

KRC supporting material

Hugh H. Kieffer File= /xtex/tes/krc/port.tex

July 20, 2010

Contents

1	Introduction	2
2	Binary file format	2
2.0.1	type52	3
3	Helplist	4
4	IDL Routines	18
5	Temperature-dependant conductivity	20
5.1	Cemented and un-cemented Particulates	20
5.2	Testing temperature dependant conductivity	20
5.2.1	2010feb17	25
6	Evolution log of KRC	25
7	Liens on KRC system or code	29
8	Code details	30
9	IDL interface	31
10	Development notes	31
10.0.2	2003jan23	31
10.0.3	2004jul05	32
10.0.4	2004Jul11	32
10.0.5	2006sep09	32
10.0.6	2009mar11	33
10.0.7	2009dec24	33
10.0.8	2010jan09	33
10.0.9	2010jan22	34
10.1	Mar 11 2009 puzzels	34
11	files to consider	36

1 Introduction

This document is a compilation of what had been various separate files of notes about the KRC system. The level of formality ranges widely!

Liens on documentation:

2 Binary file format

Other than direct access files, binary output files are in "bin5" format, described below. None contain the optional free-text section.

Description of the "bin5" file system
Hugh Kieffer 2010may12

The bin5 file system is designed to allow transfer of binary arrays between different languages, operating systems and hardware types. Each bin5 file contains a leading 512 byte (or multiple thereof) region that is ASCII text and describes the array which follows it.

The first few words are integers separated by spaces, these define the array size and type

The first integer is the number of dimensions to the array, N.

The next N integers are the sizes of each dimension, in the order of most-rapidly varying index first,

Then comes an integer that defines the word type; this follow the IDL convention. Values greater than 4 are not supported in Fortran as of 2002sep.

- 1 Byte
- 2 Integer
- 3 Longword integer
- 4 Floating point
- 5 Double-precision floating
- 6 Complex floating
- 7 String
- 8 Structure
- 9 Double-precision complex
- 10 Pointer
- 11 Object reference
- 12 Unsigned Integer
- 13 Unsigned Longword Integer
- 14 64-bit Integer
- 15 Unsigned 64-bit Integer

The next integer is the number of elements in the array, this is redundant with the product of the dimension sizes.

Files written after 1999march should have a final integer that indicates the number of leading ASCII bytes in the file; this is always a multiple of 512.

Next is some text bounded by << .. >>; the first part will usually be

"<<IDL_SIZE + headlen " , the latter part should be the creation date of the file, e.g., " Tue Jul 9 05:08:18 2002 >>"

This is followed by the "header", which is free text describing the file. Files from Hugh Kieffer may contain Keyword=value sections and/or embedded small arrays. Embedded arrays use a separator, such as |, #, or ^, between each element, the separator is doubled to indicate the limits of the embedded array. E.g.,
Geom=||7.|0.|7.|0.|0.|1.|| Such arrays are created/read by strum.pro

The last 5 bytes of the ASCII section should always be "C_END"; these should be immediately preceded by a 5- or 8-byte indication of the source hardware architecture, e.g. "x86 ".

It may be convenient to define an alias to look at the ASCII section:

```
alias bhead 'dd count=1 if=\\!*' # display first 512 bytes of one binary file
alias bhead2 'dd count=2 if=\\!*' # display first 1024 bytes of one binary file
```

Versions are available in IDL, Fortran (uses calls to C routines)

Example:

```
2 32 2 4 64 512 <<IDL_SIZE + headlen Tue Jul 9 05:08:18 2002 >>synmoonspec:
Multiply factor for ROLO irrads model r311f and Solar model= solar_bb. Synthetic
spectrum: Match mean and StdDev of Apollo with Breccia fraction: 0.05. Geom=
||7.|0.|7.|0.|0.|1.||
```

x86_64 C_END

This example defines a single-precision floating-point array sized (32,2)

Within the binary array after the header, sub-arrays are packed.

2.0.1 type52

For hourly conditions: is a 5-dimension array; dimensions are:

```
1: N24          number of hours output
2: 7            3 temperatures, 2 fluxes, 2 sets of 3 parameters plus layers
3: N4          number of latitudes
4: NDX+NSOUT   number of seasons output plus leading extra seasons
5: NCASE       number of cases
```

The first NDX extra "seasons" of each case contain values that hold for all latitudes:

```
Float of FRONT+KRCCOM, DJU5(ns), SUBS(ns), PZREF(ns), TAUD(ns), SUMF(ns)
```

True seasons contain for every hour and latitude: TSF, TPF, TAF, DOWNVIS, DOWNNIR +

```
6: float(NDJ4)+ DTM4 + TTA4+ Tmin(Nlay-) Omitting virtual first layer
7: FROST4+ AFRO4+ HEATMM+ Tmax(Nlay-) " " "
the Nlay- are as many layers as fit within the number of Hours
```

FRONT is 4 integers for the sizes used, converted to real:

```
1 NWKRC ! size of KRCCOM
2 IDX ! 1-based index of the dimension with extra values (4 for type 52)
```

3 NDX ! Number of those extra
4 NSOUT ! actually a spare

NDX=(4+NWKRC+5*NSOUT)/(N24*7*N4) ! extra Dim needed to hold case header

3 Helplist

A guide to setting up the input file is contained in *helplist.tex*, which is maintained as a plain text file for easy access. It is also included here.

KRC: PLANETARY SURFACE TEMPERATURES
HELPLIST.TXT 2010July19
Hugh Kieffer. Original code ~1969, many revisions.

Major changes:

2002jul12-17 Replace atmosphere with Delta-Eddington model, and atmospheric temperature based on solar and IR energy balance.

2008nov-2009feb Add capability for temperature-dependant thermal conductivity and 2010feb temperature-dependant specific heat and revision of KRCCOM.

The evolution of KRC code is contained in *evolve.txt*.

A crude diagram of the call architecture is in *flow.txt*

=====

METHOD

See the LaTeX document for a more detailed description: *tes/krc/jpap.tex*

Program is designed to compute surface and subsurface temperatures for a global set of latitudes at a full set of seasons, with enough depth to capture the annual thermal wave, and to compute seasonal condensation mass. For historic reasons, the code has substantial optimization. There are generalities that allow this code set to be used for any solid body with any spin vector, in any orbit (around any star); this is also the source of some of the complexity.

Method is explicit forward finite differences with exponentially increasing layer thickness and binary time increase with depths where allowed by stability. Depth parameter is scaled to the diurnal thermal skin depth. Initially starts at 18 hours with the mean temperature of a perfect conductor. Second degree perturbation is applied at the end (midnight) of the (third) day; this jumps the mean temperature of all layers and the lower boundary to equal the mean surface temperature.

Boundary condition treatment:

- Perturbation solution of quartic equation at surface for each iteration;
temperature gradient assumed uniform in top interval.
- Lower boundary may be insulating or constant-temperature.

Atmospheric Radiation:

KRC uses a one-layer atmosphere that is grey in both the solar and infra-red

regions. parametric atmosphere. The default atmospheric parameters are based on estimates of Mars' gas and aerosol properties.

Delta-Eddington model for insolation; direct onto sloped surface and diffuse, with possible twilight extension.

Atmosphere temperature based on Delta-Eddington solar absorption and IR opacity [Pre 2002jul16

First-order treatment of scattering of solar radiation.

Diurnal temperature is modeled as sinusoidal with phase shift.]

Keplerian orbital motion; seasons are at uniform increments of time. Mean orbital elements are pre-calculated for any epoch (all planets and several comets) by the PORB code set.

Units are SI; days for orbital motion. (Revised from cal-cgs, 97july)

Options:

Different Physical properties below a set layer (IC).

Regional slope

Three ways to handle seasonal global pressure variation

Atmosphere condensation:

Global integral of CO2 frost-gas budget can control surface pressure.

Allows different surface elevation for each latitude zone.

Zonal frost saturation temperature tracks local surface pressure.

Option for cap albedo to depend upon mean daily insolation.

CONVERGENCE NOTES

Convergence prediction routine can't jump more than one time constant ($\text{TAU} = X^{**2}/2$) for the total thickness. Therefore, if $X(N1)$ is small, make DDT smaller than usual. If DELJUL is much smaller than $(X(N1))^{**2}/2$, then DDT can be as large as 0.3. Otherwise DDT must be about 0 for the prediction routine to work well (it assumes the 3rd derivative to be 0).

- - - - - INPUT FILE - - - - -

All parameters for KRC are set by a formatted text file. An example is master.inp, which has default values for a 19 latitude set for a run of three martian years, with the last output to disk. Parameter values are listed below their titles, which are in many cases identical to the code name, and last character of the title is above the last location in the field. Thus, integer values MUST be aligned. Titles with a leading "[" indicate that the value is not used. The recommended procedure is to copy master.inp and edit only the values you wish to change. The number of lines of Latitudes and Elevations must match the value of N4, e.g., 2 lines for N4=11:20, entries beyond the N4 position may be left blank or contain the end of the line. The 13 lines following Elevations are a geometry matrix for Mars orientation and orbit in 2005, and should not be touched; they can be replaced by running PORBMN carefully.

The titles lines are skipped, so that you may put comments there carefully

The first input line is always KOLD,KEEP (I*), which sets file usage.

These are described near the end of this help file under DISK BINARY FILES.

If and only if there is a third non-zero integer, then will read next card as 6 debug flags, IDB1 to IDB6, which are normally zero

The next (normally second) line is free text where you can outline the purpose of your run.

If KOLD=0, then a full set of input values is read.

Change lines may follow immediately after the geometry matrix (see PARAMETER CHANGES section below). The end of definition of a "case" is indicated by a "0/" line. Two successive "0/" lines ends the run.

Items with numbers inset 2 spaces below are computed, not input. The source code for 'krccom.inc' indicates which subroutine sets many of the parameters; as the routine name in lowercase just below the parameter name.

Type 4 Title (20A4) 80 characters of anything to appear at top of each page.

Type 1 Real parameters (8F10.2) =====

Surface Properties

- 1 ALB Surface albedo
- 2 EMIS Surface emissivity
- 3 SKRC Surface thermal inertia [$J m^{-2} s^{-1/2} K^{-1}$] { cal cm * 4.184e4}
- 4 COND2 Lower material conductivity (IC>0)
- 5 DENS2 Lower material density (IC>0)
- 6 PERIOD Length of solar day in days (of 86400 seconds)
- 7 SPHT Surface specific heat [$J/(kg K)$] {cal/(g K) * 4184.}
- 8 DENS Surface density [kg/m^3] {g/cubic cm. *10}

Atmospheric Properties

- 9 CABR Atmospheric infrared back radiation coefficient
2002jul16 IR opacity of dust-free atmosphere
- 10 AMW Molecular weight of the atmosphere
- 11 [ABRPHA UNUSED [Phase of ABRAMP, degrees relative to midnight]
- 12 PTOTAL Global annual mean surface pressure at 0 elev., Pascal[=.01mb]
- 13 FANON Mass-fraction of mean atmosphere that is non-condensing
- 14 TATM Atm temp for scale-height calculations

- 15 TDEEP Fixed bottom temperature. Used if IB>=1.
16 SPHT2 Lower material specific heat (IC>0)

Dust & Slope Properties

- 17 TAUD Mean visible opacity of dust, solar wavelengths
- 18 DUSTA Single scattering albedo of dust
- 19 TAURAT Ratio of thermal to visible opacity of dust
- 20 TWILI Twilight extension angle [deg]
- 21 ARC2 Henyey-Greenstein asymmetry factor
moon = eclipse start time in local Hours
- 22 [ARC3 NOT USED coeff. for planetary heating
moon = eclipse duration in seconds 0=no eclipse
- 23 SLOPE Ground slope, degrees dip. Only pit may slope beyond pole.
- 24 SLOAZI Slope azimuth, degrees east from north. <-360 is a pit

Frost Properties

- 25 TFROST Minimum Frost saturation temperature
may be overridden by local saturation temperature (LVFT)
- 26 CFROST Frost latent heat [J/kg] {cal/gm*4184. [Not used if
- 27 AFROST Frost albedo, may be overridden (LVFA) [TFROST never
- 28 FEMIS Frost emissivity [reached
- 29 AF1 constant term in linear relation of albedo to solar flux

```

30 AF2 linear term in relation of albedo to solar flux units=1/flux
    Afrost = AF1 + AF2 * <cos incidence> SOLCON / DAU^2
31 FROEXT Frost required for unity scattering attenuation coeff. [kg/m^2]
    the greater of this and 0.01 is always used.
32 fd32    UNUSED
-----
    Thermal Solution Parameters
33 RLAY Layer thickness ratio
34 FLAY First layer thickness (in skin depths)
35 CONVF Safety factor for classical numerical convergence
    0 for no binary time division of lower layers
    >0.8 for binary time division. Larger is more conservative
36 DEPTH Total model depth (scaled) (overrides FLAY if not 0.)
37 DRSET Perturbation factor in jump convergence. If = 0., then
    all layers reset to same average as surface layer. Else,
        does quadratic curve between surface and bottom averages
38 DDT Convergence limit of temperature RMS 2nd differences
39 GGT Surface boundary condition iteration test on temperature
40 DTMAX Convergence test: RMS layer T changes in a day
-----
    Orbit Geometry & Constants
41 DJUL Starting Julian date of run -2440000(N5>0)
42 DELJUL Increment between seasons in Julian days (if N5>1)
43 SDEC Solar declination in degrees. (if N5=0)
44 DAU Distance from Sun in astronomical units (if N5=0)
45 SUBS Aerocentric longitude of Sun, in degrees. For printout
    only. Computed from date unless N5=0(for printout only)
46 SOLCON Solar constant Applied Optics 1977 v.16, p.2693: 1367.9 W/m^2
    1366.2 Based on figure in Frohlich, Observations of
    irradiance variations, Space Sci. Rev.,94,15-24,2000
47 GRAV Surface gravity. MKS-units
48 AtmCp Specific heat at constant pressure of the atmosphere [J/kg/K]
-----
    Temperature dependent conductivity. Ignored unless LKOFT set.
49 ConUp0 Constant coef for upper material
50 ConUp1 Linear in  $k=c_0+c_1x+c_2x^2+c_3x^3$  where  $x=(T-220)*0.01$ 
51 ConUp2 Quadratic "
52 ConUp3 Cubic coeff. "
53 ConLo0 Constant coef for lower material
54 ConLo1 Linear as for ConUp above
55 ConLo2 Quadratic "
56 ConLo3 Cubic coeff. "
    Temperature dependent specific heat. Ignored unless LKOFT set.
57 SphUp0 Constant coef for upper material
58 SphUp1 Linear in  $k=c_0+c_1x+c_2x^2+c_3x^3$  where  $x=(T-220)*0.01$ 
59 SphUp2 Quadratic "
60 SphUp3 Cubic coeff. "
61 SphLo0 Constant coef for lower material
62 SphLo1 Linear as for SphUp above
63 SphLo2 Quadratic "
64 SphLo3 Cubic coeff. "
-----
COMPUTED REAL*4 VALUE
65 HUGE = 3.3E38 nearly largest REAL*4 value
66 TINY = 2.0E-38 nearly smallest REAL*4 value
67 EXPMIN = 86.80 neg exponent that would almost cause underflow

```

```

68   fd60(2) Spare
69
70   RGAS = 8.3145  ideal gas constant  (MKS=J/mol/K)
71   TATMIN  Atmosphere saturation temperature
72   PRES    Local surface pressure at current season
73   OPACITY Solar opacity for current elevation and season
74   TAUIR   current thermal opacity at the zenith
75   TAUEFF  effective current thermal opacity
76   TATMJ   One-layer atmosphere temperature
77   SKYFAC  fraction of upper hemisphere that is sky
78   TFNOW  frost condensation temperature at current latitude
79   AFNOW  frost albedo  at current latitude
80   PZREF  Current surface pressure at 0 elevation, [Pascal]
81   SUMF   Global average columnar mass of frost [MKS]
82   TEQUIL Equilibrium temperature ( no diurnal variation)
83   TBLow  Numerical limit (Blowup) temperature
84   HOURO  Output Hour requested for "one-point" model
85   SCALEH Atmospheric scale height
86   BETA   Atmospheric IR absorption
87   DJU5   Current Julian date (offset 2440000 ala PORB convention)
88   DAM    Half length of daylight in degrees
89   EFROST Frost on the ground at current latitude [kg/m^2] {g/cm^2 * 10.}
90   DLAT   Current latitude
91   COND   Top material Thermal conductivity (for printout only)
92   DIFFU  Top material Thermal diffusivity (for printout only)
93   SCALE  Top material Diurnal skin depth (for printout only)
94   PI    pi
95   SIGSB  Stephan-Boltzman constant (set in KRC)
96   RAD   Degrees/radian

```

Type 2 Integer Parameters (8I10) =====

```

1  N1 # layers (including fake first layer) (lim MAXN1)
2  N2 # 'times' per day (lim MAXN2). Must be an even number,
    should be a multiple of N24 and NMHA.
3  N3 Maximum # days to iterate for solution (lim MAXN3)
    98sep03 This can be 1, but then must use DELJUL ~= PERIOD
    If N3 lt 3, first day starts on midnight. else at 18H
4  N4 # latitudes (lim MAXN4=19). Global integrations done for N4>8
5  N5 # 'seasons' total for this run. If 0, then DAU and SDEC will be
    used as entered for a single season.
6  N24 # 'hours' per day stored, should be divisor of N2 (lim MAXNH)
7  IB Bottom control: 0=insulating, 1=constant temperature
    2=start all layers =TDEEP & constant temperature
8  IC First layer (remember that 1 is air) of changed properties.
    0 or 999=homogeneous, or 3 to N1-2
-----
9  NRSET # days before reset of lower layers; >N3=no reset
10 NMHA # 'hour angles' per day for printout (no limit)
11 NRUN Run #; appears in some printout
12 JDISK Season count that disk output is to begin
13 IDOWN Season at which to read change cards
14 I14 Index in FD of flexible print
15 I15 ""
16 KPREF Mean global pressure control. 0=constant
1= follows Viking Lander curve  2=reduced by global frost, but

```


then N4 must be >8, and latitudes must be monotonic increasing
and must include both polar regions (no warning for your failure)

17 K4OUT Disk output control: See details in DISK BINARY FILES section

Three modes of direct access Fortran files; one case per file.

--KRCCOM(once), then TSF & TPF;

0=KRCCOM,LATCOM each season

1:49=KRCCOM,DAYCOM for the last latitude; each season

Modes of bin5 file for multiple cases

51=(Hours, 2 min/max, lat, seasons, cases)

52=(hours, 7 items, lat, seasons, cases)

54=[many seasons, 5 items,lats, cases]

55=[many seasons,9 items, cases]

56=[packed T hour and depth, latitude,season,case]

18 JBARE J5 season count at end of which to set frost amount to 0. 0=never

19 NMOD Spacing of season for notification. minimum of 1

20 IDISK2 NOPE Season count at which to reset deep layer temperatures

COMPUTED I*4 VALUES

21 KOLD Season index for reading starting conditions

22 id22(6) 22 and 23 used as flags for season-variable ALB and TAUD

28 NFD Number of real items read in

39 NID Number of integer items read in

30 NLD Number of logical items read in

31 N1M1 Temperature vrs depth printout limit (N1-1)

32 NLW Temperature vrs depth printout increment

33 JJO Index of starting time of first day

34 KKK Total # separately timed layers

35 N1PIB N1+IB Used to control reset of lowest layer

36 NCASE Count of input parameter sets in one run

37 J2 Index of current time of day

38 J3 Index of current day of iteration

39 J4 Index of current latitude

40 J5 Index of current "season"

Type 3 Logical Parameters (10L7) =====

1 LP1 Print program description. TPRINT(1)

2 LP2 Print all parameters and change cards (2)

3 LP3 Print hourly conditions on last day (3)

4 LP4 Print daily convergence summary (4)

5 LP5 Print latitude summary (5)

6 LP6 Print TMIN and TMAX versus latitude and layer (6)

7 LPGLOB Print global parameters each season

8 LVFA Use variable frost albedo. Uses AF1 & AF2 (real # 29,30)

9 LVFT Use variable frost temperatures

10 LKOFT Use temperature-dependent conductivity and specific heat

11 LPORB Call PORB1 just after full input set

12 LKEY Read change item from terminal after main input set

13 LSC Read change cards from input file at start of each season

14 LNOTIF spare

15 LOCAL Use each layer for scaling depth

16 LD16 Print hourly table to FORT.76 [TLATS]

17 LPTAVE Print <T>-<TSUR> at midnight for each layer [TDAY]

```

18 LD18 Output to fort.78 [TLATS] insolation and atm.rad.coefficients
19 LD19 Output to fort.79 [TLATS] insolation and atm.rad. arrays
20 LONE (Computed) Set TRUE if KRC is in the "one-point" mode
-----
-----

```

followed in 'krccom' by:

```

[real*4] TITLE(20) 80-character title
[real*4] DAYTIM(5) 20-character run date and time
=====

```

Latitude(s) (10F7.2) N4 latitudes in degrees, no internal separations.
 Latitudes to be in order; south to north. [[If last latitude is
 .LE. 0, will assume symmetric results for global integrations]]

Elevation(s) (10F7.2) N4 values in Km corresponding to latitudes

Orbital Parameters (LPORB=T) Format identical to that produced by PORB
 program set ASCII file output. So these can be directly pasted with an
 editor. see PORBCM.INC

PARAMETER CHANGES

Fortran List Directed. Change the values in KRCCOM
 White-separated, a "/" terminates the read and leaves remaining values unchanged
 The 4 items are: Integer Integer Numeric_value 'Text' / Comment
 1: Type (integer) see table below
 2: Index in array (integer), as listed in table above
 3: New value, numeric, will read as real and convert. 0.=false.
 4: Reason, text string within single quotes
 [after a / (forward slash) nothing is read, so you can use for comments]

The print file will list each change as read, followed by the title of the
 changed item. It is a good idea to look at this print to be sure you changed
 what you intended.

Type	Meaning	Valid Index
0	End of Current Changes	
1	Real Parameter	1-56
2	Integer Parameter	1-20
3	Logical Parameter	1-20
4	New Latitude Card(s) Follow	
5	New Elevation Card(s) Follow	
6	New Orbital Parm Cards Follow (LPORB Must be True)	
7	Text becomes new Title	
8	Text becomes new disk or season-variation file name	
	if index=22, read variable ALBEDO	
	if index=23, read variable TAUD	
9	Complete new set of input follows	
10	Text becomes new One-Point input file name	
11	This is a set of parameters for "one-point" model	
	For this type, 9 values must appear in a rigid format	
12	Set of 2*4 coefficients for T-dep conductivity. List-directed IO	

To start variable albedo, use input card:

8 22 0 'AlbedoFileName' / Variable albedo text file name
Can revert to constant albedo by hokey technique of using a bad name. E.g.,
8 22 0 'badName' / turn variable albedo off
Files of text table of value versus season will be read at the start of a
run. These will apply to ALL latitudes. See example valb1.tab

Variable Tau done the same way, with 22 being replaced with 23

COMMON /LATCOM/ see latcom.inc
COMMON /DAYCOM/ see daycom.inc

Because the binding routines to IDL are intolerant of any errors, the items in
the above commons have not been changed, Rather, in 2004July, and additional
common was added as a "catch-all" for any new items.

COMMON /HATCOM/ see hatcom.inc

Error Returns:

"Parameter error in TDAY(1)" : Convergence factor < .8 classic.
Instability anticipated.
"UNSTABLE; Layer..... TDAY(1):

DRSET: 0=>Reset by delta_average_T for each layer:
else: reset by {linear + DRSET*quadratic}*{<surf>-<botm>}
TDAY: LRESET Reset midnight T's for all but top layer.
LDAY Last day computations

----- Handy things -----

The first "hour" in printout and output arrays is 1/24 (strictly, 1/N24) of a
sol after midnight. E.g., the last time is midnight, not the first.

Atmospheric scale height, SCALEH, depends upon physical constants and
TATMAVE which (2007nov) is always = TATM, input. and GRAV , input

----- DISK BINARY FILES -----

The routine TDISK is used to read or write direct-access binary files or bin5
files. The first season to write is specified by JDISK, all following seasons
will go to the same file. For direct-access files, each file record consists of
KRCCOM plus LATCOM or KRCCOM plus DAYCOM.

Disk output is largely controlled by the KRC and TSEAS routines.

- - - Items which control file I/O - - - -

KOLD & KEEP on first input line

KOLD: 0= input card set follows; else=disk record number to start from,
then will read any change cards.

If LPORB in old file was True, then there must be a PORB card set
as the set of lines following the KEEP,KOLD line

KEEP: 0= close disk file after reading seasonal record KOLD;
>0= value of JJJJJ at which to start saving seasons in same disk
file [overrides JDISK].

To start from a prior seasonal run, need to determine the record

corresponding to the desired season;
KOLD=J5_target - JDISK(old) ; >0
set KEEP=1, change card J5=number of new seasons, set K4OUT.

JDISK sets the first season to save results

N5 sets the last season to run

K4OUT sets the record content:

- Will output first record of KRCCOM,ALAT,ELEV, then records of TSF & TPF
- 0 (normal) Will output records of KRCCOM+LATCOM
- +n<=50 Will output records of KRCCOM+DAYCOM for the last computed latitude.

> 50 Will write custom bin5 file at the end of a run, with dimensionality from 3 to 5 (more possible). All 5x outputs allow multiple cases, each with a "prefix" for each case consisting of with 4 size integers (converted to Float) followed by KRCCOM; after this may come vectors of parameters versus season. The next-to-last dimension is increased to allow room for the prefix to be embedded in the bin5 array. KRC input items that would change any of the bin5 dimensions are not allowed to change between cases. Each dimension is adjusted to the necessary size. Each case has the same structure; this simplifies coding although some items are then present redundantly. The number of cases allowed is set by the size of one case, and printed as MASE at the end of the first case in the print output. Cases beyond the maximum that can be stored will be executed, but not saved.

The first 4 words of the prefix, and of thus of the bin5 array, are:

- (1)=FLOAT(NWKRC) ! Number of words in KRCCOM
- (2)=FLOAT(IDX) ! 1-based index of dimension with extra values
- (3)=FLOAT(NDX) ! Number of those extra
- (4)=FLOAT(NSOUT) ! [Available of other use]

51=(N24 hours, 2: TSF TPF, N4 lats, NDX+ seasons, cases)

The prefix section contains: sub_array(seasons,5)(0-based index)

0)=DJU5 1)=SUBS 2)=PZREF 3)=TAUD 4)=SUMF

52=(N24 hours, 7 items, N4 lats, NDX+ seasons, cases)

The 7 items are: 1)=TSF 2)=TPF 3)=TAF 4)=DOWNVIS 5)=DOWNNIR

6) packed with [NDJ4,DTM4,TTA4, followed by TIN(2+

7) packed with [FROST4,AFRO4,HEATMM, followed by TAX(2+

The number of layers for TIN and TAX is the smaller of: the number computed and that fit here.

The prefix is identical to Type 51

54= (seasons, 5 items, NDX +nlat, cases)

Items are (0-based index):

0= TSF=surface temperature at 1 am, 1= TSF at 13 hours,

2= HEATMM=heat flow, 3= FROST4=frost amount,

4= TTb4 = predicted mean bottom temperature

The prefix contains DJU5

55= (seasons,NDX+ items,cases). For seasonal studies at one latitude
ITEMS intended to be recoded as needed. Initial version is 9 items:

[Tsur@ 1am,3am,1pm, spare, Tplan @1am,1pm, Surface heat flow,
frost budget, T_bottom]

The prefix contains DJU5
Can hold very large number of seasons and cases.
THIS MODE DOES NOT SUPPORT CONTINUATION RUNS

56= [vectors&items, latitudes, NDX+ seasons, cases]
The first dimension is: TSF for all hours, TPF at all hours,
T4 for all layers at midnight, then FROST4,HEATMM,TTA4
The prefix is identical to Type 51

Once a disk file is opened, any records written will go into that file until a new filename is specified (Type 8 Change line), which closes the current file.

To run & save various cases for a single season, set N5 and JDISK to 1.

To extract a detailed day by saving DAYCOM to disk, set JDISK=N5, set a new file name, and set K4OUT to desired latitude index (normally 1):

To run continuously with output every K ((1-3) days, set DELJUL=K*PERIOD
this will force prediction terms to near 0.
setting N3=1 will turn off all prediction.
set GGT large (to avoid iteration for convergence)
set NRSET=999 (to avoid reset of layers)

To continue run with new parameters (e.g., DELJUL)
3 21 1 'flag set to continue'

Note: changing DELJUL will cause reset of DJUL
Must increase the value of N5: 2 5 bigger 'Increase stopping season'
Reset will not occur because J5 continues incrementing

ASCII Output Files

krc.prt general; results, what is output is controlled by LP1:6 & LPGLOB

fort.76

```
tlats.f: mimic Mike Mellon ASCII files
      if (ld16) then
        write(76,761)subs,dlat,alb,skrc,taud,pres
761   format(/,'      Ls      Lt      A      I      TauD      P'
762   format(f7.2,f9.3,f8.3,f9.3)
        write(76,762)qh,tsfh(i),adgr(j),qs

        do i=1,n24
          j=(i*n2)/n24
          qh=i*qhs
          qs=(1.-alb)*asol(j) ! absorbed insolation
          write(76,762)qh,tsfh(i),adgr(j),qs
        enddo
```

fort.78

```
tlats.f: for average and maximum:
      if (ld18) write(78,*)cosi_(i), t_(i),ADGs(i),ADGP(i)
      if (ld18) write(78,*)j5,j4,sol,ave_a,adgir,c52,beta
```

fort.79

```
tlats.f: for each time-step
```

```

      if (ld19) write(79,*)adgr(jj),qa,direct,diffuse
col 1 = downgoing thermal radiation
col 2 = total insolation reaching surface
col 3 = direct fraction of insolation
col 4 = diffuse fraction of insolation

```

----- To run two material types (2000jan23)

Set IC to the first layer to have the lower material properties (>= 3)

Set COND2 to the lower material conductivity

Set DENS2 to the lower material density

Set SPHT2 to the lower material specific heat

If LOCAL is False, then initial setting of all layer thicknesses is based upon the scale of the upper material; if it is set True, the thickness of the lower layers is set by their scale.

TDAY no longer allows unstable (thin) layers, and will increase the thickness of the layer IC to satisfy the convergence safety factor FCONV if needed.

However, the code to check on convergence was retained.

----- Setting temperature-dependant properties

Basic Flag is L10=LKOFT . If this is true, then the 8 input parameters ConUp0 to ConLo3 must be set to yield thermal conductivity as a function of temperature for the upper and lower materials. $k=c_0 + c_1x + c_2x^2 + c_3x^3$ where $x=(T-200.)*0.01$

Correspondingly, the 8 input parameters SphUp0 to SphLo3 must be set for specific heat

One way to generate the coefficients is to run for each of the upper and lower materials the IDL procedure KOFTOP, which can call all of the temperature-dependant routines. KOFTOP allows change of its parameters, including grain radius and pressure, and will print the required parameters ready for input to KRC.

Below are sample coefficients based on Sylvain Piqueux's numerical model for un-cemented soils; the fit error is <0.1% over 120-320K. Left column is grain radius in micrometers, then the four normalized coefficients ready for inclusion in a KRC input file, followed by the thermal inertia at 220K for nominal density and specific heat.

R(mu)	c0	c1	c2	c3	Iner
10.	0.008274	0.000735	-0.000376	0.000148	89.8
20.	0.012379	0.001280	-0.000629	0.000250	109.9
50.	0.021485	0.002647	-0.001201	0.000483	144.7
100.	0.032051	0.004528	-0.001874	0.000761	176.8
200.	0.046023	0.007569	-0.002743	0.001129	211.8
500.	0.068387	0.014075	-0.003874	0.001687	258.2
1000.	0.086303	0.021288	-0.004146	0.002099	290.1
2000.	0.103743	0.030909	-0.003141	0.002535	318.0
5000.	0.127172	0.049907	0.002019	0.003469	352.1
10000.	0.149810	0.074734	0.011546	0.004939	382.2
20000.	0.185706	0.119913	0.030938	0.007877	425.5
50000.	0.283361	0.250283	0.089327	0.016714	525.6

RUNNING THE "ONE-POINT" MODE (2002mar08)

A parameter initialization file `Mone.inp` is provided. It sets the KRC system into a reasonable mode for one-point calculations. Do not change that file unless you have read this entire file.

A line near the end of that file points to a file `'one.inp'` which can contain any number of one-point conditions. `'one.inp'` is intended to be edited to contain the cases you want; however, it must maintain the input format of the sample file.

First Line is any title you wish. It must be present.
The second line is an alignment guide for the location lines. It must be there.

Each following line must start with an `'11 '`; this is a code that tells the full-up KRC that is a one-point line. The next 9 fields are read with a fixed format, and each item should be aligned with the last character of the Column title. All items must be present, each line must extend at least to the `m` in Azim; comments may extend beyond that, but they will not appear in the output file. Be sure to have a `<CR>` at the end of the last input line.

The fields (after the 11) in the one-point input are:

Ls `L_sub_S` season, in degrees
Lat Aerographic latitude in degrees
Hour Local time, in 1/24'ths of a Martian Day
Elev Surface elevation (relative to a mean surface Geoid), in Km
Alb Bolometric Albedo, dimensionless
Inerti Thermal Inertia, in SI units
Opac Atmospheric dust opacity in the Solar wavelength region
Slop_ Regional slope, in degrees from horizontal
Azim Azimuth of the down-slope direction, Degrees East of North.

The two additional columns in the output file are:

TkSur Surface kinetic temperature
TbPla Planetary bolometric brightness temperature

Try running the binary file first. If that fails, a Makefile is provided to compile and link the program; simply enter `"make krc"` and pray. If this fails, have your local guru look over the Makefile for local dependencies. Suggestions of making the Makefile more universal are welcome.

To run the program, change to the directory where the program was built, and enter `"krc"`. You should get a prompt:

```
?* Input file name or / for default =  
Mone.inp
```

If the initialization file still has this name and is in the same directory, enter a single `"/"` and `<CR>`. Otherwise, enter the full pathname to the initialization file, with no quotes and no blanks.

A second prompt is for the name of the output file:

```
?* Print file name or / for default =  
krc.prt
```

Again, if this is satisfactory, simply enter / <CR> , else enter the desired file path-name.

----- Comments on the One-point model.

The initialization file of 2002mar08 is set to compute the temperatures at the season requested without seasonal memory. It uses layers that extend to 5 diurnal skin depths. It does not treat the seasonal frost properly, so don't believe the results near the edge of the polar cap. Execution time on a circa 2001 PC may be the order of 0.01 seconds per case.

The underlying model is the full version of KRC. By modifying the initialization file, you can compute almost anything you might want. If you choose to try this, best to read all of this document.

Reading type 5x files

Routines do not access files directly unless specifically listed.

DEFINEKRC Define structures in IDL that correspond for Fortran commons
Calls: None other than IDL
Firm code of common definitions. Must be recoded if a Fortran *.inc changes

READKRCCOM Read a KRCCOM structure from a bin5 file
uses 3-element HOLD array. Returns a structure of krccom
Options to open or close bin5 file or read one case
Calls: DEFINEKRC
Files: bin5
HOLD is: 0]=logical unit 1]=number of words in a case 2]=# cases in the file

MAKEKRCVAL Make string of selected KRC inputs: Key=val
Calls: DEFINEKRC

KRCHANGE Find changes in KRC input values in common KRCCOM
Calls: READKRCCOM MAKEKRCVAL
Reads and stores krccom for first case. For each additional case, makes a list of any changes in the float, integer or logical input values.

KRCCOMLAB Print KRC common input items
all items via arguments
Calls: None

KRCLAYER Compute center depth of KRC layers
all items via arguments
Calls: none

KRCCOMLAB Print KRC common input items
all items via arguments
Calls: None

KRCSIZES Compute array and common sizes for KRC Fortran
Test procedure to compute array sizes or hours.
Must recode if any size in *.inc changes

A listing of all commons can be generated by these Linux commands:
cd /home/hkieffer/krc/src [replace top part of path with local installation]
rm allinc.txt
cat krccom.inc latcom.inc daycom.inc hatcom.inc filcom.inc units.inc porbcm.inc > allinc.txt

=====
Notes on how some aspects of the code work:

>> New file name:

TCARD reads a card of Type 8, (and index is not 22 or 23)
it calls TDISK(4,0), which closes current file and sets LOPN2=.FALSE.
TCARD then moves new file name into common
KRC checks if current (new) values of N5 and JDISK call for file output;
with LOPN2=.FALSE., KRC calls CALL TDISK (1,0) to open new file.

>> End of a case and end of a run:

TCARD sets KOUNT=0 at entry; this is incremented for every card except those of
type 0 (or less) or type 11 (one-point mode). When type 0 is encountered, if
KOUNT is positive, does normal check of changes before return with IR=1 to
indicate start of a new case; if KOUNT is zero, returns with IR=5 and prints
'END OF DATA ON INPUT UNIT'

>> Setting one-point mode.

This can be done only in the first case, and there is no way to leave the
one-point mode except to end the run.

TCARD encounters: " 10 * filename" as change card in the initial case.

sets this as new input file name, then returns with IRET=4
[Thus, nothing following this change card in initial file is read]
KRC closes prior input file, opens the new one, and reads past first two lines
then calls TCARD to read first one-point line and sets LONE=true
and drops into the top of the "case" loop.

The master one-point should have a single latitude, no binary output file.
The small number of layers, days to converge, and seasons ignores the seasonal
effect.

One-point request values are read by TCARD @ 310, which computes starting DJUL

>> Starting conditions and date

Initial N5-JDISK sets the size of output files. There could be any number of
interior seasons where parameter changes are made; based on successive values of
IDOWN.

KRC initially calls TCARD(1

For each case loop, sets IQ=TCARD_return. If one-point mode, sets IQ=1

TSEAS uses IQ as key. If this is 1, then sets J5=0 and sets DJU5 to season -1.
else, increments J5 and increments DJU5 with current DELDUL. This allows use of
variable resolution dates. (so J5 never 0 when TCARD(2 called)
If J5 equals IDISK2, then TSEAS calls TCARD(2 to read changes, and proceeds to
next season.

TLATS uses J5 as the key; if it is <= 1, then starts from equilibrium

conditions, else uses predictions from prior season

The default is that change cards cause a fresh calculation of starting conditions. Exceptions are when J5=IDOWN>0 at TCARD entry

>> Changing parameters within a seasonal run = Continue from memory.

When J5 reaches IDOWN, TSEAS calls TCARD, which will set IRET=3 before reading the new parameters. May change DELJUL to get finer seasonal resolution, but must NOT change N5

Use: Normal restrictions for what may not change for Type 5x files apply. E.g., type 56 must NOT change number of latitudes nor total number of seasons.

Set N5 to be the total number of seasons desired, including those after any number of parameter changes; it must NOT be changed later.

Set IDOWN to the season at the beginning of which wish to (first) change parameters. The next set of changes could include a revised (larger) IDOWN.

>> Use of common PORBCM

Contents are described in porbcm.inc, PORBCM is filled by TCARD calling PORB0, which reads the first 60 items in 5G15.7 from the input file and sets the value of PI. KRC references porbcm.inc but does not use it. TSEAS uses a few items to calculate LsubS. TYEAR uses the value for length of year.

4 IDL Routines

Input: idlRoutines

IDL routines 2010jul19

A number of IDL routines have been written to interface with the KRC system. The following are KRC-specific:

```
definekrc # Define structures in IDL that correspond for Fortran commons.
           Calls BYTEPAD
delcase   # Show delta between arrays changing only last index. Calls HISTFAST
lookrc    # Read any type 5x KRC bin5 models; look at change between cases.
           Calls READKRC5*
krccomlab # Print KRC common input items. Calls 0
krchange  # Find changes in KRC input values in common KRCCOM.
           Calls READKRCCOM MAKEKRCVAL
krclayer  # Compute center depth of KRC layers. Calls 0
makekrcval # Make string of selected KRC inputs: Key=val. Calls DELASTO
readkrc52 # Read RKC type 52 or 51 bin5 file; post 2004jul21. Calls BIN5
readkrc54 # Read KRC type 54 or 55 bin5 file. Calls BIN5
readkrc56 # Read KRC type 56 bin5 file. Calls BIN5
readkrccom # Read a KRCCOM structure from a bin5 file. Calls DEFINEKRC
tstp2ta   # Convert T_surf and T_plan to T_atm. Calls 0
when2start # Calc starting date for KRC to reach Ls on specific season step.
           Calls 0
```

Related to temperature-dependent properties:

```
debyefunc # Debye Function to be called for integration
koftop    # Specific heat, thermal conductivity. minerals, particulate soils
koft      # Thermal conductivity of minerals
molewt    # Molecular weights based upon common mix of isotope
partcond  # Particulate conductivity for cemented materials
sp09a     # Sylvain Piqueux unconsolidated particulate thermal conductivity
specheat  # Specific heat capacity of minerals
```

The following are utility routines called directly or indirectly by the KRC routines:

```
bin5.pro # Write/Read numeric binary files with 'standard' header. Calls 0
bytepad  # Create a Byte version of a string, padded with trailing blanks.
  Calls 0
chart    # Strip-chart plot of several variables. Calls PSYMLINE
clot     # Color plot of related curves
color24bit # Generate 256 longwords to emulate nice 8-bit color table. Calls 0
curveguide # Put guide to curves or symbols onto a plot
delast0  # Delete trailing 0's past the decimal point. Calls 0
getp     # Modify single numeric value; with prompt and limit tests. Calls 0
getpan   # Modify any elements of numeric array, with prompt and limit tests.
  Calls 0
getpsn   # Interactive input any elements of a string array, with prompt. Calls 0
graph    # Interface to graphics devices. Calls SETCOLOR
histfast # Robust, easy histogram plot, with statistics, opt row weights.
  Calls MEAN_STD SUBTITLE
kon91    # Common minimal functionality in the kon case statement.
  Calls GETPINTS GRAPH MAKE99 SETCOLOR TV2JPG TV2LP
label_curve # Place an oriented label on a curve . Calls RNDEX RTERP1
locate   # Find lower index of interval in ordered vector containing x. Calls 0
tes      # Convert Martian season L_s <-> Julian day. Calls 0
make99   # Make/print list of user options for a program. Calls 0
mean_std # Mean and standard deviation of a vector. Calls 0
psymline # Hughs convention for transferring PSYM and LINESTYLE in one. Calls 0
rndex    # Finds floating-point index of within a monotonic array. Calls LOCATE
rterp1   # real interpolation in a vector. Calls 0
setcolor # Set or modify colors, lines, plot-symbols, #plots/page.
  Calls COLOR24BIT GETP GETPAN TOOTHB STO
st0      # Make minimal string for numbers, or string arrays. Calls DELASTO
strword1 # Extract first word from a string or strarr. Calls 0
toothb   # Add a toothed color scale-bar to a window or TVPLEX panel.
  Calls GETPAN
```

The IDL program `lookrc` contains code to read all type 5x files and compare multiple cases. Although there is a lot of speciality code, all functions are isolated in elements of a large case statement. One could extract parts of the code to start your own routines.

```
.rnew lookrc
@ 11, edit file path; the default extension will be appended for each type
@ 5x, where x is 1,2,4,5,6 to read the file
@ 40, will list a guide to what was read
```

End of input: idlRoutines

5 Temperature-dependant conductivity

See HeatOfT.pdf

5.1 Cemented and un-cemented Particulates

See Cement.pdf

Input: testKofT.tex

5.2 Testing temperature dependant conductivity

2010jan

Comparison cases were run using 3 years and 19 latitudes. Non-linear terms in $k(T)$ were set to zero, and inertia of upper layers and conductivity of lower layers were set to give exactly the linear conductivity value, so that the specifications of the two cases were effectively identical. The pressure was set to follow the Viking curve, so that the polar frosts would not affect the temperate latitudes.

Two cases run with type 52 output; first LkofT False, then True. When slightly different results were found, 4 cases were run; with LkofT False, True, False, True. No differences were found between results for the same input LkofT, indicating that the True - False difference were not due to erroneous resetting of memory, such as might be caused by indices outside defined range, apart from possibly in arrays specific to the True case.

For each of the 5 arrays stored in Type 52 files, statistics were run on the difference of the two cases, True minus False. The same process was run on each of the 5 arrays in type 52 files. Routine DELCASE first gets statistics on the entire array, with two passes, the second excluding values outside 4σ of the first. In addition, statistics were run for each of the items in an array.

Type 52 arrays are:

Array ttt has:	Tsurf	Tplan	Tatm	DownVIS	DownIR		
Dimensions	24	5	19	40	4		
Contain:	hour	item	lat	seas	case		
Array uuu has:	alat	elev					
Dimensions	19	2	4				
Contain:	lat	item	case				
Array vvv has:	DJU5	SUBS	PZREF	TAUD	SUMF		
Dimensions	40	5	4				
Contain:	seas	item	case				
Array ddd has:	Tmin	Tmax					
Dimensions	19	2	19	40	4		
Contain:	lay	item	lat	seas	case		
Array ggg has:	NDJ4	DTM4	TTA4	FROST4	AFRO4	HEATMM	
Dimensions	6	19	40	4			
Contain:	item	lat	seas	case			

Results for True - False cases

	Mean	sigma	min	max
ttt				
pass1=	-0.00043529	0.0064557	-0.46111	0.48940
pass2=	-0.00048231	0.0035892	-0.026215	0.025345
Tsurf	-0.00090867	0.0092087	-0.46111	0.48940

Tplan	-0.00075997	0.0082757	-0.39853	0.40007
Tatm	-0.00032801	0.0065075	-0.052032	0.16205
DownVIS	8.2989e-06	0.00019086	-0.0019531	0.0082397
DownIR	-0.00018810	0.0034732	-0.017777	0.093485
uuu				
pass1=	0.0000	0.0000	0.0000	0.0000
alat	0.0000	0.0000	0.0000	0.0000
elev	0.0000	0.0000	0.0000	0.0000
vvv				
pass1=	-0.032836	0.12384	-0.54330	0.15485
pass2=	-0.030271	0.11871	-0.50653	0.15485
DJU5	0.0000	0.0000	0.0000	0.0000
SUBS	0.0000	0.0000	0.0000	0.0000
PZREF	0.0000	0.0000	0.0000	0.0000
TAUD	0.0000	0.0000	0.0000	0.0000
SUMF	-0.16418	0.23695	-0.54330	0.15485
ddd				
pass1=	-0.0040661	0.084181	-0.73631	0.69478
pass2=	-0.0043492	0.069489	-0.34064	0.33194
Tmin	-0.0040681	0.082764	-0.68054	0.61833
Tmax	-0.0040641	0.085578	-0.73631	0.69478
ggg				
pass1=	-0.0081047	0.16092	-2.9978	1.9007
pass2=	-0.00024668	0.044051	-0.64148	0.53574
NDJ4	0.0052632	0.088755	-1.0000	1.0000
DTM4	-0.00018950	0.0020181	-0.014841	0.013175
TTA4	-0.00042542	0.0040637	-0.041733	0.052383
FROST4	-0.052804	0.38045	-2.9978	1.9007
AFR04	0.0000	0.0000	0.0000	0.0000
HEATMM	-0.00047228	0.021726	-0.14533	0.12671

The small change in DownVIS (0.008 of max value is 673) was unexpected, it was found to be caused by the small change in scale height caused by a difference in the atmospheric temperature. This was confirmed by setting the elevation to 0, which resulted in no change of DownVIS.

To examine the problem more closely, a statement was added to TLATS which wrote out a number of variables for each latitude, season and case. It was found that the largest differences occurred in the season following that in which the number of convergence days, NDJ4, were different.

There are two tests in tday tha can trigger setting the last convergence day, both based on DTM, the RMS day-to-day change in layer temperatures at midnight. The one for DTM being nearly constant (fractional change less than DDT=0.002 seems never to be used (min value was .07); the last day is set when DTM is less than DTMAX=0.1; TDAY also requires that the DTM computed for the last day is smaller than the trigger day.

The average DTM for Tcon was .0861548, that for KofT 0.0861049

There were 14 times when the number of convergence days differed, out of 2280

The average number of days for False/True were 7.00219 ,7.00658

Added temporary code in TDAY to write out FAC7 and layer ΔT 's, initially every time but that made huge files, so only at times when calculations went all the way to the bottom layer, which is every 2 hours. Still generates 18366 lines for one latitude.

Wrote KRC43 to look at ΔT in time loop. till generates 18366 lines for one latitude. After finding the part of the run for which the number of convergence days were the same, thus being straight-forward to compare, compute statistics for each saved item for value and change between cases. These are shown in table below, along with the change normalized to the Mean Absolute value for each item.

The bottom layers for time doubling (printed in krc.prt) are: 4 6 8 10 12 20. Hence, the ΔT and its change would be expected to increase below each of these steps.

Item	Mean	StdDev	Minimum	Maximum
Value				
FAC7	6.30496264	0.00033094	6.30529356	6.30529356
dt2	-0.00953017	0.74000335	-1.60779512	1.88100445
dt3	-0.00382637	0.57072055	-1.17243278	1.32609236
dt4	-0.00016793	0.39744040	-0.71563292	0.82678801
dt5	0.00062784	0.52034736	-0.85654873	1.11210728
dt6	-0.00000143	0.22762474	-0.35643440	0.44961619
dt7	-0.00020698	0.14914675	-0.17599730	0.28826970
dt8	-0.00014462	0.07949657	-0.09983830	0.15199120
dt9	-0.00021678	0.07759106	-0.11688780	0.14435400
dt10	-0.00011440	0.03198550	-0.06418030	0.05997020
dt11	0.00000534	0.02309710	-0.06537880	0.08025720
dt12	0.00026830	0.00800052	-0.02782550	0.02934820
dt13	0.00070504	0.00932394	-0.02700320	0.02524080
dt14	0.00056112	0.00804087	-0.05450010	0.01298910
dt15	0.00114619	0.00828606	-0.00978450	0.08045830
dt16	-0.00052190	0.00812550	-0.06155470	0.01021010
dt17	0.00096459	0.00694588	-0.00888760	0.04014310
dt18	0.00022052	0.00558620	-0.01030950	0.00918710
dt19	0.00033041	0.00589143	-0.01310710	0.01515460
dt20	0.00025516	0.00500442	-0.00770330	0.00900730
Change				
FAC7	0.00000048	0.00000000	0.00000048	0.00000048
dt2	0.00000003	0.00000782	-0.00009227	0.00008285
dt3	-0.00000006	0.00000568	-0.00007927	0.00005758
dt4	-0.00000008	0.00000453	-0.00007856	0.00003415
dt5	-0.00000017	0.00000888	-0.00019383	0.00007170
dt6	-0.00000017	0.00000894	-0.00019750	0.00008839
dt7	-0.00000054	0.00002063	-0.00051940	0.00018780
dt8	-0.00000043	0.00002041	-0.00060130	0.00036010
dt9	-0.00000025	0.00004396	-0.00208550	0.00076390
dt10	0.00000087	0.00007460	-0.00359640	0.00276280
dt11	0.00001232	0.00029545	-0.00276430	0.01712380
dt12	0.00000802	0.00022708	-0.00986730	0.00709600
dt13	-0.00000051	0.00028553	-0.00953960	0.00549600
dt14	0.00001854	0.00051008	-0.00441390	0.01323000
dt15	-0.00004195	0.00053688	-0.01266470	0.00452640
dt16	0.00001029	0.00029857	-0.00293140	0.00632640
dt17	0.00000449	0.00079936	-0.00997620	0.01415670
dt18	-0.00000724	0.00038301	-0.00719460	0.00491180
dt19	0.00000661	0.00007952	-0.00060410	0.00075570
dt20	0.00000077	0.00005681	-0.00033320	0.00051780
Normalized				
FAC7	0.00000008	0.00000000	0.00000008	0.00000008
dt2	0.00000007	0.00001470	-0.00017333	0.00015564
dt3	-0.00000014	0.00001334	-0.00018623	0.00013526
dt4	-0.00000026	0.00001418	-0.00024570	0.00010682
dt5	-0.00000040	0.00002066	-0.00045090	0.00016680
dt6	-0.00000086	0.00004572	-0.00101003	0.00045205
dt7	-0.00000396	0.00015250	-0.00383924	0.00138815
dt8	-0.00000596	0.00028455	-0.00838324	0.00502049
dt9	-0.00000365	0.00063786	-0.03026020	0.01108403
dt10	0.00003065	0.00264342	-0.12744309	0.09790339
dt11	0.00060872	0.01459237	-0.13653105	0.84575814
dt12	0.00123823	0.03504863	-1.52296805	1.09523165

```

dt13 -0.00006120 0.03454603 -1.15420711 0.66496730
dt14 0.00260983 0.07179526 -0.62127244 1.86217058
dt15 -0.00586633 0.07507097 -1.77087688 0.63291633
dt16 0.00148966 0.04320403 -0.42418355 0.91545123
dt17 0.00073377 0.13071573 -1.63135839 2.31497478
dt18 -0.00148039 0.07828456 -1.47052312 1.00393558
dt19 0.00127036 0.01527329 -0.11602344 0.14513978
dt20 0.00017693 0.01308114 -0.07672276 0.11922884

```

The difference in midnight temperatures for each layer generally increases with depth, except for the deepest few layers.

Mean and stddev NEED TO DO ABS

Compare new Ames GCM run gcm1.inp with 2008oct16 gcm2b.inp

11 1 gcm1A

252

522 save in ---h

61

Case= 1 had: ALBEDO=0.25 INERTIA=270. TauDust=0.3 SLOPE=0. SloAzi=90. IC=999

Case= 2 changed: TauRati=0.5

Case= 3 changed: TauDust=0.15

Case= 4 changed: CABR=0.18

Case= 5 changed: PTOTAL=689.7

Case= 6 changed: IB=1

Case= 7 changed: IB=2

Case= 8 changed: N1=12

Case= 9 changed: N1=12 IB=1

Case= 10 changed: N1=12 IB=2

ALBEDO	EMISS	INERTIA	COND2	DENS2	PERIOD	SpHeat	DENSITY		
0.25	1.00	270.00	3.40	928.00	1.03	630.00	1600.00		
CABR	AMW	[ABRPHA	PTOTAL	FANON	T_ATM	T_DEEP	SpHeat2		
0.11	43.50	0.00	540.00	0.05	200.00	180.00	1300.00		
TauDust	Dust_A	TauRati	Twilite	ARC2	[ARC3	SLOPE	SloAzi		
0.30	0.90	1.00	0.00	0.50	0.00	0.00	90.00		
TFROST	CFROST	AFROST	F_Emis	AF1	AF2	FroExt	[fd32		
146.00	589944.00	0.65	0.95	0.54	0.00	50.00	0.00		
RLAY	FLAY	CONVF	DEPTH	DRSET	DDT	GGT	DTMAX		
1.20	0.18	4.00	0.00	0.00	0.00	50.00	0.00		
D_Jul	DelJul	SDEC	D_AU	LsubS	SolCon	Gravity	AtmCp		
11897.84	1.03	25.19	1.66	90.89	1368.00	3.73	860.00		
N1	N2	N3	N4	N5	N24	IB	IC		
29	384	1	1	20	48	0	999		
NRSET	NMHA	NRUN	JDISK	[IDOWN	Flx14	Flx15	KPREF		
333	24	1	1	0	45	65	1		
K4OUT	JBARE	Notif	IDISK2						
52	0	5	0						
LP1	LP2	LP3	LP4	LP5	LP6	LP7	LP8	LP9	LP10
0	1	0	0	0	0	0	0	0	0
LPORB	LKEY	LSC	LNOTIF	LOCAL	Prt76	LPTAVE	Prt78	Prt79	LONE
1	0	0	1	1	0	0	0	0	0

fix tauRat, run KRC gcm1B

ALBEDO	EMISS	INERTIA	COND2	DENS2	PERIOD	SpHeat	DENSITY
0.25	1.00	270.00	3.40	928.00	1.03	630.00	1600.00

CABR	AMW	[ABRPHA	PTOTAL	FANON	T_ATM	T_DEEP	SpHeat2		
0.11	43.50	-0.00	540.00	0.05	200.00	180.00	1300.00		
TauDust	Dust_A	TauRati	Twilite	ARC2	[ARC3	SLOPE	SloAzi		
0.30	0.90	1.00	0.00	0.50	-0.00	0.00	90.00		
TFROST	CFROST	AFROST	F_Emis	AF1	AF2	FroExt	[fd32		
146.00	589944.00	0.65	0.95	0.54	0.00	50.00	0.00		
RLAY	FLAY	CONVF	DEPTH	DRSET	DDT	GGT	DTMAX		
1.20	0.18	4.00	0.00	0.00	0.00	50.00	0.00		
D_Jul	DelJul	SDEC	D_AU	LsubS	SolCon	Gravity	AtmCp		
11897.84	1.03	25.19	1.66	90.89	1368.00	3.73	860.00		
ConUp0	ConUp1	ConUp2	ConUp3	ConLo0	ConLo1	ConLo2	ConLo3		
0.01	0.00	0.00	0.00	0.22	0.00	0.00	0.00		
N1	N2	N3	N4	N5	N24	IB	IC		
29	384	1	1	20	48	0	999		
NRSET	NMHA	NRUN	JDISK	[IDOWN	Flx14	Flx15	KPREF		
333	24	1	1	0	45	65	1		
K4OUT	JBARE	Notif	IDISK2						
52	0	5	0						
LP1	LP2	LP3	LP4	LP5	LP6	LPGLOB	LVFA	LVFT	LkofT
0	1	0	0	0	0	0	0	0	0
LPORB	LKEY	LSC	LNOTIF	LOCAL	Prt76	LPTAVE	Prt78	Prt79	LONE
1	0	0	1	1	0	0	0	0	0

```

1 TauRati=0.5
2 TauDust=0.15
3 CABR=0.18
4 PTOTAL=689.7
5 IB=1
6 IB=2
7 N1=12
8 N1=12 IB=1
9 N1=12 IB=2

```

Versions agree to 1.E-5 K

253: HSTATS on now-hold

ArrayDelta	Mean	StdDev	MIN	MAX
ttt	-2.61068e-08	9.85024e-06	-9.15527e-05	0.000122070
uuu	0.00000	0.00000	0.00000	0.00000
vvv	0.00000	0.00000	0.00000	0.00000
ddd	-33.5235	71.0970	-201.804	7.62939e-05
ggg	-1.72792e-07	4.59667e-06	-3.05176e-05	4.03225e-05

ddd diff is due to non-computed layers in last 3 cases being 180 in the old version and ==0 in the KofT version, determined by doing the following:

```

qq=reform(ddd-dddh)
qt=transpose(qq,[2,3,0,1])
qp=reform(qt,200,28,2)
chart,qq[*,* ,0]
chart,qp[0:139,* ,1]
for i=0,27 do print,HSTATS(ddd[i,* ,* ,*]-dddh[i,* ,* ,*],[ 'M', 'S', 'I', 'X'])
3.81470e-08 1.02201e-05 -3.05176e-05 6.10352e-05
-1.14441e-07 1.01047e-05 -4.57764e-05 6.10352e-05
-5.72205e-07 8.79100e-06 -4.57764e-05 4.57764e-05
-3.43323e-07 9.56538e-06 -4.57764e-05 6.10352e-05
-6.48499e-07 9.73109e-06 -6.10352e-05 4.57764e-05
-1.52588e-07 9.95882e-06 -4.57764e-05 6.10352e-05

```



```

1.90735e-07 9.32256e-06 -3.05176e-05 7.62939e-05
4.57764e-07 8.28536e-06 -3.05176e-05 4.57764e-05
6.48499e-07 7.87507e-06 -3.05176e-05 6.10352e-05
1.14441e-07 5.12308e-06 -1.52588e-05 4.57764e-05
1.14441e-07 3.14753e-06 -1.52588e-05 3.05176e-05
-60.4550 92.4624 -201.804 1.52588e-05
-59.1590 90.4803 -197.499 1.52588e-05
-57.8364 88.4574 -193.086 0.00000
-56.5842 86.5421 -188.884 0.00000
-55.5175 84.9105 -185.273 0.00000
-54.7373 83.7174 -182.600 0.00000
-54.2776 83.0141 -180.997 0.00000
-54.0753 82.7048 -180.277 0.00000
-54.0138 82.6108 -180.052 0.00000
-54.0016 82.5922 -180.006 0.00000
-54.0000 82.5898 -180.000 0.00000
-54.0000 82.5897 -180.000 0.00000
-54.0000 82.5897 -180.000 0.00000
-54.0000 82.5897 -180.000 0.00000
-54.0000 82.5897 -180.000 0.00000
-54.0000 82.5897 -180.000 0.00000
-54.0000 82.5897 -180.000 0.00000
-54.0000 82.5897 -180.000 0.00000

```

```
for i=0,9 do print,HSTATS(ddd[*,*,*,*,i]-dddh[*,*,*,*,i],['M','S','I','X'])
```

```

6.67572e-07 4.88859e-06 -3.05176e-05 4.57764e-05
-7.49316e-07 4.87669e-06 -4.57764e-05 3.05176e-05
-4.49589e-07 4.60751e-06 -4.57764e-05 6.10352e-05
-1.49863e-07 4.34880e-06 -4.57764e-05 6.10352e-05
-3.26974e-07 6.99981e-06 -6.10352e-05 6.10352e-05
2.17983e-06 9.36096e-06 -3.05176e-05 7.62939e-05
-5.44957e-07 3.72176e-06 -3.05176e-05 3.05176e-05
-111.745 90.0768 -201.804 3.05176e-05
-111.745 90.0768 -201.804 0.00000
-111.745 90.0768 -201.804 4.57764e-05

```

5.2.1 2010feb17

after including CP(T)

Case 1	DTIME: total, user, system=	2.6626	2.6586	0.0040
Case 2	DTIME: total, user, system=	7.6608	7.6558	0.0050
Case 1	DTIME: total, user, system=	2.7636	2.7606	0.0030
Case 2	DTIME: total, user, system=	7.7318	7.7278	0.0040
Case 3	DTIME: total, user, system=	9.9835	9.9535	0.0300
Case 4	DTIME: total, user, system=	15.3287	15.2987	0.0300

End of input: testKofT.tex

6 Evolution log of KRC

This log of modifications may not be complete.

Input: evolve.txt

evolve.txt Notes on evolution of KRC code, after the first 15 years.

85May~10-14. Dave Paige visits Flagstaff. We create new directory [hkieffer.krc.mars] in which MARS version of code is put. MARS version has larger LATCOM (JLAT changed to real*4). HELPLIST revised.

After Dave left, found that TDISK had not had JLAT change, hence was not writing LATCOM to disk.

Made new plot version, starting with 84jun comet version, but with almost entire revision using NCAR_1 routines, including new MCURVE1, CONREC1 and G01. Never got all the bugs out.

85Jun24-28 Paul Weissman visit. Found error in COMA2, otherwise no changes from Jun 84 comet version. Linked and ran and duplicated older runs.

85sep05-07 Combine the comet and mars versions into single routines which use the larger LATCOM. Only external change is reversing the meaning of LD18. Major restructuring of TLATS and TDAY to accommodate both comet and Mars; use LD20 .TRUE. if Mars, .FALSE. if comet. Other routines needed no changes.

Revise HELPLIST.

Create directory KRC.COM] for the comet-particular stuff. Move older routines for .KRC] to .KRC.COM]. Move all the routines which support both comet and mars from .KRC.MARS] to .KRC].

Delete plot routines dating from 1984 which used the smaller LATCOM.

85oct14 Add COMMON FILCOM of file names; print these in TPRINT.

Change meaning of FROST4 and AFRO4 for comet.

Minor changes in printout sequence.

Change TYPE to WRITE(IOPM in TDISK.

86oct Paul finds erroneous factor of PI in computing coma diffuse radiance; change made to source code only, not linked.

87mar29 Remove incorporation of albedo in the solar incident flux ASOL.

Add ADGDIF (diffuse solar flux) to KRCCOM.INC.

Recompile TLATS,TDAY, Link.

87jun30 TDAY: Avoid /0 if DTMJ(JJJ)=0 at "done" test.

87sep11 .MARS] TDAY & TLATS: Special versions for metamorphism.

Use ZLAT(17:19) for input of metamorphism and sublimation constants.

Use TT(J,MAXN3) to transfer metamorphism rate.

TYEAR: version of TLATS which averages daily insolation and includes

PORB in the insolation calculation. Uses AVEDAY.

87oct01 ALL Separate the use of NMHA for storage and N24 for printout.

TLATS: replace CFSOLAR with AVEDAY.

87nov22 .YEAR] versions of DAYCOM and LATCOM with larger MAXN24 and MAXN2 meant for use with TYEAR.

Most routines compiled into .YEAR] with the these.

Will need to redo TDISK if it is to be used.

87nov22 TCARD: Add report if input integers are reset into valid range.

NMHA no longer constrained.

TPRINT: redo some formats.

TDAY: Add error report if convergence is unstable.
KRC: set IOERR=IOPM rather than IOSP.
Force parameter print through call to TPRINT(2) if TDAY(1) error occurs.
.MARS]TLATS: now includes variance tests.

88sep08 .MARS]TDAY: Test moving layer T limit tests and metamorph from N24 into each time loop; so that they are done 1536 instead of 40 times per "day". Negligible effect; <.01 degree in TMETA.

97fall-98summer Incorporate one-layer atmosphere with many parameters that can be tuned to mimic Haberle-Jakosky model. Wrote LaTeX description. Build TES look-up code for computation of thermal inertia from TES observations; this interfaces with Mike Mellon model set.

1997sep idlkrc.f Build this IDL interface to call KRC.

98sep01-07 Add section to TDISK for output of bin5 files type 51 and 52. Minor code cleanup, avoiding divide-by-zero if atmosphere parameters were zero Make KRC/moon version of TLATS and TDAY by removing all atmosphere code, and including eclipse section in TLATS; commons left the same even though atmosphere results not calculated.

1999dec krc.f tcard.f Add option to continue from current condition

2002mar07 alsubs.f Created. Adopted from l_sub_s.pro

2002mar07 krc.f tcard.f Major change. Add option for "one-point" rapid runs for Surface T

2002jul Major change. Incorporate Delta-Eddington atmosphere. Found that double precision required within deding2.f

2002aug04 tdisk.f Add output file type 53=(combo at 1 lat, 2+80 seasons, 10 cases). Recode logic

2002nov01 tseas.f Have DJU5 increment by current DELJUL for each season

2004jul06 tdisk Add file style 54 Add the Common HATCOM

2004sep28 porb.f Change name of called routine ROTATE to ROTVEC to avoid library conflicts

2004sep28 tlats.f Add tests to avoid round-off to negative fluxes at night so code would run at ASU

2004sep30-Oct5 tday.f tdisk.f Add storage of surface downward fluxes every hour on last day. Revise file style 52 to include them (and a spare variable).

2005nov18 tlats Add optional solar zenith angle limit

2005nov19 tprint Add print of depth to top of 2nd layer

2005dec28 tlats Fix bug using ZENLIM. Additional comments

2006jan25 tlats Modify SKYFAC from linear with slope to $(1+\cos s)/2$

2006apr12 tdisk Change file style 54 to have both 1am and 1pm surface Temp.
2006apr22 tdisk Allow flexible number of cases for output file type 52 and 54
2006apr30 tdisk Add TTB4 to type 54 output
2006sep09 tcard Correct error: REAL*4 LSUBS should have been ALSUBS
2006sep09 Allow seasonally variable albedo and TauD.
three new routines: seasalb, seastau, readtxt360;
and changes in tseas and tcard and filcom.inc

2008oct02-25

Found error in calculation of planetary temperature; was using
hemispheric integral 1-BETA instead of $\exp(-\text{Tau_IR})$; so in effect using tau
that was too large by factor of $\text{tau_eff}/\text{Tau_IR}$.
Modify: krc.f tseas.f tlats.f tday.f tdisk.f tprint.f tcard.f
krccom.inc daycom.inc
Use slope azimuth as a flag for a pit of slope SLOPE.
Put in proper SKYFAC for conical pit.
Allow for Snow formation in cold atmosphere and fall to surface.
Move MAXBOT from daycom.inc to krccom.inc.
Replace ID22(1) and (2) with KVALB and KVTAU.
Move AMW fixed value in tlats.f into krccom in place of ABRAMP
move other physical and hardware-dependent constants into krc and krccom.
Add output file type 56 in tdisk.
Replace dual use of GGT by using new DTMAX for daily convergence.
Major modification of output Type 52.
Use IDOWN as season index at which to read some changes
TMN4Y(

2008nov11-2009feb

Add temperature-dependant conduction option
Use L10 as logical variable for k-of-T
Lengthen KRCCOM and move some inputs around
recode tday to have constant and T-dependant options
Update IDL routines that deal with KRCCOM
Remove max # seasons.

2009apr22

Begin ability to forecast deep layers based on storage of midnight values.

Scheme 1. [Coded, but not refined. Mostly a stub] Store at integral years prior
to forecast season index. If have 3 or more values, use EPREAD; should be safe
to use on all layers. Because of possible jump perturbation on the first day,
best to not have that be one of the stored dates. Does nothing if IDISK2=0. [To
remove, delete TMN4Y from hatcom, delete call to TYEARP in TSEAS, remove
tyearp.o from sources in Makefile]

Scheme 2. [Not implimented] Rolling storage of all seasons for the past year. At
forecast season; evaluate thermal delay to surface and make [complicated]
forecast.

2010jan11 Go to IMPLICIT NONE in all major routines

Change names in krccom PI>>PIVAL RAD>>RADC to improve search uniqueness
Move TINT call from tlats to tseas so as to remove 1-season delay in SUMF

Add D-line writes to fortran units for study of KofT differences.

2010apr21 Update 3 routines that deal with the 4 input lines related to temperature-dependent properties: tprint, tcard, tday , giving the extension .f2 and the older versions extension .f1. Make soft links of .f to the .f2 version Also make soft link for krccom.inc to krccom.inc2 As of this date, the base version of KRC has the temperature-dependent option, invoked by setting

Add option of 3rd integer on first input line. If not 0, then will read a card of 6 debug flags, IDB1 to IDB6, which are contained in units.inc and all are initialized as 0 in TCARD(1)

End of input: evolve.txt

7 Liens on KRC system or code

2009??

Check T_s oscillation when have night frost

Input: liens.txt

Geophysical liens

No explicit H2O condensation

Coding Limitations

Only one binary output format at a time.

2002mar09 Possible reordering of Real inputs

Type 1 Real parameters (8F10.2) =====

Surface Properties

1 ALB Surface albedo
2 EMIS Surface emissivity
3 SKRC Surface thermal inertia [$J m^{-2} s^{-1/2} K^{-1}$] { cal cm * 4.184e4}
23 SLOPE Ground slope, degrees dip. MUST NOT slope beyond pole.
24 SLOAZI Slope azimuth, degrees east from north
15 TDEEP Fixed bottom temperature. Used if IB<=1.
7 SPHT Surface specific heat [$J/Kg/K$] {cal/g/K * 4184.}
8 DENS Surface density [kg/m^3] {g/cubic cm. *10}

4 COND2 Lower material conductivity (IC>0)
5 DENS2 Lower material density (IC>0)
16 SPHT2 Lower material specific heat (IC>0)

Atmospheric Properties

9 CABR Atmospheric infrared back radiation coefficient
10 ABRAMP Amplitude of diurnal variation of CABR
11 ABRPHA Phase of ABRAMP, degrees relative to midnight.
12 PTOTAL Global annual mean surface pressure at 0 elev., Pascal[=.01mb]
13 FANON Mass-fraction of mean atmosphere that is non-condensing
14 TATM Atm temp for scale-height calculations

Dust & Slope Properties

```

17 TAUD Mean visible opacity of dust, solar wavelengths
18 DUSTA Single scattering albedo of dust
21 ARC2 Henyey-Greenstein asymmetry factor
    moon      = eclipse start time in local Hours
22 ARC3 coeff. for planetary heating
    moon      = eclipse duration in seconds 0=no eclipse
19 TAURAT Ratio of thermal to visible opacity
20 TWILI Twilight extension angle [deg]
-----
    Frost Properties
25 TFROST Minimum Frost saturation temperature
    may be overridden by local saturation temperature (LVFT)
26 CFROST Frost latent heat [J/Kg] {cal/gm*4184. [ Not used if
27 AFROST Frost albedo, may be overridden (LVFA) [ TFROST never
28 FEMIS Frost emissivity [ reached
29 AF1 constant term in linear relation of albedo to solar flux
30 AF2 linear term in relation of albedo to solar flux units=1/flux
    Afrost = AF1 + AF2 * <cos incidence> SOLCON / DAU^2
31 FROEXT Frost required for unity scattering attenuation coeff. [Kg/m^2]
32 fd32 spare
-----
    Thermal Solution Parameters
33 RLAY Layer thickness ratio
34 FLAY First layer thickness (in skin depths)
35 CONVF Safety factor for classical numerical convergence
    0 for no binary time division of lower layers
    >0 for binary time division
36 DEPTH Total model depth (scaled) (overrides FLAY if not 0.)
37 DRSET Perturbation factor in jump convergence. If = 0., then
    all layers reset to same average as surface layer.
38 DDT Convergence limit of temperature 2nd differences
39 GGT Convergence test: RMS layer T changes in a day,
    and del_T surface each time step
40 FDOWN Mass ratio factor for grid motion ( enabled by IDOWN)
-----
    Orbit Geometry & Constants
6 PERIOD Length of solar day in days (of 86400 seconds)
46 SOLCON Solar constant Applied Optics 1977 v.16, p.2693: 1367.9 W/m^2
47 GRAV Surface gravity. MKS-units
41 DJUL Starting Julian date of run (N5>0)
42 DELJUL Increment between seasons in Julian days (N5>1)
43 SDEC Solar declination in degrees. (N5=0)
44 DAU Distance from Sun in astronomical units (N5=0)
45 SUBS Aerocentric longitude of Sun, in degrees. (for printout only)
48 fd48(13) Spare

```

End of input: liens.txt

8 Code details

Input: code.tex

deding2.f has inputs of:

- OMEGA dust single scattering albedo
- G0 dust asymmetry parameter

ASUR surface albedo
COSI cosine of incidence angle
TAU dust vertical opacity

and outputs:

BOND Planetary (atm plus surface system) albedo
COLL Direct beam at bottom = collimated + aureole

RI [2,2] Diffuse irradiances:

[1, = I_0 = isotropic [2, = I_1 = asymmetric

,1] = at top of atmosphere ,2] = at bottom of atm

_____ *End of input: code.tex* _____

9 IDL interface

Development of an IDL interface to the Fortran KRC thermal Model System

2002March01 Hugh Kieffer, Kris Becker.

Source file locations:

IDL top procedure /home/hkieffer/idl/tes/KRC/topkrc.pro
IDT to Fortran wrapper /home/hkieffer/idl/externals/krcw.c
C:Fortran interface /home/hkieffer/idl/externals/idlkrc.h
Fortran top function /home/hkieffer/krc/tes/idlkrc.f
Fortran simulator for IDL /home/hkieffer/krc/tes/idummy.f
Makefile /home/hkieffer/krc/tes/Makefile
Read for initialization /home/hkieffer/krc/tes/idldummy.inp
Fortran common definitions /home/hkieffer/krc/tes/*.inc

10 Development notes

Input: notes.tex

Partial scattered notes made during development and test

10.0.2 2003jan23

Notes following tutorial for Robin Ferguson, Josh Bandfield and Tim Titus.

One-point model temperatures are influenced by elevation if the opacity is >0

Note, negative opacities can cause very long (infinite?) run times.

If $N5=0$, then the values entered for DAU and SEDC are used for a single season.

If $KPREF=2$ and $N4>8$, then will do global condensed frost integration, in which case you better have input a reasonable set of monotonically increasing latitudes.

The stored times-of-day are such that the last one is "midnight". E.g., Hour 1 is 1/24 of a sol after midnight. (in tday.f, see use of JJH)

Frost amount units are kg/m^2

Minor code changes to insure non-negative TAUD and DRSET, and to avoid opening output binary file if it would not be written to.

10.0.3 2004jul05

Desire to output the heatflow into the bottom of a frost layer
This would require output of DAYCOM for every latitude and season.
Decide to define a new common to contain simply this information.
define Array (# latitudes, # seasons, # models)
Then bin5 output only the defined part.

In TDAY, isolate the heat conduction term at the surface.
Requires a counter for model!
In TDISK, write new section for KODE=54
Actually: move all dimension-defining parameters from other commons into KRCCOM,
and create new hatcom.inc to hold new items

10.0.4 2004Jul11

There is a new version of KRC which can output the
daily average upward heatflow into the surface. This is intended for study of
the ice caps. The file output is an additional style of .bin5, selected by
K4OUT=54

In the process, I somewhat modernized the code by moving all the
dimension-defining parameters from other commons into KRCCOM and removing the
redundant definitions. I also added a new common, hatcom.inc, to transfer the new
items between routines; I am reluctant to change any sizes in the existing
commons because the C code used to read them into IDL can tolerate no mismatch
in dimensions. This common would be the place to add other additional items.

As far as I know, the new version is backward-compatible with the circa 2002
version that was distributed to Robin.

I would like to get some mileage on this new code before distributing it.

I checked, and the frost temperature does change if KPREF and LVFT are set
properly.

10.0.5 2006sep09

Design for allowing seasonally variable albedo and TauD

Without changing sizes of common, there are few places available for control
values. Choose to use ID22 indicate variable soil surface (not frost) albedo and
23 to indicate variable soil Taud. 0 will mean they are not variable. Files of
text table of value versus season can be read at the start of a run. These will
apply to ALL latitudes.

Requires logical unit to read file, closes before return
Requires additional file names, read via fcard.
requires a way to know how many season points are used.

Implimentation is complex. but can turn these variable features on and off:

To start variable albedo, input card
8 22 0 'AlbedoFileName' / Variable albedo text file name
Can revert to constant albedo by hokey technique of giving it bad name. E.g.,
8 22 0 'badName' / turn variable albedo off

Variable Tau done the same way, with 22 being replaced with 23

Found code error in tcard.f REAL*4 LSUBS ! function
should have been ALSUBS

10.0.6 2009mar11

Test1: same results for constant K

loadt CONUP0:CONLO3 to be constant at same value as L10=0 case

dtimes
1.3018 0.0080 5.1952 0.0370
1.3028 0.0080 5.2032 0.0290
1.2588 0.0080 5.1922 0.0360
1.2598 0.0080 5.1982 0.0290

Output in work1/krc/mars/kot3.t52

IDL program qk to test differences.

Number of convergence days differs by 1 for ??

Do the following for top and lower layers

- 1) decide on input paramters that influence k of T
- 2) run either kotab.f or spspread.pro to generate k of T table
- 3) fit cubic function for scaled $x=0.015*(T-220)$
koffit.pro
- 4) transfer 4 coefficients into krc input file

10.0.7 2009dec24

Decide to elimiate conflicting use of PERIOD and PI by changing porbcm.inc and all that reference it. Then will be able to reference it in tseas and tyearp.

PERIOD >> OPERIOD and PI >> PICON
Need to modify: porbcm.inc tseas.f tyearp.f
ephemr.f porb1.f porb.f porbin.f porbmn.f porbqq.f
Remove probdi and porbls as obsolete
porbin.f is not used??

10.0.8 2010jan09

run KofT compare after resetting upper I to 7 decimal places

fix tday.f to recompute fac7
make krc

modify master.inp
run krc
DTIME 2.2677 3.4385

```
IDL .rnew lookrc
11
52
```

10.0.9 2010jan22

Checking hoe one-point mode switch occurs:

```
TCARD(1 called only by KRC in single place
TCARD(2 called by KRC(2 places) and TSEAS (1 place)
```

When change card 10 encountered, tcard renames FINPUT and sets IR=4
Card 10 should only be used after the initial full parameter block.

then krc, when it gets IR=4, will switch input files (close old, open new) read
the first line of new file and write that to IOSP, and read the second line of
input file (ignored) and write a column header line to ISOP.

```
Debug direct fortran writes. All have comment !dbw
fort.41 tday 191 J,DIFFI(J),DTIMI, BLAY(J),SCONVG(J)
fort.42 tlats L101 n1,j3,j4,j5,n1pib
fort.43 tday 307 jj,jjj,j5,kn,fac7,(dtj(j),j=2,kn)
fort.44 tlats L287 j3,j4,j5,ncase,efrost,ave_a,taud,pres
& ,DTMJ(J3),DTMJ(J3P1),TMIN(2),TMAX(2)
```

End of input: notes.tex

10.1 Mar 11 2009 puzzels

Input: /home/hkieffer/krc/tes/puzzels

```
master.t52 shows DOWNVIS delta one day later than the other four.
@44 CHART,y3[1700:1800,*],parti=ita
```

Examine: in tday.f

```
294:   ATMRAD= FAC9*TATMJ**4   ! hemispheric downwelling  IR flux
298:   POWER = (1.-A)*ASOL(JJ) + FAC6F*ATMRAD
309:   ABRAD = FAC3*ASOL(JJ) + FAC6*ATMRAD   ! surface absorbed radiation
355:   DOWNNIR(IH,J4)=ATMRAD ! save downward IR flux

298:   POWER = (1.-A)*ASOL(JJ) + FAC6F*ATMRAD
309:   ABRAD = FAC3*ASOL(JJ) + FAC6*ATMRAD   ! surface absorbed radiation
354:   DOWNVIS(IH,J4)=ASOL(JJ) ! save downward solar flux
```

so DOWNNIR based on current TATM whereas DOWNVIS based in inputs from TLATS
which uses albedo at the beginning of season. So, if frost flag stored at a
season if different between two runs, DOWNVIS can be quite different the next
season.

```
-----
t55 tbot ldelts up to 3K
t52 hows lowest layer T Delta<0.5
ttt siq2=.026
```

```
uuu ==0
vvv== 0
ddd sig2=.049
ggg siq2 .038
-----
```

Return from tcard.	Conditions
1 re-start fresh	Called with IQ=1
2 read a existing file record	KOLD>0
3 continue from memory	read 2 >20 x AND IR .ne.2
4 go to one-point mode	read a one.inp file name
5 stop	successive 0/

Need way to set ultimate number of seasons for TDISK

```
J5
JDISK
JBARE
IDISK2
-----
```

All fixed by recompile all
seems to not be using latest disk

```
WRITE(*,*)'Idx,JJJ=',IDX,JJJ
Idx,JJJ= 5 24 7 4 42 0 0 4 30 0 does not include IDX
```

```
Initiated custom output: K4OUT= 52
JJJ=   5   24   7   4   41   0   0   4   30   0
Idx,JJJ= 5 24 7 4 42 0 0 4 30 0
MMM= 24 168 672 27552 0 0
KOMMON,KASE,MASE,MTOT= 1056720 28224 37 1044288
```

screen:

```
N4= 19
N4,jjj= 19 5 24 7 19 40 0 0 4 30 0
IDX= 4
NDX,K= 1 451
Idx,JJJ= 4 5 24 7 19 41 0 0 4 30 0
MMM= 24 168 3192 127680 0 0
Segmentation fault
```

bin5 contorl has 4 lats whereas 19 specified
convergence layer 16 mis prints
did not run 2nd case

===== below OK =====

```
bhead kot3.t51  5 24 2 4 44 1 4 8448 512 << 2009 Feb 24 08:20:38
bhead kot3.t52  5 24 7 4 42 3 4 84672 512 << 2009 Feb 24 07:47:04
bhead kot3.t54  4 41 5 6 1 4 1230 512 << 2009 Feb 24 08:20:38
```

bhead kot3.t55 3 41 17 1 4 697 512 << 2009 Feb 24 08:20:38

bhead kot3.t56 4 71 4 43 1 4 12212 512 << 2009 Feb 24 08:20:38

remove files, rerun, get proper sizes.

```
-rwxr--r-- 1 hkieffer 49360 Feb 24 08:26 kot3.t56*
-rwxr--r-- 1 hkieffer 5432 Feb 24 08:26 kot3.t54*
-rwxr--r-- 1 hkieffer 339200 Feb 24 08:26 kot3.t52*
-rwxr--r-- 1 hkieffer 34304 Feb 24 08:26 kot3.t51*
-rwxr--r-- 1 hkieffer 3300 Feb 24 08:26 kot3.t55*
```

So. BIN5 must overwrite old file if there is room

End of input: /home/hkieffer/krc/tes/puzzels

11 files to consider

GCMcompare.txt

12 Make notes

98may26 In tdisk.f, the units of RECL depend upon compiler options. E.g., Sun Fortran; units are bytes (ANSII standard?) if -xl[d] is not set, and 4-byte words if it is set. For Linux, the units seem to be bytes.

2010jan gfortran produces warnings:

In file tcard.f:52

```
& , 'SPEC_HEAT', 'DENSITY', 'CABR', 'AMW', 'ABRPHA', 'PTOTAL', 'FANON' !7
1
```

Warning: initialization string truncated to match variable at (1)

However, seems to have defined the entire array properly