CSU – RAMS

Model Update and Modification Details

This document contains detailed notes of changes made between model versions as development and deprecation were performed on the model by Stephen Saleeby and others at CSU starting in 2010. This is meant primarily as a reference for noting when particular changes take place in CSU-RAMS.

Prepared by:

Stephen Saleeby
Department of Atmospheric Science
Colorado State University

Last updated: 12 September, 2016

DESCRIPTION OF MODIFICATIONS MADE TO RAMS BETWEEN VERSIONS

VERSIONS ARE UPDATED WITH A DATE STAMP and/or VERSION NUMBER

Orange asterisks (*) indicate certain bug fixes or code changes that can impact general model solutions. These asterisks are not used for new code implementations that will change the solutions.

Some changes were prompted by bounds checking via the compiler. You must compile with "-Mbounds" for PGI or "-check bounds" for IFORT and do a test run.

##
Moving from Past Versions to rams6_20100504
*For the micro budgets I set the k=1 level to the k=2 level value. (file mic_driv.f90)
*Set all latent heat budgets to run over "do k=2,m1" (file mic_driv.f90)

##
Moving from rams6_20100504 to rams6_20101004:
*In geodat.f90 added statement to solve dateline SST issues
*In landuse_input.f90 put min/max functions for glonp1 for dateline issues
*In rad_driv.f90 put min/max bounds for gp and omgp to keep them physical
*Added IBUBBLE option section to RAMSIN, condition to rhhi.f90 and specification to "bubble" in ruser.f90. This added the variables:
IBUBBLE,IBDXIA,IBDXIZ,IBDYJA,IBDYJZ,IBDZK1,IBDZK2,BTHP,BRTP

##
Moving from rams6_20101004 to rams6_20101207:
*Added IDATAIN to specify different ingest datasets. Currently IDATAIN = 1 is for NARR and IDATAIN = 0 for all others. Would need to adjust this for other possible grids.
*Remove call to lambcon_adjust from asti.f90. Do not need this since RAMS6 apparently added a routine called "rotate_winds" to deal with rotating winds from different projections.
*Added new variable IDATAIN into the calls to "rotate_winds" in asti.f90. And in "rotate winds" in "map_proj.f90" insert special statement whereby the winds from NARR data are not rotated to earth-relative since they are already earth-relative in the dataset. This is a special case.
*Added special IDATAIN statement for NARR data in file "astp.f90" in routine "pressure_stage" in the possible conversion of ingested specific humidity to relative humidity. The current statement assumes in input specific humidity in g/kg, but NARR is kg/kg. We may need to add more special statements for additional datasets.
*Added hybrid power law tables for hydrometeor fall speeds that use new equation for rain, pristine, graupel, hail, but original ones for snow and aggregates. (file mic_init.f90, routine micro_master)
*To fit this isan version to correct version for NARR and selectively for other datasets: In "isnstage" in asti.f90, comment out "lambcon_adjust". In "pressure_stage" in astp.f90 add situation for NARR which gives specific humidity in kg/kg, not the assumed g/kg. In "rotate_winds" in map_proj.f90 in section for (rot_type == 'lc_rps') comment out call to "uvlc_uvll" and use call to "uevetouv" as in section for (rot_type == 'll_rps').

*Changing T(kelvin) to T(celcius) by subtracting 273.15 instead of 273.16 (file kf_cuparm.f90, kf_driver.f90, oda_proc_obs.f90, rhhi.f90, rprnt.f90, asti2.f90)
*Remove density multiplier from "colfacc = colfacr * dn0" since density was previously factored into the equation (file mic_misc.f90, routine each_column)

##
Moving from rams6_20101207 to rams6_20110712:
*New cloud nucleation lookup tables that add a dimension of solubility (epsilon). Now in a separate file mic_nuctab.f90.
*Fixed and updated dust source model. Appropriate for idealized runs. Fixed and updated aerosol, salt, and dust dry and wet deposition. Added salt_dust_deposit.f90, salt_dust_include.f90, salt_dust_sources.f90.
*Fixed and updated aerosol, salt, dust radiative effects. The version by Dave Stokowski had serious errors in calculation of single scatter albedo and asymmetry parameter.
*Jerry's fix to short wave flux routine in rrad2.f90 can still allow the occasional division by zero error. Will go with the OLAM fix for now.
*ICCNLEV for aerosol depletion and restoration and tracking of aerosol mass in some of the hydrometeor species.
*New hydrometeor initialization routines for setting up aerosol log-normal distribution based on more recent parcel model simulations.
*Limiting enemb to only be used under certain conditions. (file mic_misc.f90, routine enemb)
*Putting bounds on use of supersaturation tables in cloud nucleation. (files mic_nuc.f90 or mic_nuctab.f90, routine cloudnuc/aero_nuc_tab)
*Limit amount of cloud water transfer during homogeneous ice nucleation. (routine icenuc)
*Positive definite checks added to collection transfers for rx and cx. (routine colxfers)
*Set upper size limit of GCCN to 5.0 microns. (routine cldnuc)
*Fixed minor error in establishing k levels of cloud water after nucleation (cldnuc)
*Added output of additional radiation variables, 3D precipitation rate, vertical velocity terms budgets.

##
Moving from rams6_20110712 to rams6_20120202:
*Separated out aerosol species into CCN, GCCN, dust1, dust2, salt file, salt jet, salt spume, sub-micron and super-micron regenerated aerosol. These are all nucleated separately in cloud nucleation.
*Created file called mic_nucpre.f90 called prenuc_ccn to check all aerosol for number, mass, and size limits and determine the ratio of cloud water that each gets at nucleation based on total species surface area. Using revised formula to go from median radius to mean radius exp(1.5 * (alog(sigma_ccn))**2)
*Installed revised DeMott formula for heterogeneous ice nucleation based on all aerosol species concentration (except sea salt) > 0.5 micron diam. Apply a pre-call to this in cloudnuc to routine prenuc_ifn in order to preferentially set aside aerosols to act as IN. Then remove those from the available aerosols for cloud nucleation.

##
Moving from rams6_20121207 to rams6_20130409:
*Fixed an implementation of DeMott(2010) IN and made flag IIFN = 1,2,or 3. The user can now use it like the Meyers scheme with an IN profile in CIFNP.
*Added soil moisture, soil temperature ingest for 2 layers. Also ingest of snow water content and snow depth. For now this is just for NARR and GFS data and the user must use new degribber code to extract these data. These fields are added to the varfiles and then ingested appropriately at the model start. Routines to set up the soil and snow fields are now in leaf_init.f90.
*Made a couple minor changes to rsys.F90 and rammain.f90 for compiler and system commands. This changes allows RAMS to compile on standard Linux and Mac Linux with PGI, IFORT, or GFORTRAN.
*In sfc_driver.f90 added calls to new boundary condition subroutines. Added call to "turb_bcond" for sflux_u, sflux_v, sflux_w, sflux_r, sflux_t and added call to "rad_bcond" for rlongup and albedt. These two routines have been added after "leaf_bcond" in the file leaf3.f90

##
Moving from rams6_20130409 to rams6_20130813:
Mods to Powers Laws and Sedimentation
*Added True-Bin sedimentation in addition to the Standard version using displacement distance and "reldisp". I have kept both for now, but perhaps the old version could be removed in the future. This new Sedimentation is based on work from Adrian Loftus.
*"nembfall" in mic_driv.f90 was increased from 20 to 40 to better cover mean mass parameter space
*"nbins" was set to 41 in mic_tabs.f90. Bins are mass doubled and span the range of allowable mean masses.
*In the setup of gamma distributions across bins, the lowest and highest 5% of bins are removed to prevent including and large tail of mass and the small tail of number. Distributions are then normalized. Note that this 5% should be carefully changed if "nbins" is changed in the future.
*Added Rob Carver's power law bands to micphys.f90 and sedimentation lookup tables (mic_tabs.f90,mic_misc.f90,mic_driv.f90).
*Added dimensions of "ndensrtgt" and "nband" to sedimentation lookup tables. "nband=3" is for implementation of Rob Carvers banded power laws. This works only in the True-Bin sedimentation. "ndensrtgt=40" dimension covers parameter space for the density and rtgt effects on fall distance. See tables below:
  pcp_tab(ngr)%pcpfillc(mmzp(ngr),maxkfall,nembfall,nhcat,ndensrtgt,nband))
  pcp_tab(ngr)%pcpfillr(mmzp(ngr),maxkfall,nembfall,nhcat,ndensrtgt,nband))
  pcp_tab(ngr)%sfcpcp(maxkfall,nembfall,nhcat,ndensrtgt,nband))
  pcp_tab(ngr)%allpcp(mmzp(ngr),nembfall,nhcat,ndensrtgt,nband))
*In true-bin sedimentation we now bilinearly interpolate for mean mass and power law band lookup tables dimensions to remove noise that exists otherwise
*IPLAWS in RAMSIN: 0-Original4.3, 1-New Single Choice (R.Carver,Mitchell96) 2-New With Banded Plaws for sedimentation (R.Carver,Mitchell96) A value of 2 only works for new True-bin sedimentation
*ISEDIM added to RAMSIN: Sedimentation: 0-RelDisp, 1-True Hybrid-Bin

*In mic_driv.f90 has to add option of setting of "ch2" variable for mean mass rather than displacement distance. Also added new option for "mksedim_tab" called "mksedim_tab_trubin" and new option of "sedim" called "sedim_trubin".
*Removed mass and Vt power law table "dstprms3" in mic_init.f90. The banded power laws should replace this option.

Other Modifications
*Bug fix to "nuccldct" micro budget variable. In former versions this was possibly incorrect when ICCNLEV=0. Added separate calculations for the case of ICCNLEV=0 and for ICCNLEV>0. (Mods to mic_nuc.f90)
*Budget variables dust1cldrt, dust1drzrt, dust2cldrt, dust2drzrt are now only calculated for ICCNLEV>0 as they were incorrectly computed for ICCNLEV=0. The code would have to be greatly altered to have these budgets available for ICCNLEV=0. (Mods to mic_nuc.f90 and opspec.f90)
*In mic_tabs.f90 subroutine sxy, added line to keep number and mass bins consistent: "amkd(l)=0. !If bin number is zero, make sure mass is zero."

##
Moving from rams6_20130813 to rams6_20130901:
*Added cloud droplet nucleation lookup table for NaCl. Renamed the routines for accessing the lookup tables to delineate between (NH4)2SO4 and NaCl.
*In micphys.f90, added an array to specify soluble chemistry.
*In mic_init.f90 added a routine (aerosol_init) to set the vanthoff factor, (really the number of dissociated ions in solution) to the soluble chemistry. The vanthoff factor is used in (cal_dwet). This routine also determines the dry aerosol density based on weighted densities of the solid and soluble materials determined by the solubility fraction. The call to this routine is added for initial and history restarts in rdint.f90.
*In aerosol deposition routine, changed the condition for the call to cal_dwet (deliquescence growth routine) from vanthoff factor related to solubility fraction and RH related.
*In cal_dwet also inserted code to allow for different aerosol dry densities in deliquescence equation. Also inserted code to compute the density of the deliquesced solution particle to coincide with radius of the wet particles when used in wet and dry deposition routines. More deliquesced particles should have net density closer to that of water (1000 kg/m3).
*Fixed bugs in cal_dwet for a mix of radius units in meters and microns. Set an upper limit for wet particle sizes to 10 microns radius. Larger particles should not be allowed for gravitational settling for the equation being used since it is only appropriate for falling particles with low Reynold's numbers < ~0.1.
*In mic_init.f90 I moved the call to "prenuc_ccn" outside the supersaturation requirement so that is does an aerosol size computation and size limitation for all grid cells. Turns out this is necessary to keep particles sizes in bounds where nucleation is not occurring, but advection and diffusion and mess with the computed sizes.

##
Moving from rams6_20130901 to rams6_20131005:
*Reorganized some code structure to make things run more efficiently when NOT using ICCNLEV>=2. Just set things up so there's no unnecessary "if" checks or "do" loops.

*Got rid of ICCNLEV==3 option. Rather I introduced three flags to set in micphys.f90. These are "trackdust", "trackdustifn", and "trackepsilon". (trackdust=1 allows tracking of dust within hydrometeors and allocates 8 new arrays.) (trackdustifn=1 allows tracking of dust that enters hydrometeors from ice nucleation only and it allocates 8 new arrays.) (trackepsilon=1 allows tracking of aerosol solubility as a whole and allows prediction of epsilon (solubility fraction) for the regenerated aerosol categories. This requires 8 tracking arrays and 2 arrays that hold the amount of soluble regenerated aerosol mass. We then get epsilon from the ratio of soluble/total aerosol mass.
*Fixed a limited in IIFN=3 for DeMott nucleation. Was nucleating new ice
like "nifn = nifn - (total_in + immersed + ifnnuc)" but should be
like "nifn = nifn - ifnnuc". Fixed this in mic_nuc.f90 & mic_nucpre.f90 and updated total_in when particles enter as potential immersion freezing nuclei.

##
Moving from rams6_20131005 to rams6_20131008:
*Included bug fix found by Adele Igel to limit lapse rates in turb_diff.f90 subroutine truhor. Indices were improper.
*Throughout the model, I converted all aerosol number and mass concentrations from #/cm3 and g/cm3 over to #/kg and kg/kg. Did this so that aerosol fields are conserved during advection, since advection does not currently conserve volume fields.
*Decided to remove nuccldct and nucicect from budgets. The number concentration can get very large when accumulated and not sure how useful these are compared to the mixing ratios which come from conserved quantities. May remove other number concentration budgets as well, but not in this release.

##
Moving from rams6_20131008 to rams6_20131111:
*Removed all the print out and old rams plotting sections from RAMSIN and from the code. This is the start of a cleanup effort to remove obsolete and unnecessary portions of the code.
*Removed microphysical budgets for nucleation number concentration
*Created lateral boundary points for radiation variables since they formerly were simply set to zero. This created odd looking gradients at all boundaries when plotting, and could impact analyzed results.
*Reorganized the namelist info in rcio.f90, rname.f90, and mpass_init.f90 to make it easier to add variables in the future and keep them ordered.
*Commented out a lot of print statements just to clean up the standard output for statements we really never use. Can turn these back on if we need to.
*Got rid of calls to "numcheck" in the micro driver. These are no longer needed for IIFN=3 since we run this check in the negative adjustment call.
*Deleted references to LEVEL = 4 and 5 microphysics which used to be a bin sub-model, but the code was never present in standard RAMS. Plus, we will have the HUCM bin model in future version of CSU-RAMS and will not need these old options.
*In rams_grid.f90 subroutine newgrid, I broke up a long do loop into several smaller do loops to allow the IFORT compiler to vectorize this loop more quickly at compile time for –O2. Otherwise it is really slow to compile (many hours).
*Reordered and added model checks in opspec.f90.

*In varf_update.f90 changed the varfile variable copying (future to past) from implicit do loops to explicit do loops. I had sometimes been getting segmentation faults at this point in the code when using the INTEL fortran compiler.
*Added the rtgt factor that influences vertical spacing over topography, to the aerosol deposition routine, and input ubmin for minimum wind speed similar to what is done in leaf3 because errors occur is wind speed becomes zero.
*Included a bounds fix check in "psxfer". Occasionally snow mixing ratio was becoming negative because of a computation precision difference. This produced a negative condensate error. We now limit ice-snow transfers to positive definite values.
*Moved call to "aerosol_init" in rdint.f90 to the top of the initialization routine, so that this only has to be placed once in the code. Added passing of aero_rhosol and aero_vanthoff arrays to mpass_init.f90 to make this possible for parallel runs. Otherwise I'd have to repeat the call to aerosol_init within the timestep routine AEROSOLS.
*Altered code to allow aerosol deposition for 1 or 2 moment micro and for moisture LEVEL = 0, 1, 2,or 3, and aerosols can be allocated and initialized for any moisture level. This is useful to look at dry deposition without micro being turned on. Also aerosol radiation can be used for LEVEL>=1. (Radiation in general requires LEVEL>=1.) Note that aerosols can be initialized in the model for any microphysics level and can be treated independent of the microphysics. This is useful to see how they are transported and diffused without being active. Then the user can choose the levels of activity (ie. radiation, deposition, microphysics).
*Put call to routine "AEROSOLS" in rtimh.f90 Timestep routine. Placed this ahead of call to RADIATE. This call will run the dust and salt source models if those are active. Also will run aerosol deposition for LEVEL<=2 if active. This replaces the call to dust and salt that was formerly in the micro driver. There is still a call to aerosol deposition in the micro driver for LEVEL==3.
*Created temp arrays for all micro budgets that are assigned in mic_misc.f90 under range_check and are copied back in mic_driv. This solves a problem of passing arrays that may not be declared.
*Bounds check fix to the ccncon and ccnmas array in cloud nucleation for aerosol removal.
*Bounds check fix in auto_accret and auto_accret_ice for several variables to access collection array.
*Found and fixed a bounds check error in rrad.f90 in the event that air temperature < 180K.
*Fixed bug in leaf3.f90 whereby surface top level temperature was not getting initialized if radiation and soil model were turned off. This allowed canopy temperature of water patches to approach absolute zero.
*Added RAMSIN flag ICHECKMIC which checks for negative condensate and Nans. If turned on, this calls routine "checkmicro", which is called after MICRO in rtimh.f90.
*Added RAMSIN flag IAEROPRNT which turns on or off printing out of initial aerosol profiles for a set I,J location.
*Added RAMSIN flag IAERODEP to turn on and off aerosol wet and dry deposition.
*Added RAMSIN namelist arrays for aerosol chemistry type, initial median radius, and solubility and removed these from being hard coded in micphys.f90.
*Added RAMSIN flag IAEROHIST to allow user to reinitialize aerosol concentration profiles on a history restart and addition of new grid at history restart. This is a useful option for aerosol sensitivity tests that are run from history restart after model spinup time. Code for initialization was added in rdint.f90.

*Added RAMSIN flag IPRNTSTMT to allow user to turn key print statement on and off for the standard output. These statements can be useful for debugging, but are clutter when trying to examine useful info in the standard output.
*In RAMSIN – ICLOUD, IDRIZ, IPRIS will no longer be > 5. Setting them to 5 will allow them to be 2-moment, but these are not tied to aerosol profile setup any longer. There are separate flags for these. The flag IAEROSOL in RAMSIN is set to 0 or 1, with 1 being ON. If on, then aerosol will be initialized regardless of microphysics level. Same for IDUST and ISALT.
*In RAMSIN – PPARM should now be always 0. This is not used. There is a separate variable CIN_MAX to put the max ice nuclei variable. This is similar to how this is done for CCN_MAX, GCCN_MAX, DUST1_MAX, etc.
*Default aerosol profiles are now set to decrease similarly to air pressure. The user needs to be aware of this and set their profiles as needed in mic_init.f90.

##
Moving from rams6_20131111 to rams6_20131130:
*Removed all code in RAMS related to NCAR graphics and removed the NCAR graphics from compile-time include.mk.opt and utils compile directory. Removed ncarg_dummy.f90, rams_wgrib.c, wgrib.c from utils directories.
*Removed interface.h since it is obsolete. Removed all "include interface.h" lines of code from all routines in the model.
*Removed a number of obsolete routines that are in the code but never call and used.
*Removed large sections of code that have been commented out since early RAMS 4.3.
*Added a few more print statements under the conditions that IPRNTSTMT=1.
*In mpi/par_decomp.f90 changed the "relspeed" array dimensions from 256 to 2560. This value represents the maximum number of parallel processors on which the model can run.
*Cleaned up urban canopy code a bit. Not sure how functional this code is in its current form.
*In core/rammain.f90, increased the dimension of "taskid" from 512 to 2560. As with "relspeed", this array size represents the maximum number of processor cores the model can use. This number made need to be increased again in the future.
*In memory/grid_dims.f90, changed "MAXMACH" from 512 to 1024. This is the maximum number of processor cores the model can run on. Memory is set aside for arrays containing dimension MAXMACH. This number is limited on 32-bit systems.

##
Moving from rams6_20131130 to rams6_20131204:
VERSION 6.0.00 (will also start using version numbers in addition to date stamps)
*Added several more IPRNTSTMT=1 conditions for printing info to the standard output.
*Changed dimensions of "taskids" in rammain.f90 to "maxmach", as well as to the variables "relspeed" in par_decomp.f90.
*Removed more obsolete or undeveloped code in the utils directories. Removed everything related to grib format and netcdf format. None of this was working. Perhaps was Tremback's work in progress, but non-functional. Deleted from utils: fenvgen.F90, gdf_write.f90, gdf_write_sfc.f90, gdf_write_upa.f90, cdf_utils.F90, grib_c.c, grib_utils.f90, lamcon.f90, project.f90, rams_read_header.f90. None of these were compiled or used! Also deleted inithis-vfm.f90 which was not compiled or used.

*Deleted vformat.f90. This required removing code from utils and model files related to the old .vfm file format. Deleted calls to subroutines vfinit, vctran, vcorec, vcirec, vforec, vwrt, vfirec, cscale, cfact, viorec, vwrti, viirec, cscalei. Removed this file from compilation objects.mk files.
*Deleted getvar.F90 and error_mess.f90. These were used together and only getvar.F90 called the routine in error_mess.f90. RAMS_getvar in getvar.F90 was only called by subroutines "recycle" and "first_RAMS". It appears that "first_RAMS" is never used unless you compile with RAMS, GRIB, or WRF support. To my knowledge, no one has ever used this function which is turned on if GUESS1st='RAMS' within RAMSIN. Even then, I think the code needed substantial work to be functional. This could be a dded back in, in the future if necessary. For now, we delete first_rams.f90. We remove getvar.F90, error_mess.f90, first_rams.f90 from compilation in object.mk files within /bin and bin/utils directories.
*Removing "first_RAMS" allows us to remove "GUESS1ST" from RAMSIN and all related namelist files. Removed reference to GUESS1ST in isan/asgen.f90, isan/asti.f90, isan/avarf.f90, isan/file_inv.f90, isan/isan_coms.f90, isan/isan_name.f90.
*Removed obsolete and unused subroutines in multiple files in the utils/2.4 directory files. This includes cleaning up charutils.f90, dateutils.f90, interp_lib.f90, numutils.f90, parlib.c, polarst.f90, rnamel.f90, utils_f.f90, and utils_c.c.
*Moved an_header.f90 and rconstants.f90 to the "memory" directory. From an_header.f90, I removed the subroutines "rams_read_header" and "an_setvar_info" since these were only used in getvar.F90 and recycle.f90. Further "rams_read_header" is archaic and we have better methods to do this throughout the code as is done in nud_update.f90 to read the header file.
*Went through the code and eliminated all reference to the old .vfm file format. Reworked code sections so that hdf5 is the only i/o format. This should make the code clearer and cleaner to work with. (Files affected were landuse_input.f90, geodat.f90, and varf_update.f90; also files cu_read.f90 and cond_update.f90 which are a special case since the code here is not functional and needs work; also inithis.f90, nud_read.f90, and nud_update.f90 for history file initialization and nudging are discussed more in the section below. There are still references to .vfm in non-functional code in the routines cu_read.f90 and cond_update.f90. This will be altered later.
*Added comments in fdda/cond_update.f90, io/inithis.f90, and io/recycle.f90 that mention these features are not fully functional. The model will stop. This will be made to work in the future.
*In rtimh.f90 added call to new subroutine "non_scalar_bc" at the end of core/rams_master. I added this routine to combine all the separate calls to set boundary grid points to non-scalar, non-advected variables. This includes calls to turb_bcond, leaf_bcond, rad_bcond, sfcrad_bcond, and THERMO for only BC's. This new call to THERMO replaces the call that occurs just before analysis file write in core/rams_master.f90. We call "non_scalar_bc" in core/rams_master.f90 where THERMO used to be called directly. Now we set all non-scalar BC's in this location.
*Placed a fix in the vapor flux routines to prevent vapor "rv" from becoming negative. Tests indicate that this can sometimes occur at upper levels where vapor magnitudes are quite small. In mic_vap.f90 in call to "vapdiff", alter the following line to keep "rv" positive.
rv(k) = max(0.,(rvstr(k) + sumuy(k)) / (1.0 + sumuz(k)))
then in "vapflux", altered the following line to "rv" positive
rv(k) = max(0.,(rvstr(k) + sumuy(k) + sumvr(k)) / (1.0 + sumuz(k)))
*Placed a fix in mic_nuc.f90 in "cldnuc" to set aerosol concentration and mass to zero if either becomes zero. This was added in the section where aerosols are removed after nucleation.
*In mic_driv.f90, altered the passing of variables to "psxfer" so that k1,k2 are passed for both snow and pristine ice. This change goes along with a change in mic_vap.f90 for "psxfer". The

k1,k2 values are passed in, then the min/max are used for transferring pristine ice or snow back and forth as needed. However, an error could occur such that NEW snow could be added to the snow category or vice versa without updating the k1,k2 values resulting non-conservations of condensate mass and associated aerosol mass. So, now k1,k2 values are updated and returned.
*Included an additional bounds check in mic_vap.f90 to "psxfer" variables dvap and dnum to make sure they are always of the same sign and non-zero. I included a check statement for this and removed old statements related to "xfer exceeded".
*Removed all reference to subroutine "hydro" and "hydrol" which were incomplete routines related to subsurface water tables. Call to these routines initiated from sfc_driver.f90. Removed the file leaf3_hyd.f90 and removed its name from objects.mk for compilation. Also removed all reference to the urban canopy model, which was also an incomplete project from ATMET. Our TEB model will likely replace this urban model anyway. Removed the RAMSIN flag "IF_URBAN_CANOPY". Removed all reference to variable "cdrag" for urban drag coefficients.
*Decided to remove everything related to cumulus inversion. This is still experimental and required unavailable offline code to convert observed precipitation into some sort of convective heating profile. Removed file cuparm/cu_read.f90, subroutine rconv.f90/cu_inv_tend. Also deleted all references to variables (maxcufiles, maxcugrids, cu_times, ncufiles, fnames_cu, itotdate_cu, ncuf1, cutime1, cutime2). From RAMSIN and model, deleted flags (IF_CUINV, CU_PREFIX, TNUDCU, WT_CU_GRID, TCU_BEG, TCU_END, CU_TEL, CU_TIL).
*In surface/ruser.f90, subroutine "eng_params", I removed all reference to obsolete variables named "IZFLAT", "NTOPSMTH", "IADVL", "IADVF". Also removed routines and/or calls to routines "toptsmth" and "toposmooth" as these were not used and may need additional work.
*In varf_update.f90 changed the varfile computation of "varpf" from implicit do loops to explicit do loops. I had sometimes been getting segmentation faults at this point in the code when using the INTEL fortran compiler. Seg faults did not occur with PGF90.
*Added GNU license header to every program file.
*Added condition such that soil moisture and temperature and snow data are only attempted to be ingested for an INITIAL run and not for HISTORY. Also added a check to see if varfile for particular grid is available. If not, perform default soil moisture initialization as usual.
*In mpi/mpass_feed.f90, added a print statement condition and removed subroutine "prtout".

For making the HISTORY initialization and nudging code functional:
*In init/rdint.f90 routine "initlz" for initialization, I started making modifications for making the HISTORY initialization option work. This will require more work to be fully functional.
*In io/inithis.f90 made many changes that read/interpolate a previous history file into initial conditions for a new run that can be on a similar grid or a different grid. Changes thus far allow the new initial output file to equal the history file that was read in. More work needs to be done.
*In lib/interp_lib.f90 and fdda/nud_analysis.f90 Adele make a code update to allow history initialization and nudging for simulations run in 2D.
*Changes to fdda/nud_read.f90 to allow reading of hdf5 history files.
*Complete redo of fdda/nud_update.f90 for nudging with history files. I do not think this this ever functioned, so it required a lot of experimentation and testing to make it work.

##
Moving from rams6_20131204 to rams6_20140114:
VERSION 6.0.01
*Installed Adele Igel's change to the advection code to include a monotonic flux limiter for vertical advection and improved positive definite scheme. This is currently functional in vertical direction only. Code is available for 3D flux limiter but will require a few modifications to function best in parallel. Requires message passing an additional layer of data along boundaries. Using the 3D version without such message passing changes is still mostly fine, but not exactly binary reproducible if you change the number of running nodes.
*Removed the following unused variables from the model related to the old TOPMODEL code and Leaf-3: sporo, ssand, sclay, sorgan, root, cmin, corg, cwat, cair, cka, ckw, romin, roorg, slcond, wtroot, tmin, xmin, ratio, romean. A number of these were in the analysis output header files and will no longer be there starting with this version. This requires an update to rcio.f90 in the model and revu.

##
Moving from rams6_20140114 to rams6_20140119:
VERSION 6.0.02
*Created offline script to search for subroutine and functions that are present but never called. Used this to begin removing unused subroutines and function.
*Removing more old code and standardized some code formatting for easy "grep" of certain words such as "module", "subroutine", "function", "interface", and "entry". Avoid using these key words when making comments so these are set aside for easy searches. Also, end every subroutine with "end subroutine", end every function with "end function", end every module with "end module", and end every interface with "end interface".
*Removed moving grids file nest_move.f90 including all subroutines therein.
*Removed subroutines acctimes1, acctimes2, kf_eta_init, ll_lc2, uvll_uvlc2, staprt, nest_interpolated_topo, prtlev, exstbuffs, mkstbuffs, dnswt_off, trtend, tridiff2, tridiff2orig, friclyr, gdf_missing_sfc, ifirstchar, isnumber, is_integer, uvll_uvlc, trueps60_ps_rot, trueps60_uevetouv, trueps60_uvtoueve, ps_ps, ps_ps_rot, ismax, rsifp, rslfp, aminmax, rowcolmn, ramran, cio_f8, fillsrc_den. Some of these routines are lambert conformal grid projection routines that could be useful later. Can be accessed from original files.
*Removed RAMSIN variables "GRIDU" and "GRIDV" used for moving grids. Removed these from everywhere in the code. Also removed variables "dimove" and "djmove" from memory directory files.

##
Moving from rams6_20140119 to rams6_20140122:
VERSION 6.0.03
*Removed ADAP cut cell code since this was only partly function and has a long way to go to be functional. If you need this vertical coordinate, you will need to move to the OLAM model.
*Removed ADAP variables lpu, lpv, lpw, volt, volu, volv, aru, arv, arw.
*Removed IF_ADAP from RAMSIN and all its dependencies.
*Removed all ADAP fortran files and subroutines that are typically named "something_adap".
*Had to remove K level offsets due to lpw, lpu, and lpv.
*Also removed routines lpuwv_init, ctrlvols.

*Removed “entry” statement pr_hystatic_t and gave it its own subroutine.
*Removed functions “os”, “tsa”, “tw”, “esat”, “w”, and “tmr”. It is not immediately clear what these were intended for. There is no info on these functions.
*Cleaned up the parlibs.c somewhat.
*Standardized the code with the following characteristics:
1. “implicit none” used in ALL modules, interfaces, functions, and subroutine in fortran. This is written beginning in first column.
2. Subroutines are headed by syntax such as: “subroutine steve ()”. Spaces are included and this starts in first column. Subroutines end with “End Subroutine”.
3. Functions are headed by syntax such as: “integer function steve ()”. Spaces are included and this starts in first column. Functions end with “End Function steve”.
4. Modules are headed by syntax such as: “module steve ”. Spaces are included and this starts in first column. Modules end with “End Module steve”.
5. Interface statements are headed as: “interface”. This starts in first column. Interfaces end with “End Interface”.

##########
Moving from rams6_20140122 to rams6_20140126:
VERSION 6.0.04
*Removed global RAMS setup since this was never really functional. This required removing nesting/hemi.f90.
*Removed from mem_grid.f90: nhemgrd2, nhemt, nhemu, nhemv, and all variables with dimension “maxhp”.
*Removed NESTZ2 and NSTRATZ2 from RAMSIN and all code since these refer to the 2nd hemispheric grid that we have removed.
*Removed global dimension MAXHP.
*Changed all instances of NESTZ1 and NSTRATZ1 to NESTZ and NSTRATZ, this included changing related variables as well.
*Added Meesters et al. (2008) topographic gradient surface flux fix.
*Removed a few additional deprecated declared variables: nrzflg, lev4bins, lev5bins.
*In core/raco.f90 in subroutine prdctv, put all contents within the “jdim” if statement for 2D simulations so that V is not predicted at all in 2D simulations.
*In init/rhhi.f90 in subroutine arrsnd, I include a statement that sets input sounding V-winds to zero in the case of a 2D simulation.
*In isan/astp.f90 I made a few modifications for ingesting global lat/lon datasets since it was not accepting certain datasets due to missing grid navigation. This was fixed in the dataprep code and modified here accordingly.

##########
Moving from rams6_20140126 to rams6_20140130:
VERSION 6.0.05
*Reworked the compilation Makefile routines to combine 4 short files into 1 that the user needs to modify. Also combined the RAMS utilities directories and compilation into the full model compilation to eliminate this unnecessary extra step at compile also. It also helps to centralize the source code.

*In testing this version I stumbled upon random segmentation faults. In some instances, the computer stack overflows. So, random looking segmentation faults are sometimes not real. You can avoid this issue by increase the stack size. On Linux type the command "ulimit –s unlimited". This HAS to be done to run OLAM or it seg-faults on the first timestep. I am wondering now if this was been the source of random seg-faults in all versions of RAMS since the early days of 4.3. We used to get random seg-faults when create surface or varfiles. It is possible these were not really code errors, but rather, stack overflow. This is being noted in the user guide. This issue could be resolved with use of allocate/deallocate within subroutines that currently declare many local 3D variables on the stack.
*Changed the HDF5 calls to H5Dopen and H5Dcreate to the HDF5 version 1.8 API format.
*Fixed bounds check issue in leaf3_init.f90 for the soil data ingest section. It was looking for varfiles in simulations where varfiles do not exist.
*In mic_nuc.f90 subroutine cldnuc, a statement was input to force cloud and drizzle droplets to nucleate at their smallest allowed size.
*Removed everything related to addition of new scalars via RAMSIN flag NADDSC. This included removing the file memory/mem_scalar.f90 and all calls to added scalars via "scalar_g" and "scalarm_g" arrays in memory/alloc.f90. In memory/mem_tend.f90, removed several loops with reference to additional scalars and tendencies of "scalar_g" and "scalarm_g" arrays.
*Removed "proc_type" equals 3 and subroutine "dealloc_all" as these were related to dynamic load balancing which is a non-functional feature that was removed previously.
*Removed call to THERMO in io/ranlavg.f90 and replaced with call to "non_scalar_bc" since this seems to do a better job of setting boundary conditions for non-scalars.
*From rcio.f90/commio: Removed many variables that do not appear necessary to be sent and recorded in the analysis header file. I retained the variables necessary for history restarts and REVU. Others were removed, but I added GNU and IAEROCHEM for use in future REVU. User can do a diff command on new and original files to see the changes.
*From history_start.f90/history_start: Removed header files read-in of many variables that are not necessary for history restart. Only retained the needed ones. User can do a diff on new and original files to see the changes.
*Removed use of lambert-conformal projection. This feature was non-functional and was not recommended by ATMET to use at this point in time. Removed option IHTRAN=2, call to "grdspc_lc" from init/rams_grid.f90. Removed file init/lllc_utils.f90 and its compilation in bin/objects.mk. Removed all instances of RAMSIN variables "stdlat1" and "stdlat2". Removed "lc_ll" in lib/map_proj.f90 as it is unused.
*Fixed a limitation in the Kain-Fritsch cumulus parameterization that did not allow for more than 50 vertical model levels. Set the new default to 500 levels. Ideally this code would be substantially reworked to be more inline with RAMS memory and allocation structure.
*Removed variables "iinput", "iopunt", "iversion", "runtype_save", "naddsc" from the model.
*Changes every instance of "runtype" to "trim(runtype)" to prevent future conflicts.
*Removed RAMSIN and model variables XLITE, YLITE, ZLITE as these were never used.
*Removed the ":hist:" vtable type flag from usage in the model. This is mostly obsolete since we do not have separate history and analysis files. The analysis files are the history restart files.
*Removed routine and call to "filltab_scratch" as nothing is done for this.
*Removed passing "rmin", "rmax" thru MPI as it does not appear at all necessary.
*In mic_init.f90 where collection tables are created, I added a statement that forces the read-in of the collection table, even if it is being created currently. This fixes a number truncation

difference that arises between creating table values in-line and reading them from a file. These differences lead to different solutions and prevent binary duplication.

VERSION 6.0.06
*Changed syntax of call to subroutines to always begin with "CALL". This is done such for ease of case sensitive grepping and code tracking. A call should be like "CALL name (variables)" with spaces included.
*Removed subroutine "wet_scavenge_chate" and its commented out call, as it was never used.

VERSION 6.0.07
*Got "shdf5_info" working for history restarts. This just provides information on arrays and dimensions, but could also be quite useful in future development.
*Since we cannot test other machine dependent types, I have removed the "#if defined" statements to all types but PC_LINUX1, which is a standard Linux OS and is the primary system of choice. I combined all system dependent routines into rsys.F90 so that there is only 1 fortran file to modify for additional systems in the future. The C file utils_c.c and utils_sub_names.h would also need to be modified for additional machine types. But these are the only 3 files to modify.
*Modified many string lengths from 80 to 128 characters so these do not get so easily truncated when having to put long path lengths into RAMSIN. Otherwise model would crash and say that it could not find a particular file.
*In gridset.f90 I removed reference to variables "ngra" and "ngrb" and their condition statements, as these were tied to the old non-functional moving grids.
*Checked and reorganized all function calls to that are easily traceable. Renamed a few of them so we can grep for them more easily. Set all subroutine and function names and calls to lowercase for easier searching and matching. Made sure every subroutine that uses and function declares it as an "external".
*Removed function "LC" as it is never used and removed routine "dnswt" as it is never used.
*Removed call to "deallocate" routines for the global pointer variables such as "dealloc_micro" since these are never used, and system will clear memory if model crashes or finishes.
*Set all subroutine and function names and calls to include () at the end even if not variables are passed. This is just done for consistent grepping and searching.
*Set up the subroutine mass_flux_bc to be functional in the model for sequential simulations. This is off by default but can be turned on at the bottom of rtimh.f90.
*Turned entry statements "azero" and "azero2" into separate subroutines to get rid of entry statements.

##
Moving from rams6_20140130 to rams6_20140410:
VERSION 6.1.00
*No changes here. Just freezing this version of RAMS after a substantial amount of work, bug fixes, modifications, science changes, structural changes, and code deprecation. This will move us from versions 6.0 to 6.1 and forward.

##
Moving from rams6_20140228 to rams6_20140410:
VERSION 6.1.01 (Merged REVU back into RAMS source code to share files)
*Added a few variables need to merge the REVU code with RAMS including "anal_table" and "maxrevu".
*Comment out additional call to "H5close" that interrupts REVU functionality but does not really seem necessary for RAMS.
*Fixed the output of the "deltazn" variable in the analysis header files.

VERSION 6.1.02
*Used gfortran compiler with –Wall option to turn on all compiler warnings. Then used this output to eliminate from the code all unused but declared variables. Also eliminated all potentially uninitialized variables.

VERSION 6.1.03
*Fixed reading in of isentropic/sigma-z variable initialization files in the event that these are output prior to creation of the actual variable initialization files or "varfiles". This would be most useful for development purposes where the user needs to examine the output created during the isentropic stage of the processing of pressure level grids and conversion to RAMS isentropic and sigma-z levels.
*Also fixed a pointer initialization in mpi/mpass_full.f90.

VERSION 6.1.04
*Made a few alterations to subroutine non_scalar_bc to allow it to be called from master and node and pass the correct local indices. This required passing in the used indices.
*In core/model.f90 subroutine par_model, changed the use of "nmach" to "nmachs" in all calls to par_ready subroutine. Made changes to allow "model" and "par_model" to have a more similar structure that makes them easier to compare. This required changing the location of the time update calculation and relocating the call to compute the means in subroutine anlavg. Added passing of local domain grid point array sizes.
*In memory/alloc.f90, I made modifications to allow declaring 2D grid characteristic arrays so that these can be written to the MEAN files. For example, we really need TOPT in the MEAN files for computing many of the variables in the REVU post-processor.
*In io/anal_write.f90 subroutine anal_write, made a few fixes to output the correct fields for MEAN state files and BOTH files.
*In mpi/mpass_full.f90 subroutine node_sendanl, added print statement to see passed variables, and in subroutine master_getanl, added a condition statement to update number of received variables for MEAN state files.
*In io/ranlavg.f90 subroutine anlavg, added print statement to see list of averaged variables. Modified call to subroutine non_scalar_bc to work on parallel nodes. This required passing in the local node array sizes rather than absolute domain dimensions.
*In mpi/rnode.f90 subroutine rams_node, I moved around call location to subroutine anlavg and added the appropriate passing variables associated with subdomain sizes.

VERSION 6.1.05
*Cleaned up the passing of unnecessary variables among the routines that read in the RAMSIN namelist.
*Eliminated all remaining common blocks.
*Fixed an error in the history restart read of NDVI values.
*Fixed a runtime and history restart error related to the Chen/Cotton radiation routine that improperly assigned K levels regardless of topography.
*Further streamlined REVU including make the syntax for subroutine and calls conform to that now used in RAMS.

VERSION 6.1.06
*Made a fix so that dust-in-hydrometeor variables are not declared and used unless both IDUST>0 and dust tracking flags are turned on.
*Transferred a REVU file inventory "MAX" variable from rcommons.f90 to grid_dims.f90.
*Made a fix to collection of cloud droplets. In past versions, lookup tables in subroutine "make_autotab" in file mic_tabs.f90 started with a minimum mixing ratio of 1.e-5 kg/kg while the application of these lookup tables started at 1.e-12 kg/kg. This led to collection of cloud droplets that was too rapid. In warm rain conditions this would create rain too quickly. The solution was to increase the number of lookup table value of "nrrcr" from 10 to 30 in micphys.f90 and lower the minimum mixing ratio threshold to 1.e-12 kg/kg in subroutine "make_autotab."
*Continued modernizing and streamlining the REVU package files, including fixing some of the LEAF patch variables. Substantially streamlined the dumpout or text output option. This will now be the "TEXT" option. Made all REVU files "implicit none" compatible and eliminated unused but declared variables. Fixed the section in "iplt.f90" that determines min and max dimensions based on variable type and output ranges in REVU_IN.
*Reorganized some REVU HDF5 output code to make things a bit cleaner looking, which should help with future updating. Made things such as subroutine calls and end and header conform to that syntax used in the rest of RAMS.
*Started organizing the code to place all "MAX" array limit variables in the same file named memory/grid_dims.f90. This way, future changes to max array sizes and be done in a single location for all purposes (ie. RAMS, REVU, ISAN, ODA, etc).

VERSION 6.1.07
*Made common variables "strl1" with character length 128 and "strl2" of 256 length that can be used throughout the model for declaring max string length.
*In isan_name.f90, rnamel.f90, rname.f90, removed "maxrec" and modified call to "findgr".
*Updated grid_dims.f90 as the central location for all "max" variables in the model code.
*In ref_sounding.f90, io_params.f90, var_tables.f90, mem_oda.f90, mem_mksfc.f90, mem_varinit.f90, isan_coms.f90, isan_name.f90, rname.f90, landuse_input.f90, removed "max" variables to be moved to grid_dims.f90.
*Few minor syntax changes to REVU.
*Streamlined and standardized how RAMS and REVU read in command line arguments and the runtime command execution.
*Standardized how "character" array lengths are declared. (There was a mix of old and new fortran syntax.)

*Changed several RAMSIN flag names to be shorter for code syntax purposes. These changes are made in all relevant files throughout the model:
1. wt_nudge_grid to wt_nudge_g
2. wt_nudgec_grid to wt_nudgec
3. roda_zfact to roda_zfac
4. oda_upaprefix to oda_upapref
5. oda_sfcprefix to oda_sfcpref
6. frqstate_keep to frqst_keep
7. itrackepsilon to itrkepsilon
8. itrackdust to itrkdust
9. itrackdustifn to itrkdustifn

VERSION 6.1.08
*Added a condition statement in sedimentation routine "sedim_trubin" to keep the array condition integer "idensrtgt" from going out of bounds. Should stay between 1-40.
*Added RAMSIN variables "IAEROLBC" and "AEROTAU". Setting grid-dependent variable IAEROLBC to 1 will reset the initial aerosol vertical profile to the initial conditions along the lateral boundaries. This will function for zero gradient boundaries only (see LSFLG in RAMSIN namelist documentation). This would typically only be used on an outer domain, but could be used for nested grids if certain aerosol sources are needed. The grid dependent flag AEROTAU is a number in seconds that is used to compute the inverse time weighting. The time weight gets computed as (timestep/AEROTAU) and lets the user control the strength of resetting the aerosol profile. If AEROTAU = timestep then full aerosol resetting occurs; if AEROTAU > timestep, the resetting will be more gradual.

VERSION 6.1.09
*Removed use of RAMSIN variable DELTAY since we should always have DX=DY in the model. Some routines (ie. mxdefm) actually only use dx or dy and not both in certain calculations. So a difference would produce inconsistent results. So, throughout the code I removed DELTAY and replaced it with DELTAX as needed. DELTAX is now simply the horizontal grid spacing.
*Removed use of NSTRATY which was the Y direction grid nesting ratio. Our nest ratios should always be the same in X and Y except for 2D simulation where Y nest ratios would always be 1. So I removed NSTRATY from RAMSIN and assign it in "opspec" where NSTRATY=NSTRATX except in the 2D case.
*Got rid of snow ingest variables "snow_water_sfc" and "snow_depth_sfc" and just ingest and use these products in the already declared variables "snow_mass" and "snow_depth".
*Fixed a bug related to ingest of snow depth variable. I had been assigning snow_depth equal to 5 x the "snow_mass" value but then unintentionally adding the ingested "snow_depth". We should be using one or the other and not both. Use 5x option when snow is not ingested and use the ingested "snow_depth" when available.
*Updated revu/hvlib.f90 and revu/textout.f90 to rename "snow_water_ps" to "snow_mass_ps". And updated revu/textout.f90 to apply correct variable string length for "snow_mass_ps" and renamed gempak abbreviation from SNOC to SNOM.

VERSION 6.1.10

*In applying history initialization, found that “leaf_class”, “patch_area”, and “soil_text” need boundary assignments made before the first analysis write and first timestep. So, these were added to mksfc_sfc.f90 subroutine “sfc_read”.

*In ndvi_read.f90 subroutine “ndvi_update”, removed the boundary assignment for veg_ndvic and veg_ndvip, since these are not necessary as “veg_ndvif” is the value stored in the NDVI surface files.

* In rbnd_nonscalar.f90 subroutine “leaf_bcond” removed setting boundary conditions for “patch_area”, “leaf_class”, “veg_ndvif”, and “soil_text” since these are now set when the surface data files are read in during a model start.

*Removed use of TOPTA and TOPMA since these are redundant. TOPT = TOPTA and TOPM = TOPMA. Not sure why these were in the model since they were not used. These are also eliminated from model analysis files and REVU post-processing. Where TOPTA and TOPMA were used, they have been switched to TOPT and TOPM.

*In rdint.f90 subroutine “initlz” for history restarts, I removed calls to “grid_setup” and “make_sfcfiles” since these are already called for any model start from subroutine “rams_master”.

*In routine “inithis” began making incremental fixes and updates, but this is not yet functional.

*In revu code, changes “topo” to “topt” for consistency and removed “topta”.

VERSION 6.1.11

*In rbnd_nonscalar.f90 subroutine “leaf_bcond”, added boundary setting for variable “veg_albedo”. This had been left out for setting bc’s in the J direction.

*Added RAMSIN variable “NVEGPAT” to I/O for various nudging options and for history initialization and grid structure comparisons.

*Commented out call to “prgintrp” for history initialization since we do not just want to interpolate from new parent grid to new nested grid. Rather we will interpolate from all relevant history grids to all new grids.

*Made further changes to history initialization in subroutine “inithis”, including removal of unnecessary subroutines based on new method of interpolation to new grids. But this is still a work in progress.

*Switched array dimensions for “soil_text” in surface files to be (i,j,k,patch) rather than (k,i,j,patch) in order to conform to the rest of the model output files.

*Changed naming convention of “topo_z0” to “topzo” in topography files so that these conform to the rest of the model use of topographic roughness.

*Changed naming convention of “veg_ndvi” to “veg_ndvif” in NDVI files so that these conform to the rest of the model use of NDVI.

*Added “veg_albedo” in REVU output options. Removed “scalar1” and “scalar2” from REVU output options since these are obsolete.

*Changed use of ISFCL conditions statements to only allow values of 0 and 1.

*Changed NVGCON max to 20 in “rname.f90”. Changed vegetation class loops from (1,nvtyp) to (0,nvtyp) since types range from 0-20. Likewise, changes array dimentions of NVTYP variables to 0:NVTYP.

*Removed top level initialization of “veg_rough” and “veg_albedo” in subroutine “sfcinit_nofile” since these values are overwritten in the following call to subrouttine “vegndvi” that sets more realistic values.

VERSION 6.1.12 (HDF5 array swap in this version)
*Cyclic BCs were fixed to produce duplication whether run sequentially or in parallel. This was done by Steve Herbener and required a large overhaul of numerous MPI routines calls to cyclic boundary conditions.
*Updated the radiation BCs in "rbnd_nonscalar.f90" to include 'rshort', 'rlong', and 'cosz'.
*Included new integer variable "INITORIG" that tracks the original simulation type of INITIAL = 1 or 2 (ie. horizontal homogeneous or variable initialization). If we are running history initialization (INITIAL=3) we still need to know the original simulation type for many applications. The original value of INITIAL is stored in INITORIG and retained in output header files.
*In coriolis.f90 subroutines "corlsu" and "corlsv", put in condition statement to exit the routine if running INITIAL=2 or INITIAL=3 and INITORIG=2. Needed for history initialization.
*In kf_rconv.f90 and rconv.f90 I forced the KF or KUO convective parameterizations to only reset arrays to zero if time=0 and NOT a history initialization. Also set these parameterizations to run at times based on the same modulus calculation as we use to call radiation. Needed these to make history initialization match history restart when grid are the same.
*In rdint.f90 subroutine "initlz", the variable INITORIG is zeroed and then set to the initial value to INITIAL if we are running INITIAL = 1 or 2. Also put in statement to only call BUBBLE if INITIAL=1.
*In history_restart.f90 subroutine "history_start", added i/o statement to include "INITORIG" in the output header files.
*Made many modifications to inithis.f90 to get history initialization functioning.
*In opspec.f90 subroutine "opspec3", included statement to allow Rayleigh damping for INITIAL=3 if INITORIG=1 (ie. HH simulation).
*In rcio.f90 subroutine "commio", included INITORIG in i/o list of variables.
*In rprnt.f90 subroutine "prtopt", included statement to print initial sounding for INITIAL=3 if INITORIG=1 (ie. HH simulation).
*In hdf5_utils.f90 subroutines "shdf5_info" and "shdf5_orec", included the array switch for file output necessary for using Parallel HDF5, which is our future plan for RAMS in order to use the distributed memory version that Steve Herbener is developing. The array swap will allow direct access file opening by many software packages.
*In interp_lib.f90 subroutines "gdtost2", "htint", and htint2" made a few modifications related to horizontal and vertical interpolation. These changes allow for more precise computations among code bases and machines that offer different precision depending on arithmetic order.
*Minor change to output statements in grid_struct.f90 to be more readable.
*Updated routine "check_real" to compare arrays based on old and current grids rather than just the old grids. Also commented out some sections of subroutines "cond_update" and "nud_update" so they will compile; these routines are not yet used.
*Added variable INITORIG to memory in mem_grid.f90.
*Changed "igrid_match" variable to be 2D to compare multiple grids from history and current domains where necessary for history initialization and nudging.
*Added 3D precip rate arrays such as "pcpvr" to the negative adjustment routine so that these arrays are set to zero is associated hydrometeor mixing ratios are zero. Without this fix, it was possible to have precip rates non-zero where no hydrometeors exist, which could cause problems with aerosol precipitation scavenging.

*In mpass_dtl.f90, added variable INITORIG to the put and get routines for parallel processing and updated the nwords count.
*In rad_driv.f90 subroutines “radcomp” and “zen”, switched to a more precise and consistent computation of julian day that is needed for standard runs and history initializations runs to be consistent over time where the date changes in the simulations. Also included a more consistent day hour diagnostic that functions to find the time of day regardless of model start time. This was needed for history initialization to work precisely.
*Fix to gaussian bubble implementation in subroutine “bubble” in ruser.f90 to resolve a bubble radius versus diameter discrepancy.
*In turb_diff.f90 subroutine “truhor”, fixed a bounds check error that sometimes occurs related to calling and running subroutine “topobnd” for 2D simulations where topography does not really exist in the Y direction. The fix was to only allow the Y direction computation if simulations are 3D.
*Minor change to the default idealized mountain topography routine in ruser.f90 where the user specifies a mountain shape for idealize mountain flow simulations. The default setup is more simple and universal.
*Small print out changes to subroutine “textout” in textout.f90 to print null terminator strings on blank lines in gempak text format.

VERSION 6.1.13
Summary: This version largely improved the history file initialization option. It also removed the history file nudging, condensate file nudging, and recycle file initialization options in order to replaced these by creating history-varfiles that contain history nudging grids and condensate nudging grids. The “recycle” option for LEAF variables was made to work within the history initialization routines with the RAMSIN flag IPAST_LEAF, but it does not require a full history initialization.
*In rams_master.f90, included the option to create history-nudging files from history file data. The runtype for this is “MAKEHFILE”. It will make history nuding files for as many history files that are present in the specificed directory under “VAR_HFILE”. Also removed calls to nud_read and cond_read since these subroutines are removed. All nudging (traditional varfile, history, and condensate) will be done via varfiles with a single set of varfile interpolation times.
*In nud_analysis.f90, fixed condensate nudging weights to be more appropriate. This should be experimented with for best future usage. Switched out nud_type so that this is either 0 or 1 (ie. no nudging or nudging varfiles).
*In varf_update.f90, added condensate nudging grids to be read in from varfiles. This is only possible from history-varfiles using MAKEHFILE to ingest history file data.
*In rdint.f90, modified code to fix nudging type options, history initialization and partial history initialization using the “recycle” option. The recycle subroutine was eliminated and the recycle flag “IPASTIN” was renamed to “IPAST_LEAF”. If IPAST_LEAF = 1, but history initialization is not turned on, then history initialization is called just for the LEAF variables. This can be done for the same grid configuration or a new configuration similar to the history initialization capabilities.
*In history_start.f90, added a check section and statement that warns the user if they are trying to restart a run at a time when all model grids are not present. This is not allowed.
*In inithis.f90, continued refining the code and added capability to work with the recycle feature that currently works for LEAF variables.

*Changed io_params.f90 and rname.f90 to reflect changes related to IPAST_LEAF recycle option and history and condensate nudging flags. We no longer need a RAMSIN flag to hold the name of history or condensate nudging files. This is built into the history varfiles.
*In asgen.f90 added a variable to hold condensate nudging arrays to place in isan varfiles. Also created new routine "nudh_driver" to run instead of "isan_driver" in the event we want to create history varfiles rather than traditional varfiles based on pressure level data grids. The history varfile driver works similar to history initialization and can create history varfiles for a grid structure that differs but fits inside a history grid. History varfiles will contain the standard 5 upper level fields as well as condensate fields for nudging. They will not contain soil moisture, soil temperature, and snow fields that the standard varfiles contain since this is not compatible with soil initialization. If these are needed, the user would need to do history initialization or LEAF recycle.
*Eliminated the fortran files: recycle.f90, nud_read.f90, cond_read.f90, nud_update.f90, and cond_update.f90 since these are obsolete with the new history varfile option. Also throughout the code I eliminated references to condensate and history file nudging times since these no longer exist.
*Moved subroutine "check_real" to numutils.f90 and expanded it to do better checking between history grids and current grids for history initialization, history varfiles, and/or recycle features.
*Few other minor naming convention changes to make some code more distinct.

VERSION 6.1.14
*Applied lateral boundary conditions for precipitation rate in file bc/rbnd_nonscalar.f90 subroutine "micro_bcond". Interpolation between grids can give non-zero precip rate values on grid boundaries that interfere with visualization and contouring.
*Added condition statement in opspec.f90 to produce error if user attempts to set RAMSIN variable NVEGPAT to something greater than NPATCH-1. Inappropriate settings will result in segmentation fault and potentially erroneous surface files and erroneous results from the LEAF surface model.
*Made modifications to file micro/mic_misc.f90 subroutine "adj1" that controls the microphysics negative adjustment or positive definite checking scheme. We had seen grid boundary points containing positive mixing ratio and zero number or positive number and zero mixing ratio for some hydrometeor types. This inconsistency was fixed. Further, if mixing ratio is adjusted in this scheme then number concentration is equivalently adjusted in call to "ae1mic".
*Added REVU variables for density potential temperature "theta_rho" and buoyancy relative to liquid condensate loading "buoyancy_liquid". At this time, these should be used with caution and are experimental. This impacted files hvlib.f90, rcomp.f90, and textout.f90.

VERSION 6.1.15
*Added code related to the Loftus et al. (2008) convergence generation code for idealized initiation of convection. This code is in surface/ruser.f90 subroutine "conv_forcing" and it computes U and V tendencies that generate convergence. The code is called until either a certain time is met or a certain W threshold is met. To get the domain maximum W when in parallel, a computation of domain max "vertvel" was added to core/modsched.f90. Further, in the non-DM code I added calls to routines "master_getvertvel" and "master_putvertvel" in core/model.f90.

*Computation of W budgets in subroutine “boyanc” was reworked to prevent passing of unallocated arrays, which sometime caused segmentation faults. Also used allocation statements to limit stack memory requirement.
*In file core/radvc.f90 subroutine “fluxlimits”, changed 3D locally allocated variables to allocatable variables, then used allocate/deallocate statements to keep extra memory off the stack. This should help limit occurrence of segmentation faults related to stack memory overflow.
*In core/rtimi.f90 subroutine “tend0”, added call to convergence routine “conv_forcing”.
*In cuparm files kf_cuparm.f90 and kf_driver.f90, I removed passing of micro 3D arrays. Rather these are now accessed via the “use micro” statement. If they are allocated, then they are copied to allocatable local arrays to be used. This also prevent segmentation fault related to passing unallocated variables. Made a few other minor changes to Kain-Fritsch code related to microphysics levels. We have microphysics LEVEL=0-3 in this version and not 4-6 which used to include the primitive dumpbucket scheme. Lastly, in cuparm/kf_rconv.f90 I added a few condition statements related to hydrometeor mixing ratios that may not be allocated.
*In init/rdint.f90 I added calls to “init_tracer” which is used for custom initialization of extra scalars or tracer variables. They can be on any grid and there can be any number of these. Subroutine “init_tracer” was added to micro/mic_init.f90.
*In io/anal_write.f90, got rid of use of pointers to prevent potential memory issues.
*In io/opspec.f90, added some condition statements for use of the convergence forcing code.
*In io/rname.f90, added RAMSIN variables related to converging forcing and tracer code.
*In isan/asgen.f90, eliminated use of “nullify” statements. Use deallocate instead.
*In lib/rnamel.f90, changed the length of the read statement for RAMSIN input to be longer. This allows long path names in RAMSIN and prevents skipping long names at read time.
*In isan/isan_coms.f90 and all memory fortran files, got rid “pointer” statements, and replaced with “allocatable” statements. Also got ride of “nullify” and changed to “deallocate”.
*Added memory/mem_tracer.f90. This replaces old use of mem_scalar.f90 which was not quite working properly before. Added allocation of tracer arrays in alloc.f90 and “use mem_tracer” where necessary. Updated calls to tendency allocation arrays to add in the tracer tendency arrays.
*Eliminated calls to “c03” and “pc03” in mic_driv.f90 since these are generally unnecessary. Rather I directly call “enemb” to adjust number concentrations after cloud and ice nucleation. Also eliminated the associated subroutines from within mic_misc.f90.
*In micro/mic_drive.f90, changed sedimentation tables from pointers to allocatables.
*Fixed the fall speed coefficient for cloud and drizzle in mic_init.f90 from 3173 to 3173.e4. The new coefficient is appropriate for velocity units of m/s.
*In mic_init.f90 I eliminated several other unused but computed variables and their associated declarations in micphys.f90.
*In micphys.f90, declared RAMSIN variables related to convergence forcing code.
*In mksfc/mksfc_driver.f90, added deallocation of “sfcfile_p” arrays. Perhaps this fixes the random segmentation fault that sometimes occurs when making surface files.
*In mpi/mpass_dtl.f90 for the non-DM code, I added routines “master_getvertvel”, “master_putvertvel”, “node_getvertvel”, and “node_putvertvel” in order to pass around the domain max vertical velocity needed for the convergence forcing code. In the DM version of the model, I only had to add subroutine “set_vertvelmax_allnodes” to accomplish the same task.
*In mpi/mpass_init.f90 I added variables related to convergence code and tracer code that needs to be passed to the nodes. Also updated “nwords” for buffering.

*In mpi/node_mod.f90, changed pointer to allocatable.
*In mpi/rnode.f90, changed associated variable to allocated variable. In the non-DM code, I added calls to "node_putvertvel" and "node_getvertvel". In the DM code, I added call to "set_vertvelmax_allnodes" to pass the domain max vertical velocity information.
*In rad_drive.f90, got rid of pointers related to cloud water or no cloud water being passed into subroutine "radcomp3".
*In surface/ruser.f90, modified Cosine-squared bubble routine to also allowed 2D bubble for 2D convective idealized simulations.
*In turb/turb_ke.f90, changed associated arrays to allocated arrays.
*In turb/turb_k.f90, eliminated some of the potentially troublesome pointers that are now declared as allocatable arrays in the memory routines.
*In init/rinit.f90 subroutine "fldinit", removed calls to "azero" since all of these fields have already been zeroed out at variable allocation time. Altered the associated calls to "fldinit" within file init/rdint.f90.

VERSION 6.1.16
*Made a few cosmetic changes to the code in order to more easily track and cross-reference subroutine and function declarations with their associated calls.
*Removed additional unused subroutines from the model.
*Altered some informational print statements so that duplicate print statements do not occur when running the model in parallel.

VERSION 6.1.17 – June 30, 2015
*Updated binned riming collection kernels to be directly compatible with newer set of fall speed power laws.
*Modified subroutine "col3" which treats collision-coalescence between rain and each ice species. Applying the Milbrandt-Yau (2005) approach, via Adrian Loftus' adaptation to earlier versions of RAMS, to compute density of coalesced rain-ice so we can appropriately determine the destination category. Old RAMS version simply placed all mass and number into hail as the destination category.
*Updating of "col3" works well with the binned riming approach to ice collecting cloud droplets. The use of both options now helps retain mass and number in the graupel category since not everything is transferred to hail. Since this works well, I modified the destination category in binned riming subroutine "auto_accret_ice" to perform similar to the original riming approach in subroutine "col2". These modifications appear to work quite well together and provide good partitioning among ice categories following ice-liquid collisions.
*Had to eliminate the micro budget term related to mass transferred to hail as the destination category for rain-ice collisions since the destination category can now be variable.
*Added tracer output options in REVU.

VERSION 6.1.18 – Sep 28, 2015
*From detailed testing of ATEX case (Saleeby et al. 2015) it was decided to reset cloud mass-diameter coefficient back to 3173.0 from 3173.e4, and reset cloud nucleation droplet size to the diagnosed value rather than the smallest allowed droplet. Also turned off the advection monotonic flux limiter until further testing can be done.

*Added a conditional if statement in mic_driv.f90 after the call to "icenuc" for the lines of code that determines the new depth, or K levels, of the pristine ice layer. This was causing segmentation faults when ice was turned off in the model.
*Eliminated boundary condition options IBND and JBND =2 and =3 since these do not actually exist in the model. IBND=JBND=4 for cyclic boundary conditions is now IBND=JBND=2.
*Updated the "reflectivity_all" option in REVU to include all hydrometeor species. Added an adaptation for ice species to assume spherical particles; this adaptation takes the non-spherical mass-diameter relationships and computes an "alpha" coefficient assuming spheres. This is done for each hydrometer type at each grid cell, and is diameter dependent. Eliminated the individual hydrometeor reflectivity. These could be added back within the section for full reflectivity.
*In REVU, placed a lower limit on output data values. Anything with -1.e-20 < X < 1.e-20 is set to zero so that we do not have to output ultra-high-precision values. Some post software cannot handle these really small and high precision numbers that ultimately do not matter.
*In REVU, added a variable called "empty3d" in the event that post processing software requires place holder variables. This is necessary in Quickbeam radar simulator, in which a certain order of variables is required. If your simulations, for example, does not have drizzle turned on, then this variable is not available in REVU; but Quickbeam requires that the drizzle category not be empty. In this case we replace drizzle with "empty3d" which contains all zeros.
*Fixed declaration of NGBEGUN to integer in io/inithis.f90. This was causing model seg faults.
*Fix to file name output in REVU for the TEXT output option.
*In hvlib.f90 switched to allocate/deallocate of 3D variables need for reflectivity calculation. Use of local 3D declarations overwhelmed the stack memory on some systems.
*Updated/fixed surface wind convergence forcing code in ruser.f90 in order to allow user to input divergence rate in RAMSIN rather than the amplitude. (via Leah Grant and Sean Freeman)

VERSION 6.1.19 – Mar 14, 2016
*Added the Simple Biosphere (SiB) model version 2.5/3.2 from Lixin Lu. This is turned on by setting RAMSIN flag ISFCL=2 and setting an initial CO2 profile in RAMSIN flag CO2_INIT. Adding SiB impacts memory routines, lateral boundaries, surface field initialization, nesting, etc. The only new 3D variable related to SiB is the CO2 concentration and associated tendency, "RCO2P" and "RCO2T".
*In aero_sources.f90, fixed dust model lofting median radius since it had been set to the mean mass radius instead.
*Removed "patch_wetind" from the model since this LEAF-3 surface model variable was never used.
*Removed "scrx" from mem_mksfc.f90 and mksfc_driver.f90 since this is never used.
*Removed passing of unnecessary scratch arrays in nest_geosst.f90 and simply allocated and deallocated variables as needed.
*In hvlib.f90, changed default minimum total mixing ratio from 1.e-4 to 5.e-4kg/kg for both computations of cloud top temperature. This is rather arbitrary and it is best to get cloud top temperature from a satellite simulator.
*In hvlib.f90, made a bug fix to the dust accumulation variable "ACCPDUST" and changed the multiplier from 1.e6 to 1.e5 to get the correct units.
*In hvlib.f90, fixed output for all surface flux variables and eliminated unused or redundant output variables. The flux variables were multiplying by density, but the density multiplier is already accounted for within the model (ie. sflux_u, sflux_t, etc).

*In textout.f90, renamed a number of output variable abbreviations to help distinguish between different units for the same variable to be output.
*In leaf3.f90, eliminated passing of temporary variables (ups2, vps2, ths2, rvs2, pis2, dens2, zts2) into subroutine “leaf3” that can be allocated locally.
*Modified leaf3_init.f90 to allow initialization of both LEAF3 and SiB. SiB uses the LEAF3 initialization by default, but then has additional initialization specific to its routines since SiB is much more sophisticated than LEAF3.
*In leaf3_init.f90, eliminated passing of temporary variables (prsv, pis) which can be allocated locally and declared as individual reals rather than arrays. Similar change to initialization in ruser.f90.
*Change in sfc_driver.f90 to merge subroutine “leaf3” and “sib_driver” into a single routine to prevent have a bunch of redundant code.
*In leaf3_init.f90 subroutine “sfcinit_nofile” and ruser.f90 subroutine “sfcinit_nofile_user” and inithis.f90 subroutine “sfcinit_hstart”, added code to prevent soil_water (ie. soil moisture) from exceeding 100%. This can occur due to difference between land surface soil types in RAMS and the data source model and when history initializing on a different grid. Also adapted this difference between LEAF3 soil type porosity and that used in SiB.
*In SiB and RAMS micro, modified some code that was used to prevent division by zero. In some places, denominators were added by small numbers such as 1.e-20 or 1.e-12. However, some compilers with take (0.0 + 1.e-20) =0.0, which will give a NAN when used as the denominator. Rather, we will be using the “max” function everywhere we need this, such as “max(variable,1.e-12)”. This has not been altered everywhere in the code, so when NANs show up, be on the lookout for this type of error.
*Fix to array bounds potential error in subroutine “htint” related to array “eleva”. The array bounds error does not impact model solution since it occurs only when the associated calculation has zero in the denominator. The array bounds error value generated a NaN that would cause the model to crash.

CONTINUE BELOW FOR UPDATES STARTING WITH RAMS VERSION 6.1.20

MODEL UPDATES STARTING WITH VERSION 6.1.20 WITH ADDITION OF HUCM-SBM BIN MICROPHYSICS AND SIB CARBON-CYCLE LAND SURFACE MODEL

VERSION 6.1.20 – July 13, 2016

*Added HUCM Spectral-Bin Microphysics Model. (via Adele Igel). This impacts several parts of the code. Added subroutine "wetthrm3_bin" in rthrm.f90, condition statements in rtimh.f90 for bin-micro, condition statements in rdint.f90 for bin-micro, new output variables in anal_extra.f90, extra output conditions in anal_write.f90 for bin-micro, extra conditions for history starts in history_start.f90, condition statements for flag checking in opspec.f90, hdf5 conditions in isan_io.f90, conditions for hdf5 output for 4D bin micro in hdf5_utils.f90, changes to memory allocation for 4D bin micro with additional dimension of number of bins, inclusion of bin-micro memory allocation in mem_micro.f90, increased size of scratch arrays in mem_scratch.f90 to accommodate 4D bin-micro, adding tendency arrays for bin-micro in mem_tend.f90, condition statements in mic_misc.f90, mpi consideration for buffer sizes of 4D variables in bin-micro within mpass_cyclic.f90, mpass_init.f90, mpass_lbc.f90, node_mod.f90, and para_init.f90. Modification to rad_driv.f90 to allow bin-micro to interact with Harrington radiation. Additions to turb_k.f90 in calls to "ae1" and "sum_bins".

*Bug fix to "Gaussian" warm bubble (IBUBBLE=2 option) temperature/moisture perturbation in ruser.f90 subroutine "bubble".

*Fix to random temperature perturbations for IBUBBLE=3 option. The random bubbles were not initialized correctly in the DM code since each sub-domain was running the random number generator. Now, only the main node will compute the random number and then broadcast them to the other nodes. Also changed the default random bubble generation to be a 0.1K maximum temperature perturbation that is maximized at the surface and decreased linearly to 500m altitude. (via Leah Grant and Steve Herbener)

*Added code to allow SiB to compute the surface albedo and upward longwave radiation rather than using the LEAF3 default code when SiB is run. This impacted sfc_driver.f90, sib2_co2.f90, and rad_driv.f90. Both the sfc_driver and rad_driv call the subroutine "sfcrad", which should not be called if running SiB.

*Turn advection monotonic flux limiter back on by default. This is only for the vertical direction.

*Added base state density (DN0) to the extra output variables in analysis files.

*Fixed some errors in the budget variables related to latent heating by condensation and freezing. Also fixed the total melting budget variable to prevent double counting of some melting. Latent heating budgets are set to report dTheta(K) by default. This can be changed in micphys.f90 variable "lhrtheta". If set to "false" the latent heating budget variables will return dTemperature(K). (via Adele Igel)

*Added a lot more comments to SiB and reordered some SiB variables to group them by common function. Also linked SiB data to the LEAF3 output variables VEG_TAI, VEG_LAI, VEG_ROUGH, VEG_HEIGHT, VEG_ALBEDO, VEG_FRACAREA. And computed PATCH_ROUGH via SiB info rather than using LEAF3 defaults. These fields are 0 for first analysis file since SiB is not called until first timestep.

*Added call to "non_scalar_bc" before writing to LITE, MEAN, and BOTH file types.

*Prevent automatically writing the "extra" variables in anal_extra.f90 to LITE, MEAN, and BOTH files unless clearly directed to do so in RAMSIN. Did this since at LITE writing time, only LITE variables requested will be returned to master in non-DM code unless it's a full analysis write time. See notes in anal_write.f90 to override this.

*Added accumulation and accumulation rate for total aerosol mass. This had already been done for accumulation of dust-only. Note that this accumulation is from rain-out only and not dry deposition of aerosols to the surface. That is still work to be done, but the dry deposition rate is quite small compared to rain-out for short term simulations.
*Updated history initialization of surface quantities for SiB modifications. Prevent interpolated soil moisture from exceeding 100% due to altered soil porosity via interpolation to new grid. Avoid interpolating land surface class information since this can produce non-integer values when integers are expected. Reset a number of variables (ie. roughness, stomatal resistance) since leaf class may change via new-grid interpolation.
*Added recycle feature to SiB land surface variables.
*Altered hard-coded path to supplemental files such as DustEmission.dat, SiB land info dataset, and HUCM input files. Set this to "../etc".
*Added HUCM model and SiB model REVU output variables and REVU net radiative flux computation. Fixed units for REVU variable ACCPDUST (surface accumulated of dust through wet deposition; not thru dry deposition).
*Modified subroutine "vegndvi" to create separate routines "veg" and "ndvi" in order to have these work well with both LEAF and SiB land surface models.
*Updated HUCM bin-microphysics model to work for history restarts and history initialization and history varfiles. This affected rdint.f90, inithis.f90, and asgen.f90.

VERSION 6.1.21 – August 25, 2016
*Removed SiB diagnostic variable NEE (net ecosystem exchanged) from RAMS and REVU since this is essentially a duplicate of CO2FLX (flux of CO2 from canopy). Removed from sib2_co2.f90, sfc_driver.f90, nest_geosst.f90, rbnd_nonscalar.f90, hvlib.f90 and textout.f90.
*Added code to initialize CAS CO2 (pco2ap) SiB input variable based on initial CO2 (co2_init) and air pressure rather than hard coded value in subroutine "init_sib".
*Split SiB CO2 and PCO2AP initialization into two subroutines so that I can only initialize PCO2AP when doing history restarts with added grid. The 3D CO2 variable is automatically interpolated on history restart with added grid, so we do not want to reinitialize CO2. The reinitialized PCO2AP is overwritten if RAMSIN flag NOFILFLG=0, but is needed if NOFILFLG=2.
*Eliminated use of NOFILFLG since only option = 2 really worked. This was used in subroutine "geonest_nofile". It was called at model initialization and for history restarts with added grids. This subroutine is now only called at model initialization and not for history restarts or history initialization. For history initialization or history restart added grids, the "geonest_nofile" variables are interpolated from previous or parent grids using tested routines in inithis.f90. If the user needs to truly reset the surface data via "geonest_nofile", a call to "geonest_nofile" or "sfcinit_nofile_user" can be added to initialization.
*Eliminated "HISTORY" restart section in rdint.f90 subroutine "initlz" since there was much redundant code and since the old interpolation of history added grids does not seem to match the correct interpolation used in history initialization. The history restart was integrated into the history initialization code in inithis.f90 when grids are added at history restart; otherwise "history_start" is still called. Modified inithis.f90 to incorporate history restart added grids by preserving the original start and restart times and model duration and dealing with variables that do/do not need to be reset on history restart. Code was added to rams_master.f90 in non-DM version and rams_model.f90 in DM version to determine if doing history restart or history

initialization. History header is read in here rather than later on as previously done. A history restart causes flag INITIAL to be set to 3 if grid is added. Flag checks in opspec.f90 have been adjusted accordingly for soil initialization using ISOILDAT=1.
*For initializing with ISOILDAT=1 in the DM code, we force the model to use a single node for the soil moisture/temperature and snow field assignment. This has to be done on 1 node due to nearest neighbor interpolation. This is done in rams_model.f90. This cannot currently be done correctly on multiple nodes. The user has to run the initialization and the model will output the first analysis file. Then a parallel history restart can be run on many nodes.
*Made a bug fix to history initialization interpolation of WP and WC. Had been interpolating via T-level information, but needed to be using M-level information for vertical interpolation. Horizontal interpolation is fine on by the T-grid and M-grid for scalars and momentum variables. This required modifying input variables in all calls to "hi_interp" throughout the model.
*Eliminated code for "patch_land_average", "patch_land_unaverage", and "patch_interp" since these are not being used for interpolation. This could be incorporated into history initialization and history restart added grids for a smoother interpolation of land surface characteristics between grids, but it does not appears to offer substantial gain. This could be examined more closely in the future.
*Eliminated code in mksfc_driver.f90 related to surface file checking during history restart. This is redundant code and is now done at initialization for all grids. Also made a bug fix in use of the RAMSIN namelist flag "isstflg" in the section on NDVI; changed this to used the flag "ndviflg".
*Add SiB CO2 lateral boundary forcing for zero gradient LBCs to prevent CO2 from depleting too much over time. This simple forcing just forces LBCs to past value using a weighting function that is applied to one interior boundary perimeter. This is ok but not smooth. This is rather brute force and should be done more like varfile nudging in the future.
*Changed call to "stars" to only be called (when running SiB) if patch=1 or (initial=2 and time<0.001) or patch_area<0.009. Had to move computation of "ubmin" and "vels_pat" out of subroutine "stars" since the routine "sfclmcv" always needs these variables. Making this change speeds up the model by not running "stars" un-necessarily for SiB.
*Altered naming convention for recycle variables from "recycle_leaf" to "recycle_sfc" and RAMSIN variable "IPAST_LEAF" to "IPAST_SFC" since these are no longer exclusive to LEAF, but can also be used with SiB.
*Made a bug fix for calls to "grndvap". Calls to this routine should be passing the top snow layer or surface water layer data to the routine (layer ksn). Rather, it had been passing in layer (mzs) which is the maximum number of snow or surface water layers. However, KSN does not always equal MZS. This could cause serious errors in fluxes from snow surfaces when MZS in the RAMSIN namelist is > 1.
*Fixed REVU variables related to surface snow depth, mass, and temperature. Had to fix sections of hvlib.f90 and rcomp.f90 to make the output correct. The previous subroutines were wrong in their variable passing and computations of these variables. I do not think that they ever previously worked correctly.
*Got rid of calls to "snow_init" and deleted this subroutine. It is not necessary since idealized snow initialization can easily be added to ruser.f90.

VERSION 6.1.22 – September 12, 2016

*Altered use of NGBEGUN in history restart/initialization. For history initialization or history restart added grid, NGBEGUN will equal 1 rather than 0 if they are being restarted from a previous simulations that was beyond time=0. Since these grids are interpolated from other grids, they do not start from a time=0 sort of state, meaning that past and current momentum and pressure fields are different. From a true time=0 state, the past and current are equal and hence RAMS uses a different Asselin "filter parameter" for the first timestep in order to preserve amplitude in this special case.
*Fixed a bug in history restart added grids for 2D fields.
*Added namelist parameter ISNOWDAT which is similar in function to ISOILDAT, but gives the user control over whether to initialize snow fields from data in varfiles.
*Eliminated "alloc_leafcol" in which some 1D variables are dynamically allocated for LEAF. Rather than have a separate awkward module/routine for allocating these, I added these variables to the "leaf_coms" variable declaration and gave them static array sizes of "nzgmax+nzsmax+1". These are not large arrays, so this should be no problem.
*Got rid grid dependent polelat and polelon (ie. platn(ngrids), plonn(ngrids). This was an unnecessary lingering grid dependent variable that had hung around since the days in which RAMS was trying to be turned into a dual-hemispheric (global) model. This never worked correctly and was replaced by OLAM. So, I decided to remove the grid-dependent nature of platn and plonn and just use polelat, polelon from the RAMSIN namelist.
*Fixed radiative heating rate in Harrington and Chen-Cotton schemes so that the heating rates are in potential temperature rather than temperature. This simply required dividing the heating rate by the exner function. Set top and bottom boundaries for radiative heating rate "FTHRD".
*Moved the setting of the top and bottom boundaries for SWUP, SWDN, LWDN, LWUP to "radcalc3". These are computed on "m" levels, so it was best to set these inside radiation on radiation levels. The top boundaries are set to the top radiation level which may be further aloft than model top. This is useful for computing radiative balance.
*Fixed code that was commented out to allow analysis write of "extra" variables to LITE files that are the LEAF type 4,5,6 that have dimensions that include patches, soil layers, or snow layers. This had not been functioning, but does now. A couple of non-active placeholder LEAF variables are in the code in "anal_extra.f90" and "anal_write.f90". This has to be customized as needed and care should be taken when adding "extra" variables since the numbering process must be exact.