Table of Contents for Modified Modules:

module_NoahMP_hrldas_driver_wt.F

In MODULE_NoahMP_hrldas_driver:

- Define new water tracer variables
- Add “water_tracer_option” in the namelist
    - It will be transferred to “OPT_WT"
- Add “partial_mixing_option” in the namelist
    - It will be transferred to “IOPT_PM"
- In subroutine “land_driver_init”:
    - Read in namelist and assign “IOPT_PM”, “NSUB”, “NTRACER"
    - Assign “wvtstart” and “wvtend” values according to namelist
    - Allocate water tracer variables
    - Initialize tracer variables to be 0.0 or undefined_real
- In subroutine “land_driver_exe”:
    - Assign “wvtflag” according to whether the time step is within the tagging period or not
    - Initialize tracer variables to be 0.0 if wvtflag == 1
    - Assign “wvt_ratio” according to “wvtflag” values
    - If water tracer is activated
        - Call “noahmplsm” with optional variables passing
    - Output new tracer variables for each time step

module_hrldas_netcdf_io_wt.F

- Subroutine add_to_output_3d
    - Added option for snow_or_soil to be “TRAC” so that it can output tracer variables in the sub-soil layers
        - elseif
    - Assign “zdimid"

        - zdimid = dimid_tracer_layers_remember
- Define “dimid_tracer_layers_remember" at the beginning (line 46)
        - Integer, private :: dimid_tracer_layers_remember
- Assign “dimid_tracer_layers_remember” in subroutine “prepare_output_file_seq"
    - Add “ntracer” into the argument as input and define it
        - Integer, intent(in) :: ntracer
    - Define “dimid_tracer_layers"
        - Integer :: dimid_tracer_layers
    - Define “tracer_layers” dimension in output file and assign “dimid_tracer_layers"
        - iret = nf90_def_dim(acid, “tracer_layers”, nsnow, dimid_tracer_layers)
    - Assign “dimid_tracer_layers_remember"
        - dimid_tracer_layers_remember = dimid_tracer_layers
- For a different soil layer and depth configuration other than the default 0.1, 0.3. 0.6 1.0m, in “readinit_hrldas” subroutine, it will do the interpolation accordingly for soil temperature and soil moisture
    - By calling “init_interp"

- Subroutine “prepare_output_file_mpp"
    - Add “ntracer” into the argument when calling “prepare_output_file_seq"
    - Define “ntracer” as input argument

---

module_sf_noahmpdrv_wt.F

In subroutine “noahmplsm”:

- Add water tracer variable (2-D or 3-D) as argument and define them
    - “NTRACER” is not an optional variable because it will be needed to define dimensions
- Define 1-D tracer variables to transfer 3-D variables for column calculation
    - Lines 495
- Transfer 3-D tracer variables to 1-D variables
    - Using “if present(WVTFLAG)” statement
- Add “NSOIL” and “ZSOIL” as new argument when calling “TRANSFER_MP_PARAMETERS"
- Call “NOAHMP_SFLX"
    - Add 1-D tracer variables as optional arguments
- Update 3-D tracer variables with each column
    - Use “if present(WVTFLAG)” statement

In subroutine “TRANSFER_MP_PARAMETERS”:

- Add “NSOIL” and “ZSOIL” as new arguments and define them
    - NSOIL as input
        - INTEGER, INTENT(IN) :: NSOIL
    - ZSOIL as optional input
        - REAL, DIMENSION(1:NSOIL), INTENT(IN),OPTIONAL :: ZSOIL
- In this newer version, SOILTYPE can vary vertically but has been hard-wired to be 4 layers
    - Therefore, also adjust the dimension of SOILTYPE using NSOIL
- Define “ZSOIL_OLD” and “IZ” as local variables
    - ZSOIL_OLD to store default soil layer configuration
        - REAL, DIMENSION(1:NSOIL) :: ZSOIL_OLD = (/0.1, 0.4, 1.0, 2.0/)
    - IZ for vertical loop
        - INTEGER :: IZ
    - If (present(ZSOIL)), do the vertical loop from 1 to NSOIL
        - Redefind parameter%NROOT

module_sf_noahmplsm_wt.F

Here defines most of the new subroutines that will be used for water tracer calculation

- State them at the beginning of the code
        - Public :: tag_precip
        - Private :: calculate_dz
        - Etc (see the end of this file listing new subroutines for tracer calculation)
- Subroutine “NOAHMP_SFLX"
    - Add tracer variables as optional arguments
    - Define these tracer variables as optional variables
        - NTRACER is not optional, because it would be required to define dimension
        - Others are optional
    - Define local tracer variables
    - Call “**tag_precip**” to tag precipitation as water tracer (Line 740, after call “ATM")
        - If (present(WVTFLAG)) call tag_precip(RAIN, wvt_ratio, RAIN_TR)
    - Call “**calculate_dz**” to calculated tracer snow/soil layer thickness (Line 752)
    - Call “PRECIP_HEAT"
        - Add tracer variables as optional arguments
    - Call “ENERGY"

        - Add tracer variables as optional arguments
    - Update “SICE_TR” and “SICE_SUB"
        - If (present(WVTFLAG)) SICE_TR(:)=MAX(0.0, SMC_TR(:)-SH2O_TR(:))
    - Call “WATER”
        - Add tracer variables as optional arguments
    - Call “**calculate_lh_energy**” to calculate tracer energy fluxes
        - If (present(WVTFLAG)) call calculate_lh_energy
    - Update “SNOWH_TR” and “SNEQV_TR” if “SNOWH” is close to 0
        - If (present(WVTFLAG)) SNOWH_TR = 0.0

- Subroutine “PRECIP_HEAT"
    - Add water tracer variables as optional arguments
    - Define water tracers as optional variables
    - Define local tracer variables
    - Initialize tracer variables
        - If (present(WVTFLAG)) QINTR_TR = 0.0
    - Call “tag_prcip” to tag intercepted or through fall as tracer intercepted or through
    - Update “CANLIQ_TR"
    - Update “CANICE_TR"
    - Update “CMC_TR” (tracer canopy water)
    - Update tracer snow/rain on the ground

- Subroutine “ENERGY"
    - Add water tracer variables as optional arguments
    - Define water tracer variables as optional variables
    - Call “PHASECHANGE”
        - Add tracer variable as optional arguments

- Subroutine “PHASECHANGE”:
    - Add water tracer variables as optional arguments
    - Define water tracer variables as optional variables
    - Initialize tracer variables
    - Calculate the rate of melting/freezing for tracer snow and soil
        - Call “**MELTING_SUB**"
        - Or call “**FREEZING_SUB**"
    - Update “MLIQ_TR”, “QMELT_TR"
    - Update “SNLIQ_TR” and “SNICE_TR”, “SMC_TR” and other “SUB” variables

---

- Subroutine “WATER”:
  - Add water tracer variables as optional arguments
  - Define water tracer variables as optional variables
  - Initialize water tracer variables
  - Call “CANWATER"
    - Add tracer variables as optional arguments
  - Call “SNOWWATER"
    - Add tracer variables as optional arguments
  - For frozen ground, call “**TOPSOIL_SUB**” or update “SICE_TR”, “EDIR_TR"
  - Update “QINSUR_TR"
  - If lake, call “**get_wt_ratio**” and update “WSLAKE_TR” and “EDIR_TR”
  - If soil, call “SOILWATER"
    - Add water tracer variables as optional arguments
    - Update “EDIR_TR"
  - Update “RUNSUB_TR"
  - Update “SMC_TR” and “SMC_SUB"
  - Update “RUNSUB_TR” by adding “SNOFLOW_TR”

- Subroutine “CANWATER”:
  - Add water tracer variables as optional arguments
  - Define water tracer variables as optional variables
  - Define local tracer variables
  - Initialize tracer variables
  - Call “get_wt_ratio” to calculate tracer ratio in “CANLIQ” and “CANICE"
  - Update “CANLIQ_TR” due to evaporation/condensation
  - Update “CANICE_TR” due to sublimation/frost
  - Update “CANLIQ_TR” and “CANICE_TR" due to melting/freezing
  - Update “CANLIQ_TR” and “CANICE_TR” due to melting/freezing
  - Calculate “CMC_TR” and “ECAN_TR"

- Subroutine “SNOWWATER"
  - Add water tracer variables as optional arguments
  - Define water tracer variables as optional variables
  - Define local tracer variables
  - Call “SNOWFALL”
    - Add water tracer variables as optional argument
  - Call “COMPACT”:
    - Add water tracer variables as optional argument

  - Call “COMBINE"
    - Add water tracer variables as optional argument
  - Call “DIVIDE"
    - Add water tracer variables as optional argument
  - Call “SNOWH2O"
    - Add water tracer variables as optional argument
  - Set tracer variables in empty snow layers to zero
  - For glacier regions, update tracer variables
  - Sum up tracer snow mass for layered snow as “SNEQV_TR"
  - Reset and update “DZSNSO_TR"

- Subroutine “SNOWFALL”:
  - Add water tracer variables as optional argument
  - Define water tracer variables as optional variables
  - Update “SNOWH_TR” and “SNEQV_TR"
  - Update “SNLIQ_TR” and “SNICE_TR” and “DZSNSO_TR"

- Subroutine “COMBINE”:
  - Add water tracer variables as optional arguments
  - Define water tracer variables as optional variables
  - Update “SNEQV_TR”, “SNOWH_TR”, “PONDING1_TR”
    - Call “TOPSOIL_SUB” if sublimation removes water from soil
  - Update tracer snow variables for case of too large surface sublimation
  - Update tracer snow variables if all snow is going (converted into pounding water on soil surface)
  - For snow layers exist:
    - Call “**COMBO_TR**” to combine layers (**WHY??**)
      - Add water tracer variables as arguments
    - Shift tracer elements above this down one

- Subroutine “DIVIDE”:
  - Add water tracer variables as optional arguments
  - Define water tracer variables as optional variables
  - Update “SWICE_TR” and “SWLIQ_TR"
  - Split snow layers into to if needed
  - For splitting into more layers
    - Update each layer tracer snow vars according to snow depth
    - Call “**COMBO_TR**” if needed

    - Update “SNICE_TR”, “SNLIQ_TR” and “DZSNSO_TR"

- Subroutine “**COMBO_TR**”:
    - Modified upon “COMBO”
    - Add water tracer variables as new arguments
    - Define water tracer variables
    - Update tracer snow variables

- Subroutine “COMPACT”:
    - Add water tracer variables as optional arguments
    - Define water tracer variables as optional variables
    - Update “DZSNSO_TR” due to melting

- Subroutine “SNOWH2O”:
    - Add water tracer variables as optional argument
    - Define water tracer variables
    - If SNEQV = 0 (melting) and perhaps melt top soil snow into water
        - Call “TOPSOIL_SUB”
        - Update “SICE_TR” and “SICE_SUB"
    - For shallow snow
        - Call “get_wt_ratio” or “TOPSOIL_SUB” if needed
        - update “EDIR_TR”, “SNEQV_TR”, “SNOWH_TR"
    - For deep snow
        - Update “SNICE_TR” and “EDIR_TR"
        - Call “COMBINE"
        - Update “SNLIQ_TR”
    - Update “QIN_TR” and “QOUT_TR"
    - “QOUT_TR” being liquid water from snow bottom to soil

- Subroutine “SOILWATER"
    - Add water tracer variables as optional arguments
    - Define water tracer variables as optional variables
    - Define local tracer variables
    - Initialize tracer variables
    - If snowmelt is too large, call “**TOPSOIL_SAT**” or “**SOIL_SAT**” to update “SH2O_SUB” and “SH2O_TR"

    - Call “**get_wt_ratio**” to update “RUNSRF_TR” and “PDDUM_TR"
    - Call “**SMT_TR**” to update tracer soil water variables
    - Update “QDRAIN_TR”, “RUNSRF_TR” and “ETRAN_TR"

After subroutine “noahmp_options” before the end of END MODULE “MODULE_SF_NOAHMPLSM”:
Add all new tracer subroutines here:

- Subroutine “**tag_precip**"
    - Tag water as tracer water using wvt_ratio
- Subroutine “**calculate_dz**"
    - Compute tracer snow and soil layer depth
- Subroutine “**calculate_lh_energy**"
    - Compute latent heat due to evaporation/sublimation amount
- Subroutine “**get_wt_ratio**"
    - Calculate tracer/total water ratio
- Subroutine “**MELTING_SUB**"
    - Deals with melting in each soil sublayer
- Subroutine “**FREEZING_SUB**"
    - Deals with freezing in each soil sublayer
- Subroutine “**TOPSOIL_SUB**"
    - Deals with top soil layer “SH2O_TR” and “SICE_TR"
- Subroutine “**TOPSOIL_SAT**"
    - Deals with top soil layer for saturated situation
- Subroutine “**SOIL_SAT**"
    - Deals with non-top soil layers fro saturated situation
- Subroutine “**SMT_TR**"
    - Updates tracer soil variables
    - Each steps are explained by the comments

Note: **RED BOLD** suggests new water tracer related subroutines