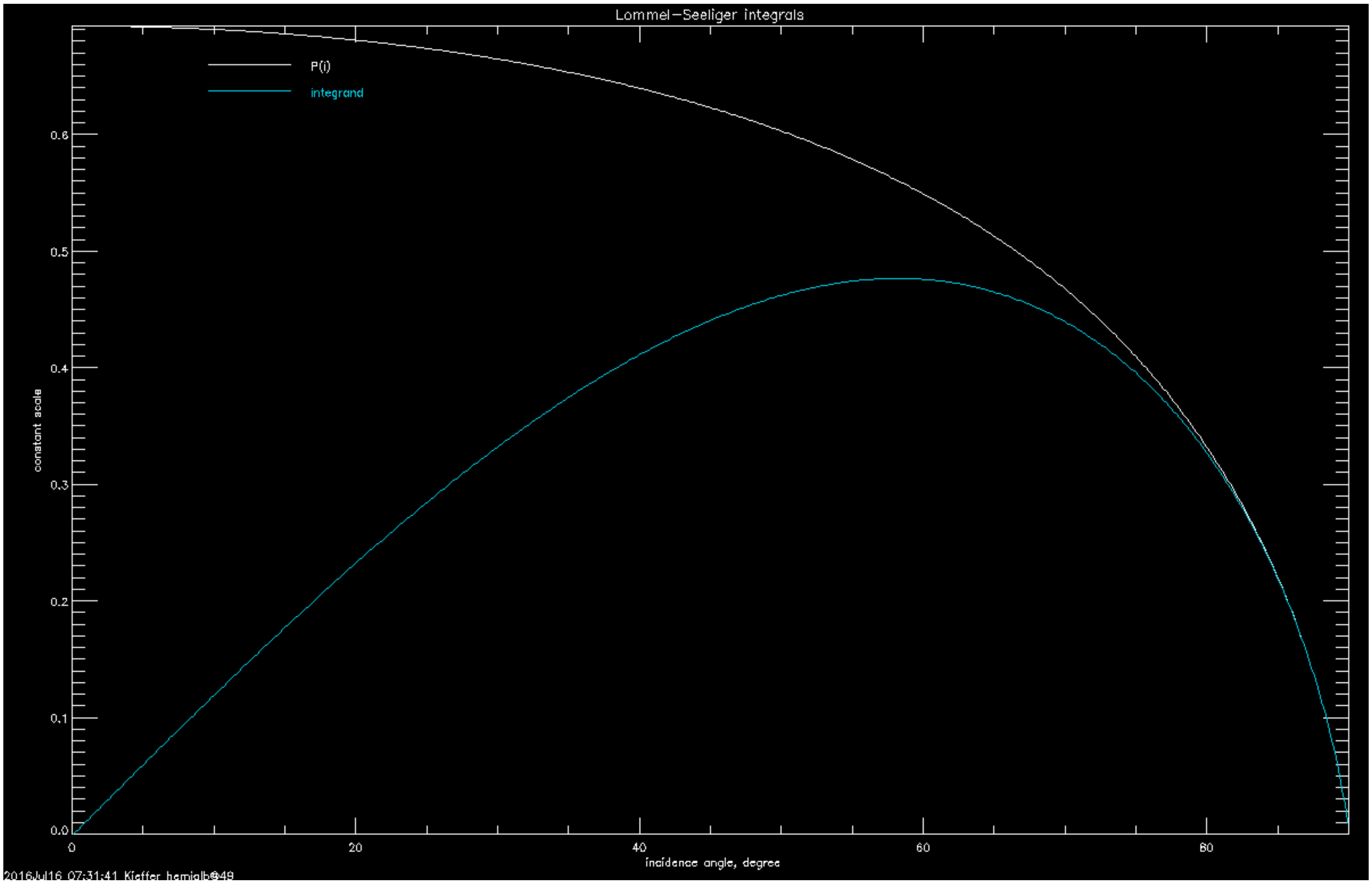

Figure 3: Hemispherical albedo for Lommel-Seeliger reflectance, and the integrand for A_s LomSee.png

Thus $x = 0.25$ yields Keihm and $x = 0.375$ yields Vasavada for $A = 0.12$.

$$A_s = 2A \int_{i=0}^{\pi/2} \left(1. + x f_3 i^3 + f_8 i^8\right) \cos i \sin i \; di = A\,[1. + 2 f_3 D_3 \; x \; + 2 f_8 D_8]$$

where (Wolframalpha.com)
$\int x^3 \cos x \sin x \; dx = \frac{3}{8}(2x^2-1)\sin x \cos x - \frac{1}{8}(2x^2-3)\cos(2x) + constant$ and $\big[_0^{\pi/2} = \frac{1}{32}\pi(\pi^2-6) \sim 0.37989752 \equiv D_3$
$\int x^8 \cos x \sin x \; dx = \frac{1}{4}(4x^6 - 42x^4 + 210x^2 - 315)\sin(2x) - \frac{1}{8}(2x^8 - 28x^6 + 210x^4 - 630x^2 + 315)\cos(2x) + constant$ and $\big[_0^{\pi/2} = \frac{80640 - 20160\pi^2 + 1680\pi^4 - 56\pi^6 + \pi^8}{1024} \sim 0.944125 \equiv D_8$

$$A_s = A\,(1.0 + 1.5683 \; x \; + 0.05944) \qquad \text{eq : KAs} \tag{13}$$

$$P_f \equiv A_h(0)/A = 1 \qquad ; \qquad P_s \equiv \frac{A_s}{A_h(0)} = 1.05944 + 1.5683 \; x \qquad \text{eq : aak} \tag{14}$$

See Figure 4
The following coded in IDL but not needed; eventually good agreement between analytic and numeric integration.

$$A_s = 2A \left[_{i=0}^{\pi/2} \; \underbrace{\frac{\sin^2 i}{2}}_{p1} + x \underbrace{\overbrace{\frac{64}{\pi^3}}^{f3} \cdot \left[\overbrace{\frac{3}{8}(2i^2-1)\sin i \cos i}^{b21} - \overbrace{\frac{i}{8}(2i^2-3)\cos 2i}^{b22} \right]}_{p2} \right. \qquad \text{eq : LLAs} \tag{15}$$

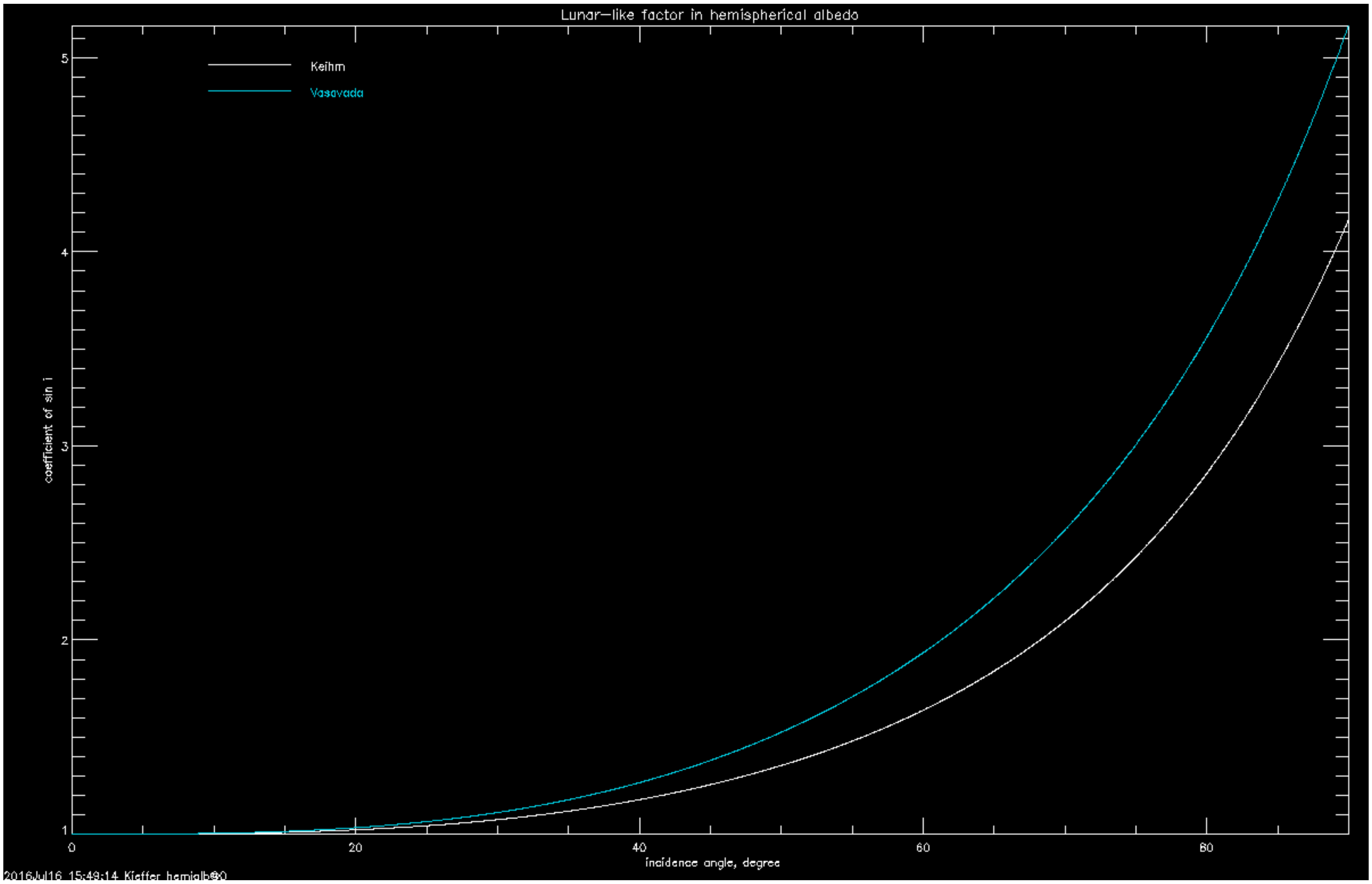

Figure 4: Lunar-like hemispherical albedo. KeihmAh.png

$$+\underbrace{\overbrace{\frac{0.14}{0.12}\frac{256}{\pi^8}}^{f8}\cdot\left[\overbrace{\frac{i}{4}(4i^6-42i^4+210i^2-315)\sin 2i}^{b31}-\overbrace{\frac{1}{8}(2i^8-28i^6+210i^4-630i^2+315)\cos 2i}^{b32}\right]}_{p3}\Bigg|$$

```
in hemialb.pro @ 46,48
f3,f8:       2.0640982       0.031476598
sum3,8=      0.38127627      0.95209557
D3,D8=       0.37989752      0.94412538
qsum=        0.72721736      0.82559132
        i          b21         b22           b31      b32        2A factor
   1.5707963   0.0000000  -0.3798975   0.0000000  38.4308746  -0.5136367
   0.0000000  -0.0000000  -0.0000000  -0.0000000  39.3750000  -1.2393910
   9.0351430e-17      -0.37989752   3.6271733e-16     -0.94412538      0.72575430
del P2,3      0.37989752      0.94412538
fun3,8=      0.38127627      0.95209557
P1                FLOAT     =       0.500000
P2                DOUBLE    =        0.78414580
P3                DOUBLE    =       0.029717855
final terms:       1.0000000       1.5682916      0.059435709
Using x=     0.250000      0.375000
total factor       1.4515086       1.6475451
```

If wish to try a lunar-like in the form $A(1+bi^a)$, integral: $x^a \sin x \cos x dx = -2^{-a-3}x^a(x^2)^{-a}((-ix)^a\Gamma(a+1,2ix)+(ix)^a\Gamma(a+1,-2ix))$ where $\Gamma(a,x)$ is the incomplete gamma function.

But using $(\cos i)^a$ would be easy: $\int \cos^a x \sin x \cos x dx = \int \cos^{a+1} x \sin x dx = -\frac{\cos^{a+2} x}{a+2} + constant$

Possible form

$$A \equiv r_h = \frac{a}{1+\frac{1-b}{b}\mu^c} \qquad \mathrm{eq:hemi} \tag{16}$$

a is the albedo for grazing incidence, b the backscatter ratio (Albedo at zenith / albedo at horizon) and c is a sharpness factor;
$c = 1$ is linear from zenith to horizon
$c =$ small drops off quickly at the horizon
$c =$ large rises quickly at zenith
All these have the non-physical propery of discontinuous 2nd derivative if the Sun gets to the zenith.

13.0.14 All

The following table lists A_h and A_s factors for several reflectance types.
yf is $A_h(0)$
ys is the spherical albedo based on $A_h(i)$
ys/yf is the spherical albedo based on $A_h(i)$ with $A_h(0)$ scaled to unity.
values greater than 1.0 are non-physical.

yid=	Lambert	Lomm-See	Kheim	Vasa	Minnk0.3	Minnk0.5	Minnk0.7	Minnk0.9
yf=	1.00000	0.15343	1.00000	1.00000	4.83322	4.18879	3.69599	3.30694
ys=	1.00000	0.50000	1.92259	2.28435	7.43572	5.58505	4.34823	3.48099
ys/yf	1.00000	3.25889	1.92259	2.28435	1.53846	1.33333	1.17647	1.05263

Each of these relations has hemispherical albedo decreasing as insolation becomes more oblique; Lommell-Seeliger is close to $cos^{0.3}i$ from 0 to 45°; see Fig. 5 and 6. $A_h(i)$ increases with i for all models here. All but Lambert and high-k Minnaert have the property of becoming larger than 1 at high incidence angles, so that the absorped power can become negative for reasonable values of ALB, see Figure 7. The only practical way to prevent such a creation of energy for all photometric models is to invoke a lower limit of 0 on adsorbed insolation; TLATS restricts $0 \leq A_H \leq 1$.

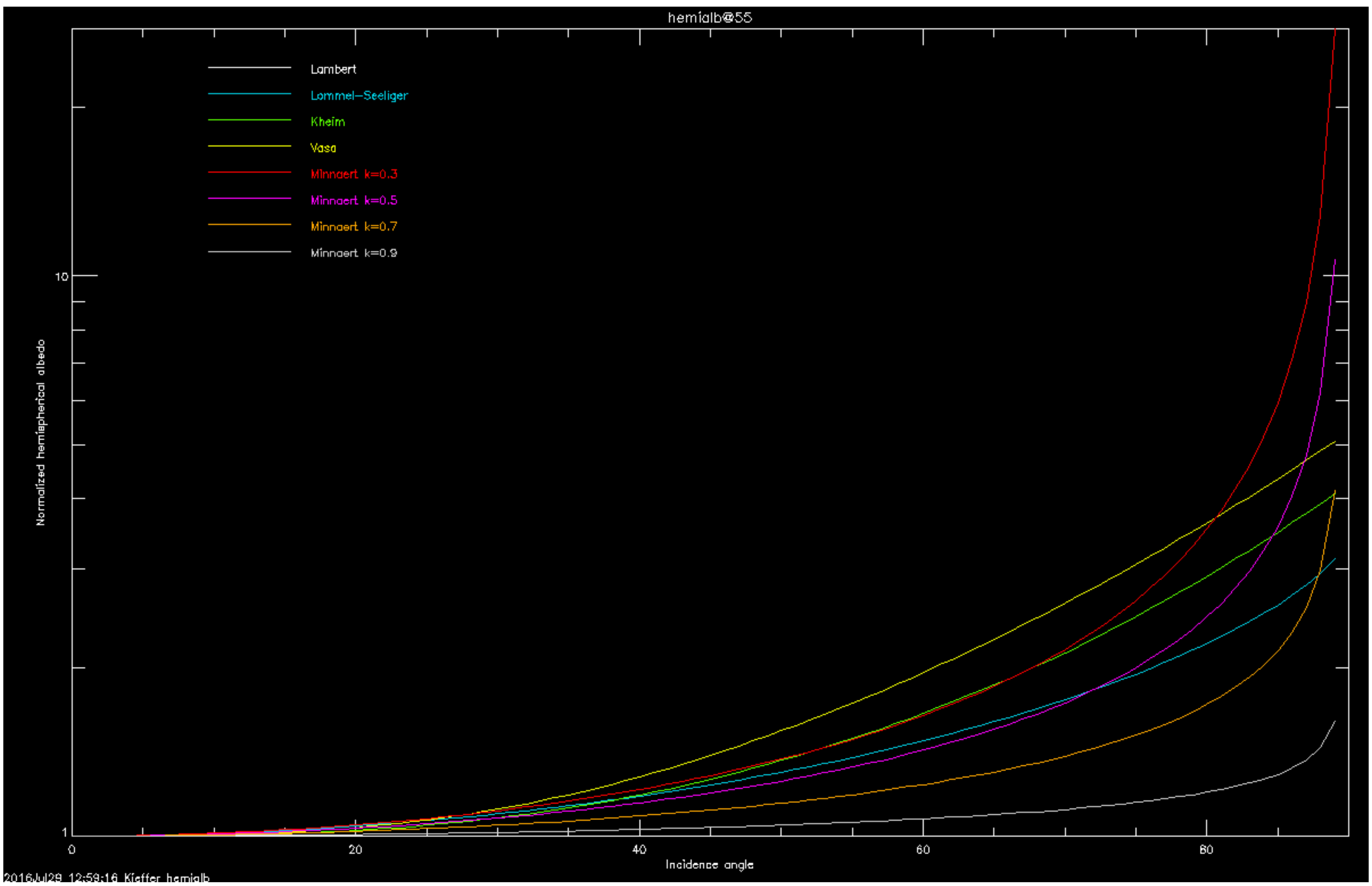

Figure 5: Hemispherical albedo as a function of incidence angle for several photometric models; see legend. Each curve normalized to the value at normal incidence. hem55n.png

13.1 Code in KRC

Version 3.4 does not allow photometric functions when there is an atmosphere because of input parameter overloading.
The use of A_h and A_s is shown in §15

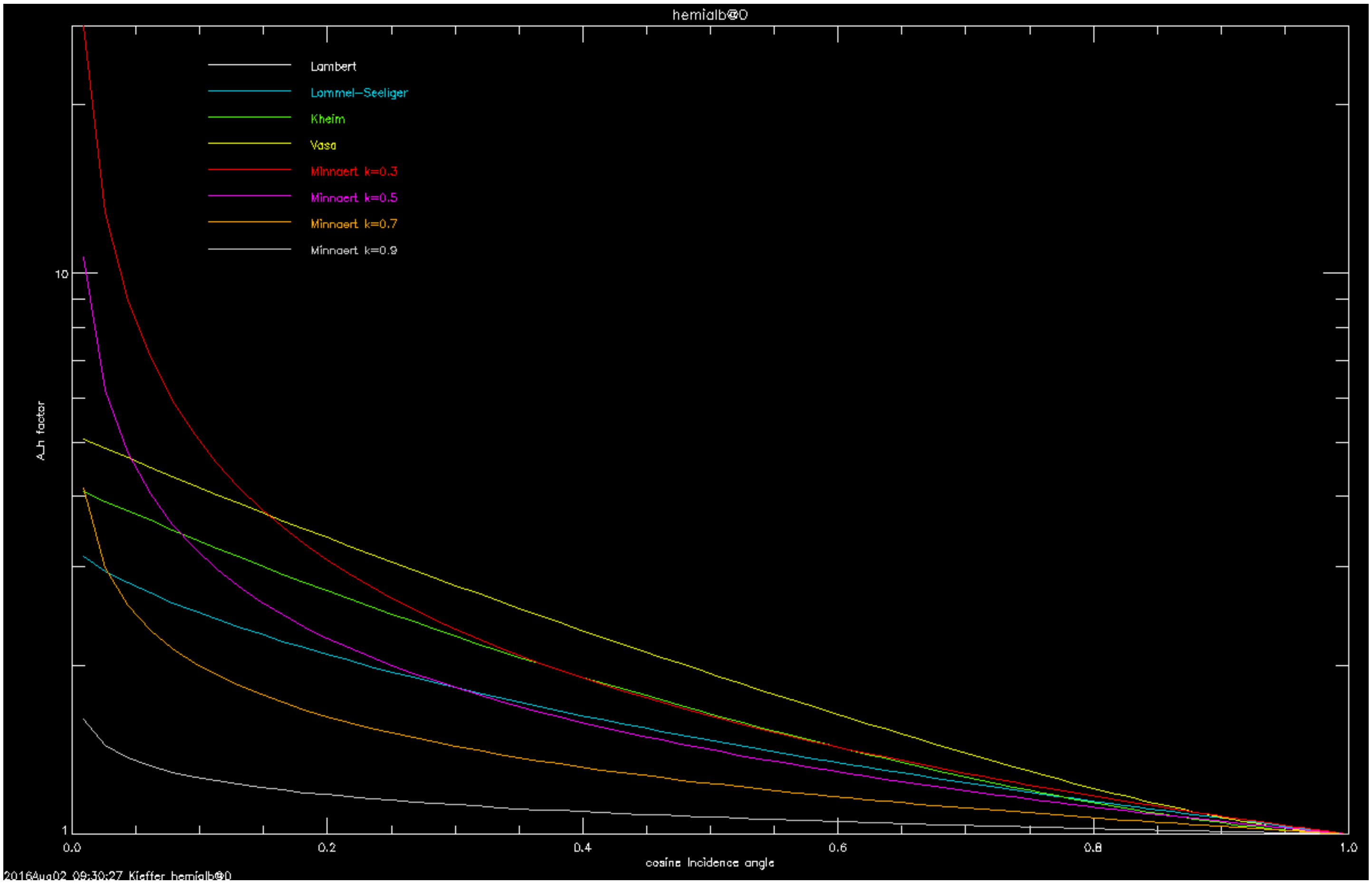

Figure 6: Same as Fig. 5 but with abscissa being $\cos i$. hem55c.png

SALB is A_s and is a constant for a case.

Frost is always considered Lambertian

Each photometric function is normalized to unity at normal incidence, so that the midday temperatures for low thermal inertia would be the same. The same scaling is applied to the spherical albedos. Thus, the absorbed flux, using the traditional KRC input parameter `ALB`$\equiv A$:

Collimated: $(1 - AP_f) \cos i$ where $P_f = A_h(i)/A_h(0)$ and AP_f=`HALB=ALBJ(JJ)` for the sloped surface.

Diffuse and bounce: $(1 - AP_s)$ where $P_s = A_S/A_h(0) =$ `ASF` and AP_s=`SALB`

Items in COMMON (see also the comments in TLATS)

```
krcccom ALB                Input albedo
hatccom SALB             ! spherical albedo of the soil
hatccom ALBJ(MAXN2)      ! hemispherical albedo at each time of day
hatccom SOLDIF(MAXN2)    ! Solar diffuse (with bounce) insolation at each time W/m^2
dayccom ASOL(MAXN2)      ! Direct solar flux on sloped surface at each time of day
dayccom ADGR(MAXN2)      ! Atm. solar heating at each time of day
```

Access to photometric models; in version 3.4.2 only when no atmosphere.

Set PTOTAL to 0.1: 1 12 .1 'PTOTAL' /

Set photometric model by ARC2: 1 21 x 'ARC2/PHT' / where x is:

0. is Lambert (the default with an atmosphere)

-1. is Lommel-Seeliger

$-1 < x < 0$ is Minneart with exponent $|x|$

$0 < x < 1.$ is Lunar-like, with x being the coefficent of i^3

Hemispheric albedos for the implimented photometric functions are shown Figure 8

The effect on surface temperature, relative to a Lambertian surface, is shown in Figure 9

The effect at all hours, latitudes, seasons and cases is shown in Figure 10

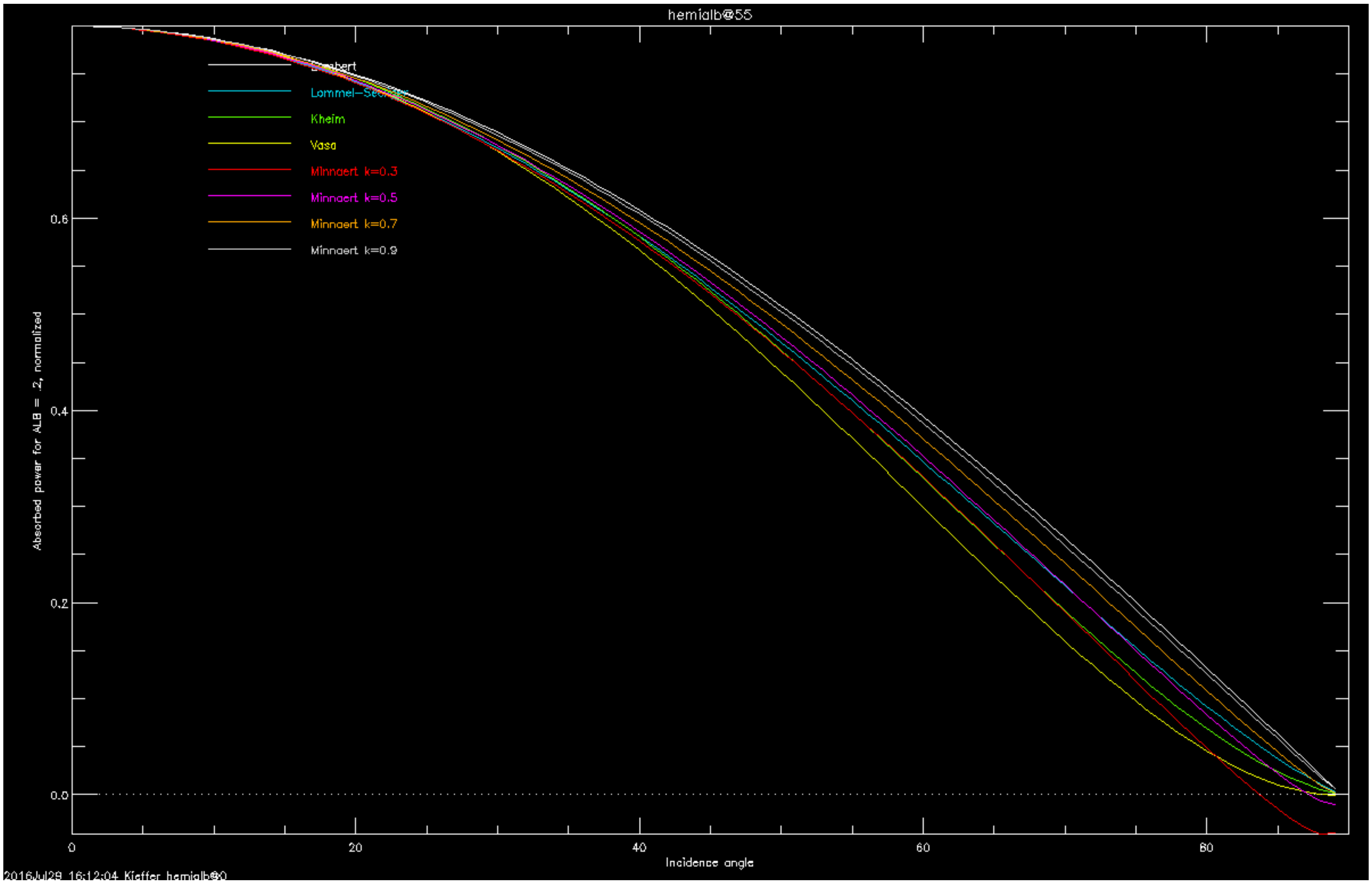

Figure 7: Absorbed power for ALB=.2 as a function of incidence angle for several photometric models; see legend. Values are $\cos i\ (1. - 0.2A_h(i)/A_h(0)$. hem55a.png

13.2 TLATS: Sequence within the hour-angle loop

.

If not atm., twilight forced to zero

Compute μ_0 (angle onto flat terrain)

    If twilight, adjust μ_0

If slope, compute i_2; no consideration of twilight

    compute α and G_1

Compute hemispheric albedo of the surface, based in [adjusted] μ_0

    If Far, frost albedo is thick-deposit value based on Tfar, as have no frost-amount.

Compute C=Collimated beam and boundary fluxs; use DE if an Atm.

Compute D=Diffuse, which does not depend on slope

Compute B=Bounce flux.

Sum C+D+B at each time

  If twilight:

    flux onto top of atmosphere unchanged, so atm heating should not be changed

    diffuse flux out bottom of atm is extended by cos3

    total energy should not change, so need to scale [C+D] by their diurnal sum.

Delta-Eddington always uses the Lambert albedo, otherwise becomes complex (undefined) to consider the twilight region. Thus, surface photometric fuction considered only for Collimated beam. However, frost is always treated as Lambertian.

Sky factor for a pit with wall slope s

Solid angle is

$$\int_0^{2\pi} \int_0^{\theta} \sin x \ dx \ d\phi = 2\pi \left[-\cos x\right]_0^{\theta} = 2\pi(1 - \cos\theta)$$

where $\theta = 90 - s$ and x is the angle from zenith

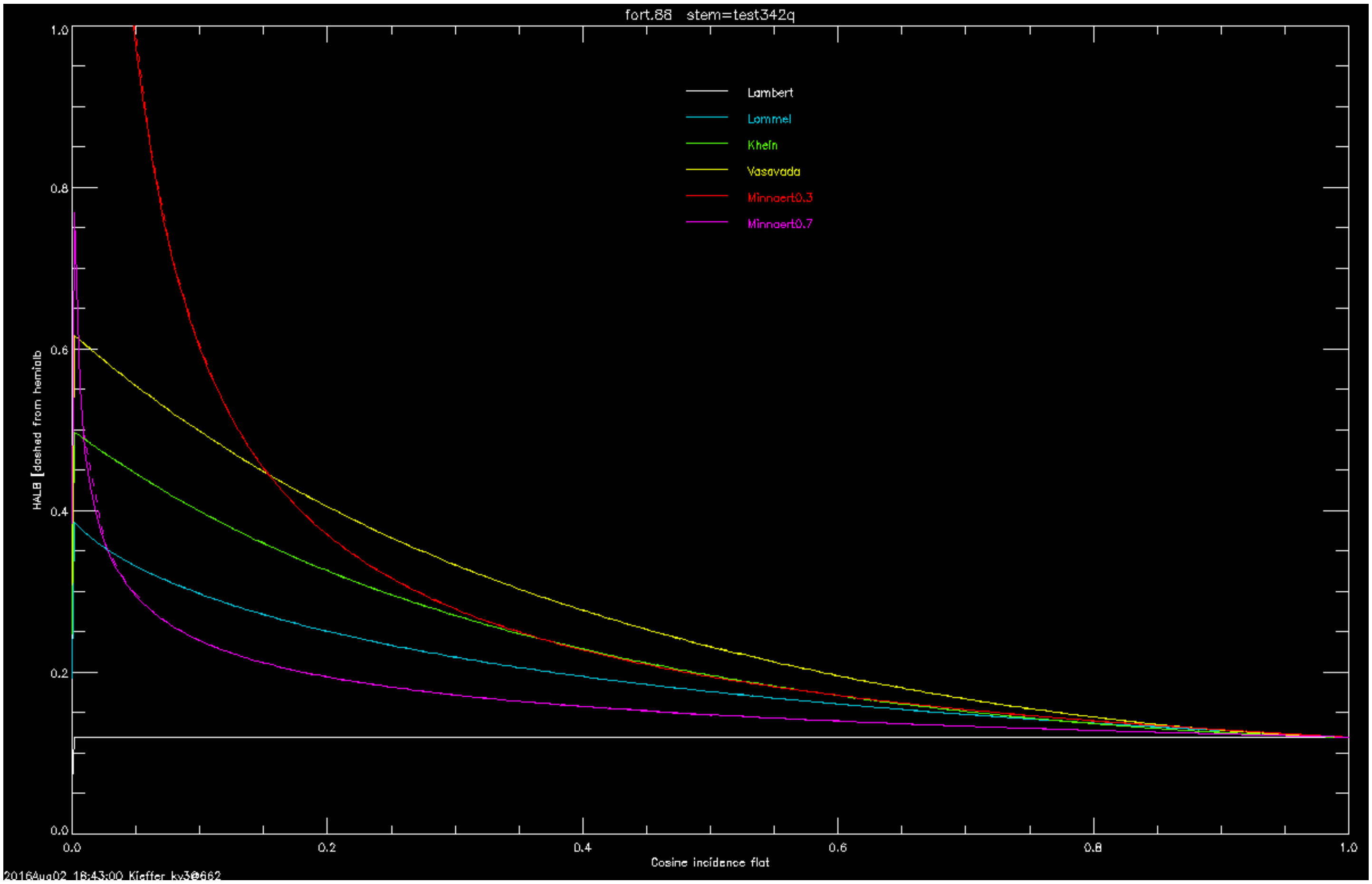

Figure 8: Hemispheric albedo computed in KRC versus cosine of the incidence angle, solid lines, at every time step. Dashed lines (largely invisible) are values at every degree computed in IDL (hemialb.pro @55). In both cases ALB=0.12 kv651.png

But projection onto a horizontal surface has an additional term $\cos i$ in the integrand.

$$\int_0^{2\pi} \int_0^{\theta} \sin x \cos x \; dx \; d\phi == 2\pi \left[\frac{\sin^2 x}{2} \right]_0^{\theta} = 2\pi \frac{\sin^2 \theta}{2}$$

.

Absorbed Direct insolation: C = Albedo * PhotoFunc * AtmTrans * SunAtMars

Albedo: constant for soil, may be variable for frosts; TDAY

PhotoFunc: cos i for Lambert, several others available; HALBF and BND2 in TLATS

AtmTrans: compute with DEDING2 for atmospheres, otherwise 1. [or 0 at night]; COLL in TLATS

SunAtMars: $1/AU^2$; TSEAS

Trying to put all the incidence angle calculations into one DownVIs in TLATS may be asking too much.

```
ASOL[jj]
     &, ASOL(MAXN2)     ! Insolation at each time of day, direct + diffuse
     &, ADGR(MAXN2)     ! Atm. solar heating at each time of day
Both use LFROST at the start of the season and do not treat a change during a season.
```

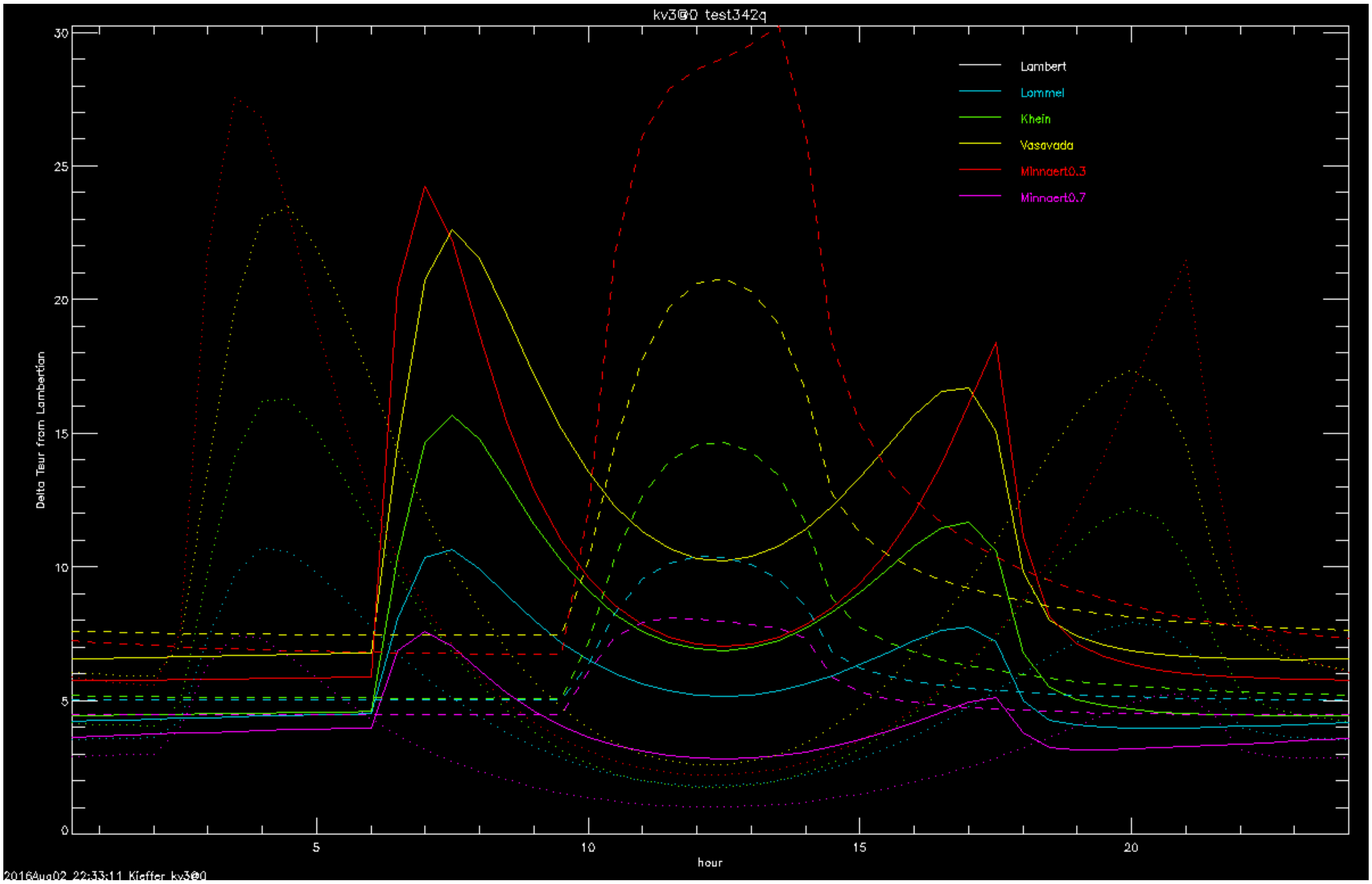

Figure 9: Effect of various photometric functions; low thermal inertia, I=50, and at 60°S for Mars orbit but no atmosphere. Ordinate is Tsur relative to the values for a Lambertian model. Solid lines are Ls=0, dashes Ls=93, dotted Ls=272. kv572.png

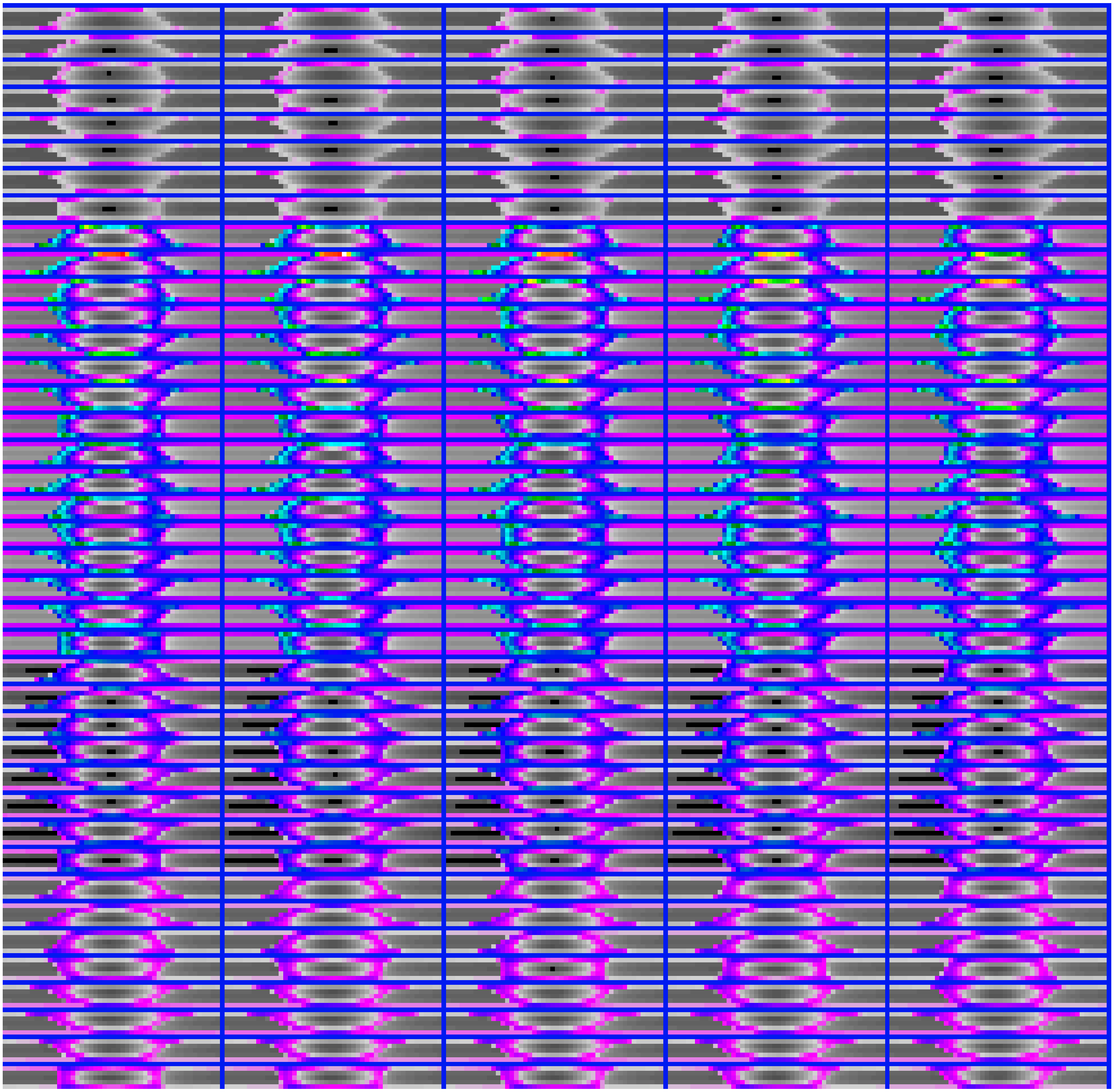

Figure 10: Effect of various photometric functions; low thermal inertia, I=50, and at the equator for Mars orbit but no atmosphere. QUILT3 image of delta temperature from a Lambertian model. Displayed value range is: 0.0 to 36.5 K. Sample is: hour(48) * 5 planes of seas*case. Line is: lat(5) * 40 groups of seas*case; Lines increase upward.. Latitudes: -60 -30 0 30 60 . Season range: 0.1 to 351.4 . From the bottom upward, models are: Lommel-Seeliger, Kheim, Vasavada, Minnaert 0.3, Minnaert 0.7 . quilt.png

14 Heat flow Details

14.0.1 Lower boundary

To stay in the numerical scheme of KRC, impose upward basal heat flow of H_g by setting

$$T_{n+1} = T_n + \frac{H_g(B_n + B_{n+1})}{2k_n} \qquad \text{eq : H1} \tag{17}$$

This value is constant for all latitudes and seasons of a single case.

14.0.2 Upper boundary

No change to numerical iteration, but starting conditions can include expected effect.

Modified from KRC paper, equation number in [13] there has additional term on the right, $+H_g$.

Surface radiation balance, from Eq.[13] for a flat surface with sub-surface heat flow H_g; revised [12] :

$$\epsilon\sigma\langle T_s^4\rangle = (1. - A)\langle S'_{(t)}\rangle + H_g + \epsilon\sigma\beta_e\langle T_a^4\rangle \qquad \text{eq : Hs} \tag{18}$$

This will be further revised with the inclusion of photometric functions, see §15.6.2 . Expansion of $\langle H_R\rangle$ using Eq. [5] and a combination of Eqs. [10] and [13], yields revision of [12];

$$\langle T_a^4\rangle = \frac{\langle H_V\rangle/\beta_e + (1 - A)\langle S'_{(t)}\rangle + H_g}{\sigma(2 - \epsilon\beta_e)} \qquad \text{eq : Ha} \tag{19}$$

14.0.3 General relations

Heat flow at any time between any two layers, Layer i of thickness B_i, conductivity k_i and temperatureT_i and the next deeper layer with values B_+, k_+ and T_+. Assume the temperature holds at the middle of each layer. Equating the heat flow in the adjacent 1/2 layers so that no energy goes into the interface at temperature T_p:

$$H_i = \frac{k_i}{B_i/2}(T_p - T_i) = \frac{k_+}{B_+/2}(T_+ - T_p) = H_+ \qquad \text{eq : H4} \tag{20}$$

yields;

$$T_p = \frac{k_iB_+T_i \ + \ k_+B_iT_+}{k_iB_+ + k_+B_i} \qquad \text{or} \qquad T_p = \frac{T_i + fT_+}{1 + f} \qquad \text{where} \qquad f = \frac{k_+B_i}{k_iB_+} \qquad \text{eq : H5} \tag{21}$$

Use either of the first relation to get heat-flow; the first yields

$$H_i = \frac{k_i}{B_i}\frac{2f}{1 + f}(T_+ - T_i) \qquad \text{eq : H6} \tag{22}$$

However, the values will be near the geothermal level only well below the annual skin-depth.

14.0.4 General expectation

Compared to a model without heat flow, for which the diurnal mean surface temperature is $\epsilon\sigma T_m^4 = S_o f(1 - A)/U^2$ where f is a "garbage" factor that accounts for the effects of illumination geometry and the atmosphere; with steady geothermal heat-flow H_g would expect a diurnal mean surface temperature $\epsilon\sigma T_H^4 = fS_o(1 - A)/U^2 + H_g$, some manipulation leads to a change in the average surface temperature:

$$T_H - T_m \simeq H_g/\left(4\epsilon\sigma T_m^3\right) \tag{23}$$

The effect will be larger at night and less in the day. With depth, the average temperature increase for layer j is,

$$\overline{T_{Hj}} - \overline{T_j} \approx +\frac{H_g}{k}\left[\sum_{i=2}^{j-1} B_i \ + B_j/2\right] \tag{24}$$

where B_i is the thickness of KRC layer i in meters.

14.0.5 Slow convergence

KRC starts with an isothermal profile. Heat-flow induces a thermal slope which grows from the lower boundary.

Characteristic time for a slab of thickness l is l^2/κ

Carslaw59=[3] [CJ] section 3.4 discusses a region $-l < x < l$ with zero initial temperature and with the $x = \pm l$ kept at constant temperature V for $T > 0$". Apart from the imposition of a linear gradient, this region $0 < x < l$ is similar to starting KRC isothermal and imposing a constant heat-flow that in infinite time would increase the lower boundary by V. Introducing the dimensionless parameters $T = \kappa t/l^2$ and $\xi = x/l$ (all this is CJ symbols) yields solution for the temperature increase $v_{(T,\xi)}$ in the form

$$\frac{v}{V} = 1 - \frac{4}{\pi}\sum_{n=0}^{\infty} \underbrace{\overbrace{\frac{(-1)^n}{2n+1}}^{a} \overbrace{e^{-(2n+1)^2\pi^2/4\cdot T}}^{b}}_{p1} \underbrace{\cos \overbrace{\frac{(2n+1)\pi}{2}}^{c} \xi}_{p2} \qquad [\qquad 3.4.4] \tag{25}$$

and the family of solutions is shown in CJ Fig 11 (p. 101), which I have recalculated [IDL routine qkrcsimp.pro] and show in Fig 11; the under- and over-braces indicate grouped terms in my numerical solution.

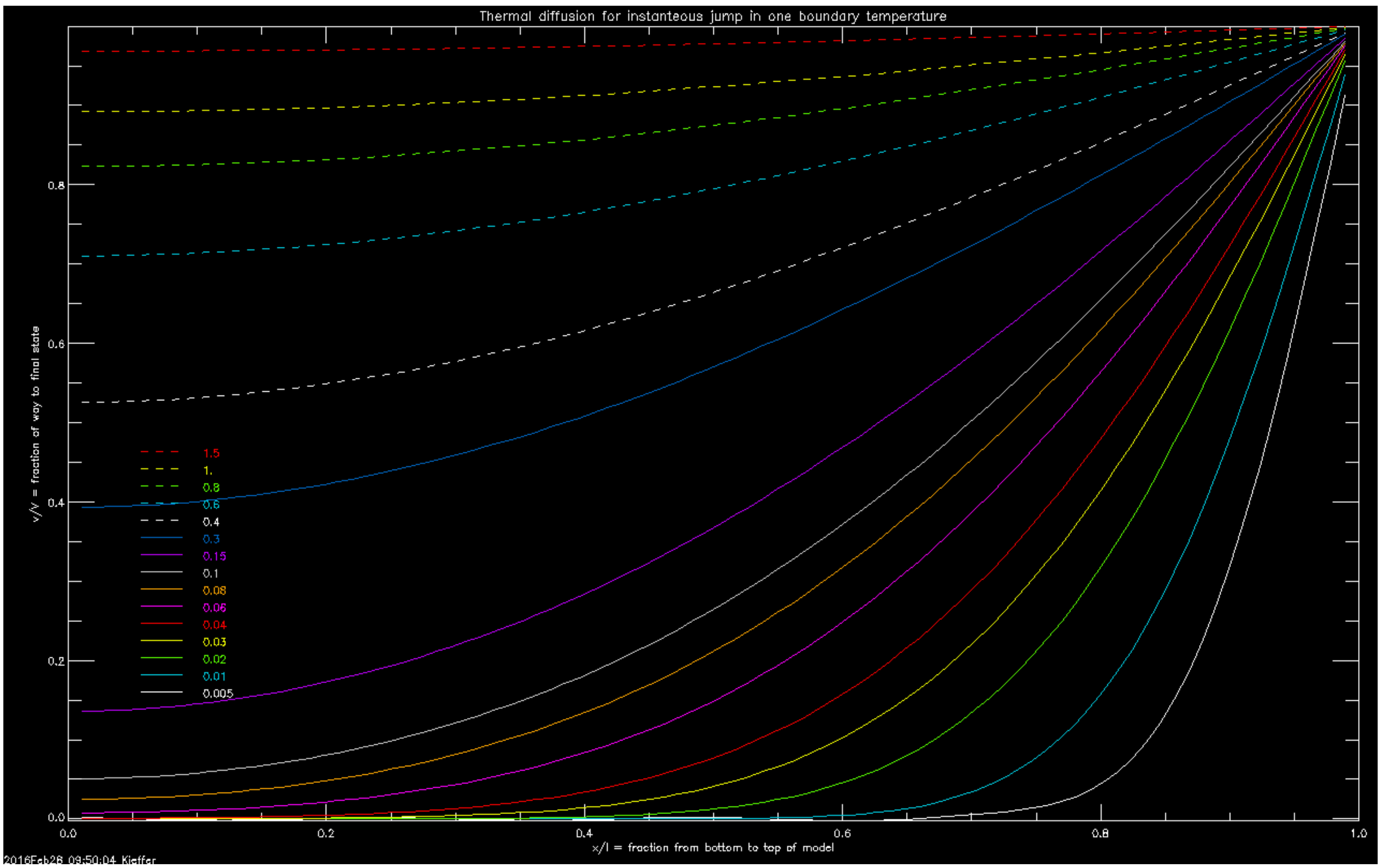

Figure 11: Temperature distribution over time for a slab with a jump in one boundary.Ordinant is fraction of the temperature rise to final state; abscissa is fraction of the way into the slab. Curve values shown in the legend are normalized time: $\kappa t/l^2$. See text. CJFig11.png

If a KRC model is thick enough to attenuate the annual cycle (about 4 times the annual skin depth, $l = 4D_y$ and $D_y = \sqrt{\kappa \frac{P_y}{\pi}}$. Then, to reach 90% effect, need $T = 1$ or $t = l^2/\kappa \Rightarrow t = 16P_y/\pi$, or about 5 years.

A relatively quick way to approximate heat-flow models is to compute a KRC model with no heat-flow, estimate the change in surface temperature and the thermal gradient on the model soil stack using the thermal conductivity profile and add these to the KRC no-heat-flow model.

14.0.6 Issues ?? CHECK

1. Heat flow determined from KRC run output temperatures does approach the lower-boundary input values. However, when IC2 invokes a second material, get only about 80% of the expected heat flow. Have not found the cause of this.

2. Ts differences for the two T-dep cases are up to about 1K at the equator and 2K for $\pm 60°$.

   Convergence test for the lower material in Version 2 was against CONVG, so a large value generated thick layers in the lower material; inappropriate. V34 cannot reproduce the exact layer and time-doubling configuration of V23. Yet, the remaining difference are much larger than expected. See the working document *V33issues.tex*

15 Far-field Detailed design

Enhancement: Option for sloped models to use temperatures from a far-field model (first case, if it was run and stored) for the far ground.

Terminology:

self mode Radiation from the far field horizontal surface assumed to be at the same temperature at the sloped surface, and of the same albedo. The atmosphere temperature is calculated based on sloped-surface temperature. This was KRC's only mode until version 3.4

far mode Radiation from the far field surface is based on prior KRC run. Atmosphere down-going radiation also based on this prior run.

far-field file == fff A file that contains the temperatures from a model of the "remote" surface and atmosphere as needed by a sloped case. A fff contains the values computed on the last day, not extrapolated to the end of a season

"flat" case A zero-slope case that matchs a sloped case in all parameters except slope and azimuth.

Sequence in the software set:

- TCARD reads the file name.
- LOPN3=true indicates that a fff is available.
- KRC checks the name length, and can call TFAR(1 to open the file. If fff open fails, TFAR will write an error message and KRC will stop to prevent a possible long run with the wrong fff.
- TSEAS calls TFAR*(2) before the first season to get all the size and date information. For each run season, it interpolates the far-field file to that season; if the nearest fff season is within 1% of a season length, it will use that season only; else it calls TFAR twice to get the bounding seasons. Tsurf, and Tatm if needed, are interpolated to the desired date and stored in COMMON.
- TLATS extracts the right latitude, and will do cubic interpolation in hour to the N2 times if needed. TLATS converts fff surface temperature to radiance at each time-step and places in COMMON as FARAD. If KRC is using an atmosphere, TLATS interpolates fff Tatm temperature for each time-step and places in COMMON as HARTA.
- TADY uses LOPN3 as the flag to indicate the fff is being used. When using fff, the atmosphere temperatures and radiation come from fff. Frost is still calculated.

15.1 Far-field for sloped surfaces

Atmosphere and far-field to be based on a no-slope run with all other conditions normally the same.
No additional changes are made for pits.
User should not invoke global frost integration (KPREF=2) for sloped runs.

For onePoint mode; more complicated: **not yet implimented**
Must specify the name of an appropriate fff; must read it successfully
fff seasons must cover the range needed by the onePoint master
fff latitudes must cover the range in the onePoint lines
Print a stern warning that a single fff is being used.
For each onePoint line, need to interpolate:
in season; same as for normal run
in latitude; requires additional code
Limitation: only allow a single fff for a onePoint run.

15.2 New Inputs

Control is by the presence of the file name defined by change line : 8 3 x name

15.3 New Outputs

No plan for **TDISK** to be able to read type 52

Size of fff is set by firm-code sizes, TSF(MAXNH,MAXN4) for each season = 96*37*8=28416 bytes for each temperature for each season. The first record in file contains KRCCOM; season start at record 2. Thus, the file size in bytes is 28416*(seasons+1)*[1,2 or 3]

15.4 New Common Items

Must set the maximum number of time-steps that can be stored in a fff. To simplify writing records in TDISK, write the existing arrays, TSF etc., these are (MAXNH,MAXN4). Because direct-access files are not open to write for more than one case at a time, it would be possible to use smaller record sizes and write

((TSF(I,J),J=1,NLAT),I=1,N24) as long as record can hold KRCCOM.

FORTRAN executes the outer loop first

However, tests indicate that must write complete arrays to direct-access records

Must also set the maximum number of time-steps that can be interpolated from an fff temperature set. This couild be smaller than MAXN2=384*4*256=393216

.

FILCOM/ CHARACTER*80 FFAR ! far-field temperatures input

HATCOM/ INTEGER MAXFF PARAMETER (MAXFF=384*4*4=6144) ! dimension of far-field times of day REAL*8 SALB ! spherical albedo of the soil

REAL*8 FARTS(MAXNH,MAXN4,2) ! far-field Tsurf/Tatm for current season

“ FARAD(MAXFF) ! far-field radiance for every time-step at current latitude

“ HARTA(MAXFF) ! far-field Tatm for every time-step at current latitude

“ SOLDIF(MAXN2) ! Solar diffuse (with bounce) insolation at each time W/m^2

UNITS/ INTEGER MINT ! the number of temperature sets in file being handled by IOD3

Type -n files as 1,2 or 3 arrays each Real*8 (MAXNH,MAXN4)

15.4.1 Array size limits

TFAR8: FTS ,FTP,FTA (MAXNH,MAXN4) for one season

TSEAS: same 3

FARTS(MAXNH,MAXN4,2) ! far-field Tsurf/Tatm for current season

TLATS: REAL*8 WORK(MAXFP=MAXNH+3) ! to hold extended hours CUBUTERP8 outputs into FARAD=KIM*NY= must be less than MAXFF

KRC runs requires N2 le MAXN2 =384*4*256=393216; runs using fff will limit N2 to MAXFF=384*4*4 = 6144

15.5 Implementation

ALERT The use of the output file flag K4OUT has been changed for version 3.4 . It now controls only the direct access type being written. Actual reading and writing of data file is controlled by the presence and length of three file names; all 3 names default to 'no'. Sees §15.6.8.

15.6 What arrays are available in earlier versions

In **TLATS** [or **TDAY**]:

for the last day computed, not extrapolated

TSF(I,J4)=TSFH(I) (hour,lat)

TPF(I,J4)=TPFH(I) (hour,lat)

TAF(IH,J4)=TATMJ (hour,lat) saved in **TDAY**

Extrapolated to the end of a season

TTS4(J4)=xof TTS , TTB4(J4)=xof TTB : surface and bottom diurnal average

TTA4(J4), midnight Atm

FROST4(J4)=EFP ! frost amount

TMN4(I,J4) , predicted TT1(layer,lat) temperature at midnight

15.6.1 Self-heating versus Far-field

Minimize code changes; aim at needing only Tsurf and Tatm from the far-field model. Assume normally will use same atmosphere parameters, although could use different!

from KRC paper: [68]...
The surface condition for a frost-free level surface is :

$$W = (1. - A)S'_{(t)} + \Omega\epsilon R_\Downarrow + k\frac{\partial T}{\partial z}_{(z=0)} - \overbrace{\Omega\epsilon\sigma}^{FAC5} T^4 \quad (jgr13) \tag{26}$$

where W is the heat flow into the surface, A is the current surface albedo, $S'_{(t)}$ is the total solar radiation onto the surface as in Eq. (1), $R_\Downarrow$ is the down-welling thermal radiation (assumed isotropic), T is the kinetic temperature of the surface, k is the thermal conductivity of the top layer. Ω is the visible fraction of the sky, ϵ is the surface emissivity and σ the Stefan-Boltzmann constant. In the absence of frost, the boundary condition is satisfied when $W = 0$.

from KRC paper: [70]...
The collimated incident beam is treated rigorously, intensities of the diffuse solar and thermal fields are modified by the fraction of sky visible, and the average reflectance and emittance of the surrounding surface (absent in the level case) are approximated as: the brightness of level terrain with the same albedo, and material having the same temperature as the target surface, respectively; this last approximation accentuates the diurnal surface temperature variation with increasing slope. Then

$$S'_{(t)} = S_M \left[\underbrace{F_\parallel \cos i_2}_{direct} + \underbrace{\Omega F^\downarrow_\ominus}_{diffuse} + \underbrace{\alpha A(G_1 \cos i F_\parallel + \Omega F^\downarrow_\ominus)}_{bounce} \right] \quad (jgr14) \tag{27}$$

$F_\parallel$ =`COLL` is the collimated beam in the Delta-Eddington model and $F^\downarrow_\ominus$=`BOTDOWN` is the down-going diffuse beam. Ω=`SKYFAC`: $\Omega \equiv 1-\alpha$ here and in Eq. (13). G_1=`G1` is the fraction of the visible surrounding surface which is illuminated. Within the brackets in Eq. (jgr 14),

the first term is the direct collimated beam, `DIRECT`

the second is the diffuse skylight directly onto the target surface, `DIFFUSE`

$F^\downarrow_\ominus$) does not depended upon slope.

the third term is light that has scattered once off the surrounding surface, `BOUNCE`

For a sloped surface, G_1 is taken as unity. As a first approximation, for depressions $G_1 = (90 - i)/s \ < 1)$ where s is the slope to the lip of the depression (the apparent horizon). For the flat-bottom of a depression, $i_2 = i_0$ when the sun is above this slope, and $\cos i_2 = 0$ when below.

ALERT: $\cos i$ factor in the bounce $F_\parallel$ term is missing in JGR paper. It is in the tlats8.f code back to at least 2011aug.

For far-field, need to expands the thermal radiation balance term, and (jgr 13) becomes

$$\underbrace{W}_{POWER} = \underbrace{\overbrace{(1. - A)}^{FAC3} S'_{(t)} + \overbrace{\Omega\epsilon}^{FAC6} R^0_\Downarrow}_{ABRAD} + \underbrace{k\frac{\partial T}{\partial z}_{(z=0)}}_{SHEATF} - \overbrace{\epsilon\sigma}^{FAC5} T^4 + \underbrace{\overbrace{(1-\Omega)\epsilon\sigma\epsilon_x}^{FAC5X} T_x^4}_{FARAD} \tag{28}$$

where $R^0_\Downarrow$ is for the equivalent no-slope case and T_x is the far-field surface temperature and ϵ_x its emissivity; $T_x \equiv T$ if self-heating. The next-to-last term is surface emission into a hemisphere and the last term is thermal radiation from the far surface. If the sloped surface has frost, T becomes fixed at the frost temperature but the equation remains the same.

With the ability of albedo to depend upon incidence angle, need to expand $S'_{(t)}$ and (jgr 13) becomes

$$\begin{aligned} \underbrace{W}_{power} = S_M \Bigg[(1. - A_{h(i_2)}) \underbrace{F_\parallel \cos i_2}_{direct} + (1 - A_s) \Bigg(\underbrace{\Omega F^\downarrow_\ominus}_{diffuse} + \underbrace{\alpha A_s(G_1 \cos i F_\parallel + \Omega F^\downarrow_\ominus)}_{bounce} \Bigg) \Bigg] \\ + \underbrace{\Omega\epsilon R^0_\Downarrow}_{atm\ IR} + \underbrace{k\frac{\partial T}{\partial z}_{(z=0)}}_{conduction} - \underbrace{\epsilon\sigma T^4}_{emission} + \underbrace{(1-\Omega)\epsilon\sigma\epsilon_x T_x^4}_{back\ radiation} \end{aligned} \tag{29}$$

where all the terms within the square brackets are normalized (are unitless).

Reformulate, under- and overbrace terms indicate FORTRAN variable names

$$\begin{aligned} \underbrace{W}_{POWER} = \underbrace{(1. - \overbrace{A_{h(i_2)}}^{ALBJ})}_{FAC3} \overbrace{S_M F_\parallel \cos i_2}^{ASOL} + \underbrace{(1 - \overbrace{A_s}^{SALB})}_{FAC3S} \underbrace{S_M \Bigg(\overbrace{\Omega F^\downarrow_\ominus}^{DIFFUSE} + \overbrace{\alpha A_s(G_1 \cos i F_\parallel + \Omega F^\downarrow_\ominus)}^{BOUNCE} \Bigg)}_{SOLDIF} \\ + \underbrace{\Omega\epsilon}_{FAC6} \underbrace{R^0_\Downarrow}_{ATMRAD} + \underbrace{k\frac{\partial T}{\partial z}_{(z=0)}}_{SHEATF} - \underbrace{\epsilon\sigma}_{FAC5} T^4 + \overbrace{\underbrace{(1-\Omega)\epsilon\sigma\epsilon_x}_{FAC5X} T_x^4}^{FARAD} \qquad \mathrm{eq:wb} \end{aligned} \tag{30}$$

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$\equiv A_h$ =ALB*AHF is composed of two factors, ALB$\equiv A_0$ 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 $\texttt{FARAD}_t$ transfered to `TDAY`. However, to then accomodate variable frost emission, need to multiply by ϵ_f/ϵ for the frost case (relatively rare).

Because frost temperature changes only with pressure, it does not need to change with Hour.

15.6.2 Equilibrium temperature

The equilibrium temperature T_e is that value of T that would make the diurnal average of W in Eq. 30 zero. Or:

$$FAC5 * T_e^4 = \overbrace{\langle FAC3 * ASOL + FAC2S * SOLDIF\rangle}^{\Delta AVEI} + \ FAC6 * \langle ATMRAD\rangle + H_g + \langle FARAD\rangle \qquad \text{eq : Te} \tag{31}$$

where $\langle\ \rangle$ represents the diurnal average.

To reach the equivalent of JRG Eq. (12), need to modify JGR Eq. (11) by allowing angle[time]-variable albedo A and adding the geothermal heat-flow term H_g to become

$$\epsilon\sigma\langle T_s^4\rangle = \langle(1. - A)S'_{(t)}\rangle + H_g + \epsilon\sigma\beta_e\langle T_a^4\rangle \qquad \text{eq : sbal} \tag{32}$$

JRG Eq. (12) then becomes:

$$\langle T_a^4\rangle = \frac{\overbrace{\langle H_V\rangle/\beta_e}^{QS} + \overbrace{\langle(1-A)S'_{(t)}\rangle}^{AVEI} + \overbrace{H_g}^{GHF}}{\sigma(2-\epsilon\beta_e)} \qquad \text{eq : Ta4} \tag{33}$$

Then ATMRAD (= FAC9*TATMJ**4) $= \sigma\beta_e\langle T_a^4\rangle$

JGR eq. (2) remains the same:

$$\overbrace{H_V}^{HUV} = \overbrace{S_M}^{SOLR} \overbrace{\left(\mu_0 - F_\ominus^\uparrow(0) - (1-A_h(t))\left[\mu_0\ F_\parallel + F_\ominus^\downarrow(\tau_v)\right]\right)}^{ATMHEAT} \qquad \text{eq : aheat} \tag{34}$$

Atmosphere IR heating is the average of H_R in JGR Eq. (5): $\langle H_R\rangle = \sigma\beta_e\left(\epsilon\langle T_s^4\rangle - 2\langle T_a^4\rangle\right)$

15.6.3 code in TLATS

Snippits of code in TLATS for radiation values placed in COMMON; omitting all the logical tests. Minor edits for clarity.

```
.
  SOLR=SOLCON/(DAU*DAU) ! solar flux at this heliocentric range
  call DEDING28 (omega,g0,avea,COSI,opacity, bond,COLL,deri)
  DIRECT=COS2*COLL ! slope is in sunlight or =0
ASOL(JJ)=QI=DIRECT*SOLR ! collimated solar onto slope surface
.
  AHF= (1.D0+COS2*DLOG(COS2/(1.D0+COS2)))/2. ! Lommel-Seeliger, e.g.
  HALB=ALB*AHF/AH0 ! normalized hemispherical albedo
ALBJ(JJ)=MIN(MAX(HALB,0.D0),1.D0) ! current hemispheric albedo
.
SKYFAC = (1.D0+ DCOS(SLOPE/RADC))/2.D0 ! effective isotropic radiation.
     call DEDING28 (omega,g0,avea,COS3,opacity, bond,COL3,deri)
     BOTDOWN=PIVAL*(DERI(1,2)+F23*DERI(2,2)) or 0 ! diffuse down at surf
  DIFFUSE=SKYFAC*BOTDOWN ! diffuse flux onto surface
     PUS=1.3333333 ! e.g. Lommel-Seeliger P_S
     SALB=PUS*ALB ! spherical albedo, for diffuse irradiance
     G1=DMIN1 (1.D0,(90.D0-AINC)/SLOPE) ! (90-i)/slope or 1.
     DIRFLAT=COSI*COLL ! or COSI if no atm. collimated onto regional flat plane
  BOUNCE=(1.D0-SKYFAC)*SALB*(G1*DIRFLAT+DIFFUSE)
  QI=DIRECT*SOLR ! collimated solar onto slope surface
SOLDIF(JJ)=(DIFFUSE+BOUNCE)*SOLR ! all diffuse, = but the direct.
AVEI=AVEI+(1.d0-ALBJ(JJ))*QI+(1.-SALB)*SOLDIF(JJ) ! .
  ATMHEAT=COSI-TOPUP-(1.-AVEA)*(BOTDOWN+COSI*COLL) ! atm. heating
ADGR(JJ)=QA=ATMHEAT*SOLR ! solar flux available for heating of atm.
.
  in TFAR, extract from fff: FELP(8)= F3FD(2) ! surface emissivity
  FAC5X=(1.-SKYFAC)*EMIS*SIGSB*DELP(8) ! last is fff surface emissivity
  Extract fff surface temperatures into WORK for the proper latitude
     add midnight wrap to both ends.
     raise to 4'th power and multiply by FAC5X
CALL CUBUTERP8 (2,WORK,NHF,TENS,N2,FARAD) ! cubic interpolation to timesteps
```

15.6.4 Code in TDAY

Snippits of code in TDAY for calculating surface temperature; omitting all the logical tests and the convergence loops. Minor edits for clarity. When there is no atmosphere:

.

FAC3S = 1.D0-SALB ! spherical absorption
FAC3 = 1.D0-ALBJ(JJ) ! hemispherical absorption
ABRAD = FAC3*ASOL(JJ)+FAC3S*SOLDIF(JJ) ! surface absorbed radiation
FAC5 = SKYFAC*EMIS*SIGSB ! if self-heating
FAC5 = EMIS*SIGSB ! if fff
FAC7 = KTT(2)/XCEN(2) ! current redone if T-dep conductivity
TS3 = TSUR**3 ! bare ground
SHEATF = FAC7*(TTJ(2)-TSUR) ! upward heat flow to surface
POWER = ABRAD +SHEATF - FAC5*TSUR*TS3 ! unbalanced flux
POWER = POWER+FARAD(JJ) ! only if fff

ERROR: of 10 deg slope, fff 30 K hotter than self. Not physical
And hour 23 and 23.5 are much colder.
daytime rise with fff= flat:NoA greater than with fff= flat:tinyA
Suggests: far view factor much too big

15.6.5 find T for W=0

Need to modify JGR Eq. 29 : " From Eq. (13), find

$$\frac{\partial W}{\partial T} = \overbrace{-k/X_2}^{FAC7} - \overbrace{4\Omega\epsilon\sigma}^{FAC45} T^3 \quad (jgr29) \tag{35}$$

where X_2 is the depth to the center of the first soil layer. " becomes

$$\frac{\partial W}{\partial T} = -\overbrace{k/X_2}^{FAC7} - \overbrace{4\epsilon\sigma}^{FAC45} T^3 \qquad \text{coded as} \qquad \overbrace{\Delta T}^{DELT} = \frac{W}{k/X_2 + 4\epsilon\sigma T^3} \tag{36}$$

i.e., Ω becomes 1, so omit this from `FAC45`

Thus, the fff must contain at least $R_{\Downarrow}$ [or Tatm] and T_x. Probably desireable that it contain KRCCOM for insurance.

To handle many seasons, could write direct access files.

–§ **Interpolation of the fff**

Generally expect that the far-field case will be run with exactly the same grid in season, latitude and time as a sloped case. Because the stored hours are less dense than time-steps, will need interpolation in at least that dimension.

Accomodate linear interpolation in season, but if the seasons are within DELSEAS=1% of a season-step, use that season without interpolation.

To avoid many complications in interpolation, require that fff contain a latitude within DLATEST=0.1 degree of those in the sloped case, and use that without interpolation.

Will interpolate smoothly (cubic spline, replicated across midnight) in hour from the stored N24 points to the N2 time-steps. Firm-code the maximum number of times steps for using a fff to a generous MAXFF=384*4*4=6144.

However, this requires that N2 for the slope run be an integral multiple, 2 or more, of N24 for the fff run.

15.6.6 Getting Tatm from TPlan and Tsurf. Not used.

in **TLATS**: tauir is in krcc8m.f;
It does vary with elevation (due to PRES) , which can be a function of latitude
it can vary with season if PZREF varies due to KPREF ne 0

```
in TLATS:
        TAUIR=(CABR+TAUVIS*TAURAT)*(PRES/PTOTAL)+TAUICE ! thermal opacity, zenith
        QA=AMIN1(0.0168455D0,AMAX1(TAUIR,62.4353D0)) ! limits 1. < FACTOR < 2.
        FACTOR= 1.50307D0 -0.121687D0*DLOG(QA) ! from fit to hemisphere integrals
        TAUEFF=FACTOR*TAUIR     ! effective hemispheric opacity
        BETA=1.-DEXP(-TAUEFF)   ! hemispheric thermal absorption of atmosphere


in TDAY:  IF (LATM) ....
      EMTIR = DEXP(-TAUIR)       ! Zenith Infrared transmission of atm
```

```
      FAC82=1.-EMTIR            !  " absorption "
      FAC9=SIGSB*BETA           ! factor for downwelling hemispheric flux
SIGSB is a constant and BETA
      IF (EFROST.GT.0.) THEN
        FAC8=EMTIR*FEMIS        ! ground effective emissivity through atmosphere
      ELSE                      ! bare ground
        FAC8=EMTIR*EMIS
      ENDIF

            ATMRAD= FAC9*TATMJ**4 ! hemispheric downwelling  IR flux
              TPFH(IH)=(FAC8*TSUR4+FAC82*TATM4)**0.25 ! planetary
              TAF(IH,J4)=TATMJ  ! save Atm Temp.
              DOWNIR(IH,J4)=ATMRAD ! save downward IR flux
```

This is a mess, as seasonal PRES is not in type -1 files. and EFROST is only in the single KRCCOM in the first record.

Alternate solution is to store Tatm in traditional type -1 files.

To get the right Tplan for output for sloped surfaces, need to have Tatm for the flat case, so would have to include that also in the type -1 files.

Could avoid worsening the large size of type -1 files by making them R*4, convert to/from R*8 in **TDISK**

Not much more work to define a new type similar to -1 but which includes TATM. This could be read/write simultaneously with type 52.

Could store the any new flags and arrays in hatcom to avoid impacting any other commons. hatcom is already included in **TLATS** and **TDAY**.

Better solution might be to put only DOWNIR and TATM is a separate type -2 file, and perhaps be able to write -1,-2 and 52 all at the same time?

This would take additional input parameters to set up, or could set the required logical flags (in **TCARD**) based on setting the file names

15.6.7 File handling in version 3.4

To utilize far-field temperatures, need to have an input data file open simultaneous with at least one output file. Version 3.4 can handle 0 to 3 data files open at once. To deal efficiently with fff for both with and without atmophere cases, two additional types of binary files have been defined, and the prior type -1 has becomes type -2.

15.6.8 Handling 3 data files at once.

Because type 52 is written after all cases done, it should be possible to have direct-access file open at the same time with little conflict. Will need more complex control logic. Note: type - requires open/close for each case.

Need to have multiple file names active, and **TCARD** must distinguish when to open/close direct access (each case) versus bin5 files (may be multi-case)

May have zero to 3 data files active at one time, determined by the second field in a change card starting '8':

5 A "bin" Type 5x (52) bin5-format to be written. Name is `FDISK`. PIO (c-level I/O) system determines the unit. LOPN4 is true when active. All interface is through **TDISK**, which calls **BINF5** to open or close.

21 A direct-access file to be written. Name is `FDIRA`. Uses IOD2, LOPN2 is true when active. Five types are available; they are distinguished by the value of K4OUT. By convention, the file extension should indicate the file type, but the KRC system makes no decisions based on this extension. All interface is through **TDISK**.

3 A fff direct access file to be read. Name is `FFAR`, it must be type -N; -1 is adequate for air-less bodies and -3 is required for atmospheres. Uses IOD3, LOPN3 is true when active. All interface is through **TFAR**, with open and close initiated by calls from **KRC**

BEWARE: seasons in this file must cover all that will be computed in later KRC run, including the spin-up. It is best to save a full year, with no wrap.

Each direct access file is opened after a new name of length four or more characters is read into FILCOM by **TCARD**
Each direct access file is closed from **TCARD** when a new name of any length is listed or from **KRC** when the run ends.
IOD2 is closed at the end of a case, as direct-access files as implimented by KRC can only hold one case.
An open file is always active!
e.g., a sloped case will use FFAR if that is open, else it will self-heat
Seasons-records are written or read from **TSEAS** by calls to **TDISK** or read by calls to **TFAR**. Sloped cases are required in abundance to address thermal beaming, which may be especially relevant for airless bodies. For these, need only the surface tempertures, so include the capability of writing direct access files with only Tsurf.

15.7 Lab notes on tests

Start with master34.inp, fewer latitudes and shorter spinup. Generate fff tm3 for flat and for fake steep self-heating slope.

Run case with epsilon slope to compare with flat, and a case with slope and azimuth identical to the fake steep case, expecting same temperatures. Output to /work/work1/krc/beam/BeamBa*

```
2016 May 25 19:07:35
Edit candi.inp for 321 and 341, run both
kv3@115, then parf[[5,6,0,1]]=['/work2/KRC/321/run/out/',    'candi' $
                              ,'/home/hkieffer/krc/tes/out/','candi341']

Doing -------------->     550
Num lat*seas*case with NDJ4 same/diff=         632         128


test341.inp: 40 seasons, 2 year spinup, 3 latitudes
   6 cases exercising zone table, photometric function, heatflow, constant KofT

----------  consistency between file types --------
test341a.inp: Double run: output in: /work/work1/krc/test/

1) 670 seasons, soly, no spinup, 5 latitudes
   6 cases exercising zone table, photometric function, heatflow, constant KofT
   54271232 May 23 23:11 v341aTest.t52
   57201408 May 23 23:11 v341aFlat.tm3

2) 40 seasons, 2 year spinup, 19 latitudes
 1 case, variable frost albedo and temperature
    1047488 May 23 23:11 v341aTest2.t52
    3495168 May 23 23:11 v341aFlat2.tm3

kv3@ 147 for both pairs of files confirms that t52 and tm3 temperatures
are identical.

--------- comparison of 341 to older versions ------

321//VerTest.inp
670 seasons soly, no spinup 5 latitudes
 6 cases test KofT with/without atmosphere



@2 pari 7=3 8=11 17=2

@45  Case 1-Case 3
Item in ttt Mean     Std    mean_ABS_std
    Tsurf  -0.417   2.814   0.480   2.804
    Tplan   1.633   4.516   3.654   3.115
     Tatm   0.000   0.548   0.450   0.313
  DownVIS   0.030   0.380   0.030   0.380
   DownIR   0.000   0.319   0.244   0.205

 dt=reform(t1[*,0,jlat,*])
clot,dt   shows that last hour are all near -.8, read within +-.2

@45  Case 2-Case 4
Item in ttt Mean     Std    mean_ABS_std
    Tsurf  -0.241   2.141   0.375   2.122
    Tplan   1.818   4.046   3.539   2.675
     Tatm   0.000   0.543   0.440   0.318
  DownVIS   0.020   0.363   0.020   0.363
```

```
   DownIR   0.000   0.318   0.240   0.209

2016 May 24 08:03:02
KRCINDIFF: test for changes. Input limits:       64     120     220
 11 10   ABRPHA      27.955     -0.0000      27.955
 22 21ARC3/Safe     0.80010     -0.0000     0.80010
 32 31     fd32      3182.5      0.0000      3182.5
 76 75    TATMJ      184.59      184.61   -0.015201
117 16    K4OUT      -3      52     -55
118 17    JBARE    9999       0    9999
134 33      KKK       4       8      -4


.rnew kv3
@114  4  232 342
@11 1=test342f
@111 123
@12  7=0 20=3  23=
@402 Clot DOWNVIS and Tsurf for all cases     WRONG? slope greater max
   WRONG? Tsurf far is 36K greater than self at noon
@45   stats on case deltas
@46   plot case delta for DOWNVIS and Tsurf

@12  0=34 23=-3
 stem='v342Flatf'   read the type -n for 1st case
@51    yields FOUT=dblArray[5, 2], TTOU= dblarr[48, 5, 3, 40]
t4=ttt[*,0:2,*,*,0] & t4=transpose(t4,[0,2,1,3])
HISTFAST,ttou-t4 ; all 0
  stem='v342Flata' & pari[23]=-1     read the type -n for 2nd case
@51    yields  TTOU= dblarr[48, 5, 1, 40]
t4=ttt[*,0,*,*,1] & t4=transpose(t4,[0,2,1,3])
HISTFAST,ttou-t4 ; all 0
```

Far-field heating has a smaller effect than self-heating; see Figure 12. An artifact of KRC is that atmospheric pressure can be constant even when frost forms so that the amount of frost is not limited; this limits night temperatures and can delay the dawn temperature rise, as seen in the first case in the Figure.

15.7.1 fff types

All lunar-like with Mars orbit, 5 lats, 40 seasons. Cases:
1: tiny atmosphere, save /work1/krc/test/v342Flatf.tm3
2: no atm, save v342Flata.tm1
3: 10 degree slope, save v342Flatb.tm1
4: same slope, use fff, save v342Flatc.tm1

15.7.2 temporary

tday 686 TAF(IH,J4)=TATMJ ! save Atm Temp. Done only on the saved hour, so would need to either use as is or interpolate to each time step. C0uld do the interpolation in **TLATS** time-step loop

15.7.3 Reminder, FORTRAN file types

http://www.fortran.com/fortran/F77_std/rjcnf.html is the F77 definition OPEN arguments that control the nature of the file.
ACCESS:
'SEQUENTIAL'=default
'APPEND'
'DIRECT'
RECL must also be given, since all I/O transfers are done in multiples of fixed-size records.
UNFORMATTED is the default
FORM:
'FORMATTED'=default for sequential. Each record is terminated with a newline character; that is, each record actually

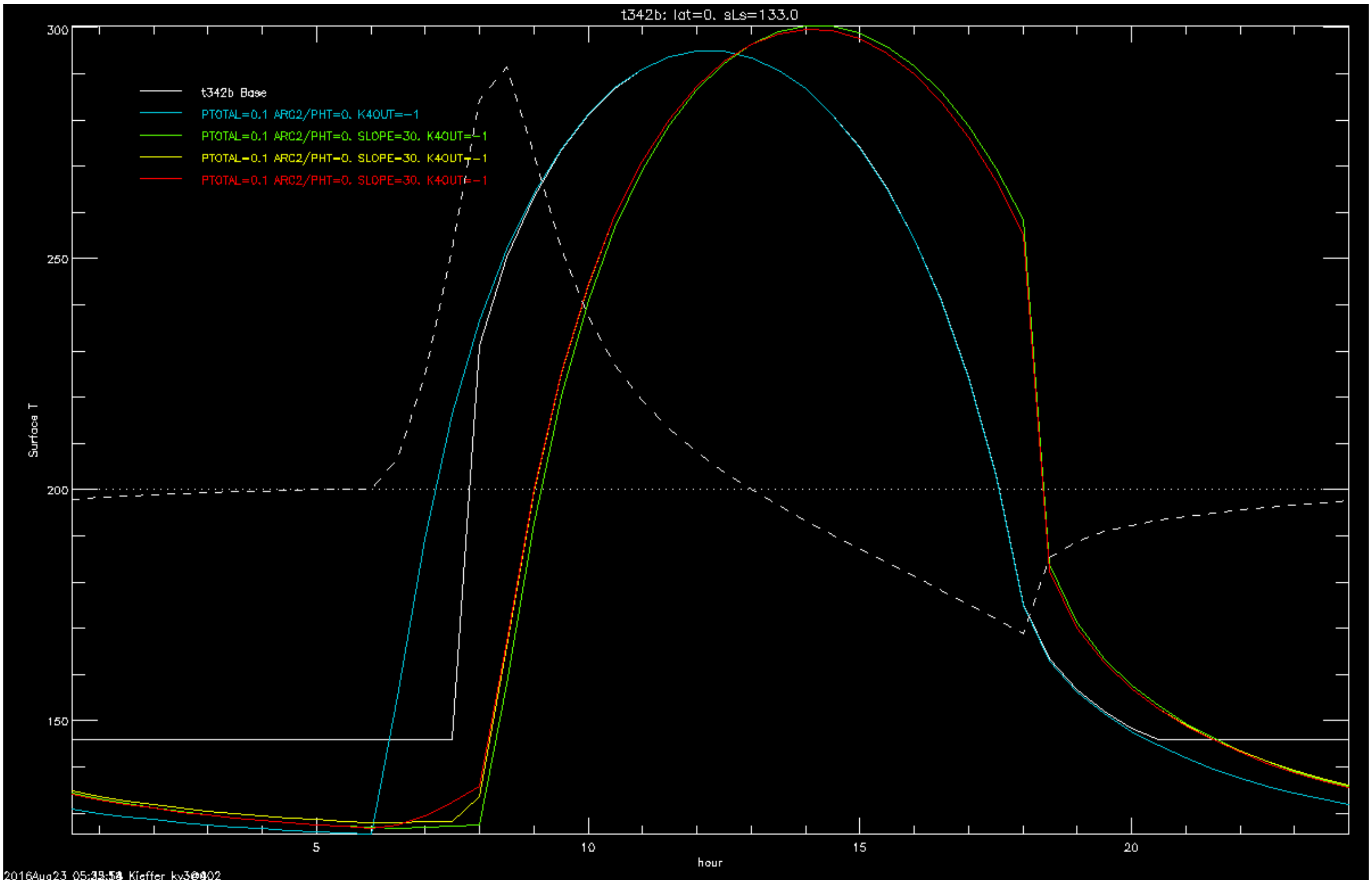

Figure 12: Test with slope of 30° dipping to the West for a nominal asteroid surface in Mars orbit; diurnal surface kinetic temperatures at the equator at $L_S = 134$. The 5 cases in the legend are: top=white: Flat with surface pressure (P_T) 1.1 Pa and clear with frost temperature 146; 2=blue: flat with no atmosphere; 3=green: sloped with no atmosphere and self-heating; 4=yellow: sloped with no atmosphere and far-field from first case, which is artificial; bottom=red: sloped with no atmosphere and far-field from 2nd case, which is realistic. The dashed line shows the difference self-heating minus far-flat-heating magnified by a factor of ten and offset from 200; it reaches 9K near dawn. tslope.png

has one extra character.

'UNFORMATTED' the size of each transfer depends upon the data transferred.

Each record is preceded and terminated with an INTEGER*4 count, making each record 8 characters longer than normal. This convention is not shared with other languages, so it is useful only for communicating between FORTRAN programs.

'PRINT'

RECL=rl: required if ACCESS='DIRECT' and ignored otherwise.

rl is an integer expression for the length in characters of each record of a file. rl must be positive.

If -xl[d] is set, rl is number of words, and record length is rl*4.

else, rl is number of characters [bytes], and record length is rl.

-xl does not occur in the KRC Makefile, as of 2016may11. Excerpt from **TDISK**:

```
  IF (K4OUT.LT.0) THEN  !  K4OUT is negative  .tm1
     NWTOT=2*MAXNH*MAXN4
  NRECL=8*NWTOT    ! bytes: or  NRECL=NWTOT  ! depends upon compiler <<<<
  OPEN (UNIT=IOD2,FILE=FDISK,ACCESS='DIRECT',STATUS=CSTAT,RECL=NRECL,...
```

15.7.4 Early plan and timing tests

Although dicussed in this early email, changing files to Real*4 has not been impliments in V3.4.2

```
Robin:

With regard to the need for additional stored information to enable use of a
 flat far field (fff) for sloped surfaces in KRC.
```

```
I have reached a compromise solution that optionally adds atmospheric
 temperature (Ta) as a third array after Surface kinetic temperature (Ts) and
 top-of-atmosphere nadir brightness temperature (Tp, planetary temperature) in
 the type -1 file. This is called type -3; the file extension would be .tm3

An added advantage of type -3 is that it allows relatively easy calculation
 (estimation) of the brightness temperature for off-nadir viewing: e.g.,
B=(emis*Ts^4 - Tp^4)/( Ta^4-emis*Ts^4)  B is the transmission of the atmosphere
tau = -ln(B)                        is the atmosphere column opacity
C=exp(-tau/cos e)                   where e is the off-nadir (emittance) angle
To^4=(1-C)emis*Ts^4 + C*Ta^4        To is the off-nadir brightness temperature

Use of fff requires an additional TDISK-like routine to read-only a type -3
 file, called TDIF3. This results from needing to have two KRC direst-access
 files open simultaneously.  TDISK has been modified to allow writing a type 52
 simultaneously with either -1 or -3.

Changes required to read a -3 file to get Ts and Tp in the fashion of a -1 file:
 FORTRAN: Change the RECL argument in the OPEN statement.
   (Beware, its meaning depends upon the compiler setting of -xl[d])

Note that type -1 and -3 file records are fixed size set by the MAXN4=37 and
 MAXNH=96 and the word type (currently REAL*8) so -3 will be MAXNH*MAXN4*3=10656
 words or 85248 bytes each. The first record contains KRCCOM, currently 1704
 bytes. A typical KRC model of 40 seasons is 3.4 Mbytes

-----

I am considering changeing the type of the Ts, Tp and Ta arrays
 written to type -3 files from REAL*8 to REAL*4 simply to keep the size
 down. This would be done transparently to users of TDISK. KRCCOM would still be
 REAL*8, but there is no need for double precision in the final temperatures.  I
 believe the time to do REAL*4 <-> REAL*8 conversion is trivial.

Changes required to accomodate REAL*4 in the files for other readers:
If user really wants REAL*8 temperatures,
  READ arrays as REAL*4, also define REAL*8 arrays
  Loop over hour and latitude with
   TP8(I,J)=TP4(I,J) and similar for Ts and Ta
  Loop limits could be MAXNH and MAXN4 or the actual N24 and N4

I have tested timing to read type -3 files and to convert between R*4 and R*8.
  For a typical file with 40 seasons (does not matter how many hours or
 latitudes as the array sizes are fixed at the maximum allowed). For type -3,
 reading takes 0.6 to 0.9 ms, but this may be highly dependent upon caching.
  Conversion of the maximum set of hours and latitudes for all seasons takes 1.6
 ms (array was filled with random temperatures).
```

16 Plans for next release

Current concept for thermal beaming is as a post-run process and as such has no impact on KRC.

"Moon-ready" KRC will require changes to several of the KRC commons, and will require an additional loop to handle longitude. This will be a major restructuring.

17 Error codes

The code-set has progressed toward a consistent usage: IRET is the return code argument in a routine, IRL is the name in a call to a lower routine.

TCARD: 1 = normal start 2 = restarted from disk record
3 = continue from current conditions 4 = Switch to "one-point" mode
5 = END of data (no more change cards) in input file
6= Error reading internal buffer 7= End while reading internal buffer
TDAY: 1=normal return **TDAY**(2, 2=numerical blowup
TDAY(1, 3=Some layer unstable 4=Too many Layers generated by zone table
TLATS:
1=normal 2,3,4= error of same number in **TDAY**
C 5=no matching latitude in fff
TSEAS: returns from:
TCARD(2: if > 5, then +10 4=switch to onePoint 5=End of input file
TDAY(1 if $\neq$ 1, then +20
TLATS no change
TFAR 41: Tatm needed but not in the fff.
KRC: 4=PARAMETER ERROR IN TDAY(1)

18 Test runs

Because of the extensive testing done with KRC version 321, see "KRC version 2 and 3: Thin/deep layers and long runs", that is considered the base for testing of later versions.

18.0.5 A: Obsolete

Test Run A is primarily a body in Mars orbit with no atmosphere; it used a 2-year spin-up followed by 1 year that is saved. It had 15 cases defined in *krc33.inp*, produced *M33A.prt* and *M33A.t52*; the differences in diurnal temperatures from case 0 for the last day of the last season are shown in Figure 13. Case 0 had: No atmosphere, was homogeneous with depth, used 44 layers and Lambert albedo. Differences from this case are listed below:
1: Normal Mars with atmosphere and Lambertian soil. Two materials
TUN radiance output to fort.77, renamed to *M33A.77*
2: Lommel-Seeliger albedo
3: Minnaert=0.7 albedo
4: 1 W/m2 heat-flow
5: 2nd material at IC2=9
6: 2nd material at IC2=9, 1 W/m2 heat-flow
7: 30 layers
8: 30 layers, 1 W/m2 heat-flow
9: 30 layers , KofT
10: 30 layers , KofT, 1 W/m2 heat-flow
11: less time doubling
12: no time doubling
13: zone table: zoneX (similar to 2 materials)
14: zone table: zoneY (exercise most table options)

Test Run B is basically the Version 3.3 standard for 3 latitudes: 0, -30 and -45; with the 2nd material starting with IC2=7, used 44 layers and Lambert albedo, had a 2 year spin-up and ran for a total of 20 years. It had 12 cases defined in *krc33B.inp*, and produced *M33B.prt* and *M33B.t52*. Case 1,7: are the version 3.3 standard Mars homogeneous with depth; the second case number is with 100 milli-Watt/m^2 geothermal head-flow. Case differences from this are listed below: (1-based index)
2,8: Two materials with IC2=7
3,9: use T-dep materials
4,10: Two materials with IC2=7 and use T-dep materials
5,11: zone table: zoneX (similar to 2 materials)
6,12: zone table: *Grott07.tab*, which follows Grott07=[5] with their higher conductivity

Heat flow of 100 milli-Watt/m^2 is about a factor of 3 higher than expected for Mars, but chosen to make the effects of heat-flow easier to see in plots.

The effect of heat flow is shown in Figure 14. The effect of 20mW/m^2 is 1K at about index 36, KRC layer 38, depth 2.7m .

18.0.6 nill Atmosphere

A test of the progression toward low atmosphere pressures was run, 342c. One must be careful to ensure that the Photometry parameter ARC2/PHT is 0 for the nill-Pressure case to ensure that it is Lambertian, as the atmosphere cases are forced to be Lambertian in the TLATS code.

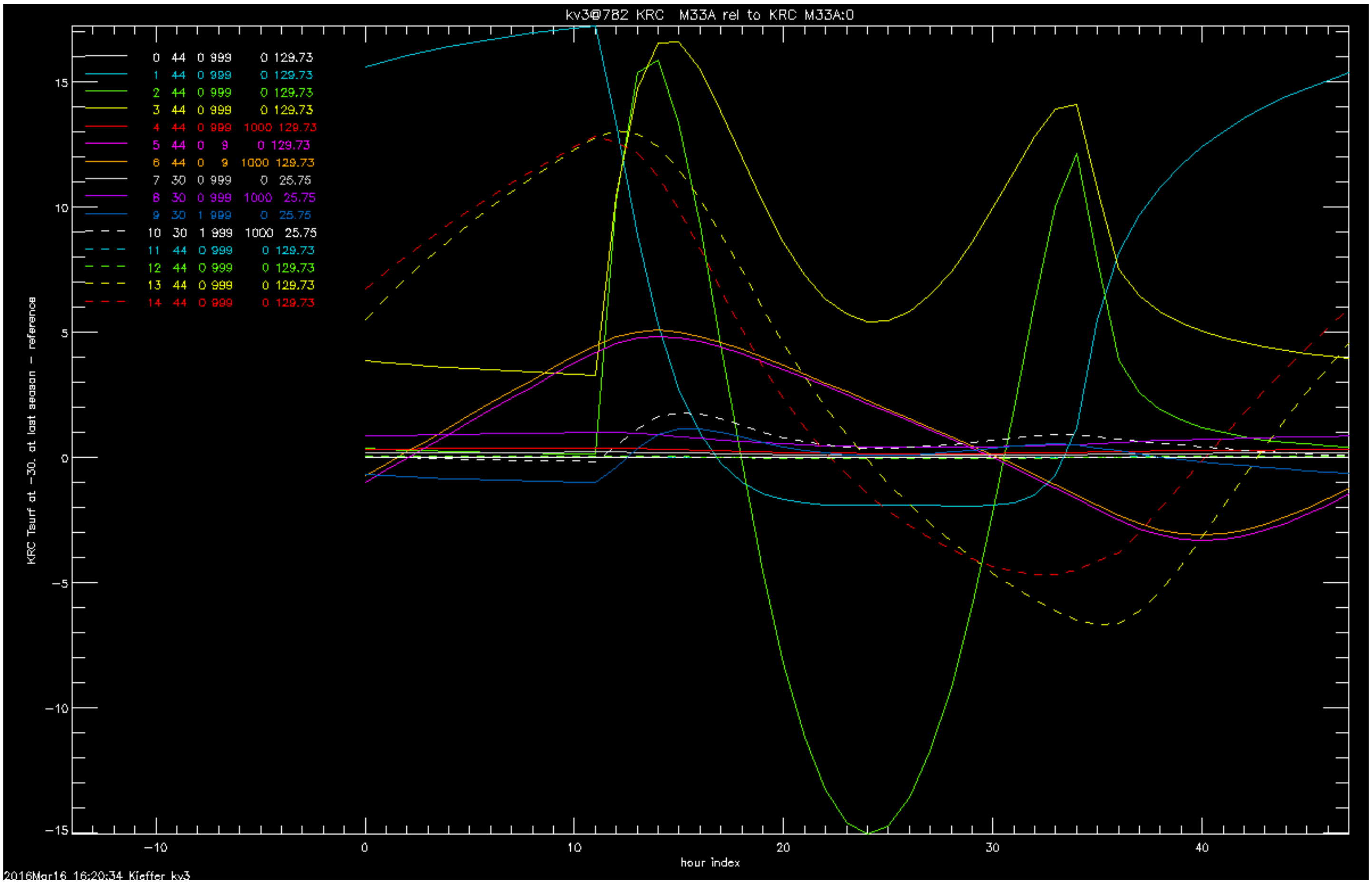

Figure 13: Diurnal surface temperature in Version 3.3 test run A at the last season (Ls=0) at latitude 0 relative to the first case. Difference in cases described in the text. kv782M33A.png

Results are shown in Figure 15

18.0.7 Version tests

Standard comparison of version 321 and 342 was with a set of 5 latitudes for every sol with 15 layers and no spin-up, Figure 16; runs had the same six cases:

1: base=Mars std atmosphere, surface with T-constant properties
2: as case 1, but with T-dependent properties
3: as case 1, with T-con. but with variable frost temperature and albedo
4: no atmosphere, Lambertian, T-constant properties
5: as case 4, but T-dep. properties
6: as case 5, but the T-dependence set to zero

A second standard comparison used 20 layers for 19 latitudes for 40 seasons after a two-year spin-up but only case 1; Figure 17).

For both of these runs, CONVF was 2 for V321, yielding lower layer of time doubling 2 4 6 8 10 12 14 15or20; and CONVF was 3 for V342, yielding 4 5 7 9 11 13 15 [20].

A run of 5 latitudes with 20 layers for 40 seasons after a two-year spin-up but CONVF=3 for both versions, yielding lower layer of time doubling 4 5 7 9 11 13 15 20 for both versions, had smaller differences: Figure 18

Matching with version 232, 321 and 342 were done using for 5 latitudes (-60, -30, 0, 30, 60), 40 seasons with a 2-year spinup, 20 layers with similar time doubling (Lower layer of time doubling: 4 5 7 9 11 13 15 20 for V321 and V342; 4 6 8 10 12 20 for V232) were done for representative Martian physical properties; I=200 over ice begining at 6 cm. Each run had the same 8 cases:

1: base=Mars std atmosphere, surface with T-constant properties
2: as case 1, but with T-dep. properties
3: as case 1, but with variable frost temperature and albedo
4: as case 1, but 30 ° slope to 90° azimuth (West)
5: no atmosphere, Lambertian, T-constant properties

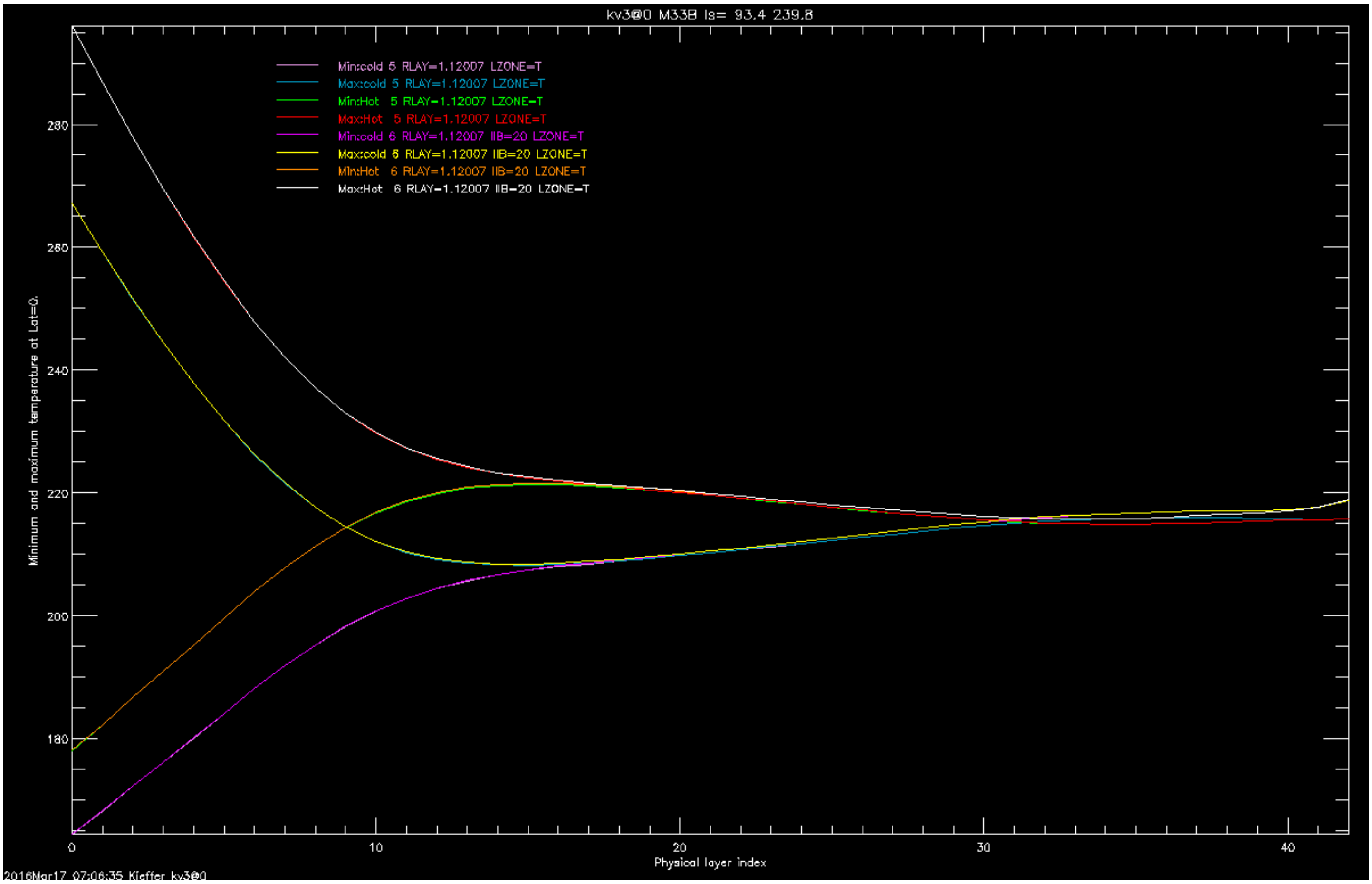

Figure 14: Temperature versus depth for a model similar to the higher conductivity model of Grott07=[5], but at latitude 30S which has larger seasonal variation than the planned InSight landing area. Abscissa is 0-based physical layer index, which is roughly proportional to the log of depth. Ordinate is temperature. First 4 curves are without heat-flow and the second 4 with heat-flow of 20 mW/m^2; the sets virtually overlay for index less than 25. Of each set, the first 2 are at the cold season, Ls=93°; second 2 at the hot season, Ls= 240°. Of each pair, the first is the minimum diurnal temperature and the second is the maximum. Colors lower in the legend can overwrite those above. kv7847.png

6: as case 5, but T-dep. properties
7: as case 6, but the T-dependence set to zero
8: as case 5, but 30 ° slope to 90° azimuth (West)

Run times: 232= 2.445 sec, 321= 3.036 341= 3.0445

Differences in Tsurf are shown in Figures 18, 19 and 20. The difference for T-dep cases results from small changes in the layer thicknesses as discussed in the Alert in §3.

DownVis: 342-321 has deltas up to 8 for the atm. cases. Unexplained.
looks like revised amplitude. but ratio largest at noon, and varies with season
232-321 has deltas up to 15 that resemble the delta Ts.
no-atm. cases identically zero

DownIR: 232-321: delta up to 8 for the atm cases.

@570 342b-321 lat=0, last season:
Ts: case 2 and 5 (T-dep) 0.4K cool in morning, 0.8K hot after dusk
Insolation low by up to 80 at noon
342b-321, deltas at roundoff. downir deltas largest for case 2, and 3:5 all the sma.

342c: downir values for 341 seem to be the same for all cases! Fixed
downir is ATMRAD

```
KRCINDIFF: test for changes. Input limits:       64     120     220
                    V342       V321
out   i     Label      Arg1        Arg2        Arg1-Arg2
 34 33     FLAY      0.21600      0.18000    0.036000 < but yields same layers
 65 64     HUGE 1.0000e+308  3.3000e+38 1.0000e+308
```

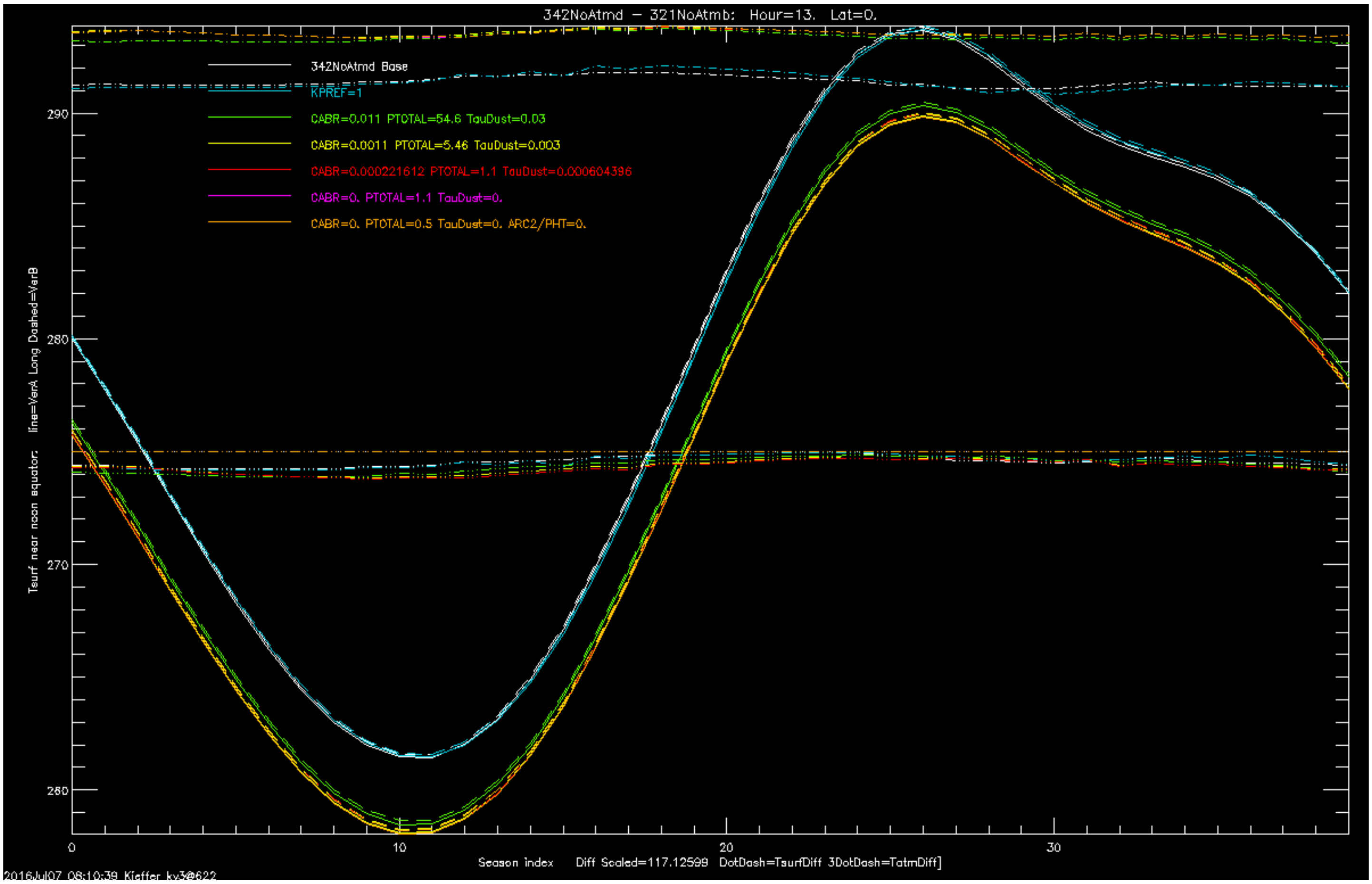

Figure 15: Surface temperatures as a function of atmospheric pressure; Mars seasonal variation of the equator at 13 hours after a t2-year spin-up. Solid lines are V342, long dash are V321. White is standard mars conditions with constant pressure, blue is pressure following the Viking lander annual variation. Green, yellow and red, are constant pressures of 1/10, 1/100 and 1/500 Mars, reducing the dust opacity and the CO_2background opacity by the same fraction. Purple is also .1/500 Mars (1.1 mBar) but with dust CO_2background opacity both set to zero; orange is no atmosphere. 622c.png

```
 66 65     TINY 1.0000e-307  2.0000e-38 -2.0000e-38
 67 66   EXPMIN      700.00      86.800      613.20
 71 70   TATMIN      143.39      143.39 -1.4859e-05
 72 71     PRES      912.60      912.60  -0.0020989
 73 72  OPACITY     0.50143     0.50143 -1.1532e-06
 74 73    TAUIR     0.30921     0.30921 -7.1116e-07
 75 74   TAUEFF     0.61843     0.61843 -1.4223e-06
 76 75    TATMJ      165.99      165.99 -0.00011073
 82 81   TEQUIL      154.52      194.78     -40.258  < ??
 85 84   SCALEH      8.2048      8.2047  3.7947e-05
 86 85     BETA     0.46121     0.46121 -7.6633e-07
117 16    K4OUT      -2       52      -54
118 17    JBARE    9999        0    9999

Using Aug 30 13:35 tlats;
 82 81   TEQUIL      192.24      194.78     -2.5368


321----------
      AVEA=ALB ! surface albedo; will be frost if frosty at end of prior day
      AVEI=AMAX1((1.-AVEA)*AVEI/DFLOAT(N2),0.) ! average absorbed insolation
      IF (LATM) THEN             !v-v-v-v-v  with atmosphere
        TSEQ4=BETA*TAEQ4+AVEI/(SIGSB*AVEE) ! equilib T_s^4
```

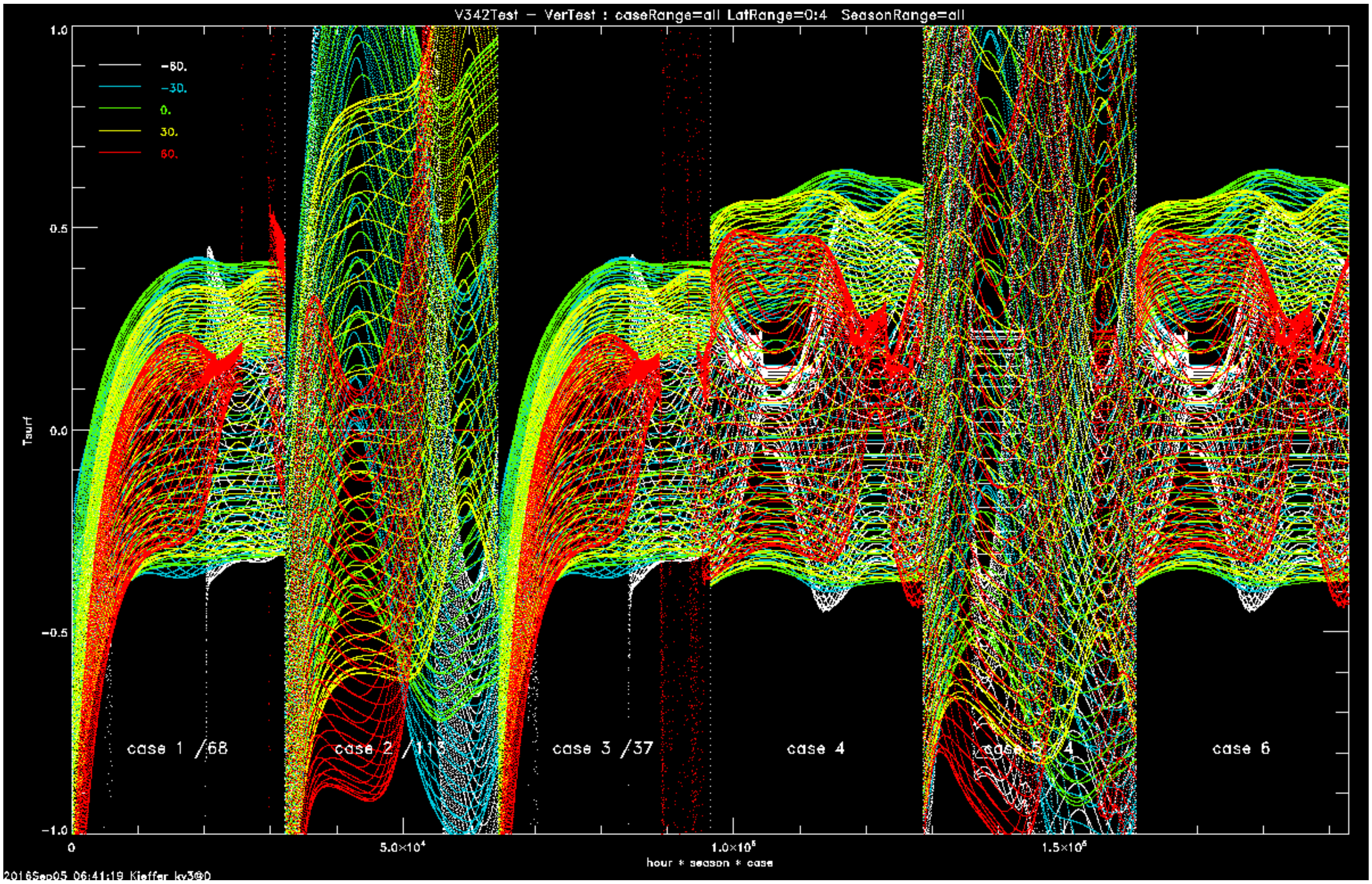

Figure 16: Difference of surface temperature for V342-V321 for each sol for five latitudes as shown in the legend. See text. 342soly-321.png

```
        TEQUIL = AMAX1( TSEQ4,1.D4)**0.25D0 ! equilib T_s, min of 10.
      ELSE                      ! no atmosphere
        TEQUIL = ((1.D0-AVEA)*AVEI/(SIGSB*AVEE))**0.25D0 ! equilib T_s

342----------
      AVEA=ALB ! surface albedo; will be frost if frosty at end of prior day

      IF (LATM) THEN              !v-v-v-v-v  with atmosphere
        PHOG=0.                   ! force to be Lambert
        KOP=1                     ! Lambert flag
      ELSE                      ! +-+-+-+-+  no atm. may use photometric functions
        PHOG=ARC2                 ! reassigned to be the photometric value

             AVEA=ALB*AHF   where AHF is photometric factor, =1 for Lambert
      AVEI=AMAX1((1.-AVEA)*AVEI/DFLOAT(N2),0.) ! average absorbed insolation
      IF (LATM) THEN
        TSEQ4=BETA*TAEQ4+(AVEI+GHF)/(SIGSB*AVEE) ! equilib T_s^4
        TEQUIL = AMAX1( TSEQ4,1.D4)**0.25D0 ! equilib T_s, min of 10.
      ELSE                      ! no atmosphere
        TEQUIL = (((1.D0-AVEA)*AVEI+GHF)/(SIGSB*AVEE))**0.25D0 ! equilib T_s
```

```
from 232 to 321 to 342, Tlats grew from 480 to 506 to 679 lines
 tday:  569 to 487 (deletion of many debug) to  823
```

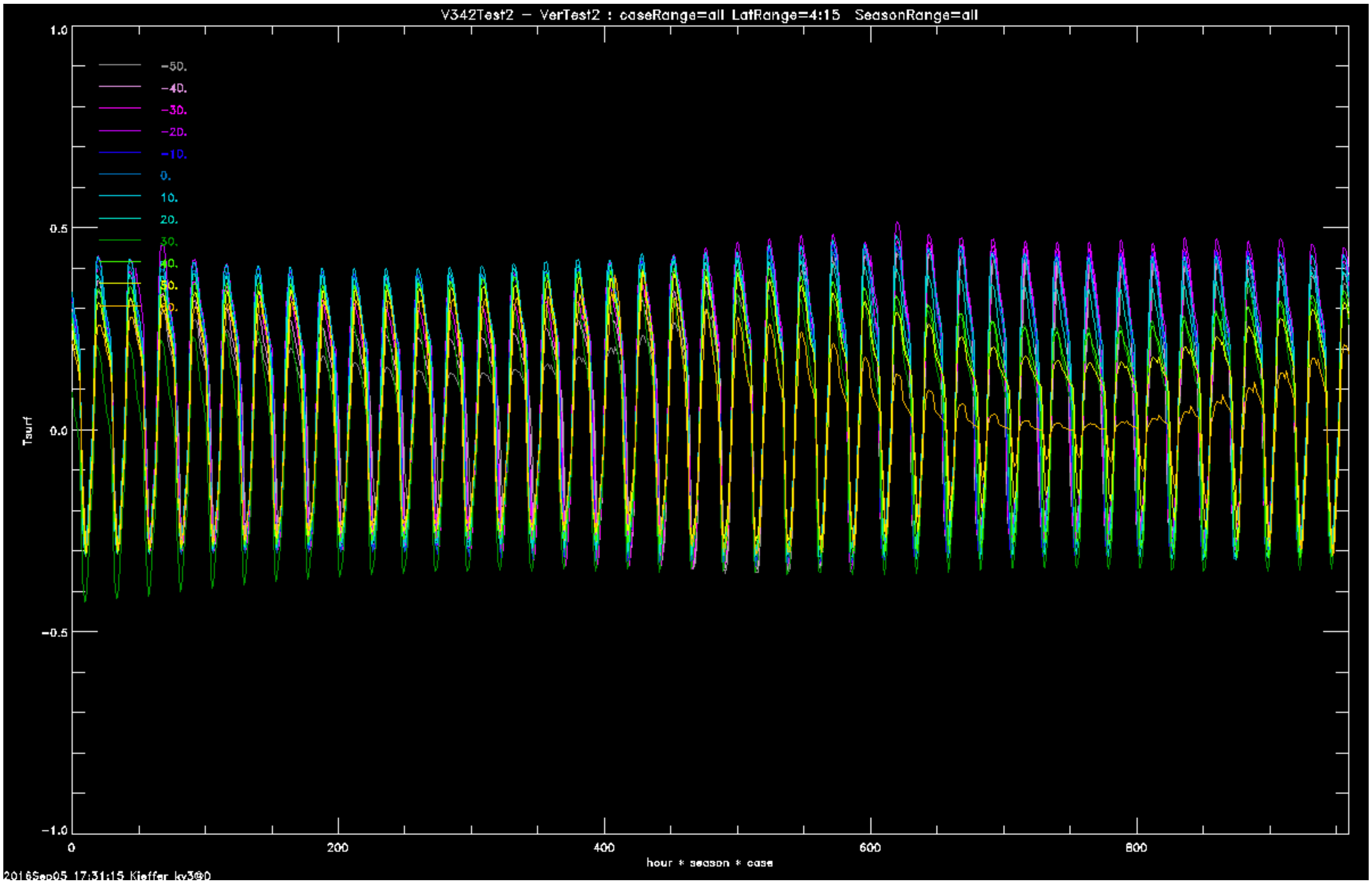

Figure 17: Difference of surface temperature for V342-V321 for 19 latitudes for a runs with 20 layers for 40 seasons after a two-year spin-up. 342vt2-321.png

```
2016 Sep 2 18:45:40 check processing of  232 .t52 versus .tm1
a) kv3 @114 4 232 321 @11 6=V232test2 17=.tm1   @201 202 207 @252
    @12 0=23 23=-1 @50   @51 -> 0   @53
b)  kv3 @114 4 342 342  @11 17=.tm2  @201 202 207 @252
    @12 0=43 23=-2 @50   @51 -> 0   @53
All identical.

         ___THICKNESS____     __CENTER_DEPTH__    Total  Converg.
 LAYER    scale      meter      scale      meter   kg/m^2  factor
    2    0.2160     0.0070    0.1080     0.0035    11.224   2.851
   20   39.3144     1.2768  199.4641     6.4780  6639.141  15.790

         ___THICKNESS____     __CENTER_DEPTH__  Conductiv. Density Sp.Heat     Total Converg.
 LAYER  D_scale      meter    D_scale      meter      W/m-K   kg/m^3   J/kg     kg/m^2  factor

    2    0.2160     0.0070     0.1080     0.0035 0.3864E-01  1600.00  647.00    11.224   2.851
   20    5.7506     1.2768    30.5485     6.4780  2.770        928.00 1711.00  6639.141  15.790

    2    0.2160     0.0076     0.1080     0.0038 0.3951E-01  1600.00  568.17    12.112   2.851
   20    5.7506     1.4720    30.5485     7.4642  3.226        928.00 1499.21  7647.643  15.790


T-global= 190.09
```

18.0.8 IDL notes

```
.rnew kv3
```

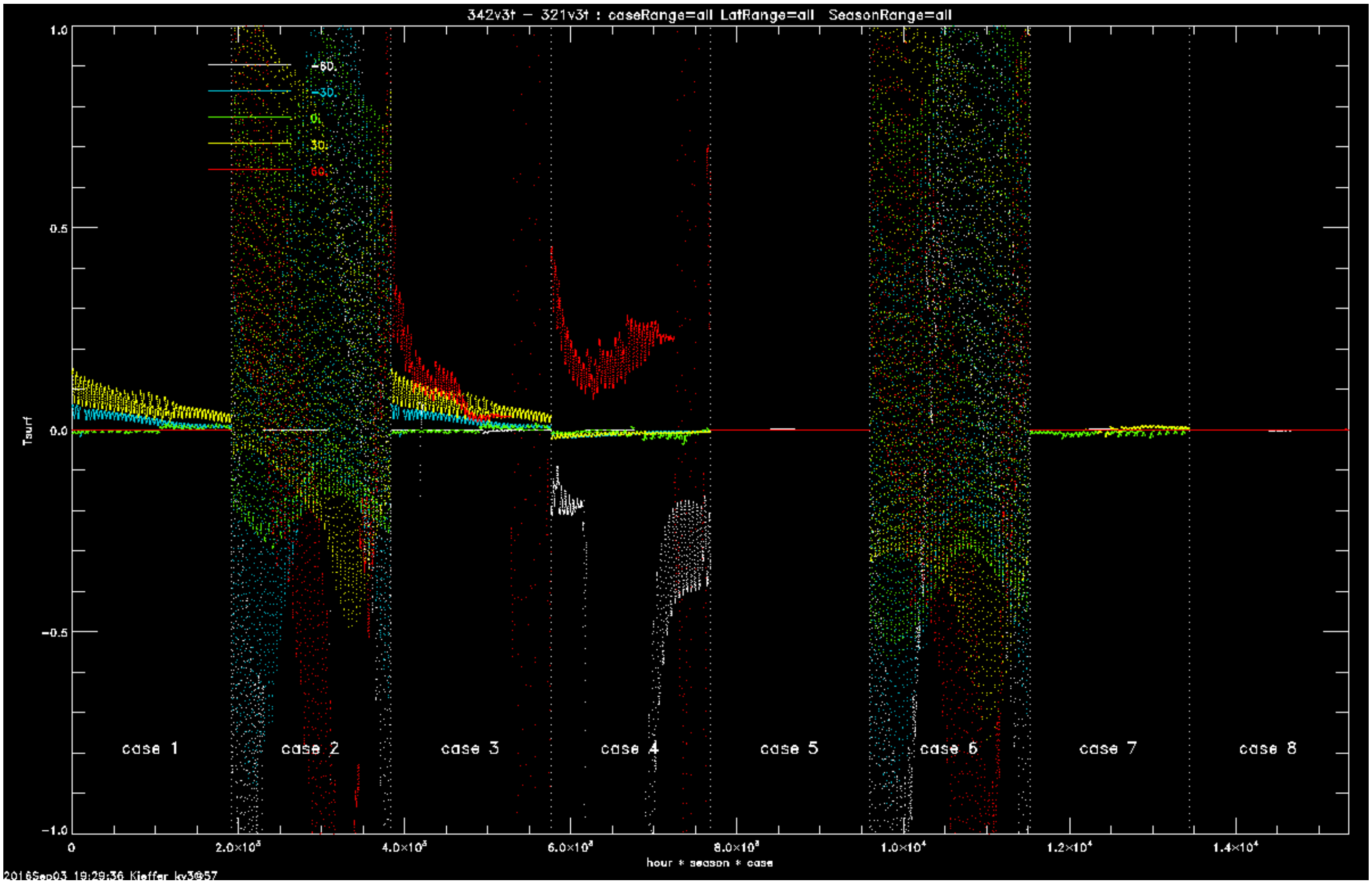

Figure 18: V342-V321 for 8 cases. Abscissa is hours * season * case. Ordinant is Tsurf of 342 - 321. Five cases have differences less than about 0.1 K, apart from winter frost edges. Those with T-dep properties have differences are up to 1.5 K, apart from frost edge. kv51.png

```
@114: 4, 321 342 ,  111 123
@65
@66:  46 48 55 (respond many 1) -2  to prepare for HALB comparison
@ 661
@ 651 662

@402   CLOT DownVis then Tsur for all cases
@56  t 0    Select array and item
@561        Prepare the difference
@562        Stats versus latitude
@563        QUILT3 and make dd

CLOT,reform(qy[*,0,0,*])

CLOT,reform(ttt[*,0,2,3,*]),caset,locc=1  DIurnal T, equator, one season, all cases

@12 7=0 8=1  check effect of TAUD and CABR when no atm,
@ 56 561 ...result == 0
@12 7=-1 8=0 compare to lambert
@13 9=0 not absolute
@ 56 561
 clot,reform(ttt[*,0,2,3,*]),caset,locc=1
 clot,reform(ttt[*,3,2,3,*]),caset,locc=1
```

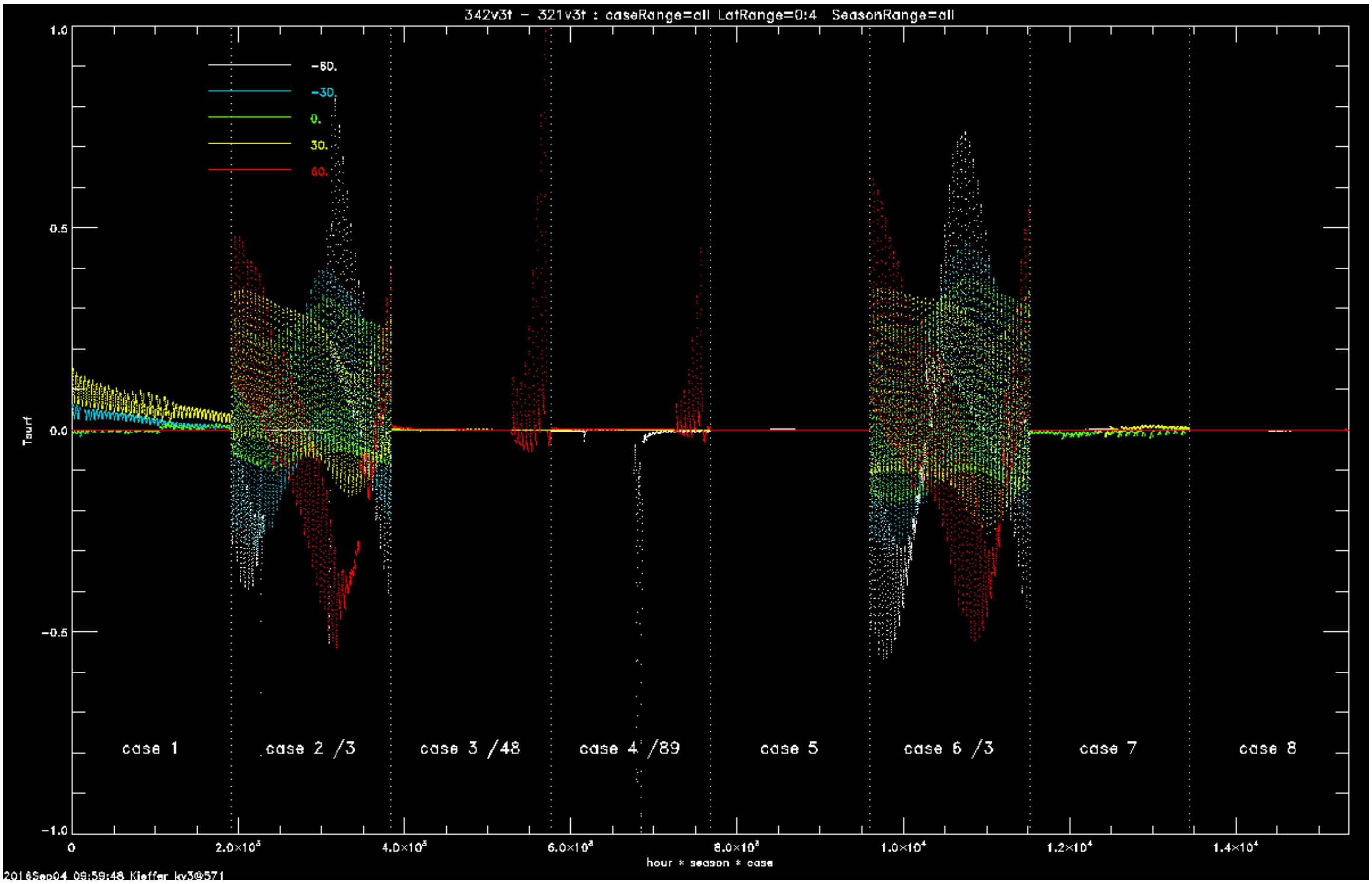

Figure 19: Same as Fig. 18. Cases with large excursions are shown attenuated by factors listed in the figure. kv571.png

References

[1] R.C. Birkebak. Thermophysical properties of lunar materials from the Apollo missions. In *Advances in Heat Transfer (T.F. Irvine, Jr., and J.P. Harnet, Eds.)*, volume 10, pages 1–37, 1974.

[2] R.C. Birkebak, C.J. Cremers, and J.P. Dawson. Directional spectral and total reflectance of lunar material. *Proc. Apollo 11 Lunar Sci. Conf.*, 3:1993–2000, 1970.

[3] H. S. Carslaw and J. C. Jaeger. *Conduction of heat in solids.* Oxford University Press, London, 1959. 2nd ed., 510 pp.

[4] H. B. Dwight. *Tables of Integrals and other Mathematical Data.* MacMillan, New York, 1961. 4th ed., 336 pp.

[5] M. Grott, J. Helbert, and R. Nadalini. Thermal structure of Martian soil and the measurability of the planetary heat flow. *Journal of Geophysical Research (Planets)*, 112:9004, September 2007.

[6] B. W. Hapke. *Theory of Reflectance Spectroscopy.* Cambridge University Press, Cambridge, 1993. 455 pp.

[7] S. J. Keihm. Interpretation of the lunar microwave brightness temperature spectrum - Feasibility of orbital heat flow mapping. *Icarus*, 60:568–589, December 1984.

[8] H. H. Kieffer. Thermal model for analysis of Mars infrared mapping. *Jour. Geophys. Res., Planets*, 118:451–470, 2013.

[9] A. R. Vasavada, J. L. Bandfield, B. T. Greenhagen, P. O. Hayne, M. A. Siegler, J.-P. Williams, and D. A. Paige. Lunar equatorial surface temperatures and regolith properties from the Diviner Lunar Radiometer Experiment. *J.Geophys. Res. (Planets)*, 117:E00H18, April 2012.

A Lunar albedoes

A.1 Email from Sylvain

05/05/2016 04:18 PM

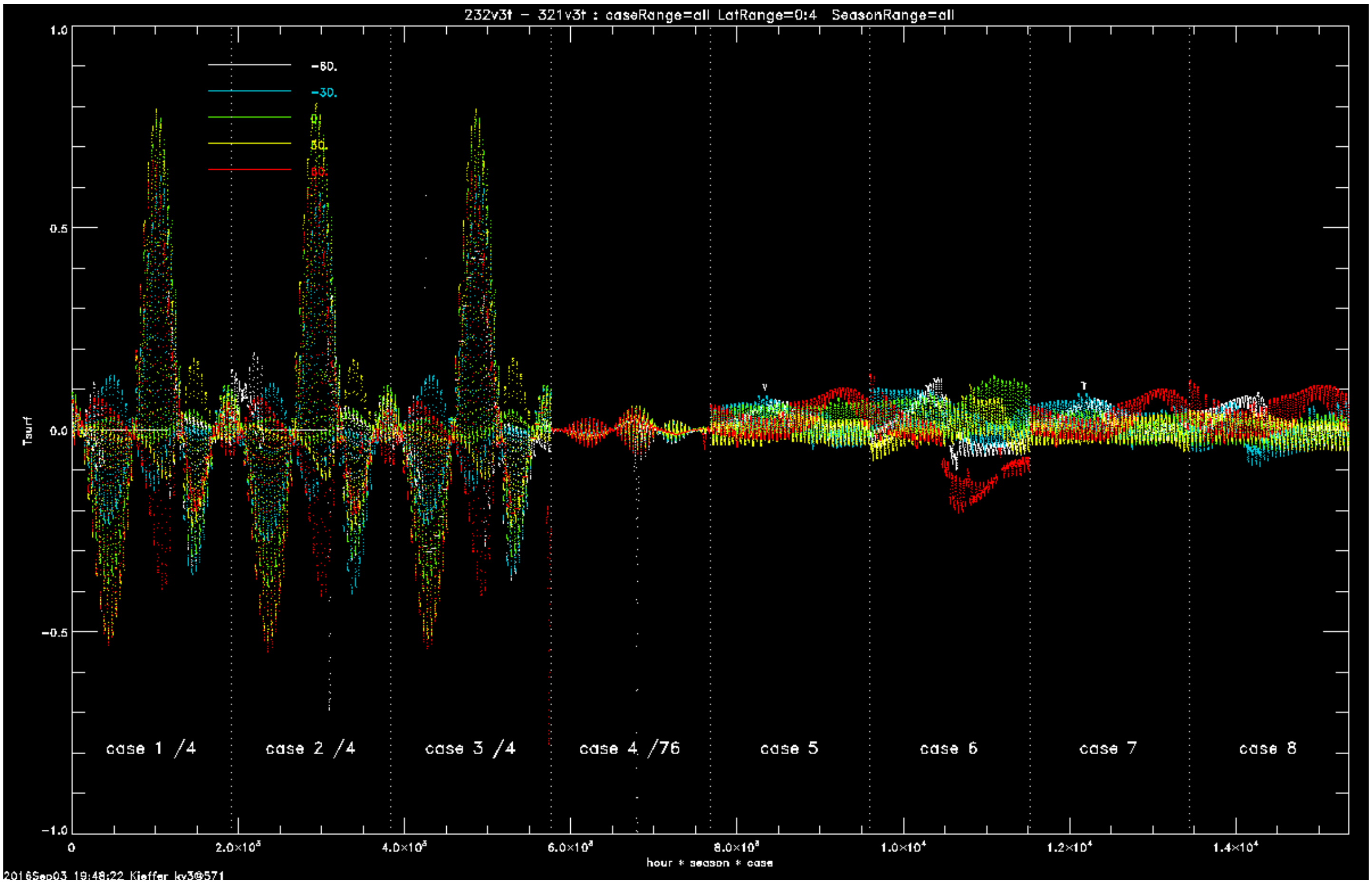

Figure 20: V232-V321 for 6 cases. See caption for Fig. 19 Abscissa is hours * season * case. Ordinate is Tsurf of 232 - 321. Four cases with no atmosphere have differences less than about 0.1 K, those cases with atmosphere are shown attenuated by a factor listed in the figure; the differences are up to 3.2K, apart from seasonal cap edge. kv571b.png

```
As I am diving in the new KRC capabilities (photometric functions,
temperature-dependent properties at depth, etc.) as well as what's potentially
missing for airless bodies, there is this one other thing that bugs me:

KRC can now use a choice of surface photometric functions (Minnaert, etc.) but
an integration is necessary to turn each one of the chosen function into
hemispherical reflectance = surface albedo that depends on solar incidence. I
started to think more about these things after seeing a presentation for a
student of David Paige that compared hemispherically integrated photometric
functions from lab measurements vs. theoretical formulations and existing
formulation (Keihm 1984 and others).

I can't figure out where in KRC this integration is done, but more importantly,
why not using a Kheim1984-like formulation which actually directly gives the
hemispherically integrated albedo as a function of solar incidence (i.e. the
integration is already done for us, without having KRC doing it) as opposed to
-as of now- the photometric function that must be integrated by a black box (to
most users unless they look at the KRC codes) in KRC?

This way, the user would directly provide a and b (from Keihm 1984 Eq. A5) as
inputs, which would directly lead to the surface albedo, without relying on KRC
for the intermediate hemispherical integration. Again, a and b are from Equation
A5 in the Keihm paper we discussed a few months ago (attached with this email)
and would be the 2 input provided by the user. Is there a reason not to do that?
```

```
And while I am looking at the Keihm paper and some of my old notes from a
telecon last year with you and Phil, I wrote that KRC could integrate a
temperature-dependent surface emissivity (Keihm, Eq. A4). Do you still believe
that an upcoming version of KRC could integrate that capability?

05/06/2016 09:25 AM

I think what is throwing me off is the opposite trends between the photometric
functions found in the general optical literature and the functions described by
the lunar folks.

The Minnaert/Lommel-Seeliger/other functions have the bond albedo decreasing
when the Sun gets closer to the horizon; the bond albedo derived from lunar
observations and laboratory work increase when the Sun gets closer to the
horizon. Vasavada et al. refined the Keihm 1984 formulation with new values for
a and b to best fit the Diviner data (See Vasavada paper Eq.(1) vs Keihm
A5). Also see a presentation by Emilie Foote and David Paige compiling
laboratory measurements (specifically the last 2 slides) also confirming the
opposite trends.

With a negative exponent, the Minnaert function can get somewhat close to the
Kheim or Vasavada formulation (except near the limb as you mention in the
documentation where the bond albedo > 1) but the current input system in KRC
cannot understand negative exponents (ARC2 < 0 => Lommel-Seeliger).
```

A.2 Hemispheric Albedo

In */work2/Reprints/* I have Keihm84=[7] as *Lunar/Keihm84.pdf*, Vasavada12=[9] as *Photom/Vasavada12Lunar.pdf* and the Foote and Paige presentation as *Photom/Foote13DivinerOxford.pdf* .

Keihm84=[7] has (equation A5):

$$A(\theta) = 0.12 + 0.03\,(\theta/45)^3 + 0.14\,(\theta/90)^8 \tag{37}$$

Extracts from Vasavada12=[9]

§3.1 [25]: " Within this constrained data set, we find that for darker surfaces, the dependence of reflectance on phase angle can be removed by dividing by$\mu_0^{1.3}$. This is a slightly stronger dependence than for a Lambertian surface (i.e., dividing by μ_0). It is difficult to assess its appropriateness for brighter surfaces due to (unresolved) surface slopes that cause higher levels of scatter in the data."

§5.2 [44]: " Because the Diviner solar reflectance data used in section 3.1 are measured normal to the surface, they cannot be used to define the full bidirectional reflectance of the surface. But daytime temperatures, being close to radiative equilibrium with the instantaneous insolation, can be used to infer the angular dependence of albedo. We find that the Apollo-derived formulation of Keihm [1984] reproduces the observations well, where

$$A(\theta) = A_0 + a\,(\theta/45)^3 + b\,(\theta/90)^8\,, \quad (1) \tag{38}$$

and A_0 is our Diviner normal albedo at each longitude. We derive a best fit value of $a = 0.045$ (modified from 0.03) and keep Keihms value of $b = 0.14$."

Converting to a normalized form:

Can represent Keihm as

$$A(\theta) = 0.12\left[1. + 0.25\,(\theta/45)^3 + 1.17\,(\theta/90)^8\right] \tag{39}$$

Vasavada would be, at least for $A_0 = .12$,

$$A(\theta) = A_0\left[1. + 0.375\,(\theta/45)^3 + 1.17\,(\theta/90)^8\right] \tag{40}$$

In TLATS, code as factor=1 + $f_1\theta^3$ + $f_2\theta^8$ where θ is in radians and $f_1 = x/(\pi/4)^3$ where x is derived from the input parameter and $f_2 = 1.17/(\pi/2)^8$

Both are encoded into KRC 3.4 by over-using the single available photometry parameter: -x: $0 < x < 1$

In some later version of KRC, isolate several input parameters to handle hemispheric albedo. Possible form:

$$A(\theta) = c_0\left[1. + c_1\,(\theta/45)^f + c_2\,(\theta/90)^g\right] \tag{41}$$

A.2.1 Foote

We have developed a simplified BRDF function that takes the following form:

$$REFF(i,e,g) = \frac{2X}{\mu_0 + \mu + Y}\left(1 + B(g)\right)p(g)$$

Where X and Y are constants.

Where

$$B(g) = \frac{b_0}{1 + \frac{\tan(g/2)}{h}} \qquad \text{and} \qquad p(g) = \frac{(1-c)(1-b^2)}{(1+2b\cos(g)+b^2)^{3/2}} + \frac{c(1-b^2)}{(1-2b\cos(g)+b^2)^{3/2}} \tag{42}$$

Adjustable parameters: $b,\ b_0,\ c, h,\ Y,\ X$.

A.2.2 Birkebak Apollo samples

Birkebak measured hemispherical albedo (reciprocal method) of several Apollo soil samples; his term is "directional reflectance", Birkebak70=[2] and Birkebak74=[1] .

Spectral observations were made at 15,30,45 and 60°, weighted with the solar irradiance and fit with polynomials in i up to 6'th degree; Birkebak74=[1] Tables IV and V list the coefficients. However, the analytic expressions were forced to go thorough unity at 90 degrees. The analytic fits are shown in Figure 22

His Table III lists "solar albedo" at six discrete incidence angles from earlier work; these are shown in Figure 21 and Figure 22

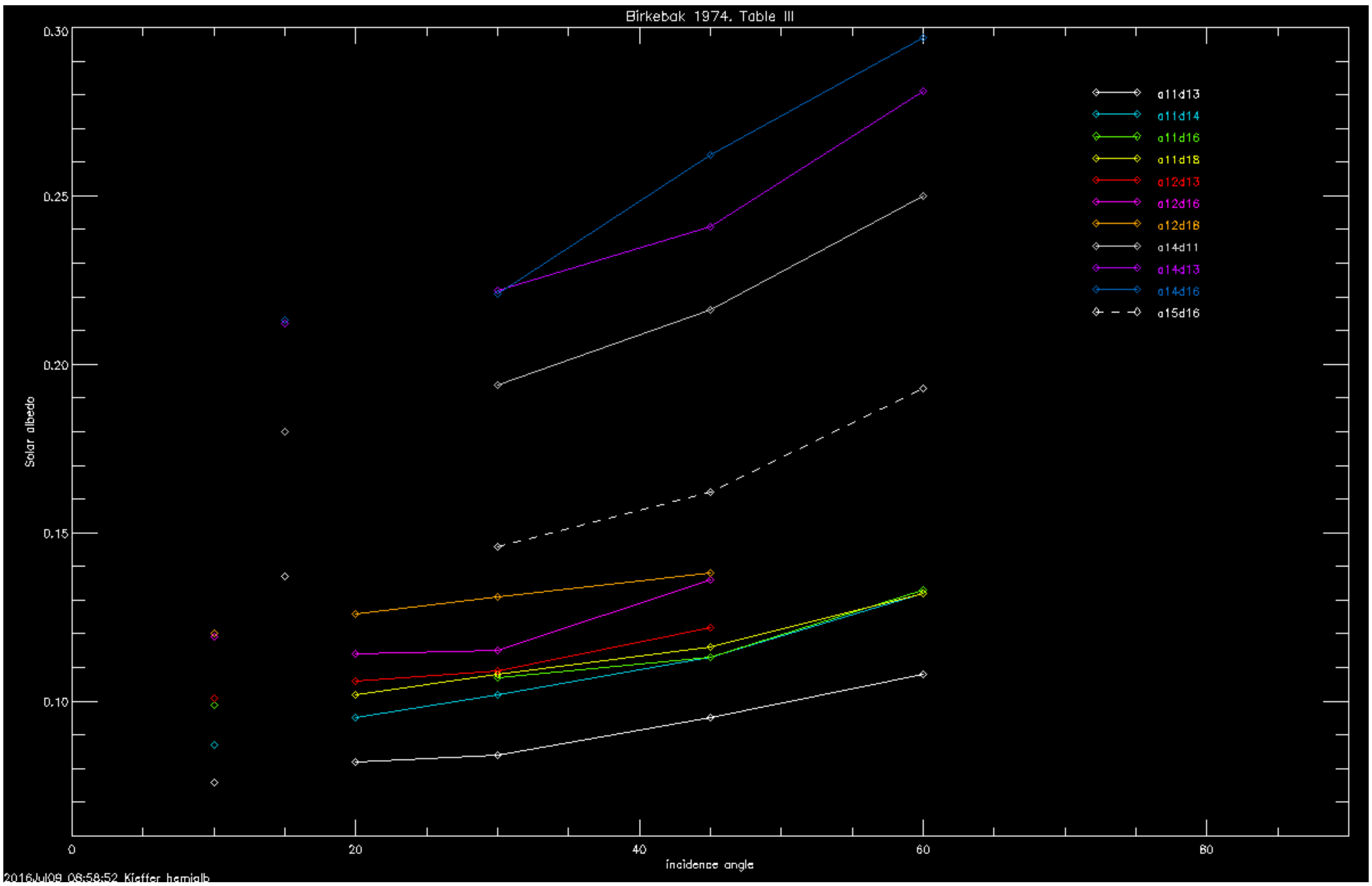

Figure 21: Solar albedo for specific incidence angle, from Birkebak table III. Lines omitted across missing data. Birk2.png

Some of the published fits pass through all the data points, others have up to about .01 residuals, but most have too much curvature to look realistic.

A.2.3 My try at fits. FUTURE

Not many data points, so can't use many coefficients

$A_h = A_0(1. + c_0(\theta/90.)^{c_1})$, one non-linear term, can use BRENTX See what the family of curves looks like. Begin coding: hemialb.pro @71

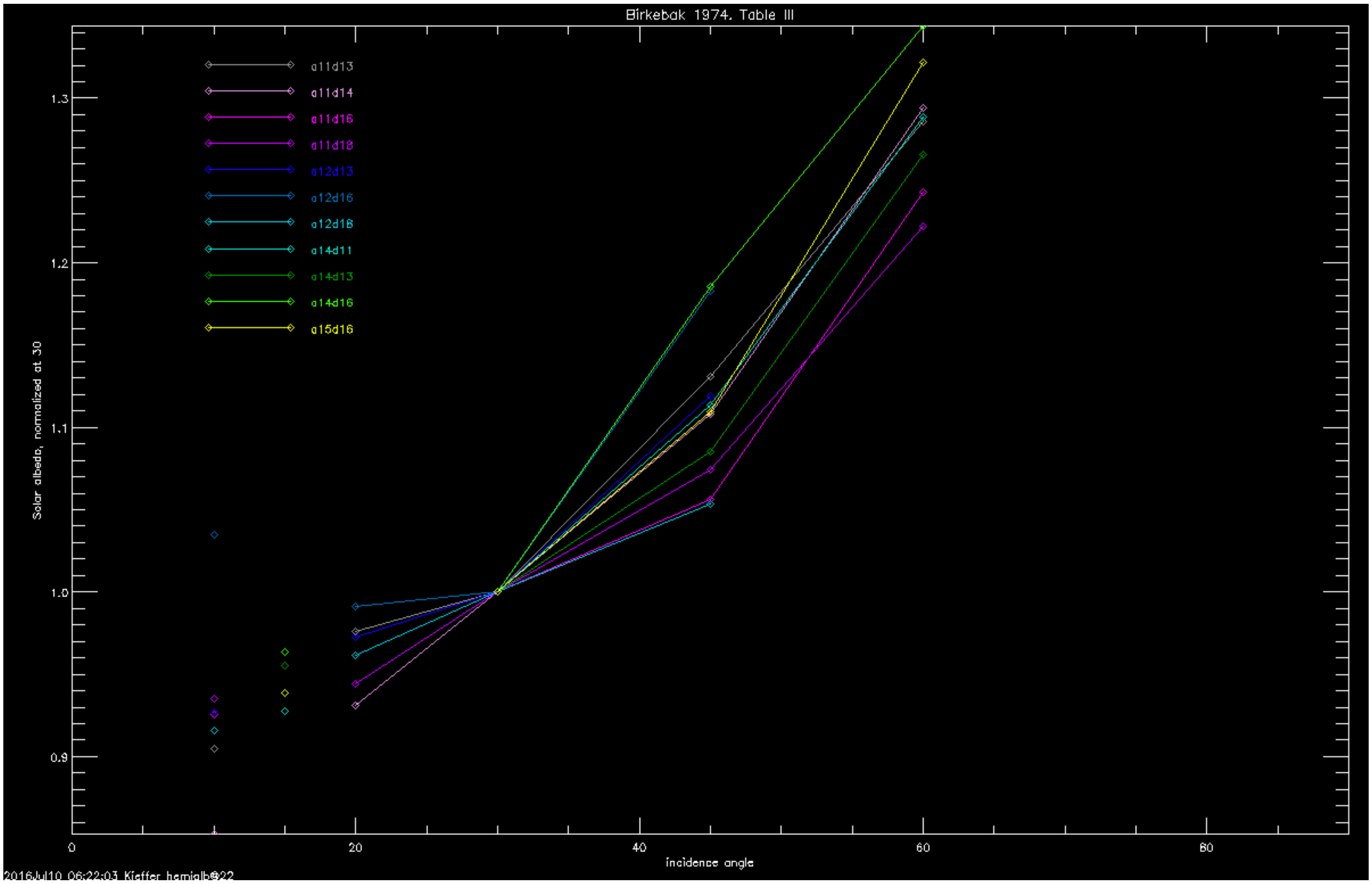

Figure 22: Solar albedo for specific incidence angle, Data same as Figure 21, but normalized to values at i=30°. Apollo 12 with density 1600 seems wayward. Birk3.png

B Effect of photometric function

Results for each type of photometric function in KRC v3.4.x are shown in Figures 23 and 24. Some normalization is missing for Lommel-Seeliger. The Vasavada relation appears unrealistic.

C Tuning

Using a nominal Mars case for I=300 with a year of 40 seasons after a 2-year spin-up; latitudes 0, -30 and -45° as the southern hemisphere as the more extreme seasons. Tried various geometric ratios RLAY and number of layers N1; FLAY was adjusted so that the bottom depths are all the same.

Ran a case with the maximum number of layers, RLAY= 1.12, and 8-fold increase in times steps as the reference, which takes about 8 times as long to run. Case inputs and results for 30°S are in Table 1; Temperature through the day of the last season are shown in Figures 25 and performance is in Figure 26.

Efficiency is basically displacement down perpendicular to the dotted line in Fig. 26. Cases 7,11 and 14 are similar, with case 8 being the best (cases 5 and 12 identical). The fine time-step cases show that reducing the convergence safety factor in this case saves about 0.30 sec (5%) for each reduction of 1, with small MAR, 7 mK, for the first and another 21 mK for the second. Cases 3,4 and 5 tested the same CONVG changes, also indicating that the major performance improvement is going from CONVG of 1 to 2.

Although the MAR for all 3 latitudes are similar, the generality of these results has not been explored.

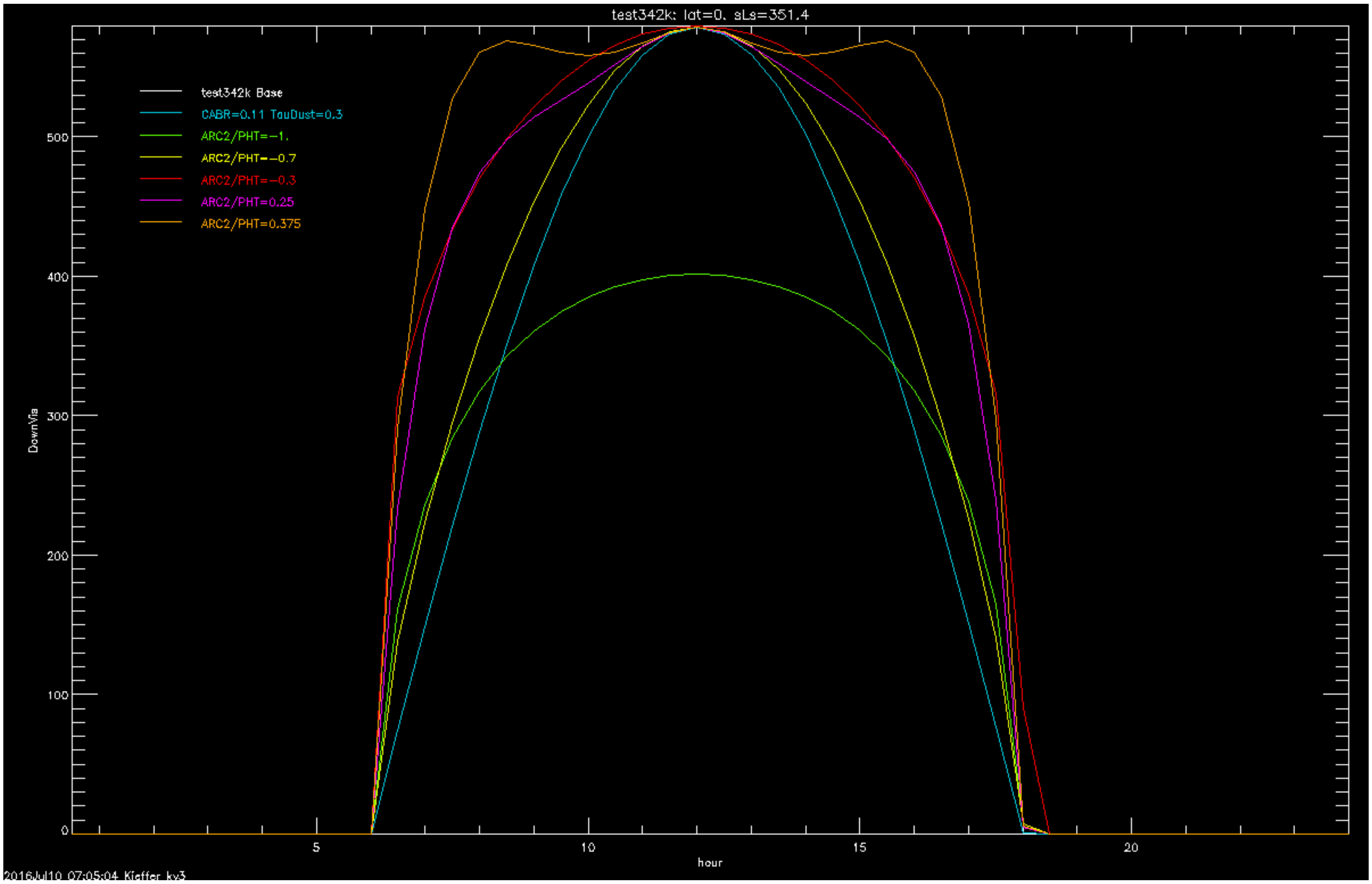

Figure 23: Downward solar flux at the surface for different photometric functions, no atmosphere. First case (in legend, from the top) is Lambertian (2nd is same, testing no atm), third is Lommel-Seeliger, 4th and 5th are Minnaert with k= 0.7and 0.3, 6th is Keihm, 7th is Vasavada. kv402v.png

Table 1: Cases for the tuning run. 'Deep' is depth to the bottom in units of D; 'Sconv' is the average convergence safety factor; 'secs' is the execution time for the case (all lats and seasons); MAR is the mean absolute residual of surface temperature for 30°S; 'Tdel' is the temperature difference at 7 Hours. The first 3 cases are identical except for CONVG, and are cases 3:5 3. Case 6 has 1/2 the time steps. Case 7 is similar to case 2, but must use a larger first layer and smaller RLAY because of the larger time-step. Last are 3 sets of constant RLAY, each with increasing number of layers.

i	RLAY	FLAY	CONVF	N1	N2	Deep	Sconv	secs	MAR	Tdel
0	1.120	0.063	3.00	50	12288	134.36	7.69	6.388	0.000	0.000
1	1.120	0.063	2.00	50	12288	134.36	7.69	6.088	0.007	-0.006
2	1.120	0.063	1.00	50	12288	134.36	7.69	5.696	0.028	-0.004
3	1.150	0.100	1.00	39	1536	134.36	2.44	0.791	0.084	-0.106
4	1.150	0.100	3.00	39	1536	134.36	2.44	0.921	0.054	-0.133
5	1.150	0.100	2.00	39	1536	134.36	2.44	0.884	0.053	-0.143
6	1.150	0.100	2.00	39	768	134.36	1.22	0.487	0.129	-0.359
7	1.100	0.127	2.00	50	1536	134.36	3.95	0.927	0.038	-0.073
8	1.120	0.124	2.00	44	1536	134.36	3.78	0.884	0.038	-0.075
9	1.120	0.099	2.00	46	1536	134.36	2.39	0.954	0.061	-0.150
10	1.120	0.079	2.00	48	1536	134.36	1.52	1.016	0.078	-0.198
11	1.150	0.115	2.00	38	1536	134.36	3.24	0.852	0.043	-0.098
12	1.150	0.100	2.00	39	1536	134.36	2.44	0.884	0.053	-0.143
13	1.150	0.076	2.00	41	1536	134.36	1.39	0.951	0.070	-0.205
14	1.200	0.114	2.00	31	1536	134.36	3.16	0.781	0.053	-0.092
15	1.200	0.095	2.00	32	1536	134.36	2.19	0.813	0.062	-0.155
16	1.200	0.079	2.00	33	1536	134.36	1.52	0.844	0.069	-0.197

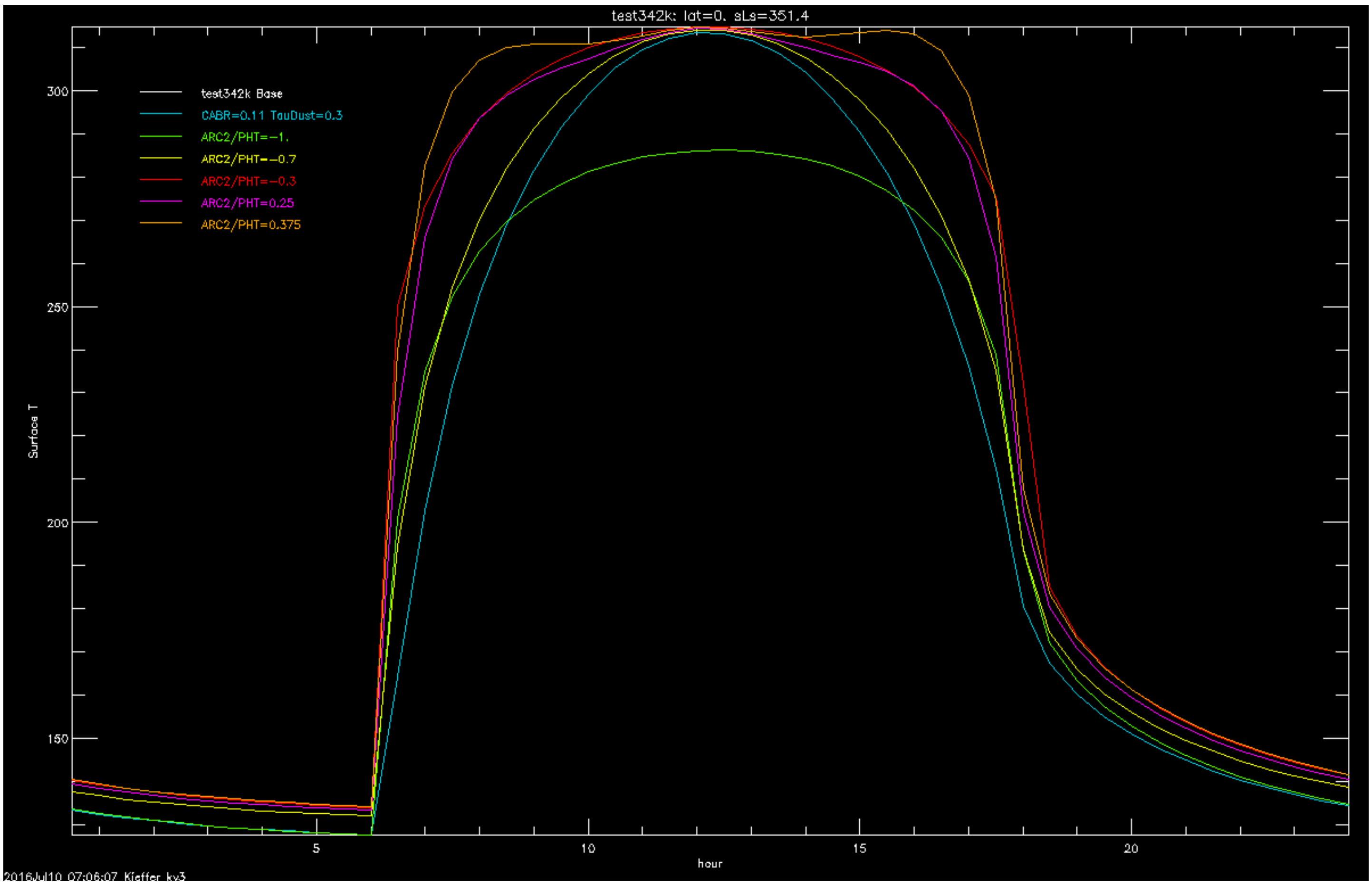

Figure 24: Surface temperature for a lunar material (I=50, no atmosphere, Mars orbit and length of day) for different photometric functions. Different curves as identified in Fig. 23. kv402.png

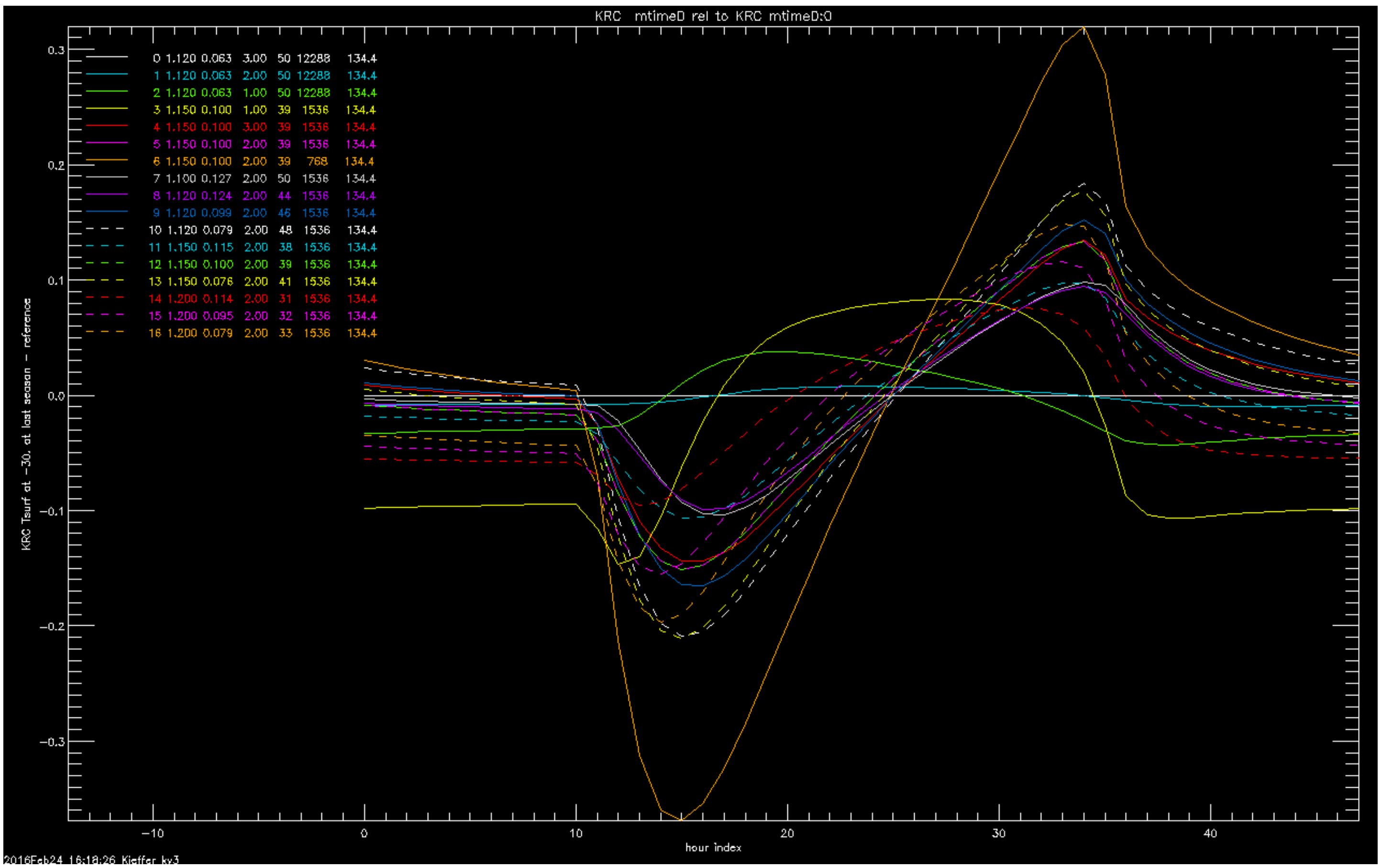

Figure 25: Results for I=300 on Mars with 40 seasons after a 2-year spin-up. Run mtimeD; 17 cases, most with 1536 times per sol, referenced to 12288 times per sol with CONVG=3. Legend columns are: 0-based case index, RLAY, FLAY, CONVG, N1, N2, bottom depth in units of D. kv782Dm30.png

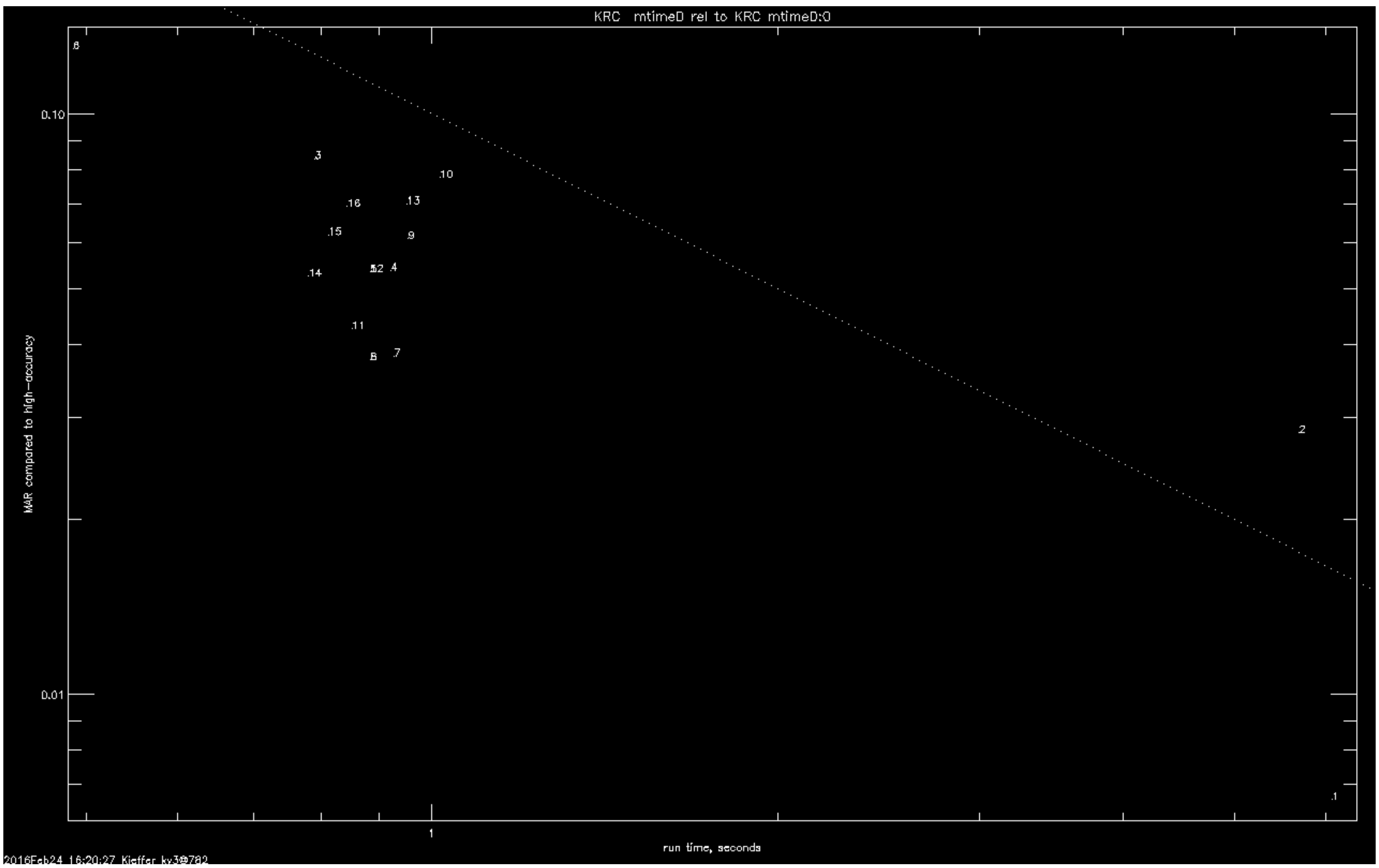

Figure 26: Efficiency results for similar homogeneous cases; abscissa is log of execution time for the case (all latitudes and seasons), ordinate is the Mean Absolute Residual (MAR) of surface temperature for 30°S relative to the reference case. kv782bDm30.png