CLASS-CTEM

Generated by Doxygen 1.8.9.1

Mon Jan 30 2017 10:33:41

Contents

Chapter 1

Main Page

1.1 Overview of the Canadian Terrestrial Ecosystem Model (CTEM)

Version 1 of the CTEM is the terrestrial carbon cycle component of the second generation Canadian Earth System Model (CanESM2) [6] where it is coupled to version 2.7 of the Canadian Land Surface Scheme (CLASS). CTEM v. 2.0 [38] is currently coupled to the CLASS v. 3.6 [51]. The coupled CLASS–CTEM model is capable of being run online in the CanESM model or offline, as is the case in this study, driven by observation-based meteorological forcings. CTEM models terrestrial ecosystem processes for nine PFTs, two of which are crop PFTs (see Table tab:compparams}), by tracking the flow of carbon through three living vegetation components (leaves, stem and roots) and two dead carbon pools (litter and soil). The carbon budget equations for the model's five pools are summarized in Sect. rate_change_eqns} of the Appendix. The amount of carbon in these five carbon pools is simulated prognostically. CLASS models the land surface energy and water balance and calculates liquid and frozen soil moisture, and soil temperature for three soil layers (with thicknesses 0.1, 0.25 and 3.75\, m). In the CLASS–CTEM framework, CLASS uses structural vegetation attributes (including LAI, vegetation height, canopy mass and rooting depth) simulated by CTEM, and CTEM uses soil moisture, soil temperature and net radiation calculated by CLASS. Combined, CLASS and CTEM simulate the atmosphere–land fluxes of energy, water and CO_2.

Version 1.0 of CTEM is described in a collection of papers detailing parametrization of photosynthesis, autotrophic and heterotrophic respiration [7]; phenology, carbon allocation, biomass turnover and conversion of biomass to structural attributes [10]; dynamic root distribution [8]; and disturbance (fire) [9]. These processes are modelled over prescribed fractional coverage of nine PFTs [53] and determine the structural vegetation dynamics including vegetation biomass, LAI, vegetation height, fraction of roots in each of the three soil layers, leaf onset and offset times and primary CO_2 fluxes of gross primary productivity (GPP) and NPP. A full description of CTEM and changes from v. 1.0 to v. 2.0 are included in the Appendix.

A parametrization for competition between PFTs in an earlier version of CTEM is described by [3] [11] where it was evaluated at select locations. Here we present CTEM v. 2.0, which builds upon the model framework of CTEM v. 1.0 and can be run in two different modes, either (i) using specified fractional coverage of its nine PFTs, or (ii) allowing the fractional coverage of its seven non-crop PFTs to be dynamically determined based on competition between PFTs. The parametrization for simulating competition between PFTs is summarized in Sect. compmain}. The fire parametrization has also been refined in the new model version as described in Appendix fire}. The C↩ LASS–CTEM modelling framework has the capability of representing the sub-grid scale variability of PFTs using either a composite or a mosaic configuration [33] [36]. In the composite (or single tile) configuration, the vegetation attributes for all PFTs present in a grid cell are averaged and used in energy and water balance calculations that determine the physical land surface conditions including soil moisture, soil temperature and thickness and fractional coverage of snow (if present). In the mosaic (or multi-tile) configuration each PFT is allocated its own tile for which separate energy and water balance calculations are performed. As a result, the simulated carbon balance evolves somewhat differently in the two configurations despite being driven with identical climate forcing (see [36]). The results presented in this paper are obtained using the composite configuration.

1.2 Overview of the Canadian Land Surface Scheme (CLASS)

The Canadian Land Surface Scheme, CLASS, was originally developed for use with the Canadian Global Climate Model or GCM (Verseghy, 1991; Verseghy et al., 1993). This documentation describes version 3.6 of CLASS, which was released in December of 2011. The table at the end of this overview summarizes the development of CLASS from the late 1980's to the present.

The basic function of CLASS is to integrate the energy and water balances of the land surface forward in time from an initial starting point, making use of atmospheric forcing data to drive the simulation. When CLASS is run in coupled mode with a global or regional atmospheric model, the required forcing data are passed to it at each time step over each modeled grid cell from the atmospheric driver. CLASS then performs its internal calculations, evaluating a suite of prognostic and diagnostic variables such as albedo and surface radiative and turbulent fluxes, which are in turn passed back to the driver. CLASS can also be run in uncoupled or offline mode, using forcing data derived from field measurements, and the output values of its prognostic and diagnostic variables can then be validated against observations. Version 3.6 includes a single-column offline driver which can be used for this purpose.

CLASS models separately the energy and water balances of the soil, snow, and vegetation canopy (see the diagram below). The basic prognostic variables consist of the temperatures and the liquid and frozen moisture contents of the soil layers; the mass, temperature, density, albedo and liquid water content of the snow pack; the temperature of the vegetation canopy and the mass of intercepted rain and snow present on it; the temperature and depth of ponded water on the soil surface; and an empirical vegetation growth index. These variables must be initialized, and a set of physical parameters describing the soil and vegetation existing on the modelled area must be assigned background values, at the beginning of the simulation.

At each time step, CLASS calculates the bulk characteristics of the vegetation canopy on the basis of the vegetation types present over the modelled area. In a pre-processing step, each vegetation type is assigned representative values of parameters such as albedo, roughness length, annual maximum and minimum plant area index, rooting depth and so on (see the section on "Data Requirements"). These values are then aggregated over four main vegetation categories identified by CLASS: needleleaf trees, broadleaf trees, crops, and grass (i.e. short vegetation). The physiological characteristics of the vegetation in each category are determined at the current time step using the aggregated background parameters and assumed annual or diurnal variation functions. These physiological characteristics are then aggregated to produce the bulk canopy characteristics for the current time step.

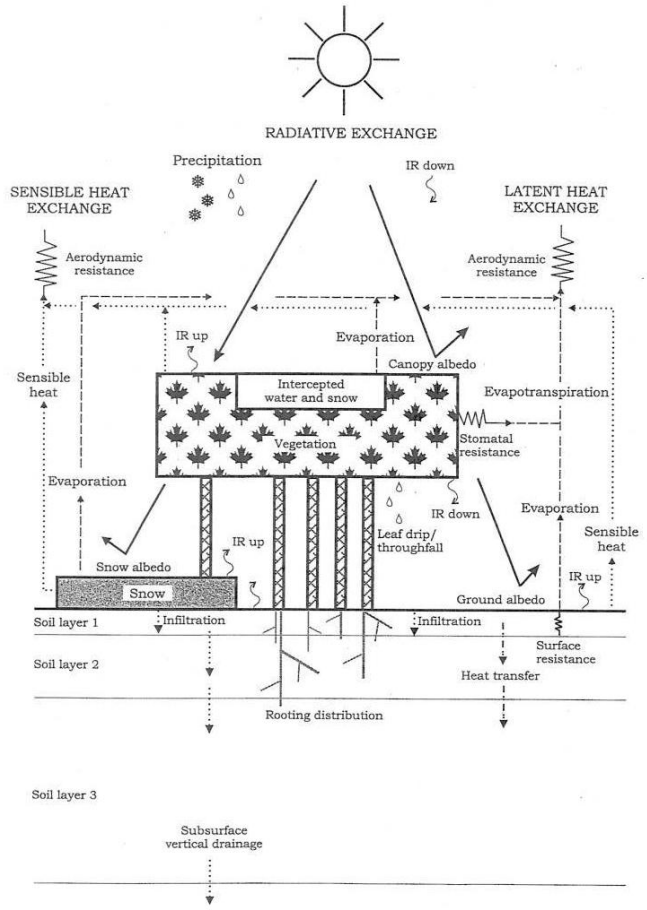

Figure 1.1: Schematic Diagram Of CLASS

In performing the surface flux calculations the modeled area is divided into up to four subareas: bare soil, vegetation over soil, snow over bare soil, and vegetation over snow. The fractional snow coverage is determined using the concept of a threshold snow depth. If the calculated snow depth is less than this value, the snow depth is set to the threshold value and the fractional snow cover is calculated on the basis of conservation of snow mass. The fluxes are calculated for each of the four subareas, and these and the prognostic variables are then areally averaged before being passed back to the atmospheric model.

Originally CLASS performed only one set of these calculations for each grid cell of the model domain. In more recent versions, a "mosaic" option has been added to handle sub-grid scale heterogeneity more effectively. When this option is utilized, each grid cell is divided into a user-specified number of mosaic "tiles", and the CLAS↩ S calculations are performed in turn over each. The surface fluxes are averaged, but the prognostic variables are kept separate for each of the tiles of the mosaic between time steps.

The section on the CLASS offline driver, RUNCLASS, provides information on how a CLASS run is typically performed, from assigning the background and initial values of variables, through calling the high-level CLASS subroutines, to calculating values of diagnostic and output variables. A gather-scatter operation is included in the driver, mimicking the practice in atmospheric models of "gathering" land surface points on latitude circles onto long vectors prior to the calculations, for improved computational efficiency on vector supercomputers. For CLASS, the mosaic tiles on each of the modelled grid cells are "gathered" onto long arrays prior to calling the CLASS subroutines (thus collapsing the first two dimensions of the arrays into one), and subsequently "scattered" back onto the grid cells before performing the diagnostic averaging calculations.

The two sections following the one that describes the driver provide detailed descriptions first of the common block and other preliminary routines that are called before the run is launched, and then of the pre- and post-processing routines that are called at the beginning and end of each time step. The next three sections detail the three main CLASS subroutines together with the auxiliary subroutines that they call: CLASSA, which handles the calculation of

the albedos and other surface parameters; CLASST, which addresses the evaluation of the surface energy balance and related variables; and CLASSW, which performs the surface water balance calculations and the re-aggregation of the prognostic variables. The sub-section on each subroutine contains a dictionary of the variables passed into and out of it, with units. The final section provides a list of references cited.

1.2.1 Developmental history of CLASS

1.0	April 1989	Basic thermal and hydrological model of snow and soil.
2.0	August 1991	Addition of vegetation thermal and hydrological model.
2.1	May 1993	Full vectorization of code to enable efficienr running on vector supercomputers.
2.2	April 1994	Augmentation of diagnostic calculations; incorporation of in-line comments throughout; development of a parallel stand-alone version of the model for use with field data.
2.3	December 1994	Revisions to diagnostic calculations; new near-surface atmospheric stability functions.
2.4	August 1995	Complete set of water budget diagnostic calculations; parametrizations of organic soils and rock soils; allowance for inhomegeneity between soil layers; incorporation of variable surface detention capacity.
2.5	January 1996	Completion of energy budget diagnostic calculations.
2.6	August 1997	Revisions to surface stability function calculations.
2.7	December 1997	Incorporation of variable soil permeable depth; calculation of soil thermal and hydraulic properties based on textural composition; modified surface temperature iteration scheme.
3.0	December 2002	Improved treatment of soil evaporation; complete treatment of organic soils; new canopy conductance formulation; preliminary routines for lateral movement of soil water; enhanced snow density and snow interception; improved turbulent transfer from vegetation; mosaic formulation.
3.1	April 2005	Faster surface temperature iteration scheme; refinements to leaf boundary resistance formulation; improved treatment of snow sublimation and interception; transition to Fortran 90 and single precision var
3.2	May 2006	Option for multiple soil layers at depth; additional liquid water content of snow pack; revised radiation transmission in vegetation.
3.3	December 2006	Separate temperature profile curve fit for snow and soil; multiple-layer option for ice sheets; water and energy balance checks for each time step; modifications to soil hydraulic conductivity calculatio
3.4	April 2008	Streamline and clean up code; updated soil thermal conductivity calculations; revisions to handling of water stored on vegetation.
3.5	December 2010	Updated field capacity calculation; revised treatment of water on canopy; reworked calculation of baseflow.
3.6	December 2011	Revised ponding depth over organic soils; revised snow albedo refreshment threshold; new snow thermal conductivity algorithm; interface with Canadian Terrestrial Ecosystem Mode

1.3 Data Requirements

This section describes the three types of data that are required to run CLASS: atmospheric forcing data, surface vegetation and soil data, and initial values for the prognostic variables.

1.3.1 Forcing Data

At each time step, for each grid cell or modelled area, the following atmospheric forcing data are required:

- FCLOROW Fractional cloud cover [ ]
- FDLROW Downwelling longwave sky radiation $[Wm^{-2}]$
- FSIHROW Near infrared shortwave radiation incident on a horizontal surface $[Wm^{-2}]$
- FSVHROW Visible shortwave radiation incident on a horizontal surface $[Wm^{-2}]$
- PREROW Surface precipitation rate $[kgm^{-2}s^{-1}]$

- PRESROW Surface air pressure $[P_a]$
- QAROW Specific humidity at reference height $[kgkg^{-1}]$
- TAROW Air temperature at reference height [K]
- ULROW Zonal component of wind velocity $[ms^{-1}]$
- VLROW Meridional component of wind velocity $[ms^{-1}]$
- VMODROW Wind speed at reference height $[ms^-1]$
- ZBLDROW Atmospheric blending height for surface roughness length averaging [m]
- ZRFHROW Reference height associated with forcing air temperature and humidity [m]
- ZRFMROW Reference height associated with forcing wind speed [m]

When assembling the forcing data, the following guidelines should be noted:

1) CLASS ordinarily requires that the forcing incoming shortwave radiation be partitioned into the visible and near-infrared components. If these are not available, however, they can each be roughly estimated as approximately half of the total incoming solar radiation.

2) The fractional cloud cover is used to calculate the direct and diffuse components of the incoming shortwave radiation. If it is not available it can be estimated on the basis of the solar zenith angle and the occurrence of precipitation (see the section on the RUNCLASS driver).

3) For atmospheric models, the air temperature supplied to CLASS should be the lowest level air temperature extrapolated using the dry adiabatic lapse rate to the bottom of the atmosphere, i.e. to where the wind speed is zero and the pressure is equal to the surface pressure P a . For field data, the actual measured air temperature at the reference height should be used, since in this case the adiabatic extrapolation is performed within CLASS.

4) Atmospheric models provide the zonal and meridional components of the wind velocity, but CLASS does not actually require information on wind direction. Thus, if only the scalar wind speed is available, either ULROW or VLROW can be set to it, and the other to zero. (Both of these terms, plus the scalar wind speed VMODROW, must be supplied to CLASS.)

5) In atmospheric models the forcing wind speed, air temperature and specific humidity are obtained from the lowest modelled atmospheric layer, and thus the reference height will be the height above the “surface” (i.e. the location where the wind speed is zero and the pressure is equal to the surface pressure P a) corresponding to that lowest layer. Some atmospheric models use a vertical co-ordinate system in which the momentum and thermodynamic levels are staggered, and if so, ZFRMROW and ZRFHROW will have different values. If that is the case, the switch ISLFD in the CLASS driver should be set to 2, so that the subroutines FLXSURFZ and DIASURFZ are called (see the RUNCLASS documentation), since the other options do not support different reference heights. In the case of field data, the reference height is the height above the ground surface at which the variables are measured. If the measurement height for wind speed is different from that for the air temperature and specific humidity, again the ISLFD switch in the CLASS driver should be set to 2. (Note that neither ZRFHROW nor ZRFMROW may be smaller than the vegetation canopy height, as this will cause the model run to crash.)

6) If the surface being modelled is a heterogeneous one, care must be taken to ensure that the reference heights are greater than the “blending height”, the distance above the surface at which the atmospheric variables are not dominated by any one surface type. In principle this height depends on the length scale of the roughness elements; it is usually of the order of 50-100 m. In CLASS the blending height is used in averaging the roughness lengths over the modelled area, and is read in separately from ZRFMROW and ZRFHROW as ZBLDROW.

7) CLASS is able to run with total incoming precipitation, partitioning it into rainfall and snowfall on the basis of empirically derived equations. If the rainfall rate (RPREROW) and snowfall rate (SPREROW) are available, they should be used instead. The READ statement in the CLASS driver should be modified accordingly, and the switch IPCP should be set to 4.

8) The length of the time step should be carefully considered in assembling the forcing data. CLASS has been designed to run at a time step of 30 minutes or less, and the explicit prognostic time stepping scheme used for the soil, snow and vegetation variables is based on this assumption. Longer time steps may lead to the appearance of numerical instabilities in the modelled prognostic variables.

1.3.2 Vegetation Data

For each of the four main vegetation categories (needleleaf trees, broadleaf trees, crops and grass), the following data are required for each mosaic tile over each grid cell or modelled area:

- ALICROT Average near-IR albedo of vegetation category when fully-leafed [ ]
- ALVCROT Average visible albedo of vegetation category when fully-leafed [ ]
- CMASROT Annual maximum canopy mass for vegetation category $[kgm^{-2}]$
- FCANROT Annual maximum fractional coverage of modelled area [ ]
- LNZ0ROT Natural logarithm of maximum vegetation roughness length [ ]
- PAMNROT Annual minimum plant area index of vegetation category [ ]
- PAMXROT Annual maximum plant area index of vegetation category [ ]
- PSGAROT Soil moisture suction coefficient (used in stomatal resistance calculation) [ ]
- PSGBROT Soil moisture suction coefficient (used in stomatal resistance calculation) [ ]
- QA50ROT Reference value of incoming shortwave radiation (used in stomatal resistance calculation) $[Wm^{-2}]$
- ROOTROT Annual maximum rooting depth of vegetation category [m]
- RSMNROT Minimum stomatal resistance of vegetation category $[sm^{-1}]$
- VPDAROT Vapour pressure deficit coefficient (used in stomatal resistance calculation) [ ]
- VPDBROT Vapour pressure deficit coefficient (used in stomatal resistance calculation) [ ]

CLASS models the physiological characteristics of trees as remaining constant throughout the year except for the leaf area index and plant area index, which vary seasonally between the limits defined by PAMXROT and PAMN↩ ROT. The areal coverage of crops varies from zero in the winter to FCANROT at the height of the growing season, and their physiological characteristics undergo a corresponding cycle. Grasses remain constant year-round. (For full details of these calculations, see the documentation for subroutine APREP). Urban areas are also treated as "vegetation" in the CLASS code, and have associated values for FCANROT, ALVCROT, ALICROT and LNZ0ROT. Thus these arrays have a third dimension of 5 rather than 4.

Ideally the above vegetation parameters should be measured at the modelled location. Of course this is not always possible, especially when running over a large modelling domain. As a guide, the table in Appendix A provides representative values for the 20 vegetation types recognized by the Canadian GCM. If more than one type of vegetation in a given category is present on the modelled area, the parameters for the category should be areally averaged over the vegetation types present. For the stomatal resistance parameters, typical values of these for the four principal vegetation types are given below:

	RSMN	QA50	VPDA	VPDB	PSGA	PSGB
Needleleaf trees	200.0	30.0	0.65	1.05	100.0	5.0
Broadleaf trees	125.0	40.0	0.50	0.60	100.0	5.0
Crops	85.0	30.0	0.50	1.00	100.0	5.0
Grass	100.0	30.0	0.50	1.00	100.0	5.0

1.3.3 Soil Data

The following information is required for each modelled soil layer:

- DELZ Layer thickness [m]

- ZBOT Depth of bottom [m]

The standard operational configuration for CLASS consists of three soil layers, of thicknesses 0.10 m, 0.25 m and 3.75 m, and thus of bottom depths 0.10, 0.35 and 4.10 m respectively. CLASS versions 3.2 and higher support other options: the third soil layer may be replaced with a larger number of thinner layers, and/or the bottom of the soil profile may be extended below 4.10 m. However, because the temperature stepping scheme used in CLASS is of an explicit formulation, care must be taken not to make the layers too thin, since this may lead to numerical instability problems.

For each of the modelled soil layers on each of the mosaic tiles, the following texture data are required:

- CLAYROT Percentage clay content
- ORGMROT Percentage organic matter content
- SANDROT Percentage sand content

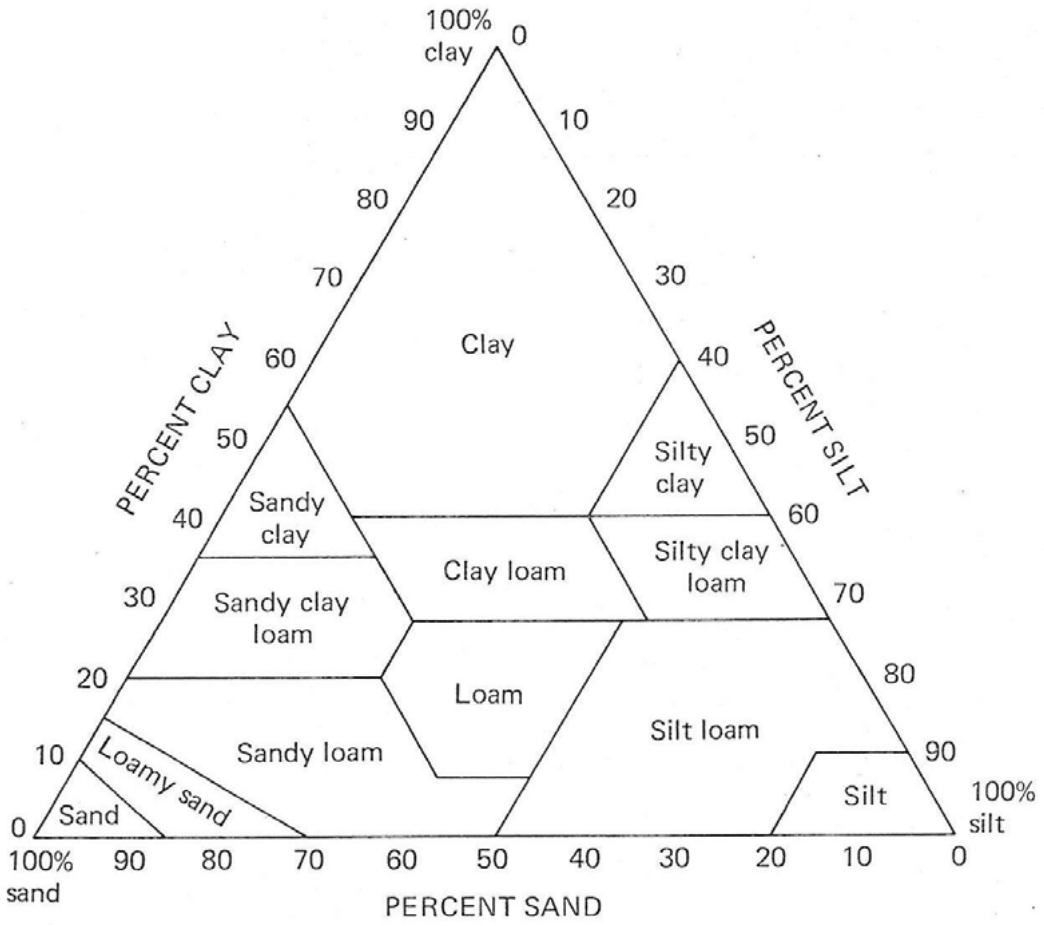

Figure 1.2: Percent Sand

1) For mineral soils, the percentages of sand, clay and organic matter content need not add up to 100%, since the residual is assigned to silt content. If the exact sand, clay and organic matter contents are not known, estimates can be made for the general soil type on the basis of the standard USDA texture triangle shown above. Organic matter contents in mineral soils are typically not more than a few percent.

2) If the soil layer is a fully organic one, SANDROT, CLAYROT and ORGMROT are used differently. The sand content is assigned a flag value of -2, and the organic matter content may be assigned a flag value of 1, 2 or 3 depending on whether the peat texture is fibric, hemic or sapric (see Letts et al., 2000). The current default is for

the first layer to be assumed as fibric, the second as hemic and any lower layers as sapric. CLAYROT is not used and is set to zero.

3) If the layer consists of rock, SANDROT is assigned a flag value of -3. If it is part of a continental ice sheet, it is assigned a flag value of -4. In both cases, CLAYROT and ORGMROT are not used and are set to zero.

SANDROT, CLAYROT and ORGMROT are utilized in the calculation of the soil layer thermal and hydraulic properties in subroutine CLASSB. If the measured values of these properties are available, they should be used instead.

For each of the mosaic tiles over the modelled area, the following surface parameters must be specified:

- DRNROT Soil drainage index
- FAREROT Fractional coverage of mosaic tile on the modelled area
- MIDROT Mosaic tile type identifier (1 for land surface, 0 for inland lake)
- SDEPROT Soil permeable depth [m]

1) The soil permeable depth, i.e. the depth to bedrock, may be less than the modelled thermal depth of the soil profile. This permeable depth is indicated by the variable SDEPROT. If the depth to bedrock occurs within a soil layer rather than at the interface between two layers, CLASS assigns the specified mineral or organic soil characteristics to the part of the layer above bedrock, and values corresponding to rock to the portion below.

2) The drainage index, DRNROT, is set to 1 except in cases of deep soils where it is desired to suppress drainage from the bottom of the soil profile (e.g. in bogs, or in deep soils with a high water table). In this case it is set to 0.

When the standard three-layer soil configuration is used, CLASS provides a means of accounting for the possibility of the depth to bedrock falling within the thick third layer, and therefore of phase changes of water taking place in only the upper part of the layer, by introducing the variable TBASROT, which refers to the temperature of the lower part of the layer containing the bedrock. At the beginning of the time step the temperature of the upper part of the layer is disaggregated from the overall average layer temperature using the saved value of TBASROT. The heat flow between the upper part of the soil layer and the lower part is diagnosed from the heat flux at the top of the layer. The upper layer temperature and TBASROT are stepped ahead separately, and the net heat flux in the upper part of the layer is used in the phase change of water if appropriate. The upper layer temperature and TBASROT are re-aggregated at the end of the time step to yield once again the overall average layer temperature.

Two variables, assumed to be constant over the grid cell, are provided if required for atmospheric model runs:

- GGEOROW Geothermal heat flux $[Wm^{-2}]$
- Z0ORROW Orographic roughness length [m]

Unless the soil depth is very large and/or the run is very long, the geothermal heat flux can be set to zero. Z0↩ ORROW is the surface roughness length representing the contribution of orography or other terrain effects to the overall roughness, which becomes important when the modelled grid cell is very large (e.g. in a GCM). For field studies it can be set to zero.

Finally, four parameters are required for modelling lateral movement of soil water: GRKFROT, WFCIROT, WFSF↩ ROT and XSLPROT. However, the routines for interflow and streamflow modelling are still under development, so unless the user is involved in this development, these parameters can be set to arbitrary values, since they will not be used.

1.3.4 Initialization of Prognostic Variables

CLASS requires initial values of the land surface prognostic variables, either from the most recent atmospheric model integration or from field measurements. These are listed below, with guidelines for specifying values for each.

- ALBSROT Snow albedo [ ]
- CMAIROT Aggregated mass of vegetation canopy $[kgm^{-2}]$

- GROROT Vegetation growth index [ ]
- QACROT Specific humidity of air within vegetation canopy space $[kgkg^{-1}]$
- RCANROT Intercepted liquid water stored on canopy $[kgm^{-2}]$
- RHOSROT Density of snow $[kgm^{-3}]$
- SCANROT Intercepted frozen water stored on canopy $[kgm^{-2}]$
- SNOROT Mass of snow pack $[kgm^{-2}]$
- TACROT Temperature of air within vegetation canopy [K]
- TBARROT Temperature of soil layers [K]
- TBASROT Temperature of bedrock in third soil layer [K]
- TCANROT Vegetation canopy temperature [K]
- THICROT Volumetric frozen water content of soil layers $[m^3m^{-3}]$
- THLQROT Volumetric liquid water content of soil layers $[m^3m^{-3}]$
- TPNDROT Temperature of ponded water [K]
- TSFSROT Ground surface temperature over subarea [K]
- TSNOROT Snowpack temperature [K]
- WSNOROT Liquid water content of snow pack $[kgm^{-2}]$
- ZPNDROT Depth of ponded water on surface [m]

1) TBARROT, THLQROT and THICROT are required for each of the modelled soil layers. Thin soil layers near the surface equilibrate quickly, but thicker, deeper layers respond more slowly, and long-term biases can be introduced into the simulation if their temperatures and moisture contents are not initialized to reasonable values. For the moisture contents, it may be better to err on the low side, since soil moisture recharge typically takes place on shorter time scales than soil moisture loss. Field capacity is commonly used as an initial value. If the soil layer temperature is above freezing, the liquid moisture content would be set to the field capacity and the frozen moisture content to zero; if the layer temperature is below zero, the liquid moisture content would be set to the minimum value and the frozen moisture content to the field capacity minus the minimum value. Very deep soil temperatures do not have a large effect on surface fluxes, but errors in their initial values can adversely effect hydrological simulations. If the standard three-layer soil configuration is being used, TBASROT should be set to the third soil layer temperature; otherwise it can be arbitrarily set to zero. For rock or ice layers, THLQROT and THICROT should both be set to zero.

2) It is best to begin a simulation in snow-free conditions, so that the snow simulation can start from the simplest possible state where SNOROT, TSNOROT, ALBSROT, RHOSROT and WSNOROT are all initialized to zero. If erroneous values of the snow variables are specified as initial conditions, this can lead to a persistent bias in the land surface simulation.

3) The vegetation canopy has a relatively small heat capacity and water storage capacity compared with the soil, so its temperature and intercepted water stores equilibrate quite quickly. TCANROT and TACROT can be initialized to the air temperature and QACROT to the air specific humidity. RCANROT and SCANROT can be initialized to zero. CMAIROT, which is used only in the diagnostic energy balance check during the time step, can also be set to zero.

4) GROROT should be initialized to 1 during the growing season and to 0 otherwise.

5) Surface ponded water is a small term and is ephemeral in nature, so ZPNDROT and TPNDROT can both be initialized to zero. TSFSROT is included simply to provide a first guess for the surface temperature iteration in the next time step, so it can be initialized to an arbitrary value. For the snow-covered subareas of the surface it can be set to the freezing point of water; for the snow-free subareas it can be set to the temperature of the first soil layer.

1.4 Running CLASS-CTEM

Some points:

1. runclass36ctem.f is the primary driver via which most input data is read (CTEM's initialization files are read in io_driver.f90) and subroutines are called.
2. The model initialization files include one for CLASS (SITENAME.INI) and one for CTEM (SITENAME.CTM). Examples provided in the Benchmarks folder.
3. The file containing meteorological data (SITENAME.MET). Also provided in the Benchmarks folder.
4. A Makefile is included which compiles all .f and .f90 files and generates an executable called CLASS36C↩TEM. The standard Makefile, currently set up for pgf90 (but can also be setup for xlf and gfortran) may not be suitable for your Unix station and might require modification. You may need your system administrator's help to do this. Note that when the code is compiled it generates .mod files that corresponds to the Fortran modules.

Other than the .INI, .CTM and .MET files which are mandatory, some other CTEM related files are also required depending on which CTEM functionality is switched on. These required files are discussed further down in this user guide.

1.4.1 One thing to note about composite versus mosaic running of the model

In the composite mode, the structural vegetation attributes (including leaf area index, vegetation height, rooting depth) of PFTs that exist in a grid cell are averaged in proportion to their fractional coverages and then used in the grid-averaged energy and water balance calculations. As a result the entire grid cell is characterized by land surface physical environment (including soil temperature, soil moisture, fractional snow cover, and net radiation) that is common to all PFTs. In contrast, in the mosaic mode a grid box is split into multiple tiles representing individual PFTs for each of which energy, water and carbon balance calculations are performed separately. The mosaic mode is, however, able to represent all forms of sub-grid scale variabilities. In principle, the mosaic mode may be used to represent tiles that are characterized by different soil types. For the purposes of coupling to CTEM, however, each tile represents each PFT.

In the composite mode then there is only one tile which covers the whole grid cell and the fractional coverage of CLASS's 4 PFTs and CTEM's 9 PFTs are specified or dynamically modelled by CTEM. In the mosaic mode, the grid cell is divided into multiple tiles and each tile is assumed to be covered by only one PFT (for simplicity) although the entire grid cell is still assumed to be characterized by the same soil type. [38] describe the composite and mosaic approaches for the CLASS-CTEM modelling framework.

Although pretty clever, running the mosaic version and interpreting the model results can be a logical nightmare, especially with competition on where the fractions of different tiles/mosaic change with time. So if you are new at this, please consider running the model in the composite mode.

1.4.2 The Input Files

1.4.2.1 The MET file (Meteorological forcings)

The .MET files contain (usually) half-hourly values of all seven variables that are needed to drive the CLASS land surface scheme. Please see the Benchmarks folder for an example .MET file. The format of the .MET file cannot be changed since CLASS reads in these data using a fixed FORTRAN format (1X,I2,I3,I5,I6,2F9.2,E14.4,F9.2,E12.↩3,F8.2,F12.2,3F9.2,F9.4).

1.4.2.2 The INI file (CLASS's initialization file)

The .INI file is the initialization file for CLASS, and includes initialization data for soil moisture, temperature, and other related fields. Like the .MET file, the .INI file also uses fixed FORTRAN formats to read data, so caution must

be taken when changing the .INI file fields. In addition, note that when the code is run with dynamic vegetation (when CTEM is switched on) several of the vegetation- related fields specified in the .INI file are replaced by those estimated by CTEM. These fields include:

- Leaf area index (LAI)
- Log of roughness length
- Visible and near-infrared albedos
- Canopy mass
- Rooting depth

In addition, CTEM calculates stomatal resistance values so parameters related to CLASS' stomatal resistance formulation and the associated parameters in the .INI file (RSMN, QA50, VPDA, VPDB, PSGA and PSGB) do not matter.

If CLASS is being run on its own, without CTEM, proper specification of vegetation related attributes and parameters is crucial.

Version 2.0 of CTEM adds the capability to divide each cell into several mosaics or 'tiles' as mentioned above. When running for multiple tiles, Both the CLASS and CTEM calculations are performed over each tile. The prognostic variables are saved for each tile in addition to grid averaged values.

A sample .INI file is found in the Benchmarks folder. The INI file has very speficic formatting. You can see it in runclass36ctem.f where the INI file is read and written (search for file units 10 and 100, respectively). Below is an explanation of the variables found in the INI file:

- ZRFM and ZRFH (m): The reference height at which the climate variables (wind speed, temperature and specific humidity) are provided. If the model is driven by the atmospheric model forcing data, these heights would vary by time step. If the model is driven by field data (such as the sample file here), these heights refer to the measurement height of these variables.
- ZBLD (m): Atmospheric blending height. Here it is assigned a value of 50 m.
- GC: GCM surface description variable. For land surface (including inland water) it has a value of -1.
- Plant growth index: CLASS uses specified maximum and minimum leaf area indices (LAIs) for each of its four plant functional types (PFTs). The LAI goes from this minimum to the maximum specified LAI at a specified rate and dependent on the soil temperature. Plant growth index determines what the LAI would be at initialization. This is little tricky if you aren't familiar with CLASS, so it's best left at zero. In addition, when coupled to CTEM this doesn't matter because then CLASS uses LAI simulated by CTEM. When CTEM is switched on then all vegetation-related fields are estimated by CTEM. Competition between PFTs is now simulated and so if a user switches on this option then the fractional coverage of PFTs that grow in a grid cell is also simulated dynamically by CTEM. However, please read the discussion related to each of CTEM's capabilities associated with each switch further down this user guide.
- NLTEST and NMTEST: Number of grid cells and the number of mosaic tiles. NLTEST can not be greater than NLAT. Also, NMTEST must not be greater than NMOS. Typically this standalone code is run for 1 grid cell (NLTEST=1) with 1 or more mosaic tiles (NMTEST=1 or more).
- RSMN (s m-1), QA50 (W/m2), VPDA, VPDB, PSGA and PSGB: Used in stomatal resistance calculation in CLASS subroutine CANALB. Note that these values do not matter when CTEM is switched on. Typical values for the four CLASS vegetation categories are listed above in section **Vegetation Data**.
- DRN: Set it to 1 to allow drainage. Set it to 0 if drainage is suppressed at the bottom of the soil layer. Coupling of CLASS 3.6 to CTEM required a lot of tuning because it seems CLASS 3.6 is drier than CLASS 2.7. As a result a value of 0.1 has been lately used the CCCma Earth system model.
- SDEP (m): Depth to bedrock in the soil profile
- XSLP, GRKF, WFSF, WFCI: Parameters used when running MESH code, not used in Version 3.6 of CLASS.

- TBAR, THLQ and THIC: Thin soil layers near the surface equilibrate quickly, but thicker, deeper layers respond more slowly. Long-term biases can be introduced into the simulation if the soil layers temperatures and moisture contents are not initialized accurately. For the moisture contents, it is better to err on the low side, since soil moisture recharge typically takes place on shorter scales than soil moisture loss. Very deep soil temperatures do not have a large effect on surface fluxes, but errors in their initial values can affect hydrological simulations. For rock or ice layers, THLQ and THIC should both be set to zero.
- RCAN (kg/m2): Intercepted liquid water stored on canopy. RCAN can be initialized to zero.
- SCAN (kg/m2): Intercepted frozen water stored on canopy The vegetation canopy has a relatively small heat capacity and water storage capacity relative to the soil, so its temperature and intercepted water stores equilibrate quickly. TCAN can be initialized to the air temperature. SCAN can be initialized to zero.
- SNO (kg/m2): Mass of snow pack.
- ALBS: Snow albedo.
- RHOS (kg/m3): Snow density

The above three variables and TSNO can be all initialized to 0 in snow-free conditions.

- TPND and ZPND: Surface ponded water is a small term and is ephemeral in nature, so ZPNDROW and TPNDROW can both be initialized to zero.
- GRO: Vegetation growth index, should be initialized to 1 during the growing season and to 0 otherwise. If CTEM is switched on, these values do not matter.
- DELZ and ZBOT: The standard operational configuration for CLASS consists of three soil layers, of thicknesses 0.10 m, 0.25 m and 3.75 m, and thus of bottom depths 0.10, 0.35 and 4.10 m respectively. Version 3.6 supports other options: the third soil layer may be replaced with a larger number of thinner layers, and/or the bottom of the soil profile may be extended below 4.10 m. However, care must be taken not to make the soil layers too thin since this may lead to numerical instability.

1.4.2.3 Typical values of vegetation-related fields for CLASS-only simulations

If you run CLASS without CTEM you will need to specify all vegetation-related parameters. The table below shows typical values you may want to use for different vegetation types.

Biome	CLASS PFT	Visible albedo	Near-infrared Albedo	Roughness length (m)
Needleleaf evergreen forest	1	0.03	0.19	1.5
Needleleaf deciduous forest	1	0.03	0.19	1.0
Broadleaf evergreen forest	2	0.03	0.23	3.5
Broadleaf cold deciduous forest	2	0.05	0.29	2.0
Broadleaf tropical evergreen forest	2	0.03	0.23	3.0
Broadleaf tropical drought deciduous forest	2	0.05	0.29	0.8
Broadleaf evergreen shrub	4	0.03	0.19	0.05
Broadleaf deciduous shrub	2	0.05	0.29	0.15
Broadleaf thorn shrub	2	0.06	0.32	0.15
Short grass and forbs	4	0.06	0.34	0.02
Long grass	4	0.05	0.31	0.08
Arable	3	0.06	0.34	0.08
Rice	3	0.06	0.36	0.08
Sugarcane	3	0.05	0.31	0.35
Maize	3	0.05	0.33	0.25
Cotton	3	0.07	0.43	0.10
Irrigated crop	3	0.07	0.36	0.08
Tundra	4	0.05	0.29	0.01
Swamp	4	0.03	0.25	0.05
Urban	5	0.09	0.15	1.35

Biome	CLASS PFT	Max. LAI (m2/m2)	Min. LAI (m2/m2)	Aboveground biomass (kg/m
Needleleaf evergreen forest	1	2.0	1.6	25.0
Needleleaf deciduous forest	1	2.0	0.5	15.0
Broadleaf evergreen forest	2	8.0	8.0	50.0
Broadleaf cold deciduous forest	6.0	0.5	20.0	2.0
Broadleaf tropical evergreen forest	8.0	8.0	40.0	5.0
Broadleaf tropical drought deciduous forest	4.0	4.0	15.0	5.0
Broadleaf evergreen shrub	2.0	2.0	2.0	0.2
Broadleaf deciduous shrub	4.0	0.5	8.0	1.0
Broadleaf thorn shrub	3.0	3.0	8.0	5.0
Short grass and forbs	3.0	3.0	1.5	1.2
Long grass	4.0	4.0	3.0	1.2
Arable	4.0	0.0	2.0	1.2
Rice	6.5	0.0	2.0	1.2
Sugarcane	5.0	0.0	5.0	1.0
Maize	4.0	0.0	5.0	1.5
Cotton	5.0	0.0	2.0	2.0
Irrigated crop	4.0	0.0	2.0	5.0
Tundra	1.5	1.5	0.2	0.1
Swamp	1.5	1.5	1.0	5.0
Urban	-	-	-	-

1.4.2.4 CLASS and CTEM PFTs

While CLASS models all physical processes related to energy and water balance for four PFTs, CTEM models its terrestrial ecosystem processes for nine PFTs. The nine PFTs of CTEM, however, line up with CLASS' four PFTs perfectly as shown in Table 4. Needleleaf trees are divided into their evergreen and deciduous versions, broadleaf trees are divided into evergreen and cold and drought/dry deciduous versions, and crops and grasses are divided into their C3 and C4 versions based on their photosynthetic pathway.

	CTEM PFTs	CLASS PFTs
1.	Needleleaf Evergreen	Needleleaf
2.	Needleleaf Deciduous	
3.	Broadleaf Evergreen	Broadleaf
4.	Broadleaf Cold Deciduous	
5.	Broadleaf Drought/Dry Deciduous	
6.	C3 Crop	Crops
7.	C4 Crop	
8.	C3 Grass	Grasses
9.	C4 Grass	

Chapter 2

CONTRIBUTING

This contribution guide is only a draft.

CLASS-CTEM code uses an self-documenting tool called Doxygen. All new code should follow the doyxgen conventions and formatting.

Doxygen

Overview

Doxygen is a tool for producing documentation from commented code. For input it supports many code languages, including Fortran. Both fixed form code such as Fortran 77 (extension .f or .for) and free form code such as Fortran 90 (extension .f90) are recognized. There are many different supported outputs, including LaTeX and HTML.

Quick Start (if you have made change to the code and doxygen comments)

1. `Download Doxygen` to your computer

1. Run the command 'doxygen Doxyfile'.

1. View the produced HTML by opening 'index.html' from within the newly created 'html' folder within your current directory.

1. View the produced LaTeX pdf by using the created makefile: A. Run 'make' on the command line in the 'latex' folder in your current project's directory. A. Open the new pdf 'refman.pdf'

Documenting The Code Through Comments Doxygen Recognizes

Doxygen will ignore regular code comments. Thus comments meant to be purely internal to the code can still be made. Commenting is slightly different depending on the version of Fortran being used.

Free Form Fortran (.f90)

Start a comment the regular way with a '!' and then add either '<' or '>' to make it into a comment block that Doxygen will recognize. To continue the comment on another line '!!' may also be used instead of '!>'.

Fixed Form Fortran (.f, .for)

On a new line (not directly after other code) start a regular comment with the capital letter 'C', then add either '<' or '>'. Comments can again be carried onto another line either using 'C!' or 'C>'. It also appears to be fine for the older structured code to also use the newer style's type of comment.

'<' versus '>'

'>' : This indicates to Doxygen that the comment refers to the code after the comment. '<' : This indicates to Doxygen that the comment refers to the code before the comment.

For example;

```
top fortran code snippet
!> comment 1
middle fortran code snippet
!< comment 2
bottom fortran code snippet
```

will have both comments being related to the middle code snippet.

As copied from the 'getting started' documentation page: "Place a special documentation block in front of the declaration or definition of the member, class or namespace. For file, class and namespace members it is also allowed to place the documentation directly after the member."

In general, comment blocks must be before the code (and thus use '!>' and '!!' but not '!<'), with a few exceptions such as class and namespace members.

Beyond Standard Comment Blocks

To make 'pretty' formatting, you can use `markdown` and doyxgen will understand it (the switch for this is turned on in our doxyfile).

Sometimes there is information that should be documented but is not necessarily related to any particular part of the code. In this case structural commands or other specific commands can be put directly before the comment to tell Doxygen where to put the documentation.

For example:

```
!>
```

Chapter 3

The coupled Canadian Land Surface Scheme and Canadian Terrestrial Ecosystem Model (CLASS-CTEM)

Please see the `wiki` for instructions on how to access the CLASS-CTEM manual and run the model.

The https://gitlab.com/jormelton/classctem/blob/master/CONTRIBUTING.md "Contribution Guide" details how your code changes can be made to conform to the doxygen standard.

Chapter 4

Module Index

4.1 Modules

Here is a list of all modules:

Chapter 5

Data Type Index

5.1 Data Types List

Here are the data types with brief descriptions:

Chapter 6

File Index

6.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 7

Module Documentation

7.1 Competition_scheme_bioclim

Canadian Terrestrial Ecosystem Model (CTEM) Bioclimatic Parameters Estimation Subroutine.

- subroutine, public competition_scheme::bioclim (iday, ta, precip, netrad, il1, il2, nilg, leapnow, tcurm, srpcuryr, dftcuryr, inibioclim, tmonth, anpcpcur, anpecur, gdd5cur, surmncur, defmncur, srplscur, defctcur, twarmm, tcoldm, gdd5, aridity, srplsmon, defctmon, anndefct, annsrpls, annpcp, dry_season_length)

7.1.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Bioclimatic Parameters Estimation Subroutine.

The mortality associated with bioclimatic criteria, $m_{bioclim}$, ensures that PFTs do not venture outside their bioclimatic envelopes. The bioclimatic criteria that determine PFT existence are listed in Table tab:pftparams} for tree PFTs. Bioclimatic limits are not used for the C_3 and C_4 grass PFTs. The bioclimatic limits represent physiological limits to PFT survival that are either not captured in the model or processes that are not sufficiently described by empirical observations to allow their parametrization. Some examples of the latter include a plant's resistance to frost damage and xylem cavitation limits due to moisture stress. The bioclimatic criteria include the minimum coldest month air temperature (T_{min}^{cold}), the maximum coldest month air temperature (T_{max}^{cold}), the maximum warmest month air temperature (T_{max}^{warm}), the minimum number of annual growing degree days above $5\,C$ ($GDD5_{min}$), the minimum annual aridity index (ratio of potential evapotranspiration to precipitation; $arid_{min}$) and the minimum dry season length in a year ($dryseason_{min}$), where the dry season length represents the number of consecutive months with precipitation less than potential evaporation. The bioclimatic indices are updated on a 25 year timescale ($T = 25$) such that the slowly changing value of a bioclimatic index $X(t+1)$ for time $t+1$ is updated using its previous year's value $X(t)$ and its value $x(t)$ for the current year as

$$X(t+1) = X(t)e^{-1/T} + x(t)(1 - e^{-1/T}).$$

Equation (efold}) implies that $63\,\%$ of a sudden change in the value of a bioclimatic index Δx is reflected in $X(t)$ in T years $(1 - e^{T(-1/T)} = 1 - e^{-1} = 0.63)$, while $86\,\%$ of the change is reflected in $2T$ years $(1 - e^{2T(-1/T)} = 1 - e^{-2} = 0.86)$.

7.1.2 Function/Subroutine Documentation

7.1.2.1 subroutine, public competition_scheme::bioclim (integer, intent(in) *iday,* real, dimension(nilg), intent(in) *ta,* real, dimension(nilg), intent(in) *precip,* real, dimension(nilg), intent(in) *netrad,* integer, intent(in) *il1,* integer, intent(in) *il2,* integer, intent(in) *nilg,* logical, intent(in) *leapnow,* real, dimension(nilg), intent(inout) *tcurm,* real, dimension(nilg), intent(inout) *srpcuryr,* real, dimension(nilg), intent(inout) *dftcuryr,* logical, intent(inout) *inibioclim,* real, dimension(12,nilg), intent(inout) *tmonth,* real, dimension(nilg), intent(inout) *anpcpcur,* real, dimension(nilg), intent(inout) *anpecur,* real, dimension(nilg), intent(inout) *gdd5cur,* real, dimension(nilg), intent(inout) *surmncur,* real, dimension(nilg), intent(inout) *defmncur,* real, dimension(nilg), intent(inout) *srplscur,* real, dimension(nilg), intent(inout) *defctcur,* real, dimension(nilg), intent(inout) *twarmm,* real, dimension(nilg), intent(inout) *tcoldm,* real, dimension(nilg), intent(inout) *gdd5,* real, dimension(nilg), intent(inout) *aridity,* real, dimension(nilg), intent(inout) *srplsmon,* real, dimension(nilg), intent(inout) *defctmon,* real, dimension(nilg), intent(inout) *anndefct,* real, dimension(nilg), intent(inout) *annsrpls,* real, dimension(nilg), intent(inout) *annpcp,* real, dimension(nilg), intent(inout) *dry_season_length*)

Parameters

in	*iday*	

Find current month

Find if we are at end of month or not

Update monthly temperature for the current month, and other variables. at the end of the month we will have average of all daily temperatures for the current month.

If its the end of the month then store the monthly temperature and set tcurm equal to zero. also check if this month had water deficit or surplus

this loop doubles up the size of the "wet_dry_mon_index" matrix

If its the end of year, then find the temperature of the warmest and the coldest month

Update long term moving average of bioclimatic parameters in an e-folding sense

7.2 Competition_scheme_existence

Canadian Terrestrial Ecosystem Model (CTEM) PFT Existence Subroutine.

- subroutine, public competition_scheme::existence (iday, il1, il2, nilg, sort, nol2pfts, twarmm, tcoldm, gdd5, aridity, srplsmon, defctmon, anndefct, annsrpls, annpcp, pftexist, dry_season_length)

7.2.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) PFT Existence Subroutine.

7.2.2 Function/Subroutine Documentation

7.2.2.1 subroutine, public competition_scheme::existence (integer, intent(in) *iday,* integer, intent(in) *il1,* integer, intent(in) *il2,* integer, intent(in) *nilg,* integer, dimension(icc), intent(in) *sort,* integer, dimension(ican), intent(in) *nol2pfts,* real, dimension(nilg), intent(in) *twarmm,* real, dimension(nilg), intent(in) *tcoldm,* real, dimension(nilg), intent(in) *gdd5,* real, dimension(nilg), intent(in) *aridity,* real, dimension(nilg), intent(in) *srplsmon,* real, dimension(nilg), intent(in) *defctmon,* real, dimension(nilg), intent(in) *anndefct,* real, dimension(nilg), intent(in) *annsrpls,* real, dimension(nilg), intent(in) *annpcp,* logical, dimension(nilg,icc), intent(out) *pftexist,* real, dimension(nilg), intent(in) *dry_season_length*)

Parameters

in	*iday*	

Constants and parameters are located in ctem_params.f90

go through all grid cells and based on bioclimatic parameters decide if a given pft should exist or not.

needleleaf evergreen

needleleaf deciduous

broadleaf evergreen

broadleaf deciduous cold (see note below pft 5 too)

broadleaf deciduous dry

We don't want both broadleaf species co-existing so if it has PFT 5 remove PFT 4.

c3 and c4 crops

c3 grass

if(tcoldm(i).le.tcoldmax(sort(j)))then

else pftexist(i,j)=.false. endif

c4 grass

if(tcoldm(i).ge.tcoldmin(sort(j)))then

else pftexist(i,j)=.false. endif

7.3 Competition_scheme_competition

Canadian Terrestrial Ecosystem Model (CTEM) PFT Competition Subroutine.

- subroutine, public competition_scheme::competition (iday, il1, il2, nilg, nol2pfts, nppveg, dofire, leapnow, pftexist, geremort, intrmort, gleafmas, bleafmas, stemmass, rootmass, litrmass, soilcmas, grclarea, lambda, burnvegf, sort, pstemmass, pgleafmass, fcancmx, fcanmx, vgbiomas, gavgltms, gavgscms, bmasveg, add2allo, colrate, mortrate)

7.3.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) PFT Competition Subroutine.

7.3.2 Function/Subroutine Documentation

7.3.2.1 subroutine, public competition_scheme::competition (integer, intent(in) *iday,* integer, intent(in) *il1,* integer, intent(in) *il2,* integer, intent(in) *nilg,* integer, dimension(ican), intent(in) *nol2pfts,* real, dimension(nilg,icc), intent(inout) *nppveg,* logical, intent(in) *dofire,* logical, intent(in) *leapnow,* logical, dimension(nilg,icc), intent(in) *pftexist,* real, dimension(nilg,icc), intent(in) *geremort,* real, dimension(nilg,icc), intent(in) *intrmort,* real, dimension(nilg,icc), intent(inout) *gleafmas,* real, dimension(nilg,icc), intent(inout) *bleafmas,* real, dimension(nilg,icc), intent(inout) *stemmass,* real, dimension(nilg,icc), intent(inout) *rootmass,* real, dimension(nilg,iccp1), intent(inout) *litrmass,* real, dimension(nilg,iccp1), intent(inout) *soilcmas,* real, dimension(nilg), intent(in) *grclarea,* real, dimension(nilg,icc), intent(in) *lambda,* real, dimension(nilg,icc), intent(in) *burnvegf,* integer, dimension(icc), intent(in) *sort,* real, dimension(nilg,icc), intent(in) *pstemmass,* real, dimension(nilg,icc), intent(in) *pgleafmass,* real, dimension(nilg,icc), intent(inout) *fcancmx,* real, dimension(nilg,ican), intent(inout) *fcanmx,* real, dimension(nilg), intent(inout) *vgbiomas,* real, dimension(nilg), intent(inout) *gavgltms,* real, dimension(nilg), intent(inout) *gavgscms,* real, dimension(nilg,icc), intent(inout) *bmasveg,* real, dimension(nilg,icc), intent(out) *add2allo,* real, dimension(nilg,icc), intent(out) *colrate,* real, dimension(nilg,icc), intent(out) *mortrate*)

Parameters

in	*iday*	

set competition parameters according to the model chosen

First, let's adjust the fractions if fire is turned on.

Since the biomass pools could have changed, update bmasveg.

Do our usual initialization

initial rank/superiority order for simulating competition. since crops are not in competition their rank doesn't matter and therefore we only have icc-2 ranks corresponding to the remaining pfts. the first icc-4 are tree pfts and the last two are the c3 and c4 grasses.

Estimate colonization and mortality rate for each pft, except for crops whose fractional coverage is prescribed.

colonization rate (1/day). the factor (deltat/963.62) converts npp from u-mol co2-c/m2.sec -> kg c/m2.day

mortality rate is the sum of growth related mortality, intrinsic mortality, and an additional mortality that kicks in when long term averaged bioclimatic conditions become unfavourable for a pft. this last term is based on the binary array pftexist.

—> from here on we assume that we only have icc-numcrops pfts <-— since crops are not part of the competition.

based on npp for each pft find the competition ranks / superiority order for simulating competition. note that crops are not in competition, so the competition is between the remaining pfts. in addition pfts which shouldn't exist in the grid cell because of unfavourable values of long-term climatic conditions are considered inferior.

bubble sort according to colonization rates NOTE - this only works if no tree species are indexed at positions > numtreepfts, i.e. the trees must be a contiguous unit at the start of the indexes. JM Jun 2014

the rank of c3 and c4 grass is also determined on the basis of their npp but grasses are always assumed to be inferior to tree pfts

with the ranks of all pfts in all grid cells we can now simulate competition between them. for lotka-volterra eqns we need a minimum seeding fraction otherwise the pfts will not expand at all.

arrange colonization and mortality rates, and fractions, according to superiority ranks

update fractions and check if all fractions are +ve

with the minimum seeding fraction prescription, especially for lotka volterra eqns the total veg fraction may exceed 1. to prevent this we need to adjust fractional coverage of all non-crop pfts that do not have the minimum fraction.

check again that total veg frac doesn't exceed 1.

map delfrac to chngfrac so that we get change in fraction corresponding to the actual number of pfts

—> from here on we get back to our usual icc pfts <-—

get bare fraction

check if a pft's fractional cover is increasing or decreasing

check if bare fraction increases or decreases

now that we know the change in fraction for every pft we use its npp for spatial expansion and litter generation. we also spread vegetation biomass uniformly over the new fractions, and generate additional litter from mortality if the fractions decrease.

three things can happen here

1. fraciord = 0, which means all npp that was used for expansion becomes litter, due to self/expansion thinning and mortality.
2. fraciord = 1, which means a part of or full npp is used for expansion but some litter may also be generated. the part of npp that is used for expansion needs to be allocated to leaves, stem, and root. rather than doing this here we will let the allocation part handle this. so allocation module will allocate not only the npp that is used for pure vertical expansion but also this npp. but we will do our part here and spread the vegetation biomass over the new increased fraction.
3. fraciord = -1, which means all of the npp is to be used for litter generation but in addition some more litter will be generated from mortality of the standing biomass.

All npp used for expansion becomes litter plus there is additional mortality of the standing biomass. the npp that becomes litter is now spread over the whole grid cell. all biomass from fraction that dies due to mortality is also distributed over the litter pool of whole grid cell.

if bare fraction decreases then chop off the litter and soil c from the decreased fraction and add it to grsumlit & grsumsoc for spreading over the whole grid cell. if bare fraction increases then spread its litter and soil c uniformly over the increased fraction.

if a pft is not supposed to exist as indicated by pftexist and its fractional coverage is really small then get rid of the pft all together and spread its live and dead biomass over the grid cell.

adjust litter and soil c mass densities for increase in barefrac over the bare fraction.

spread litter and soil c over all pfts and the barefrac

get fcanmxs for use by class based on the new fcancmxs

update grid averaged vegetation biomass, and litter and soil c densities

and finally we check the c balance. we were supposed to use a fraction of npp for competition. some of it is used for expansion (this is what we save for allocation), and the rest becomes litter. so for each pft the total c mass in vegetation and litter pools must all add up to the same value as before competition.

check balance over the bare fraction

7.4 Ctem_params_initpftpars

- subroutine, public ctem_params::initpftpars (compete)

7.4.1 Detailed Description

7.4.2 Function/Subroutine Documentation

7.4.2.1 subroutine, public ctem_params::initpftpars (logical, intent(in) *compete*)

Parameters

in	*compete*	true if the competition scheme is on.

7.5 Ctem_statevars

7.6 Disturbance_scheme_disturb

Canadian Terrestrial Ecosystem Model (CTEM) Disturbance Subroutine.

- subroutine, public disturbance_scheme::disturb (stemmass, rootmass, gleafmas, bleafmas,thliq,wiltsm,fieldsm,uwind,vwind,ligh litrmass,prbfrhuc, rmatctem, extnprob, popdon,il1,il2,sort, nol2pfts,grclarea,thice,popdin, lucemcom,dofire,currlat,iday,fsnow,isa

7.6.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Disturbance Subroutine.

CTEM v. 2.0 represents disturbance as both natural and human-influenced fires. The original fire parametrization corresponding to CTEM v. 1.0 is described in [9]. The parametrization has since been adapted and used in several other DGVMs [29] [28] [39] [32]. CTEM v. 2.0 incorporates changes suggested in these studies as well as several new improvements.

Fire in CTEM is simulated using a process-based scheme of intermediate complexity that accounts for all elements of the fire triangle: fuel load, combustibility of fuel, and an ignition source. CTEM represents the probability of a fire occurrence ($P_{\rm f}$), for a representative area of $500\,km^2$ (a_{rep}), as

$$P_{\rm f} = P_{\rm b} P_{\rm i} P_{\rm m},$$

where the right hand side terms represent the fire probabilities that are conditioned on (i) the availability of biomass as a fuel source ($P_{\rm b}$), (ii) the combustibility of the fuel based on its moisture content ($P_{\rm m}$), and (iii) the presence of an ignition source ($P_{\rm i}$). The probability of fire and the subsequent calculations are performed for each PFT present in a grid cell (but the PFT index α is omitted for clarity in Eq. fieya}). Since the CTEM parametrization is based on one fire per day per representative area, the representative area has to be sufficiently small that the requirement of only one fire per day is reasonable, yet sufficiently large such that it is not possible to burn the entire representative area in 1 day. Based on MODIS observed fire counts in Fig. 1 of [32], $500\,km^2$ is an appropriate size to not have more than one fire per day and still be a large enough area to be assumed representative of the grid cell as a whole.

The $P_{\rm b}$ term depends on the aboveground biomass (B_{ag}) available for sustaining a fire (which includes the green and brown leaf mass, stem mass and litter mass, $B_{ag} = C_{\rm L} + C_{\rm S} + C_{\rm D}$). Below a lower threshold of aboveground biomass (B_{low}; $0.2\,kg\,C\,m^{-2}$; similar to [41], and [30]), fire is not sustained and thus has a probability of 0. Above a biomass of $1.0\,kg\,C\,m^{-2}$ (B_{high}), $P_{\rm b}$ is set to 1 as the amount of fuel available is assumed sufficient for fire. $P_{\rm b}$ is then calculated using the aboveground biomass, B_{ag} ($kg\,C\,m^{-2}$) with a linear variation between the upper and lower thresholds as

$$P_{\rm b} = \max\left[0, \min\left(1, \frac{B_{ag} - B_{low}}{B_{high} - B_{low}}\right)\right].$$

The linear decrease of $P_{\rm b}$ from B_{high} to B_{low} reflects the fragmentation of fuel that occurs as biomass decreases. Fuel fragmentation impacts upon area burned as it impedes the fire spread rate [22].

The probability of fire based on the presence of ignition sources ($P_{\rm i}$) is influenced by both natural (lightning) and anthropogenic agents (either intentional or accidental). An initial lightning scalar, ϑ_F, that varies between 0 and 1 is found as

$$\vartheta_F = \max\left[0, \min\left(1, \frac{F_c 2g - F_{low}}{F_{high} - F_{low}}\right)\right],$$

where F_{low} and F_{high} represent lower and upper thresholds of cloud-to-ground lightning strikes ($F_c 2g$, $flashes\,km^{-2}\,month^{-1}$) , respectively. Similar to Eq. (eqn:Pb}), below the lower threshold (F_{low}; $0.25\,flashes\,km^{-2}\,month^{-1}$), ϑ_F is 0 implying lightning strikes are not sufficient to cause fire ignition, above the upper threshold (F_{high}; $10.0\,flashes\,km^{-2}\,month^{-1}$) ϑ_F is 1, as ignition sources now do not pose a constraint on fire. The amount of cloud-to-ground lightning, $F_c 2g$, is a fraction of the total lightning based on the relationship derived by [44] (approximation of their Eqs. 1 and 2) as

$$F_c 2g = 0.22 \exp(0.0059 \times |\Phi|) F_{tot},$$

where Φ is the grid cell latitude in degrees and F_{tot} is the total number of lightning $flashes\,km^{-2}\,month^{-1}$ (both cloud-to-cloud and cloud-to-ground). The probability of fire due to natural ignition, P_i, n, depends on the lightning

scalar, ϑ_F, as

$$P_i, n = y(\vartheta_F) - y(0)(1 - \vartheta_F) + \vartheta_F[1 - y(1)]y(\vartheta_F) = \frac{1}{1 + \exp\left(\frac{0.8 - \vartheta_F}{0.1}\right)}.$$

Fire probability due to ignition caused by humans, P_i, h, is parametrized following [29] with a dependence on population density, p_{d} ($number of people\, km^{-2}$)

$$P_i, h = \min\left[1, \left(\frac{p_{\mathrm{d}}}{p_{thres}}\right)^{0.43}\right],$$

where p_{thres} is a population threshold ($300\, people\, km^{-2}$) above which $P_{i,h}$ is 1. The probability of fire conditioned on ignition, P_{i}, is then the total contribution from both natural and human ignition sources

$$P_{\mathrm{i}} = \max[0, \min\{1, P_{i,n} + (1 - P_{i,n})P_{i,h}\}].$$

The population data used to calculate probability of fire ignition caused by humans and anthropogenic fire suppression (discussed further down in this section) is based on the HYDE 3.1 data set [27]

The probability of fire due to the combustibility of the fuel, P_{m}, is dependent on the soil moisture in vegetation's root zone and in the litter layer. The root-zone soil wetness (ϕ_{root}, Eq. degsoilsat}) is used as a surrogate for the vegetation moisture content and the soil wetness of the top soil layer as a surrogate for the litter moisture content. If a grid cell is covered by snow, P_{m} is set to zero. The probability of fire conditioned on soil wetness in vegetation's rooting zone, $P_{m,V}$, is then

$$P_{m,V} = 1 - \tanh\left[\left(\frac{1.75\ \phi_{root}}{E_{\mathrm{V}}}\right)^2\right],$$

where E_{V} is the extinction soil wetness above which $P_{f,V}$ is reduced to near zero and is set to 0.30.

The probability of fire based on the moisture content in the *duff* layer, $P_{m,D}$, which includes the brown leaf mass (grasses only) and litter mass ($B_{duff} = C_{L,b} + C_{\mathrm{D}}$; $kg\, C\, m^{-2}$), is calculated in a similar way but uses the soil wetness of the first soil layer, (ϕ_1, Eq. phitheta}), as a surrogate for the moisture in the duff layer itself as

$$P_{m,D} = 1 - \tanh\left[\left(\frac{1.75\phi_1}{E_{\mathrm{D}}}\right)^2\right],$$

where the extinction soil wetness for the litter layer, E_{D}, is set to 0.50, which yields a higher probability of fire for the litter layer than for the vegetation for the same soil wetness. P_{m} is then the weighted average of $P_{m,V}$ and $P_{m,D}$ given by

$$P_{\mathrm{m}} = P_{m,V}(1 - f_{duff}) + P_{m,D} f_{duff} f_{duff} = \frac{B_{duff}}{B_{ag}}$$

where f_{duff} is the duff fraction of aboveground combustible biomass.

The area burned (a) is assumed to be elliptical in shape for fires based upon the wind speed and properties of an ellipse

$$a(t) = \pi \frac{l}{2}\frac{w}{2} = \frac{\pi}{2}(v_{\mathrm{d}} + v_{\mathrm{u}})v_{\mathrm{p}}t^2,$$

where l (m) and w (m) are the lengths of major and minor axes of the elliptical area burnt; v_{d} ($km\, h^{-1}$) and v_{u} ($km\, h^{-1}$) are the fire spread rates in the downwind and upwind directions, respectively; v_{p} ($km\, h^{-1}$) is the fire spread rate perpendicular to the wind direction and t is the time (h).

The fire spread rate in the downwind direction (v_{d}) is represented as

$$v_{\mathrm{d}} = v_{d,max}\, g(u)\, h(\phi_{r,d})$$

where $v_{d,max}$ ($km\, h^{-1}$) is the PFT-specific maximum fire spread rate from [32], which is set to zero for crop PFTs (see also ctem_params.f90). The functions $g(u)$ accounts for the effect of wind speed and $h(\phi_{r,d})$ accounts for the effect of rooting zone and duff soil wetness on the fire spread rate, as discussed below.

The wind speed (u; $km\,h^{-1}$) is used to determine the length (l) to breadth (w) ratio, $L_{\rm b}$, of the elliptical area burned by fire

$$L_{\rm b} = \frac{l}{w} = \frac{v_{\rm d} + v_{\rm u}}{2v_{\rm p}} = 1 + 10[1 - \exp(-0.06u)]$$

and its head to back ratio, $H_{\rm b}$, following [32], as

$$H_{\rm b} = \frac{v_{\rm d}}{v_{\rm u}} = \frac{L_{\rm b} + (L_{\rm b}^2 - 1)^{0.5}}{L_{\rm b} - (L_{\rm b}^2 - 1)^{0.5}},$$

which help determine the fire spread rate in the direction perpendicular to the wind speed and in the downward direction. Equations (lb}) and (hb}) are combined to estimate the wind scalar $g(u)$ as

$$g(u) = g(0)\frac{2.0L_{\rm b}}{(1 + 1/H_{\rm b})}\frac{g(u)}{g(0)} = \frac{v_{\rm d}}{v_{\rm p}} = \frac{2.0L_{\rm b}}{(1 + 1/H_{\rm b})},$$

which varies between 0.05 and 1. The lower limit is imposed by the $g(0)$ term, which has a value of 0.05 and represents the fire spread rate in the absence of wind ($u = 0$); the upper limit is assigned a maximum value of 1. The fire spread rate in the absence of wind is essentially the spread rate in the direction perpendicular to the wind speed ($v_{\rm p}$). The value of the $g(0)$ term is derived by considering the case where the wind speed becomes very large. As $u \to \infty$ then $L_{\rm b} \to 11$ and $H_{\rm b} \to 482$, while $g(\infty) = 1$ due to its definition, which yields $g(0) = 0.0455 \approx 0.05$.

The dependence of fire spread rate on the rooting zone and duff soil wetness, $h(\phi_{r,d})$ is represented as

$$h(\phi_{r,d}) = h(\phi_{root})(1 - f_{duff}) + h(\phi_1)f_{duff}h(\phi_{root}) = \left(1 - min\left(1, \frac{\phi_{root}}{E_{\rm V}}\right)\right)^2 h(\phi_1) = \left(1 - min\left(1, \frac{\phi_1}{E_{\rm D}}\right)\right)^2.$$

Both $h(\phi_{root})$ and $h(\phi_1)$ gradually decrease from 1 (when soil wetness is 0 and soil moisture does not constrain fire spread rate) to 0 when soil wetness exceeds the respective extinction wetness thresholds, $E_{\rm V}$ and $E_{\rm D}$.

With fire spread rate determined, and the geometry of the burned area defined, the area burned in 1 day, $a_{\rm 1d}$ ($km^2\,day^{-1}$), following [32], is calculated as

$$a_{\rm 1d} = \frac{\pi v_{\rm d}^2 t^2}{4L_{\rm b}}\left(1 + \frac{1}{H_{\rm b}}\right)^2 = \frac{\pi v_{\rm d}^2 (24^2)}{4L_{\rm b}}\left(1 + \frac{1}{H_{\rm b}}\right)^2$$

by setting t equal to $24\,h$.

The fire extinguishing probability, q, is used to calculate the duration (τ, $days$) of the fire, which in turn is used to calculated the area burned over the duration of the fire, $a_{\tau d}$. q is represented following [29] as

$$q = 0.5 + \frac{\max\left[0, 0.9 - \exp(-0.025\,p_{\rm d})\right]}{2},$$

which yields a value of q that varies from 0.5 to 0.95 as population density, $p_{\rm d}$ ($numberofpeople\,km^{-2}$), increases from zero to infinity. Higher population density thus implies a higher probability of fire being extinguished. q represents the probability that a fire will be extinguished on the same day it initiated and the probability that it will continue to the next day is ($1 - q$). Assuming individual days are independent, the probability that the fire will still be burning on day τ is $(1 - q)^\tau$. The probability that a fire will last exactly $$ days, $P()$, is the product of the probability that the fire still exists at day τ and the probability it will be extinguished on that day hence $P(\tau) = q(1 - q)^\tau$. This yields an exponential distribution of fire duration whose expected value is

$$\overline{\tau} = E(\tau) = \sum_{\tau=0}^{\infty} \tau\, q(1 - q)^\tau = \frac{1 - q}{q}.$$

Based on this fire duration and the area burned in 1 day (Eq. aburned}), the area burned over the duration of the fire ($a_{\tau \rm d}$) (but still implemented in 1 day since the model does not track individual fires over their duration, $km^2\,day^{-1}$) is calculated as

$$a_{\tau \rm d} = E(a_{\rm 1d}\tau^2) = \sum_{\tau=0}^{\infty} a_{\rm 1d}\,\tau^2 q(1 - q)^\tau = a_{\rm 1d}\,\frac{(1 - q)(2 - q)}{q^2}.$$

Finally, and reintroducing the PFT index α, the area burned is extrapolated for a PFT α ($A_{\mathrm{b},\alpha}$, $km^2\,day^{-1}$) to the whole grid cell as

$$A_{\mathrm{b},\alpha} = P_{f,\alpha}\, a_{\tau \mathrm{d},\alpha} \frac{A_{\mathrm{g}} f_\alpha}{a_{rep}},$$

where A_{g} is area of a grid cell (km^2), f_α the fractional coverage of PFT α and a_{rep} the representative area of $500\,km^2$, as mentioned earlier. Area burned over the whole grid cell (A_{b}, $km^2\,day^{-1}$) is then calculated as the sum of area burned for individual PFTs,

$$A_{\mathrm{b}} = \sum_{\alpha=1}^{N} A_{\mathrm{b},\alpha}.$$

Fire emits CO_2, other trace gases, and aerosols as biomass is burned while plant mortality and damage due to fire contribute to the litter pool. The emissions of a trace gas/aerosol species j from PFT α, $E_{\alpha,j}$ ($gspecies(m^{-2} gridcellarea) day^{-1}$) are obtained from a vector of carbon densities $\vec{C}_\alpha = (C_{\mathrm{L}}, C_{\mathrm{S}}, C_{\mathrm{R}}, C_{\mathrm{D}})_\alpha$ ($kg\,C\,m^{-2}$) for its leaf, stem, root and litter components, multiplied by a vector of combustion factors $mho_\alpha = (mho_{\mathrm{L}}, mho_{\mathrm{S}}, mho_{\mathrm{R}}, mho_{\mathrm{D}})_\alpha$, which determines what fraction of leaf, stem, root and litter components gets burned, multiplied by a vector of emissions factors $\Upsilon_j = (\Upsilon_{\mathrm{L}}, \Upsilon_{\mathrm{S}}, \Upsilon_{\mathrm{R}}, \Upsilon_{\mathrm{D}})_j$ ($gspecies(kg\,C\,dryorganicmatter)^{-1}$), and by the area burned $A_{\mathrm{b},\alpha}$ for that PFT.

The dot product of $\vec{C}_\alpha$, Υ_j and mho_α thus yields emissions per unit grid cell area of species j from PFT α,

$$E_{\alpha,j} = ((\vec{C}_\alpha \cdot mho_\alpha) \cdot \Upsilon_j) \frac{A_{\mathrm{b},\alpha}}{A_{\mathrm{g}}} \frac{1000}{450},$$

where the constant 1000 converts $\vec{C}_\alpha$ from $kg\,C\,m^{-2}$ to $g\,C\,m^{-2}$ and the constant 450 ($g\,C\,(kgdryorganicmatter)^{-1}$) converts biomass from carbon units to dry organic matter [32]. The corresponding loss of carbon ($kg\,C\,m^{-2}\,day^{-1}$) from the three live vegetation components (L, S, R) and the litter pool (D) of PFT α is given by

$$H_{\alpha,i} = C_{\alpha,i} mho_i \left(\frac{A_{\mathrm{b},\alpha}}{A_{\mathrm{g}}} \right) \quad i = L, S, R, D.$$

The PFT-specific combustion factors for leaf (mho_{L}), stem (mho_{S}), root (mho_{R}) and litter (mho_{D}) components are summarized in ctem_params.f90. Emission factors for all species of trace gases and aerosols (CO_2, CO, CH_4, H_2, $NHMC$, NO_x, N_2O, total particulate matter, particulate matter less than $2.5\,\mu m$ in diameter, and black and organic carbon) are based on an updated set by [2] listed in Tables 3 and 4 of [32].

Litter generated by fire is based on similar mortality factors, which reflect a PFT's susceptibility to damage due to fire $\vec{\Theta}_\alpha = (\Theta_{\mathrm{L}}, \Theta_{\mathrm{S}}, \Theta_{\mathrm{R}})_\alpha$ (fraction). The contribution to litter pool of each PFT due to plant mortality associated with fire ($kg\,C\,m^{-2}\,day^{-1}$) is calculated as

$$M_\alpha = (\vec{C}_\alpha \cdot \Theta_\alpha) \frac{A_{\mathrm{b},\alpha}}{A_{\mathrm{g}}},$$

which is the sum of contribution from individual live vegetation pools

$$M_{\alpha,i} = C_{\alpha,i} \Theta_{\alpha,i} \left(\frac{A_{\mathrm{b},\alpha}}{A_{\mathrm{g}}} \right) \quad i = L, S, R.$$

The carbon loss terms associated with combustion of vegetation components and litter ($H_{\alpha,i}, i = L, S, R, D$) and mortality of vegetation components ($M_{\alpha,i}, i = L, S, R$) due to fire are used in Eqs. (rate_change_eqns_live↩ _pools}) and (rate_change_eqns_dead_pools}), which describe the rate of change of carbon in model's five pools (however, listed there without the PFT subscript α). The PFT-specific mortality factors for leaf (Θ_{L}), stem (Θ_{S}) and root (Θ_{R}) components are listed in ctem_params.f90.

When CTEM is run with prescribed PFT fractional cover, the area of PFTs does not change and the fire-related emissions of CO_2, other trace gases and aerosols , and generation of litter act to thin the remaining biomass. When competition between PFTs for space is allowed, fire both thins the remaining biomass and through plant mortality creates bare ground, which is subsequently available for colonization. The creation of bare ground depends on the susceptibility of each PFT to stand replacing fire (ζ_{r}, fraction) (see also ctem_params.f90) and the PFT area burned. The fire-related mortality rate, m_{dist} (day^{-1}), used in Eq. (mortality}), is then

$$m_{dist,\alpha} = \zeta_{\mathrm{r},\alpha} \frac{A_{\mathrm{b},\alpha}}{f_\alpha A_{\mathrm{g}}}.$$

After bare ground generation associated with fire, the thinned biomass is spread uniformly over the remaining fraction of a PFT. However, it is ensured that the carbon density of the remaining biomass does not increase to a value above what it was before the fire occurred.

7.6.2 Function/Subroutine Documentation

7.6.2.1 subroutine, public disturbance_scheme::disturb (real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *bleafmas,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,ignd) *wiltsm,* real, dimension(ilg,ignd) *fieldsm,* real, dimension(ilg) *uwind,* real, dimension(ilg) *vwind,* real, dimension(ilg) *lightng,* real, dimension(ilg,icc) *fcancmx,* real, dimension(ilg,iccp1) *litrmass,* real, dimension(ilg) *prbfrhuc,* real, dimension(ilg,icc,ignd) *rmatctem,* real, dimension(ilg) *extnprob,* logical, intent(in) *popdon,* integer *il1,* integer *il2,* integer, dimension(icc) *sort,* integer, dimension(ican) *nol2pfts,* real, dimension(ilg) *grclarea,* real, dimension(ilg,ignd) *thice,* real, dimension(ilg) *popdin,* real, dimension(ilg) *lucemcom,* logical *dofire,* real, dimension(ilg) *currlat,* integer *iday,* real, dimension(ilg) *fsnow,* integer, dimension(ilg,ignd) *isand*)

Parameters

in	*popdon*	if set true use population density data to calculate fire extinguishing probability and probability of fire due to human causes, or if false, read directly from .ctm file
	il1	il1=1
	il2	il2=ilg
	sort	index for correspondence between 9 pfts and size 12 of parameters vectors
	nol2pfts	number of level 2 ctem pfts
	dofire	boolean, if true allow fire, if false no fire.
	stemmass	stem mass for each of the 9 ctem pfts, kgc/m^2
	rootmass	root mass for each of the 9 ctem pfts, kgc/m^2
	gleafmas	green leaf mass for each of the 9 ctem pfts, kgc/m^2
	bleafmas	brown leaf mass
	thliq	liquid soil moisture content
	wiltsm	wilting point soil moisture content
	fieldsm	field capacity soil moisture content
	uwind	wind speed, m/s
	vwind	wind speed, m/s
	fcancmx	fractional coverages of ctem's 9 pfts
	lightng	total $lightning, flashes/(km^2.year)$ it is assumed that cloud to ground lightning is some fixed fraction of total lightning.
	litrmass	litter mass for each of the 9 pfts
	prbfrhuc	probability of fire due to human causes
	extnprob	fire extinguishing probability
	rmatctem	fraction of roots in each soil layer for each pft
	thice	frozen soil moisture content over canopy fraction
	popdin	population density $(people/km^2)$
	lucemcom	land use change (luc) related combustion emission losses, $u - molco2/m2.sec$
	fsnow	fraction of snow simulated by class
	grclarea	gcm grid cell area, km^2

initialize required arrays to zero, or assign value

if not simulating fire, leave the subroutine now.

initialization ends

Find pft areas before

area in km^2

Find the probability of fire as a product of three functions with dependence on total biomass, soil moisture, and lightning

1. Dependence on total biomass

don't allow it to bring in crops since they are not allowed to burn.

Root biomass is not used to initiate fire. For example if the last fire burned all grass leaves, and some of the roots were left, its unlikely these roots could catch fire.

Find average biomass over the vegetated fraction

Sum up the vegetated area

calculate bterm for individual PFTs as well

1. Dependence on soil moisture

This is calculated in a way such that the more dry the root zone of a pft type is, and more fractional area is covered with that pft, the more likely it is that fire will get started. that is the dryness factor is weighted by fraction of roots in soil layers, as well as according to the fractional coverage of different pfts. the assumption here is that if there is less moisture in root zone, then it is more likely the vegetation will be dry and thus the likeliness of fire is more.

First find the dryness factor for each soil layer.

If there is snow on the ground, similarly if not soil, do not allow fire so set betadrgt to 0 for all soil layers otherwise calculate as per normal.

Now find weighted value of this dryness factor averaged over the rooting depth, for each pft

Next find this dryness factor averaged over the vegetated fraction $avgdryns(i) = avgdryns(i) + drgtstrs(i,j) * fcancmx(i,j)$

The litter and brown leaves are not affected by the soil water potential therefore they will react only to the moisture conditions (our proxy here is the upper soil moisture). If it is dry they increase the probability of fire corresponding to the proportion of total C they contribute. Only allow if there is no snow.

Use average root zone vegetation dryness to find likelihood of fire due to moisture.

calculate mterm for each PFT

duff fraction for each PFT, Vivek

$drgtstrs(i,j)$ is ϕ_{root} in Melton and Arora GMDD (2015) paper

no fire likelihood due to moisture if no vegetation

1. dependence on lightning

Dependence on lightning is modelled in a simple way which implies that a large no. of lightning flashes are more likely to cause fire than few lightning flashes.

New approximation of Price and Rind equation. It was developed from a more complete dataset than Prentice and Mackerras. Lightning comes in in units of $flashes/km^2/yr$ so divide by 12 to make per month.

Determine the probability of fire due to human causes this is based upon the population density from the .popd read-in file

——————— Number of fire calculations -———————\

This is not used in CTEM in general but we keep the code here for testing purposes

calculate natural and anthorpogenic ignitions/km2.day the constant 0.25 assumes not all c2g lightning hits causes ignitions, only 0.25 of them do the constant (1/30.4) converts c2g lightning from flashes/km2.month to flashes/km2.day MAKE SURE LIGHTNING IS IN UNITS OF FLASHES/KM2.MONTH

Eqs. (4) and (5) of Li et al. 2012 doi:10.5194/bg-9-2761-2012 + also see corrigendum

calculate fire suppression also as a function of population density. Li et al. (2012) formulation is quite similar to what we already use based on Kloster et al. (2010, I think) but we use Kloster's formulation together with our fire extingishing probability. Here, the fire suppression formulation is just by itself

——————— Number of fire calculations -———————//

calculate fire probability for each PFT. Recall that lightning term is still grid averaged

——————— Number of fire calculations -———————\

This is not used in CTEM in general but we keep the code here for testing purposes

calculate total number of ignitions based natural and anthorpogenic ignitions for the whole grid cell

finally calculate number of fire, noting that not all ignitions turn into fire because moisture and biomass may now allow that to happen, and some of those will be suppressed due to fire fighting efforts

——————— Number of fire calculations -———————//

Calculate area burned for each PFT, make sure it's not greater than the area available, then find area and fraction burned for the whole gridcell

soil moisture dependence on fire spread rate

wind speed, which is gridcell specific

Length to breadth ratio from Li et al. (2012)

head to back ratio from Li et al. (2012).

dependence of spread rate on wind

fire spread rate per PFT

area burned in 1 day for that PFT

fire extinguishing probability as a function of grid-cell averaged population density

account for low suppression in Savanna regions, see above for increase in ignition due to cultural practices

area multipler to calculate area burned over the duration of the fire

per PFT area burned, km^2

Calculate gridcell area burned and fraction

Finally estimate amount of litter generated from each pft, and each vegetation component (leaves, stem, and root) based on their resistance to combustion. Update the veg pools due to combustion.

Set aside these pre-disturbance stem and root masses for use in burntobare subroutine.

Update the pools:

Output the burned area per PFT (the units here are burned fraction of each PFTs area. So if a PFT has 50% gridcell cover and 50% of that burns it will have a burnvegf of 0.5 (which then translates into a gridcell fraction of 0.25). This units is for consistency outside of this subroutine.

If .not. dofire then we enter here and perform the calculations for the emissions since we might have some from luc.

We also estimate CO_2 emissions from each of these components. Note that the litter which is generated due to disturbance is uniformly distributed over the entire area of a given pft, and this essentially thins the vegetation biomass. If compete is not on, this does not change the vegetation fractions, if competition is on a fraction will become bare. That is handled in burntobare subroutine called from competition subroutine.

Calculate the emissions of trace gases and aerosols based upon how much plant matter was burnt

Sum all pools that will be converted to emissions/aerosols (gc/m^2)

Add in the emissions due to luc fires (deforestation) the luc emissions are converted from $umolco_2m-2s-1togcm-2$ (day-1) before adding to tot_emit

Convert burnt plant matter from carbon to dry organic matter using a conversion factor, assume all parts of the plant has the same ratio of carbon to dry organic matter. units: $kgdom/m^2$

Convert the dom to emissions/aerosols using emissions factors units: $gspecies/m^2$

FLAG for the optimization of popd effect on fire I am taking the lterm out as the 'temp' var. So I have made tmp be dimension ilg and I overwrite the lterm (which is written to an output file) here in its place.

7.7 Disturbance_scheme_burntobare

- subroutine, public disturbance_scheme::burntobare (il1, il2, nilg, sort, pvgbioms, pgavltms, pgavscms, fcancmx, burnvegf, stemmass, rootmass, gleafmas, bleafmas, litrmass, soilcmas, pstemmass, pgleafmass, nppveg)

7.7.1 Detailed Description

7.7.2 Function/Subroutine Documentation

7.7.2.1 subroutine, public disturbance_scheme::burntobare (integer, intent(in) *il1,* integer, intent(in) *il2,* integer, intent(in) *nilg,* integer, dimension(icc), intent(in) *sort,* real, dimension(nilg), intent(in) *pvgbioms,* real, dimension(nilg), intent(in) *pgavltms,* real, dimension(nilg), intent(in) *pgavscms,* real, dimension(nilg,icc), intent(inout) *fcancmx,* real, dimension(nilg,icc), intent(in) *burnvegf,* real, dimension(nilg,icc), intent(inout) *stemmass,* real, dimension(nilg,icc), intent(inout) *rootmass,* real, dimension(nilg,icc), intent(inout) *gleafmas,* real, dimension(nilg,icc), intent(inout) *bleafmas,* real, dimension(nilg,iccp1), intent(inout) *litrmass,* real, dimension(nilg,iccp1), intent(inout) *soilcmas,* real, dimension(nilg,icc), intent(in) *pstemmass,* real, dimension(nilg,icc), intent(in) *pgleafmass,* real, dimension(nilg,icc), intent(inout) *nppveg*)

Parameters

in	*nilg*	no. of grid cells in latitude circle (this is passed in as either ilg or nlat depending on mos/comp)
in	*sort*	index for correspondence between 9 ctem pfts and size 12 of parameter vectors
in	*pvgbioms*	initial veg biomass
in	*pgavltms*	initial litter mass
in	*pgavscms*	initial soil c mass
in,out	*fcancmx*	initial fractions of the ctem pfts
in	*burnvegf*	per PFT fraction burned of that PFTs area
in,out	*gleafmas*	green leaf carbon mass for each of the 9 ctem pfts, kgc/m^2
in,out	*bleafmas*	brown leaf carbon mass for each of the 9 ctem pfts, kgc/m^2
in,out	*stemmass*	stem carbon mass for each of the 9 ctem pfts, kgc/m^2
in,out	*rootmass*	roots carbon mass for each of the 9 ctem pfts, kgc/m^2
in,out	*nppveg*	npp for individual pfts, $u - molco_2/m^2.sec$
in,out	*soilcmas*	soil carbon mass for each of the 9 ctem pfts + bare, kgc/m^2
in,out	*litrmass*	litter carbon mass for each of the 9 ctem pfts + bare, kgc/m^2
in	*pstemmass*	grid averaged stemmass prior to disturbance, kgc/m^2
in	*pgleafmass*	grid averaged rootmass prior to disturbance, kgc/m^2

Do some initializations

Account for disturbance creation of bare ground. This occurs with relatively low frequency and is PFT dependent. We must adjust the amount of bare ground created to ensure that we do not increase the density of the remaining vegetation.

Test the pftfraca to ensure it does not cause densification of the exisiting biomass Trees compare the stemmass while grass compares the root mass.

the pstemmass is from before the fire occurred, i.e. no thinning!

adjust the bare frac to accomodate for the changes

the pgleafmass is from before the fire occurred, i.e. no thinning!

adjust the bare frac to accomodate for the changes

Soil and litter carbon are treated such that we actually transfer the carbon to the bare fraction since it would remain in place as a location was devegetated In doing so we do not adjust the litter or soilc density on the remaining vegetated fraction. But we do adjust it on the bare fraction to ensure our carbon balance works out.

only do checks if we actually shifted fractions here.

check if total grid average biomass density is same before and after adjusting fractions

only do checks if we actually shifted fractions here.

then add the bare ground in.

7.8 Hetresg

Canadian Terrestrial Ecosystem Model (CTEM) Heterotrophic respiration Subroutine.

- subroutine, public heterotrophic_respiration::hetresg (litrmass, soilcmas, delzw,thpor,il1,il2,tbar,psisat, b,thliq,zbotw,thiceg,frac,isnow,isand,

7.8.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Heterotrophic respiration Subroutine.

Heterotrophic Respiration Subroutine For Bare Fraction

Heterotrophic respiration, $R_{\rm h}$ ($mol\,CO_2\,m^{-2}\,s^{-1}$), in CTEM is based on respiration from the litter (which includes contributions from the stem, leaf and root components), $R_{h,D}$, and soil carbon, $R_{h,H}$, pools,

$$R_{\rm h} = R_{h,D} + R_{h,H}.$$

Heterotrophic respiration is regulated by soil temperature and moisture and is calculated on a daily time step. The original heterotrophic respiration scheme is described in [7] while the modified parametrization used in CTEM v. 2.0 is detailed in [37] and is briefly described here. Respiration from the litter and soil carbon pools takes the following basic form

$$R_{{\rm h},i} = 2.64 \times 10^{-6}\,\varsigma_i C_i f_{15}(Q_{10}) f(\Psi)_i, i = {\rm D}, {\rm H}.$$

The soil carbon and litter respiration depends on the amount of carbon in these components ($C_{\rm H}$ and $C_{\rm D}$; $kg\,C\,m^{-2}$) and on a PFT-dependent respiration rate specified at $15\,C$ ($\varsigma_{\rm H}$ and $\varsigma_{\rm D}$; $kg\,C\,(kg\,C)^{-1}\,yr^{-1}$; see also ctem_params.f90). The constant 2.64×10^{-6} converts units from $kg\,C\,m^{-2}\,yr^{-1}$ to $mol\,CO_2\,m^{-2}\,s^{-1}$.

The effect of soil moisture is accounted for via dependence on soil matric potential ($f(\Psi)$), described later. The temperature dependency of microbial soil respiration rates has been estimated by several different formulations, ranging from simple Q_{10} (exponential) to Arrhenius-type formulations (see review by [34]). In CTEM, soil temperature influences heterotrophic respiration through a temperature-dependent Q_{10} function ($f_{15}(Q_{10})$). The value of Q_{10} itself is assumed to be a function of temperature following a hyperbolic tan function:

$$Q_{10} = 1.44 + 0.56\,\tanh[0.075(46.0 - T_i)], i = {\rm D}, {\rm H},$$

where $T_{\{D,H\}}$ is the temperature of either the litter or soil carbon pool (C), respectively. The parametrization is a compromise between the temperature-independent Q_{10} commonly found in many terrestrial ecosystem models [18] and the temperature-dependent Q_{10} of [26]. While a constant Q_{10} yields an indefinitely increasing respiration rate with increasing temperature, the formulation of [26] gives a continuously increasing Q_{10} under decreasing temperature, which leads to unreasonably high soil and litter carbon pools at high latitudes in CTEM. The CTEM parametrization avoids these issues with a Q_{10} value of about 2.0 for temperatures less than $20\,C$, while a decreasing value of Q_{10} at temperatures above $20\,C$ ensures that the respiration rate does not increase indefinitely.

The temperature of the litter pool is a weighted average of the temperature of the top soil layer (T_1) and the root temperature ($T_{\rm R}$) as litter consists of leaf, stem, and root litter ($T_{\rm D} = 0.7T_1 + 0.3T_{\rm R}$). The temperature of the soil carbon pool is calculated as the mean soil temperature in the rooting zone based upon the fraction of roots in each soil layer and their temperature. The carbon in each soil layer is not explicitly tracked but assumed to adopt an exponential distribution [25].

The response of heterotrophic respiration to soil moisture is formulated through soil matric potential (Ψ; MPa). While soil matric potential values are usually negative, the formulation uses absolute values to allow its logarithm to be taken. Absolute values of soil matric potential are high when soil is dry and low when it is wet. The primary premise of soil moisture control on heterotrophic respiration is that heterotrophic respiration is constrained both when the soils are dry (due to reduced microbial activity) and when they are wet (due to impeded oxygen supply to microbes) with optimum conditions in-between. The exception is the respiration from the litter component, which is assumed to be continually exposed to air, and thus never oxygen deprived, even when soil moisture content is high ($0.04 > |\Psi| \geq |\Psi_{sat}|$, where Ψ_{sat} is the soil matric potential at saturation). The soil moisture dependence thus varies between 0 and 1 with matric potential as follows:

for $0.04 > |\Psi| \geq |\Psi_{sat}|$

$$f(\Psi)_{\rm H} = 1 - 0.5\frac{\log(0.04) - \log|\Psi|}{\log(0.04) - \log|\Psi_{sat}|} f(\Psi)_D = 1;$$

for $0.06 \geq |\Psi| \geq 0.04$

$$f(\Psi)_{\{}D, H\} = 1;$$

for $100.0 \geq |\Psi| > 0.06$

$$f(\Psi)_{\{}D, H\} = 1 - 0.8\frac{\log|\Psi| - \log(0.06)}{\log(100) - \log(0.06)};$$

for $|\Psi| > 100.0$

$$f(\Psi)_{\{}D, H\} = 0.2.$$

Heterotrophic respiration for bare ground is treated separately in CTEM. The carbon contributions to the bare ground litter and soil carbon pools come via processes such as creation of bare ground due to fire, competition between PFTs and land use change. The heterotrophic respiration is sensitive to temperature and moisture in the same manner as vegetated areas using Eqs. (lithet})–(lastpsi}). The base respiration rates of $\varsigma_{D,bare}$ and $\varsigma_{H,bare}$ are set to 0.5605 and $0.02258\, kg\, C\, (kg\, C)^{-1}\, yr^{-1}$, respectively.

The amount of humidified litter, which is transferred from the litter to the soil carbon pool ($C_{\rm D\rightarrow H}$) is modelled as a fraction of litter respiration ($R_{h,D}$) as

$$C_{\rm D\rightarrow H} = \chi\, R_{h,D}$$

where χ (see also ctem_params.f90) is the PFT-dependent humification factor and varies between 0.4 and 0.5. For crops, χ is set to 0.1 to account for reduced transfer of humidified litter to the soil carbon pool which leads to loss in soil carbon when natural vegetation is converted to croplands. Over the bare ground fraction χ is set to 0.45.

With heterotrophic respiration known, net ecosystem productivity (NEP) is calculated as

$$NEP = G_{canopy} - R_{\rm m} - R_{\rm g} - R_{\rm h}.$$

7.8.2 Function/Subroutine Documentation

7.8.2.1 subroutine, public heterotrophic_respiration::hetresg (real, dimension(ilg,icc+1) *litrmass,* real, dimension(ilg,icc+1) *soilcmas,* real, dimension(ilg,ignd) *delzw,* real, dimension(ilg,ignd) *thpor,* integer *il1,* integer *il2,* real, dimension(ilg,ignd) *tbar,* real, dimension(ilg,ignd) *psisat,* real, dimension(ilg,ignd) *b,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,ignd) *zbotw,* real, dimension(ilg,ignd) *thiceg,* real, dimension(ilg) *frac,* integer *isnow,* integer, dimension(ilg,ignd) *isand*)

Parameters

il1	il1=1
il2	il2=ilg
isnow	integer telling if bare fraction is fg (0) or fgs (1), isnow is changed to isnow(ilg) in classt of class version higher than 3.4 for coupling with ctem
litrmass	litter mass for the 8 pfts + bare in kgc/m^2
soilcmas	soil carbon mass for the 8 pfts + bare in kgc/m^2
tbar	soil temperature, k
thliq	liquid soil moisture content in 3 soil layers
zbotw	bottom of soil layers
frac	fraction of ground (fg) or snow over ground (fgs)
psisat	saturation matric potential
b	parameter b of clapp and hornberger
thpor	porosity

Estimate temperature of the litter and soil carbon pools.

We estimate the temperature of the soil c pool assuming that soil carbon over the bare fraction is distributed exponentially. note that bare fraction may contain dead roots from different pfts all of which may be distributed differently. For simplicity we do not track each pft's dead root biomass and assume that distribution of soil carbon over the bare fraction can be described by a single parameter.

find moisture scalar for soil c decomposition

find moisture scalar for litter decomposition

7.9 Hetresv

- subroutine, public heterotrophic_respiration::hetresv (fcan,fct, litrmass, soilcmas,delzw,thpor, il1,il2,tbar,psisat, b, thliq,roottemp,zbotw,sort,isand, thicec,

7.9.1 Detailed Description

7.9.2 Function/Subroutine Documentation

7.9.2.1 subroutine, public heterotrophic_respiration::hetresv (real, dimension(ilg,icc) *fcan,* real, dimension(ilg) *fct,* real, dimension(ilg,icc+1) *litrmass,* real, dimension(ilg,icc+1) *soilcmas,* real, dimension(ilg,ignd) *delzw,* real, dimension(ilg,ignd) *thpor,* integer *il1,* integer *il2,* real, dimension(ilg,ignd) *tbar,* real, dimension(ilg,ignd) *psisat,* real, dimension(ilg,ignd) *b,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,icc) *roottemp,* real, dimension(ilg,ignd) *zbotw,* integer, dimension(icc) *sort,* integer, dimension(ilg,ignd) *isand,* real, dimension(ilg,ignd) *thicec*)

Parameters

il1	il1=1
il2	il2=ilg
sort	index for correspondence between 9 pfts and 12 values in the parameters vectors
fcan	fractional coverage of ctem's 9 pfts
fct	sum of all fcan, fcan & fct are not used at this time but could be used at some later stage
litrmass	litter mass for the 9 pfts + bare in kgc/m^2
tbar	soil temperature, k
soilcmas	soil carbon mass for the 9 pfts + bare in kgc/m^2
thliq	liquid soil moisture content in 3 soil layers
roottemp	root temperature as estimated in mainres subroutine
zbotw	bottom of soil layers
thicec	liquid soil moisture content in 3 soil layers in canopy covered subarea
psisat	saturation matric potential
b	parameter b of clapp and hornberger
thpor	porosity

7.10 io_driver_read_from_ctm

This subroutine reads in the restart/starting conditions from the .CTM file. The input values are checked and possibly adjusted if the run is intended to be with competition on and if the start_bare flag is true.

- subroutine, public io_driver::read_from_ctm (nltest, nmtest, FCANROT, FAREROT, RSMNROT, QA50ROT, VPDAROT, VPDBROT, PSGAROT, PSGBROT, DRNROT, SDEPROT, XSLPROT, GRKFROT, WFSFROT, WFCIROT, MIDROT, SANDROT, CLAYROT, ORGMROT, TBARROT, THLQROT, THICROT, TCANROT, TSNOROT, TPNDROT, ZPNDROT, RCANROT, SCANROT, SNOROT, ALBSROT, RHOSROT, GROROT, argbuff, onetile_perPFT)

7.10.1 Detailed Description

This subroutine reads in the restart/starting conditions from the .CTM file. The input values are checked and possibly adjusted if the run is intended to be with competition on and if the start_bare flag is true.

7.10.2 Function/Subroutine Documentation

7.10.2.1 **subroutine, public io_driver::read_from_ctm (integer, intent(in) *nltest,* integer, intent(inout) *nmtest,* real, dimension(nlat,nmos,icp1), intent(inout) *FCANROT,* real, dimension(nlat,nmos), intent(inout) *FAREROT,* real, dimension(nlat,nmos,ican), intent(inout) *RSMNROT,* real, dimension(nlat,nmos,ican), intent(inout) *QA50ROT,* real, dimension(nlat,nmos,ican), intent(inout) *VPDAROT,* real, dimension(nlat,nmos,ican), intent(inout) *VPDBROT,* real, dimension(nlat,nmos,ican), intent(inout) *PSGAROT,* real, dimension(nlat,nmos,ican), intent(inout) *PSGBROT,* real, dimension(nlat,nmos), intent(inout) *DRNROT,* real, dimension(nlat,nmos), intent(inout) *SDEPROT,* real, dimension(nlat,nmos), intent(inout) *XSLPROT,* real, dimension(nlat,nmos), intent(inout) *GRKFROT,* real, dimension(nlat,nmos), intent(inout) *WFSFROT,* real, dimension(nlat,nmos), intent(inout) *WFCIROT,* integer, dimension(nlat,nmos), intent(inout) *MIDROT,* real, dimension(nlat,nmos,ignd), intent(inout) *SANDROT,* real, dimension(nlat,nmos,ignd), intent(inout) *CLAYROT,* real, dimension(nlat,nmos,ignd), intent(inout) *ORGMROT,* real, dimension(nlat,nmos,ignd), intent(inout) *TBARROT,* real, dimension(nlat,nmos,ignd), intent(inout) *THLQROT,* real, dimension(nlat,nmos,ignd), intent(inout) *THICROT,* real, dimension(nlat,nmos), intent(inout) *TCANROT,* real, dimension(nlat,nmos), intent(inout) *TSNOROT,* real, dimension(nlat,nmos), intent(inout) *TPNDROT,* real, dimension(nlat,nmos), intent(inout) *ZPNDROT,* real, dimension(nlat,nmos), intent(inout) *RCANROT,* real, dimension(nlat,nmos), intent(inout) *SCANROT,* real, dimension(nlat,nmos), intent(inout) *SNOROT,* real, dimension(nlat,nmos), intent(inout) *ALBSROT,* real, dimension(nlat,nmos), intent(inout) *RHOSROT,* real, dimension(nlat,nmos), intent(inout) *GROROT,* character(80), intent(in) *argbuff,* logical, intent(in) *onetile_perPFT*)

Read from CTEM initialization file (.CTM)

The following three variables are needed to run CTEM. 1) min & 2) max leaf area index are needed to break class lai into dcd and evg for trees (for crops and grasses it doesn't matter much). 3) dvdfcanrow is needed to divide needle & broad leaf into dcd and evg, and crops & grasses into c3 and c4 fractions.

Rest of the initialization variables are needed to run CTEM but if starting from bare ground initialize all live and dead c pools from zero. suitable values of extnprobgrd and prbfrhucgrd would still be required. set stdaln to 1 for operation in non-gcm stand alone mode, in the CTEM initialization file.

If fire and competition are on, save the stemmass and rootmass for use in burntobare subroutine on the first timestep.

Take the first tile value now and put it over the other tiles

Check that a competition or luc run has the correct number of mosaics. if it is not a start_bare run, then nmtest should equal nmos

if this run uses the competition or lnduseon parameterization and starts from bare ground, set up the model state here. this overwrites what was read in from the .ini and .ctm files. for composite runs (the composite set up is after this one for mosaics)

store the read-in crop fractions as we keep them even when we start bare. FLAG: this is setup assuming that crops are in mosaics 6 and 7. JM Apr 9 2014.

check the number of mosaics that came from the .ini file

we need to transfer some initial parameterization info to all mosaics, so set all values to that of the first mosaic.

set the number of mosaics to icc+1

if (nmtest .ne. nmos)

set the initial conditions for the pfts

initalize to zero

set the seed amount for each pft in its mosaic

initial conditions always required

then adjusted below for the actual mosaic makeup

set up for composite runs when start_bare is on and compete or landuseon

store the read-in crop fractions as we keep them even when we start bare. FLAG: this is setup assuming that crops are in pft number 6 and 7. and the first tile contains the information for the grid cell (assumes we have crops in every tile too! JM Apr 9 2014.

initalize to zero, these will be filled in by the luc or competition subroutines.

initial conditions always required

7.11 io_driver_write_ctm_rs

After a set period is complete the restart file for CTEM (.CTM_RS) is written this restart file contains all of the CTEM level information needed to to restart the model to the same state.

- subroutine, public io_driver::write_ctm_rs (nltest, nmtest, FCANROT, argbuff)

7.11.1 Detailed Description

After a set period is complete the restart file for CTEM (.CTM_RS) is written this restart file contains all of the CTEM level information needed to to restart the model to the same state.

7.11.2 Function/Subroutine Documentation

7.11.2.1 subroutine, public io_driver::write_ctm_rs (integer, intent(in) *nltest,* integer, intent(inout) *nmtest,* real, dimension(nlat,nmos,icp1), intent(inout) *FCANROT,* character(80), intent(in) *argbuff*)

if landuseon or competition, then we need to recreate the dvdfcanrow so do so now

check to ensure that the dvdfcanrow's add up to 1 across a class-level pft

Lastly check if the different pfts accidently add up $>$ 1.0 after rounding to the number of sig figs used in the output this rounds to 3 decimal places. if you are found to be over or under, arbitrarily reduce one of the pfts. the amount of the change will be inconsequential.

We can write out the first tile value since these are the same across an entire gridcell.

Just write in the first tiles value since all tiles in a gridcell are the same.

7.12 lo_driver_create_outfiles

All output files are initialized in this subroutine.

- subroutine, public io_driver::create_outfiles (argbuff, title1, title2, title3, title4, title5, title6, name1, name2, name3, name4, name5, name6, place1, place2, place3, place4, place5, place6)

7.12.1 Detailed Description

All output files are initialized in this subroutine.

7.12.2 Function/Subroutine Documentation

7.12.2.1 subroutine, public io_driver::create_outfiles (character(80), intent(in) *argbuff,* character(4), intent(in) *title1,* character(4), intent(in) *title2,* character(4), intent(in) *title3,* character(4), intent(in) *title4,* character(4), intent(in) *title5,* character(4), intent(in) *title6,* character(4), intent(in) *name1,* character(4), intent(in) *name2,* character(4), intent(in) *name3,* character(4), intent(in) *name4,* character(4), intent(in) *name5,* character(4), intent(in) *name6,* character(4), intent(in) *place1,* character(4), intent(in) *place2,* character(4), intent(in) *place3,* character(4), intent(in) *place4,* character(4), intent(in) *place5,* character(4), intent(in) *place6*)

begin:

the ctem output file suffix naming convention is as follows: ".CT##{time}" where the ## is a numerical identifier, {time} is any of H, D, M, or Y for half hourly, daily, monthly, or yearly, respectively.

stand alone mode, includes half-hourly and daily output

ctem half hourly output files

ctem daily output files

ctem_on & not parallelrun

CLASS MONTHLY FOR BOTH PARALLEL MODE AND STAND ALONE MODE

CLASS YEARLY OUTPUT FILES

CTEM monthly output files

Monthly disturbance

CTEM yearly output files

Annual disturbance

ctem pft fractions MONTHLY

ctem pft fractions YEARLY

Methane(wetland) MONTHLY

Methane(wetland) YEARLY

ctem_on & parallelrun

7.13 io_driver_class_monthly_aw

Accumulate and write out the monthly CTEM outputs.

- subroutine, public io_driver::class_monthly_aw (IDAY, IYEAR, NCOUNT, NDAY, SBC, DELT, nltest, nmtest, ALVSROT, FAREROT, FSVHROW, ALIRROT, FSIHROW, GTROT, FSSROW, FDLROW, HFSROT, ROF↩ ROT, PREROW, QFSROT, QEVPROT, SNOROT, TAROW, WSNOROT, TBARROT, THLQROT, THICROT, TFREZ, QFCROT, QFGROT, QFNROT, QFCLROT, QFCFROT, FSGVROT, FSGSROT, FSGGROT, ACTL↩ YR, FTABLE)

7.13.1 Detailed Description

Accumulate and write out the monthly CTEM outputs.

7.13.2 Function/Subroutine Documentation

7.13.2.1 **subroutine, public io_driver::class_monthly_aw (integer, intent(in) *IDAY,* integer, intent(in) *IYEAR,* integer, intent(in) *NCOUNT,* integer, intent(in) *NDAY,* real, intent(in) *SBC,* real, intent(in) *DELT,* integer, intent(in) *nltest,* integer, intent(in) *nmtest,* real, dimension(nlat,nmos), intent(in) *ALVSROT,* real, dimension(nlat,nmos), intent(in) *FAREROT,* real, dimension(nlat), intent(in) *FSVHROW,* real, dimension(nlat,nmos), intent(in) *ALIRROT,* real, dimension(nlat), intent(in) *FSIHROW,* real, dimension(nlat,nmos), intent(in) *GTROT,* real, dimension(nlat), intent(in) *FSSROW,* real, dimension(nlat), intent(in) *FDLROW,* real, dimension(nlat,nmos), intent(in) *HFSROT,* real, dimension(nlat,nmos), intent(in) *ROFROT,* real, dimension(nlat), intent(in) *PREROW,* real, dimension(nlat,nmos), intent(in) *QFSROT,* real, dimension(nlat,nmos), intent(in) *QEVPROT,* real, dimension(nlat,nmos), intent(in) *SNOROT,* real, dimension(nlat), intent(in) *TAROW,* real, dimension(nlat,nmos), intent(in) *WSNOROT,* real, dimension(nlat,nmos,ignd), intent(in) *TBARROT,* real, dimension(nlat,nmos,ignd), intent(in) *THLQROT,* real, dimension(nlat,nmos,ignd), intent(in) *THICROT,* real, intent(in) *TFREZ,* real, dimension(nlat,nmos,ignd), intent(in) *QFCROT,* real, dimension(nlat,nmos), intent(in) *QFGROT,* real, dimension(nlat,nmos), intent(in) *QFNROT,* real, dimension(nlat,nmos), intent(in) *QFCLROT,* real, dimension(nlat,nmos), intent(in) *QFCFROT,* real, dimension(nlat,nmos), intent(in) *FSGVROT,* real, dimension(nlat,nmos), intent(in) *FSGSROT,* real, dimension(nlat,nmos), intent(in) *FSGGROT,* real, dimension(nlat,nmos), intent(in) *ACTLYR,* real, dimension(nlat,nmos), intent(in) *FTABLE*)**

Parameters

in	*fsgvrot*	Diagnosed net shortwave radiation on vegetation canopy
in	*fsgsrot*	Diagnosed net shortwave radiation on ground snow surface
in	*fsggrot*	Diagnosed net shortwave radiation on ground surface

Accumulate output data for monthly averaged fields for class grid-mean. for both parallel mode and stand alone mode

7.14 io_driver_class_annual_aw

- subroutine, public io_driver::class_annual_aw (IDAY, IYEAR, NCOUNT, NDAY, SBC, DELT, nltest, nmtest, ALVSROT, FAREROT, FSVHROW, ALIRROT, FSIHROW, GTROT, FSSROW, FDLROW, HFSROT, ROFROT, PREROW, QFSROT, QEVPROT, TAROW, QFCROT, FSGVROT, FSGSROT, FSGGROT, ACTLYR, FTABLE, leapnow)

7.14.1 Detailed Description

7.14.2 Function/Subroutine Documentation

7.14.2.1 subroutine, public io_driver::class_annual_aw (integer, intent(in) *IDAY,* integer, intent(in) *IYEAR,* integer, intent(in) *NCOUNT,* integer, intent(in) *NDAY,* real, intent(in) *SBC,* real, intent(in) *DELT,* integer, intent(in) *nltest,* integer, intent(in) *nmtest,* real, dimension(nlat,nmos), intent(in) *ALVSROT,* real, dimension(nlat,nmos), intent(in) *FAREROT,* real, dimension(nlat), intent(in) *FSVHROW,* real, dimension(nlat,nmos), intent(in) *ALIRROT,* real, dimension(nlat), intent(in) *FSIHROW,* real, dimension(nlat,nmos), intent(in) *GTROT,* real, dimension(nlat), intent(in) *FSSROW,* real, dimension(nlat), intent(in) *FDLROW,* real, dimension(nlat,nmos), intent(in) *HFSROT,* real, dimension(nlat,nmos), intent(in) *ROFROT,* real, dimension(nlat), intent(in) *PREROW,* real, dimension(nlat,nmos), intent(in) *QFSROT,* real, dimension(nlat,nmos), intent(in) *QEVPROT,* real, dimension(nlat), intent(in) *TAROW,* real, dimension(nlat,nmos,ignd), intent(in) *QFCROT,* real, dimension(nlat,nmos), intent(in) *FSGVROT,* real, dimension(nlat,nmos), intent(in) *FSGSROT,* real, dimension(nlat,nmos), intent(in) *FSGGROT,* real, dimension(nlat,nmos), intent(in) *ACTLYR,* real, dimension(nlat,nmos), intent(in) *FTABLE,* logical, intent(in) *leapnow*)

Parameters

in	*leapnow*	true if this year is a leap year. Only used if the switch 'leap' is true.
in	*fsgvrot*	Diagnosed net shortwave radiation on vegetation canopy
in	*fsgsrot*	Diagnosed net shortwave radiation on ground snow surface
in	*fsggrot*	Diagnosed net shortwave radiation on ground surface

Accumulate output data for yearly averaged fields for class grid-mean. for both parallel mode and stand alone mode

ADD INITIALIZTION FOR YEARLY ACCUMULATED ARRAYS

IDAY.EQ.365/366 .AND. NDAY

7.15 io_driver_ctem_daily_aw

Accumulate and write out the daily CTEM outputs.

- subroutine, public io_driver::ctem_daily_aw (nltest, nmtest, iday, FAREROT, iyear, jdstd, jdsty, jdendd, jdendy, grclarea, onetile_perPFT)

7.15.1 Detailed Description

Accumulate and write out the daily CTEM outputs.

7.15.2 Function/Subroutine Documentation

7.15.2.1 subroutine, public io_driver::ctem_daily_aw (integer, intent(in) *nltest,* integer, intent(in) *nmtest,* integer, intent(in) *iday,* real, dimension(:,:), intent(in) *FAREROT,* integer, intent(in) *iyear,* integer, intent(in) *jdstd,* integer, intent(in) *jdsty,* integer, intent(in) *jdendd,* integer, intent(in) *jdendy,* real, dimension(:), intent(in) *grclarea,* logical, intent(in) *onetile_perPFT*)

write daily ctem results

Reset the grid and tile average variables.

First some unit conversions:

Now for the bare fraction of the grid cell.

Aggregate to the tile avg vars:

Do the bare ground also:

Calculation of grid averaged variables

Write daily ctem results

First the per PFT values to file .CT01D

File: .CT01D

File .CT02D

File $*$.CT03D

File .CT04D

File $*$.CT06D

Now write out the bare fraction values (only needed if you have vars that are affected by barefrac values)

File: .CT01D

File $*$.CT03D

Now write out the tile average values for each tile if the tile number is greater than 1 (nmtest $>$ 1).

File: .CT01D

File .CT02D

File .CT03D

File .CT04D

Finally do the grid avg values:

File: .CT01D

File .CT02D

File .CT03D

File .CT04D

File ∗.CT06D

File .CT08D

7.16 Io_driver_ctem_monthly_aw

- subroutine, public io_driver::ctem_monthly_aw (nltest, nmtest, iday, FAREROT, iyear, nday, onetile_perPFT)

7.16.1 Detailed Description

7.16.2 Function/Subroutine Documentation

7.16.2.1 subroutine, public io_driver::ctem_monthly_aw (integer, intent(in) *nltest,* integer, intent(in) *nmtest,* integer, intent(in) *iday,* real, dimension(:,:), intent(in) *FAREROT,* integer, intent(in) *iyear,* integer, intent(in) *nday,* logical, intent(in) *onetile_perPFT*)

Accumulate monthly outputs

Accumulate monthly outputs at the per PFT level.

Also do the bare ground

Accumulate monthly outputs at the per tile level.

take mean wind speed and convert to km/h

Do the mid-month variables (these are not accumulated, we just keep the mid month value for printing in the monthly file)

Do the bare fraction too

Now find the per tile values:

Also add in the bare fraction contributions.

Now find the gridcell level values:

Do the end of month variables

Convert some quantities into per day values

Find the monthly outputs at the per tile level from the outputs at the per PFT level

Find the monthly outputs at the per grid cell level from the outputs at the per tile level

Write to file .CT01M

First the per PFT values:

Now write out the bare fraction values:

Now write out the tile average values for each tile if the tile number is greater than 1 (nmtest $>$ 1).

Write to file .CT06M

First the per PFT values:

Now write out the tile average values for each tile if the tile number is greater than 1 (nmtest $>$ 1).

Add fraction of each pft and bare

Prepare the monthly vars for the next month:

Reset all end of month accumulated arrays

if(iday.eq.monthend(nt+1))

nt=1,nmon

7.17 lo_driver_ctem_annual_aw

- subroutine, public io_driver::ctem_annual_aw (nltest, nmtest, iday, FAREROT, iyear, onetile_perPFT, leapnow)

7.17.1 Detailed Description

7.17.2 Function/Subroutine Documentation

7.17.2.1 subroutine, public io_driver::ctem_annual_aw (integer, intent(in) *nltest,* integer, intent(in) *nmtest,* integer, intent(in) *iday,* real, dimension(:,:), intent(in) *FAREROT,* integer, intent(in) *iyear,* logical, intent(in) *onetile_perPFT,* logical, intent(in) *leapnow*)

Parameters

in	*leapnow*	true if this year is a leap year. Only used if the switch 'leap' is true.

Accumulate yearly outputs

Accumulate the variables at the per PFT level

Also do the bare fraction amounts

Accumulate the variables at the per tile level

The pools are looked at just at the end of the year.

Add values to the per tile vars

Add values to the per gridcell vars

Write to annual output files:

File .CT01Y

Now do the bare fraction of the grid cell. Only soil c, hetres and litter are relevant so the rest are set to 0.

Write out the per tile values

Finally write out the per gridcell values

Write to file .CT06Y

Write out the per tile values

Finally write out the per gridcell values

Write fraction of each pft and bare

Write out the per tile values

Reset all annual vars in preparation:

7.18 io_driver_close_outfiles

- subroutine, public io_driver::close_outfiles ()

7.18.1 Detailed Description

7.18.2 Function/Subroutine Documentation

7.18.2.1 subroutine, public io_driver::close_outfiles ( )

CLOSE CLASS OUTPUT FILES

then the CTEM ones

7.19 Landuse_change_initialize_luc

Canadian Terrestrial Ecosystem Model (CTEM) LUC Initial Read-In Subroutine.

- subroutine, public landuse_change::initialize_luc (iyear, lucdat, nmtest, nltest, nol2pfts, cyclemet, cylucyr, lucyr, fcanrow, farerow, nfcancmxrow, pfcancmxrow, fcancmxrow, reach_eof, start_bare, compete, onetile_↩ perPFT)

7.19.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) LUC Initial Read-In Subroutine.

7.19.2 Function/Subroutine Documentation

7.19.2.1 subroutine, public landuse_change::initialize_luc (integer, intent(in) *iyear,* character(80), intent(in) *lucdat,* integer, intent(in) *nmtest,* integer, intent(in) *nltest,* integer, dimension(ican), intent(in) *nol2pfts,* logical, intent(in) *cyclemet,* integer, intent(in) *cylucyr,* integer, intent(out) *lucyr,* real, dimension(nlat,nmos,icp1), intent(inout) *fcanrow,* real, dimension(nlat,nmos), intent(inout) *farerow,* real, dimension(nlat,nmos,icc), intent(out) *nfcancmxrow,* real, dimension(nlat,nmos,icc), intent(out) *pfcancmxrow,* real, dimension(nlat,nmos,icc), intent(out) *fcancmxrow,* logical, intent(inout) *reach_eof,* logical, intent(in) *start_bare,* logical, intent(in) *compete,* logical, intent(in) *onetile_perPFT*)

Parameters

in	*onetile_perpft*	if you are running with one tile per PFT in mosaic mode, set to true. Changes how competition is run. Specifically it allows competition between tiles. This is not recommended for any case where you don't have one PFT in each tile as it has not been tested for that.

Initialize barefraction to 1.0

reset the composite fcanrow as it is appended on later in a loop

it is the first year, so prepare the luc data:

open the luc file

Skip first three rows:

get first year of luc data note we load the nfcancmx, not pfcancmx array. this is because this nfcancmx value is passed to the pfcancmx array at the start of each simulation year

next update our luc data if either of the following conditions are met:

1) we are cycling the met data and the luc year we just read in is less than the year we want to cycle over (assuming it is not defaulted to -9999) or,

2) we are not cycling over the met data so we just want to get the same year of luc data as the met data we read in above in preparation for our transient run.

If you are running with start_bare on, take in only the crop fractions, set rest to seed. If compete, but not start bare, then just make sure you have at least seed for each pft.

not starting bare, but still make sure you have at least seed

Keep track of the non-crop nfcancmx for use in loop below. pftarrays keeps track of the nfcancmxrow for all non-crops indexposj and indexposm store the index values of the non-crops in a continuous array for use later. n and k are then the indexes used by these arrays.

check that in making these seed fraction we haven't made our total fraction more than 1.0.

Find out which of the non-crop PFTs covers the largest area.

j is then the nmos index and m is the icc index of the PFT with the largest area

Reduce the most dominant PFT by barf and minbare. The extra amount is to ensure we don't have trouble later with an extremely small bare fraction. barf is a negative value.

Find out which of the non-crop PFTs covers the largest area.

j is then the nmos index and m is the icc index of the PFT with the largest area

Reduce the most dominant PFT by barf and minbare. The extra amount is to ensure we don't have trouble later with an extremely small bare fraction. barf is a negative value.

get fcans for use by class using the nfcancmxs just read in

this tile has some plants so overwrite the seed fraction with an actual fraction

note: the seed fraction has already been assigned in runclassctem prior to entering this subroutine.

(re)find the bare fraction for farerow(i,iccp1)

check that the bare fraction is possible ($>$0) and if not then reduce the other pfts proportionally to make a non-negative bare ground fraction.

competition requires a 'seed' fraction so make sure the bare ground is also that big. for prescribed runs you just need it to be possible ($>$0).

assign the present pft fractions from those just read in

back up one year in the luc file this is because we were setting things up here, we will later call readin_luc so want the file to be rewound prior to that to the proper start year.

end of the luc file is reached. close and tell main program to exit

7.20 Landuse_change_readin_luc

Canadian Terrestrial Ecosystem Model (CTEM) LUC Annual Read-In Subroutine.

- subroutine, public landuse_change::readin_luc (iyear, nmtest, nltest, lucyr, nfcancmxrow, pfcancmxrow, reach_eof, compete, onetile_perPFT)

7.20.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) LUC Annual Read-In Subroutine.

7.20.2 Function/Subroutine Documentation

7.20.2.1 subroutine, public landuse_change::readin_luc (integer, intent(in) *iyear,* integer, intent(in) *nmtest,* integer, intent(in) *nltest,* integer, intent(inout) *lucyr,* real, dimension(nlat,nmos,icc), intent(out) *nfcancmxrow,* real, dimension(nlat,nmos,icc), intent(in) *pfcancmxrow,* logical, intent(inout) *reach_eof,* logical, intent(in) *compete,* logical, intent(in) *onetile_perPFT*)

Parameters

in	*onetile_perpft*	if you are running with one tile per PFT in mosaic mode, set to true. Changes how competition is run. Specifically it allows competition between tiles. This is not recommended for any case where you don't have one PFT in each tile as it has not been tested for that.

it is subsequent years so read in and adjust the luc file info.

If compete is on, then only take in the crop fraction. Set the other fractions to the same as before. These will be adjusted in adjust_luc_fracs.

(re)find the bare fraction for farerow(i,iccp1)

end of the luc file is reached. close and tell main program to exit

7.21 Landuse_change_luc

Canadian Terrestrial Ecosystem Model (CTEM) Land Use Change Subroutine.

- subroutine, public landuse_change::luc (il1,il2,nilg,nol2pfts,grclarea, pfcancmx, nfcancmx,iday,todfrac,yesfrac, interpol, compete,leapnow,

7.21.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Land Use Change Subroutine.

The land use change (LUC) module of CTEM is based on [4] and briefly described here. When the area of crop P↩ FTs changes, CTEM generates LUC emissions. In the simulation where fractional coverage of PFTs is specified, the changes in fractional coverage of crop PFTs are made consistent with changes in the fractional coverage of natural non-crop PFTs. That is, an increase or decrease in the area of crop PFTs is associated with a corresponding decrease or increase in the area of non-crop PFTs. This approach is taken by [53], which allows one to reconstruct historical land cover given a spatial data set of changes in crop area over the historical period and an estimate of potential natural land cover for the pre-industrial period (as described in Sect. methods}). When competition between PFTs for space is allowed, only the fractional coverage of crop PFTs is specified. Similar to a simulation with prescribed PFT fractions, when the area of crop PFTs increases, the fractional coverage of non-crop PF↩ Ts is decreased in proportion to their existing coverage [53]. Alternatively, and in contrast to the simulation with prescribed PFT fractions, when the area of crop PFTs decreases then the generated bare fraction is available for recolonization by non-crop PFTs.

A decrease in the area of natural non-crop PFTs, associated with an increase in area of crop PFTs, results in deforested biomass (while the term *deforested* implies clearing of forests, the same processes can occur in grasslands as well and is meant here to imply removal of the biomass). The deforested biomass is divided into three components: (i) the component that is combusted or used for fuel wood immediately after natural vegetated is deforested and which contributes to atmospheric CO_2, (ii) the component left as slash or used for pulp and paper products and (iii) the component that is used for long-lasting wood products. The fractions allocated to these three components depend on whether the PFTs are woody or herbaceous and on their aboveground vegetation biomass density (see Table 1 of [4]). To account for the timescales involved, the fraction allocated to slash or pulp and paper products is transferred to the model's litter pool and the fraction allocated to long-lasting wood products is allocated to the model's soil carbon pool. Land use change associated with a decrease in the area of natural vegetation thus redistributes carbon from living vegetation to dead litter and soil carbon pools and emits CO_2 to the atmosphere through direct burning of the deforested biomass. The net result is positive LUC carbon emissions from land to the atmosphere.

When croplands are abandoned, the area of natural PFTs increases. In simulations with prescribed fractional coverage of PFTs this results in a decreased carbon density for all model pools as the same amount of carbon is spread over a larger fraction of the grid cell. This reduced density implies that natural vegetation is able to take up carbon as it comes into equilibrium with the driving climate and atmospheric CO_2 concentration. This creates the carbon sink associated with abandonment of croplands as natural vegetation grows in its place. In simulations with competition between PFTs, the abandoned land is treated as bare ground, which is subsequently available for recolonization, as mentioned above. As natural vegetation expands into bare ground it takes up carbon, again creating the carbon sink associated with abandonment of croplands. The net result is negative LUC carbon emissions as carbon is taken from atmosphere to grow vegetation over the area that was previously a cropland.

7.21.2 Function/Subroutine Documentation

7.21.2.1 subroutine, public landuse_change::luc (integer *il1,* integer *il2,* integer *nilg,* integer, dimension(ican) *nol2pfts,* real, dimension(nilg) *grclarea,* real, dimension(nilg,icc) *pfcancmx,* real, dimension(nilg,icc) *nfcancmx,* integer *iday,* real, dimension(nilg,icc) *todfrac,* real, dimension(nilg,icc) *yesfrac,* logical *interpol,* logical *compete,* logical *leapnow*)

Parameters

il1	il1=1
il2	il2=nilg
nilg	no. of grid cells in latitude circle(this is passed in as either ilg or nlat depending on comp/mos)
iday	day of year
nol2pfts	number of level 2 pfts
leapnow	true if this year is a leap year. Only used if the switch 'leap' is true.
interpol	if todfrac & yesfrac are provided then interpol must be set to false so that this subroutine doesn't do its own interpolation using pfcancmx and nfcancmx which are year end values
compete	true if the competition subroutine is on.
pfcancmx	previous max. fractional coverages of ctem's 9 pfts.
nfcancmx	next max. fractional coverages of ctem's 9 pfts.
todfrac	today's fractional coverage of all pfts
yesfrac	yesterday's fractional coverage of all pfts
grclarea	gcm grid cell area, km2

Constants and parameters are located in ctem_params.f90

find/use provided current and previous day's fractional coverage if competition is on, we will adjust these later.

perform interpolation

use provided values but still check they are not -ve

If competition is on, we need to adjust the other fractions for the increase/decrease in cropland as only the crops areas is now specified.

check if this year's fractional coverages have changed or not for any pft

initialization ends

if land use change has taken place then get fcanmxs for use by class based on the new fcancmxs

check if the interpol didn't mess up the barefrac. if so, take the extra amount from the pft with the largest area. jm apr 24 2013. but you can't take it from crops!

find previous day's bare fraction using previous day's fcancmxs

check if the interpol didn't mess up the pbarefra. if so, take the extra amount from the pft with the largest area. jm apr 24 2013. but you can't take it from crops!

based on sizes of 3 live pools and 2 dead pools we estimate the total amount of c in each grid cell.

```
find above ground biomass and treatment index for combust, paper,
and furniture
```

check if a pft's fractional cover is increasing or decreasing

check if bare fraction increases of decreases

if the fractional coverage of pfts increases then spread their live & dead biomass uniformly over the new fraction. this effectively reduces their per m2 c density.

if bare fraction increases then spread its litter and soil c uniformly over the increased fraction

if any of the pfts fractional coverage decreases, then we chop the aboveground biomass and treat it according to our rules (burn it, and convert it into paper and furniture). the below ground live biomass and litter of this pfts gets assimilated into litter of all pfts (uniformly spread over the whole grid cell), and soil c from the chopped off fraction of this pft, gets assimilated into soil c of all existing pfts as well.

chop off above ground biomass

rootmass needs to be chopped as well and all of it goes to the litter/paper pool

keep adding chopped off biomass for each pft to get the total for a grid cell for diagnostics

find what's burnt, and what's converted to paper & furniture

keep adding all this for a given grid cell

litter from the chopped off fraction of the chopped off pft needs to be assimilated, and so does soil c from the chopped off fraction of the chopped pft

if bare fraction decreases then chop off the litter and soil c from the decreased fraction and add it to grsumlit & grsumsoc for spreading over the whole grid cell

calculate if the chopped off biomass equals the sum of grsumcom(i), grsumpap(i) & grsumfur(i)

spread chopped off stuff uniformly over the litter and soil c pools of all existing pfts, including the bare fraction.

convert the available c into density

the combusted c is used to find the c flux that we can release into the atmosphere.

this is flux in kg c/m2.day that will be emitted

convert all land use change fluxes to u-mol co2-c/m2.sec

and finally we see if the total amount of carbon is conserved

total litter mass (before + input from chopped off biomass) and after must be same

for conservation totcmass(i) must be equal to ntotcmas(i) plus combustion carbon losses

update grid averaged vegetation biomass, and litter and soil c densities

just like total amount of carbon must balance, the grid averagred densities must also balance

7.22 Landuse_change_adjust_luc_fracs

this subroutine adjusts the amount of each pft to ensure that the fraction of gridcell bare ground is >0.

7.22.1 Detailed Description

this subroutine adjusts the amount of each pft to ensure that the fraction of gridcell bare ground is >0.

7.23 Landuse_change_adjust_fracs_comp

This subroutine is used when compete = true. It adjusts the amount of each pft to allow expansion of cropland.

7.23.1 Detailed Description

This subroutine is used when compete = true. It adjusts the amount of each pft to allow expansion of cropland.

Chapter 8

Data Type Documentation

8.1 ctem_statevars::class_moyr_output Type Reference

CLASS's monthly outputs.

Public Attributes

- real, dimension(nlat) **alvsacc_mo**
- real, dimension(nlat) **aliracc_mo**
- real, dimension(nlat) **flutacc_mo**
- real, dimension(nlat) **fsinacc_mo**
- real, dimension(nlat) **flinacc_mo**
- real, dimension(nlat) **hfsacc_mo**
- real, dimension(nlat) **qevpacc_mo**
- real, dimension(nlat) **snoacc_mo**
- real, dimension(nlat) **wsnoacc_mo**
- real, dimension(nlat) **rofacc_mo**
- real, dimension(nlat) **preacc_mo**
- real, dimension(nlat) **evapacc_mo**
- real, dimension(nlat) **transpacc_mo**
- real, dimension(nlat) **taacc_mo**
- real, dimension(nlat) **actlyr_mo**
- real, dimension(nlat) **ftable_mo**
- real, dimension(nlat) **actlyr_min_mo**
- real, dimension(nlat) **actlyr_max_mo**
- real, dimension(nlat) **ftable_min_mo**
- real, dimension(nlat) **ftable_max_mo**
- real, dimension(nlat) altotacc_mo

  Broadband albedo.
- real, dimension(nlat) groundevap

  evaporation and sublimation from the ground surface (formed from QFG and QFN), kg /m/mon
- real, dimension(nlat) canopyevap

  evaporation and sublimation from the canopy (formed from QFCL and QFCF), kg /m/mon
- integer, dimension(nlat) altotcntr_m

  Used to count the number of time steps with the sun above the horizon.
- real **fsstar_mo**
- real **flstar_mo**
- real **qh_mo**
- real **qe_mo**

- real, dimension(nlat, ignd) **tbaracc_mo**
- real, dimension(nlat, ignd) **thlqacc_mo**
- real, dimension(nlat, ignd) **thicacc_mo**
- real, dimension(nlat) **alvsacc_yr**
- real, dimension(nlat) **aliracc_yr**
- real, dimension(nlat) **flutacc_yr**
- real, dimension(nlat) **fsinacc_yr**
- real, dimension(nlat) **flinacc_yr**
- real, dimension(nlat) **hfsacc_yr**
- real, dimension(nlat) **qevpacc_yr**
- real, dimension(nlat) **rofacc_yr**
- real, dimension(nlat) **preacc_yr**
- real, dimension(nlat) **evapacc_yr**
- real, dimension(nlat) **transpacc_yr**
- real, dimension(nlat) **taacc_yr**
- real, dimension(nlat) **actlyr_yr**
- real, dimension(nlat) **actlyr_min_yr**
- real, dimension(nlat) **actlyr_max_yr**
- real, dimension(nlat) **ftable_yr**
- real, dimension(nlat) **ftable_min_yr**
- real, dimension(nlat) **ftable_max_yr**
- real, dimension(nlat) altotacc_yr

  Broadband albedo.
- integer, dimension(nlat) altotcntr_yr

  Used to count the number of time steps with the sun above the horizon.
- real **fsstar_yr**
- real **flstar_yr**
- real **qh_yr**
- real **qe_yr**

8.1.1 Detailed Description

CLASS's monthly outputs.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.2 ctem_statevars::ctem_annual Type Reference

CTEM's average annual values (per PFT)

Public Attributes

- real, dimension(nlat, nmos, icc) **laimaxg_yr**
- real, dimension(nlat, nmos, icc) **stemmass_yr**
- real, dimension(nlat, nmos, icc) **rootmass_yr**
- real, dimension(nlat, nmos, icc) **npp_yr**
- real, dimension(nlat, nmos, icc) **gpp_yr**
- real, dimension(nlat, nmos, icc) **vgbiomas_yr**
- real, dimension(nlat, nmos, icc) **autores_yr**
- real, dimension(nlat, nmos, iccp1) **totcmass_yr**

- real, dimension(nlat, nmos, iccp1) **litrmass_yr**
- real, dimension(nlat, nmos, iccp1) **soilcmas_yr**
- real, dimension(nlat, nmos, iccp1) **nep_yr**
- real, dimension(nlat, nmos, iccp1) **litres_yr**
- real, dimension(nlat, nmos, iccp1) **soilcres_yr**
- real, dimension(nlat, nmos, iccp1) **hetrores_yr**
- real, dimension(nlat, nmos, iccp1) **nbp_yr**
- real, dimension(nlat, nmos, icc) **emit_co2_yr**
- real, dimension(nlat, nmos, icc) **emit_co_yr**
- real, dimension(nlat, nmos, icc) **emit_ch4_yr**
- real, dimension(nlat, nmos, icc) **emit_nmhc_yr**
- real, dimension(nlat, nmos, icc) **emit_h2_yr**
- real, dimension(nlat, nmos, icc) **emit_nox_yr**
- real, dimension(nlat, nmos, icc) **emit_n2o_yr**
- real, dimension(nlat, nmos, icc) **emit_pm25_yr**
- real, dimension(nlat, nmos, icc) **emit_tpm_yr**
- real, dimension(nlat, nmos, icc) **emit_tc_yr**
- real, dimension(nlat, nmos, icc) **emit_oc_yr**
- real, dimension(nlat, nmos, icc) **emit_bc_yr**
- real, dimension(nlat, nmos, icc) **bterm_yr**
- real, dimension(nlat, nmos, icc) **mterm_yr**
- real, dimension(nlat, nmos, icc) **burnfrac_yr**
- real, dimension(nlat, nmos, icc) **smfuncveg_yr**
- real, dimension(nlat, nmos, icc) **veghght_yr**

8.2.1 Detailed Description

CTEM's average annual values (per PFT)

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.3 ctem_statevars::ctem_gridavg Type Reference

CTEM's grid average variables.

Public Attributes

- real, dimension(nlat) **wsnorot_g**
- real, dimension(nlat) **rofsrot_g**
- real, dimension(nlat) **snorot_g**
- real, dimension(nlat) **rhosrot_g**
- real, dimension(nlat) **rofrot_g**
- real, dimension(nlat) **zpndrot_g**
- real, dimension(nlat) **rcanrot_g**
- real, dimension(nlat) **scanrot_g**
- real, dimension(nlat) **trofrot_g**
- real, dimension(nlat) **troorot_g**
- real, dimension(nlat) **trobrot_g**
- real, dimension(nlat) **roforot_g**
- real, dimension(nlat) **rofbrot_g**
- real, dimension(nlat) **trosrot_g**

- real, dimension(nlat) **fsgvrot_g**
- real, dimension(nlat) **fsgsrot_g**
- real, dimension(nlat) **flgvrot_g**
- real, dimension(nlat) **flgsrot_g**
- real, dimension(nlat) **hfscrot_g**
- real, dimension(nlat) **hfssrot_g**
- real, dimension(nlat) **hevcrot_g**
- real, dimension(nlat) **hevsrot_g**
- real, dimension(nlat) **hmfcrot_g**
- real, dimension(nlat) **hmfnrot_g**
- real, dimension(nlat) **htcsrot_g**
- real, dimension(nlat) **htccrot_g**
- real, dimension(nlat) **fsggrot_g**
- real, dimension(nlat) **flggrot_g**
- real, dimension(nlat) **hfsgrot_g**
- real, dimension(nlat) **hevgrot_g**
- real, dimension(nlat) **cdhrot_g**
- real, dimension(nlat) **cdmrot_g**
- real, dimension(nlat) **sfcurot_g**
- real, dimension(nlat) **sfcvrot_g**
- real, dimension(nlat) **actlyr_g**
- real, dimension(nlat) **ftable_g**
- real, dimension(nlat) **fc_g**
- real, dimension(nlat) **fg_g**
- real, dimension(nlat) **fcs_g**
- real, dimension(nlat) **fgs_g**
- real, dimension(nlat) **pcfcrot_g**
- real, dimension(nlat) **pclcrot_g**
- real, dimension(nlat) **pcpgrot_g**
- real, dimension(nlat) **qfcfrot_g**
- real, dimension(nlat) **qfgrot_g**
- real, dimension(nlat, ignd) **qfcrot_g**
- real, dimension(nlat) **rofcrot_g**
- real, dimension(nlat) **rofnrot_g**
- real, dimension(nlat) **wtrsrot_g**
- real, dimension(nlat) **wtrgrot_g**
- real, dimension(nlat) **pcpnrot_g**
- real, dimension(nlat) **qfclrot_g**
- real, dimension(nlat) **qfnrot_g**
- real, dimension(nlat) **wtrcrot_g**
- real, dimension(nlat) **gpp_g**
- real, dimension(nlat) **npp_g**
- real, dimension(nlat) **nbp_g**
- real, dimension(nlat) **socres_g**
- real, dimension(nlat) **autores_g**
- real, dimension(nlat) **litres_g**
- real, dimension(nlat) **dstcemls3_g**
- real, dimension(nlat) **litrfall_g**
- real, dimension(nlat) **rml_g**
- real, dimension(nlat) **rms_g**
- real, dimension(nlat) **rg_g**
- real, dimension(nlat) **leaflitr_g**
- real, dimension(nlat) **tltrstem_g**
- real, dimension(nlat) **tltrroot_g**
- real, dimension(nlat) **nep_g**

- real, dimension(nlat) **hetrores_g**
- real, dimension(nlat) **dstcemls_g**
- real, dimension(nlat) **humiftrs_g**
- real, dimension(nlat) **rmr_g**
- real, dimension(nlat) **tltrleaf_g**
- real, dimension(nlat) **gavgltms_g**
- real, dimension(nlat) **vgbiomas_g**
- real, dimension(nlat) **gavglai_g**
- real, dimension(nlat) **gavgscms_g**
- real, dimension(nlat) **gleafmas_g**
- real, dimension(nlat) **bleafmas_g**
- real, dimension(nlat) **stemmass_g**
- real, dimension(nlat) **rootmass_g**
- real, dimension(nlat) **litrmass_g**
- real, dimension(nlat) **soilcmas_g**
- real, dimension(nlat) **slai_g**
- real, dimension(nlat) **ailcg_g**
- real, dimension(nlat) **ailcb_g**
- real, dimension(nlat) **veghght_g**
- real, dimension(nlat) **rootdpth_g**
- real, dimension(nlat) **roottemp_g**
- real, dimension(nlat) **totcmass_g**
- real, dimension(nlat) **tcanoacc_out_g**
- real, dimension(nlat) **burnfrac_g**
- real, dimension(nlat) **smfuncveg_g**
- real, dimension(nlat) **lucemcom_g**
- real, dimension(nlat) **lucltrin_g**
- real, dimension(nlat) **lucsocin_g**
- real, dimension(nlat) **emit_co2_g**
- real, dimension(nlat) **emit_co_g**
- real, dimension(nlat) **emit_ch4_g**
- real, dimension(nlat) **emit_nmhc_g**
- real, dimension(nlat) **emit_h2_g**
- real, dimension(nlat) **emit_nox_g**
- real, dimension(nlat) **emit_n2o_g**
- real, dimension(nlat) **emit_pm25_g**
- real, dimension(nlat) **emit_tpm_g**
- real, dimension(nlat) **emit_tc_g**
- real, dimension(nlat) **emit_oc_g**
- real, dimension(nlat) **emit_bc_g**
- real, dimension(nlat) **bterm_g**
- real, dimension(nlat) **lterm_g**
- real, dimension(nlat) **mterm_g**
- real, dimension(nlat) **ch4wet1_g**
- real, dimension(nlat) **ch4wet2_g**
- real, dimension(nlat) **wetfdyn_g**
- real, dimension(nlat) **ch4dyn1_g**
- real, dimension(nlat) **ch4dyn2_g**
- real, dimension(nlat) **ch4_soills_g**
- real, dimension(nlat, icc) **afrleaf_g**
- real, dimension(nlat, icc) **afrstem_g**
- real, dimension(nlat, icc) **afrroot_g**
- real, dimension(nlat, icc) **lfstatus_g**
- real, dimension(nlat, icc) **rmlvegrow_g**
- real, dimension(nlat, icc) **anvegrow_g**

- real, dimension(nlat, ignd) **rmatctem_g**
- real, dimension(nlat, ignd) **hmfgrot_g**
- real, dimension(nlat, ignd) **htcrot_g**
- real, dimension(nlat, ignd) **tbarrot_g**
- real, dimension(nlat, ignd) **thlqrot_g**
- real, dimension(nlat, ignd) **thicrot_g**
- real, dimension(nlat, ignd) **gflxrot_g**
- real, dimension(nlat) **fsstar_g**
- real, dimension(nlat) **flstar_g**
- real, dimension(nlat) **qh_g**
- real, dimension(nlat) **qe_g**
- real, dimension(nlat) **snomlt_g**
- real, dimension(nlat) **beg_g**
- real, dimension(nlat) **gtout_g**
- real, dimension(nlat) **tpn_g**
- real, dimension(nlat) **altot_g**
- real, dimension(nlat) **tcn_g**
- real, dimension(nlat) **tsn_g**
- real, dimension(nlat) **zsn_g**

8.3.1 Detailed Description

CTEM's grid average variables.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.4 ctem_statevars::ctem_gridavg_annual Type Reference

CTEM's grid average annual values.

Public Attributes

- real, dimension(nlat) **laimaxg_yr_g**
- real, dimension(nlat) **stemmass_yr_g**
- real, dimension(nlat) **rootmass_yr_g**
- real, dimension(nlat) **litrmass_yr_g**
- real, dimension(nlat) **soilcmas_yr_g**
- real, dimension(nlat) **npp_yr_g**
- real, dimension(nlat) **gpp_yr_g**
- real, dimension(nlat) **nep_yr_g**
- real, dimension(nlat) **nbp_yr_g**
- real, dimension(nlat) **hetrores_yr_g**
- real, dimension(nlat) **autores_yr_g**
- real, dimension(nlat) **litres_yr_g**
- real, dimension(nlat) **soilcres_yr_g**
- real, dimension(nlat) **vgbiomas_yr_g**
- real, dimension(nlat) **totcmass_yr_g**
- real, dimension(nlat) **emit_co2_yr_g**
- real, dimension(nlat) **emit_co_yr_g**
- real, dimension(nlat) **emit_ch4_yr_g**
- real, dimension(nlat) **emit_nmhc_yr_g**

- real, dimension(nlat) **emit_h2_yr_g**
- real, dimension(nlat) **emit_nox_yr_g**
- real, dimension(nlat) **emit_n2o_yr_g**
- real, dimension(nlat) **emit_pm25_yr_g**
- real, dimension(nlat) **emit_tpm_yr_g**
- real, dimension(nlat) **emit_tc_yr_g**
- real, dimension(nlat) **emit_oc_yr_g**
- real, dimension(nlat) **emit_bc_yr_g**
- real, dimension(nlat) **smfuncveg_yr_g**
- real, dimension(nlat) **luc_emc_yr_g**
- real, dimension(nlat) **lucltrin_yr_g**
- real, dimension(nlat) **lucsocin_yr_g**
- real, dimension(nlat) **burnfrac_yr_g**
- real, dimension(nlat) **bterm_yr_g**
- real, dimension(nlat) **lterm_yr_g**
- real, dimension(nlat) **mterm_yr_g**
- real, dimension(nlat) **ch4wet1_yr_g**
- real, dimension(nlat) **ch4wet2_yr_g**
- real, dimension(nlat) **wetfdyn_yr_g**
- real, dimension(nlat) **ch4dyn1_yr_g**
- real, dimension(nlat) **ch4dyn2_yr_g**
- real, dimension(nlat) **ch4soills_yr_g**
- real, dimension(nlat) **veghght_yr_g**

8.4.1 Detailed Description

CTEM's grid average annual values.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.5 ctem_statevars::ctem_gridavg_monthly Type Reference

CTEM's grid average monthly values.

Public Attributes

- real, dimension(nlat) **laimaxg_mo_g**
- real, dimension(nlat) **stemmass_mo_g**
- real, dimension(nlat) **rootmass_mo_g**
- real, dimension(nlat) **litrmass_mo_g**
- real, dimension(nlat) **soilcmas_mo_g**
- real, dimension(nlat) **litrfall_mo_g**
- real, dimension(nlat) **humiftrs_mo_g**
- real, dimension(nlat) **npp_mo_g**
- real, dimension(nlat) **gpp_mo_g**
- real, dimension(nlat) **nep_mo_g**
- real, dimension(nlat) **nbp_mo_g**
- real, dimension(nlat) **hetrores_mo_g**
- real, dimension(nlat) **autores_mo_g**
- real, dimension(nlat) **litres_mo_g**
- real, dimension(nlat) **soilcres_mo_g**

- real, dimension(nlat) **vgbiomas_mo_g**
- real, dimension(nlat) **totcmass_mo_g**
- real, dimension(nlat) **emit_co2_mo_g**
- real, dimension(nlat) **emit_co_mo_g**
- real, dimension(nlat) **emit_ch4_mo_g**
- real, dimension(nlat) **emit_nmhc_mo_g**
- real, dimension(nlat) **emit_h2_mo_g**
- real, dimension(nlat) **emit_nox_mo_g**
- real, dimension(nlat) **emit_n2o_mo_g**
- real, dimension(nlat) **emit_pm25_mo_g**
- real, dimension(nlat) **emit_tpm_mo_g**
- real, dimension(nlat) **emit_tc_mo_g**
- real, dimension(nlat) **emit_oc_mo_g**
- real, dimension(nlat) **emit_bc_mo_g**
- real, dimension(nlat) **smfuncveg_mo_g**
- real, dimension(nlat) **luc_emc_mo_g**
- real, dimension(nlat) **lucltrin_mo_g**
- real, dimension(nlat) **lucsocin_mo_g**
- real, dimension(nlat) **burnfrac_mo_g**
- real, dimension(nlat) **bterm_mo_g**
- real, dimension(nlat) **lterm_mo_g**
- real, dimension(nlat) **mterm_mo_g**
- real, dimension(nlat) **ch4wet1_mo_g**
- real, dimension(nlat) **ch4wet2_mo_g**
- real, dimension(nlat) **wetfdyn_mo_g**
- real, dimension(nlat) **ch4dyn1_mo_g**
- real, dimension(nlat) **ch4dyn2_mo_g**
- real, dimension(nlat) **ch4soills_mo_g**

8.5.1 Detailed Description

CTEM's grid average monthly values.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.6 ctem_statevars::ctem_monthly Type Reference

CTEM's variables monthly averaged (per pft)

Public Attributes

- real, dimension(nlat, nmos, icc) **laimaxg_mo**
- real, dimension(nlat, nmos, icc) **stemmass_mo**
- real, dimension(nlat, nmos, icc) **rootmass_mo**
- real, dimension(nlat, nmos, icc) **litrfallveg_mo**
- real, dimension(nlat, nmos, iccp1) **humiftrsveg_mo**
- real, dimension(nlat, nmos, icc) **npp_mo**
- real, dimension(nlat, nmos, icc) **gpp_mo**
- real, dimension(nlat, nmos, icc) **vgbiomas_mo**
- real, dimension(nlat, nmos, icc) **autores_mo**
- real, dimension(nlat, nmos, iccp1) **totcmass_mo**

- real, dimension(nlat, nmos, iccp1) **litrmass_mo**
- real, dimension(nlat, nmos, iccp1) **soilcmas_mo**
- real, dimension(nlat, nmos, iccp1) **nep_mo**
- real, dimension(nlat, nmos, iccp1) **litres_mo**
- real, dimension(nlat, nmos, iccp1) **soilcres_mo**
- real, dimension(nlat, nmos, iccp1) **hetrores_mo**
- real, dimension(nlat, nmos, iccp1) **nbp_mo**
- real, dimension(nlat, nmos, icc) **emit_co2_mo**
- real, dimension(nlat, nmos, icc) **emit_co_mo**
- real, dimension(nlat, nmos, icc) **emit_ch4_mo**
- real, dimension(nlat, nmos, icc) **emit_nmhc_mo**
- real, dimension(nlat, nmos, icc) **emit_h2_mo**
- real, dimension(nlat, nmos, icc) **emit_nox_mo**
- real, dimension(nlat, nmos, icc) **emit_n2o_mo**
- real, dimension(nlat, nmos, icc) **emit_pm25_mo**
- real, dimension(nlat, nmos, icc) **emit_tpm_mo**
- real, dimension(nlat, nmos, icc) **emit_tc_mo**
- real, dimension(nlat, nmos, icc) **emit_oc_mo**
- real, dimension(nlat, nmos, icc) **emit_bc_mo**
- real, dimension(nlat, nmos, icc) **burnfrac_mo**
- real, dimension(nlat, nmos, icc) **bterm_mo**
- real, dimension(nlat, nmos, icc) **mterm_mo**
- real, dimension(nlat, nmos, icc) **smfuncveg_mo**

8.6.1 Detailed Description

CTEM's variables monthly averaged (per pft)

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.7 ctem_statevars::ctem_switches Type Reference

switches for running CTEM, read from the joboptions file

Public Attributes

- logical **ctem_on**
- logical parallelrun

  set this to be true if model is run in parallel mode for multiple grid cells, output is limited to monthly & yearly grid-mean only. else the run is in stand alone mode, in which output includes half-hourly and daily and mosaic-mean as well.
- logical cyclemet

  to cycle over only a fixed number of years (nummetcylyrs) starting at a certain year (metcylyrst) if cyclemet, then put co2on = false and set an appopriate setco2conc, also if popdon is true, it will choose the popn and luc data for year metcylyrst and cycle on that.
- logical dofire

  boolean, if true allow fire, if false no fire.
- logical **run_model**
- logical **met_rewound**
- logical **reach_eof**
- logical compete

logical boolean telling if competition between pfts is on or not

- logical start_bare

  set this to true if competition is true, and if you wish to start from bare ground. if this is set to false, the ini and ctm file info will be used to set up the run. NOTE: This still keeps the crop fractions (while setting all pools to zero)

- logical rsfile

  set this to true if restart files (.ini_rs and .ctm_rs) are written at the end of each year. these files are necessary for checking whether the model reaches equilibrium after running for a certain years. set this to false if restart files are not needed (known how many years the model will run)

- logical lnduseon

  logical switch to run the land use change subroutine or not.

- logical co2on

  use co_2 time series, set to false if cyclemet is true

- logical ch4on

  use CH_4 time series, set to false if cyclemet is true the CO_2 timeseries is in the same input file as the CO_2 one.

- logical popdon

  if set true use population density data to calculate fire extinguishing probability and probability of fire due to human causes, or if false, read directly from .ctm file

- logical inibioclim

  switch telling if bioclimatic parameters are being initialized from scratch (false) or being initialized from some spun up values(true).

- logical start_from_rs

  if true, this option copies the _RS INI and CTM files to be the .INI and .CTM files and then starts the run as per normal. it is handy when spinning up so you don't have to do a complicated copying of the RS files to restart from them. NOTE! This will not work on hadar or spica, instead you have to manually move the files and set this to .false.

- logical leap

  set to true if all/some leap years in the .MET file have data for 366 days also accounts for leap years in .MET when cycling over meteorology (cyclemet)

- logical dowetlands

  if true allow wetland methane emission

- logical obswetf

  observed wetland fraction

- logical **transient_run**
- character(80) **titlec1**
- character(80) **titlec2**
- character(80) **titlec3**

8.7.1 Detailed Description

switches for running CTEM, read from the joboptions file

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.8 ctem_statevars::ctem_tile_level Type Reference

CTEM's variables per tile.

Public Attributes

- real, dimension(nlat, nmos) **leaflitr_t**
- real, dimension(nlat, nmos) **tltrleaf_t**
- real, dimension(nlat, nmos) **tltrstem_t**
- real, dimension(nlat, nmos) **tltrroot_t**
- real, dimension(nlat, nmos) **ailcg_t**
- real, dimension(nlat, nmos) **ailcb_t**
- real, dimension(nlat, nmos, ignd) **rmatctem_t**
- real, dimension(nlat, nmos) **veghght_t**
- real, dimension(nlat, nmos) **rootdpth_t**
- real, dimension(nlat, nmos) **roottemp_t**
- real, dimension(nlat, nmos) **slai_t**
- real, dimension(nlat, nmos) **afrroot_t**
- real, dimension(nlat, nmos) **afrleaf_t**
- real, dimension(nlat, nmos) **afrstem_t**
- real, dimension(nlat, nmos) **laimaxg_t**
- real, dimension(nlat, nmos) **stemmass_t**
- real, dimension(nlat, nmos) **rootmass_t**
- real, dimension(nlat, nmos) **litrmass_t**
- real, dimension(nlat, nmos) **gleafmas_t**
- real, dimension(nlat, nmos) **bleafmas_t**
- real, dimension(nlat, nmos) **soilcmas_t**
- real, dimension(nlat, nmos) **emit_co2_t**
- real, dimension(nlat, nmos) **emit_co_t**
- real, dimension(nlat, nmos) **emit_ch4_t**
- real, dimension(nlat, nmos) **emit_nmhc_t**
- real, dimension(nlat, nmos) **emit_h2_t**
- real, dimension(nlat, nmos) **emit_nox_t**
- real, dimension(nlat, nmos) **emit_n2o_t**
- real, dimension(nlat, nmos) **emit_pm25_t**
- real, dimension(nlat, nmos) **emit_tpm_t**
- real, dimension(nlat, nmos) **emit_tc_t**
- real, dimension(nlat, nmos) **emit_oc_t**
- real, dimension(nlat, nmos) **emit_bc_t**
- real, dimension(nlat, nmos) **bterm_t**
- real, dimension(nlat, nmos) **mterm_t**
- real, dimension(nlat, nmos) **smfuncveg_t**
- real, dimension(ilg) **fsnowacc_t**
- real, dimension(ilg) **tcansacc_t**
- real, dimension(ilg) **tcanoaccgat_t**
- real, dimension(ilg) **taaccgat_t**
- real, dimension(ilg) **uvaccgat_t**
- real, dimension(ilg) **vvaccgat_t**
- real, dimension(ilg, ignd) **tbaraccgat_t**
- real, dimension(ilg, ignd) **tbarcacc_t**
- real, dimension(ilg, ignd) **tbarcsacc_t**
- real, dimension(ilg, ignd) **tbargacc_t**
- real, dimension(ilg, ignd) **tbargsacc_t**
- real, dimension(ilg, ignd) **thliqcacc_t**
- real, dimension(ilg, ignd) **thliqgacc_t**
- real, dimension(ilg, ignd) **thliqacc_t**
- real, dimension(ilg, ignd) **thicecacc_t**
- real, dimension(ilg, ignd) **thicegacc_t**
- real, dimension(ilg, icc) **ancsvgac_t**
- real, dimension(ilg, icc) **ancgvgac_t**
- real, dimension(ilg, icc) **rmlcsvga_t**
- real, dimension(ilg, icc) **rmlcgvga_t**

8.8.1 Detailed Description

CTEM's variables per tile.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.9 ctem_statevars::ctem_tileavg_annual Type Reference

CTEM's variables per tile annual values.

Public Attributes

- real, dimension(nlat, nmos) **laimaxg_yr_t**
- real, dimension(nlat, nmos) **stemmass_yr_t**
- real, dimension(nlat, nmos) **rootmass_yr_t**
- real, dimension(nlat, nmos) **npp_yr_t**
- real, dimension(nlat, nmos) **gpp_yr_t**
- real, dimension(nlat, nmos) **vgbiomas_yr_t**
- real, dimension(nlat, nmos) **autores_yr_t**
- real, dimension(nlat, nmos) **totcmass_yr_t**
- real, dimension(nlat, nmos) **litrmass_yr_t**
- real, dimension(nlat, nmos) **soilcmas_yr_t**
- real, dimension(nlat, nmos) **nep_yr_t**
- real, dimension(nlat, nmos) **litres_yr_t**
- real, dimension(nlat, nmos) **soilcres_yr_t**
- real, dimension(nlat, nmos) **hetrores_yr_t**
- real, dimension(nlat, nmos) **nbp_yr_t**
- real, dimension(nlat, nmos) **emit_co2_yr_t**
- real, dimension(nlat, nmos) **emit_co_yr_t**
- real, dimension(nlat, nmos) **emit_ch4_yr_t**
- real, dimension(nlat, nmos) **emit_nmhc_yr_t**
- real, dimension(nlat, nmos) **emit_h2_yr_t**
- real, dimension(nlat, nmos) **emit_nox_yr_t**
- real, dimension(nlat, nmos) **emit_n2o_yr_t**
- real, dimension(nlat, nmos) **emit_pm25_yr_t**
- real, dimension(nlat, nmos) **emit_tpm_yr_t**
- real, dimension(nlat, nmos) **emit_tc_yr_t**
- real, dimension(nlat, nmos) **emit_oc_yr_t**
- real, dimension(nlat, nmos) **emit_bc_yr_t**
- real, dimension(nlat, nmos) **burnfrac_yr_t**
- real, dimension(nlat, nmos) **smfuncveg_yr_t**
- real, dimension(nlat, nmos) **bterm_yr_t**
- real, dimension(nlat, nmos) **luc_emc_yr_t**
- real, dimension(nlat, nmos) **lterm_yr_t**
- real, dimension(nlat, nmos) **lucsocin_yr_t**
- real, dimension(nlat, nmos) **mterm_yr_t**
- real, dimension(nlat, nmos) **lucltrin_yr_t**
- real, dimension(nlat, nmos) **ch4wet1_yr_t**
- real, dimension(nlat, nmos) **ch4wet2_yr_t**
- real, dimension(nlat, nmos) **wetfdyn_yr_t**
- real, dimension(nlat, nmos) **ch4dyn1_yr_t**
- real, dimension(nlat, nmos) **ch4dyn2_yr_t**
- real, dimension(nlat, nmos) **ch4soills_yr_t**
- real, dimension(nlat, nmos) **veghght_yr_t**

8.9.1 Detailed Description

CTEM's variables per tile annual values.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.10 ctem_statevars::ctem_tileavg_monthly Type Reference

CTEM's variables per tile monthly values.

Public Attributes

- real, dimension(nlat, nmos) **laimaxg_mo_t**
- real, dimension(nlat, nmos) **stemmass_mo_t**
- real, dimension(nlat, nmos) **rootmass_mo_t**
- real, dimension(nlat, nmos) **litrfall_mo_t**
- real, dimension(nlat, nmos) **humiftrs_mo_t**
- real, dimension(nlat, nmos) **npp_mo_t**
- real, dimension(nlat, nmos) **gpp_mo_t**
- real, dimension(nlat, nmos) **vgbiomas_mo_t**
- real, dimension(nlat, nmos) **autores_mo_t**
- real, dimension(nlat, nmos) **totcmass_mo_t**
- real, dimension(nlat, nmos) **litrmass_mo_t**
- real, dimension(nlat, nmos) **soilcmas_mo_t**
- real, dimension(nlat, nmos) **nep_mo_t**
- real, dimension(nlat, nmos) **litres_mo_t**
- real, dimension(nlat, nmos) **soilcres_mo_t**
- real, dimension(nlat, nmos) **hetrores_mo_t**
- real, dimension(nlat, nmos) **nbp_mo_t**
- real, dimension(nlat, nmos) **emit_co2_mo_t**
- real, dimension(nlat, nmos) **emit_co_mo_t**
- real, dimension(nlat, nmos) **emit_ch4_mo_t**
- real, dimension(nlat, nmos) **emit_nmhc_mo_t**
- real, dimension(nlat, nmos) **emit_h2_mo_t**
- real, dimension(nlat, nmos) **emit_nox_mo_t**
- real, dimension(nlat, nmos) **emit_n2o_mo_t**
- real, dimension(nlat, nmos) **emit_pm25_mo_t**
- real, dimension(nlat, nmos) **emit_tpm_mo_t**
- real, dimension(nlat, nmos) **emit_tc_mo_t**
- real, dimension(nlat, nmos) **emit_oc_mo_t**
- real, dimension(nlat, nmos) **emit_bc_mo_t**
- real, dimension(nlat, nmos) **burnfrac_mo_t**
- real, dimension(nlat, nmos) **smfuncveg_mo_t**
- real, dimension(nlat, nmos) **bterm_mo_t**
- real, dimension(nlat, nmos) **luc_emc_mo_t**
- real, dimension(nlat, nmos) **lterm_mo_t**
- real, dimension(nlat, nmos) **lucsocin_mo_t**
- real, dimension(nlat, nmos) **mterm_mo_t**
- real, dimension(nlat, nmos) **lucltrin_mo_t**
- real, dimension(nlat, nmos) **ch4wet1_mo_t**
- real, dimension(nlat, nmos) **ch4wet2_mo_t**

- real, dimension(nlat, nmos) **wetfdyn_mo_t**
- real, dimension(nlat, nmos) **ch4dyn1_mo_t**
- real, dimension(nlat, nmos) **ch4dyn2_mo_t**
- real, dimension(nlat, nmos) **ch4soills_mo_t**
- real, dimension(nlat, nmos) **wind_mo_t**

8.10.1 Detailed Description

CTEM's variables per tile monthly values.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.11 ctem_statevars::veg_gat Type Reference

CTEM's 'gat' vars.

Public Attributes

- real, dimension(ilg, icc) **ailcmin**
- real, dimension(ilg, icc) **ailcmax**
- real, dimension(ilg, icc) **dvdfcan**
- real, dimension(ilg, icc) gleafmas

  green leaf mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(ilg, icc) bleafmas

  brown leaf mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(ilg, icc) stemmass

  stem mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(ilg, icc) rootmass

  root mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(ilg, icc) pstemmass

  stem mass from previous timestep, is value before fire. used by burntobare subroutine
- real, dimension(ilg, icc) pgleafmass

  root mass from previous timestep, is value before fire. used by burntobare subroutine
- real, dimension(ilg, icc) fcancmx

  max. fractional coverage of ctem's 9 pfts, but this can be modified by land-use change, and competition between pfts
- real, dimension(ilg) gavglai

  grid averaged green leaf area index
- real, dimension(ilg) lightng

  total lightning frequency, flashes/km2.year
- real, dimension(ilg) **tcanoaccgat_out**
- real, dimension(ilg, ican) zolnc

  lumped log of roughness length for class' 4 pfts
- real, dimension(ilg, ican) ailc

  lumped lai for class' 4 pfts
- real, dimension(ilg, icc) ailcg

  green lai for ctem's 9 pfts
- real, dimension(ilg, icc) ailcgs

  GREEN LAI FOR CANOPY OVER SNOW SUB-AREA.
- real, dimension(ilg, icc) fcancs

FRACTION OF CANOPY OVER SNOW FOR CTEM's 9 PFTs.

- real, dimension(ilg, icc) fcanc

  FRACTIONAL COVERAGE OF 8 CARBON PFTs, CANOPY OVER GROUND.
- real, dimension(ilg) co2conc

  ATMOS. CO2 CONC. IN PPM.
- real, dimension(ilg) **ch4conc**
- real, dimension(ilg, icc) co2i1cg

  INTERCELLULAR CO2 CONC FOR 8 PFTs FOR CANOPY OVER GROUND SUBAREA (Pa) - FOR SINGLE/SUNLIT LEAF.
- real, dimension(ilg, icc) co2i1cs

  SAME AS ABOVE BUT FOR SHADED LEAF (above being co2i1cg)
- real, dimension(ilg, icc) co2i2cg

  INTERCELLULAR CO2 CONC FOR 8 PFTs FOR CANOPY OVER SNOWSUBAREA (Pa) - FOR SINGLE/SUNLIT LEAF.
- real, dimension(ilg, icc) co2i2cs

  SAME AS ABOVE BUT FOR SHADED LEAF (above being co2i2cg)
- real, dimension(ilg, icc) ancsveg

  net photosynthetic rate for ctems 9 pfts for canopy over snow subarea
- real, dimension(ilg, icc) ancgveg

  net photosynthetic rate for ctems 9 pfts for canopy over ground subarea
- real, dimension(ilg, icc) rmlcsveg

  leaf respiration rate for ctems 9 pfts forcanopy over snow subarea
- real, dimension(ilg, icc) rmlcgveg

  leaf respiration rate for ctems 9 pfts forcanopy over ground subarea
- real, dimension(ilg, icc) slai

  storage/imaginary lai for phenology purposes
- real, dimension(ilg, icc) ailcb

  brown lai for ctem's 9 pfts. for now we assume only grasses can have brown lai
- real, dimension(ilg) **canres**
- real, dimension(ilg, icc) flhrloss

  fall or harvest loss for deciduous trees and crops, respectively, kgc/m^2 *il1*
- real, dimension(ilg, icc) grwtheff

  growth efficiency. change in biomass per year per unit max. lai (kgc/m^2 *)/(m2/m2), for use in mortality subroutine*
- real, dimension(ilg, icc) lystmmas

  stem mass at the end of last year
- real, dimension(ilg, icc) lyrotmas

  root mass at the end of last year
- real, dimension(ilg, icc) tymaxlai

  this year's maximum lai
- real, dimension(ilg) vgbiomas

  grid averaged vegetation biomass, kgc/m^2
- real, dimension(ilg) gavgltms

  grid averaged litter mass, kgc/m^2
- real, dimension(ilg) gavgscms

  grid averaged soil c mass, kgc/m^2
- real, dimension(ilg, icc) stmhrlos

  stem harvest loss for crops, kgc/m^2
- real, dimension(ilg, ican, ignd) rmatc

  fraction of roots for each of class' 4 pfts in each soil layer
- real, dimension(ilg, icc, ignd) rmatctem

  fraction of roots for each of ctem's 9 pfts in each soil layer

- real, dimension(ilg, iccp1) litrmass

  litter mass for each of the 9 ctem pfts + bare, kgc/m^2
- real, dimension(ilg, iccp1) soilcmas

  soil carbon mass for each of the 9 ctem pfts + bare, kgc/m^2
- real, dimension(ilg, icc) vgbiomas_veg

  vegetation biomass for each pft
- real, dimension(ilg, icc) emit_co2

  carbon dioxide
- real, dimension(ilg, icc) emit_co

  carbon monoxide
- real, dimension(ilg, icc) emit_ch4

  methane
- real, dimension(ilg, icc) emit_nmhc

  non-methane hydrocarbons
- real, dimension(ilg, icc) emit_h2

  hydrogen gas
- real, dimension(ilg, icc) emit_nox

  nitrogen oxides
- real, dimension(ilg, icc) emit_n2o

  nitrous oxide
- real, dimension(ilg, icc) emit_pm25

  particulate matter less than 2.5 um in diameter
- real, dimension(ilg, icc) emit_tpm

  total particulate matter
- real, dimension(ilg, icc) emit_tc

  total carbon
- real, dimension(ilg, icc) emit_oc

  organic carbon
- real, dimension(ilg, icc) emit_bc

  black carbon
- real, dimension(ilg) burnfrac

  areal fraction burned due to fire for every grid cell (%)
- real, dimension(ilg, icc) burnvegf

  per PFT fraction burned of that PFT's area
- real, dimension(ilg, icc) **smfuncveg**
- real, dimension(ilg) popdin

  population density (people / km^2)
- real, dimension(ilg, icc) bterm

  biomass term for fire probabilty calc
- real, dimension(ilg) lterm

  lightning term for fire probabilty calc
- real, dimension(ilg, icc) mterm

  moisture term for fire probabilty calc
- real, dimension(ilg) extnprob

  fire extingusinging probability
- real, dimension(ilg) prbfrhuc

  probability of fire due to human causes
- real, dimension(ilg, 12) **mlightng**
- real, dimension(ilg) dayl_max

  maximum daylength for that location (hours)
- real, dimension(ilg) dayl

daylength for that location (hours)

- real, dimension(ilg, icc) bmasveg

  total (gleaf + stem + root) biomass for each ctem pft, kgc/m^2
- real, dimension(ilg, ican) cmasvegc

  total canopy mass for each of the 4 class pfts. recall that class requires canopy mass as an input, and this is now provided by ctem. kg/m^2.
- real, dimension(ilg, icc) veghght

  vegetation height (meters)
- real, dimension(ilg, icc) rootdpth

  99% soil rooting depth (meters) both veghght & rootdpth can be used as diagnostics to see how vegetation grows above and below ground, respectively
- real, dimension(ilg) rml

  leaf maintenance respiration (u-mol co2/m2.sec)
- real, dimension(ilg) rms

  stem maintenance respiration (u-mol co2/m2.sec)
- real, dimension(ilg, icc) tltrleaf

  total leaf litter fall rate (u-mol co2/m2.sec)
- real, dimension(ilg, icc) tltrstem

  total stem litter fall rate (u-mol co2/m2.sec)
- real, dimension(ilg, icc) tltrroot

  total root litter fall rate (u-mol co2/m2.sec)
- real, dimension(ilg, icc) leaflitr

  leaf litter fall rate (u-mol co2/m2.sec). this leaf litter does not include litter generated due to mortality/fire
- real, dimension(ilg, icc) roottemp

  root temperature, k
- real, dimension(ilg, icc) afrleaf

  allocation fraction for leaves
- real, dimension(ilg, icc) afrstem

  allocation fraction for stem
- real, dimension(ilg, icc) afrroot

  allocation fraction for root
- real, dimension(ilg, icc) wtstatus

  soil water status used for calculating allocation fractions
- real, dimension(ilg, icc) ltstatus

  light status used for calculating allocation fractions
- real, dimension(ilg) rmr

  root maintenance respiration (u-mol co2/m2.sec)
- real, dimension(ilg, 8) slopefrac

  prescribed fraction of wetlands based on slope only(0.025, 0.05, 0.1, 0.15, 0.20, 0.25, 0.3 and 0.35 percent slope thresholds)
- real, dimension(ilg) **wetfrac_pres**
- real, dimension(ilg, 12) **wetfrac_mon**
- real, dimension(ilg) ch4wet1

  methane flux from wetlands calculated using hetrores in umol ch4/m2.s
- real, dimension(ilg) ch4wet2

  methane flux from wetlands calculated using npp in umol ch4/m2.s
- real, dimension(ilg) wetfdyn

  dynamic wetland fraction
- real, dimension(ilg) ch4dyn1

  methane flux from wetlands calculated using hetrores and wetfdyn, in umol ch4/m2.s
- real, dimension(ilg) ch4dyn2

methane flux from wetlands calculated using npp and wetfdyn, in umol ch4/m2.s

- real, dimension(ilg) ch4_soills

  Methane uptake into the soil column ($mg CH_4 m^{-2} s^{-1}$)
- real, dimension(ilg) lucemcom

  land use change (luc) related combustion emission losses, u-mol co2/m2.sec
- real, dimension(ilg) lucltrin

  luc related inputs to litter pool, u-mol co2/m2.sec
- real, dimension(ilg) lucsocin

  luc related inputs to soil c pool, u-mol co2/m2.sec
- real, dimension(ilg) npp

  net primary productivity
- real, dimension(ilg) nep

  net ecosystem productivity
- real, dimension(ilg) nbp

  net biome productivity
- real, dimension(ilg) gpp

  gross primary productivity
- real, dimension(ilg) hetrores

  heterotrophic respiration
- real, dimension(ilg) autores

  autotrophic respiration
- real, dimension(ilg) **soilcresp**
- real, dimension(ilg) rm

  maintenance respiration
- real, dimension(ilg) rg

  growth respiration
- real, dimension(ilg) litres

  litter respiration
- real, dimension(ilg) socres

  soil carbon respiration
- real, dimension(ilg) dstcemls

  carbon emission losses due to disturbance, mainly fire
- real, dimension(ilg) litrfall

  total litter fall (from leaves, stem, and root) due to all causes (mortality, turnover, and disturbance)
- real, dimension(ilg) humiftrs

  transfer of humidified litter from litter to soil c pool
- real, dimension(ilg, icc) gppveg

  gross primary productity for each pft
- real, dimension(ilg, iccp1) nepveg

  net ecosystem productity for bare fraction expnbaln(i)=0.0 amount of c related to spatial expansion Not used JM Jun 2014 OR net ecosystem productity for each pft
- real, dimension(ilg, iccp1) nbpveg

  net biome productity for bare fraction OR net biome productity for each pft
- real, dimension(ilg, icc) nppveg

  npp for individual pfts, u-mol co2/m2.sec
- real, dimension(ilg, iccp1) **hetroresveg**
- real, dimension(ilg, icc) **autoresveg**
- real, dimension(ilg, iccp1) **litresveg**
- real, dimension(ilg, iccp1) **soilcresveg**
- real, dimension(ilg, icc) **rmlvegacc**
- real, dimension(ilg, icc) rmsveg

stem maintenance resp. rate for each pft

- real, dimension(ilg, icc) rmrveg

  root maintenance resp. rate for each pft
- real, dimension(ilg, icc) rgveg

  growth resp. rate for each pft
- real, dimension(ilg, icc) litrfallveg

  litter fall in kgc/m^2 for each pft
- real, dimension(ilg, iccp1) **humiftrsveg**
- real, dimension(ilg, icc) rothrlos

  root death as crops are harvested, kgc/m^2
- real, dimension(ilg, icc) pfcancmx

  previous year's fractional coverages of pfts
- real, dimension(ilg, icc) nfcancmx

  next year's fractional coverages of pfts
- real, dimension(ilg, ican) **alvsctm**
- real, dimension(ilg, ican) paic

  plant area index for class' 4 pfts. this is the sum of leaf area index and stem area index.
- real, dimension(ilg, ican) slaic

  storage lai. this will be used as min. lai that class sees so that it doesn't blow up in its stomatal conductance calculations.
- real, dimension(ilg, ican) **alirctm**
- real, dimension(ilg) **cfluxcg**
- real, dimension(ilg) **cfluxcs**
- real, dimension(ilg) dstcemls3

  carbon emission losses due to disturbance (fire at present) from litter pool
- real, dimension(ilg, icc) anveg

  net photosynthesis rate for each pft
- real, dimension(ilg, icc) rmlveg

  leaf maintenance resp. rate for each pft
- real, dimension(ilg) twarmm

  temperature of the warmest month (c)
- real, dimension(ilg) tcoldm

  temperature of the coldest month (c)
- real, dimension(ilg) gdd5

  growing degree days above 5 c
- real, dimension(ilg) aridity

  aridity index, ratio of potential evaporation to precipitation
- real, dimension(ilg) srplsmon

  number of months in a year with surplus water i.e. precipitation more than potential evaporation
- real, dimension(ilg) defctmon

  number of months in a year with water deficit i.e. precipitation less than potential evaporation
- real, dimension(ilg) anndefct

  annual water deficit (mm)
- real, dimension(ilg) annsrpls

  annual water surplus (mm)
- real, dimension(ilg) annpcp

  annual precipitation (mm)
- real, dimension(ilg) dry_season_length

  length of dry season (months)
- real, dimension(ilg) tcurm

  temperature of the current month (c)

- real, dimension(ilg) srpcuryr

  water surplus for the current year
- real, dimension(ilg) dftcuryr

  water deficit for the current year
- real, dimension(12, ilg) tmonth

  monthly temperatures
- real, dimension(ilg) anpcpcur

  annual precipitation for current year (mm)
- real, dimension(ilg) anpecur

  annual potential evaporation for current year (mm)
- real, dimension(ilg) gdd5cur

  growing degree days above 5 c for current year
- real, dimension(ilg) surmncur

  number of months with surplus water for current year
- real, dimension(ilg) defmncur

  number of months with water deficit for current year
- real, dimension(ilg) srplscur

  water surplus for the current month
- real, dimension(ilg) defctcur

  water deficit for the current month
- real, dimension(ilg, icc) geremort

  growth efficiency related mortality (1/day)
- real, dimension(ilg, icc) intrmort

  intrinsic (age related) mortality (1/day)
- real, dimension(ilg, icc) lambda

  Used to determine the colonization rate.
- real, dimension(ilg, icc) cc

  colonization rate & mortality rate
- real, dimension(ilg, icc) mm

  colonization rate & mortality rate
- logical, dimension(ilg, icc) pftexist

  logical array indicating pfts exist (t) or not (f)
- integer, dimension(ilg, 2) colddays

  cold days counter for tracking days below a certain temperature threshold for ndl dcd and crop pfts.
- integer, dimension(ilg) **icount**
- integer, dimension(ilg, icc) lfstatus

  leaf phenology status
- integer, dimension(ilg, icc) pandays

  days with positive net photosynthesis (an) for use in the phenology subroutine
- integer, dimension(ilg) stdaln

  an integer telling if ctem is operated within gcm (=0) or in stand alone mode (=1). this is used for fire purposes. see comments just above where disturb subroutine is called.

8.11.1 Detailed Description

CTEM's 'gat' vars.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

8.12 ctem_statevars::veg_rot Type Reference

CTEM's 'rot' vars.

Public Attributes

- real, dimension(nlat, nmos, icc) **ailcmin**
- real, dimension(nlat, nmos, icc) **ailcmax**
- real, dimension(nlat, nmos, icc) **dvdfcan**
- real, dimension(nlat, nmos, icc) gleafmas

  green leaf mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(nlat, nmos, icc) bleafmas

  brown leaf mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(nlat, nmos, icc) stemmass

  stem mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(nlat, nmos, icc) rootmass

  root mass for each of the 9 ctem pfts, kgc/m^2
- real, dimension(nlat, nmos, icc) pstemmass

  stem mass from previous timestep, is value before fire. used by burntobare subroutine
- real, dimension(nlat, nmos, icc) pgleafmass

  root mass from previous timestep, is value before fire. used by burntobare subroutine
- real, dimension(nlat, nmos, icc) fcancmx

  max. fractional coverage of ctem's 9 pfts, but this can be modified by land-use change, and competition between pfts
- real, dimension(nlat, nmos) gavglai

  grid averaged green leaf area index
- real, dimension(nlat, nmos, ican) zolnc

  lumped log of roughness length for class' 4 pfts
- real, dimension(nlat, nmos, ican) ailc

  lumped lai for class' 4 pfts
- real, dimension(nlat, nmos, icc) ailcg

  green lai for ctem's 9 pfts
- real, dimension(nlat, nmos, icc) ailcgs

  GREEN LAI FOR CANOPY OVER SNOW SUB-AREA.
- real, dimension(nlat, nmos, icc) fcancs

  FRACTION OF CANOPY OVER SNOW FOR CTEM's 9 PFTs.
- real, dimension(nlat, nmos, icc) fcanc

  FRACTIONAL COVERAGE OF 8 CARBON PFTs, CANOPY OVER SNOW.
- real, dimension(nlat, nmos) co2conc

  ATMOS. CO2 CONC. IN PPM.
- real, dimension(nlat, nmos) **ch4conc**
- real, dimension(nlat, nmos, icc) co2i1cg

  INTERCELLULAR CO2 CONC FOR 8 PFTs FOR CANOPY OVER GROUND SUBAREA (Pa) - FOR SINGLE/SU↩ NLIT LEAF.
- real, dimension(nlat, nmos, icc) co2i1cs

  SAME AS ABOVE BUT FOR SHADED LEAF (above being co2i1cg)
- real, dimension(nlat, nmos, icc) co2i2cg

  INTERCELLULAR CO2 CONC FOR 8 PFTs FOR CANOPY OVER SNOWSUBAREA (Pa) - FOR SINGLE/SUNLIT LEAF.
- real, dimension(nlat, nmos, icc) co2i2cs

  SAME AS ABOVE BUT FOR SHADED LEAF (above being co2i2cg)
- real, dimension(nlat, nmos, icc) ancsveg

net photosynthetic rate for ctems 9 pfts for canopy over snow subarea

- real, dimension(nlat, nmos, icc) ancgveg

  net photosynthetic rate for ctems 9 pfts for canopy over ground subarea
- real, dimension(nlat, nmos, icc) rmlcsveg

  leaf respiration rate for ctems 9 pfts forcanopy over snow subarea
- real, dimension(nlat, nmos, icc) rmlcgveg

  leaf respiration rate for ctems 9 pfts forcanopy over ground subarea
- real, dimension(nlat, nmos, icc) slai

  storage/imaginary lai for phenology purposes
- real, dimension(nlat, nmos, icc) ailcb

  brown lai for ctem's 9 pfts. for now we assume only grasses can have brown lai
- real, dimension(nlat, nmos) **canres**
- real, dimension(nlat, nmos, icc) flhrloss

  fall or harvest loss for deciduous trees and crops, respectively, kgc/m^2 *il1*
- real, dimension(nlat, nmos, icc) grwtheff

  growth efficiency. change in biomass per year per unit max. lai (kgc/m^2 *)/(m2/m2), for use in mortality subroutine*
- real, dimension(nlat, nmos, icc) lystmmas

  stem mass at the end of last year
- real, dimension(nlat, nmos, icc) lyrotmas

  root mass at the end of last year
- real, dimension(nlat, nmos, icc) tymaxlai

  this year's maximum lai
- real, dimension(nlat, nmos) vgbiomas

  grid averaged vegetation biomass, kgc/m^2
- real, dimension(nlat, nmos) gavgltms

  grid averaged litter mass, kgc/m^2
- real, dimension(nlat, nmos) gavgscms

  grid averaged soil c mass, kgc/m^2
- real, dimension(nlat, nmos, icc) stmhrlos

  stem harvest loss for crops, kgc/m^2
- real, dimension(nlat, nmos, ican, ignd) rmatc

  fraction of roots for each of class' 4 pfts in each soil layer
- real, dimension(nlat, nmos, icc, ignd) rmatctem

  fraction of roots for each of ctem's 9 pfts in each soil layer
- real, dimension(nlat, nmos, iccp1) litrmass

  litter mass for each of the 9 ctem pfts + bare, kgc/m^2
- real, dimension(nlat, nmos, iccp1) soilcmas

  soil carbon mass for each of the 9 ctem pfts + bare, kgc/m^2
- real, dimension(nlat, nmos, icc) vgbiomas_veg

  vegetation biomass for each pft
- real, dimension(nlat, nmos, icc) emit_co2

  carbon dioxide
- real, dimension(nlat, nmos, icc) emit_co

  carbon monoxide
- real, dimension(nlat, nmos, icc) emit_ch4

  methane
- real, dimension(nlat, nmos, icc) emit_nmhc

  non-methane hydrocarbons
- real, dimension(nlat, nmos, icc) emit_h2

  hydrogen gas
- real, dimension(nlat, nmos, icc) emit_nox

nitrogen oxides

- real, dimension(nlat, nmos, icc) emit_n2o

  nitrous oxide
- real, dimension(nlat, nmos, icc) emit_pm25

  particulate matter less than 2.5 um in diameter
- real, dimension(nlat, nmos, icc) emit_tpm

  total particulate matter
- real, dimension(nlat, nmos, icc) emit_tc

  total carbon
- real, dimension(nlat, nmos, icc) emit_oc

  organic carbon
- real, dimension(nlat, nmos, icc) emit_bc

  black carbon
- real, dimension(nlat, nmos) burnfrac

  areal fraction burned due to fire for every grid cell (%)
- real, dimension(nlat, nmos, icc) burnvegf

  per PFT fraction burned of that PFT's area
- real, dimension(nlat, nmos, icc) **smfuncveg**
- real, dimension(nlat, nmos) popdin

  population density $(people/km^2)$
- real, dimension(nlat, nmos, icc) bterm

  biomass term for fire probabilty calc
- real, dimension(nlat, nmos) lterm

  lightning term for fire probabilty calc
- real, dimension(nlat, nmos, icc) mterm

  moisture term for fire probabilty calc
- real, dimension(nlat, nmos) extnprob

  fire extingusinging probability
- real, dimension(nlat, nmos) prbfrhuc

  probability of fire due to human causes
- real, dimension(nlat, nmos, 12) **mlightng**
- real, dimension(nlat) dayl_max

  maximum daylength for that location (hours)
- real, dimension(nlat) dayl

  daylength for that location (hours)
- real, dimension(nlat, nmos, icc) bmasveg

  total (gleaf + stem + root) biomass for each ctem pft, kgc/m^2
- real, dimension(nlat, nmos, ican) cmasvegc

  total canopy mass for each of the 4 class pfts. recall that class requires canopy mass as an input, and this is now provided by ctem. kg/m^2.
- real, dimension(nlat, nmos, icc) veghght

  vegetation height (meters)
- real, dimension(nlat, nmos, icc) rootdpth

  99% soil rooting depth (meters) both veghght & rootdpth can be used as diagnostics to see how vegetation grows above and below ground, respectively
- real, dimension(nlat, nmos) rml

  leaf maintenance respiration (u-mol co2/m2.sec)
- real, dimension(nlat, nmos) rms

  stem maintenance respiration (u-mol co2/m2.sec)
- real, dimension(nlat, nmos, icc) tltrleaf

  total leaf litter fall rate (u-mol co2/m2.sec)

- real, dimension(nlat, nmos, icc) tltrstem

  total stem litter fall rate (u-mol co2/m2.sec)
- real, dimension(nlat, nmos, icc) tltrroot

  total root litter fall rate (u-mol co2/m2.sec)
- real, dimension(nlat, nmos, icc) leaflitr

  leaf litter fall rate (u-mol co2/m2.sec). this leaf litter does not include litter generated due to mortality/fire
- real, dimension(nlat, nmos, icc) roottemp

  root temperature, k
- real, dimension(nlat, nmos, icc) afrleaf

  allocation fraction for leaves
- real, dimension(nlat, nmos, icc) afrstem

  allocation fraction for stem
- real, dimension(nlat, nmos, icc) afrroot

  allocation fraction for root
- real, dimension(nlat, nmos, icc) wtstatus

  soil water status used for calculating allocation fractions
- real, dimension(nlat, nmos, icc) ltstatus

  light status used for calculating allocation fractions
- real, dimension(nlat, nmos) rmr

  root maintenance respiration (u-mol co2/m2.sec)
- real, dimension(nlat, nmos, 8) slopefrac

  prescribed fraction of wetlands based on slope only(0.025, 0.05, 0.1, 0.15, 0.20, 0.25, 0.3 and 0.35 percent slope thresholds)
- real, dimension(nlat, nmos) ch4wet1

  methane flux from wetlands calculated using hetrores in umol ch4/m2.s
- real, dimension(nlat, nmos) ch4wet2

  methane flux from wetlands calculated using npp in umol ch4/m2.s
- real, dimension(nlat, nmos) wetfdyn

  dynamic wetland fraction
- real, dimension(nlat, nmos) ch4dyn1

  methane flux from wetlands calculated using hetrores and wetfdyn, in umol ch4/m2.s
- real, dimension(nlat, nmos) ch4dyn2

  methane flux from wetlands calculated using npp and wetfdyn, in umol ch4/m2.s
- real, dimension(nlat, nmos, 12) **wetfrac_mon**
- real, dimension(nlat, nmos) ch4_soills

  Methane uptake into the soil column ($mg CH_4 m^{-2} s^{-1}$)
- real, dimension(nlat, nmos) lucemcom

  land use change (luc) related combustion emission losses, u-mol co2/m2.sec
- real, dimension(nlat, nmos) lucltrin

  luc related inputs to litter pool, u-mol co2/m2.sec
- real, dimension(nlat, nmos) lucsocin

  luc related inputs to soil c pool, u-mol co2/m2.sec
- real, dimension(nlat, nmos) npp

  net primary productivity
- real, dimension(nlat, nmos) nep

  net ecosystem productivity
- real, dimension(nlat, nmos) nbp

  net biome productivity
- real, dimension(nlat, nmos) gpp

  gross primary productivity
- real, dimension(nlat, nmos) hetrores

heterotrophic respiration

- real, dimension(nlat, nmos) autores

  autotrophic respiration
- real, dimension(nlat, nmos) **soilcresp**
- real, dimension(nlat, nmos) rm

  maintenance respiration
- real, dimension(nlat, nmos) rg

  growth respiration
- real, dimension(nlat, nmos) litres

  litter respiration
- real, dimension(nlat, nmos) socres

  soil carbon respiration
- real, dimension(nlat, nmos) dstcemls

  carbon emission losses due to disturbance, mainly fire
- real, dimension(nlat, nmos) litrfall

  total litter fall (from leaves, stem, and root) due to all causes (mortality, turnover, and disturbance)
- real, dimension(nlat, nmos) humiftrs

  transfer of humidified litter from litter to soil c pool
- real, dimension(nlat, nmos, icc) gppveg

  !gross primary productity for each pft
- real, dimension(nlat, nmos, iccp1) nepveg

  net ecosystem productity for bare fraction expnbaln(i)=0.0 amount of c related to spatial expansion Not used JM Jun 2014 OR net ecosystem productity for each pft
- real, dimension(nlat, nmos, iccp1) nbpveg

  net biome productity for bare fraction OR net biome productity for each pft
- real, dimension(nlat, nmos, icc) nppveg

  npp for individual pfts, u-mol co2/m2.sec
- real, dimension(nlat, nmos, iccp1) **hetroresveg**
- real, dimension(nlat, nmos, icc) **autoresveg**
- real, dimension(nlat, nmos, iccp1) **litresveg**
- real, dimension(nlat, nmos, iccp1) **soilcresveg**
- real, dimension(nlat, nmos, icc) **rmlvegacc**
- real, dimension(nlat, nmos, icc) rmsveg

  stem maintenance resp. rate for each pft
- real, dimension(nlat, nmos, icc) rmrveg

  root maintenance resp. rate for each pft
- real, dimension(nlat, nmos, icc) rgveg

  growth resp. rate for each pft
- real, dimension(nlat, nmos, icc) litrfallveg

  litter fall in kgc/m^2 for each pft
- real, dimension(nlat, nmos, iccp1) **humiftrsveg**
- real, dimension(nlat, nmos, icc) rothrlos

  root death as crops are harvested, kgc/m^2
- real, dimension(nlat, nmos, icc) pfcancmx

  previous year's fractional coverages of pfts
- real, dimension(nlat, nmos, icc) nfcancmx

  next year's fractional coverages of pfts
- real, dimension(nlat, nmos, ican) **alvsctm**
- real, dimension(nlat, nmos, ican) paic

  plant area index for class' 4 pfts. this is the sum of leaf area index and stem area index.
- real, dimension(nlat, nmos, ican) slaic

storage lai. this will be used as min. lai that class sees so that it doesn't blow up in its stomatal conductance calculations.

- real, dimension(nlat, nmos, ican) **alirctm**
- real, dimension(nlat, nmos) **cfluxcg**
- real, dimension(nlat, nmos) **cfluxcs**
- real, dimension(nlat, nmos) dstcemls3

  carbon emission losses due to disturbance (fire at present) from litter pool
- real, dimension(nlat, nmos, icc) anveg

  net photosynthesis rate for each pft
- real, dimension(nlat, nmos, icc) rmlveg

  leaf maintenance resp. rate for each pft
- logical, dimension(nlat, nmos, icc) pftexist

  logical array indicating pfts exist (t) or not (f)
- integer, dimension(nlat, nmos, 2) colddays

  cold days counter for tracking days below a certain temperature threshold for ndl dcd and crop pfts.
- integer, dimension(nlat, nmos) **icount**
- integer, dimension(nlat, nmos, icc) lfstatus

  leaf phenology status
- integer, dimension(nlat, nmos, icc) pandays

  days with positive net photosynthesis (an) for use in the phenology subroutine
- integer, dimension(nlat, nmos) stdaln

  an integer telling if ctem is operated within gcm (=0) or in stand alone mode (=1). this is used for fire purposes. see comments just above where disturb subroutine is called.
- real, dimension(nlat, nmos) **preacc_m**
- real, dimension(nlat, nmos) **gtacc_m**
- real, dimension(nlat, nmos) **qevpacc_m**
- real, dimension(nlat, nmos) **hfsacc_m**
- real, dimension(nlat, nmos) **hmfnacc_m**
- real, dimension(nlat, nmos) **rofacc_m**
- real, dimension(nlat, nmos) **snoacc_m**
- real, dimension(nlat, nmos) **ovracc_m**
- real, dimension(nlat, nmos) **wtblacc_m**
- real, dimension(nlat, nmos, ignd) **tbaracc_m**
- real, dimension(nlat, nmos, ignd) **thlqacc_m**
- real, dimension(nlat, nmos, ignd) **thicacc_m**
- real, dimension(nlat, nmos, ignd) **thalacc_m**
- real, dimension(nlat, nmos) **alvsacc_m**
- real, dimension(nlat, nmos) **aliracc_m**
- real, dimension(nlat, nmos) **rhosacc_m**
- real, dimension(nlat, nmos) **tsnoacc_m**
- real, dimension(nlat, nmos) **wsnoacc_m**
- real, dimension(nlat, nmos) **snoare_m**
- real, dimension(nlat, nmos) **tcanacc_m**
- real, dimension(nlat, nmos) **rcanacc_m**
- real, dimension(nlat, nmos) **scanacc_m**
- real, dimension(nlat, nmos) altotacc_m

  Daily broadband albedo.
- integer, dimension(nlat) altotcntr_d

  Used to count the number of time steps with the sun above the horizon.
- real, dimension(nlat, nmos) **groacc_m**
- real, dimension(nlat, nmos) **fsinacc_m**
- real, dimension(nlat, nmos) **flinacc_m**
- real, dimension(nlat, nmos) **taacc_m**

- real, dimension(nlat, nmos) **uvacc_m**
- real, dimension(nlat, nmos) **presacc_m**
- real, dimension(nlat, nmos) **qaacc_m**
- real, dimension(nlat, nmos) **evapacc_m**
- real, dimension(nlat, nmos) **flutacc_m**
- real, dimension(nlat, nmos) **tcanrs**
- real, dimension(nlat, nmos) **tsnors**
- real, dimension(nlat, nmos) **tpndrs**
- real, dimension(nlat, nmos, ican) **csum**
- real, dimension(nlat, nmos, ignd) **tbaraccrow_m**
- real, dimension(nlat, nmos) **tcanoaccrow_m**
- real, dimension(nlat, nmos) **uvaccrow_m**
- real, dimension(nlat, nmos) **vvaccrow_m**
- real, dimension(nlat, nmos) **tcanoaccrow_out**
- real, dimension(nlat, nmos) **qevpacc_m_save**
- real, dimension(nlat, nmos) twarmm

  temperature of the warmest month (c)
- real, dimension(nlat, nmos) tcoldm

  temperature of the coldest month (c)
- real, dimension(nlat, nmos) gdd5

  growing degree days above 5 c
- real, dimension(nlat, nmos) aridity

  aridity index, ratio of potential evaporation to precipitation
- real, dimension(nlat, nmos) srplsmon

  number of months in a year with surplus water i.e. precipitation more than potential evaporation
- real, dimension(nlat, nmos) defctmon

  number of months in a year with water deficit i.e. precipitation less than potential evaporation
- real, dimension(nlat, nmos) anndefct

  annual water deficit (mm)
- real, dimension(nlat, nmos) annsrpls

  annual water surplus (mm)
- real, dimension(nlat, nmos) annpcp

  annual precipitation (mm)
- real, dimension(nlat, nmos) dry_season_length

  length of dry season (months)

8.12.1 Detailed Description

CTEM's 'rot' vars.

The documentation for this type was generated from the following file:

- ctem_statevars.f90

Chapter 9

File Documentation

9.1 allocate.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Allocation Subroutine.

Functions/Subroutines

- subroutine allocate (lfstatus,thliq,ailcg,ailcb,il1,il2,sand,clay,rmatctem, gleafmas, stemmass, rootmass,sort, nol2pfts, fcancmx, isand,

9.1.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Allocation Subroutine.

Positive NPP is allocated daily to the leaf, stem and root components, which generally causes their respective biomass to increase, although the biomass may also decrease depending on the autotrophic respiration flux of a component. Negative NPP generally causes net carbon loss from the components. While CTEM offers the ability to use both specified constant or dynamically calculated allocation fractions for leaves, stems and roots, in practice the dynamic allocation fractions are primarily used. The formulation used in CTEM v. 2.0 differs from that for CTEM v. 1.0 as described in [10] only in the parameter values.

The dynamic allocation to the live plant tissues is based on the light, water and leaf phenological status of vegetation. The preferential allocation of carbon to the different tissue pools is based on three assumptions: (i) if soil moisture is limiting, carbon should be preferentially allocated to roots for greater access to water, (ii) if LAI is low, carbon should be allocated to leaves for enhanced photosynthesis and finally (iii) carbon is allocated to the stem to increase vegetation height and lateral spread of vegetation when the increase in LAI results in a decrease in light penetration.

The vegetation water status, W, is determined as a linear scalar quantity that varies between 0 and 1 for each PFT and calculated by weighting the degree of soil saturation (${i}({i})$, Eq. phitheta}) with the fraction of roots in each soil layer

$$W = \phi_{root} = \sum_{i=1}^{g} \phi_i(\theta_i) r_i.$$

The light status, L, is parametrized as a function of LAI and nitrogen extinction coefficient, k_{n} (PFT-dependent; see also ctem_params.f90), as

$$L = \{ \exp(-k_{\mathrm{n}} LAI), \quad treesandcrops \max\left(0, 1 - \frac{LAI}{4.5}\right), \quad grasses.$$

For PFTs with a stem component (i.e. tree and crop PFTs), the fractions of positive NPP allocated to stem (a_{fS}),

leaf (a_{fL}) and root (a_{fR}) components are calculated as

$$a_{fS} = \frac{\epsilon_{\rm S} + \omega_{\rm a}(1-L)}{1+\omega_{\rm a}(2-L-W)}$$

$$a_{fR} = \frac{\epsilon_{\rm R} + \omega_{\rm a}(1-W)}{1+\omega_{\rm a}(2-L-W)},$$

$$a_{fL} = \frac{\epsilon_{\rm L}}{1+\omega_{\rm a}(2-L-W)} = 1 - a_{fS} - a_{fR}.$$

The base allocation fractions for each component (leaves – $\epsilon_{\rm L}$, stem – $\epsilon_{\rm S}$, and roots – $\epsilon_{\rm R}$) are PFT-dependent (see also ctem_params.f90) and sum to 1, i.e. $\epsilon_{\rm L} + \epsilon_{\rm S} + \epsilon_{\rm R} = 1$. The parameter $\omega_{\rm a}$, which varies by PFT (see also ctem_params.f90), determines the sensitivity of the allocation scheme to changes in W and L. Larger values of $\omega_{\rm a}$ yield higher sensitivity to changes in L and W.

Grasses do not have a stem component (i.e. $a_{fS} = 0$) and the allocation fractions for leaf and root components are given by

$$a_{fL} = \frac{\epsilon_{\rm L} + \omega_{\rm a} L}{1+\omega_{\rm a}(1+L-W)}, a_{fR} = \frac{\epsilon_{\rm R} + \omega_{\rm a}(1-W)}{1+\omega_{\rm a}(1+L-W)}.$$

The above equations ensure that the allocation fractions add up to one ($a_{fL} + a_{fR} + a_{fS} = 1$).

The dynamic allocation fractions are superseded under three conditions. First, during the leaf onset for crops and deciduous trees, all carbon must be allocated to leaves ($a_{fL} = 1$, $a_{fS} = a_{fR} = 0$). Second, the proportion of stem plus root biomasses to leaf biomass must satisfy the relationship:

$$C_{\rm S} + C_{\rm R} = \eta C_{\rm L}^{\kappa},$$

where $C_{\rm S}$, $C_{\rm R}$ and $C_{\rm L}$ are the carbon in the stem, root and leaves, respectively. The parameter η is PFT-specific (see also ctem_params.f90) and parameter κ has a value of 1.6 for trees and crops and 1.2 for grasses. Both parameters are based on the Frankfurt Biosphere Model (FBM) [35]. This constraint (Eq. propwoody}) is based on the physical requirement of sufficient stem and root tissues to support a given leaf biomass. As grasses have no stem component, Eq. (propwoody}) determines their root to shoot ratio (i.e. the ratio of belowground to aboveground biomass). The final condition ensures that a minimum realistic root to shoot ratio is maintained for all PFTs (lr_{min}, see also ctem_params.f90). Root mass is required for nutrient and water uptake and support for the aboveground biomass. If the minimum root to shoot ratio is not being maintained, carbon is allocated preferentially to roots.

9.1.2 Function/Subroutine Documentation

9.1.2.1 subroutine allocate (integer, dimension(ilg,icc) *lfstatus,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,icc) *ailcg,* real, dimension(ilg,icc) *ailcb,* integer *il1,* integer *il2,* real, dimension(ilg,ignd) *sand,* real, dimension(ilg,ignd) *clay,* real, dimension(ilg,icc,ignd) *rmatctem,* real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* integer, dimension(icc) *sort,* integer, dimension(ican) *nol2pfts,* real, dimension(ilg,icc) *fcancmx,* integer, dimension(ilg,ignd) *isand*)

Parameters

il1	input: il1=1
il2	input: il2=ilg
lfstatus	input: leaf status. an integer indicating if leaves are in "max. growth", "normal growth", "fall/harvest", or "no leaves" mode. see phenolgy subroutine for more details.
sort	input: index for correspondence between 9 pfts and the 12 values in parameters vectors
nol2pfts	input: number of level 2 ctem pfts

ailcg	input: green or live leaf area index
ailcb	input: brown or dead leaf area index
thliq	input: liquid soil moisture content in 3 soil layers
rootmass	input: root mass for each of the 9 ctem pfts, kg c/m2
rmatctem	input: fraction of roots in each soil layer for each pft
gleafmas	input: green or live leaf mass in kg c/m2, for the 9 pfts
stemmass	input: stem mass for each of the 9 ctem pfts, kg c/m2
sand	input: percentage sand
clay	input: percentage clay
fcancmx	input: max. fractional coverage of ctem's 9 pfts, but this can be modified by land-use change, and competition between pfts

Estimate field capacity and wilting point soil moisture contents

Wilting point corresponds to matric potential of 150 m field capacity corresponds to hydarulic conductivity of 0.10 mm/day -> 1.157x1e-09 m/s

Calculate liquid soil moisture content, and wilting and field capacity soil moisture contents averaged over the root zone. note that while the soil moisture content is same under the entire gcm grid cell, soil moisture averaged over the rooting depth is different for each pft because of different fraction of roots present in each soil layer.

Using liquid soil moisture content together with wilting and field capacity soil moisture contents averaged over the root zone, find soil water status.

Calculate light status as a function of lai and light extinction parameter. for now set nitrogen status equal to 1, which means nitrogen is non-limiting.

allocation to roots is determined by min. of water and nitrogen status

now that we know water, light, and nitrogen status we can find allocation fractions for leaves, stem, and root components. note that allocation formulae for grasses are different from those for trees and crops, since there is no stem component in grasses.

if using constant allocation factors then replace the dynamically calculated allocation fractions.

make sure allocation fractions add to one

the allocation fractions calculated above are overridden by two rules.

rule 1 which states that at the time of leaf onset which corresponds to leaf status equal to 1, more c is allocated to leaves so that they can grow asap. in addition when leaf status is "fall/harvest" then nothing is allocated to leaves.

for grasses we use the usual allocation even at leaf onset

rule 2 overrides rule 1 above and makes sure that we do not allow the amount of leaves on trees and crops (i.e. pfts 1 to 7) to exceed an amount such that the remaining woody biomass cannot support. if this happens, allocation to leaves is reduced and most npp is allocated to stem and roots, in a proportion based on calculated afrstem and afrroot. for grasses this rule essentially constrains the root:shoot ratio, meaning that the model grasses can't have lots of leaves without having a reasonable amount of roots.

make sure that root:shoot ratio is at least equal to rtsrmin. if not allocate more to root and decrease allocation to stem.

finally check if all allocation fractions are positive and check again they all add to one.

9.2 APREP.f File Reference

Purpose: Calculate various land surface parameters.

Functions/Subroutines

- subroutine aprep (FC, FG, FCS, FGS, PAICAN, PAICNS, FSVF, FSVFS, FRAINC, FSNOWC, FRAICS, F↩ SNOCS, RAICAN, RAICNS, SNOCAN, SNOCNS, DISP, DISPS, ZOMLNC, ZOMLCS, ZOELNC, ZOELC↩ S, ZOMLNG, ZOMLNS, ZOELNG, ZOELNS, CHCAP, CHCAPS, CMASSC, CMASCS, CWLCAP, CWFCAP,

CWLCPS, CWFCPS, RBCOEF, ZPLIMC, ZPLIMG, ZPLMCS, ZPLMGS, HTCC, HTCS, HTC, FROOT, FR↩
OOTS, WTRC, WTRS, WTRG, CMAI, PAI, PAIS, AIL, FCAN, FCANS, PSIGND, FCANMX, ZOLN, PAIMAX, PAIMIN, CWGTMX, ZRTMAX, PAIDAT, HGTDAT, THLIQ, THICE, TBAR, RCAN, SNCAN, TCAN, GROWTH, ZSNOW, TSNOW, FSNOW, RHOSNO, SNO, Z0ORO, ZBLEND, ZPLMG0, ZPLMS0, TA, RHOAIR, RAD↩
J, DLON, RHOSNI, DELZ, DELZW, ZBOTW, THPOR, THLMIN, PSISAT, BI, PSIWLT, HCPS, ISAND, ILG, IL1, IL2, JL, IC, ICP1, IG, IDAY, IDISP, IZREF, IWF, IPAI, IHGT, RMAT, H, HS, CWCPAV, GROWA, GROWN, GROWB, RRESID, SRESID, FRTOT, FRTOTS, FCANCMX, ICTEM, ctem_on, RMATC, AILC, PAIC, AILCG, L2MAX, NOL2PFTS, AILCGS, FCANCS, FCANC, ZOLNC, CMASVEGC, SLAIC)

This subroutine is hard-coded to handle the standard four vegetation categories recognized by CLASS (needleleaf trees, broadleaf trees, crops and grass), so a call to abort is performed if the number of vegetation classes, IC, is not equal to 4. A set of diagnostic and accumulator arrays is then initialized to zero, and the liquid water suction in the soil is set to an arbitrarily high value.

9.2.1 Detailed Description

Purpose: Calculate various land surface parameters.

9.2.2 Function/Subroutine Documentation

9.2.2.1 subroutine aprep (real, dimension (ilg) *FC,* real, dimension (ilg) *FG,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FGS,* real, dimension(ilg) *PAICAN,* real, dimension(ilg) *PAICNS,* real, dimension (ilg) *FSVF,* real, dimension (ilg) *FSVFS,* real, dimension(ilg) *FRAINC,* real, dimension(ilg) *FSNOWC,* real, dimension(ilg) *FRAICS,* real, dimension(ilg) *FSNOCS,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *RAICNS,* real, dimension(ilg) *SNOCAN,* real, dimension(ilg) *SNOCNS,* real, dimension (ilg) *DISP,* real, dimension (ilg) *DISPS,* real, dimension(ilg) *ZOMLNC,* real, dimension(ilg) *ZOMLCS,* real, dimension(ilg) *ZOELNC,* real, dimension(ilg) *ZOELCS,* real, dimension(ilg) *ZOMLNG,* real, dimension(ilg) *ZOMLNS,* real, dimension(ilg) *ZOELNG,* real, dimension(ilg) *ZOELNS,* real, dimension (ilg) *CHCAP,* real, dimension(ilg) *CHCAPS,* real, dimension(ilg) *CMASSC,* real, dimension(ilg) *CMASCS,* real, dimension(ilg) *CWLCAP,* real, dimension(ilg) *CWFCAP,* real, dimension(ilg) *CWLCPS,* real, dimension(ilg) *CWFCPS,* real, dimension(ilg) *RBCOEF,* real, dimension(ilg) *ZPLIMC,* real, dimension(ilg) *ZPLIMG,* real, dimension(ilg) *ZPLMCS,* real, dimension(ilg) *ZPLMGS,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg,ig) *FROOT,* real, dimension(ilg,ig) *FROOTS,* real, dimension (ilg) *WTRC,* real, dimension (ilg) *WTRS,* real, dimension (ilg) *WTRG,* real, dimension (ilg) *CMAI,* real, dimension (ilg,ic) *PAI,* real, dimension (ilg,ic) *PAIS,* real, dimension (ilg,ic) *AIL,* real, dimension (ilg,ic) *FCAN,* real, dimension (ilg,ic) *FCANS,* real, dimension(ilg) *PSIGND,* real, dimension(ilg,icp1) *FCANMX,* real, dimension (ilg,icp1) *ZOLN,* real, dimension(ilg,ic) *PAIMAX,* real, dimension(ilg,ic) *PAIMIN,* real, dimension(ilg,ic) *CWGTMX,* real, dimension(ilg,ic) *ZRTMAX,* real, dimension(ilg,ic) *PAIDAT,* real, dimension(ilg,ic) *HGTDAT,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg) *RCAN,* real, dimension (ilg) *SNCAN,* real, dimension (ilg) *TCAN,* real, dimension(ilg) *GROWTH,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *FSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *SNO,* real, dimension (ilg) *Z0ORO,* real, dimension(ilg) *ZBLEND,* real, dimension(ilg) *ZPLMG0,* real, dimension(ilg) *ZPLMS0,* real, dimension (ilg) *TA,* real, dimension(ilg) *RHOAIR,* real, dimension (ilg) *RADJ,* real, dimension (ilg) *DLON,* real, dimension(ilg) *RHOSNI,* real, dimension (ig) *DELZ,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *ZBOTW,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLMIN,* real, dimension(ilg,ig) *PSISAT,* real, dimension (ilg,ig) *BI,* real, dimension(ilg,ig) *PSIWLT,* real, dimension (ilg,ig) *HCPS,* integer, dimension (ilg,ig) *ISAND,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IC,* integer *ICP1,* integer *IG,* integer *IDAY,* integer *IDISP,* integer *IZREF,* integer *IWF,* integer *IPAI,* integer *IHGT,* real, dimension (ilg,ic,ig) *RMAT,* real, dimension (ilg,ic) *H,* real, dimension (ilg,ic) *HS,* real, dimension(ilg) *CWCPAV,* real, dimension (ilg) *GROWA,* real, dimension (ilg) *GROWN,* real, dimension (ilg) *GROWB,* real, dimension(ilg) *RRESID,* real, dimension(ilg) *SRESID,* real, dimension (ilg) *FRTOT,* real, dimension(ilg) *FRTOTS,* real, dimension(ilg,ictem) *FCANCMX,* integer *ICTEM,* logical *ctem_on,* real, dimension(ilg,ic,ig) *RMATC,* real, dimension (ilg,ic) *AILC,* real, dimension (ilg,ic) *PAIC,* real, dimension(ilg,ictem) *AILCG,* integer *L2MAX,* integer, dimension(ic) *NOL2PFTS,* real, dimension (ilg,ictem) *AILCGS,* real, dimension (ilg,ictem) *FCANCS,* real, dimension(ilg,ictem) *FCANC,* real, dimension(ilg,ic) *ZOLNC,* real, dimension(ilg,ic) *CMASVEGC,* real, dimension(ilg,ic) *SLAIC*)

This subroutine is hard-coded to handle the standard four vegetation categories recognized by CLASS (needleleaf trees, broadleaf trees, crops and grass), so a call to abort is performed if the number of vegetation classes, IC, is not equal to 4. A set of diagnostic and accumulator arrays is then initialized to zero, and the liquid water suction in the soil is set to an arbitrarily high value.

Parameters

fc	Subarea fractional coverage of modelled area (X) [ ]
fg	Subarea fractional coverage of modelled area (X) [ ]
fcs	Subarea fractional coverage of modelled area (X) [ ]
fgs	Subarea fractional coverage of modelled area (X) [ ]
paican	Plant area index of canopy over bare ground (Λ_p) [ ]
paicns	Plant area index of canopy over snow (Λ_p) [ ]
fsvf	Sky view factor for bare ground under canopy (χ) [ ]
fsvfs	Sky view factor for snow under canopy (χ) [ ]
frainc	Fractional coverage of canopy by liquid water over snow-free subarea [ ]
fsnowc	Fractional coverage of canopy by frozen water over snow-free subarea [ ]
fraics	Fractional coverage of canopy by liquid water over snow-covered subarea [ ]
fsnocs	Fractional coverage of canopy by frozen water over snow-covered subarea [ ]
raican	Intercepted liquid water stored on canopy over bare ground (W_l) [kgm^{-2}]
raicns	Intercepted liquid water stored on canopy over snow (W_l) [kgm^{-2}]
snocan	Intercepted frozen water stored on canopy over bare soil (W_f) [kgm^{-2}]
snocns	Intercepted frozen water stored on canopy over snow (W_f) [kgm^{-2}]
disp	Displacement height of vegetation over bare ground (d) [m]
disps	Displacement height of vegetation over snow (d) [m]
zomlnc	Logarithm of roughness length for momentum of vegetation over bare ground [ ]
zomlcs	Logarithm of roughness length for momentum of vegetation over snow [ ]
zoelnc	Logarithm of roughness length for heat of vegetation over bare ground [ ]
zoelcs	Logarithm of roughness length for heat of vegetation over snow [ ]
zomlng	Logarithm of roughness length for momentum of bare ground [ ]
zomlns	Logarithm of roughness length for momentum of snow [ ]
zoelng	Logarithm of roughness length for heat of bare ground [ ]
zoelns	Logarithm of roughness length for heat of snow [ ]
rbcoef	Parameter for calculation of leaf boundary resistance (C_{rb})
chcap	Heat capacity of canopy over bare ground [$Jm^{-2}K^{-1}$]
chcaps	Heat capacity of canopy over snow [$Jm^{-2}K^{-1}$]
cmassc	Mass of canopy over bare ground [kgm^{-2}]
cmascs	Mass of canopy over snow [kgm^{-2}]
cwlcap	Storage capacity of canopy over bare ground for liquid water ($W_{l,max}$) [kgm^{-2}]
cwfcap	Storage capacity of canopy over bare ground for frozen water ($W_{f,max}$) [kgm^{-2}]
cwlcps	Storage capacity of canopy over snow for liquid water ($W_{l,max}$) [kgm^{-2}]
cwfcps	Storage capacity of canopy over snow for frozen water ($W_{f,max}$) [kgm^{-2}]
zplimc	Maximum water ponding depth for ground under canopy [m]
zplimg	Maximum water ponding depth for bare ground [m]
zplmcs	Maximum water ponding depth for ground under snow under canopy [m]
zplmgs	Maximum water ponding depth for ground under snow [m]
htcc	Diagnosed internal energy change of vegetation canopy due to conduction and/or change in mass [Wm^{-2}]
htcs	Diagnosed internal energy change of snow pack due to conduction and/or change in mass [Wm^{-2}]

wtrc	Diagnosed residual water transferred off the vegetation canopy [$kgm^{-2}s^{-1}$]
wtrs	Diagnosed residual water transferred into or out of the snow pack [$kgm^{-2}s^{-1}$]
wtrg	Diagnosed residual water transferred into or out of the soil [$kgm^{-2}s^{-1}$]
cmai	Aggregated mass of vegetation canopy [kgm^{-2}]
froot	Fraction of total transpiration contributed by soil layer [ ]
htc	Diagnosed internal energy change of soil layer due to conduction and/or change in mass [Wm^{-2}]
pai	Plant area index of vegetation category over bare ground [ ]
pais	Plant area index of vegetation category over snow [ ]
ail	Leaf area index of vegetation category over bare ground [ ]
fcan	Fractional coverage of vegetation category over bare ground (X_i) [ ]
fcans	Fractional coverage of vegetation category over snow (X_i) [ ]
psignd	Minimum liquid moisture suction in soil layers [m]
fcanmx	Maximum fractional coverage of modelled area by vegetation category [ ]
zoln	Natural logarithm of maximum roughness length of vegetation category [ ]
paimax	Maximum plant area index of vegetation category [ ]
paimin	Minimum plant area index of vegetation category [ ]
cwgtmx	Maximum canopy mass for vegetation category [kgm^{-2}]
zrtmax	Maximum rooting depth of vegetation category [m]
paidat	Optional user-specified value of plant area indices of vegetation categories to override CL↩ ASS-calculated values [ ]
hgtdat	Optional user-specified values of height of vegetation categories to override CLASS-calculated values [m]
thliq	Volumetric liquid water content of soil layers (θ l) [m^3m^{-3}]
thice	Frozen water content of soil layers under vegetation [m^3m^{-3}]
tbar	Temperature of soil layers [K]
rcan	Intercepted liquid water stored on canopy (W_l) [kgm^{-2}]
sncan	Intercepted frozen water stored on canopy (W_f) [kgm^{-2}]
tcan	Vegetation canopy temperature [K]
growth	Vegetation growth index [ ]
zsnow	Depth of snow pack (z_s) [m]
tsnow	Snowpack temperature [K]
fsnow	Diagnosed fractional snow coverage [ ]
rhosno	Density of snow (s) [kgm^{-3}]
sno	Mass of snow pack (W_s) [kgm^{-2}]
ta	Air temperature at reference height [K]
rhoair	Density of air [kgm^{-3}]
dlon	Longitude of grid cell (east of Greenwich) [degrees]
z0oro	Orographic roughness length [m]
zblend	Atmospheric blending height for surface roughness length averaging (z_b) [m]
rhosni	Density of fresh snow (ρ s,f) [kgm^{-3}]
zplmg0	Maximum water ponding depth for snow-free subareas (user-specified when running MESH code) [m]
zplms0	Maximum water ponding depth for snow-covered subareas (user-specified when running M↩ ESH code) [m]
radj	Latitude of grid cell (positive north of equator) [rad]
delzw	Permeable thickness of soil layer [m]
zbotw	Depth to permeable bottom of soil layer [m]

thpor	Pore volume in soil layer (θ p) [m^3m^{-3}]
thlmin	Residual soil liquid water content remaining after freezing or evaporation [m^3m^{-3}]
psisat	Soil moisture suction at saturation (Ψ sat) [m]
bi	Clapp and Hornberger empirical "b" parameter [ ]
psiwlt	Soil moisture suction at wilting point (Ψ w) [m]
hcps	Volumetric heat capacity of soil particles [Jm^{-3}]
isand	Sand content flag
delz	Soil layer thickness [m]
ailcg	GREEN LAI FOR USE WITH PHTSYN SUBROUTINE
ailcgs	GREEN LAI FOR CANOPY OVER SNOW SUB-AREA
fcanc	FRACTION OF CANOPY OVER GROUND FOR CTEM's 9 PFTs
fcancs	FRACTION OF CANOPY OVER SNOW FOR CTEM's 9 PFTs

In the 120 loop, the growth index for crops, GROWA, is calculated (if CLASS is not being run coupled to CTEM). This is done by referring to the three-dimensional array GROWYR, which contains values corresponding to the four Julian days of the year on which crops are planted, on which they reach maturity, on which harvesting begins, and on which the harvest is complete, for each ten-degree latitude half-circle in each hemisphere. These are generic, average dates, approximated using information gleaned from annual UN FAO (Food and Agriculture Organization) reports. (In the tropics, it is assumed that areas classified as agricultural are constantly under cultivation, so all four values are set to zero.)

First, the latitude of the modelled area is converted from a value in radians to a value from 1 to 18, IN, corresponding to the index of the latitude circle (1 for latitudes $80-90^oS$, 18 for latitudes $80-90^oN$). Then the hemisphere index, NL, is set to 1 for the Eastern Hemisphere, and 2 for the Western Hemisphere. If the planting date for the modelled area is zero (indicating a location in the tropics), GROWA is set to 1. Otherwise, GROWA is set to 1 if the day of the year lies between the maturity date and the start of the 54harvest, and to zero if the day of the year lies between the end of the harvest and the planting date. For dates in between, the value of GROWA is interpolated between 0 and 1. Checks are performed at the end to ensure that GROWA is not less than 0 or greater than 1. If the calculated value of GROWA is vanishingly small, it is set to zero.

In the 150 loop the other three growth indices are evaluated, as well as the vegetation heights and plant area indices for the four vegetation categories over snow-covered and snow-free ground. The background growth index for trees, GROWTH, is evaluated separately in subroutine CGROW. It varies from a value of 0 for dormant or leafless periods to 1 for fully-leafed periods, with a sixty-day transition between the two. When senescence begins, it is set instantaneously to -1 and thereafter increases over a sixty-day period back to 0. (The onset of spring budburst and fall senescence are triggered by near-zero values of the air temperature and the first soil layer temperature.) For needleleaf trees, the growth index GROWN is simply set to the absolute value of GROWTH. For broadleaf trees, the transition period is assumed to last thirty days instead of sixty, and so the growth index GROWB is set to the absolute value of double the value of GROWTH, with upper and lower limits of 1 and 0. Finally, the growth index of grasses is set to 1 all year round.

A branch in the code occurs next, depending on the value of the flag IHGT. If IHGT=0, the values of vegetation height calculated by CLASS are to be used. For trees and grass, the vegetation height under snow-free conditions is assumed to be constant year-round, and is calculated as 10 times the exponential of ZOLN, the logarithm of the maximum vegetation roughness length. For crops, this maximum height is multiplied by GROWA. If IHGT=1, vegetation heights specified by the user are utilized instead. This height H for each of the four vegetation categories is used to calculate the height HS over snow-covered areas. For needleleaf and broadleaf trees, HS is set to H. For crops and grass, HS is calculated by subtracting the snow depth ZSNOW from H, to account for the burying of short vegetation by snow.

If CLASS is being run uncoupled to CTEM, a second branch now occurs, depending on the value of the flag IPAI. If IPAI=0, the values of plant area index calculated by CLASS are to be used. For all four vegetation categories, the plant area index over snow-free ground, PAI, is determined by interpolating between the annual maximum and minimum plant area indices using the growth index. If IPAI=1, plant area index values specified by the user are utilized instead. For trees, the plant area index over snow- covered ground, PAIS, is set to PAI. For crops and grass, if H$>$0, PAIS is set to PAI scaled by the ratio of HS/H; otherwise, it is set to zero. Lastly, the leaf area indices for the four vegetation categories over snow-free ground, AIL, are determined from the PAI values. For needleleaf trees, AIL is estimated as 0.90 PAI; for broadleaf trees it is estimated as the excess PAI over the annual minimum value. For crops and grass AIL is assumed to be equal to PAI. (If CLASS is being run coupled to CTEM, the CTEM-generated values of PAI and AIL are used instead.)

In the 175 loop, the fractional coverage of the modelled area by each of the four vegetation categories is calculated, for snow-free (FCAN) and snow-covered ground (FCANS). For needleleaf and broadleaf trees, FCAN is set to the maximum coverage FCANMX of the vegetation category, scaled by the snow- free fraction of the modelled area, 1-FSNOW. For crops and grass, this calculation is modified for cases where the plant area index has been calculated as falling below a threshold value owing to growth stage or burying by snow. (If CLASS is being run coupled to C↩ TEM, this threshold value is set to 0.05; otherwise it is set to 1.) In such cases the vegetation coverage is assumed to become discontinuous, and so an additional multiplication by PAI is performed to produce a reduced value of FCAN, and PAI is reset to the threshold value. An identical procedure is followed to determine the FCANS values.

The areal fractions of each of the four CLASS subareas, vegetation over bare soil (FC), bare soil (FG), vegetation over snow (FCS) and snow (FGS) are then calculated, using the FCAN and FCANS values and FSNOW. Checks are carried out, and adjustments performed if necessary, to ensure that none of the four subareas is vanishingly small. The values of FSNOW and of the four FCANs and FCANSs are recalculated accordingly. Finally, checks are carried out to ensure that each of the four subareas is greater than zero, and that their sum is unity. If this is not the case, a call to abort is performed. In the last part of the 175 loop, the limiting ponding depth for surface water is determined for each of the four subareas. If the flag IWF is zero, indicating that lateral flow of water within the soil is to be neglected, these values are assigned as follows. If the index ISAND of the first soil layer is -3 or -4, indicating a rock surface or an ice sheet, the bare soil ponding limit, ZPLIMG, is set to 1 mm; otherwise ZPLIMG is set to 2 mm. If the fractional area of snow on bare soil is greater than zero, the subarea ponding limit ZPLMGS is set to the weighted average of ZPLIMG over the areas where snow has not buried vegetation and where it has buried crops, and to 3 mm over areas where it has buried grass; otherwise to zero. If the fractional area of canopy over bare soil is greater than zero, the subarea ponding depth ZPLIMC is set to the weighted average of 1 cm under trees and 3 mm under crops and grass; otherwise to zero. If the fractional area of canopy over snow is greater than zero, the subarea ponding depth ZPLMCS is also currently set to the weighted average of 1 cm under trees and 3 mm under crops and grass; otherwise to zero. Finally, if the flag IWF is greater than zero, indicating that lateral flow of soil water is being modelled, externally derived user-specified values of the ponding limit for the four subareas are assigned.

In loop 200, calculations are done related to the interception of water on vegetation. First, the plant area indices of the composite vegetation canopy over the snow-free and snow-covered subareas are calculated as weighted averages of the plant area indices of the four vegetation categories over each subarea. The liquid water interception capacity $W_{l,max}$ on each of the two subareas is calculated as $W_{l,max} = 0.20\Lambda_p$ where Λ_p is the plant area index of the composite canopy. This simple relation has been found to work well for a wide range of canopy types and precipitation events (see Verseghy et al, 1993). If either the average amount of liquid water on the canopy, RC↩ AN, or the total cancpy coverage, FC+FCS, is less than a small threshold value, the value of RCAN is stored in a residual water array RRESID, and RCAN is set to zero. Next the intercepted liquid water is partitioned between FC and FCS. First RCAN is re- evaluated as an average value over the canopy-covered area only, rather than over the whole modelled area. Then the intercepted liquid water amounts on vegetation over snow-free (RAICAN) and snow- covered areas (RAICNS) are calculated by making use of the relations $W_{L,0}/\Lambda_{p,0} = W_{L,s}/\Lambda_{p,s}$ and $W_l(X_0 + X_s) = W_{l,0}X_0 + W_{l,s}X_s$ where W_l is the liquid water on the canopy, X is the fractional area, and the subscripts 0 and s refer to snow-free and snow-covered areas respectively.

For snow interception on the canopy, a modified calculation of the plant area indices $\Lambda_{p,0}$ and $\Lambda_{p,s}$ is performed, assigning a weight of 0.7 to the plant area index of needleleaf trees, to account for the effect of needle clumping. The interception capacity for snow, $W_{f,max}$, is calculated following Bartlett et al. (2006), using a relation developed by Schmidt and Gluns (1991): $W_{f,max} = 6.0\Lambda_p[0.27 + 46.0\rho_{s,f}]$ where

In loops 250 and 275, calculations of the displacement height and the logarithms of the roughness lengths for heat and momentum are performed for the vegetated subareas. The displacement height d_i and the roughness length $z_{0,i}$ for the separate vegetation categories are obtained as simple ratios of the canopy height H: $d_i = 0.70H$ $z_{0,i} = 0.10H$

The averaged displacement height d over the vegetated subareas is only calculated if the flag IDISP has been set to 1. If IDISP = 0, this indicates that the atmospheric model is using a terrain-following coordinate system, and thus the displacement height is treated as part of the "terrain". If DISP = 1, d is calculated as a logarithmic average over the vegetation categories: $Xln(d) = \Sigma[X_i ln(d_i)]$ where X is the fractional coverage of the subarea. The averaged roughness length for momentum z_{0m} over the subarea is determined based on the assumption that averaging should be performed on the basis of the drag coefficient formulation. Thus, following Delage et al. (1999), and after Mason (1988): $X/ln^2(z_b/z_{0m}) = \Sigma[X_i/ln^2(z_b/z_{0i})]$

The averaged roughness length for heat z_{0e} over the subarea is calculated as a geometric mean over the vegetation

categories: $z_{0e}z_{0mX} = \Pi(z_{0i}^{2Xi})$

In loop 300, calculations of the logarithms of the roughness lengths for heat and momentum are performed for the bare ground and snow-covered subareas. Background values of $ln(z_{om})$ for soil, snow cover, ice sheets and urban areas are passed into the subroutine through common blocks. In CLASS, urban areas are treated very simply, as areas of bare soil with a high roughness length. The subarea values of $ln(z_{om})$ for bare soil and snow are therefore adjusted for the fractional coverage of urban area. Values for the ratio between the roughness lengths for momentum and heat for bare soil and snow are also passed in via common blocks. These are used to derive subarea values of $ln(z_{oe})$ from $ln(z_{om})$.

In loop 325, an adjustment is applied to $ln(z_{om})$ if the effect of terrain roughness needs to be taken into account. If the surface orographic roughness length is not vanishingly small, its logarithm, LZ0ORO, is calculated. If it is greater than the calculated logarithm of the roughness length for momentum of any of the subareas, these are reset to LZ0ORO.

In loop 350, the canopy mass is calculated as a weighted average over the vegetation categories, for canopy over bare soil (CMASSC) and over snow (CMASCS). (For crops over bare soil, the mass is adjusted according to the growth index; for crops and grass over snow, the mass is additionally adjusted to take into account burying by snow.) If IDISP = 0, indicating that the vegetation displacement height is part of the "terrain", the mass of air within the displacement height is normalized by the vegetation heat capacity and added to the canopy mass. If IZREF = 2, indicating that the bottom of the atmosphere is taken to lie at the local roughness length rather than at the ground surface, the mass of air within the roughness length is likewise normalized by the vegetation heat capacity and added to the canopy mass. The canopy heat capacities over bare soil (CHCAP) and over snow (CHCAPS) are evaluated from the respective values of canopy mass and of intercepted liquid water and snow. The aggregated canopy mass CMAI is recalculated, and is used to determine the change in internal energy of the canopy, HTCC, owing to growth or disappearance of the vegetation.

In the 450 and 500 loops, the fraction of plant roots in each soil layer is calculated. If CLASS is being run coupled to CTEM, the CTEM-derived values are assigned. Otherwise, for each vegetation category the rooting depth ZROOT is set to the background maximum value, except in the case of crops, for which it is set to the maximum scaled by GROWA. If the soil permeable depth is less than ZROOT, ZROOT is set to this depth instead. Values are then assigned in the matrix RMAT, which stores the fraction of roots in each vegetation category for each soil layer. According to Feddes et al. (1974), the fractional root volume R(z) below a depth z is well represented for many varieties of plants by the following exponential function: $R(z) = a_1 exp(-3.0z) + a_2$.

Making use of the boundary conditions $R(0) = 1$ and $R(z_r) = 0$, where z_r is the rooting depth ZROOT, it can be seen that the fraction of roots within a soil depth interval Δz can be obtained as the difference between R(z) evaluated at the top (z_T) and bottom (z_B) of the interval: $R(\Delta z) = [exp(-3.0z_T) - exp(-3.0z_B)]/[1 - exp(-3.0z_r)]$

The total fraction of roots in each soil layer, FROOT, can then be determined as a weighted average over the four vegetation categories.

In loop 450, a leaf boundary resistance parameter C_{rb} , incorporating the plant area indices of the four vegetation subareas, is also calculated for later use in subroutine TSOLVC: $C_{rb} = C_l \Lambda_{p,i}^{0.5}/0.75 \bullet [1 - exp(-0.75\Lambda_{p,i}^{0.5})]$ where C_l is a parameter that varies with the vegetation category. The aggregated value of C_{rb} is obtained as a weighted average over the four vegetation categories over bare ground and snow cover.

In loop 600, the sky view factor χ of the ground underlying the canopy is calculated for the vegetated subareas. The standard approach is to determine χ as an exponential function of the plant area index $\Lambda_p : \chi = exp[-c\Lambda_p]$ where c is a constant depending on the vegetation category. The subarea values of χ are obtained as weighted averages over the four vegetation categories.

In the 650 loop, the fraction of the total transpiration of water by plants that is extracted from each soil layer is determined. This is done by weighting the values of FROOT calculated above by the relative soil moisture suction in each layer: $(\Psi_w - \Psi_i)/(\Psi_w - \Psi_{sat})$ where Ψ_i , the soil moisture suction in the layer, is obtained as $\Psi_i = \Psi_{sat}(\theta_{l,i}/\theta_p)^{-b}$ In these equations Ψ_w is the soil moisture suction at the wilting point, Ψ_{sat} is the suction at saturation, $\theta_{l,i}$ is the volumetric liquid water content of the soil layer, θ_p is the pore volume, and b is an empirical parameter developed by Clapp and Hornberger (1978). The layer values of FROOT are then re-normalized so that their sum adds up to unity. In this loop, the representative soil moisture suction PSIGND is also calculated for later use in the vegetation stomatal resistance formulation, as the minimum value of Ψ_i and Ψ_w over all the soil layers.

Finally, in loop 800 the aggregated canopy plant area indices PAICAN and PAICNS are set back to their original

values, from the modified values used for the snow interception calculations above; and if CLASS is being run coupled with CTEM, a set of CTEM-related calculations is performed.

9.3 balcar.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Carbon Balance Subroutine.

Functions/Subroutines

- subroutine balcar (gleafmas, stemmass, rootmass, bleafmas, litrmass, soilcmas, ntchlveg, ntchsveg, ntchrveg, tltrleaf, tltrstem, tltrroot, glcaemls, blcaemls, stcaemls, rtcaemls, ltrcemls, ltresveg, scresveg, humtrsvg, pglfmass, pblfmass, pstemass, protmass, plitmass, psocmass, vgbiomas, repro_cost, pvgbioms, gavgltms, pgavltms, gavgscms, pgavscms, galtcels, repro_cost_g, npp, autores, hetrores, gpp, nep, litres, socres, dstcemls, nbp, litrfall, humiftrs, il1, il2)

9.3.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Carbon Balance Subroutine.

unless mentioned all pools are in kg c/m2 unless mentioned all fluxes are in units of u-mol co2/m2.sec

9.3.2 Function/Subroutine Documentation

9.3.2.1 subroutine balcar (real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* real, dimension(ilg,icc) *bleafmas,* real, dimension(ilg,icc+1) *litrmass,* real, dimension(ilg,icc+1) *soilcmas,* real, dimension(ilg,icc) *ntchlveg,* real, dimension(ilg,icc) *ntchsveg,* real, dimension(ilg,icc) *ntchrveg,* real, dimension(ilg,icc) *tltrleaf,* real, dimension(ilg,icc) *tltrstem,* real, dimension(ilg,icc) *tltrroot,* real, dimension(ilg,icc) *glcaemls,* real, dimension(ilg,icc) *blcaemls,* real, dimension(ilg,icc) *stcaemls,* real, dimension(ilg,icc) *rtcaemls,* real, dimension(ilg,icc) *ltrcemls,* real, dimension(ilg,icc+1) *ltresveg,* real, dimension(ilg,icc+1) *scresveg,* real, dimension(ilg,icc+1) *humtrsvg,* real, dimension(ilg,icc) *pglfmass,* real, dimension(ilg,icc) *pblfmass,* real, dimension(ilg,icc) *pstemass,* real, dimension(ilg,icc) *protmass,* real, dimension(ilg,icc+1) *plitmass,* real, dimension(ilg,icc+1) *psocmass,* real, dimension(ilg) *vgbiomas,* real, dimension(ilg,icc) *repro_cost,* real, dimension(ilg) *pvgbioms,* real, dimension(ilg) *gavgltms,* real, dimension(ilg) *pgavltms,* real, dimension(ilg) *gavgscms,* real, dimension(ilg) *pgavscms,* real, dimension(ilg) *galtcels,* real, dimension(ilg) *repro_cost_g,* real, dimension(ilg) *npp,* real, dimension(ilg) *autores,* real, dimension(ilg) *hetrores,* real, dimension(ilg) *gpp,* real, dimension(ilg) *nep,* real, dimension(ilg) *litres,* real, dimension(ilg) *socres,* real, dimension(ilg) *dstcemls,* real, dimension(ilg) *nbp,* real, dimension(ilg) *litrfall,* real, dimension(ilg) *humiftrs,* integer *il1,* integer *il2*)

Parameters

il1	other variables: il1=1
il2	other variables: il2=ilg
stemmass	pools (after being updated): stem mass for each of the 9 ctem pfts
rootmass	pools (after being updated): root mass for each of the 9 ctem pfts
gleafmas	pools (after being updated): green leaf mass for each of the 9 ctem pfts
bleafmas	pools (after being updated): brown leaf mass for each of the 9 ctem pfts
litrmass	pools (after being updated): litter mass over the 9 pfts and the bare fraction of the grid cell
soilcmas	pools (after being updated): soil carbon mass over the 9 pfts and the bare fraction of the grid cell

ntchlveg	fluxes for each pft: net change in leaf biomass
ntchsveg	fluxes for each pft: net change in stem biomass
ntchrveg	fluxes for each pft: net change in root biomass the net change is the difference between allocation and autotrophic respiratory fluxes
tltrleaf	fluxes for each pft: total leaf litter falling rate
tltrstem	fluxes for each pft: total stem litter falling rate
tltrroot	fluxes for each pft: total root litter falling rate
glcaemls	fluxes for each pft: carbon emission losses mainly due to fire: green leaf carbon emission losses
blcaemls	fluxes for each pft: carbon emission losses mainly due to fire: brown leaf carbon emission losses
stcaemls	fluxes for each pft: carbon emission losses mainly due to fire: stem carbon emission losses
rtcaemls	fluxes for each pft: carbon emission losses mainly due to fire: root carbon emission losses
ltrcemls	fluxes for each pft: carbon emission losses mainly due to fire: litter carbon emission losses
ltresveg	fluxes for each pft: litter respiration for each pft + bare fraction
scresveg	fluxes for each pft: soil c respiration for each pft + bare fraction
humtrsvg	fluxes for each pft: humification for each pft + bare fraction
pglfmass	pools (before being updated): previous green leaf mass
pblfmass	pools (before being updated): previous brown leaf mass
pstemass	pools (before being updated): previous stem mass
protmass	pools (before being updated): previous root mass
plitmass	pools (before being updated): previous litter mass
psocmass	pools (before being updated): previous soil c mass
npp	grid averaged flux: net primary productivity
vgbiomas	pools (after being updated): grid averaged pools: vegetation biomass
pvgbioms	pools (before being updated): grid average pools: previous vegetation biomass
gavgltms	pools (after being updated): grid averaged pools: litter mass
pgavltms	pools (before being updated): grid average pools: previous litter mass
gavgscms	pools (after being updated): grid averaged pools: soil carbon mass
pgavscms	pools (before being updated): grid average pools: previous soil c mass
autores	grid averaged flux: autotrophic respiration
hetrores	grid averaged flux: heterotrophic respiration
gpp	grid averaged flux: gross primary productivity
nep	grid averaged flux: net primary productivity
litres	grid averaged flux: litter respiration
socres	grid averaged flux: soil carbon respiration
dstcemls	grid averaged flux: carbon emission losses due to disturbance, mainly fire
nbp	grid averaged flux: net biome productivity
litrfall	grid averaged flux: combined (leaves, stem, and root) total litter fall rate
humiftrs	grid averaged flux: humification
repro_cost	pools (after being updated): amount of C transferred to litter due to reproductive tissues
galtcels	grid averaged flux: carbon emission losses from litter
repro_cost_g	grid averaged flux: amount of C used to generate reproductive tissues

to check c budget we go through each pool for each vegetation type.

green and brown leaves

stem

root

litter over all pfts

litter over the bare fraction

soil carbon over the bare fraction

grid averaged fluxes must also balance

vegetation biomass

litter

soil carbon

9.4 bio2str.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Biomass To Structural Attributes Conversion Subroutine.

Functions/Subroutines

- subroutine bio2str (gleafmas, bleafmas, stemmass, rootmass,il1,il2,fcancmx,zbotw,delzw, nol2pfts,soildpth,

9.4.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Biomass To Structural Attributes Conversion Subroutine.

The time-varying biomass in the leaves ($C_{\rm L}$), stem ($C_{\rm S}$) and root ($C_{\rm R}$) components is used to calculate the structural attributes of vegetation for the energy and water balance calculations by CLASS.

Leaf biomass is converted to LAI using specific leaf area (SLA, $m^2\,(kg\,C)^{-1}$), which itself is assumed to be a function of leaf lifespan ($\tau_{\rm L}$; see also ctem_params.f90)

$$SLA = \gamma_{\rm L}\tau_{\rm L}^{-0.5} LAI = C_{\rm L} SLA$$

where $\gamma_{\rm L}$ is a constant with value equal to $25\,m^2\,(kg\,C)^{-1}\,yr^{0.5}$.

The vegetation height (H; m) is calculated for tree, crop and grass PFTs as

$$H = \{\min\left(10.0C_{\rm S}^{0.385}, 45\right) trees (C_{\rm S}+C_{\rm L})^{0.385} crops 3.5(C_{L,g}+0.55\,C_{L,b})^{0.5} grasses,$$

where $C_{L,g}$ is the green leaf biomass and $C_{L,b}$ is the brown leaf biomass that is scaled by 0.55 to reduce its contribution to the plant height. CTEM explicitly tracks brown leaf mass for grass PFTs. The turnover of green grass leaves, due to normal aging or stress from drought and/or cold, does not contribute to litter pool directly as the leaves first turn brown. The brown leaves themselves turnover to litter relatively rapidly $(\tau_{L,b} = 0.1\,\tau_{\rm L})$.

CTEM dynamically simulates root distribution and depth in soil following [8]. The root distribution takes an exponential form and roots grow and deepen with increasing root biomass. The cumulative root fraction at depth z is given by

$$f_{\rm R}(z) = 1 - \exp(-\iota z).$$

Rooting depth ($d_{\rm R}$; m), which is defined to be the depth containing $99\,\%$ of the root mass, is found by setting z equal to $d_{\rm R}$ and $f_{\rm R} = 0.99$, which yields

$$d_{\rm R} = \frac{-\ln(1-f_{\rm R})}{\iota} = \frac{-\ln(1-0.99)}{\iota} = \frac{4.605}{\iota}.$$

The parameter ι that describes the exponential root distribution is calculated as

$$\iota = \bar{\iota}\left(\frac{\overline{C_{\rm R}}}{C_{\rm R}}\right)^{0.8},$$

where $\bar{\iota}$ represents the PFT-specific mean root distribution profile parameter and $\overline{C_{\rm R}}$ the average root biomass derived from [24] (see also ctem_params.f90). Equation (iota}) yields a lower (higher) value of ι than $\bar{\iota}$ when root biomass $C_{\rm R}$ is higher (lower) than the PFT-specific mean root biomass $\overline{C_{\rm R}}$, resulting in a deeper (shallower) root profile than the mean root profile.

The rooting depth $d_{\rm R}$ is checked to ensure it does not exceed the soil depth. If so, $d_{\rm R}$ is set to the soil depth and ι is recalculated as $\iota = 4.605/d_{\rm R}$ (see Eq. rootterm1} for derivation of 4.605 term). The new value of ι is used to determine the root distribution profile adjusted to the shallower depth. Finally, the root distribution profile is used to calculate fraction of roots in each of the model's soil layers.

9.4.2 Function/Subroutine Documentation

9.4.2.1 subroutine bio2str (real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *bleafmas,* real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* integer *il1,* integer *il2,* real, dimension(ilg,icc) *fcancmx,* real, dimension(ilg,ignd) *zbotw,* real, dimension(ilg,ignd) *delzw,* integer, dimension(ican) *nol2pfts,* real, dimension(ilg) *soildpth*)

Parameters

il1	input: il1=1
il2	input: il2=ilg
nol2pfts	input: number of level 2 pfts
gleafmas	input: green or live leaf mass in kg c/m2, for the 9 pfts
bleafmas	input: brown or dead leaf mass in kg c/m2, for the 9 pfts
stemmass	input: stem biomass in kg c/m2, for the 9 pfts
rootmass	input: root biomass in kg c/m2, for the 9 pfts
fcancmx	input: max. fractional coverages of ctem's 9 pfts. this is different from fcanc and fcancs (which may vary with snow depth). fcancmx doesn't change, unless of course its changed by land use change or dynamic vegetation.
delzw	input: thicknesses of the 3 soil layers
zbotw	input: bottom of soil layers
soildpth	input: soil depth (m)

Constants and parameters are located in ctem_params.f90

class' original root parameterization has deeper roots than ctem's default values based on literature. in the coupled model this leads to lower evapotranspiration (et) values. an option is provided here to deepen roots, but this will also increase photosynthesis and vegetation biomass slightly, due to more access to soil water. so while use of deeper roots is desirable in the coupled global model, one may decide to use ctem's default parameterizarion for stand alone simulations, and set deeproots to .false.

—— 1. conversion of leaf biomass into leaf area index -——

find specific leaf area (sla, m2/kg) using leaf life span

convert leaf biomass into lai. brown leaves could have less lai than the green leaves for the same leaf mass. for now we assume sla of brown leaves is fracbofg times that of green leaves.

also find stem area index as a function of stem biomass

plant area index is sum of green and brown leaf area indices and stem area index

make class see some minimum pai, otherwise it runs into numerical problems

get fcancmx weighted leaf area index for use by class needle leaf evg + dcd = total needle leaf broad leaf evg + dcd cld + dcd dry = total broad leaf crop c3 + c4 = total crop grass c3 + c4 = total grass also add brown lai. note that although green + brown lai is to be used by class for energy and water balance calculations, stomatal conductance estimated by the photosynthesis subroutine is only based on green lai. that is although both green+brown leaves intercept water and light, only the green portion photosynthesizes. also lump stem and plant area indices for class' 4 pfts

for crops and grasses set the minimum lai to a small number, other wise class will never run tsolvc and thus phtsyn and ctem will not be able to grow crops or grasses.

—— 2. conversion of stem biomass into roughness length -——

class uses log of roughness length (zoln) as an input parameter. when vegetation grows and dies as per ctem, then zoln is provided by ctem.

1. convert stem biomass into vegetation height for trees and crops, and convert leaf biomass into vegetation height for grass
2. convert vegetation height into roughness length & take its log
3. lump this for ctem's 9 pfts into class' 4 pfts

—— 3. estimating fraction of roots in each soil layer for —— —— ctem's each vegetation type, using root biomass ——

estimate parameter b of variable root profile parameterization

use b to estimate 99% rooting depth

if estimated rooting depth is greater than soil depth, or the maximum rooting depth then adjust rooting depth and parameter alpha

also find "a" (parameter determining root profile). this is the "a" which depends on time varying root biomass

using parameter "a" we can find fraction of roots in each soil layer just like class

if rootdepth is shallower than the bottom of current layer and is deeper than bottom of the previous top layer

if rootdepth is shallower than the bottom of 2nd layer

if rootdepth is shallower than the bottom of 3rd layer or even the deeper layer (ignd>3)

make sure all fractions (of roots in each layer) add to one.

lump rmatctem(i,9,ignd) into rmatc(i,4,ignd) for use by class

-————————— 4. calculate storage lai ---—————————

need a minimum slai to be able to grow from scratch. consider this as model seeds.

— 5. calculate total vegetation biomass for each ctem pft, and – ----———— canopy mass for each class pft ------————

since class uses canopy mass and not total vegetation biomass as an input, we find canopy mass as a sum of stem and leaf mass, for each class pft, i.e. only above ground biomass.

if there is no vegetation canopy mass will be abszero. this should essentially mean more bare ground, but since we are not changing fractional coverages at present, we pass a minimum canopy mass to class so that it doesn't run into numerical problems.

— 6. calculate albedo for class' 4 pfts based on specified -— ----— albedos of ctem 9 pfts and their fractional coveraes --—

9.5 CANADD.f File Reference

Purpose: Calculate canopy interception of rainfall and snowfall, and determine rainfall/snowfall rates at ground surface as a result of throughfall and unloading.

Functions/Subroutines

- subroutine canadd (IWATER, R, TR, S, TS, RAICAN, SNOCAN, TCAN, CHCAP, HTCC, ROFC, ROVG, P↩ CPN, PCPG, FI, FSVF, CWLCAP, CWFCAP, CMASS, RHOSNI, TSURX, RDRIP, SDRIP, ILG, IL1, IL2, JL)

9.5.1 Detailed Description

Purpose: Calculate canopy interception of rainfall and snowfall, and determine rainfall/snowfall rates at ground surface as a result of throughfall and unloading.

9.5.2 Function/Subroutine Documentation

9.5.2.1 subroutine canadd (integer *IWATER,* real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg) *S,* real, dimension (ilg) *TS,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension (ilg) *TCAN,* real, dimension (ilg) *CHCAP,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *ROFC,* real, dimension (ilg) *ROVG,* real, dimension (ilg) *PCPN,* real, dimension (ilg) *PCPG,* real, dimension (ilg) *FI,* real, dimension (ilg) *FSVF,* real, dimension(ilg) *CWLCAP,* real, dimension(ilg) *CWFCAP,* real, dimension (ilg) *CMASS,* real, dimension(ilg) *RHOSNI,* real, dimension (ilg) *TSURX,* real, dimension (ilg) *RDRIP,* real, dimension (ilg) *SDRIP,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

r	Rainfall rate over subarea in question $[ms^{-1}]$
tr	Temperature of rainfall [C]
s	Snowfall rate over subarea in question $[ms^{-1}]$
ts	Temperature of snowfall [C]
raican	Intercepted liquid water stored on the canopy $[kgm^{-2}](W_{l,c})$
snocan	Intercepted frozen water stored on the canopy $[kgm^{-2}](W_{f,c})$
tcan	Temperature of vegetation canopy $[K](T_c)$
chcap	Heat capacity of vegetation canopy $[Jm^{-2}K^{-1}](C_c)$
htcc	Internal energy change of canopy due to changes in temperature and/or mass $[Wm^{-2}](I_c)$
rofc	Liquid/frozen water runoff from vegetation $[kgm^{-2}s^{-1}]$
rovg	Liquid/frozen water runoff from vegetation to ground surface $[kgm^{-2}s^{-1}]$
pcpn	Precipitation incident on snow pack $[kgm^{-2}s^{-1}]$
pcpg	Precipitation incident on ground $[kgm^{-2}s^{-1}]$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
fsvf	Sky view factor of surface under vegetation canopy [ ]
cwlcap	Interception storage capacity of vegetation for liquid water $[kgm^{-2}]$
cwfcap	Interception storage capacity of vegetation for frozen water $[kgm^{-2}](W_{f,max})$
cmass	Mass of vegetation canopy $[kgm^{-2}]$
rhosni	Density of fresh snow $[kgm^{-3}]$
tsurx	Ground or snow surface temperature of subarea [K]

The calculations in this subroutine are performed if the rainfall or snowfall rates over the modelled area are greater than zero, or if the intercepted liquid water RAICAN or frozen water SNOCAN is greater than zero (to allow for unloading). The throughfall of rainfall or snowfall incident on the canopy, RTHRU or STHRU, is calculated from FSVF, the canopy gap fraction or sky view factor. The remaining rainfall or snowfall is assigned to interception, as RINT and SINT. The resulting temperature of liquid water on the canopy, TRCAN, is calculated as a weighted average of RAICAN at the canopy temperature TCAN, and RINT at the rainfall temperature TR. The resulting temperature of frozen water on the canopy, TSCAN, is calculated as a weighted average of SNOCAN at the canopy temperature TCAN, and SINT at the snowfall temperature TS.

Calculations are now done to ascertain whether the total liquid water on the canopy exceeds the liquid water interception capacity CWLCAP. If such is the case, this excess is assigned to RDRIP, water dripping off the canopy. The rainfall rate reaching the surface under the canopy is calculated as RDRIP + RTHRU, and the temperature of this water flux is calculated as a weighted average of RDRIP at a temperature of TRCAN, and RTHRU at the original rainfall temperature TR. The remaining intercepted water becomes CWLCAP. Otherwise, the rainfall rate reaching the surface under the canopy is set to RTHRU, and the liquid water on the canopy RAICAN is augmented by RINT.

Interception and unloading of snow on the canopy is calculated using a more complex method. The amount of snow intercepted during a snowfall event over a time step, $\Delta W_{f,i}$, or SLOAD, is obtained from the initial intercepted snow amount $W_{f,c}$ and the interception capacity $W_{f,max}$, following Hedstrom and Pomeroy (1998), as:

$$\Delta W_{f,i} = (W_{f,max} \breve{} W_{f,c})[1 \breve{} exp(-S_{int}/W_{f,max})]$$

where S_{int} is the amount of snow incident on the canopy during the time step. The amount of snow not stored by interception, SWXCES, is calculated as SINT – SLOAD. Between and during precipitation events, snow is unloaded from the canopy through wind gusts and snow densification. These effects of these processes are estimated using an empirical exponential relationship for the snow unloading rate $W_{f,u}$ or SNUNLD, again following Hedstrom and Pomeroy (1998):

$$W_{f,u} = W_{f,c} + \Delta W_{f,i} exp(-U\Delta t)$$

where U is a snow unloading coefficient, assigned a value of $0.1d^{-1}$ or $1.157 * 10^{-6}s{-}1$. The sum of SWXCES and SNUNLD is assigned to SDRIP, the snow or frozen water falling off the canopy. The snowfall rate reaching the surface under the canopy is calculated as SDRIP + STHRU, and the temperature of this water flux is calculated as a weighted average of SDRIP at a temperature of TSCAN, and STHRU at the original snowfall temperature TS. The frozen water stored on the canopy is recalculated as SNOCAN + SINT – SWXCES - SNUNLD. Otherwise, the snowfall rate reaching the surface under the canopy is set to STHRU, and SNOCAN is augmented by SINT.

In the final section of the subroutine, the initial heat capacity and temperature of the canopy are saved in temporary variables. The new canopy heat capacity is calculated as a weighted average over the specific heats of the liquid

and frozen water stored on the canopy and the canopy mass. The canopy temperature is calculated as a weighted average over the stored liquid and frozen water at the updated temperatures TRCAN and TSCAN, and the vegetation mass at the original temperature TCAN. Then the change in internal energy I_c of the vegetation canopy as a result of the water movement above is calculated as the difference in I_c before and after these processes:

$$\Delta I_c = X_i \Delta [C_c T_c] / \Delta t$$

where C_c represents the canopy heat capacity, T_c the canopy temperature, Δt the length of the time step, and X_i the fractional coverage of the subarea under consideration relative to the modelled area.

Finally, the rainfall and snowfall temperatures are converted to degrees C. (In the absence of precipitation, both are set equal to the surface temperature of the subarea to avoid floating point errors in later subroutines.) For subareas with a snow cover (IWATER = 2), the water running off the canopy and the precipitation incident on the snow pack are updated using RDRIP and SDRIP. For subareas without snow cover (IWATER = 1), the water running off the canopy is updated using RDRIP and SDRIP, the precipitation incident on the snow pack is augmented by SDRIP, and the precipitation incident on bare ground is augmented by RDRIP.

9.6 CANALB.f File Reference

Purpose: Calculate vegetation albedos, transmissivities and stomatal resistances.

Functions/Subroutines

- subroutine canalb (ALVSCN, ALIRCN, ALVSCS, ALIRCS, TRVSCN, TRIRCN, TRVSCS, TRIRCS, RC, R↩ CS, ALVSC, ALIRC, RSMIN, QA50, VPDA, VPDB, PSIGA, PSIGB, FC, FCS, FSNOW, FSNOWC, FSNOCS, FCAN, FCANS, PAI, PAIS, AIL, PSIGND, FCLOUD, COSZS, QSWINV, VPD, TA, ACVDAT, ACIDAT, ALVS↩ GC, ALIRGC, ALVSSC, ALIRSC, ILG, IL1, IL2, JL, IC, ICP1, IG, IALC, CXTEFF, TRVS, TRIR, RCACC, RCG, RCV, RCT, GC)

9.6.1 Detailed Description

Purpose: Calculate vegetation albedos, transmissivities and stomatal resistances.

9.6.2 Function/Subroutine Documentation

9.6.2.1 **subroutine canalb (real, dimension(ilg) *ALVSCN,* real, dimension(ilg) *ALIRCN,* real, dimension(ilg) *ALVSCS,* real, dimension(ilg) *ALIRCS,* real, dimension(ilg) *TRVSCN,* real, dimension(ilg) *TRIRCN,* real, dimension(ilg) *TRVSCS,* real, dimension(ilg) *TRIRCS,* real, dimension (ilg) *RC,* real, dimension (ilg) *RCS,* real, dimension (ilg,icp1) *ALVSC,* real, dimension (ilg,icp1) *ALIRC,* real, dimension (ilg,ic) *RSMIN,* real, dimension (ilg,ic) *QA50,* real, dimension (ilg,ic) *VPDA,* real, dimension (ilg,ic) *VPDB,* real, dimension (ilg,ic) *PSIGA,* real, dimension (ilg,ic) *PSIGB,* real, dimension (ilg) *FC,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FSNOW,* real, dimension(ilg) *FSNOWC,* real, dimension(ilg) *FSNOCS,* real, dimension (ilg,ic) *FCAN,* real, dimension (ilg,ic) *FCANS,* real, dimension (ilg,ic) *PAI,* real, dimension (ilg,ic) *PAIS,* real, dimension (ilg,ic) *AIL,* real, dimension(ilg) *PSIGND,* real, dimension(ilg) *FCLOUD,* real, dimension (ilg) *COSZS,* real, dimension(ilg) *QSWINV,* real, dimension (ilg) *VPD,* real, dimension (ilg) *TA,* real, dimension(ilg,ic) *ACVDAT,* real, dimension(ilg,ic) *ACIDAT,* real, dimension(ilg) *ALVSGC,* real, dimension(ilg) *ALIRGC,* real, dimension(ilg) *ALVSSC,* real, dimension(ilg) *ALIRSC,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IC,* integer *ICP1,* integer *IG,* integer *IALC,* real, dimension(ilg,ic) *CXTEFF,* real, dimension (ilg) *TRVS,* real, dimension (ilg) *TRIR,* real, dimension (ilg,ic) *RCACC,* real, dimension (ilg,ic) *RCG,* real, dimension (ilg,ic) *RCV,* real, dimension (ilg) *RCT,* real, dimension (ilg) *GC*)**

Parameters

alvscn	Visible albedo of vegetation over bare ground [ ]
alircn	Near-IR albedo of vegetation over bare ground [ ]
alvscs	Visible albedo of vegetation over snow [ ]
alircs	Near-IR albedo of vegetation over snow [ ]
trvscn	Visible transmissivity of vegetation over bare ground $[](\tau_c)$
trircn	Near-IR transmissivity of vegetation over bare ground $[](\tau_c)$
trvscs	Visible transmissivity of vegetation over snow $[](\tau_c)$
trircs	Near-IR transmissivity of vegetation over snow $[](\tau_c)$
rc	Stomatal resistance of vegetation over bare ground $[sm^{-1}](r_c)$
rcs	Stomatal resistance of vegetation over snow $[sm^{-1}]$
alvsc	Background average visible albedo of vegetation category [ ]
alirc	Background average near-infrared albedo of vegetation category [ ]
rsmin	Minimum stomatal resistance of vegetation category $[sm^{-1}](r_{s,min})$
qa50	Reference value of incoming shortwave radiation for vegetation category (used in stomatal resistance calculation) $[Wm^{-2}](K\downarrow_{1/2})$
vpda	Vapour pressure deficit coefficient for vegetation category (used in stomatal resistance calculation) $[](c_{v1})$
vpdb	Vapour pressure deficit coefficient for vegetation category (used in stomatal resistance calculation) $[](c_{v2})$
psiga	Soil moisture suction coefficient for vegetation category (used in stomatal resistance calculation) $[](c_{\Psi 1})$
psigb	Soil moisture suction coefficient for vegetation category (used in stomatal resistance calculation) $[](c_{\Psi 2})$
fcan	Fractional coverage of vegetation category over bare ground $[](X_i)$
fcans	Fractional coverage of vegetation category over snow [ ]
pai	Plant area index of vegetation category over bare ground $[](\Lambda_p)$
pais	Plant area index of vegetation category over snow $[](\Lambda_p)$
acvdat	Optional user-specified value of canopy visible albedo to override CLASS-calculated value [ ]
acidat	Optional user-specified value of canopy near-infrared albedo to override CLASS- calculated value [ ]
ail	Leaf area index of vegetation category over bare ground $[](\Lambda)$
fc	Fractional coverage of canopy over bare ground [ ]
fcs	Fractional coverage of canopy over snow [ ]
fsnow	Diagnosed fractional snow coverage [ ]
fsnowc	Fractional coverage of canopy by frozen water over snow-free subarea [ ]
fsnocs	Fractional coverage of canopy by frozen water over snow-covered subarea [ ]
psignd	Minimum liquid moisture suction in soil layers $[m](\Psi_s)$
fcloud	Fractional cloud cover $[](X_i)$
coszs	Cosine of solar zenith angle [ ]
qswinv	Visible radiation incident on horizontal surface $[Wm^{-2}](K)$
vpd	Vapour pressure deficit of air $[mb]\Delta e$
ta	Air temperature at reference height $[K](T_a)$
alvsgc	Visible/near-IR albedo of bare ground under vegetation [ ]
alirgc	Visible/near-IR albedo of bare ground under vegetation [ ]
alvssc	Visible/near-IR albedo of snow under vegetation [ ]
alirsc	Visible/near-IR albedo of snow under vegetation [ ]

The transmissivity τ_c of a vegetation canopy to shortwave radiation is obtained by applying a form of Beer's law of radiation transfer in non-scattering media: $\tau_c = exp[-\kappa\Lambda_p]$ where κ is the canopy extinction coefficient and Λ_p is the plant area index. The extinction coefficient is calculated after Goudriaan (1988) as $\kappa = \epsilon O/cos(Z)$ where epsilon is a correction factor less than or equal to 1, accounting for forward-scattering of radiation and non-random leaf distributions (i.e. clumping), O represents the mean projected leaf area fraction perpendicular to the incoming radiation, and Z is the zenith angle of the incoming radiation. For crops, grass and needleleaf trees, the distribution of leaf angles is assumed to be spherical, and thus O = 0.5. For broadleaf trees, the preferred leaf orientation tends to be horizontal; thus O = cos(Z).

In the clear-sky case, incoming shortwave radiation is dominated by the direct beam; thus, the transmissivity for clear skies $\tau_{c,0}$ is evaluated by simply setting Z to Z_s, the solar zenith angle. An extensive search of the literature

for values of epsilon appropriate to the four vegetation categories yielded the following results for the extinction coefficient:

$\kappa = 0.3/cos(Z_s)(needleleaftrees)$ $\kappa = 0.4(broadleaftrees, fullcanopy)$ $\kappa = 0.4/cos(Z_s)(broadleaftrees, leafless)$
$\kappa = 0.4/cos(Z_s)(cropsandgrass)$

In the visible range of the shortwave spectrum, scattering is less important because of high leaf absorptivities. The following results were obtained for visible radiation:

$\kappa = 0.4/cosZ_s(needleleaftrees)$ $\kappa = 0.7(broadleaftrees, fullcanopy)$ $\kappa = 0.4/cosZ_s(broadleaftrees, leafless)$
$\kappa = 0.5/cosZ_s(cropsandgrass)$

In the case of overcast skies, the hemispherical distribution of incoming shortwave radiation is modelled using the generally accepted "standard overcast sky" distribution (e.g. Steven and Unsworth, 1980), where the shortwave radiation D(Z) emanating from a sky zenith angle Z is approximated as

$D(Z) = D(0)[(1 + 1.23cosZ)/2.23].$

Integration of the cloudy-sky transmissivity $tau_{c,cloudy}$ over the sky hemisphere is performed using a simple weighting function proposed by Goudriaan (1988):

$\tau_{c,cloudy} = 0.3\tau_c(Z = 15^o) + 0.5\tau_c(Z = 45^o) + 0.2\tau_c(Z = 75^o)$

The albedo α_c of a vegetation canopy is, like the transmissivity, dependent in principle on the zenith angle Z of the incoming radiation and the mean projected leaf area fraction O. However, in practice the observed diurnal and seasonal variation of vegetation albedo tends to be slight for closed canopies or for radiation zenith angles larger than a few degrees. Since the former is generally true in the tropics and the latter in the extratropics, the diurnal variation of visible and near-infrared canopy albedo is neglected in CLASS.

Corrections are applied to the average albedos to account for the effects of intercepted snow on the canopy and incomplete canopy closure. The presence of snow on the canopy will increase the albedo; this increase will be larger in the visible range of the spectrum, since the visible albedo of vegetation is typically very small. Making use of data presented by Leonard and Eschner (1968), the visible albedo of snow-covered vegetation is set to 0.17. The near-infrared albedo, being larger in magnitude, will be affected to a lesser degree, and is therefore set to the background near-infrared albedo plus 0.04.

If the canopy closure is incomplete, a fraction chi of the ground or snow under the canopy will be visible through gaps in it. This fraction is equal to the sky view factor of the underlying surface, which is calculated using an equation analogous to that for the canopy transmissivity, as an exponential function of the plant area index Λ_p:

$\chi = exp[-c\Lambda_p]$

where c is a constant depending on the vegetation category. The overall albedo α_T is calculated as a weighted average of the canopy albedo and the underlying ground or snow albedo (the latter weighted by the canopy transmissivity, to account for the decreased downwelling shortwave radiation at the surface):

$\alpha_T = (1\breve{}\chi)\alpha_c + \chi\tau_c\alpha_0$

where $alpha_0$ is the albedo of the surface under the canopy.

At the beginning of the subroutine, a series of work arrays is initialized to zero. Then the transmissivity and albedo of each vegetation category are calculated in turn, first over bare soil and then over a snow pack. After each of the latter sets of calculations, consistency checks are carried out to ensure that the calculated albedos and transmissivities are not less than 0 or greater than 1, and that the transmissivity is not greater than 90% of the non-reflected radiation.

The clear-sky and cloudy-sky transmissivities in the visible range are calculated first, and then the clear-sky and cloudy-sky transmissivities in the total shortwave spectrum, using the extinction coefficient equations above. (For broadleaf trees, the minimum of the full-canopy and leafless calculated values is used.) The overall transmissivity in the visible range at the current time step is obtained as an average of the clear and cloudy-sky values, weighted according to the fractional cloud cover. The effective overall extinction coefficient CXTEFF in the visible range is computed for use later on in the stomatal resistance calculation. The overall transmissivity for the entire shortwave spectrum at the current time step is similarly obtained as an average of the clear and cloudy-sky values, weighted according to the fractional cloud cover. The near-infrared transmissivity is calculated from the total and visible transmissivities, making use of the assumption that visible and near-infrared radiation each typically comprise 50% of the incoming shortwave radiation. Lastly, the aggregated visible and near-infrared transmissivities for the bulk canopy are incremented using the current values weighted by the fractional coverage of the vegetation category. At

the end, the sum is normalized by the total fractional vegetation coverage in the subarea.

For the albedos, first the sky view factor and the near-infrared albedo of snow-covered vegetation are calculated. Next, a branch directs further processing depending on the value of the flag IALC. If IALC = 0, the CLASS-calculated visible and near-infrared vegetation albedos are used. These albedos are corrected for the presence of intercepted snow, using an average of the snow- covered and background snow-free albedos weighted according to FSN↩ OWC over snow-free, and FSNOCS over snow-covered subareas (representing the ratio of the amount of snow present on the canopy relative to the interception capacity). The correction for incomplete canopy closure is applied to the resulting visible and near- infrared albedos as described above. If IALC = 1, user- specified albedos for the vegetation canopy are used instead of the CLASS values. These are assumed to incorporate the effects of incomplete canopy closure, but not of the presence of snow. Thus the CLASS values for the albedo of snow-covered vegetation and the albedo of snow under the canopy are still used in the averaging. Finally, the aggregated visible and near-infrared albedos for the bulk canopy are incremented using the current values weighted by the fractional coverage of the vegetation category. At the end, the sum is normalized by the total fractional vegetation coverage in the subarea.

In the final section, the stomatal resistance r_c of the vegetation canopy is determined. Based on the analysis of Schulze et al. (1995), the unstressed stomatal resistance $r_{c,u}$ for a given vegetation category can be calculated as a function of the incoming visible shortwave radiation $K\downarrow$:

$r_{c,u} = r_{s,min}\kappa_e / ln[K\downarrow + K\downarrow_{1/2} / \kappa_e / K\downarrow exp(-\kappa_e \Lambda) + K\downarrow_{1/2} / \kappa_e]$

where $r_{s,min}$ is the minimum stomatal resistance for the vegetation category, κ_e is the extinction coeffient for visible radiation (CXTEFF above), Λ is the leaf area index, and $K\downarrow_{1/2}$ is the value of $K\downarrow$ at which $r_{c,u} = 2r_{s,min}$.

Suboptimum environmental conditions for transpiration may lead to stresses on the plant, causing the stomatal resistance to be greater than its unstressed value. The effects of these stresses are modelled by defining functions of the air temperature T_a, the air vapour pressure deficit Δe, and the soil moisture suction Ψ_s. These functions are used to derive $r_{c,i}$ of each vegetation category on the basis of $r_{c,u,i}$:

$r_{c,i} = f(T_a) f(\Delta_e) f(\Psi_s) * r_{c,u,i}$

The air temperature function $f(T_a)$ has a value of 1 for temperatures between 5^oC and 40^oC, and an arbitrary large value of 250 for temperatures less than -5^oC and greater than 50^oC. Between these points it varies in a linear fashion. For the vapour pressure deficit function $f(\Delta e)$, two alternate forms are provided, after Oren et al. (1999) and Wu et al. (2000) respectively:

$f(\Delta e) = [(\Delta e/10.0)^{cv2}]/c_{v1}$ and $f(\Delta e) = 1/[exp(-c_{v1}\Delta e/10.0)]$

where c_{v1} and c_{v2} are parameters depending on the vegetation category. If c_{v2} is greater than zero, the first form is used; if not, the second form is used. The soil moisture suction function $f(\Psi_s)$ is expressed, following Choudhury and Idso (1985) and Fisher et al. (1981), as:

$f(\Psi_s) = 1 + (\Psi_s / c_{\Psi 1})^{c\Psi 2}$

where $c_{\Psi 1}$ and $c_{\Psi 2}$ are parameters depending on the vegetation category. Finally, the aggregated stomatal resistance for the canopy over the bare ground subarea is obtained as a weighted average over the vegetation categories. (It is assumed that transpiration is suppressed when snow is present under the canopy, so rc for this subarea is set to a large number). Since, following the electrical analogy, resistances act in parallel, the aggregated resistance for the subarea of canopy over bare ground is obtained as an average of inverses:

$X/r_c = \Sigma(X_i / r_{c,i})$

The calculations described above pertaining to stomatal resistances are performed in loops 850, 900 and 950. In the 850 loop, $f(T_a)$ is evaluated. In the 900 loop, for each vegetation category in turn, $f(\Delta e)$, $f(\Psi_s)$ and $r_{c,i}$ are determined. $r_{c,i}$ is assigned upper and lower limits of 5000 and 10 sm^{-1} respectively, and the accumulated stomatal resistance for the canopy is incremented by $X_i/r_{c,i}$. In loop 950, FRMAX, the maximum value of F↩ ROOT, the fraction of transpiration apportioned to each soil layer, is determined. If FRMAX is vanishingly small, transpiration is suppressed by setting the stomatal resistances over the vegetated subareas to a large number. If the incoming visible radiation is small, the stomatal resistances are likewise set to a large number. Otherwise, the stomatal resistance of vegetation over snow is set to a large number, and the normalized, aggregated stomatal resistance of vegetation over soil is obtained by inverting the accumulated value.

9.7 CANVAP.f File Reference

Purpose: Update liquid and frozen water stores on canopy and in soil in response to calculated sublimation, evaporation and transpiration rates.

Functions/Subroutines

- subroutine canvap (EVAP, SUBL, RAICAN, SNOCAN, TCAN, THLIQ, TBAR, ZSNOW, WLOST, CHCAP, Q↩ FCF, QFCL, QFN, QFC, HTCC, HTCS, HTC, FI, CMASS, TSNOW, HCPSNO, RHOSNO, FROOT, THPOR, THLMIN, DELZW, EVLOST, RLOST, IROOT, IG, ILG, IL1, IL2, JL, N)

9.7.1 Detailed Description

Purpose: Update liquid and frozen water stores on canopy and in soil in response to calculated sublimation, evaporation and transpiration rates.

9.7.2 Function/Subroutine Documentation

9.7.2.1 subroutine canvap (real, dimension (ilg) *EVAP,* real, dimension (ilg) *SUBL,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension (ilg) *TCAN,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *WLOST,* real, dimension (ilg) *CHCAP,* real, dimension (ilg) *QFCF,* real, dimension (ilg) *QFCL,* real, dimension (ilg) *QFN,* real, dimension (ilg,ig) *QFC,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *FI,* real, dimension (ilg) *CMASS,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *HCPSNO,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg,ig) *FROOT,* real, dimension(ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *DELZW,* real, dimension(ilg) *EVLOST,* real, dimension (ilg) *RLOST,* integer, dimension (ilg) *IROOT,* integer *IG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

thliq	Volumetric liquid water content of soil
tbar	Temperature of soil layer $[K](T_g)$
qfc	Transpired water removed from soil layer $[kgm^{-2}s^{-1}]$
htc	Internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}](I_g)$
evap	Evapotranspiration rate from vegetation canopy $[ms^{-1}]$
subl	Calculated sublimation rate from vegetation canopy $[ms^{-1}]$
raican	Intercepted liquid water stored on the canopy $[kgm^{-2}]$
snocan	Intercepted frozen water stored on the canopy $[kgm^{-2}]$
tcan	Temperature of vegetation canopy $[K](T_c)$
zsnow	Depth of snow pack $[m](z_g)$
wlost	Residual amount of water that cannot be supplied by surface stores $[kgm^{-2}]$
chcap	Heat capacity of vegetation canopy $[Jm^{-2}K^{-1}](C_c)$
qfcf	Sublimation from frozen water in canopy interception store $[kgm^{-2}s^{-1}]$
qfcl	Evaporation from liquid water in canopy interception store $[kgm^{-2}s^{-1}]$
qfn	Sublimation from snow pack $[kgm^{-2}s^{-1}]$
htcc	Internal energy change of canopy due to changes in temperature and/or mass $[Wm^{-2}](I_c)$
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}](I_s)$
froot	Fractional contribution of soil layer to transpiration [ ]

thpor	Pore volume in soil layer $[m^3m^{-3}]$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}]$
delzw	Permeable depth of soil layer $[m](\Delta z_{g,w})$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
cmass	Mass of vegetation canopy $[kgm^{-2}]$
tsnow	Temperature of the snow pack [C]
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}](C_s)$
rhosno	Density of snow pack $[kgm^{-3}]$

The calculated fluxes of liquid and frozen water from the canopy to the overlying air, obtained as outputs of WPREP, are applied to the liquid and frozen intercepted water stores on the vegetation canopy, and to the liquid and frozen moisture stores at the surface and in the soil. Since there may not be sufficient water in one or more of these stores to sustain the calculated rate over the whole time step, a hierarchy of operations is followed as described below. The change of internal energy HTC in the canopy, snow and soil layers as a result of these processes is calculated as the difference in HTC between the beginning and end of the subroutine:

$$\Delta I_c = X_i \Delta(C_c T_c)/\Delta t$$

$$\Delta I_s = X_i \Delta(C_s T_s z_s)/\Delta t$$

where the C terms represent volumetric heat capacities and the T terms temperatures of the canopy and snow pack, Δt is the length of the time step, z_s the snow depth, and X_i the fractional coverage of the subarea under consideration relative to the modelled area. For the soil layers, since only the liquid water content is affected by these calculations, the change in internal energy of each layer is calculated from the change in liquid water content θ_l as:

$$\Delta I_g = X_i C_w \Delta z_{g,w} \Delta(T_g \theta_l)/\Delta t$$

Sublimation is addressed first. The predicted mass of sublimated water SLOST is calculated and compared to the frozen water in the canopy interception store, SNOCAN. If SLOST $\leq$ SNOCAN, all of the sublimated water is subtracted from SNOCAN. If not, the excess sublimation is calculated as SLOST-SNOCAN, QFCF is corrected for the canopy sublimation difference, and SNOCAN is set to zero. Next, the new value of SLOST is compared to the snowpack mass, calculated as ZSNOW$*$RHOSNO. If SLOST $\leq$ ZSNOW$*$RHOSNO, all of the remaining sublimated water is taken from the snow pack, and QFN is modified to reflect this loss. Otherwise, the excess sublimation is calculated as SLOST - ZSNOW$*$RHOSNO, QFN is adjusted accordingly, and ZSNOW is set to zero. There now remain no further frozen moisture stores from which sublimated water can be taken (frozen water in the soil is assumed to be immobile), so the remaining energy that had been assigned to sublimation is assigned to canopy evaporation instead, and QFCL is duly recalculated. This means, however, that a small imbalance will arise in the water budget owing to the difference between the latent heats of sublimation and evaporation. This imbalance is assigned to the housekeeping variable WLOST.

Now canopy evaporation is addressed. It is assumed that all intercepted liquid water evaporates before transpiration begins, since there is a canopy stomatal resistance associated with transpiration and there is none associated with evaporation. The predicted mass of evaporated water RLOST is calculated and compared to the liquid water in the canopy interception store, RAICAN. If RLOST $\leq$ RAICAN, all of the evaporated water is subtracted from R↩ AICAN. If not, the excess evaporation is calculated as RLOST - RAICAN, and QFCL is corrected for the canopy evaporation difference. This excess evaporation is now treated as transpiration. An initial check is done by referring to the diagnostic flag IROOT, which was set to 1 at the beginning of the subroutine if there was water available for transpiration in any of the soil layers. If IROOT is zero, no transpiration can occur and the excess evaporation is stored in the temporary variable EVLOST.

9.8 CGROW.f File Reference

Purpose: Evaluate growth index used in calculating vegetation parameters for forests.

Functions/Subroutines

- subroutine cgrow (GROWTH, TBAR, TA, FC, FCS, ILG, IG, IL1, IL2, JL)

9.8.1 Detailed Description

Purpose: Evaluate growth index used in calculating vegetation parameters for forests.

9.8.2 Function/Subroutine Documentation

9.8.2.1 subroutine cgrow (real, dimension(ilg) *GROWTH,* real, dimension (ilg,ig) *TBAR,* real, dimension(ilg) *TA,* real, dimension(ilg) *FC,* real, dimension(ilg) *FCS,* integer *ILG,* integer *IG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

growth	Tree growth index [ ]
tbar	Temperature of soil layers [K]
ta	Air temperature [K]
fc	Fractional coverage of vegetation without snow on modelled area [ ]
fcs	Fractional coverage of vegetation with underlying snow pack on modelled area [ ]

The growth index that is calculated here varies from a value of 1 for periods when the trees are mature and fully leaved, to 0 for dormant and leafless periods, with a linear transition between the two. The transition periods are assumed to last for sixty days; therefore during these periods the growth index is incremented by $\Delta t/5.184x10^6$ where Δt is the time step in seconds.

The transition period from dormant to fully leafed is triggered when both the air temperature and the temperature of the first soil layer are above 2 C. If one of these conditions is not met afterwards, the growth index is reset back to 0. Increments are added continuously thereafter until the index reaches 1.

The transition from fully leafed to dormant is triggered when either the air temperature or the temperature of the first soil layer falls below 2 C. When this first happens at the end of the fully-leafed period, the growth index is set instantaneously to -1 and increments are continuously added from that point until the index reaches 0.

The absolute value of this growth index is utilized for performing calculations of various forest vegetation parameters in subroutine APREP; thus its shape as used there is that of a symmetrical trapezoidal function.

9.9 CHKWAT.f File Reference

Purpose: Check for closure of surface water budget, and for unphysical values of certain variables.

Functions/Subroutines

- subroutine chkwat (ISFC, PCPR, EVAP, RUNOFF, WLOST, RAICAN, SNOCAN, RAICNI, SNOCNI, ZPOND, ZPONDI, THLIQ, THICE, THLIQI, THICEI, ZSNOW, RHOSNO, XSNOW, SNOWI, WSNOW, WSNOWI, FCS, FGS, FI, BAL, THPOR, THLMIN, DELZW, ISAND, IG, ILG, IL1, IL2, JL, N)

9.9.1 Detailed Description

Purpose: Check for closure of surface water budget, and for unphysical values of certain variables.

9.9.2 Function/Subroutine Documentation

9.9.2.1 subroutine chkwat (integer *ISFC,* real, dimension (ilg) *PCPR,* real, dimension (ilg) *EVAP,* real, dimension(ilg) *RUNOFF,* real, dimension (ilg) *WLOST,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension(ilg) *RAICNI,* real, dimension(ilg) *SNOCNI,* real, dimension (ilg) *ZPOND,* real, dimension(ilg) *ZPONDI,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension(ilg,ig) *THLIQI,* real, dimension(ilg,ig) *THICEI,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *XSNOW,* real, dimension (ilg) *SNOWI,* real, dimension (ilg) *WSNOW,* real, dimension(ilg) *WSNOWI,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FGS,* real, dimension (ilg) *FI,* real, dimension (ilg) *BAL,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *DELZW,* integer, dimension (ilg,ig) *ISAND,* integer *IG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

pcpr	Precipitation rate over modelled subarea $[kgm^{-2}s^{-1}]$
evap	Evapotranspiration rate over modelled subarea $[kgm^{-2}s^{-1}]$
runoff	Total runoff over modelled subarea [m]
wlost	Residual amount of water that cannot be supplied by surface stores $[kgm^{-2}]$
raican	Intercepted liquid water on canopy at end of time step $[kgm^{-2}]$
snocan	Intercepted frozen water on canopy at end of time step $[kgm^{-2}]$
raicni	Intercepted liquid water on canopy at beginning of time step $[kgm^{-2}]$
snocni	Intercepted frozen water on canopy at beginning of time step $[kgm^{-2}]$
zpond	Depth of ponded water on ground at end of time step [m]
zpondi	Depth of ponded water on ground at beginning of time step [m]
zsnow	Depth of snow pack [m]
rhosno	Density of snow pack $[kgm^{-3}]$
xsnow	Switch to indicate presence of snow cover [ ]
snowi	Snow pack mass at beginning of time step $[kgm^{-2}]$
wsnow	Liquid water content of snow pack at end of time step $[kgm^{-2}]$
wsnowi	Liquid water content of snow pack at beginning of time step $[kgm^{-2}]$
fcs	Fractional coverage of canopy over snow on modelled area [ ]
fgs	Fractional coverage of snow over bare ground on modelled area [ ]
fi	Fractional coverage of subarea in question on modelled area [ ]
thliq	Volumetric liquid water content of soil layers at end of time step $[m^3m^{-3}]$
thice	Volumetric frozen water content of soil layers at end of time step $[m^3m^{-3}]$
thliqi	Volumetric frozen water content of soil layers at beginning of time step $[m^3m^{-3}]$
thicei	Volumetric frozen water content of soil layers at beginning of time step $[m^3m^{-3}]$
thpor	Pore volume in soil layer $[m^3m^{-3}]$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}]$
delzw	Permeable depth of soil layer [m]

This subroutine is called from CLASSW to perform water balance checks for each of the four subareas. The flag ISFC indicates which subarea is being addressed: ISFC=1 for vegetation over snow, ISFC=2 for snow over bare ground, ISFC=3 for vegetation over bare ground, and ISFC=4 for bare ground. If a problem is discovered, a flag is set to the index of the modelled area, and a call to XIT is performed with an error message. Checks for unphysical values of certain water balance variables are performed against an accuracy limit ACCLMT, currently set to $1x10^{-3}kgm^{-2}$ or m^3m^{-3}. The overall water balance of the subarea is checked against an accuracy limit BALLMT, currently set to $1x10^{-1}kgm^{-2}$. (These values reflect expected roundoff errors associated with 32-bit computation.)

In loop 100, for canopy-covered subareas, the intercepted rain RAICAN and snow SNOCAN are checked to ensure that if they are negative, they are vanishingly small. A similar check is done for the runoff.

In the 150 loop, for all areas that are not continental ice sheets (ISAND=-4), the liquid water content in each soil layer is checked to ensure that it is not larger than the pore volume and that it is not smaller than the minimum liquid water content (except for rock layers). The ice content is similarly checked to ensure that the sum of it, converted to an equivalent liquid water content, plus the minimum water content, is not greater than the pore volume (except for rock layers). It is also checked to ensure that if it is negative, it is vanishingly small.

Finally, in loop 300, the overall water balance BAL is calculated and compared to BALLMT. BAL is evaluated as the residual of the precipitation, the evaporation, the runoff, the water loss term WLOST, the change in canopy intercepted liquid and frozen water (for vegetation-covered areas), the change in surface ponded water, the change in snow pack and snow liquid water content (for snow-covered areas), and the changes in the soil layer liquid and frozen water contents. If the absolute value of BAL is greater than BALLMT, a flag is set, all of the terms entering BAL are printed out, and a call to XIT is performed.

9.10 CLASSA.f File Reference

Purpose: Organize calculation of radiation-related and other surface parameters.

Functions/Subroutines

- subroutine classa (FC, FG, FCS, FGS, ALVSCN, ALIRCN, ALVSG, ALIRG, ALVSCS, ALIRCS, ALVSSN, ALIRSN, ALVSGC, ALIRGC, ALVSSC, ALIRSC, TRVSCN, TRIRCN, TRVSCS, TRIRCS, FSVF, FSVFS, RAICAN, RAICNS, SNOCAN, SNOCNS, FRAINC, FSNOWC, FRAICS, FSNOCS, DISP, DISPS, ZOMLNC, ZOMLCS, ZOELNC, ZOELCS, ZOMLNG, ZOMLNS, ZOELNG, ZOELNS, CHCAP, CHCAPS, CMASSC, CMASCS, CWLCAP, CWFCAP, CWLCPS, CWFCPS, RC, RCS, RBCOEF, FROOT, FROOTS, ZPLIMC, ZPLIMG, ZPLMCS, ZPLMGS, ZSNOW, WSNOW, ALVS, ALIR, HTCC, HTCS, HTC, ALTG, ALSNO, TRSNOWC, TRSNOWG, WTRC, WTRS, WTRG, CMAI, FSNOW, FCANMX, ZOLN, ALVSC, ALIRC, PAIMAX, PAIMIN, CWGTMX, ZRTMAX, RSMIN, QA50, VPDA, VPDB, PSIGA, PSIGB, PAIDAT, HGTDAT, ACVDAT, ACIDAT, ASVDAT, ASIDAT, AGVDAT, AGIDAT, ALGWET, ALGDRY, ALGWV, ALGWN, ALGDV, ALGDN, THLIQ, THICE, TBAR, RCAN, SNCAN, TCAN, GROWTH, SNO, TSNOW, RHOSNO, ALBSNO, ZBLEND, Z0ORO, SNOLIM, ZPLMG0, ZPLMS0, FCLOUD, TA, VPD, RHOAIR, COSZS, FSDB, FSFB, REFSNO, BCSNO, QSWINV, RADJ, DLON, RHOSNI, DELZ, DELZW, ZBOTW, THPOR, THLMIN, PSISAT, BI, PSIWLT, HCPS, ISAND, FCANCMX, ICTEM, ctem_on, RMATC, ZOLNC, CMASVEGC, AILC, PAIC, L2MAX, NOL2PFTS, SLAIC, AILCG, AILCGS, FCANC, FCANCS, IDAY, ILG, IL1, IL2, NBS, JL, N, IC, ICP1, IG, IDISP, IZREF, IWF, IPAI, IHGT, IALC, IALS, IALG, ISNOALB, IGRALB, ALVSCTM, ALIRCTM)

9.10.1 Detailed Description

Purpose: Organize calculation of radiation-related and other surface parameters.

9.10.2 Function/Subroutine Documentation

9.10.2.1 subroutine classa (real, dimension (ilg) *FC,* real, dimension (ilg) *FG,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FGS,* real, dimension(ilg) *ALVSCN,* real, dimension(ilg) *ALIRCN,* real, dimension (ilg) *ALVSG,* real, dimension (ilg) *ALIRG,* real, dimension(ilg) *ALVSCS,* real, dimension(ilg) *ALIRCS,* real, dimension(ilg) *ALVSSN,* real, dimension(ilg) *ALIRSN,* real, dimension(ilg) *ALVSGC,* real, dimension(ilg) *ALIRGC,* real, dimension(ilg) *ALVSSC,* real, dimension(ilg) *ALIRSC,* real, dimension(ilg) *TRVSCN,* real, dimension(ilg) *TRIRCN,* real, dimension(ilg) *TRVSCS,* real, dimension(ilg) *TRIRCS,* real, dimension (ilg) *FSVF,* real, dimension (ilg) *FSVFS,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *RAICNS,* real, dimension(ilg) *SNOCAN,* real, dimension(ilg) *SNOCNS,* real, dimension(ilg) *FRAINC,* real, dimension(ilg) *FSNOWC,* real, dimension(ilg) *FRAICS,* real, dimension(ilg) *FSNOCS,* real, dimension (ilg) *DISP,* real, dimension (ilg) *DISPS,* real, dimension(ilg) *ZOMLNC,* real, dimension(ilg) *ZOMLCS,* real, dimension(ilg) *ZOELNC,* real, dimension(ilg) *ZOELCS,* real, dimension(ilg) *ZOMLNG,* real, dimension(ilg) *ZOMLNS,* real, dimension(ilg) *ZOELNG,* real, dimension(ilg) *ZOELNS,* real, dimension (ilg) *CHCAP,* real, dimension(ilg) *CHCAPS,* real, dimension(ilg) *CMASSC,* real, dimension(ilg) *CMASCS,* real, dimension(ilg) *CWLCAP,* real, dimension(ilg) *CWFCAP,* real, dimension(ilg) *CWLCPS,* real, dimension(ilg) *CWFCPS,* real, dimension (ilg) *RC,* real, dimension (ilg) *RCS,* real, dimension(ilg) *RBCOEF,* real, dimension (ilg,ig) *FROOT,* real, dimension(ilg,ig) *FROOTS,* real, dimension(ilg) *ZPLIMC,* real, dimension(ilg) *ZPLIMG,* real, dimension(ilg) *ZPLMCS,* real, dimension(ilg) *ZPLMGS,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *WSNOW,* real, dimension (ilg) *ALVS,* real, dimension (ilg) *ALIR,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension(ilg,nbs) *ALTG,* real, dimension(ilg,nbs) *ALSNO,* real, dimension(ilg) *TRSNOWC,* real, dimension(ilg,nbs) *TRSNOWG,* real, dimension (ilg) *WTRC,* real, dimension (ilg) *WTRS,* real, dimension (ilg) *WTRG,* real, dimension (ilg) *CMAI,* real, dimension (ilg) *FSNOW,* real, dimension(ilg,icp1) *FCANMX,* real, dimension (ilg,icp1) *ZOLN,* real, dimension (ilg,icp1) *ALVSC,* real, dimension (ilg,icp1) *ALIRC,* real, dimension(ilg,ic) *PAIMAX,* real, dimension(ilg,ic) *PAIMIN,* real, dimension(ilg,ic) *CWGTMX,* real, dimension(ilg,ic) *ZRTMAX,* real, dimension (ilg,ic) *RSMIN,* real, dimension (ilg,ic) *QA50,* real, dimension (ilg,ic) *VPDA,* real, dimension (ilg,ic) *VPDB,* real, dimension (ilg,ic) *PSIGA,* real, dimension (ilg,ic) *PSIGB,* real, dimension(ilg,ic) *PAIDAT,* real, dimension(ilg,ic) *HGTDAT,* real, dimension(ilg,ic) *ACVDAT,* real, dimension(ilg,ic) *ACIDAT,* real, dimension(ilg) *ASVDAT,* real, dimension(ilg) *ASIDAT,* real, dimension(ilg) *AGVDAT,* real, dimension(ilg) *AGIDAT,* real, dimension(ilg) *ALGWET,* real, dimension(ilg) *ALGDRY,* real, dimension(ilg) *ALGWV,* real, dimension(ilg) *ALGWN,* real, dimension(ilg) *ALGDV,* real, dimension(ilg) *ALGDN,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg) *RCAN,* real, dimension (ilg) *SNCAN,* real, dimension (ilg) *TCAN,* real, dimension(ilg) *GROWTH,* real, dimension (ilg) *SNO,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension(ilg) *ALBSNO,* real, dimension(ilg) *ZBLEND,* real, dimension (ilg) *Z0ORO,* real, dimension(ilg) *SNOLIM,* real, dimension(ilg) *ZPLMG0,* real, dimension(ilg) *ZPLMS0,* real, dimension(ilg) *FCLOUD,* real, dimension (ilg) *TA,* real, dimension (ilg) *VPD,* real, dimension(ilg) *RHOAIR,* real, dimension (ilg) *COSZS,* real, dimension(ilg,nbs) *FSDB,* real, dimension(ilg,nbs) *FSFB,* real, dimension(ilg) *REFSNO,* real, dimension(ilg) *BCSNO,* real, dimension(ilg) *QSWINV,* real, dimension (ilg) *RADJ,* real, dimension (ilg) *DLON,* real, dimension(ilg) *RHOSNI,* real, dimension (ig) *DELZ,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *ZBOTW,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLMIN,* real, dimension(ilg,ig) *PSISAT,* real, dimension (ilg,ig) *BI,* real, dimension(ilg,ig) *PSIWLT,* real, dimension (ilg,ig) *HCPS,* integer, dimension (ilg,ig) *ISAND,* real, dimension(ilg,ictem) *FCANCMX,* integer *ICTEM,* logical *ctem_on,* real, dimension(ilg,ic,ig) *RMATC,* real, dimension(ilg,ic) *ZOLNC,* real, dimension(ilg,ic) *CMASVEGC,* real, dimension (ilg,ic) *AILC,* real, dimension (ilg,ic) *PAIC,* integer *L2MAX,* integer, dimension(ic) *NOL2PFTS,* real, dimension(ilg,ic) *SLAIC,* real, dimension (ilg,ictem) *AILCG,* real, dimension(ilg,ictem) *AILCGS,* real, dimension(ilg,ictem) *FCANC,* real, dimension(ilg,ictem) *FCANCS,* integer *IDAY,* integer *ILG,* integer *IL1,* integer *IL2,* integer *NBS,* integer *JL,* integer *N,* integer *IC,* integer *ICP1,* integer *IG,* integer *IDISP,* integer *IZREF,* integer *IWF,* integer *IPAI,* integer *IHGT,* integer *IALC,* integer *IALS,* integer *IALG,* integer *ISNOALB,* integer *IGRALB,* real, dimension(ilg,ic) *ALVSCTM,* real, dimension(ilg,ic) *ALIRCTM*)

Parameters

fc	Subarea fractional coverage of modelled area [ ]
fg	Subarea fractional coverage of modelled area [ ]
fcs	Subarea fractional coverage of modelled area [ ]
fgs	Subarea fractional coverage of modelled area [ ]
alvscn	Visible albedo of vegetation over bare ground [ ]
alircn	Near-IR albedo of vegetation over bare ground [ ]
alvsg	Visible albedo of open bare ground [ ]
alirg	Near-IR albedo of open bare ground [ ]
alvscs	Visible albedo of vegetation over snow [ ]
alircs	Near-IR albedo of vegetation over snow [ ]
alvssn	Visible albedo of open snow cover [ ]
alirsn	Near-IR albedo of open snow cover [ ]
alvsgc	Visible albedo of bare ground under vegetation [ ]
alirgc	Near-IR albedo of bare ground under vegetation [ ]
alvssc	Visible albedo of snow under vegetation [ ]
alirsc	Near-IR albedo of snow under vegetation [ ]
trvscn	Visible transmissivity of vegetation over bare ground [ ]
trircn	Near-IR transmissivity of vegetation over bare ground [ ]
trvscs	Visible transmissivity of vegetation over snow [ ]
trircs	Near-IR transmissivity of vegetation over snow [ ]
fsvf	Sky view factor for bare ground under canopy [ ]
fsvfs	Sky view factor for snow under canopy [ ]
raican	Intercepted liquid water stored on canopy over bare ground $[kgm^{-2}]$
raicns	Intercepted liquid water stored on canopy over snow $[kgm^{-2}]$
snocan	Intercepted frozen water stored on canopy over bare soil $[kgm^{-2}]$
snocns	Intercepted frozen water stored on canopy over snow $[kgm^{-2}]$
frainc	Fractional coverage of canopy by liquid water over snow-free subarea [ ]
fsnowc	Fractional coverage of canopy by frozen water over snow-free subarea [ ]
fraics	Fractional coverage of canopy by liquid water over snow-covered subarea [ ]
fsnocs	Fractional coverage of canopy by frozen water over snow-covered subarea [ ]
disp	Displacement height of vegetation over bare ground [m]
disps	Displacement height of vegetation over snow [m]
zomlnc	Logarithm of roughness length for momentum of vegetation over bare ground [ ]
zomlcs	Logarithm of roughness length for momentum of vegetation over snow [ ]
zoelnc	Logarithm of roughness length for heat of vegetation over bare ground [ ]
zoelcs	Logarithm of roughness length for heat of vegetation over snow [ ]
zomlng	Logarithm of roughness length for momentum of bare ground [ ]
zomlns	Logarithm of roughness length for momentum of snow [ ]
zoelng	Logarithm of roughness length for heat of bare ground [ ]
zoelns	Logarithm of roughness length for heat of snow [ ]
chcap	Heat capacity of canopy over bare ground $[Jm^{-2}K^{-1}]$
chcaps	Heat capacity of canopy over snow $[Jm^{-2}K^{-1}]$
cmassc	Mass of canopy over bare ground $[kgm^{-2}]$
cmascs	Mass of canopy over snow $[kgm^{-2}]$

cwlcap	Storage capacity of canopy over bare ground for liquid water $[kgm^{-2}]$
cwfcap	Storage capacity of canopy over bare ground for frozen water $[kgm^{-2}]$
cwlcps	Storage capacity of canopy over snow for liquid water $[kgm^{-2}]$
cwfcps	Storage capacity of canopy over snow for frozen water $[kgm^{-2}]$
rc	Stomatal resistance of vegetation over bare ground $[sm^{-1}]$
rcs	Stomatal resistance of vegetation over snow $[sm^{-1}]$
zplimc	Maximum water ponding depth for ground under canopy [m]
zplimg	Maximum water ponding depth for bare ground [m]
zplmcs	Maximum water ponding depth for ground under snow under canopy [m]
zplmgs	Maximum water ponding depth for ground under snow [m]
rbcoef	Parameter for calculation of leaf boundary resistance
trsnowc	Short-wave transmissivity of snow pack [ ]
zsnow	Depth of snow pack $[m](z_s)$
wsnow	Liquid water content of snow pack $[kgm^{-2}]$
alvs	Diagnosed total visible albedo of land surface [ ]
alir	Diagnosed total near-infrared albedo of land surface [ ]
htcc	Diagnosed internal energy change of vegetation canopy due to conduction and/or change in mass $[Wm^{-2}]$
htcs	Diagnosed internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}]$
wtrc	Diagnosed residual water transferred off the vegetation canopy $[kgm^{-2}s^{-1}]$
wtrs	Diagnosed residual water transferred into or out of the snow pack $[kgm^{-2}s^{-1}]$
wtrg	Diagnosed residual water transferred into or out of the soil $[kgm^{-2}s^{-1}]$
cmai	Aggregated mass of vegetation canopy $[kgm^{-2}]$
fsnow	Diagnosed fractional snow coverage [ ]
froot	Fraction of total transpiration contributed by soil layer [ ]
froots	Fraction of total transpiration contributed by snow-covered soil layer [ ]
htc	Diagnosed internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}]$
fcanmx	Maximum fractional coverage of modelled area by vegetation category [ ]
zoln	Natural logarithm of maximum roughness length of vegetation category [ ]
alvsc	Background average visible albedo of vegetation category [ ]
alirc	Background average near-infrared albedo of vegetation category [ ]
paimax	Maximum plant area index of vegetation category [ ]
paimin	Minimum plant area index of vegetation category [ ]
cwgtmx	Maximum canopy mass for vegetation category $[kgm^{-2}]$
zrtmax	Maximum rooting depth of vegetation category [m]
rsmin	Minimum stomatal resistance of vegetation category $[sm^{-1}]$
qa50	Reference value of incoming shortwave radiation for vegetation category (used in stomatal resistance calculation) $[Wm^{-2}]$
vpda	Vapour pressure deficit coefficient for vegetation category (used in stomatal resistance calculation) [ ]
vpdb	Vapour pressure deficit coefficient for vegetation category (used in stomatal resistance calculation) [ ]
psiga	Soil moisture suction coefficient for vegetation category (used in stomatal resistance calculation) [ ]
psigb	Soil moisture suction coefficient for vegetation category (used in stomatal resistance calculation) [ ]

paidat	Optional user-specified value of plant area indices of vegetation categories to override CLASS-calculated values [ ]
hgtdat	Optional user-specified values of height of vegetation categories to override CLASS- calculated values [m]
acvdat	Optional user-specified value of canopy visible albedo to override CLASS-calculated value [ ]
acidat	Optional user-specified value of canopy near-infrared albedo to override CLASS- calculated value [ ]
thliq	Volumetric liquid water content of soil layers $[m^3 m^{-3}]$
thice	Volumetric frozen water content of soil layers $[m^3 m^{-3}]$
tbar	Temperature of soil layers [K]
asvdat	Optional user-specified value of snow visible albedo to override CLASS-calculated value [ ]
asidat	Optional user-specified value of snow near- infrared albedo to override CLASS-calculated value [ ]
agvdat	Optional user-specified value of ground visible albedo to override CLASS-calculated value [ ]
agidat	Optional user-specified value of ground near- infrared albedo to override CLASS-calculated value [ ]
algwv	Reference albedo for saturated soil (visible) [ ]
algwn	Reference albedo for saturated soil (NIR) [ ]
algdv	Reference albedo for dry soil (visible) [ ]
algdn	Reference albedo for dry soil (NIR) [ ]
algwet	Reference albedo for saturated soil [ ]
algdry	Reference albedo for dry soil [ ]
rhosni	Density of fresh snow $[kgm^{-3}]$
z0oro	Orographic roughness length [m]
rcan	Intercepted liquid water stored on canopy $[kgm^{-2}]$
sncan	Intercepted frozen water stored on canopy $[kgm^{-2}]$
tcan	Vegetation canopy temperature [K]
growth	Vegetation growth index [ ]
sno	Mass of snow pack $[kgm^{-2}](W_s)$
tsnow	Snowpack temperature [K]
rhosno	Density of snow $[kgm^{-3}](\rho_s)$
albsno	Snow albedo [ ]
fcloud	Fractional cloud cover [ ]
ta	Air temperature at reference height [K]
vpd	Vapour pressure deficit of air [mb]
rhoair	Density of air $[kgm^{-3}]$
coszs	Cosine of solar zenith angle [ ]
qswinv	Visible radiation incident on horizontal surface $[Wm^{-2}]$
dlon	Longitude of grid cell (east of Greenwich) [degrees]
zblend	Atmospheric blending height for surface roughness length averaging [m]
snolim	Limiting snow depth below which coverage is $<$ 100% $[m](z_{s,lim})$
zplmg0	Maximum water ponding depth for snow-free subareas (user-specified when running MESH code) [m]
zplms0	Maximum water ponding depth for snow-covered subareas (user-specified when running MESH code) [m]
radj	Latitude of grid cell (positive north of equator) [rad]
delzw	Permeable thickness of soil layer [m]
zbotw	Depth to permeable bottom of soil layer [m]
thpor	Pore volume in soil layer $[m^3 m^{-3}]$

thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3 m^{-3}]$
psisat	Soil moisture suction at saturation [m]
bi	Clapp and Hornberger empirical "b" parameter [ ]
psiwlt	Soil moisture suction at wilting point [m]
hcps	Volumetric heat capacity of soil particles $[Jm^{-3}]$
isand	Sand content flag
delz	Soil layer thickness [m]

In the first loop, the depth of snow z_s is calculated from the snow mass W_s and density ϱ_s as: $z_s = W_s/\varrho_s$.

If the calculated value of z_s is less than the limiting snow depth $z_{s,lim}$, the snow cover is deemed to be discontinuous. The fractional snow coverage X_s of the modelled area is evaluated as $X_s = z_s/z_{s,lim}$ and the snow depth is reset to $z_{s,lim}$. The water content of the snow pack is corrected according to the new snow fractional area.

The subarea albedo and transmissivity arrays (for canopy, bare ground, canopy over snow and snow over bare ground) are next initialized to zero, and the four CLASSA subsidiary subroutines are called in turn: APREP to evaluate various model parameters for the four subareas, SNOALBA to calculate the snow albedo and transmissivity, GRALB to calculate the ground surface albedo, and CANALB to calculate the canopy albedo, transmissivity and stomatal resistance. Finally, the overall visible and near- infrared albedos for the modelled area are determined as weighted averages over the four subareas.

9.11 CLASSB.f File Reference

Purpose: Assign thermal and hydraulic properties to soil layers based on sand/clay content, or soil type. Also calculate permeable thickness of soil layers, and wet and dry surface albedo for mineral soils.

Functions/Subroutines

- subroutine classb (THPOR, THLRET, THLMIN, BI, PSISAT, GRKSAT, THLRAT, HCPS, TCS, THFC, THLW, PSIWLT, DELZW, ZBOTW, ALGWET, ALGDRY, ALGWV, ALGWN, ALGDV, ALGDN, SAND, CLAY, ORGM, SOCI, DELZ, ZBOT, SDEPTH, ISAND, IGDR, NL, NM, IL1, IL2, IM, IG, IGRALB)

9.11.1 Detailed Description

Purpose: Assign thermal and hydraulic properties to soil layers based on sand/clay content, or soil type. Also calculate permeable thickness of soil layers, and wet and dry surface albedo for mineral soils.

9.11.2 Function/Subroutine Documentation

9.11.2.1 subroutine classb (real, dimension (nl,nm,ig) *THPOR,* real, dimension(nl,nm,ig) *THLRET,* real, dimension(nl,nm,ig) *THLMIN,* real, dimension (nl,nm,ig) *BI,* real, dimension(nl,nm,ig) *PSISAT,* real, dimension(nl,nm,ig) *GRKSAT,* real, dimension(nl,nm,ig) *THLRAT,* real, dimension (nl,nm,ig) *HCPS,* real, dimension (nl,nm,ig) *TCS,* real, dimension (nl,nm,ig) *THFC,* real, dimension (nl,nm,ig) *THLW,* real, dimension(nl,nm,ig) *PSIWLT,* real, dimension (nl,nm,ig) *DELZW,* real, dimension (nl,nm,ig) *ZBOTW,* real, dimension(nl,nm) *ALGWET,* real, dimension(nl,nm) *ALGDRY,* real, dimension (nl,nm) *ALGWV,* real, dimension (nl,nm) *ALGWN,* real, dimension (nl,nm) *ALGDV,* real, dimension (nl,nm) *ALGDN,* real, dimension (nl,nm,ig) *SAND,* real, dimension (nl,nm,ig) *CLAY,* real, dimension (nl,nm,ig) *ORGM,* real, dimension (nl,nm) *SOCI,* real, dimension (ig) *DELZ,* real, dimension (ig) *ZBOT,* real, dimension(nl,nm) *SDEPTH,* integer, dimension (nl,nm,ig) *ISAND,* integer, dimension (nl,nm) *IGDR,* integer *NL,* integer *NM,* integer *IL1,* integer *IL2,* integer *IM,* integer *IG,* integer *IGRALB*)

Parameters

thpor	Pore volume $[m^3m^{-3}](\theta_p)$
thlret	Liquid water retention capacity for organic soil $[m^3m^{-3}](\theta_{ret})$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}](\theta_{min})$
bi	Clapp and Hornberger empirical parameter [ ] (b)
psisat	Soil moisture suction at saturation [m] (Ψ_{sat})
grksat	Hydraulic conductivity of soil at saturation $[ms^{-1}](K_{sat})$
thlrat	Fractional saturation of soil at half the saturated hydraulic conductivity [ ] (f_{inf})
hcps	Volumetric heat capacity of soil matter $[Jm^{-3}K^{-1}](C_g)$
tcs	Thermal conductivity of soil $[Wm^{-1}K^{-1}](\tau_g)$
thfc	Field capacity $[m^3m^{-3}](\theta_{fc})$
psiwlt	Soil moisture suction at wilting point [m] (Ψ_{wilt})
delzw	Thickness of permeable part of soil layer [m]
zbotw	Depth of bottom of permeable part of soil layer [m]
algwet	All-wave albedo of wet soil for modelled area [ ]
algdry	All-wave albedo of dry soil for modelled area [ ]
isand	Sand content flag
igdr	Index of soil layer in which bedrock is encountered
sand	Percent sand content of soil layer [percent] (X_{sand})
clay	Percent clay content of soil layer [percent] (X_{clay})
orgm	Percent organic matter content of soil layer [percent]
delz	Thickness of soil layer [m]
zbot	Depth of bottom of soil layer [m]
sdepth	Permeable depth of soil column (depth to bedrock) [m] (z_b)
igralb	IF IGRALB IS SET TO 0, THE WET AND DRY SOIL ALBEDOS ARE CALCULATED ON T↩ HE BASIS OF SOIL TEXTURE. IF IT IS SET TO 1, THEY ARE ASSIGNED VALUES BASED ON THE NCAR CLM SOIL "COLOUR" DATASET.

In the first section of code, two integer flags are evaluated: IGDR, the index of the soil layer in which the bottom of the soil permeable depth, z_b , occurs; and ISAND, the value of the SAND variable for each soil layer converted to an integer. ISAND is used throughout the CLASS code as a flag to determine which branches of code to execute.

In loop 200, calculations are done to determine the permeable thickness DELZW of each soil layer. If the land cover is an ice sheet (indicated by an ISAND value of -4 in the top layer), the value of DELZW in each layer is set to DELZ, the standard thickness of the corresponding thermal layer, and the soil flag is set to -4. If the layer consists of rock, indicated by an ISAND value of -3, DELZW is set to 0. For soil layers where z_b occurs below the bottom of the layer, DELZW is set to DELZ; for layers where z_b occurs near the top of the layer, DELZW is set to 0 and ISAND is set to -3. For the soil layer containing z_b , DELZW is set to the distance from the top of the layer to z_b , and is further constrained to be $\geq$ 5 cm, to avoid overshoots in the water movement calculations. For all layers, the distance to the bottom of its respective permeable depth, ZBOTW, is set to the depth of the top of the layer plus DELZW. At the end of the loop, if the soil is a mineral one, the wet and dry albedo values are calculated using simple empirical functions derived from values given in Wilson and Henderson-Sellers (1985). (The dry albedo calculation varies slightly depending on whether or not CTEM is being run.)

In loop 300, various thermal and hydraulic soil properties are assigned to each of the soil layers, depending on soil type. Values of ISAND greater than zero indicate mineral soil. The pore volume θ_p , the saturated hydraulic conductivity K_{sat} , and the soil moisture suction at saturation Ψ sat are calculated from the percentage sand content X_{sand} , and the hydraulic parameter b is calculated from the percentage clay content X_{clay} , based on empirical relationships given in Cosby et al. (1984): $\theta_p = (-0.126X_{sand} + 48.9)/100.0$ $b = 0.159X_{clay} + 2.91$ $\Psi_{sat} = 0.01exp(-0.0302X_{sand} + 4.33)$ $K_{sat} = 7.0556x10 - 6exp(0.0352X_{sand} - 2.035)$

The fractional saturation of the soil at half the saturated hydraulic conductivity, f_{inf} , is calculated by inverting the Clapp and Hornberger (1978) expression relating hydraulic conductivity K to liquid water content of the soil θ_l : $K = K_{sat}(\theta_l/\theta_p)(2b + 3)$

Thus, $f_{inf} = 0.51/(2b + 3)$

The residual soil liquid water content remaining after evaporation or freezing, θ_{min} , and the liquid water retention capacity, θ_{ret} , are both set for mineral soils to a textbook value of 0.04. The volumetric sand, silt + clay and organic matter components of the soil matrix are derived by converting the percent values to volume fractions. The overall volumetric heat capacity of the soil material, C_g , is then calculated as a weighted average: $C_g = \Sigma(C_{sand}\theta_{sand} + C_{fine}\theta_{fine} + C_{org}\theta_{org})/(1 - \theta_p)$ where the subscript "fine" refers to the silt and clay particles

taken together. The thermal conductivity of the soil material, τ_g , is likewise calculated as a weighted average over the thermal conductivities of the components: $\tau_g = \Sigma(\tau_{sand}\theta_{sand} + \tau_{fine}\theta_{fine} + \tau_{org}\theta_{org})/(1 - \theta_p)$

The field capacity θ_{fc} , that is, the liquid water content of the soil at which gravitational drainage effectively ceases, is calculated by setting the expression for K above to a value of 0.1 mm d^{-1} , and solving for the liquid water content: $\theta_{fc} = \theta_p(1.157x10^{-9}/K_{sat})^{1/(2b+3)}$

The only exception is the field capacity of the lowest permeable layer in mineral soils (layer IGDR), which is determined using an expression developed by Soulis et al. (2010), which takes into account the permeable depth of the whole overlying soil: $\theta_{fc} = \theta_p/(b-1) \bullet (\Psi_{sat}b/z_b)^{1/b} \bullet [(3b+2)^{(b-1)/b} - (2b+2)^{(b-1)/b}]$

The soil moisture suction Ψ_{wilt} at the wilting point (the liquid water content at which plant roots can no longer draw water from the soil) is set to 150 m.

9.12 CLASSBD.f File Reference

Purpose: Assign values to parameters in CLASS common blocks. CLASS incorporates several kinds of parameters in its common blocks. Some are defined specifically for use in the CLASS code; some are also shared with the atmospheric model (if running in coupled mode).

9.12.1 Detailed Description

Purpose: Assign values to parameters in CLASS common blocks. CLASS incorporates several kinds of parameters in its common blocks. Some are defined specifically for use in the CLASS code; some are also shared with the atmospheric model (if running in coupled mode).

In routine CLASSBD, values are primarily assigned to the parameters that are specific to the CLASS code and are passed through it via common blocks CLASS1 through CLASS8. The table below lists the scalar parameters, their

definitions, and their designated values with units.

Name	Definition	Value	Units
VKC	Von Karman Constant	0.40	-
CT	Drag Coefficient for water	1.1510^{-3}	-
VMIN	Minimum wind speed	0.1	ms^{-1}
TCW	Thermal conductivity of water	0.57	$Wm^{-1}K^{-1}$
TCICE	Thermal conductivity of ice	2.24	$Wm^{-1}K^{-1}$
TCSAND	Thermal conductivity of sand particles	2.5	$Wm^{-1}K^{-1}$
TCCLAY	Thermal conductivity of fine mineral particles	2.5	$Wm^{-1}K^{-1}$
TCOM	Thermal conductivity of organic matter	0.25	$Wm^{-1}K^{-1}$
TCDRYS	Thermal conductivity of dry mineral soil	0.275	$Wm^{-1}K^{-1}$
RHOSOL	Density of soil mineral matter	2.6510^3	kgm^{-3}
RHOOM	Density of soil organic matter	1.3010^3	kgm^{-3}
HCPW	Volumetric heat capacity of water	4.18710^6	$Jm^{-3}K^{-1}$
HCPICE	Volumetric heat capacity of ice	$1.92571 0^6$	$Jm^{-3}K^{-1}$
HCPSOL	Volumetric heat capacity of mineral matter	2.2510^6	$Jm^{-3}K^{-1}$
HCPOM	Volumetric heat capacity of organic matter	2.5010^6	$Jm^{-3}K^{-1}$
HCPSND	Volumetric heat capacity of sand particles	2.1310^6	$Jm^{-3}K^{-1}$
HCPCLY	Volumetric heat capacity of fine mineral particles	2.3810^6	$Jm^{-3}K^{-1}$
SPHW	Specific heat of water	4.18610^3	$Jkg^{-1}K^{-1}$
SPHICE	Specific heat of ice	2.1010^3	$Jkg^{-1}K^{-1}$
SPHVEG	Specific heat of vegetation matter	2.7010^3	$Jkg^{-1}K^{-1}$
RHOW	Density of water	1.010^3	kgm^{-3}
RHOICE	Density of ice	40.91710^3	kgm^{-3}
TCGLAC	Thermal conductivity of ice sheets	2.24	$Wm^{-1}K^{-1}$
CLHMLT	Latent heat of freezing of water	0.33410^6	Jkg^{-1}
CLHVAP	Latent heat of vaporization of water	$2.5011 0^6$	Jkg^{-1}
ZOLNG	Natural log of roughness length of soil	-4.605	-
ZOLNS	Natural log of roughness length of snow	-6.908	-
ZOLNI	Natural log of roughness length of ice	-6.215	-
ZORATG	Ratio of soil roughness length for momentum to roughness length for heat	3.0	-
ALVSI	Visible albedo of ice	0.95	-
ALIRI	Near-infrared albedo of ice	0.73	-
ALVSO	Visible albedo of organic matter	0.05	-
ALIRO	Near-infrared albedo of organic matter	0.30	-
ALBRCK	Albedo of rock	0.27	-

.

Values are also assigned to several non-scalar parameters, as follows:

1) The crop growth descriptor array GROWYR (see the documentation for subroutine APREP);

2) Three parameters for the four main vegetation categories recognized by CLASS (needleleaf trees, broadleaf trees, crops and grass): ZORAT, the ratio of the roughness length for momentum to the roughness length for heat (currently set to 1); CANEXT, an attenuation coefficient used in calculating the sky view factor for vegetation canopies (variable c in the documentation for subroutine CANALB); and XLEAF, a leaf dimension factor used in calculating the leaf boundary resistance (variable Cl in the documentation for subroutine APREP);

3) Six hydraulic parameters associated with the three basic types of organic soils (fibric, hemic and sapric): THP↩ ORG, THRORG, THMORG, BORG, PSISORG and GRKSORG (see the documentation for subroutine CLASSB). The table below lists their values, derived from the work of Letts et al. (2000), alongside the symbols used in the CLASSB documentation.

Name	Symbol	Fibric peat	Hemic peat	Sapric peat
THPORG	θ_p	0.93	0.88	0.83
BORG	b	2.7	6.1	12.0
GRKSORG	K_{sat}	2.810^{-4}	2.010^{-6}	1.010^{-7}
PSISORG	psi_{sat}	0.0103	0.0102	0.0101
THMORG	$theta_{min}$	0.04	0.15	0.22
THRORG	$theta_{ret}$	0.275	0.62	0.705

Finally, if CLASS is being run offline, values must be assigned to the parameters listed in the first table which would normally be assigned in the GCM:

GCM Name	Definition	Value	Units
DELTIM	Time Step	Varies by run	s
CELZRO	Freezing point of water	273.16	K
GAS	Gas constant	287.04	$Jkg^{-1}K^{-1}$
GASV	Gas constant for water vapour	461.50	$Jkg^{-1}K^{-1}$
G	Acceleration due to gravity	9.80616	ms^{-1}
SIGMA	Stefan-Boltzmann constant	5.6679610^{-8}	$Wm^{-2}K^{-4}$
CPRES	Specific heat of air	1.0046410^{3}	$Jkg^{-1}K^{-1}$
CPI	Pi	3.14159265	-

The parameters in common blocks PHYCON and CLASSD2 that do not have corresponding values assigned in the GCM are assigned their RPN values, and the remaining parameters in the GCM common blocks PARAMS, PARAM1, PARAM3 and TIMES are assigned dummy values.

9.13 CLASSD.f File Reference

Purpose: Assign values to parameters in CLASS common blocks. CLASS incorporates several kinds of parameters in its common blocks. Some are defined specifically for use in the CLASS code; some are also shared with the atmospheric model (if running in coupled mode).

Functions/Subroutines

- subroutine **classd**

9.13.1 Detailed Description

Purpose: Assign values to parameters in CLASS common blocks. CLASS incorporates several kinds of parameters in its common blocks. Some are defined specifically for use in the CLASS code; some are also shared with the atmospheric model (if running in coupled mode).

In subroutine CLASSD, for consistency throughout the code, some parameters in the CLASS common blocks are set equal to their corresponding values in the common blocks PARAMS, PARAM1, PARAM3 and TIMES from the CGCM. Also, some parameters in the common block PHYCON which is used in the RPN subroutines DIASURFZ and FLXSURFZ are set equal to their values defined in the CGCM or in the block data routine CLASSBD. The parameters in question are as follows:

CLASS/RPN name	GCM name	Definition	Units
DELT	DELTIM	Time step	s
TFREZ	CELZRO	Freezing point of water	K
RGAS	GAS	Gas constant	$Jkg^{-1}K^{-1}$
RGASV	GASV	Gas constant for water vapour	$Jkg^{-1}K^{-1}$
GRAV	G	Acceleration due to gravity	ms^{-1}
CGRAV	G	Acceleration due to gravity	ms^{-1}
SBC	SIGMA	Stefan-Boltzmann constant	$Wm^{-2}K^{-4}$
CKARM	VKC	Von Karman constant	-
SPHAIR	CPRES	Specific heat of air	$Jkg^{-1}K^{-1}$
CPD	CPRES	Specific heat of air	$Jkg^{-1}K^{-1}$
PI	CPI	Pi	-

9.14 CLASSG.f File Reference

Purpose: Gather variables from two-dimensional arrays (latitude circle x mosaic tiles) onto long vectors for optimum processing efficiency on vector supercomputers.

Functions/Subroutines

- subroutine classg (TBARGAT, THLQGAT, THICGAT, TPNDGAT, ZPNDGAT, TBASGAT, ALBSGAT, TSNO↩ GAT, RHOSGAT, SNOGAT, TCANGAT, RCANGAT, SCANGAT, GROGAT, CMAIGAT, FCANGAT, LNZ0GAT, ALVCGAT, ALICGAT, PAMXGAT, PAMNGAT, CMASGAT, ROOTGAT, RSMNGAT, QA50GAT, VPDAGA↩ T, VPDBGAT, PSGAGAT, PSGBGAT, PAIDGAT, HGTDGAT, ACVDGAT, ACIDGAT, TSFSGAT, WSNOGAT, THPGAT, THRGAT, THMGAT, BIGAT, PSISGAT, GRKSGAT, THRAGAT, HCPSGAT, TCSGAT, IGDRGAT, THFCGAT, THLWGAT, PSIWGAT, DLZWGAT, ZBTWGAT, VMODGAT, ZSNLGAT, ZPLGGAT, ZPLSGA↩ T, TACGAT, QACGAT, DRNGAT, XSLPGAT, GRKFGAT, WFSFGAT, WFCIGAT, ALGWVGAT, ALGWNGAT, ALGDVGAT, ALGDNGAT, ALGWGAT, ALGDGAT, ASVDGAT, ASIDGAT, AGVDGAT, AGIDGAT, ISNDGAT, RADJGAT, ZBLDGAT, Z0ORGAT, ZRFMGAT, ZRFHGAT, ZDMGAT, ZDHGAT, FSVHGAT, FSIHGAT, FS↩ DBGAT, FSFBGAT, FSSBGAT, CSZGAT, FSGGAT, FLGGAT, FDLGAT, ULGAT, VLGAT, TAGAT, QAGAT, PRESGAT, PREGAT, PADRGAT, VPDGAT, TADPGAT, RHOAGAT, RPCPGAT, TRPCGAT, SPCPGAT, TS↩ PCGAT, RHSIGAT, FCLOGAT, DLONGAT, GGEOGAT, GUSTGAT, REFGAT, BCSNGAT, DEPBGAT, ILMOS, JLMOS, NML, NL, NT, NM, ILG, IG, IC, ICP1, NBS, TBARROT, THLQROT, THICROT, TPNDROT, ZPNDR↩ OT, TBASROT, ALBSROT, TSNOROT, RHOSROT, SNOROT, TCANROT, RCANROT, SCANROT, GROROT, CMAIROT, FCANROT, LNZ0ROT, ALVCROT, ALICROT, PAMXROT, PAMNROT, CMASROT, ROOTROT, RSMNROT, QA50ROT, VPDAROT, VPDBROT, PSGAROT, PSGBROT, PAIDROT, HGTDROT, ACVDROT, ACIDROT, TSFSROT, WSNOROT, THPROT, THRROT, THMROT, BIROT, PSISROT, GRKSROT, THRAR↩ OT, HCPSROT, TCSROT, IGDRROT, THFCROT, THLWROT, PSIWROT, DLZWROT, ZBTWROT, VMODL, ZSNLROT, ZPLGROT, ZPLSROT, TACROT, QACROT, DRNROT, XSLPROT, GRKFROT, WFSFROT, WF↩ CIROT, ALGWVROT, ALGWNROT, ALGDVROT, ALGDNROT, ALGWROT, ALGDROT, ASVDROT, ASIDR↩ OT, AGVDROT, AGIDROT, ISNDROT, RADJ, ZBLDROW, Z0ORROW, ZRFMROW, ZRFHROW, ZDMROW, ZDHROW, FSVHROW, FSIHROW, FSDBROL, FSFBROL, FSSBROL, CSZROW, FSGROL, FLGROL, F↩ DLROL, ULROW, VLROW, TAROW, QAROW, PRESROW, PREROW, PADRROW, VPDROW, TADPROW, RHOAROW, RPCPROW, TRPCROW, SPCPROW, TSPCROW, RHSIROW, FCLOROW, DLONROW, GG↩ EOROW, GUSTROL, REFROT, BCSNROT, DEPBROW)

9.14.1 Detailed Description

Purpose: Gather variables from two-dimensional arrays (latitude circle x mosaic tiles) onto long vectors for optimum processing efficiency on vector supercomputers.

9.14.2 Function/Subroutine Documentation

9.14.2.1 subroutine classg (real, dimension(ilg,ig) *TBARGAT,* real, dimension(ilg,ig) *THLQGAT,* real, dimension(ilg,ig) *THICGAT,* real, dimension(ilg) *TPNDGAT,* real, dimension(ilg) *ZPNDGAT,* real, dimension(ilg) *TBASGAT,* real, dimension(ilg) *ALBSGAT,* real, dimension(ilg) *TSNOGAT,* real, dimension(ilg) *RHOSGAT,* real, dimension (ilg) *SNOGAT,* real, dimension(ilg) *TCANGAT,* real, dimension(ilg) *RCANGAT,* real, dimension(ilg) *SCANGAT,* real, dimension (ilg) *GROGAT,* real, dimension(ilg) *CMAIGAT,* real, dimension(ilg,icp1) *FCANGAT,* real, dimension(ilg,icp1) *LNZ0GAT,* real, dimension(ilg,icp1) *ALVCGAT,* real, dimension(ilg,icp1) *ALICGAT,* real, dimension(ilg,ic) *PAMXGAT,* real, dimension(ilg,ic) *PAMNGAT,* real, dimension(ilg,ic) *CMASGAT,* real, dimension(ilg,ic) *ROOTGAT,* real, dimension(ilg,ic) *RSMNGAT,* real, dimension(ilg,ic) *QA50GAT,* real, dimension(ilg,ic) *VPDAGAT,* real, dimension(ilg,ic) *VPDBGAT,* real, dimension(ilg,ic) *PSGAGAT,* real, dimension(ilg,ic) *PSGBGAT,* real, dimension(ilg,ic) *PAIDGAT,* real, dimension(ilg,ic) *HGTDGAT,* real, dimension(ilg,ic) *ACVDGAT,* real, dimension(ilg,ic) *ACIDGAT,* real, dimension(ilg,4) *TSFSGAT,* real, dimension(ilg) *WSNOGAT,* real, dimension (ilg,ig) *THPGAT,* real, dimension (ilg,ig) *THRGAT,* real, dimension (ilg,ig) *THMGAT,* real, dimension (ilg,ig) *BIGAT,* real, dimension(ilg,ig) *PSISGAT,* real, dimension(ilg,ig) *GRKSGAT,* real, dimension(ilg,ig) *THRAGAT,* real, dimension(ilg,ig) *HCPSGAT,* real, dimension (ilg,ig) *TCSGAT,* integer, dimension(ilg) *IGDRGAT,* real, dimension(ilg,ig) *THFCGAT,* real, dimension(ilg,ig) *THLWGAT,* real, dimension(ilg,ig) *PSIWGAT,* real, dimension(ilg,ig) *DLZWGAT,* real, dimension(ilg,ig) *ZBTWGAT,* real, dimension(ilg) *VMODGAT,* real, dimension(ilg) *ZSNLGAT,* real, dimension(ilg) *ZPLGGAT,* real, dimension(ilg) *ZPLSGAT,* real, dimension (ilg) *TACGAT,* real, dimension (ilg) *QACGAT,* real, dimension (ilg) *DRNGAT,* real, dimension(ilg) *XSLPGAT,* real, dimension(ilg) *GRKFGAT,* real, dimension(ilg) *WFSFGAT,* real, dimension(ilg) *WFCIGAT,* real, dimension(ilg) *ALGWVGAT,* real, dimension(ilg) *ALGWNGAT,* real, dimension(ilg) *ALGDVGAT,* real, dimension(ilg) *ALGDNGAT,* real, dimension(ilg) *ALGWGAT,* real, dimension(ilg) *ALGDGAT,* real, dimension(ilg) *ASVDGAT,* real, dimension(ilg) *ASIDGAT,* real, dimension(ilg) *AGVDGAT,* real, dimension(ilg) *AGIDGAT,* integer, dimension(ilg,ig) *ISNDGAT,* real, dimension(ilg) *RADJGAT,* real, dimension(ilg) *ZBLDGAT,* real, dimension(ilg) *Z0ORGAT,* real, dimension(ilg) *ZRFMGAT,* real, dimension(ilg) *ZRFHGAT,* real, dimension (ilg) *ZDMGAT,* real, dimension (ilg) *ZDHGAT,* real, dimension(ilg) *FSVHGAT,* real, dimension(ilg) *FSIHGAT,* real, dimension(ilg,nbs) *FSDBGAT,* real, dimension(ilg,nbs) *FSFBGAT,* real, dimension(ilg,nbs) *FSSBGAT,* real, dimension (ilg) *CSZGAT,* real, dimension (ilg) *FSGGAT,* real, dimension (ilg) *FLGGAT,* real, dimension (ilg) *FDLGAT,* real, dimension (ilg) *ULGAT,* real, dimension (ilg) *VLGAT,* real, dimension (ilg) *TAGAT,* real, dimension (ilg) *QAGAT,* real, dimension(ilg) *PRESGAT,* real, dimension (ilg) *PREGAT,* real, dimension(ilg) *PADRGAT,* real, dimension (ilg) *VPDGAT,* real, dimension(ilg) *TADPGAT,* real, dimension(ilg) *RHOAGAT,* real, dimension(ilg) *RPCPGAT,* real, dimension(ilg) *TRPCGAT,* real, dimension(ilg) *SPCPGAT,* real, dimension(ilg) *TSPCGAT,* real, dimension(ilg) *RHSIGAT,* real, dimension(ilg) *FCLOGAT,* real, dimension(ilg) *DLONGAT,* real, dimension(ilg) *GGEOGAT,* real, dimension(ilg) *GUSTGAT,* real, dimension (ilg) *REFGAT,* real, dimension(ilg) *BCSNGAT,* real, dimension(ilg) *DEPBGAT,* integer, dimension (ilg) *ILMOS,* integer, dimension (ilg) *JLMOS,* integer *NML,* integer *NL,* integer *NT,* integer *NM,* integer *ILG,* integer *IG,* integer *IC,* integer *ICP1,* integer *NBS,* real, dimension(nl,nt,ig) *TBARROT,* real, dimension(nl,nt,ig) *THLQROT,* real, dimension(nl,nt,ig) *THICROT,* real, dimension(nl,nt) *TPNDROT,* real, dimension(nl,nt) *ZPNDROT,* real, dimension(nl,nt) *TBASROT,* real, dimension(nl,nm) *ALBSROT,* real, dimension(nl,nm) *TSNOROT,* real, dimension(nl,nm) *RHOSROT,* real, dimension (nl,nm) *SNOROT,* real, dimension(nl,nt) *TCANROT,* real, dimension(nl,nt) *RCANROT,* real, dimension(nl,nt) *SCANROT,* real, dimension (nl,nt) *GROROT,* real, dimension(nl,nt) *CMAIROT,* real, dimension(nl,nt,icp1) *FCANROT,* real, dimension(nl,nt,icp1) *LNZ0ROT,* real, dimension(nl,nt,icp1) *ALVCROT,* real, dimension(nl,nt,icp1) *ALICROT,* real, dimension(nl,nt,ic) *PAMXROT,* real, dimension(nl,nt,ic) *PAMNROT,* real, dimension(nl,nt,ic) *CMASROT,* real, dimension(nl,nt,ic) *ROOTROT,* real, dimension(nl,nt,ic) *RSMNROT,* real, dimension(nl,nt,ic) *QA50ROT,* real, dimension(nl,nt,ic) *VPDAROT,* real, dimension(nl,nt,ic) *VPDBROT,* real, dimension(nl,nt,ic) *PSGAROT,* real, dimension(nl,nt,ic) *PSGBROT,* real, dimension(nl,nt,ic) *PAIDROT,* real, dimension(nl,nt,ic) *HGTDROT,* real, dimension(nl,nt,ic) *ACVDROT,* real, dimension(nl,nt,ic) *ACIDROT,* real, dimension(nl,nt,4) *TSFSROT,* real, dimension(nl,nt) *WSNOROT,* real, dimension (nl,nt,ig) *THPROT,* real, dimension (nl,nt,ig) *THRROT,* real, dimension (nl,nt,ig) *THMROT,* real, dimension (nl,nt,ig) *BIROT,* real, dimension(nl,nt,ig) *PSISROT,* real, dimension(nl,nt,ig) *GRKSROT,* real, dimension(nl,nt,ig) *THRAROT,* real, dimension(nl,nt,ig) *HCPSROT,* real, dimension (nl,nt,ig) *TCSROT,* integer, dimension(nl,nt) *IGDRROT,* real, dimension(nl,nt,ig) *THFCROT,* real, dimension(nl,nt,ig) *THLWROT,* real, dimension(nl,nt,ig) *PSIWROT,* real, dimension(nl,nt,ig) *DLZWROT,* real, dimension(nl,nt,ig) *ZBTWROT,* real, dimension (nl) *VMODL,* real, dimension(nl,nt) *ZSNLROT,* real, dimension(nl,nt) *ZPLGROT,* real, dimension(nl,nt) *ZPLSROT,* real, dimension (nl,nt) *TACROT,* real, dimension (nl,nt) *QACROT,* real, dimension (nl,nt) *DRNROT,* real, dimension(nl,nt) *XSLPROT,* real, dimension(nl,nt) *GRKFROT,* real, dimension(nl,nt) *WFSFROT,* real, dimension(nl,nt) *WFCIROT,* real, dimension(nl,nt) *ALGWVROT,* real, dimension(nl,nt) *ALGWNROT,* real, dimension(nl,nt) *ALGDVROT,* real, dimension(nl,nt) *ALGDNROT,* real, dimension(nl,nt) *ALGWROT,* real, dimension(nl,nt) *ALGDROT,* real, dimension(nl,nt) *ASVDROT,* real, dimension(nl,nt) *ASIDROT,* real, dimension(nl,nt) *AGVDROT,* real, dimension(nl,nt) *AGIDROT,* integer, dimension(nl,nt,ig) *ISNDROT,* real, dimension (nl) *RADJ,* real, dimension(nl) *ZBLDROW,* real, dimension(nl) *Z0ORROW,* real, dimension(nl) *ZRFMROW,* real, dimension(nl) *ZRFHROW,* real, dimension (nl) *ZDMROW,* real, dimension (nl) *ZDHROW,* real, dimension(nl) *FSVHROW,* real, dimension(nl) *FSIHROW,* real, dimension(nl,nbs) *FSDBROL,* real, dimension(nl,nbs) *FSFBROL,* real, dimension(nl,nbs) *FSSBROL,* real, dimension (nl) *CSZROW,* real, dimension (nl) *FSGROL,* real, dimension (nl) *FLGROL,* real, dimension (nl) *FDLROL,* real, dimension (nl) *ULROW,* real, dimension (nl) *VLROW,* real, dimension (nl) *TAROW,* real, dimension (nl) *QAROW,* real, dimension(nl) *PRESROW,* real, dimension(nl) *PREROW,* real, dimension(nl) *PADRROW,* real, dimension (nl) *VPDROW,* real, dimension(nl) *TADPROW,* real, dimension(nl) *RHOAROW,* real, dimension(nl) *RPCPROW,* real, dimension(nl) *TRPCROW,* real, dimension(nl) *SPCPROW,* real, dimension(nl) *TSPCROW,* real, dimension(nl) *RHSIROW,* real, dimension(nl) *FCLOROW,* real,

Parameters

tbarrot	Temperature of soil layers [K]
thlqrot	Volumetric liquid water content of soil layers $[m^3 m^{-3}]$
thicrot	Frozen water content of soil layers under vegetation $[m^3 m^{-3}]$
tpndrot	Temperature of ponded water [K]
zpndrot	Depth of ponded water on surface [m]
tbasrot	Temperature of bedrock in third soil layer [K]
albsrot	Snow albedo [ ]
tsnorot	Snowpack temperature [K]
rhosrot	Density of snow $[kgm^{-3}]$
snorot	Mass of snow pack $[kgm^{-2}]$
tcanrot	Vegetation canopy temperature [K]
rcanrot	Intercepted liquid water stored on canopy $[kgm^{-2}]$
scanrot	Intercepted frozen water stored on canopy $[kgm^{-2}]$
grorot	Vegetation growth index [ ]
cmairot	Aggregated mass of vegetation canopy $[kgm^{-2}]$
tsfsrot	Ground surface temperature over subarea [K]
tacrot	Temperature of air within vegetation canopy [K]
qacrot	Specific humidity of air within vegetation canopy $[kgkg^{-1}]$
wsnorot	Liquid water content of snow pack $[kgm^{-2}]$
ilmos	Index of latitude grid cell corresponding to current element of gathered vector of land surface variables [ ]
jlmos	Index of mosaic tile corresponding to current element of gathered vector of land surface variables [ ]
fcanrot	Maximum fractional coverage of modelled area by vegetation category [ ]
lnz0rot	Natural logarithm of maximum roughness length of vegetation category [ ]
alvcrot	Background average visible albedo of vegetation category [ ]
alicrot	Background average near-infrared albedo of vegetation category [ ]
pamxrot	Maximum plant area index of vegetation category [ ]
pamnrot	Minimum plant area index of vegetation category [ ]
cmasrot	Maximum canopy mass for vegetation category $[kgm^{-2}]$
rootrot	Maximum rooting depth of vegetation category [m]
rsmnrot	Minimum stomatal resistance of vegetation category $[sm^{-1}]$
qa50rot	Reference value of incoming shortwave radiation for vegetation category (used in stomatal resistance calculation) $[Wm^{-2}]$
vpdarot	Vapour pressure deficit coefficient for vegetation category (used in stomatal resistance calculation) [ ]
vpdbrot	Vapour pressure deficit coefficient for vegetation category (used in stomatal resistance calculation) [ ]
psgarot	Soil moisture suction coefficient for vegetation category (used in stomatal resistance calculation) [ ]
psgbrot	Soil moisture suction coefficient for vegetation category (used in stomatal resistance calculation) [ ]
paidrot	Optional user-specified value of plant area indices of vegetation categories to override CL↩ ASS-calculated values [ ]
hgtdrot	Optional user-specified values of height of vegetation categories to override CLASS-calculated values [m]

acvdrot	Optional user-specified value of canopy visible albedo to override CLASS- calculated value [ ]
acidrot	Optional user-specified value of canopy near-infrared albedo to override CLASS- calculated value [ ]
thprot	Pore volume in soil layer $[m^3m^{-3}]$
thrrot	Liquid water retention capacity for organic soil $[m^3m^{-3}]$
thmrot	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}]$
birot	Clapp and Hornberger empirical "b" parameter [ ]
psisrot	Soil moisture suction at saturation [m]
grksrot	Saturated hydraulic conductivity of soil layers $[ms^{-1}]$
thrarot	Fractional saturation of soil behind the wetting front [ ]
hcpsrot	Volumetric heat capacity of soil particles $[Jm^{-3}]$
tcsrot	Thermal conductivity of soil particles $[Wm^{-1}K^{-1}]$
thfcrot	Field capacity $[m^3m^{-3}]$
thlwrot	Wilting point $[m^3m^{-3}]$
psiwrot	Soil moisture suction at wilting point [m]
dlzwrot	Permeable thickness of soil layer [m]
zbtwrot	Depth to permeable bottom of soil layer [m]
drnrot	Drainage index at bottom of soil profile [ ]
xslprot	Surface slope (used when running MESH code) [degrees]
grkfrot	WATROF parameter used when running MESH code [ ]
wfsfrot	WATROF parameter used when running MESH code [ ]
wfcirot	WATROF parameter used when running MESH code [ ]
algwrot	Reference albedo for saturated soil [ ]
algdrot	Reference albedo for dry soil [ ]
asvdrot	Optional user-specified value of snow visible albedo to override CLASS- calculated value [ ]
asidrot	Optional user-specified value of snow near-infrared albedo to override CLASS- calculated value [ ]
agvdrot	Optional user-specified value of ground visible albedo to override CLASS- calculated value [ ]
agidrot	Optional user-specified value of ground near-infrared albedo to override CLASS- calculated value [ ]
zsnlrot	Limiting snow depth below which coverage is $<$ 100% [m]
zplgrot	Maximum water ponding depth for snow- free subareas (user-specified when running MESH code) [m]
zplsrot	Maximum water ponding depth for snow- covered subareas (user-specified when running MESH code) [m]
isndgat	Sand content flag
igdrgat	Index of soil layer in which bedrock is encountered
zrfmrow	Reference height associated with forcing wind speed [m]
zrfhrow	Reference height associated with forcing air temperature and humidity [m]
zdmrow	User-specified height associated with diagnosed anemometer-level wind speed [m]
zdhrow	User-specified height associated with diagnosed screen-level variables [m]
fsvhrow	Visible radiation incident on horizontal surface $[Wm^{-2}]$
fsihrow	Near-infrared radiation incident on horizontal surface $[Wm^{-2}]$
cszrow	Cosine of solar zenith angle [ ]
fdlrol	Downwelling longwave radiation at bottom of atmosphere $[Wm^{-2}]$
ulrow	Zonal component of wind speed $[ms^{-1}]$
vlrow	Meridional component of wind speed $[ms^{-1}]$

tarow	Air temperature at reference height [K]
qarow	Specific humidity at reference height $[kgkg^{-1}]$
presrow	Surface air pressure [Pa]
prerow	Surface precipitation rate $[kgm^{-2}s^{-1}]$
padrrow	Partial pressure of dry air [Pa]
vpdrow	Vapour pressure deficit [mb]
tadprow	Dew point temperature of air [K]
rhoarow	Density of air $[kgm^{-3}]$
zbldrow	Atmospheric blending height for surface roughness length averaging [m]
z0orrow	Orographic roughness length [m]
rpcprow	Rainfall rate over modelled area $[ms^{-1}]$
trpcrow	Rainfall temperature [K]
spcprow	Snowfall rate over modelled area $[ms^{-1}]$
tspcrow	Snowfall temperature [K]
rhsirow	Density of fresh snow $[kgm^{-3}]$
fclorow	Fractional cloud cover [ ]
dlonrow	Longitude of grid cell (east of Greenwich) [degrees]
ggeorow	Geothermal heat flux at bottom of soil profile $[Wm^{-2}]$
radj	Latitude of grid cell (positive north of equator) [rad]
vmodl	Wind speed at reference height $[ms^{-1}]$

9.15 CLASSI.f File Reference

Purpose: Evaluate atmospheric variables and rainfall/snowfall rates over modelled area.

Functions/Subroutines

- subroutine classi (VPD, TADP, PADRY, RHOAIR, RHOSNI, RPCP, TRPCP, SPCP, TSPCP, TA, QA, PCPR, RRATE, SRATE, PRESSG, IPCP, NL, IL1, IL2)

9.15.1 Detailed Description

Purpose: Evaluate atmospheric variables and rainfall/snowfall rates over modelled area.

9.15.2 Function/Subroutine Documentation

9.15.2.1 subroutine classi (real, dimension (nl) *VPD,* real, dimension (nl) *TADP,* real, dimension (nl) *PADRY,* real, dimension(nl) *RHOAIR,* real, dimension(nl) *RHOSNI,* real, dimension (nl) *RPCP,* real, dimension (nl) *TRPCP,* real, dimension (nl) *SPCP,* real, dimension (nl) *TSPCP,* real, dimension (nl) *TA,* real, dimension (nl) *QA,* real, dimension (nl) *PCPR,* real, dimension (nl) *RRATE,* real, dimension (nl) *SRATE,* real, dimension(nl) *PRESSG,* integer *IPCP,* integer *NL,* integer *IL1,* integer *IL2*)

Parameters

vpd	Vapour pressure deficit of air $[mb](e_d)$
tadp	Dew point temperature of air [K]
padry	Partial pressure of dry air $[Pa](p_{dry})$
rhoair	Density of air $[kgm^{-3}](\rho_a)$

rhosni	Density of fresh snow $[kgm^{-3}](\rho_{s,i})$
rpcp	Calculated rainfall rate over modelled area $[ms^{-1}]$
trpcp	Rainfall temperature over modelled area [C]
spcp	Calculated snowfall rate over modelled area $[ms^{-1}]$
tspcp	Snowfall temperature over modelled area [C]
ta	Air temperature at reference height $[K](T_a)$
qa	Specific humidity at reference height $[kgkg^{-1}](q_a)$
pressg	Surface atmospheric pressure $[Pa](p)$
pcpr	Precipitation rate over modelled area $[kgm^{-2}s^{-1}]$
rrate	Input rainfall rate over modelled area $[kgm^{-2}s^{-1}]$
srate	Input snowfall rate over modelled area $[kgm^{-2}s^{-1}]$

In the first section, the air vapour pressure deficit, dry air pressure, air density and dew point temperature are calculated. The vapour pressure deficit e_d (in units of mb) is obtained from the saturated and actual vapour pressures of the air, e_a and $e_{a,sat}$ respectively (in units of Pa), as $e_d = [e_{a,sat} - e_a]/100.0$

The air vapour pressure is obtained from the specific humidity q_a using the formula

$e_a = q_a p/[0.622 + 0.378 q_a]$

where p is the surface atmospheric pressure. For the saturated vapour pressure, a standard empirical equation is utilized relating $e_{a,sat}$ to the temperature T_a and the freezing point T_f:

$e_{a,sat} = 611.0 exp[17.269 * (T_a \check{} T_f)/(T_a \check{} 35.86)]\ T_a \geq T_f$

$e_{a,sat} = 611.0 exp[21.874 (T_a \check{} T_f)/(T_a \check{} 7.66)]\ T_a < T_f$

The partial pressure of dry air, p_{dry}, is obtained by subtracting e_a from p, and the density of the air is calculated as the sum of the densities of the dry air and the water vapour: $\rho_a = p_{dry}/R_d T_a + e_a/R_v T_a$

where R_d and R_v are the gas constants for dry air and water vapour respectively. The dew point temperature of the air is evaluated by substituting e_a for $e_{a,sat}$ on the left-hand side of the appropriate equation above, and solving for T_a.

Finally, if IPCP = 4, this indicates that the partitioning between rainfall and snowfall has been done outside of C↩ LASS. The rainfall and snowfall rates RRATE and SRATE that have been passed into the subroutine are therefore simply assigned to RPCP and SPCP.

9.16 CLASSS.f File Reference

Purpose: Scatter variables from long, gathered vectors back onto original two-dimensional arrays (latitude circle x mosaic tiles).

Functions/Subroutines

- subroutine classs (TBARROT, THLQROT, THICROT, TSFSROT, TPNDROT, ZPNDROT, TBASROT, ALBS↩ ROT, TSNOROT, RHOSROT, SNOROT, GTROT, TCANROT, RCANROT, SCANROT, GROROT, CMAIROT, TACROT, QACROT, WSNOROT, REFROT, BCSNROT, EMISROT, SALBROT, CSALROT, ILMOS, JLMOS, NML, NL, NT, NM, ILG, IG, IC, ICP1, NBS, TBARGAT, THLQGAT, THICGAT, TSFSGAT, TPNDGAT, ZPND↩ GAT, TBASGAT, ALBSGAT, TSNOGAT, RHOSGAT, SNOGAT, GTGAT, TCANGAT, RCANGAT, SCANGAT, GROGAT, CMAIGAT, TACGAT, QACGAT, WSNOGAT, REFGAT, BCSNGAT, EMISGAT, SALBGAT, CSAL↩ GAT)

9.16.1 Detailed Description

Purpose: Scatter variables from long, gathered vectors back onto original two-dimensional arrays (latitude circle x mosaic tiles).

9.16.2 Function/Subroutine Documentation

9.16.2.1 subroutine classs (real, dimension(nl,nt,ig) *TBARROT,* real, dimension(nl,nt,ig) *THLQROT,* real, dimension(nl,nt,ig) *THICROT,* real, dimension(nl,nt,4) *TSFSROT,* real, dimension(nl,nt) *TPNDROT,* real, dimension(nl,nt) *ZPNDROT,* real, dimension(nl,nt) *TBASROT,* real, dimension(nl,nm) *ALBSROT,* real, dimension(nl,nm) *TSNOROT,* real, dimension(nl,nm) *RHOSROT,* real, dimension (nl,nm) *SNOROT,* real, dimension (nl,nm) *GTROT,* real, dimension(nl,nt) *TCANROT,* real, dimension(nl,nt) *RCANROT,* real, dimension(nl,nt) *SCANROT,* real, dimension (nl,nt) *GROROT,* real, dimension(nl,nt) *CMAIROT,* real, dimension (nl,nt) *TACROT,* real, dimension (nl,nt) *QACROT,* real, dimension(nl,nt) *WSNOROT,* real, dimension (nl,nm) *REFROT,* real, dimension(nl,nm) *BCSNROT,* real, dimension(nl,nm) *EMISROT,* real, dimension(nl,nm,nbs) *SALBROT,* real, dimension(nl,nm,nbs) *CSALROT,* integer, dimension (ilg) *ILMOS,* integer, dimension (ilg) *JLMOS,* integer *NML,* integer *NL,* integer *NT,* integer *NM,* integer *ILG,* integer *IG,* integer *IC,* integer *ICP1,* integer *NBS,* real, dimension(ilg,ig) *TBARGAT,* real, dimension(ilg,ig) *THLQGAT,* real, dimension(ilg,ig) *THICGAT,* real, dimension(ilg,4) *TSFSGAT,* real, dimension(ilg) *TPNDGAT,* real, dimension(ilg) *ZPNDGAT,* real, dimension(ilg) *TBASGAT,* real, dimension(ilg) *ALBSGAT,* real, dimension(ilg) *TSNOGAT,* real, dimension(ilg) *RHOSGAT,* real, dimension (ilg) *SNOGAT,* real, dimension (ilg) *GTGAT,* real, dimension(ilg) *TCANGAT,* real, dimension(ilg) *RCANGAT,* real, dimension(ilg) *SCANGAT,* real, dimension (ilg) *GROGAT,* real, dimension(ilg) *CMAIGAT,* real, dimension (ilg) *TACGAT,* real, dimension (ilg) *QACGAT,* real, dimension(ilg) *WSNOGAT,* real, dimension (ilg) *REFGAT,* real, dimension(ilg) *BCSNGAT,* real, dimension(ilg) *EMISGAT,* real, dimension(ilg,nbs) *SALBGAT,* real, dimension(ilg,nbs) *CSALGAT*)

Parameters

salbrot	Suffix ROT refers to variables on original two-dimensional arrays.
csalrot	Suffix ROT refers to variables on original two-dimensional arrays.
tbarrot	Temperature of soil layers [K] Suffix ROT refers to variables on original two-dimensional arrays.
thlqrot	Volumetric liquid water content of soil layers $[m^3m^{-3}]$ Suffix ROT refers to variables on original two-dimensional arrays.
thicrot	Volumetric frozen water content of soil layers $[m^3m^{-3}]$ Suffix ROT refers to variables on original two-dimensional arrays.
tsfsrot	Ground surface temperature over subarea [K] Suffix ROT refers to variables on original two-dimensional arrays.
tpndrot	Temperature of ponded water [K] Suffix ROT refers to variables on original two-dimensional arrays.
zpndrot	Depth of ponded water on surface [m] Suffix ROT refers to variables on original two-dimensional arrays.
tbasrot	Temperature of bedrock in third soil layer [K] Suffix ROT refers to variables on original two-dimensional arrays.
albsrot	Snow albedo [ ] Suffix ROT refers to variables on original two-dimensional arrays.
tsnorot	Snowpack temperature [K] Suffix ROT refers to variables on original two-dimensional arrays.
rhosrot	Density of snow $[kgm^{-3}]$ Suffix ROT refers to variables on original two-dimensional arrays.
snorot	Mass of snow pack $[kgm^{-2}]$ Suffix ROT refers to variables on original two-dimensional arrays.
gtrot	Suffix ROT refers to variables on original two-dimensional arrays.
tcanrot	Vegetation canopy temperature [K] Suffix ROT refers to variables on original two-dimensional arrays.
rcanrot	Intercepted liquid water stored on canopy $[kgm^{-2}]$ Suffix ROT refers to variables on original two-dimensional arrays.
scanrot	Intercepted frozen water stored on canopy $[kgm^{-2}]$ Suffix ROT refers to variables on original two-dimensional arrays.

grorot	Vegetation growth index [ ] Suffix ROT refers to variables on original two-dimensional arrays.
tacrot	Temperature of air within vegetation canopy [K] Suffix ROT refers to variables on original two-dimensional arrays.
qacrot	Specific humidity of air within vegetation canopy space $[kgkg^{-1}]$ Suffix ROT refers to variables on original two-dimensional arrays.
wsnorot	Liquid water content of snow pack $[kgm^{-2}]$ Suffix ROT refers to variables on original two-dimensional arrays.
cmairot	Aggregated mass of vegetation canopy $[kgm^{-2}]$ Suffix ROT refers to variables on original two-dimensional arrays.
refrot	Suffix ROT refers to variables on original two-dimensional arrays.
bcsnrot	Suffix ROT refers to variables on original two-dimensional arrays.
emisrot	Suffix ROT refers to variables on original two-dimensional arrays.
salbgat	Suffix GAT refers to variables on gathered long vectors.
csalgat	Suffix GAT refers to variables on gathered long vectors.
tbargat	Temperature of soil layers [K] Suffix GAT refers to variables on gathered long vectors.
thlqgat	Volumetric liquid water content of soil layers $[m^3m^{-3}]$ Suffix GAT refers to variables on gathered long vectors.
thicgat	Volumetric frozen water content of soil layers $[m^3m^{-3}]$ Suffix GAT refers to variables on gathered long vectors.
tsfsgat	Ground surface temperature over subarea [K] Suffix GAT refers to variables on gathered long vectors.
tpndgat	Temperature of ponded water [K] Suffix GAT refers to variables on gathered long vectors.
zpndgat	Depth of ponded water on surface [m] Suffix GAT refers to variables on gathered long vectors.
tbasgat	Temperature of bedrock in third soil layer [K] Suffix GAT refers to variables on gathered long vectors.
albsgat	Snow albedo [ ] Suffix GAT refers to variables on gathered long vectors.
tsnogat	Snowpack temperature [K] Suffix GAT refers to variables on gathered long vectors.
rhosgat	Density of snow $[kgm^{-3}]$ Suffix GAT refers to variables on gathered long vectors.
snogat	Mass of snow pack $[kgm^{-2}]$ Suffix GAT refers to variables on gathered long vectors.
gtgat	Suffix GAT refers to variables on gathered long vectors.
tcangat	Vegetation canopy temperature [K] Suffix GAT refers to variables on gathered long vectors.
rcangat	Intercepted liquid water stored on canopy $[kgm^{-2}]$ Suffix GAT refers to variables on gathered long vectors.
scangat	Intercepted frozen water stored on canopy $[kgm^{-2}]$ Suffix GAT refers to variables on gathered long vectors.
grogat	Vegetation growth index [ ] Suffix GAT refers to variables on gathered long vectors.
tacgat	Temperature of air within vegetation canopy [K] Suffix GAT refers to variables on gathered long vectors.
qacgat	Specific humidity of air within vegetation canopy space $[kgkg^{-1}]$ Suffix GAT refers to variables on gathered long vectors.
wsnogat	Liquid water content of snow pack $[kgm^{-2}]$ Suffix GAT refers to variables on gathered long vectors.
cmaigat	Aggregated mass of vegetation canopy $[kgm^{-2}]$ Suffix GAT refers to variables on gathered long vectors.
refgat	Suffix GAT refers to variables on gathered long vectors.
bcsngat	Suffix GAT refers to variables on gathered long vectors.
emisgat	Suffix GAT refers to variables on gathered long vectors.

The prognostic variables are scattered from the long, gathered arrays (collapsing the latitude and mosaic dimensions into one) back onto the original arrays using the pointer vectors generated in GATPREP.

9.17 CLASST.f File Reference

Purpose: Call subroutines to perform surface energy budget calculations.

Functions/Subroutines

- subroutine classt (TBARC, TBARG, TBARCS, TBARGS, THLIQC, THLIQG, THICEC, THICEG, HCPC, HCPG, TCTOPC, TCBOTC, TCTOPG, TCBOTG, GZEROC, GZEROG, GZROCS, GZROGS, G12C, G12G, G12CS, G12GS, G23C, G23G, G23CS, G23GS, QFREZC, QFREZG, QMELTC, QMELTG, EVAPC, EVAPCG, EVAPG, EVAPCS, EVPCSG, EVAPGS, TCANO, TCANS, RAICAN, SNOCAN, RAICNS, SNOCNS, CHCAP, CHCAPS, TPONDC, TPONDG, TPNDCS, TPNDGS, TSNOCS, TSNOGS, WSNOCS, WSNOGS, RHOSCS, RHOSGS, ITERCT, CDH, CDM, QSENS, TFLUX, QEVAP, EVAP, EVPPOT, ACOND, EVAPB, GT, QG, ST, SU, SV, SQ, SRH, GTBS, SFCUBS, SFCVBS, USTARBS, FSGV, FSGS, FSGG, FLGV, FLGS, FLGG, HFSC, HFSS, HFSG, HEVC, HEVS, HEVG, HMFC, HMFN, HTCC, HTCS, HTC, QFCF, QFCL, DRAG, WTABLE, ILMO, UE, HBL, TAC, QAC, ZREFM, ZREFH, ZDIAGM, ZDIAGH, VPD, TADP, RHOAIR, QSWINV, QSWINI, QLWIN, UWIND, VWIND, TA, QA, PADRY, FC, FG, FCS, FGS, RBCOEF, FSVF, FSVFS, PRESSG, VMOD, ALVSCN, ALIRCN, ALVSG, ALIRG, ALVSCS, ALIRCS, ALVSSN, ALIRSN, ALVSGC, ALIRGC, ALVSSC, ALIRSC, TRVSCN, TRIRCN, TRVSCS, TRIRCS, RC, RCS, WTRG, QLWAVG, FRAINC, FSNOWC, FRAICS, FSNOCS, CMASSC, CMASCS, DISP, DISPS, ZOMLNC, ZOELNC, ZOMLNG, ZOELNG, ZOMLCS, ZOELCS, ZOMLNS, ZOELNS, TBAR, THLIQ, THICE, TPOND, ZPOND, TBASE, TCAN, TSNOW, ZSNOW, RHOSNO, WSNOW, THPOR, THLRET, THLMIN, THFC, THLW, TRSNOWC, TRSNOWG, ALSNO, FSSB, FROOT, FROOTS, RADJ, PCPR, HCPS, TCS, TSFSAV, DELZ, DELZW, ZBOTW, FTEMP, FVAP, RIB, ISAND, AILCG, AILCGS, FCANC, FCANCS, CO2CONC, CO2I1CG, CO2I1CS, CO2I2CG, CO2I2CS, COSZS, XDIFFUS, SLAI, ICTEM, ctem_on, RMATCTEM, FCANCMX, L2MAX, NOL2PFTS, CFLUXCG, CFLUXCS, ANCSVEG, ANCGVEG, RMLCSVEG, RMLCGVEG, TCSNOW, GSNOW, ITC, ITCG, ITG, ILG, IL1, IL2, JL, N, IC, IG, IZREF, ISLFD, NLANDCS, NLANDGS, NLANDC, NLANDG, NLANDI, NBS, ISNOALB, LFSTATUS, DAYL, DAYL_MAX)

9.17.1 Detailed Description

Purpose: Call subroutines to perform surface energy budget calculations.

9.17.2 Function/Subroutine Documentation

9.17.2.1 subroutine classt (real, dimension (ilg,ig) *TBARC,* real, dimension (ilg,ig) *TBARG,* real, dimension(ilg,ig) *TBARCS,* real, dimension(ilg,ig) *TBARGS,* real, dimension(ilg,ig) *THLIQC,* real, dimension(ilg,ig) *THLIQG,* real, dimension(ilg,ig) *THICEC,* real, dimension(ilg,ig) *THICEG,* real, dimension (ilg,ig) *HCPC,* real, dimension (ilg,ig) *HCPG,* real, dimension(ilg,ig) *TCTOPC,* real, dimension(ilg,ig) *TCBOTC,* real, dimension(ilg,ig) *TCTOPG,* real, dimension(ilg,ig) *TCBOTG,* real, dimension(ilg) *GZEROC,* real, dimension(ilg) *GZEROG,* real, dimension(ilg) *GZROCS,* real, dimension(ilg) *GZROGS,* real, dimension (ilg) *G12C,* real, dimension (ilg) *G12G,* real, dimension (ilg) *G12CS,* real, dimension (ilg) *G12GS,* real, dimension (ilg) *G23C,* real, dimension (ilg) *G23G,* real, dimension (ilg) *G23CS,* real, dimension (ilg) *G23GS,* real, dimension(ilg) *QFREZC,* real, dimension(ilg) *QFREZG,* real, dimension(ilg) *QMELTC,* real, dimension(ilg) *QMELTG,* real, dimension (ilg) *EVAPC,* real, dimension(ilg) *EVAPCG,* real, dimension (ilg) *EVAPG,* real, dimension(ilg) *EVAPCS,* real, dimension(ilg) *EVPCSG,* real, dimension(ilg) *EVAPGS,* real, dimension (ilg) *TCANO,* real, dimension (ilg) *TCANS,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension(ilg) *RAICNS,* real, dimension(ilg) *SNOCNS,* real, dimension (ilg) *CHCAP,* real, dimension(ilg) *CHCAPS,* real, dimension(ilg) *TPONDC,* real, dimension(ilg) *TPONDG,* real, dimension(ilg) *TPNDCS,* real, dimension(ilg) *TPNDGS,* real, dimension(ilg) *TSNOCS,* real, dimension(ilg) *TSNOGS,* real, dimension(ilg) *WSNOCS,* real, dimension(ilg) *WSNOGS,* real, dimension(ilg) *RHOSCS,* real, dimension(ilg) *RHOSGS,* integer, dimension(ilg,6,50) *ITERCT,* real, dimension (ilg) *CDH,* real, dimension (ilg) *CDM,* real, dimension (ilg) *QSENS,* real, dimension (ilg) *TFLUX,* real, dimension (ilg) *QEVAP,* real, dimension (ilg) *EVAP,* real, dimension(ilg) *EVPPOT,* real, dimension (ilg) *ACOND,* real, dimension (ilg) *EVAPB,* real, dimension (ilg) *GT,* real, dimension (ilg) *QG,* real, dimension (ilg) *ST,* real, dimension (ilg) *SU,* real, dimension (ilg) *SV,* real, dimension (ilg) *SQ,* real, dimension (ilg) *SRH,* real, dimension (ilg) *GTBS,* real, dimension(ilg) *SFCUBS,* real, dimension(ilg) *SFCVBS,* real, dimension(ilg) *USTARBS,* real, dimension (ilg) *FSGV,* real, dimension (ilg) *FSGS,* real, dimension (ilg) *FSGG,* real, dimension (ilg) *FLGV,* real, dimension (ilg) *FLGS,* real, dimension (ilg) *FLGG,* real, dimension (ilg) *HFSC,* real, dimension (ilg) *HFSS,* real, dimension (ilg) *HFSG,* real, dimension (ilg) *HEVC,* real, dimension (ilg) *HEVS,* real, dimension (ilg) *HEVG,* real, dimension (ilg) *HMFC,* real, dimension (ilg) *HMFN,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *QFCF,* real, dimension (ilg) *QFCL,* real, dimension (ilg) *DRAG,* real, dimension(ilg) *WTABLE,* real, dimension (ilg) *ILMO,* real, dimension (ilg) *UE,* real, dimension (ilg) *HBL,* real, dimension (ilg) *TAC,* real, dimension (ilg) *QAC,* real, dimension (ilg) *ZREFM,* real, dimension (ilg) *ZREFH,* real, dimension(ilg) *ZDIAGM,* real, dimension(ilg) *ZDIAGH,* real, dimension (ilg) *VPD,* real, dimension (ilg) *TADP,* real, dimension(ilg) *RHOAIR,* real, dimension(ilg) *QSWINV,* real, dimension(ilg) *QSWINI,* real, dimension (ilg) *QLWIN,* real, dimension (ilg) *UWIND,* real, dimension (ilg) *VWIND,* real, dimension (ilg) *TA,* real, dimension (ilg) *QA,* real, dimension (ilg) *PADRY,* real, dimension (ilg) *FC,* real, dimension (ilg) *FG,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FGS,* real, dimension(ilg) *RBCOEF,* real, dimension (ilg) *FSVF,* real, dimension (ilg) *FSVFS,* real, dimension(ilg) *PRESSG,* real, dimension (ilg) *VMOD,* real, dimension(ilg) *ALVSCN,* real, dimension(ilg) *ALIRCN,* real, dimension (ilg) *ALVSG,* real, dimension (ilg) *ALIRG,* real, dimension(ilg) *ALVSCS,* real, dimension(ilg) *ALIRCS,* real, dimension(ilg) *ALVSSN,* real, dimension(ilg) *ALIRSN,* real, dimension(ilg) *ALVSGC,* real, dimension(ilg) *ALIRGC,* real, dimension(ilg) *ALVSSC,* real, dimension(ilg) *ALIRSC,* real, dimension(ilg) *TRVSCN,* real, dimension(ilg) *TRIRCN,* real, dimension(ilg) *TRVSCS,* real, dimension(ilg) *TRIRCS,* real, dimension (ilg) *RC,* real, dimension (ilg) *RCS,* real, dimension (ilg) *WTRG,* real, dimension(ilg) *QLWAVG,* real, dimension(ilg) *FRAINC,* real, dimension(ilg) *FSNOWC,* real, dimension(ilg) *FRAICS,* real, dimension(ilg) *FSNOCS,* real, dimension(ilg) *CMASSC,* real, dimension(ilg) *CMASCS,* real, dimension (ilg) *DISP,* real, dimension (ilg) *DISPS,* real, dimension(ilg) *ZOMLNC,* real, dimension(ilg) *ZOELNC,* real, dimension(ilg) *ZOMLNG,* real, dimension(ilg) *ZOELNG,* real, dimension(ilg) *ZOMLCS,* real, dimension(ilg) *ZOELCS,* real, dimension(ilg) *ZOMLNS,* real, dimension(ilg) *ZOELNS,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg) *TPOND,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TBASE,* real, dimension (ilg) *TCAN,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *WSNOW,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLRET,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *THFC,* real, dimension (ilg,ig) *THLW,* real, dimension(ilg) *TRSNOWC,* real, dimension(ilg,nbs) *TRSNOWG,* real, dimension(ilg,nbs) *ALSNO,* real, dimension(ilg,nbs) *FSSB,* real, dimension (ilg,ig) *FROOT,* real, dimension(ilg,ig) *FROOTS,* real, dimension (ilg) *RADJ,* real, dimension (ilg) *PCPR,* real, dimension (ilg,ig) *HCPS,* real, dimension (ilg,ig) *TCS,* real, dimension(ilg,4) *TSFSAV,* real, dimension (ig) *DELZ,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *ZBOTW,* real, dimension (ilg) *FTEMP,* real, dimension (ilg) *FVAP,* real, dimension (ilg) *RIB,* integer, dimension (ilg,ig) *ISAND,* real, dimension(ilg,ictem) *AILCG,* real, dimension(ilg,ictem) *AILCGS,* real, dimension(ilg,ictem) *FCANC,* real, dimension(ilg,ictem) *FCANCS,* real, dimension(ilg) *CO2CONC,* real, dimension(ilg,ictem) *CO2I1CG,* real, dimension(ilg,ictem) *CO2I1CS,* real, dimension(ilg,ictem) *CO2I2CG,* real, dimension(ilg,ictem) *CO2I2CS,* real, dimension(ilg) *COSZS,* real, dimension(ilg) *XDIFFUS,* real, dimension(ilg,ictem) *SLAI,* integer *ICTEM,* logical *ctem_on,* real, dimension(ilg,ictem,ig) *RMATCTEM,* real, dimension(ilg,ictem) *FCANCMX,* integer *L2MAX,* integer, dimension(ic) *NOL2PFTS,* real, dimension(ilg) *CFLUXCG,* real, dimension(ilg) *CFLUXCS,* real, dimension(ilg,ictem) *ANCSVEG,* real, dimension(ilg,ictem) *ANCGVEG,* real, dimension(ilg,ictem) *RMLCSVEG,* real, dimension(ilg,ictem) *RMLCGVEG,* real, dimension(ilg) *TCSNOW,* real, dimension(ilg) *GSNOW,* integer *ITC,* integer *ITCG,* integer *ITG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N,* integer *IC,* integer *IG,* integer *IZREF,* integer *ISLFD,* integer *NLANDCS,* integer *NLANDGS,* integer *NLANDC,* integer *NLANDG,* integer *NLANDI,* integer *NBS,* integer *ISNOALB,* integer, dimension(ilg,ictem) *LFSTATUS,* real, dimension(ilg) *DAYL,* real, dimension(ilg) *DAYL_MAX*)

Parameters

nlandcs	Number of modelled areas that contain subareas of canopy over bare ground / bare ground / canopy over snow /snow
nlandgs	Number of modelled areas that contain subareas of canopy over bare ground / bare ground / canopy over snow /snow
nlandc	Number of modelled areas that contain subareas of canopy over bare ground / bare ground / canopy over snow /snow
nlandg	Number of modelled areas that contain subareas of canopy over bare ground / bare ground / canopy over snow /snow
nlandi	Number of modelled areas that are ice sheets [ ]
itc	Flag to select iteration scheme for canopy temperature
itcg	Flag to select iteration scheme for surface under canopy
itg	Flag to select iteration scheme for ground or snow surface
izref	Flag governing treatment of surface roughness length
islfd	Flag governing options for surface stability functions and diagnostic calculations
acond	Diagnosed product of drag coefficient and wind speed over modelled area $[ms^{-1}]$
cdh	Surface drag coefficient for heat $[](C_{DH})$
cdm	Surface drag coefficient for momentum $[](C_{DM})$
chcap	Surface drag coefficient for momentum $[](C_{DM})$
chcaps	Heat capacity of canopy over bare ground $[Jm^{-2}K^{-1}]$
drag	Surface drag coefficient under neutral stability [ ]
evap	Diagnosed total surface water vapour flux over modelled area $[kgm^{-2}s^{-1}]$
evapb	Evaporation efficiency at ground surface [ ]
evapc	Evaporation from vegetation over ground $[ms^{-1}]$
evapcg	Evaporation from ground under vegetation $[ms^{-1}]$
evapcs	Evaporation from vegetation over snow $[ms^{-1}]$
evapg	Evaporation from bare ground $[ms^{-1}]$
evapgs	Evaporation from snow on bare ground $[ms^{-1}]$
evpcsg	Evaporation from snow under vegetation $[ms^{-1}]$
evppot	Diagnosed potential evapotranspiration $[kgm^{-2}s^{-1}](E_p)$
flgg	Diagnosed net longwave radiation at soil surface $[Wm^{-2}]$
flgs	Diagnosed net longwave radiation at snow surface $[Wm^{-2}]$
flgv	Diagnosed net longwave radiation on vegetation canopy $[Wm^{-2}]$
fsgg	Diagnosed net shortwave radiation at soil surface $[Wm^{-2}]$
fsgs	Diagnosed net shortwave radiation at snow surface $[Wm^{-2}]$
fsgv	Diagnosed net shortwave radiation on vegetation canopy $[Wm^{-2}]$
g12c	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g12cs	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g12g	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g12gs	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g23c	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
g23cs	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
g23g	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
g23gs	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
gt	Diagnosed effective surface black-body temperature $[K](T_{0,eff})$
gzeroc	Subarea heat flux at soil surface $[Wm^{-2}]$

gzerog	Subarea heat flux at soil surface $[Wm^{-2}]$
gzrocs	Subarea heat flux at soil surface $[Wm^{-2}]$
gzrogs	Subarea heat flux at soil surface $[Wm^{-2}]$
hbl	Height of the atmospheric boundary layer [m]
hcpc	Heat capacity of soil layers under vegetation $[Jm^{-3}K^{-1}]$
hcpg	Heat capacity of soil layers in bare areas $[Jm^{-3}K^{-1}]$
hevc	Diagnosed latent heat flux on vegetation canopy $[Wm^{-2}]$
hevg	Diagnosed latent heat flux at soil surface $[Wm^{-2}]$
hevs	Diagnosed latent heat flux at snow surface $[Wm^{-2}]$
hfsc	Diagnosed sensible heat flux on vegetation canopy $[Wm^{-2}]$
hfsg	Diagnosed sensible heat flux at soil surface $[Wm^{-2}]$
hfss	Diagnosed sensible heat flux at snow surface $[Wm^{-2}]$
hmfc	Diagnosed energy associated with phase change of water on vegetation $[Wm^{-2}]$
hmfn	Diagnosed energy associated with phase change of water in snow pack $[Wm^{-2}]$
htc	Diagnosed internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}]$
htcc	Diagnosed internal energy change of vegetation canopy due to conduction and/or change in mass $[Wm^{-2}]$
htcs	Diagnosed internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}]$
ilmo	Inverse of Monin-Obukhov roughness length (m^{-1})
iterct	Counter of number of iterations required to solve surface energy balance for the elements of the four subareas
qac	Specific humidity of air within vegetation canopy space $[kgkg^{-1}]$
qevap	Diagnosed total surface latent heat flux over modelled area $[Wm^{-2}]$
qfcf	Sublimation from frozen water on vegetation $[kgm^{-2}s^{-1}]$
qfcl	Evaporation from liquid water on vegetation $[kgm^{-2}s^{-1}]$
qfrezc	Heat sink to be used for freezing water on ground under canopy $[Wm^{-2}]$
qfrezg	Heat sink to be used for freezing water on bare ground $[Wm^{-2}]$
qg	Diagnosed surface specific humidity $[kgkg^{-1}]$
qlwavg	Upwelling longwave radiation from land surface $[Wm^{-2}]$
qmeltc	Heat to be used for melting snow under canopy $[Wm^{-2}]$
qmeltg	Heat to be used for melting snow on bare ground $[Wm^{-2}]$
qsens	Diagnosed total surface sensible heat flux over modelled area $[Wm^{-2}]$
raican	Intercepted liquid water stored on canopy over ground $[kgm^{-2}]$
raicns	Intercepted liquid water stored on canopy over snow $[kgm^{-2}]$
rhoscs	Density of snow under vegetation $[kgm^{-3}]$
rhosgs	Density of snow in bare areas $[kgm^{-3}]$
snocan	Intercepted frozen water stored on canopy over ground $[kgm^{-2}]$
snocns	Intercepted frozen water stored on canopy over snow $[kgm^{-2}]$
sq	Diagnosed screen-level specific humidity $[kgkg^{-1}]$
srh	Diagnosed screen-level relative humidity [%]
st	Diagnosed screen-level air temperature [K]
su	Diagnosed anemometer-level zonal wind $[ms^{-1}]$
sv	Diagnosed anemometer-level meridional wind $[ms^{-1}]$
tac	Temperature of air within vegetation canopy [K]
tbarc	Subarea temperatures of soil layers [C]
tbarcs	Subarea temperatures of soil layers [C]

tbarg	Subarea temperatures of soil layers [C]
tbargs	Subarea temperatures of soil layers [C]
tcano	Temperature of canopy over ground [K]
tcans	Temperature of canopy over snow [K]
tcbotc	Thermal conductivity of soil at bottom of layer $[Wm^{-1}K^{-1}]$
tcbotg	Thermal conductivity of soil at bottom of layer $[Wm^{-1}K^{-1}]$
tctopc	Thermal conductivity of soil at top of layer $[Wm^{-1}K^{-1}]$
tctopg	Thermal conductivity of soil at top of layer $[Wm^{-1}K^{-1}]$
tflux	Product of surface drag coefficient, wind speed and surface-air temperature difference $[Kms^{-1}]$
thicec	Frozen water content of soil layers under vegetation $[m^3m^{-3}]$
thiceg	Frozen water content of soil layers in bare areas $[m^3m^{-3}]$
thliqc	Liquid water content of soil layers under vegetation $[m^3m^{-3}]$
thliqg	Liquid water content of soil layers in bare areas $[m^3m^{-3}]$
tpondc	Subarea temperature of surface ponded water [C]
tpondg	Subarea temperature of surface ponded water [C]
tpndcs	Subarea temperature of surface ponded water [C]
tpndgs	Subarea temperature of surface ponded water [C]
tsfsav	Ground surface temperature over subarea [K]
tsnocs	Temperature of snow pack under vegetation [K]
tsnogs	Temperature of snow pack in bare areas [K]
ue	Friction velocity of air $[ms^{-1}]$
wsnocs	Liquid water content of snow pack under vegetation $[kgm^{-2}]$
wsnogs	Liquid water content of snow pack in bare areas $[kgm^{-2}]$
wtable	Depth of water table in soil [m]
wtrg	Diagnosed residual water transferred into or out of the soil $[kgm^{-2}s^{-1}]$
zrefm	Reference height associated with forcing wind speed [m]
zrefh	Reference height associated with forcing air temperature and humidity [m]
zdiagm	User-specified height associated with diagnosed anemometer-level wind speed [m]
zdiagh	User-specified height associated with diagnosed screen-level variables [m]
vpd	Vapour pressure deficit [mb]
tadp	Dew point temperature of air [K]
rhoair	Density of air $[kgm^{-3}](\rho_a)$
qswinv	Visible radiation incident on horizontal surface $[Wm^{-2}]$
qswini	Near-infrared radiation incident on horizontal surface $[Wm^{-2}]$
qlwin	Downwelling longwave radiation at bottom of atmosphere $[Wm^{-2}]$
uwind	Zonal component of wind speed $[ms^{-1}](U_a)$
vwind	Meridional component of wind speed $[ms^{-1}](V_a)$
ta	Air temperature at reference height $[K](T_a)$
qa	Specific humidity at reference height $[kgkg^{-1}]$
padry	Partial pressure of dry air $[Pa](p_{dry})$
fc	Subarea fractional coverage of modelled area [ ]
fg	Subarea fractional coverage of modelled area [ ]
fcs	Subarea fractional coverage of modelled area [ ]
fgs	Subarea fractional coverage of modelled area [ ]
rbcoef	Parameter for calculation of leaf boundary resistance
fsvf	Sky view factor for bare ground under canopy [ ]

fsvfs	Sky view factor for snow under canopy [ ]
pressg	Surface atmospheric pressure [Pa]
vmod	Wind speed at reference height $[ms^{-1}]$
alvscn	Visible/near-IR albedo of vegetation over bare ground [ ]
alircn	Visible/near-IR albedo of vegetation over bare ground [ ]
alvsg	Visible/near-IR albedo of open bare ground [ ]
alirg	Visible/near-IR albedo of open bare ground [ ]
alvscs	Visible/near-IR albedo of vegetation over snow [ ]
alircs	Visible/near-IR albedo of vegetation over snow [ ]
alvssn	Visible/near-IR albedo of open snow cover [ ]
alirsn	Visible/near-IR albedo of open snow cover [ ]
alvsgc	Visible/near-IR albedo of bare ground under vegetation [ ]
alirgc	Visible/near-IR albedo of bare ground under vegetation [ ]
alvssc	Visible/near-IR albedo of snow under vegetation [ ]
alirsc	Visible/near-IR albedo of snow under vegetation [ ]
trvscn	Visible/near-IR transmissivity of vegetation over bare ground [ ]
trircn	Visible/near-IR transmissivity of vegetation over bare ground [ ]
trvscs	Visible/near-IR transmissivity of vegetation over snow [ ]
trircs	Visible/near-IR transmissivity of vegetation over snow [ ]
rc	Stomatal resistance of vegetation over bare ground $[sm^{-1}]$
rcs	Stomatal resistance of vegetation over snow $[sm^{-1}]$
frainc	Fractional coverage of canopy by liquid water over snow-free subarea [ ]
fsnowc	Fractional coverage of canopy by frozen water over snow-free subarea [ ]
fraics	Fractional coverage of canopy by liquid water over snow-covered subarea [ ]
fsnocs	Fractional coverage of canopy by frozen water over snow-covered subarea [ ]
cmassc	Mass of canopy over bare ground $[kgm^{-2}]$
cmascs	Mass of canopy over snow $[kgm^{-2}]$
disp	Displacement height of vegetation over bare ground [m] (d)
disps	Displacement height of vegetation over snow [m] (d)
zomlnc	Logarithm of roughness length for momentum of vegetation over bare ground [ ]
zoelnc	Logarithm of roughness length for heat of vegetation over bare ground [ ]
zomlng	Logarithm of roughness length for momentum of bare ground [ ]
zoelng	Logarithm of roughness length for heat of bare ground [ ]
zomlcs	Logarithm of roughness length for momentum of vegetation over snow [ ]
zoelcs	Logarithm of roughness length for heat of vegetation over snow [ ]
zomlns	Logarithm of roughness length for momentum of snow [ ]
zoelns	Logarithm of roughness length for heat of snow [ ]
tpond	Temperature of ponded water [K]
zpond	Depth of ponded water on surface [m]
tbase	Temperature of bedrock in third soil layer [K]
tcan	Vegetation canopy temperature [K]
tsnow	Snowpack temperature [K]
zsnow	Depth of snow pack [m]
rhosno	Density of snow $[kgm^{-3}]$
wsnow	Liquid water content of snow pack $[kgm^{-2}]$
radj	Latitude of grid cell (positive north of equator) [rad] (φ)
pcpr	Surface precipitation rate $[kgm^{-2}s^{-1}]$

tbar	Temperature of soil layers [K]
thliq	Volumetric liquid water content of soil layers $[m^3m^{-3}]$
thice	Volumetric frozen water content of soil layers $[m^3m^{-3}]$
thpor	Pore volume in soil layer $[m^3m^{-3}]$
thlret	Liquid water retention capacity for organic soil $[m^3m^{-3}]$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}]$
thfc	Field capacity $[m^3m^{-3}]$
hcps	Heat capacity of soil material $[Jm^{-3}K^{-1}]$
tcs	Thermal conductivity of soil particles $[Wm^{-1}K^{-1}]$
delz	Overall thickness of soil layer [m]
delzw	Permeable thickness of soil layer [m]
zbotw	Depth to permeable bottom of soil layer [m]
isand	Sand content flag
ailcg	GREEN LAI FOR USE WITH PHOTOSYNTHESIS SUBTROUTINE FOR CANOPY OVER GROUND SUBAREA
ailcgs	GREEN LAI FOR USE WITH PHOTOSYNTHESIS SUBTROUTINE FOR CANOPY OVER SNOW SUBAREA
fcanc	FRACTIONAL COVERAGE OF 8 CARBON PFTs, CANOPY OVER GROUND
fcancs	FRACTIONAL COVERAGE OF 8 CARBON PFTs, CANOPY OVER SNOW
co2conc	ATMOS. CO2 CONC. IN PPM
co2i1cg	INTERCELLULAR CO2 CONC FOR 8 PFTs FOR CANOPY OVER GROUND SUBAREA (Pa) - FOR SINGLE/SUNLIT LEAF
co2i1cs	SAME AS ABOVE BUT FOR SHADED LEAF
co2i2cg	INTERCELLULAR CO2 CONC FOR 8 PFTs FOR CANOPY OVER SNOWSUBAREA (Pa) - FOR SINGLE/SUNLIT LEAF
co2i2cs	SAME AS ABOVE BUT FOR SHADED LEAF
coszs	COSINE OF SUN'S ZENITH ANGLE
xdiffus	FRACTION OF DIFFUSED RADIATION
slai	STORAGE LAI. SEE PHTSYN SUBROUTINE FOR MORE DETAILS.
rmatctem	FRACTION OF ROOTS IN EACH SOIL LAYER FOR EACH OF CTEM's 8 PFTs
fcancmx	MAX. FRACTIONAL COVERAGE OF CTEM PFTs
ancsveg	NET PHOTOSYNTHETIC RATE FOR CTEM's 8 PFTs FOR CANOPY OVER SNOW SUB↩ AREA
ancgveg	NET PHOTOSYNTHETIC RATE FOR CTEM's 8 PFTs FOR CANOPY OVER GROUND S↩ UBAREA
rmlcsveg	LEAF RESPIRATION RATE FOR CTEM's 8 PFTs FOR CANOPY OVER SNOW SUBAREA
rmlcgveg	LEAF RESPIRATION RATE FOR CTEM's 8 PFTs FOR CANOPY OVER GROUND SUBA↩ REA
ictem	8 (CTEM's PLANT FUNCTIONAL TYPES)
ctem_on	TRUE GIVES COUPLING TO CTEM
lfstatus	LEAF PHENOLOGICAL STATUS (SEE PHENOLOGY)
dayl_max	MAXIMUM DAYLENGTH FOR THAT LOCATION
dayl	DAYLENGTH FOR THAT LOCATION

First, two parameters are calculated for later use in the CLASST subroutines: the corrected wind speed v_a , and the Coriolis parameter f_{cor} (describing the effect of the earth's rotation on the movement of air according to the reference frame of the earth's surface). The wind speed correction is applied because it is assumed that air is never completely still, so v_a is specified as the maximum of VMOD and a limiting lower value of $0.1ms^{-1}$. The Coriolis parameter is calculated from the angular velocity Ω of the earth's rotation (7.29 x 10 -5 radians/s), and the latitude φ: $f_{cor} = 2\Omega sin\varphi$

The packing and unpacking of binary files may cause small shifts in the values of variables at certain points in the model run, so checks are performed on the depth of ponded water and on the soil liquid and frozen moisture contents to ensure that unphysical values have not occurred. If the ponded water depth is vanishingly small or less than zero, it is set to zero. If the soil liquid water content falls below the set minimum value, it is set to the minimum value. If the soil frozen water content is less than zero, it is set back to zero. If the sum of the liquid water content and frozen water content converted to an equivalent liquid amount is greater than the pore volume, both are re-normalized by the pore volume. (This treatment of frozen water is employed in recognition of the fact that water

expands upon freezing and can therefore occupy a greater volume than the nominal pore volume of the soil.) The small changes in internal energy and water content of the soil layers resulting from these operations are accounted for by updating the diagnostic variables HTC and WTRG.

If CLASS is being run in coupled mode with CTEM, several CTEM variables are initialized to zero. Subroutine TPREP is then called, to carry out the initialization of a number of variables and to do preparatory calculations of various parameters associated with the energy and water budget calculations.

The energy budget calculations that follow are performed over the four subareas, of canopy over snow (CS), snow over ground (GS), canopy over bare ground (C) and bare ground (G). First, a counter NLANDx is defined for each subarea, containing the number of modelled areas in which the subarea occurs. The subarea calculations are only done if the relevant counter is greater than zero. A counter NLANDI is also set to the number of modelled areas that are ice sheets (indicated by ISAND = -4). (This is used later in subroutine CLASSW to toggle the call to subroutine ICEBAL.)

For each subarea, the roughness lengths for momentum and heat, ZOM and ZOH, are obtained from their logarithms, and are then used to determine various height intervals in the atmosphere for subsequent calculations. These heights are derived from the input variables ZREFM and ZREFH, the reference heights corresponding to the input values of wind speed and of air temperature and humidity respectively, and ZDIAGM and ZDIAGH, the heights at which the diagnostic values of anemometer wind speed and screen level temperature and humidity are to be determined. The form of the calculations depends on the value of the flag IZREF. If IZREF = 1, the zero plane is taken to lie at the physical ground surface (as with field measurements); if IZREF = 2, the zero plane is taken to lie at the local roughness height for momentum(as with atmospheric models). The variables ZRSLDM and ZRSLDH are the height differences used in subroutine DRCOEF to express the interval between the conceptual bottom of the atmosphere and the reference heights for wind speed and for temperature and humidity respectively; the variables ZRSLFM and ZRSLFH are the corresponding height differences used in subroutine FLXSURFZ. If IZREF = 1, ZRSLDM and ZRSLDH are set to ZREFM and ZREFH minus the displacement height DISP or DISPS, and ZRSLFM and ZRSLFH are set to ZREFM and ZREFH minus the roughness length ZOM and DISP or DISPS. If IZREF = 2, ZRSLDM and ZRSLDH are set to ZREFM and ZREFH plus the roughness height ZOM, and ZRSLFM and ZRSLFH are set to ZREFM and ZREFH minus DISP or DISPS. (In the absence of a vegetation canopy, the displacement height is zero.) The variables ZDSLM and ZDSLH are the heights above the bottom of the modelled atmosphere at which the diagnostic values are to be calculated. If IZREF = 1, they are set to ZDIAGM and ZDI↩ AGH minus ZOM; if IZREF = 2 they are simply set to ZDIAGM and ZDIAGH. At the end of the branch in the code, the ratios ZOSCLM and ZOSCLH, which are used in subroutine DRCOEF, are calculated as ZOM/ZRSLDM and ZOH/ZRSLDH respectively.

Several other local parameters are also calculated. The potential temperature is the temperature that air would have if brought adiabatically (without addition or removal of heat) to a given height. The potential temperature of the air at the reference height, $T_{a,pot}$, is calculated relative to the height at which the horizontal wind speed goes to zero, using the dry adiabatic lapse rate, $dT/dz = -g/c_p$, where g is the acceleration due to gravity and c_p is the specific heat at constant pressure:. Thus, $T_{a,pot} = T_a + z_{ref,h}g/c_p$ where T_a is the temperature of the air at the reference height and $z_{ref,h}$ is the height interval, equivalent to ZRSLFH defined above. If CLASS is being run coupled to an atmospheric model, i.e. if IZREF=2, the air temperature at the reference height has already been adiabatically extrapolated before being passed to CLASS. Otherwise, the correction is performed using the above equation.

The virtual potential temperature of the air at the reference height, $T_{a,v}$, is the potential temperature adjusted for the reduction in air density due to the presence of water vapour. This is applied in order to enable the use of the equation of state for dry air. $T_{a,v}$ can be approximated as: $T_{a,v} = T_{a,pot}[1 + 0.61q_a]$ where q_a is the specific humidity of the air at the reference height.

The bulk Richardson number Ri_B , used in the calculations of the atmospheric stability functions in subroutine DRCOEF, is formulated as: $Ri_B = [T_0 \check{} T_{a,v}](-gz_{ref})/(T_{a,v}v_a^2)$ where T_0 is the surface temperature. For ease of calculation later on, the factor multiplying $[T_0 \check{} T_{a,v}]$ on the right-hand side of the above equation is evaluated and assigned to a coefficient CRIB, using ZRSLDM for z_{ref} . The drag coefficient under neutral stability, C_{DN} , is expressed using basic flux-gradient analysis as: $C_{DN} = k^2/[ln(z_{ref}) \check{} ln(z_0)]^2$ where k is the von Karman constant and z_0 is the roughness length. ZRSLDM is used for z_{ref} and the logarithm of the local roughness length for $ln(z_0)$, and the neutral drag coefficient DRAG over the modelled area is obtained as a weighted average over the four subareas.

For the two subareas with canopy cover, the wind speed of the air at the canopy top, $v_{a,c}$, is obtained by applying the classic logarithmic wind law for the wind speed v(z) at a height z: $kv(z)/v_* = ln[(z \check{} d)/z_0]$ where v_* is the friction velocity and d is the displacement height. Thus, $v_{a,c}$ at the canopy height H can be related to v a at the

reference height z_{ref} as: $v_{a,c} = v_a[ln(H \check{} d) \check{} ln(z_0)]/[ln(z_{ref}) \check{} ln(z_0)]$

The vegetation height is calculated as $10z_0$. Local values of the temperature of the canopy air TAC and the humidity of the canopy air QAC are assigned to variables TACCS/TACCO and QACCS/QACCO respectively.

At this point calls are made to a series of subroutines addressing the calculation of the energy balance components of the subarea in question. The calls are summarized in the table below.

		CS	GS	C	G
CWCALC	Freezing/thawing of liquid/frozen water on canopy	YES		YES	
TNPREP	Set coefficients for temperature calculations in soil	YES	YES	YES	YES
TSPREP	Set coefficients for temperature calculations of snow	YES	YES		
TSOLVC	Calculate components of canopy energy balance	YES		YES	
TSOLVE	Calculate components of ground or snow energy balance		YES		YES
TSPOST	Heat conduction in snow pack	YES	YES		
TNPOST	Heat conduction in soil	YES	YES	YES	YES

After these calls, various diagnostic calculations are performed. First the screen-level temperature and humidity, and the anemometer-level zonal and meridional wind speeds, are calculated. Three options are provided for doing this, indicated by the flag ISLFD. If ISLFD = 0, the simple approach used in the Canadian GCM is selected. The ratio of the square root of the surface drag coefficient for momentum, C_{DM} , to that of the neutral drag coefficient C_{DN} , is calculated for the screen height z_s (RATFC1) and the anemometer height (RATFCA1), to give a measure of the degree of atmospheric instability. If the bulk Richardson number RIB is positive (indicating stable conditions), RATFC1 is adopted for the screen-level calculations; if RIB is negative (indicating unstable conditions), the ratio used is the minimum of the ratio of the drag coefficient for heat C_{DH} to C_{DN} , and $(z_s/z_{ref})^{1/3}$, a measure of the depth of the convection. These ratios are applied to the calculation of the screen and anemometer level variables. If the ratios are large, indicating strong coupling with the atmosphere, the screen level variables tend to the values at the reference height; if the ratio is small, they tend to the values at the surface. At the end of the loop, the CCCma subroutine SCREENRH is called to evaluate the screen-level relative humidity.

If ISLFD= 1 or 2, the more rigorous calculations in subroutines SLDIAG and DIASURFZ are followed. The calculations done in SLDIAG are consistent with the modelling approach used in subroutine DRCOEF to determine the atmospheric stability functions, so when ISLFD = 1, DRCOEF and SLDIAG are called. The calculations done in DIASURFZ are consistent with the modelling approach used in subroutine FLXSURFZ for the atmospheric stability functions, so when ISLFD = 2, FLXSURFZ and DIASURFZ are called.

A number of additional diagnostic variables are calculated as weighted averages over the four subareas. For the most part, these calculations are straightforward; only the calculation of the porential evapotranspiration E_p (EVPPOT) involves some complexity. E_p is defined as the evapotranspiration that would occur under ambient atmospheric conditions if the soil were completely saturated and the vegetation canopy were completely water-covered, i.e. if there were no surface resistances to evaporation: $E_p = \rho_a C_{DH} v_a [q_{0,sat} \check{} q_a]$ where ρ_a is the density of air and $q_{0,sat}$ is the saturated specific humidity at the surface. For the ground or snow surface $q_{0,sat}$ was calculated in subroutine TSOLVE. For the canopy, the saturated specific humidity at the canopy air temperature, $q_{ac,sat}$, is used. This is obtained from the mixing ratio at saturation, $w_{ac,sat}$: $q_{ac,sat} = w_{ac,sat}/[1 + w_{ac,sat}]$

The mixing ratio is a function of the saturation vapour pressure $e_{ac,sat}$ at the canopy air temperature: $w_{ac,sat} = 0.622 e_{ac,sat}/(p_{dry})$

A standard empirical equation for the saturation vapour pressure dependence on the temperature T is used:

$$e_{sat} = 611.0 exp[17.269(T \check{} T_f)/(T \check{} 35.86)] T \geq T_f$$

$$e_{sat} = 611.0 exp[21.874(T \check{} T_f)/(T \check{} 7.66)] T < T_f$$

where T_f is the freezing point.

At the end of the code dealing with the four subareas, several more diagnostic variables are evaluated. Again, these calculations are generally straightforward. The effective black-body surface temperature $T_{0,eff}$ is obtained by inverting the Stefan-Boltzmann equation: $L \uparrow = \sigma T_{0,eff}^4$ where $L \uparrow$ is the outgoing longwave radiation and σ is the Stefan-Boltzmann constant. The evaporation efficiency parameter EVAPB is calculated as the ratio of the actual evapotranspiration to the potential evapotranspiration.

9.18 CLASSW.f File Reference

Purpose: Call subroutines to perform surface water budget calculations.

Functions/Subroutines

- subroutine classw (THLIQ, THICE, TBAR, TCAN, RCAN, SNCAN, RUNOFF, TRUNOF, SNO, TSNOW, R↩ HOSNO, ALBSNO, WSNOW, ZPOND, TPOND, GROWTH, TBASE, GFLUX, PCFC, PCLC, PCPN, PCPG, QFCF, QFCL, QFN, QFG, QFC, HMFC, HMFG, HMFN, HTCC, HTCS, HTC, ROFC, ROFN, ROVG, WTR↩ S, WTRG, OVRFLW, SUBFLW, BASFLW, TOVRFL, TSUBFL, TBASFL, EVAP, QFLUX, RHOAIR, TBARC, TBARG, TBARCS, TBARGS, THLIQC, THLIQG, THICEC, THICEG, HCPC, HCPG, RPCP, TRPCP, SPCP, TSPCP, PCPR, TA, RHOSNI, GGEO, FC, FG, FCS, FGS, TPONDC, TPONDG, TPNDCS, TPNDGS, EVA↩ PC, EVAPCG, EVAPG, EVAPCS, EVPCSG, EVAPGS, QFREZC, QFREZG, QMELTC, QMELTG, RAICAN, SNOCAN, RAICNS, SNOCNS, FSVF, FSVFS, CWLCAP, CWFCAP, CWLCPS, CWFCPS, TCANO, TCANS, CHCAP, CHCAPS, CMASSC, CMASCS, ZSNOW, GZEROC, GZEROG, GZROCS, GZROGS, G12C, G12↩ G, G12CS, G12GS, G23C, G23G, G23CS, G23GS, TSNOCS, TSNOGS, WSNOCS, WSNOGS, RHOSCS, RHOSGS, ZPLIMC, ZPLIMG, ZPLMCS, ZPLMGS, TSFSAV, TCTOPC, TCBOTC, TCTOPG, TCBOTG, FR↩ OOT, FROOTS, THPOR, THLRET, THLMIN, BI, PSISAT, GRKSAT, THLRAT, THFC, XDRAIN, HCPS, DELZ, DELZW, ZBOTW, XSLOPE, GRKFAC, WFSURF, WFCINT, ISAND, IGDR, IWF, ILG, IL1, IL2, N, JL, IC, IG, IGP1, IGP2, NLANDCS, NLANDGS, NLANDC, NLANDG, NLANDI)

9.18.1 Detailed Description

Purpose: Call subroutines to perform surface water budget calculations.

9.18.2 Function/Subroutine Documentation

9.18.2.1 subroutine classw (real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg) *TCAN,* real, dimension (ilg) *RCAN,* real, dimension (ilg) *SNCAN,* real, dimension(ilg) *RUNOFF,* real, dimension(ilg) *TRUNOF,* real, dimension (ilg) *SNO,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension(ilg) *ALBSNO,* real, dimension (ilg) *WSNOW,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TPOND,* real, dimension(ilg) *GROWTH,* real, dimension (ilg) *TBASE,* real, dimension (ilg,ig) *GFLUX,* real, dimension (ilg) *PCFC,* real, dimension (ilg) *PCLC,* real, dimension (ilg) *PCPN,* real, dimension (ilg) *PCPG,* real, dimension (ilg) *QFCF,* real, dimension (ilg) *QFCL,* real, dimension (ilg) *QFN,* real, dimension (ilg) *QFG,* real, dimension (ilg,ig) *QFC,* real, dimension (ilg) *HMFC,* real, dimension (ilg,ig) *HMFG,* real, dimension (ilg) *HMFN,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *ROFC,* real, dimension (ilg) *ROFN,* real, dimension (ilg) *ROVG,* real, dimension (ilg) *WTRS,* real, dimension (ilg) *WTRG,* real, dimension(ilg) *OVRFLW,* real, dimension(ilg) *SUBFLW,* real, dimension(ilg) *BASFLW,* real, dimension(ilg) *TOVRFL,* real, dimension(ilg) *TSUBFL,* real, dimension(ilg) *TBASFL,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *QFLUX,* real, dimension (ilg) *RHOAIR,* real, dimension(ilg,ig) *TBARC,* real, dimension(ilg,ig) *TBARG,* real, dimension(ilg,ig) *TBARCS,* real, dimension(ilg,ig) *TBARGS,* real, dimension(ilg,ig) *THLIQC,* real, dimension(ilg,ig) *THLIQG,* real, dimension(ilg,ig) *THICEC,* real, dimension(ilg,ig) *THICEG,* real, dimension (ilg,ig) *HCPC,* real, dimension (ilg,ig) *HCPG,* real, dimension (ilg) *RPCP,* real, dimension (ilg) *TRPCP,* real, dimension (ilg) *SPCP,* real, dimension (ilg) *TSPCP,* real, dimension (ilg) *PCPR,* real, dimension (ilg) *TA,* real, dimension(ilg) *RHOSNI,* real, dimension (ilg) *GGEO,* real, dimension (ilg) *FC,* real, dimension (ilg) *FG,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FGS,* real, dimension(ilg) *TPONDC,* real, dimension(ilg) *TPONDG,* real, dimension(ilg) *TPNDCS,* real, dimension(ilg) *TPNDGS,* real, dimension (ilg) *EVAPC,* real, dimension(ilg) *EVAPCG,* real, dimension (ilg) *EVAPG,* real, dimension(ilg) *EVAPCS,* real, dimension(ilg) *EVPCSG,* real, dimension(ilg) *EVAPGS,* real, dimension(ilg) *QFREZC,* real, dimension(ilg) *QFREZG,* real, dimension(ilg) *QMELTC,* real, dimension(ilg) *QMELTG,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension(ilg) *RAICNS,* real, dimension(ilg) *SNOCNS,* real, dimension (ilg) *FSVF,* real, dimension (ilg) *FSVFS,* real, dimension(ilg) *CWLCAP,* real, dimension(ilg) *CWFCAP,* real, dimension(ilg) *CWLCPS,* real, dimension(ilg) *CWFCPS,* real, dimension (ilg) *TCANO,* real, dimension (ilg) *TCANS,* real, dimension (ilg) *CHCAP,* real, dimension(ilg) *CHCAPS,* real, dimension(ilg) *CMASSC,* real, dimension(ilg) *CMASCS,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *GZEROC,* real, dimension(ilg) *GZEROG,* real, dimension(ilg) *GZROCS,* real, dimension(ilg) *GZROGS,* real, dimension (ilg) *G12C,* real, dimension (ilg) *G12G,* real, dimension (ilg) *G12CS,* real, dimension (ilg) *G12GS,* real, dimension (ilg) *G23C,* real, dimension (ilg) *G23G,* real, dimension (ilg) *G23CS,* real, dimension (ilg) *G23GS,* real, dimension(ilg) *TSNOCS,* real, dimension(ilg) *TSNOGS,* real, dimension(ilg) *WSNOCS,* real, dimension(ilg) *WSNOGS,* real, dimension(ilg) *RHOSCS,* real, dimension(ilg) *RHOSGS,* real, dimension(ilg) *ZPLIMC,* real, dimension(ilg) *ZPLIMG,* real, dimension(ilg) *ZPLMCS,* real, dimension(ilg) *ZPLMGS,* real, dimension(ilg,4) *TSFSAV,* real, dimension(ilg,ig) *TCTOPC,* real, dimension(ilg,ig) *TCBOTC,* real, dimension(ilg,ig) *TCTOPG,* real, dimension(ilg,ig) *TCBOTG,* real, dimension (ilg,ig) *FROOT,* real, dimension (ilg,ig) *FROOTS,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLRET,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *BI,* real, dimension(ilg,ig) *PSISAT,* real, dimension(ilg,ig) *GRKSAT,* real, dimension(ilg,ig) *THLRAT,* real, dimension (ilg,ig) *THFC,* real, dimension(ilg) *XDRAIN,* real, dimension (ilg,ig) *HCPS,* real, dimension (ig) *DELZ,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *ZBOTW,* real, dimension(ilg) *XSLOPE,* real, dimension(ilg) *GRKFAC,* real, dimension(ilg) *WFSURF,* real, dimension(ilg) *WFCINT,* integer, dimension(ilg,ig) *ISAND,* integer, dimension (ilg) *IGDR,* integer *IWF,* integer *ILG,* integer *IL1,* integer *IL2,* integer *N,* integer *JL,* integer *IC,* integer *IG,* integer *IGP1,* integer *IGP2,* integer *NLANDCS,* integer *NLANDGS,* integer *NLANDC,* integer *NLANDG,* integer *NLANDI*)

Parameters

iwf	Flag governing lateral soil water flow calculations
nlandcs	Number of modelled areas that contain subareas of canopy over snow
nlandgs	Number of modelled areas that contain subareas of snow
nlandc	Number of modelled areas that contain subareas of canopy over bare ground
nlandg	Number of modelled areas that contain subareas of bare ground
nlandi	Number of modelled areas that are ice sheets [ ]
thliq	Volumetric liquid water content of soil layers $[m^3 m^{-3}]$
thice	Volumetric frozen water content of soil layers $[m^3 m^{-3}]$
tbar	Temperature of soil layers [K]
gflux	Heat flux at interfaces between soil layers $[W m^{-2}]$
tcan	Vegetation canopy temperature [K]
rcan	Intercepted liquid water stored on canopy $[kg m^{-2}]$
sncan	Intercepted frozen water stored on canopy $[kg m^{-2}]$
runoff	Total runoff from soil $[mor kg m^{-2} s^{-1}]$
sno	Mass of snow pack $[kg m^{-2}]$
tsnow	Snowpack temperature [K]
rhosno	Density of snow $[kg m^{-3}]$
albsno	Snow albedo [ ]
zpond	Depth of ponded water on surface [m]
tpond	Temperature of ponded water [K]
growth	Vegetation growth index [ ]
tbase	Temperature of bedrock in third soil layer [K]
trunof	Temperature of total runoff [K]
wsnow	Liquid water content of snow pack $[kg m^{-2}]$
pcfc	Frozen precipitation intercepted by vegetation $[kg m^{-2} s^{-1}]$
pclc	Liquid precipitation intercepted by vegetation $[kg m^{-2} s^{-1}]$
pcpn	Precipitation incident on snow pack $[kg m^{-2} s^{-1}]$
pcpg	Precipitation incident on ground $[kg m^{-2} s^{-1}]$
qfcf	Sublimation from frozen water on vegetation $[kg m^{-2} s^{-1}]$
qfcl	Evaporation from liquid water on vegetation $[kg m^{-2} s^{-1}]$
qfn	Sublimation from snow pack $[kg m^{-2} s^{-1}]$
qfg	Evaporation from ground $[kg m^{-2} s^{-1}]$
hmfc	Diagnosed energy associated with phase change of water on vegetation $[W m^{-2}]$
hmfn	Diagnosed energy associated with phase change of water in snow pack $[W m^{-2}]$
htcc	Diagnosed internal energy change of vegetation canopy due to conduction and/or change in mass $[W m^{-2}]$
htcs	Diagnosed internal energy change of snow pack due to conduction and/or change in mass $[W m^{-2}]$
rofc	Liquid/frozen water runoff from vegetation $[kg m^{-2} s^{-1}]$
rofn	Liquid water runoff from snow pack $[kg m^{-2} s^{-1}]$
rovg	Liquid/frozen water runoff from vegetation to ground surface $[kg m^{-2} s^{-1}]$
wtrs	Diagnosed residual water transferred into or out of the snow pack $[kg m^{-2} s^{-1}]$
wtrg	Diagnosed residual water transferred into or out of the soil $[kg m^{-2} s^{-1}]$
ovrflw	Overland flow from top of soil column

subflw	Interflow from sides of soil column $[kgm^{-2}s^{-1}]$
basflw	Base flow from bottom of soil column $[morkgm^{-2}s^{-1}]$
tovrfl	Temperature of overland flow from top of soil column [K]
tsubfl	Temperature of interflow from sides of soil column [K]
tbasfl	Temperature of base flow from bottom of soil column [K]
evap	Diagnosed total surface water vapour flux over modelled area $[kgm^{-2}s^{-1}]$
qfc	Water removed from soil layers by transpiration $[kgm^{-2}s^{-1}]$
hmfg	Diagnosed energy associated with phase change of water in soil layers $[Wm^{-2}]$
htc	Diagnosed internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}]$
rpcp	Rainfall rate over modelled area $[ms^{-1}]$
trpcp	Rainfall temperature over modelled area [C]
spcp	Snowfall rate over modelled area $[ms^{-1}]$
tspcp	Snowfall temperature over modelled area [C]
pcpr	Surface precipitation rate $[kgm^{-2}s^{-1}]$
ta	Air temperature at reference height [K]
tbarc	Subarea temperatures of soil layers [C]
tbarg	Subarea temperatures of soil layers [C]
tbarcs	Subarea temperatures of soil layers [C]
tbargs	Subarea temperatures of soil layers [C]
thliqc	Liquid water content of soil layers under vegetation $[m^3m^{-3}]$
thliqg	Liquid water content of soil layers in bare areas $[m^3m^{-3}]$
thicec	Frozen water content of soil layers under vegetation $[m^3m^{-3}]$
thiceg	Frozen water content of soil layers in bare areas $[m^3m^{-3}]$
hcpc	Heat capacity of soil layers under vegetation $[Jm^{-3}K^{-1}]$
hcpg	Heat capacity of soil layers in bare areas $[Jm^{-3}K^{-1}]$
tctopc	Thermal conductivity of soil at top of layer (vegetation over ground) $[Wm^{-1}K^{-1}]$
tcbotc	Thermal conductivity of soil at bottom of layer (vegetation over ground) $[Wm^{-1}K^{-1}]$
tctopg	Thermal conductivity of soil at top of layer (bare ground) $[Wm^{-1}K^{-1}]$
tcbotg	Thermal conductivity of soil at bottom of layer (bare ground) $[Wm^{-1}K^{-1}]$
froot	Fraction of total transpiration contributed by soil layer [ ]
froots	Fraction of total transpiration contributed by snow-covered soil layer [ ]
tsfsav	Ground surface temperature over subarea [K]
fc	Subarea fractional coverage of modelled area [ ]
fg	Subarea fractional coverage of modelled area [ ]
fcs	Subarea fractional coverage of modelled area [ ]
fgs	Subarea fractional coverage of modelled area [ ]
tpondc	Subarea temperature of surface ponded water [C]
tpondg	Subarea temperature of surface ponded water [C]
tpndcs	Subarea temperature of surface ponded water [C]
tpndgs	Subarea temperature of surface ponded water [C]
evapc	Evaporation from vegetation over ground $[ms^{-1}]$
evapcg	Evaporation from ground under vegetation $[ms^{-1}]$
evapg	Evaporation from bare ground $[ms^{-1}]$
evapcs	Evaporation from vegetation over snow $[ms^{-1}]$
evpcsg	Evaporation from snow under vegetation $[ms^{-1}]$
evapgs	Evaporation from snow on bare ground $[ms^{-1}]$

qfrezc	Heat sink to be used for freezing water on ground under canopy $[Wm^{-2}]$
qfrezg	Heat sink to be used for freezing water on bare ground $[Wm^{-2}]$
qmeltc	Heat to be used for melting snow under canopy $[Wm^{-2}]$
qmeltg	Heat to be used for melting snow on bare ground $[Wm^{-2}]$
raican	Intercepted liquid water stored on canopy over ground $[kgm^{-2}]$
snocan	Intercepted frozen water stored on canopy over ground $[kgm^{-2}]$
raicns	Intercepted liquid water stored on canopy over snow $[kgm^{-2}]$
snocns	Intercepted frozen water stored on canopy over snow $[kgm^{-2}]$
fsvf	Sky view factor of ground under vegetation canopy [ ]
fsvfs	Sky view factor of snow under vegetation canopy [ ]
cwlcap	Storage capacity of canopy over bare ground for liquid water $[kgm^{-2}]$
cwfcap	Storage capacity of canopy over bare ground for frozen water $[kgm^{-2}]$
cwlcps	Storage capacity of canopy over snow for liquid water $[kgm^{-2}]$
cwfcps	Storage capacity of canopy over snow for frozen water $[kgm^{-2}]$
tcano	Temperature of canopy over ground [K]
tcans	Temperature of canopy over snow [K]
chcap	Heat capacity of canopy over bare ground $[Jm^{-2}K^{-1}]$
chcaps	Heat capacity of canopy over snow $[Jm^{-2}K^{-1}]$
cmassc	Mass of canopy over bare ground $[kgm^{-2}]$
cmascs	Mass of canopy over snow $[kgm^{-2}]$
zsnow	Depth of snow pack [m]
rhosni	Density of fresh snow $[kgm^{-3}]$
gzeroc	Subarea heat flux at soil surface $[Wm^{-2}]$
gzerog	Subarea heat flux at soil surface $[Wm^{-2}]$
gzrocs	Subarea heat flux at soil surface $[Wm^{-2}]$
gzrogs	Subarea heat flux at soil surface $[Wm^{-2}]$
g12c	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g12g	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g12cs	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g12gs	Subarea heat flux between first and second soil layers $[Wm^{-2}]$
g23c	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
g23g	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
g23cs	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
g23gs	Subarea heat flux between second and third soil layers $[Wm^{-2}]$
tsnocs	Temperature of snow pack under vegetation [K]
tsnogs	Temperature of snow pack in bare areas [K]
wsnocs	Liquid water content of snow pack under vegetation $[kgm^{-2}]$
wsnogs	Liquid water content of snow pack in bare areas $[kgm^{-2}]$
rhoscs	Density of snow under vegetation $[kgm^{-3}]$
rhosgs	Density of snow in bare areas $[kgm^{-3}]$
zplimc	Subarea maximum ponding depth [m]
zplimg	Subarea maximum ponding depth [m]
zplmcs	Subarea maximum ponding depth [m]
zplmgs	Subarea maximum ponding depth [m]
ggeo	Geothermal heat flux at bottom of soil profile $[Wm^{-2}]$
thpor	Pore volume in soil layer $[m^3m^{-3}]$
thlret	Liquid water retention capacity for organic soil [m3 m-3]

thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}]$
bi	Clapp and Hornberger empirical "b" parameter [ ]
grksat	Saturated hydraulic conductivity of soil layer $[ms^{-1}]$
psisat	Soil moisture suction at saturation [m]
thlrat	Fractional saturation of soil behind the wetting front [ ]
thfc	Field capacity $[m^3m^{-3}]$
hcps	Heat capacity of soil material $[Jm^{-3}K^{-1}]$
delzw	Overall thickness of soil layer [m]
zbotw	Depth to permeable bottom of soil layer [m]
xdrain	Drainage index at bottom of soil profile [ ]
xslope	Surface slope (used when running MESH code) [degrees]
grkfac	WATROF parameter used when running MESH code [ ]
wfsurf	WATROF parameter used when running MESH code [ ]
wfcint	WATROF parameter used when running MESH code [ ]
delz	Overall thickness of soil layer [m]
isand	Sand content flag
igdr	Index of soil layer in which bedrock is encountered

First, subroutine WPREP is called to initialize various arrays and produce parameters for the four subareas of canopy over snow (CS), snow on ground (GS), canopy over ground (C) and bare ground (G). Then, for each of the four subareas, if the number of modelled areas containing that subarea is greater than zero, a series of subroutines is called. The subroutines associated with each subarea are listed in the table below.

CANVAP	Evaporation/sublimation of water from vegetation canopy	CS,C
CANADD	Addition of rainfall/snowfall to canopy; throughfall and drip	CS,C
CWCALC	Freezing/thawing of liquid/frozen water on canopy	CS,C
SUBCAN	Precipitaiton and condensation under canopy	CS,C
TWCALC	Freezing/thawing of liquid/frozen water in soil	CS,GS,C,G
SNOVAP	Sublimaiton from snow pack	CS,GS,C,G
TFREEZ	Freezing of ponded water on soil	CS,GS,C,G
TMELT	Melting of snow pack	CS,GS
SNOADD	Accumulation of snow on ground	CS,GS,C,G
SNINFL	Infiltration of rain into snow pack	CS,GS
ICEBAL	Energy and water budget of ice sheets	GS,G
GRINFL	Infiltraiton of water into soil	CS,GS,C,G
GRDRAN	Soil water movement in response to gravity and suction forces	CS,GS,C,G
TMCALC	Step ahead soil layer temperatures, check for freezing/thawing	CS,GS,C,G
CHKWAT	Check subarea moisture balances for closure	CS,GS,C,G
SNOALBW	Temporal variation of snow albedo and density	CS,GS

After these calls have been done, average values of the main prognostic variables over the modelled area are determined by performing weighted averages over the four subareas, and checks are carried out to identify and remove vanishingly small values. First the bedrock temperature in the third soil layer, the total runoff and the runoff temperature are calculated. Then the total runoff and the overland flow, interflow and baseflow are converted from units of m to $kgm^{-2}s^{-1}$. The total surface water vapour flux over the modelled area is updated to account for the residual amounts of evaporative demand over the four subareas that could not be supplied by surface stores (WLSTCS, WLSTGS, WLOSTC and WLOSTG, variables that are defined internally in this subroutine).

The temperature of the vegetation canopy TCAN and the amount of intercepted liquid water RCAN are calculated as weighted averages over the two canopy subareas. A flag is set to trigger a call to abort if TCAN is less than -100 C or greater than 100 C. If RCAN is vanishingly small, it is added to the overland flow and to the total runoff, and their respective temperatures are recalculated. The diagnostic arrays ROFC, ROVG, PCPG and HTCC are updated, and RCAN is set to zero. The amount of intercepted snow SNCAN is likewise calculated as a weighted average over the two canopy subareas. If SNCAN is vanishingly small, it is added to the overland flow and to the total runoff, and their respective temperatures are recalculated. The diagnostic arrays ROFC, ROVG, PCPG and HTCC are updated, and SNCAN is set to zero. If there is no canopy present, TCAN is set to zero.

At the end of the 600 loop, the depth of ponded water ZPOND and its temperature TPOND over the modelled area are calculated as weighted averages over the four subareas. If ZPOND is vanishingly small, then as in the case

of intercepted water, it is added to the overland flow and to the total runoff, and their respective temperatures are recalculated. The diagnostic array HTC is updated, and ZPOND and TPOND are set to zero.

In the 650 loop, values of the snow prognostic variables are calculated as weighted averages over the four subareas. The weightings for the subareas include the four internally-defined CLASSW variables XSNOCS, XSNOGS, XSNOWC and XSNOWG, which are set in subroutine CHKWAT to 1 if the subarea snow depth is greater than zero, and to zero otherwise. If the snow depth over the CS and GS subareas is greater than zero (meaning that there was a pre-existing snow cover at the beginning of the time step), the average snow albedo ALBSNO is preferentially set to the average over these two subareas. Otherwise ALBSNO is set to the average over the C and G subareas (where snow has just been added in the current time step). The snow temperature TSNOW and density RHOSNO are set to weighted averages over the four subareas, using the internally-defined subarea volumetric heat capacities HCPSCS/GS/C/G and RHOSCS/GS/C/G. Finally the snow depth ZSNOW is calculated from the subarea depths; the liquid water content of the snow pack WSNOW is obtained as a weighted average over the CS and GS subareas (assuming that freshly fallen snow does not yet contain liquid water); and the snow mass is determined from ZSNOW and RHOSNO. As in the case of intercepted and ponded water, if the snow mass is vanishingly small it and its liquid water content are added to the overland flow and to the total runoff, and their respective temperatures are recalculated. The diagnostic arrays ROFN, PCPG and HTCS are updated, and TSNOW, RHOSNO, SNO and WSNOW are set to zero. Flags are set to trigger calls to abort if TSNOW is less than 0 K or greater than 0.001 C. Finally, the three abort flags set thus far are checked, and calls to abort are performed if they are greater than zero.

In the 700 loop, the temperature of each soil layer is calculated as a weighted average over the four subareas. In the case of the third soil layer., if the standard three-layer configuration is being modelled (with a very thick third soil layer of 3.75 m), the subarea layer temperatures TBARCS/GS/C/G and the layer heat capacities HCPCS/GS/C/G apply to the permeable depth DELZW of the layer, and the bedrock temperature TBASE and the rock heat capacity HCPSND to the remainder, DELZ-DELZW. The averaging is carried out accordingly. In all other soil layers, the layer temperature applies to the whole thickness, whose heat capacity is a weighted average of HCPCS/GS/C/G over DELZW and HCPSND over DELZ-DELZW. The volumetric liquid water content THLIQ, the volumetric frozen water content THICE, and the heat flux at the soil layer interfaces GFLUX are calculated as simple weighted averages over the subareas. A flag is set to trigger a call to abort if the soil layer temperature is less than -100 C or greater than 100 C, and after the end of the loop, a call to abort is performed if the flag is greater than zero.

Finally, subroutine CGROW is called to update the vegetation growth index.

9.19 CLASSZ.f File Reference

Purpose: Check for energy and water balance closure over modelled area.

Functions/Subroutines

- subroutine classz (ISTEP, CTVSTP, CTSSTP, CT1STP, CT2STP, CT3STP, WTVSTP, WTSSTP, WTGSTP, FSGV, FLGV, HFSC, HEVC, HMFC, HTCC, FSGS, FLGS, HFSS, HEVS, HMFN, HTCS, FSGG, FLGG, HFSG, HEVG, HMFG, HTC, PCFC, PCLC, QFCF, QFCL, ROFC, WTRC, PCPN, QFN, ROFN, WTRS, PCPG, QFG, QFC, ROF, WTRG, CMAI, RCAN, SCAN, TCAN, SNO, WSNOW, TSNOW, THLIQ, THICE, HCPS, THPOR, DELZW, TBAR, ZPOND, TPOND, DELZ, FCS, FGS, FC, FG, IL1, IL2, ILG, IG, N)

9.19.1 Detailed Description

Purpose: Check for energy and water balance closure over modelled area.

9.19.2 Function/Subroutine Documentation

9.19.2.1 subroutine classz (integer *ISTEP,* real, dimension(ilg) *CTVSTP,* real, dimension(ilg) *CTSSTP,* real, dimension(ilg) *CT1STP,* real, dimension(ilg) *CT2STP,* real, dimension(ilg) *CT3STP,* real, dimension(ilg) *WTVSTP,* real, dimension(ilg) *WTSSTP,* real, dimension(ilg) *WTGSTP,* real, dimension (ilg) *FSGV,* real, dimension (ilg) *FLGV,* real, dimension (ilg) *HFSC,* real, dimension (ilg) *HEVC,* real, dimension (ilg) *HMFC,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *FSGS,* real, dimension (ilg) *FLGS,* real, dimension (ilg) *HFSS,* real, dimension (ilg) *HEVS,* real, dimension (ilg) *HMFN,* real, dimension (ilg) *HTCS,* real, dimension (ilg) *FSGG,* real, dimension (ilg) *FLGG,* real, dimension (ilg) *HFSG,* real, dimension (ilg) *HEVG,* real, dimension (ilg,ig) *HMFG,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *PCFC,* real, dimension (ilg) *PCLC,* real, dimension (ilg) *QFCF,* real, dimension (ilg) *QFCL,* real, dimension (ilg) *ROFC,* real, dimension (ilg) *WTRC,* real, dimension (ilg) *PCPN,* real, dimension (ilg) *QFN,* real, dimension (ilg) *ROFN,* real, dimension (ilg) *WTRS,* real, dimension (ilg) *PCPG,* real, dimension (ilg) *QFG,* real, dimension (ilg,ig) *QFC,* real, dimension (ilg) *ROF,* real, dimension (ilg) *WTRG,* real, dimension (ilg) *CMAI,* real, dimension (ilg) *RCAN,* real, dimension (ilg) *SCAN,* real, dimension (ilg) *TCAN,* real, dimension (ilg) *SNO,* real, dimension (ilg) *WSNOW,* real, dimension (ilg) *TSNOW,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *HCPS,* real, dimension (ilg,ig) *THPOR,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TPOND,* real, dimension (ig) *DELZ,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FGS,* real, dimension (ilg) *FC,* real, dimension (ilg) *FG,* integer *IL1,* integer *IL2,* integer *ILG,* integer *IG,* integer *N*)

Parameters

istep	Flag indicating position at beginning or end of time step
ctvstp	Change in internal energy of vegetation over current time step $[Wm^{-2}]$
ctsstp	Change in internal energy of snow pack over current time step $[Wm^{-2}]$
ct1stp	Change in internal energy of first soil layer over current time step $[Wm^{-2}]$
ct2stp	Change in internal energy of second soil layer over current time step $[Wm^{-2}]$
ct3stp	Change in internal energy of third soil layer over current time step $[Wm^{-2}]$
wtvstp	Change in vegetation mass over current time step $[kgm^{-2}]$
wtsstp	Change in snow mass over current time step $[kgm^{-2}]$
wtgstp	Change in soil water storage over current time step $[kgm^{-2}]$
fsgv	Diagnosed net shortwave radiation on vegetation canopy $[Wm^{-2}](K_{*,c})$
flgv	Diagnosed net longwave radiation on vegetation canopy $[Wm^{-2}](L_{*,c})$
hfsc	Diagnosed sensible heat flux on vegetation canopy $[Wm^{-2}](Q_{H,c})$
hevc	Diagnosed latent heat flux on vegetation canopy $[Wm^{-2}](Q_{E,c})$
hmfc	Diagnosed energy associated with phase change of water on vegetation $[Wm^{-2}](Q_{M,c})$
htcc	Diagnosed internal energy change of vegetation canopy due to conduction and/or change in mass $[Wm^{-2}](Q_{I,c})$
fsgs	Diagnosed net shortwave radiation at snow surface $[Wm^{-2}](K_{*,s})$
flgs	Diagnosed net longwave radiation at snow surface $[Wm^{-2}](L_{*,s})$
hfss	Diagnosed sensible heat flux at snow surface $[Wm^{-2}](Q_{H,s})$
hevs	Diagnosed latent heat flux at snow surface $[Wm^{-2}](Q_{E,s})$
hmfn	Diagnosed energy associated with phase change of water in snow pack $[Wm^{-2}](Q_{M,s})$
htcs	Diagnosed internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}](Q_{I,s})$
fsgg	Diagnosed net shortwave radiation at soil surface $[Wm^{-2}](K_{*,g})$
flgg	Diagnosed net longwave radiation at soil surface $[Wm^{-2}](L_{*,g})$
hfsg	Diagnosed sensible heat flux at soil surface $[Wm^{-2}](Q_{H,g})$
hevg	Diagnosed latent heat flux at soil surface $[Wm^{-2}](Q_{E,g})$
hmfg	Diagnosed energy associated with phase change of water in soil layers $[Wm^{-2}](Q_{M,g})$
htc	Diagnosed internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}](Q_{I,g})$
pcfc	Diagnosed frozen precipitation intercepted by vegetation $[kgm^{-2}s^{-1}](P_{f,c})$
pclc	Diagnosed liquid precipitation intercepted by vegetation $[kgm^{-2}s^{-1}](P_{l,c})$
qfcf	Diagnosed vapour flux from frozen water on vegetation $[kgm^{-2}s^{-1}](E_{f,c})$
qfcl	Diagnosed vapour flux from liquid water on vegetation $[kgm^{-2}s^{-1}](E_{l,c})$
rofc	Liquid/frozen water runoff from vegetation $[kgm^{-2}s^{-1}](R_c)$
wtrc	Diagnosed water transferred off the vegetation canopy $[kgm^{-2}s^{-1}](A_c)$
pcpn	Diagnosed precipitation incident on snow pack $[kgm^{-2}s^{-1}](P_s)$
qfn	Diagnosed water vapour flux from snow pack $[kgm^{-2}s^{-1}](E_s)$
rofn	Liquid water runoff from snow pack $[kgm^{-2}s^{-1}](R_s)$
wtrs	Diagnosed water transferred into or out of the snow pack $[kgm^{-2}s^{-1}](A_s)$
pcpg	Diagnosed precipitation incident on ground $[kgm^{-2}s^{-1}](P_g)$
qfg	Diagnosed water vapour flux from ground surface $[kgm^{-2}s^{-1}](E_g)$
qfc	Diagnosed vapour flux from transpiration over modelled area $[Wm^{-2}](E_c)$
rof	Total runoff from soil $[kgm^{-2}s^{-1}](R_g)$

wtrg	Diagnosed water transferred into or out of the soil $[kgm^{-2}s^{-1}](A_g)$
cmai	Current mass of vegetation canopy $[kgm^{-2}](W_c)$
rcan	Intercepted liquid water stored on canopy $[kgm^{-2}](W_{l,c})$
scan	Intercepted frozen water stored on canopy $[kgm^{-2}](W_{f,c})$
tcan	Vegetation canopy temperature $[K](T_c)$
sno	Mass of snow pack $[kgm^{-2}](W_s)$
wsnow	Liquid water content of snow pack $[kgm^{-2}](W_{l,s})$
tsnow	Snowpack temperature $[K](T_s)$
thliq	Volumetric liquid water content of soil layers $[m^3m^{-3}](\theta_l)$
thice	Volumetric frozen water content of soil layers $[m^3m^{-3}](\theta_f)$
hcps	Volumetric heat capacity of soil particles $[Jm^{-3}](C_g)$
thpor	Pore volume in soil layer $[m^3m^{-3}]$
delzw	Permeable thickness of soil layer $[m](\Delta_z w)$
delz	Total thickness of soil layer $[m](\Delta_z)$
tbar	Temperature of soil layers $[K](T_g)$
zpond	Depth of ponded water on surface $[m](z_p)$
tpond	Total thickness of soil layer $[m](\Delta_z)$
fcs	Fractional coverage of vegetation over snow on modelled area [ ]
fgs	Fractional coverage of snow over bare ground on modelled area [ ]
fc	Fractional coverage of vegetation over bare ground on modelled area [ ]
fg	Fractional coverage of bare ground on modelled area [ ]

In this subroutine, checks are carried out to ensure that the change in energy storage in each of the components of the modelled area (canopy, snow and soil) is equal to the sum of the energy fluxes into and out of them; and that the change in moisture storage in each of the components is equal to the sum of the water fluxes into and out of them. The subroutine is called twice, once at the beginning (ISTEP=0) and once at the end (ISTEP=1) of each time step. At the beginning, the instantaneous energy and moisture storage terms are evaluated, and at the end the differences over the time step are calculated:

Change in canopy energy storage = $\Delta[(c_c W_c + c_w W_{l,c} + c_i W_{f,c})T_c]/\Delta t$

Change in snow energy storage = $\Delta[(C_i W_s/\rho_i + C_w W_{l,s}/\rho_w)T_s]/\Delta t$

Change in soil layer energy storage = $\Delta[(C_w\theta_l + C_i\theta_f + C_g\theta_g)\Delta z_w + C_b(\Delta z \breve{} \Delta z_w)]T_j/\Delta t$ (For the first soil layer, the numerator contains the additional term $C_w z_p T_p$.)

Change in canopy moisture storage = $\Delta[W_{l,c} + W_{f,c}]$

Change in snow moisture storage = $\Delta[W_s + W_{l,s}]$

Change in soil moisture storage = $\Delta[(\theta_l\rho_w + \theta_f\rho_i)\Delta z_w + z_p\rho_w]$

The net energy and moisture fluxes are also evaluated at the end of the time step:

Net energy flux for canopy = $K_{*,c} + L_{*,c} \breve{} Q_{H,c} \breve{} Q_{E,c} \breve{} Q_{M,c} + Q_{I,c}$

Net energy flux for snow = $K_{*,s} + L_{*,s} \breve{} Q_{H,s} \breve{} Q_{E,s} \breve{} Q_{M,s} + Q_{I,s}$

Net energy flux for first soil layer = $K_{*,g} + L_{*,g} \breve{} Q_{H,g} \breve{} Q_{E,g} \breve{} Q_{M,1} + Q_{I,1}$

Net energy flux for other soil layers = $-Q_{M,j} + Q_{I,j}$

Net moisture flux for canopy = $P_{l,c} + P_{f,c} \breve{} E_{l,c} \breve{} E_{f,c} \breve{} R_c + A_c$

Net moisture flux for snow = $P_s \breve{} E_s \breve{} R_s + A_s$

Net moisture flux for soil = $P_g \breve{} E_g \breve{} R_g + A_g - E_c$

In these equations the K_* terms refer to net shortwave radiation, the L_* terms to net longwave radiation, the Q_H terms to sensible heat flux, the Q_E terms to latent heat flux, the Q_M terms to heat associated with melting or freezing of water, and the Q_I terms to changes in heat storage caused by conduction or redistribution of water. The P terms refer to precipitation, the E terms to evaporation, the R terms to runoff and the A terms to water transferred between different components of the landscape. The subscript 1 refers to the first soil layer, and j to a generalized other layer.

Finally, each change in energy or moisture storage is compared in turn with the corresponding net flux of energy or moisture, and if the difference is greater than a selected threshold value, an error message is printed out and the

run is stopped.

9.20 competition_map.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Mapping For Competition Subroutine.

Functions/Subroutines

- subroutine competition_map (nml, ilmos, jlmos, grclarea,faregat,fcancmx,nppveg, geremort,intrmort, gleafmas, bleafmas, stemmass,rootmass, litrmass, soilcmas,pftexist,lambda, bmasveg, burnvegf,add2allo,cc,mm,fcanmx,vgbiomas, gavgltms, gavgscms,ta, precip, netrad,tcurm, srpcuryr,dftcuryr,tmonth, anpcpcur, anpecur,gdd5cur, surmncur, defmncur, srplscur,defctcur,twarmm,tcoldm,gdd5,aridity, srplsmon, defctmon, anndefct,annsrpls,annpcp,dry↩ _season_length,lucemcom,lucltrin, lucsocin,pfcancmx,nfcancmx, pstemmass,pgleafmass,

9.20.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Mapping For Competition Subroutine.

9.20.2 Function/Subroutine Documentation

9.20.2.1 **subroutine competition_map (integer *nml,* integer, dimension(ilg) *ilmos,* integer, dimension(ilg) *jlmos,* real, dimension(ilg) *grclarea,* real, dimension(ilg) *faregat,* real, dimension(ilg,icc) *fcancmx,* real, dimension(ilg,icc) *nppveg,* real, dimension(ilg,icc) *geremort,* real, dimension(ilg,icc) *intrmort,* real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *bleafmas,* real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* real, dimension(ilg,iccp1) *litrmass,* real, dimension(ilg,iccp1) *soilcmas,* logical, dimension(ilg,icc) *pftexist,* real, dimension(ilg,icc) *lambda,* real, dimension(ilg,icc) *bmasveg,* real, dimension(ilg,icc) *burnvegf,* real, dimension(ilg,icc) *add2allo,* real, dimension(ilg,icc) *cc,* real, dimension(ilg,icc) *mm,* real, dimension(ilg,ican) *fcanmx,* real, dimension(ilg) *vgbiomas,* real, dimension(ilg) *gavgltms,* real, dimension(ilg) *gavgscms,* real, dimension(ilg) *ta,* real, dimension(ilg) *precip,* real, dimension(ilg) *netrad,* real, dimension(ilg) *tcurm,* real, dimension(ilg) *srpcuryr,* real, dimension(ilg) *dftcuryr,* real, dimension(12,ilg) *tmonth,* real, dimension(ilg) *anpcpcur,* real, dimension(ilg) *anpecur,* real, dimension(ilg) *gdd5cur,* real, dimension(ilg) *surmncur,* real, dimension(ilg) *defmncur,* real, dimension(ilg) *srplscur,* real, dimension(ilg) *defctcur,* real, dimension(ilg) *twarmm,* real, dimension(ilg) *tcoldm,* real, dimension(ilg) *gdd5,* real, dimension(ilg) *aridity,* real, dimension(ilg) *srplsmon,* real, dimension(ilg) *defctmon,* real, dimension(ilg) *anndefct,* real, dimension(ilg) *annsrpls,* real, dimension(ilg) *annpcp,* real, dimension(ilg) *dry_season_length,* real, dimension(ilg) *lucemcom,* real, dimension(ilg) *lucltrin,* real, dimension(ilg) *lucsocin,* real, dimension(ilg,icc) *pfcancmx,* real, dimension(ilg,icc) *nfcancmx,* real, dimension(ilg,icc) *pstemmass,* real, dimension(ilg,icc) *pgleafmass*)**

Parameters

nml	nml total number of mosaic tiles with pft fractions larger than 1, see gatprep.f
ilmos	indices for scattering, see gatprep.f
jlmos	indices for scattering, see gatprep.f
fcancmx	fractional coverage of ctem's 10 pfts in each mosaic
faregat	fractional coverage of each ctem pft in each mosaic tile
grclarea	area of the grid cell, km^2
nppveg	npp for each pft type /m2 of vegetated area [u-mol co2-c/m2.sec]
geremort	growth related mortality (1/day)
intrmort	intrinsic (age related) mortality (1/day)

gleafmas	green leaf mass for each of the 10 ctem pfts, kgc/m^2
bleafmas	brown leaf mass for each of the 10 ctem pfts, kgc/m^2
stemmass	stem mass for each of the 10 ctem pfts, kgc/m^2
rootmass	root mass for each of the 10 ctem pfts, kgc/m^2
litrmass	litter mass for each of the 10 ctem pfts + bare, kgc/m^2
soilcmas	soil carbon mass for each of the 10 ctem pfts + bare, kgc/m^2
lambda	fraction of npp that is used for horizontal expansion
bmasveg	total (gleaf + stem + root) biomass for each ctem pft, kgc/m^2
burnvegf	fractional areas burned, for ctem pfts
add2allo	npp kgc/m^2.day that is used for expansion and subsequently allocated to leaves, stem, and root via the allocation part of the model.
cc	colonization rate & mortality rate
mm	colonization rate & mortality rate
fcanmx	fractional coverage of class' 4 pfts
vgbiomas	grid averaged vegetation biomass, kgc/m^2
gavgltms	grid averaged litter mass, kgc/m^2
gavgscms	grid averaged soil c mass, kgc/m^2
lucemcom	land use change (luc) related combustion emission losses, u-mol co2/m2.sec
lucltrin	luc related inputs to litter pool, u-mol co2/m2.sec
lucsocin	luc related inputs to soil c pool, u-mol co2/m2.sec
pfcancmx	previous year's fractional coverages of pfts
nfcancmx	next year's fractional coverages of pfts
pftexist	logical array indicating pfts exist (t) or not (f)
ta	mean daily temperature, k
precip	daily precipitation (mm/day)
netrad	daily net radiation (w/m2)
tcurm	temperature of the current month (c)
srpcuryr	water surplus for the current year
dftcuryr	water deficit for the current year
tmonth	monthly temperatures
anpcpcur	annual precipitation for current year (mm)
anpecur	annual potential evaporation for current year (mm)
gdd5cur	growing degree days above 5 c for current year
surmncur	number of months with surplus water for current year
defmncur	number of months with water deficit for current year
srplscur	water surplus for the current month
defctcur	water deficit for the current month
twarmm	temperature of the warmest month (c)
tcoldm	temperature of the coldest month (c)
gdd5	growing degree days above 5 c
aridity	aridity index, ratio of potential evaporation to precipitation
srplsmon	number of months in a year with surplus water i.e. precipitation more than potential evaporation
defctmon	number of months in a year with water deficit i.e. precipitation less than potential evaporation
anndefct	annual water deficit (mm)
annsrpls	annual water surplus (mm)
annpcp	annual precipitation (mm)

dry_season_length	length of dry season (months)

parameters used

note the structure of parameter vectors which clearly shows the class pfts (along rows) and ctem sub-pfts (along columns)

needle leaf	evg1	evg2	dcd
broad leaf	evg	dcd-cld	dcd-dry
crops	c3	c4	—
grasses	c3	c4	—

scattering the pft index in each mosaic (fcancmx) to pft index in each mosaic of each grid cell (fcancmxrow) nml, ilmos and jlmos are referring to gatprep.f

mapping the pft areal fraction in each mosaic of each grid cell (farerow) to pft fraction in each grid cell (fare_cmp)

9.21 competition_mod.f90 File Reference

Central module for all competition scheme-related operations.

Functions/Subroutines

- subroutine, public competition_scheme::bioclim (iday, ta, precip, netrad, il1, il2, nilg, leapnow, tcurm, srpcuryr, dftcuryr, inibioclim, tmonth, anpcpcur, anpecur, gdd5cur, surmncur, defmncur, srplscur, defctcur, twarmm, tcoldm, gdd5, aridity, srplsmon, defctmon, anndefct, annsrpls, annpcp, dry_season_length)

- subroutine, public competition_scheme::existence (iday, il1, il2, nilg, sort, nol2pfts, twarmm, tcoldm, gdd5, aridity, srplsmon, defctmon, anndefct, annsrpls, annpcp, pftexist, dry_season_length)

- subroutine, public competition_scheme::competition (iday, il1, il2, nilg, nol2pfts, nppveg, dofire, leapnow, pftexist, geremort, intrmort, gleafmas, bleafmas, stemmass, rootmass, litrmass, soilcmas, grclarea, lambda, burnvegf, sort, pstemmass, pgleafmass, fcancmx, fcanmx, vgbiomas, gavgltms, gavgscms, bmasveg, add2allo, colrate, mortrate)

9.21.1 Detailed Description

Central module for all competition scheme-related operations.

Competition between PFTs in CTEM is based upon modified L–V equations [3] [11]. The L–V equations **[lotka1925elements]** [52] have been adapted from their initial application for simulating predator–prey interactions in ecosystem models as described below.

Competition parametrization

The change in fractional coverage (f) of a PFT α through time, $\frac{\mathrm{d}f_\alpha}{\mathrm{d}t}$, is expressed as the result of mortality, and competition and colonization (CC) interactions with the other PFTs present in a grid cell and bare ground, collectively represented as B where $\alpha \notin B$:

$$\frac{\mathrm{d}f_\alpha}{\mathrm{d}t} = g(f_\alpha, f_B) - m_\alpha f_\alpha.$$

The CC interactions are represented symbolically by the $g(f_\alpha, f_B)$ function. Mortality is assumed to be proportional to the number density of plants and represented by the mortality term, $m_\alpha f_\alpha$. The PFT-dependent mortality rate (m_α; day^{-1}) (described further in Sect. mort}) produces bare ground via a number of processes, and that bare

ground is subsequently available for colonization. We consider the fractional coverage for N PFTs plus bare ground (f_{N+1} = f_{bare}) where $\sum_{j=1}^{N+1} f_j = 1$. For competition between unequal competitors, the PFTs are ranked in terms of their dominance. If PFT α is the most dominant, it will invade the area of other PFTs and the bare ground (f_B, $\alpha \notin B$). Woody PFTs are all more dominant than grass PFTs since trees can successfully invade grasses by overshading them [48] and thus are ranked higher. Within tree or grass PFTs the dominance rank of a PFT is calculated based upon its colonization rate (c_α; day^{-1}) with higher colonization rates giving a higher dominance ranking. For the general case of PFT α with a dominance rank of i, we describe the ranking from most dominant to least as 1, 2, ..., $i-1$, i, $i+1$, ..., N. Equation (concepteqn}) can then be reformulated following a phenomenological approach as

$$\frac{\mathrm{d}f_\alpha}{\mathrm{d}t} = f_\alpha^b(c_{\alpha,i+1}f_{i+1} + c_{\alpha,i+2}f_{i+2} + \ldots + c_{\alpha,N}f_N) - f_\alpha(c_{1,\alpha}f_1^b + c_{2,\alpha}f_2^b + \ldots + c_{(i-1),\alpha}f_{i-1}^b) - m_\alpha f_\alpha,$$

where the exponent b is an empirical parameter, which controls the behaviour of the L–V equations. In the original L–V formulation, b is 1, but we modify the L–V relations by using $b = 0$ following [3] [11] (implications of this choice are expanded upon below). The fractional cover of PFT α then changes depending on the gains it makes into the area of less dominant PFTs and the losses it suffers due to mortality and encroachment by more dominant PFTs. The rate of change of the bare fraction, f_{bare}, is expressed as

$$\frac{\mathrm{d}f_{bare}}{\mathrm{d}t} = \sum_{\beta=1}^{N}(m_\beta f_\beta - c_{\beta,bare}f_\beta^b f_{bare}).$$

The rate at which PFT α invades another PFT β is given by

$$c_{\alpha,\beta}f_\alpha^b f_\beta = c_\alpha\left(\frac{c_{\alpha,\beta}}{c_\alpha}\right)f_\alpha^b f_\beta = c_\alpha\delta_{\alpha,\beta}f_\alpha^b f_\beta.$$

A PFT invading bare ground has an unimpeded *invasion* rate, c_α. The ratio of the invasion rate by PFT α into area covered by another PFT β and its unimpeded invasion rate ($\frac{c_{\alpha,\beta}}{c_\alpha}$) gives the relative efficiency of colonization, termed $\delta_{\alpha,\beta}$, which is a scalar between 0 and 1. δ is 1 for invasion of any PFT into bare ground and 1 for tree PFT invasion into grass PFTs. If a PFT β has a lower dominance ranking than another PFT α then $\delta_{\beta,\alpha} = 0$ implying that sub-dominant PFTs do not invade dominant PFTs, but get invaded by them, i.e. $\delta_{\alpha,\beta} = 1$. Equation (full}) can then be written more succinctly for each PFT as

$$\frac{\mathrm{d}f_\alpha}{\mathrm{d}t} = \sum_{\beta=1}^{N+1}(c_\alpha\delta_{\alpha,\beta}f_\alpha^b f_\beta - c_\beta\delta_{\beta,\alpha}f_\alpha f_\beta^b) - m_\alpha f_\alpha.$$

The value of parameter b is related to the manner in which two PFTs interact, represented by $f_\alpha^b f_\beta$, in Eqs. (full})–(coloniz}). As a result, the value of b affects the equilibrium solution for fractional coverage of PFTs as well as how f_i evolves over time.

For the usual form of the L–V equations with $b = \delta = 1$, and for the case of a grid cell with two PFTs, the competition–colonization equations are

$$\frac{\mathrm{d}f_1}{\mathrm{d}t} = c_1f_1(f_2+f_{bare}) - m_1f_1 = c_1f_1(1-f_1) - m_1f_1, \frac{\mathrm{d}f_2}{\mathrm{d}t} = c_2f_2f_{bare} - c_1f_1f_2 - m_2f_2 = c_2f_2(1-f_1-f_2) - c_1f_1f_2 - m_2f_2,$$

where the dominant PFT 1 invades PFT 2 and the bare fraction, and PFT 2 invades only the bare fraction. The equilibrium solutions for f_1 and f_2 in this case are

$$f_1 = max\left[\frac{c_1 - m_1}{c_1}, 0\right],$$

$$f_2 = max\left[\frac{c_2 - c_2f_1 - c_1f_1 - m_2}{c_2}, 0\right] = max\left[\frac{(c_2 - m_2) - (1 + \frac{c_2}{c_1})(c_1 - m_1)}{c_2}, 0\right],$$

In Eq. (f2_eq_b_eq_1}), as long as $(c_1 - m_1) > (c_2 - m_2)$ the equilibrium solution for f_2 will always be zero and coexistence is not possible.

For $b = 0$ and $\delta = 1$, the competition–colonization equations are

$$\frac{\mathrm{d}f_1}{\mathrm{d}t} = c_1(f_2+f_{bare})-m_1 f_1 = c_1(1-f_1)-m_1 f_1, \frac{\mathrm{d}f_2}{\mathrm{d}t} = c_2 f_{bare}-c_1 f_2-m_2 f_2 = c_2(1-f_1-f_2)-c_1 f_2-m_2 f_2,$$

and the corresponding equilibrium fractions are

$$f_1 = \frac{c_1}{c_1 + m_1},$$

$$f_2 = \frac{c_2(1 - f_1)}{(c_1 + c_2 + m_2)}.$$

In Eqs. (f_equil_b_eq_0_1}) and (f_equil_b_eq_0_2}), as long as $m_1 > 0$ and $c_2 > 0$, then PFT 2 will always exist and equilibrium coexistence is possible. Values of parameter b between 1 and 0 yield equilibrium values of f_2 that vary between 0 (Eq. f2_eq_b_eq_1}) and those obtained using Eq. (f_equil_b_eq_0_2}). $b = 0$ yields a maximum value of equilibrium f_2 allowing PFT 2 to coexist maximally.

In the standard L–V equations for predator–prey interactions coexistence is possible because the predator depends on prey for its food and so the predator population suffers as the prey population declines. This is in contrast to the application of the equations for competition between PFTs where the dominant PFT does not depend on sub-dominant PFTs for its existence and is thus able to exclude them completely. The PFTs interact with each other through the invasion term $(-c_\beta \delta_{\beta,\alpha} f_\alpha f_\beta^b)$ in Eq. (compact}), where $\delta_{\alpha,\beta} = 1$ or 0 depending on whether PFT α can or cannot invade PFT β, respectively, as mentioned earlier. This interaction through invasion is represented by $-c_1 f_1 f_2$ in Eq.(cc_eq_b_eq_1_2}) (for $b = 1$) and by $-c_1 f_2$ in Eq. (cc_eq_b_eq_0_1}) (for $b = 0$). The magnitude of this interaction thus depends on the value of parameter b. When $b=1$ the interaction is proportional to the product of the fractional coverage of the two PFTs ($f_1 f_2$). When $b = 0$, the interaction is proportional to the fractional coverage of the PFT being invaded (f_2). The use of $b = 0$ thus reduces the product term $f_\alpha^b f_\beta$ to f_β and implies that the invasion of sub-dominant PFT β does not depend on the current fractional coverage of the dominant PFT α. This case may be thought of as corresponding to the general availability of the seeds of the dominant PFT α that may germinate and invade the coverage of the sub-dominant PFT β provided the climate is favourable, even if PFT α does not exist in the grid cell, i.e. $f_\alpha = 0$ (in the case where $f_\alpha = 0$, the PFT is always assumed to have a dormant seed bank in the grid cell given the long lifetimes of seeds and their wide dispersion). In contrast, in the standard version of the L–V equations, as implemented for predator–prey interactions, b always equals 1 since the amount of predation, and hence the reduction in the number of prey, depends on the product of the number of predators and the number of prey. Using $b = 0$ is thus consistent with invasion of the sub-dominant PFT β being unaffected by the fractional coverage of the dominant PFT α.

Colonization rate

The PFT-dependent colonization rate (c_α; day^{-1}) is calculated based on the fraction (Λ_α) of positive NPP ($kg\,C\,m^{-2}\,day^{-1}$) that is used for spatial expansion

$$c_\alpha = \Lambda_\alpha \, NPP_\alpha \, \xi_\alpha,$$

where ξ_α ($(kg\,C)^{-1}\,m^2$) is the inverse sapling density calculated as the reciprocal of vegetation biomass ($C_{veg,\alpha}$; $kg\,C\,m^{-2}$) multiplied by a PFT-dependent constant ($S_{sap,\alpha}$; unitless; see also ctem_params.f90)

$$\xi_\alpha = \frac{1}{S_{sap,\alpha} \, \max[0.25, \min(5.0, C_{veg,\alpha})]}.$$

The fraction of NPP used for spatial expansion, Λ_α, is calculated using the leaf area index (LAI_α; $m^2\,leaf\,(m^2\,ground)^{-1}$) of a PFT

$$\Lambda_\alpha = \min(\lambda_{max}, \max(\lambda_{1,\alpha}, \lambda_{2,\alpha})),$$

$$if LAI_\alpha \leq LAI_{min,\alpha}: \quad \lambda_{1,\alpha} = 0 if LAI_{min,\alpha} < LAI_\alpha < LAI_{max,\alpha}: \quad \lambda_{1,\alpha} = \frac{LAI_\alpha - LAI_{min,\alpha}}{LAI_{max,\alpha} - LAI_{min,\alpha}} \lambda_{max} if LAI_\alpha \geq LAI_n$$

$if LAI_\alpha > 0.25 LAI_{min,\alpha}: \quad \lambda_{2,\alpha} = \cosh(0.115(LAI_\alpha - 0.25 LAI_{min,\alpha})) - 1 if LAI_\alpha \leq 0.25 LAI_{min,\alpha}: \quad \lambda_{2,\alpha} = 0$

The original formulation of [3] only considered $\lambda_{1,\alpha}$ but here we adjust the parametrization with the addition of $\lambda_{2,\alpha}$, which ensures that a small fraction of NPP is used for spatial expansion even at very low LAI values. This additional constraint allows for improved fractional coverage of grasses in arid regions. Similar to $S_{sap,\alpha}$, $LAI_{min,\alpha}$ and $LAI_{max,\alpha}$ are PFT-dependent parameters (see also ctem_params.f90).

The value of λ_{max} is set to 0.1 so that a maximum of 10\,{%} of daily NPP can be used for spatial expansion. Finally, Λ_α is set to zero for tree PFTs when they are in a full leaf-out mode and all NPP is being used for leaf expansion (see Appendix phenol}).

9.22 competition_unmap.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Unmapping For Competition.

Functions/Subroutines

- subroutine competition_unmap (nml, ilmos, jlmos, nol2pfts,fare_cmp,nppveg_cmp, geremort_cmp,intrmort↩ _cmp, gleafmas_cmp, bleafmas_cmp,stemmass_cmp, rootmass_cmp, litrmass_cmp,soilcmas_cmp, pftexist_cmp,lambda_cmp,bmasveg_cmp, burnvegf_cmp, add2allo_cmp,cc_cmp,mm_cmp,fcanmx_↩ cmp,vgbiomas_cmp, grclarea_cmp,gavgltms_cmp, gavgscms_cmp,ta_cmp,precip_cmp,netrad_cmp,tcurm↩ _cmp, srpcuryr_cmp, dftcuryr_cmp,tmonth_cmp, anpcpcur_cmp, anpecur_cmp,gdd5cur_cmp, surmncur↩ _cmp, defmncur_cmp,srplscur_cmp, defctcur_cmp,twarmm_cmp,tcoldm_cmp,gdd5_cmp, aridity_↩ cmp,srplsmon_cmp, defctmon_cmp, anndefct_cmp,annsrpls_cmp,annpcp_cmp,dry_season_length↩ _cmp,lucemcom_cmp, lucltrin_cmp, lucsocin_cmp,pfcancmx_cmp, nfcancmx_cmp, pstemmass_↩ cmp,pgleafmass_cmp,

9.22.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Unmapping For Competition.

9.22.2 Function/Subroutine Documentation

9.22.2.1 **subroutine competition_unmap (integer *nml,* integer, dimension(ilg) *ilmos,* integer, dimension(ilg) *jlmos,* integer, dimension(ican) *nol2pfts,* real, dimension(nlat,icc) *fare_cmp,* real, dimension(nlat,icc) *nppveg_cmp,* real, dimension(nlat,icc) *geremort_cmp,* real, dimension(nlat,icc) *intrmort_cmp,* real, dimension(nlat,icc) *gleafmas_cmp,* real, dimension(nlat,icc) *bleafmas_cmp,* real, dimension(nlat,icc) *stemmass_cmp,* real, dimension(nlat,icc) *rootmass_cmp,* real, dimension(nlat,iccp1) *litrmass_cmp,* real, dimension(nlat,iccp1) *soilcmas_cmp,* logical, dimension(nlat,icc) *pftexist_cmp,* real, dimension(nlat,icc) *lambda_cmp,* real, dimension(nlat,icc) *bmasveg_cmp,* real, dimension(nlat,icc) *burnvegf_cmp,* real, dimension(nlat,icc) *add2allo_cmp,* real, dimension(nlat,icc) *cc_cmp,* real, dimension(nlat,icc) *mm_cmp,* real, dimension(nlat,ican) *fcanmx_cmp,* real, dimension(nlat) *vgbiomas_cmp,* real, dimension(nlat) *grclarea_cmp,* real, dimension(nlat) *gavgltms_cmp,* real, dimension(nlat) *gavgscms_cmp,* real, dimension(nlat) *ta_cmp,* real, dimension(nlat) *precip_cmp,* real, dimension(nlat) *netrad_cmp,* real, dimension(nlat) *tcurm_cmp,* real, dimension(nlat) *srpcuryr_cmp,* real, dimension(nlat) *dftcuryr_cmp,* real, dimension(12,nlat) *tmonth_cmp,* real, dimension(nlat) *anpcpcur_cmp,* real, dimension(nlat) *anpecur_cmp,* real, dimension(nlat) *gdd5cur_cmp,* real, dimension(nlat) *surmncur_cmp,* real, dimension(nlat) *defmncur_cmp,* real, dimension(nlat) *srplscur_cmp,* real, dimension(nlat) *defctcur_cmp,* real, dimension(nlat) *twarmm_cmp,* real, dimension(nlat) *tcoldm_cmp,* real, dimension(nlat) *gdd5_cmp,* real, dimension(nlat) *aridity_cmp,* real, dimension(nlat) *srplsmon_cmp,* real, dimension(nlat) *defctmon_cmp,* real, dimension(nlat) *anndefct_cmp,* real, dimension(nlat) *annsrpls_cmp,* real, dimension(nlat) *annpcp_cmp,* real, dimension(nlat) *dry_season_length_cmp,* real, dimension(nlat) *lucemcom_cmp,* real, dimension(nlat) *lucltrin_cmp,* real, dimension(nlat) *lucsocin_cmp,* real, dimension(nlat,icc) *pfcancmx_cmp,* real, dimension(nlat,icc) *nfcancmx_cmp,* real, dimension(nlat,icc) *pstemmass_cmp,* real, dimension(nlat,icc) *pgleafmass_cmp*)

Parameters

nml	total number of mosaic tiles with pft fractions larger than 1, see gatprep.f
ilmos	indices for scattering, see gatprep.f
jlmos	indices for scattering, see gatprep.f
fare_cmp	fractional coverage of ctem's 10 pfts in each latitudinal grid cell
nppveg_cmp	npp for each pft type of vegetated area in each latitudinal grid cell
geremort_cmp	growth related mortality in each latitudinal grid cell
intrmort_cmp	intrinsic (age related) mortality in each latitudinal grid cell
gleafmas_cmp	green leaf mass for each of the 10 ctem pfts in each latitudinal grid cell
bleafmas_cmp	brown leaf mass for each of the 10 ctem pfts in each latitudinal grid cell
stemmass_cmp	stem mass for each of the 10 ctem pfts in each latitudinal grid cell
rootmass_cmp	root mass for each of the 10 ctem pfts in each latitudinal grid cell
litrmass_cmp	litter mass for each of the 10 ctem pfts + bare, in each latitudinal grid cell
soilcmas_cmp	soil carbon mass for each of the 10 ctem pfts + bare, in each latitudinal grid cell
lambda_cmp	fraction of npp that is used for horizontal expansion in each latitudinal grid cell
bmasveg_cmp	total (gleaf + stem + root) biomass for each ctem pft, kgc/m^2 in each latitudinal grid cell
burnvegf_cmp	fractional areas burned, for ctem pfts in each latitudinal grid cell
add2allo_cmp	npp kgc/m^2.day in each latitudinal grid cell that is used for expansion and subsequently allocated to leaves, stem, and root via the allocation part of the model.
cc_cmp	colonization rate & mortality rate in each latitudinal grid cell
mm_cmp	colonization rate & mortality rate in each latitudinal grid cell
fcanmx_cmp	fractional coverage of class' 4 pfts in each latitudinal grid cell
grclarea_cmp	area of the grid cell, km^2
vgbiomas_cmp	grid averaged vegetation biomass, kgc/m^2
gavgltms_cmp	grid averaged litter mass, kgc/m^2
gavgscms_cmp	grid averaged soil c mass, kgc/m^2
lucemcom_cmp	land use change (luc) related combustion emission losses in each latitudional grid cell, u-mol co2/m2.sec
lucltrin_cmp	luc related inputs to litter pool, in each latitudional grid cell, u-mol co2/m2.sec
lucsocin_cmp	luc related inputs to soil c pool, in each latitudional grid cell, u-mol co2/m2.sec
pfcancmx_cmp	previous year's fractional coverages of pfts in each latitudinal grid cell
nfcancmx_cmp	next year's fractional coverages of pfts in each latitudinal grid cell
pstemmass_cmp	stem mass from previous timestep, is value before fire. used by burntobare subroutine
pgleafmass_cmp	root mass from previous timestep, is value before fire. used by burntobare subroutine
pftexist_cmp	logical array indicating pfts exist (t) or not (f) in each latitudinal grid cell
ta_cmp	mean daily temperature (k) in each latitudinal grid cell
precip_cmp	daily precipitation (mm/day) in each latitudinal grid cell
netrad_cmp	daily net radiation (w/m2) in each latitudinal grid cell
tcurm_cmp	temperature of the current month (c) in each latitudinal grid cell
srpcuryr_cmp	water surplus for the current year in each latitudinal grid cell
dftcuryr_cmp	water deficit for the current year in each latitudinal grid cell
tmonth_cmp	monthly temperatures in each latitudinal grid cell
anpcpcur_cmp	annual precipitation for current year (mm) in each latitudinal grid cell
anpecur_cmp	annual potential evaporation for current year (mm) in each latitudinal grid cell
gdd5cur_cmp	growing degree days above 5 c for current year in each latitudinal grid cell

surmncur_cmp	number of months with surplus water for current year in each latitudinal grid cell
defmncur_cmp	number of months with water deficit for current year in each latitudinal grid cell
srplscur_cmp	water surplus for the current month in each latitudinal grid cell
defctcur_cmp	water deficit for the current month in each latitudinal grid cell
twarmm_cmp	temperature of the warmest month (c) in each latitudinal grid cell
tcoldm_cmp	temperature of the coldest month (c) in each latitudinal grid cell
gdd5_cmp	growing degree days above 5 c in each latitudinal grid cell
aridity_cmp	aridity index, ratio of potential evaporation to precipitation in each latitudinal grid cell
srplsmon_cmp	number of months in a year with surplus water i.e. precipitation more than potential evaporation in each latitudinal grid cell
defctmon_cmp	number of months in a year with water deficit i.e. precipitation less than potential evaporation in each latitudinal grid cell
anndefct_cmp	annual water deficit (mm) in each latitudinal grid cell
annsrpls_cmp	annual water surplus (mm) in each latitudinal grid cell
annpcp_cmp	annual precipitation (mm) in each latitudinal grid cell
dry_season_↩ length_cmp	length of dry season (months) in each latitudional grid cell

parameters used

note the structure of parameter vectors which clearly shows the class pfts (along rows) and ctem sub-pfts (along columns)

needle leaf	**evg1**	**evg2**	**dcd**
broad leaf	**evg**	**dcd-cld**	**dcd-dry**
crops	**c3**	**c4**	**—**
grasses	**c3**	**c4**	**—**

initialization

note that netradrow is not initialized here because it is updated

unmapping the pft fraction in each grid cell (fare_cmp) back to the pft index in each mosaic of each grid cell (fcancmxrow) and update the pft areal fraction in each mosaic of each grid cell (farerow)

check if bare fraction is existed

unmapping back to fcancmxrow and update farerow

gathering the pft index in each mosaic of each grid cell (fcancmxrow) to the pft index in each mosaic (fcancmx) nml, ilmos and jlmos are referring to gatprep.f

9.23 CONTRIBUTING.md File Reference

9.23.1 Detailed Description

!> documentation for this file

The '

9.24 ctem.f90 File Reference

The basic model structure of CTEM includes three live vegetation components (leaf (L), stem (S) and root (R)) and two dead carbon pools (litter or detritus (D) and soil carbon (H)). The amount of carbon in these pools (C_{L}, C_{S}, C_{R}, C_{D}, C_{H}, $kgCm^{-2}$) is tracked prognostically through the fluxes in and out of them. The rate change equations

for carbon in these pools are summarized in Sect. rate_change_eqns} after the processes leading to the calculation of fluxes in and out of these pools are introduced in the following sections.

Functions/Subroutines

- subroutine ctem (fcancmx,fsnow,sand,clay,il1,il2,iday,radj,tcano,tcans,tbarc,tbarcs,tbarg,tbargs,ta,delzw,ancsveg,ancgveg, rmlcsveg,rmlcgveg,zbotw,thliqc,thliqg,deltat,uwind,vwind,lightng,prbfrhuc,extnprob,stdaln,tbar,popdon,nol2pfts, pfcancmx, nfcancmx,lnduseon,thicec, soildpth, spinfast,todfrac,compete,netrad,precip,psisat, grclarea,popdin, dofire,dowetlands, obswetf, isand,faregat, onetile_perPFT, wetfrac, slopefrac,bi,thpor,thiceg, currlat,ch4conc,G↩ RAV,RHOW,RHOICE,leapnow,

9.24.1 Detailed Description

The basic model structure of CTEM includes three live vegetation components (leaf (L), stem (S) and root (R)) and two dead carbon pools (litter or detritus (D) and soil carbon (H)). The amount of carbon in these pools ($C_{\rm L}$, $C_{\rm S}$, $C_{\rm R}$, $C_{\rm D}$, $C_{\rm H}$, $kgCm^{-2}$) is tracked prognostically through the fluxes in and out of them. The rate change equations for carbon in these pools are summarized in Sect. rate_change_eqns} after the processes leading to the calculation of fluxes in and out of these pools are introduced in the following sections.

9.24.2 Function/Subroutine Documentation

9.24.2.1 subroutine ctem (real, dimension(ilg,icc), intent(inout) *fcancmx,* real, dimension(ilg), intent(in) *fsnow,* real, dimension(ilg,ignd), intent(in) *sand,* real, dimension(ilg,ignd), intent(in) *clay,* integer, intent(in) *il1,* integer, intent(in) *il2,* integer, intent(in) *iday,* real, dimension(ilg), intent(in) *radj,* real, dimension(ilg), intent(in) *tcano,* real, dimension(ilg), intent(in) *tcans,* real, dimension(ilg,ignd), intent(in) *tbarc,* real, dimension(ilg,ignd), intent(in) *tbarcs,* real, dimension(ilg,ignd), intent(in) *tbarg,* real, dimension(ilg,ignd), intent(in) *tbargs,* real, dimension(ilg), intent(in) *ta,* real, dimension(ilg,ignd), intent(in) *delzw,* real, dimension(ilg,icc), intent(inout) *ancsveg,* real, dimension(ilg,icc), intent(inout) *ancgveg,* real, dimension(ilg,icc), intent(inout) *rmlcsveg,* real, dimension(ilg,icc), intent(inout) *rmlcgveg,* real, dimension(ilg,ignd), intent(in) *zbotw,* real, dimension(ilg,ignd), intent(in) *thliqc,* real, dimension(ilg,ignd), intent(in) *thliqg,* real, intent(in) *deltat,* real, dimension(ilg), intent(in) *uwind,* real, dimension(ilg), intent(in) *vwind,* real, dimension(ilg), intent(in) *lightng,* real, dimension(ilg), intent(in) *prbfrhuc,* real, dimension(ilg), intent(inout) *extnprob,* integer, dimension(ilg), intent(in) *stdaln,* real, dimension(ilg,ignd), intent(in) *tbar,* logical, intent(inout) *popdon,* integer, dimension(ican) *nol2pfts,* real, dimension(ilg,icc), intent(in) *pfcancmx,* real, dimension(ilg,icc), intent(in) *nfcancmx,* logical, intent(in) *lnduseon,* real, dimension(ilg,ignd), intent(in) *thicec,* real, dimension(ilg), intent(in) *soildpth,* integer, intent(in) *spinfast,* real, dimension(ilg,icc), intent(in) *todfrac,* logical, intent(in) *compete,* real, dimension(ilg), intent(in) *netrad,* real, dimension(ilg), intent(in) *precip,* real, dimension(ilg,ignd), intent(in) *psisat,* real, dimension(ilg), intent(in) *grclarea,* real, dimension(ilg), intent(inout) *popdin,* logical, intent(in) *dofire,* logical, intent(in) *dowetlands,* logical, intent(in) *obswetf,* integer, dimension(ilg,ignd), intent(in) *isand,* real, dimension(ilg), intent(in) *faregat,* logical, intent(in) *onetile_perPFT,* real, dimension(ilg), intent(in) *wetfrac,* real, dimension(ilg,8), intent(in) *slopefrac,* real, dimension(ilg,ignd), intent(in) *bi,* real, dimension(ilg,ignd), intent(in) *thpor,* real, dimension(ilg,ignd), intent(in) *thiceg,* real, dimension(ilg), intent(in) *currlat,* real, dimension(ilg), intent(in) *ch4conc,* real, intent(in) *GRAV,* real, intent(in) *RHOW,* real, intent(in) *RHOICE,* logical, intent(in) *leapnow*)

Parameters

in	*lnduseon*	logical switch to run the land use change subroutine or not.
in	*compete*	logical boolean telling if competition between pfts is on or not
in	*dofire*	boolean, if true allow fire, if false no fire.
in	*dowetlands*	if true allow wetland methane emission
in	*obswetf*	if true, use read-in observed wetland fraction
in	*onetile_perpft*	if you are running with one tile per PFT in mosaic mode, set to true. Changes how competition is run. Specifically it allows competition between tiles. This is not recommended for any case where you don't have one PFT in each tile as it has not been tested for that.
in	*leapnow*	true if this year is a leap year. Only used if the switch 'leap' is true.
in	*iday*	day of year
in	*spinfast*	spinup factor for soil carbon whose default value is 1. as this factor increases the soil c pool will come into equilibrium faster. reasonable value for spinfast is between 5 and 10. when spinfast.ne.1 then the balcar subroutine is not run.
in	*il1*	il1=1
in	*il2*	il2=ilg (no. of grid cells in latitude circle)
in	*fsnow*	fraction of snow simulated by class
in	*sand*	percentage sand
in	*clay*	percentage clay
in	*radj*	latitude in radians
in	*tcano*	canopy temperature for canopy over ground subarea, K
in	*tcans*	canopy temperature for canopy over snow subarea, K
in	*tbar*	soil temperature, k
in	*tbarc*	soil temperature for canopy over ground subarea, K
in	*tbarcs*	soil temperature for canopy over snow subarea
in	*tbarg*	soil temperature for ground subarea
in	*tbargs*	soil temperature for snow over ground subarea
in	*psisat*	saturated soil matric potential (m)
in	*bi*	Brooks and Corey b term
in	*thpor*	Soil porosity
in	*ta*	air temp, K
in	*delzw*	thicknesses of the 3 soil layers
in	*zbotw*	bottom of soil layers
in	*soildpth*	soil depth (m)
in	*thliqc*	liquid mois. content of 3 soil layers, for canopy over snow and canopy over ground subareas
in	*thliqg*	liquid mois. content of 3 soil layers, for ground and snow over ground subareas
in	*thicec*	Frozen soil moisture content for canopy over snow and canopy over ground subareas
in	*thiceg*	Frozen soil moisture content for ground and snow over ground subareas

in	*precip*	daily precipitation (mm/day)
in	*netrad*	daily net radiation (w/m2)
in	*lightng*	total lightning frequency, flashes/km2.year
in	*prbfrhuc*	probability of fire due to human causes
in	*pfcancmx*	previous year's fractional coverages of pfts
in	*nfcancmx*	next year's fractional coverages of pfts
in	*todfrac*	max. fractional coverage of ctem's 9 pfts by the end of the day, for use by land use subroutine
in	*ch4conc*	Atmospheric CH_4 concentration at the soil surface (ppmv)
in	*stdaln*	an integer telling if ctem is operated within gcm (=0) or in stand alone mode (=1). this is used for fire purposes. see comments just above where disturb subroutine is called.
in	*grav*	Acceleration due to gravity ($m s^{-1}), (CLASS param) passed in to avoid the common block structure.
in	*rhow*	Density of water ($kg m^{-3}), (CLASS param) passed in to avoid the common block structure.
in	*rhoice*	Density of ice ($kg m^{-3}), (CLASS param) passed in to avoid the common block structure.
in,out	*popdon*	if set true use population density data to calculate fire extinguishing probability and probability of fire due to human causes, or if false, read directly from .ctm file
in,out	*ancsveg*	net photosynthetic rate for ctems 9 pfts for canopy over snow subarea
in,out	*ancgveg*	net photosynthetic rate for ctems 9 pfts for canopy over ground subarea
in,out	*rmlcsveg*	leaf respiration rate for ctems 9 pfts forcanopy over snow subarea
in,out	*rmlcgveg*	leaf respiration rate for ctems 9 pfts forcanopy over ground subarea
in,out	*extnprob*	fire extingusinging probability
in,out	*fcancmx*	max. fractional coverage of ctem's 9 pfts, but this can be modified by land-use change, and competition between pfts
in,out	*popdin*	population density $(people/km^2)$
	nol2pfts	number of level 2 ctem pfts

Constants and parameters are located in ctem_params.f90

Begin calculations

Generate the sort index for correspondence between 9 pfts and the 12 values in the parameter vectors

If you intend to have competition and LUC between tiles then set onetile_perPFT to true. NOTE: Turning onetile_↩ perPFT to true is usually not the behaviour desired unless you are running with one PFT on each tile and want them to compete for space across tiles. So in general keep this as False. JM Jan 2016.

this is composite/mosaic mode (in mosaic competition only occurs WITHIN a tile)

Calculate bioclimatic parameters for estimating pfts existence

If first day of year then based on updated bioclimatic parameters find if pfts can exist or not. If .not. inibioclim then it is the first year of a run that you do not have the climatological means already in the CTM file. After one year inibioclim is set to true and the climatological means are used from the first year.

Call competition subroutine which on the basis of previous day's npp estimates changes in fractional coverage of pfts

If landuse is on, then implelement luc, change fractional coverages, move biomasses around, and estimate luc related combustion emission losses.

Land use change and competition for mosaics needs mapping and unmapping of the pfts. Composite does not require these extra steps.

Check if number of mosaics is equal to the number of pfts plus one bare, e.g., nmos=iccp1

check for fcancmx(i,j). this should be either 0 or 1 for competition to work.

competition_map scatters and maps the array with indices of (ilg,icc) to (nlat,icc) for preparation for competition

Calculate bioclimatic parameters for estimating pfts existence

If first day of year then based on updated bioclimatic parameters find if pfts can exist or not. If .not. inibioclim then it is the first year of a run that you do not have the climatological means already in the CTM file. After one year inibioclim is set to true and the climatological means are used from the first year.

call competition subroutine which on the basis of previous day's npp estimates changes in fractional coverage of pfts

Competition_unmap unmaps and gathers the array with indices (nlat,icc) back to (ilg,icc) after competition is done

initialize required arrays to zero

Store green and brown leaf, stem, and root biomass, and litter and soil c pool mass in arrays. knowing initial sizes of all pools and final sizes at the end of this subroutine, we check for conservation of mass.

Initialization ends

Find fc and fcs based on fcancmx

Autotrophic respiration

Leaf respiration is calculated in phtsyn subroutine, while stem and root maintenance respiration are calculated here.

We treat canopy over ground and canopy over snow subareas separately because stem temperature (for which we use canopy temperature as a surrogate) can be different for these two subareas.

Find maintenance respiration for canopy over snow sub-area in umol co2/m2/sec

Find maintenance respiration for canopy over ground sub-area

If ailcg/gleafmas is zero, i.e. real leaves are not on, then make maintenance respiration and gpp from storage/imaginary lai equal to zero so that we don't use these numbers in carbon budget.

Find vegetation averaged leaf, stem, and root respiration, and gpp using values from canopy over ground and canopy over snow subareas

Now that we know maintenance respiration from leaf, stem, and root and gpp, we can find growth respiration for each vegetation

Calculate grid-averaged rates of rm, rg, npp, and gpp

```
 Heterotrophic respiration part starts
```

Find heterotrophic respiration rates for canopy over ground subarea

Find heterotrophic respiration rates from bare ground subarea

Find heterotrophic respiration rates from snow over ground subarea

Find vegetation averaged litter and soil c respiration rates using values from canopy over ground and canopy over snow subareas

Find litter and soil c respiration rates averaged over the bare fraction of the grid cell using values from ground and snow over ground sub-areas.

Find grid averaged litter and soil c respiration rates

Update the litter and soil c pools based on litter and soil c respiration rates found above. also transfer humidified litter to the soil c pool.

Convert u mol co2/m2.sec -> kgc/m^2 respired over the model time step

Update litter and soil c pools

Estimate soil respiration. this is sum of heterotrophic respiratio and root maintenance respiration.

But over the bare fraction there is no live root.

Find grid averaged humification and soil respiration rates

Find CH4 wetland area (if not prescribed) and emissions:

Calculate the methane that is oxidized by the soil sink

Estimate allocation fractions for leaf, stem, and root components.

Note: fieldsm and wiltsm are calculated in allocate. They are called THFC and PSIWLT in the CLASS part of the model. They are recalculated here in CTEM to avoid passing them through the coupler in the coupled model. The CTEM calculated versions of fieldsm and wiltsm are also used in disturb and phenolgy. JM. Jan 14 2015.

Estimate fraction of npp that is to be used for horizontal expansion (lambda) during the next day (i.e. this will be determining the colonization rate in competition).

We use the following new function to smooth the transition for lambda as an abrupt linear increase does not give good results. JM Jun 2014

If tree and leaves still coming out, or if npp is negative, then do not expand

Maintenance respiration also reduces leaf, stem, and root biomass. when npp for a given pft is positive then this is taken care by allocating +ve npp amongst the leaves, stem, and root component. when npp for a given pft is negative then maintenance respiration loss is explicitly deducted from each component.

Convert npp and maintenance respiration from different components from units of u mol co2/m2.sec -> kgc/m^2 sequestered or respired over the model time step (deltat)

Remove the cost of making reproductive tissues. This cost can only be removed when NPP is positive.

i.e. if lfstatus.eq.4 and since we do not have any real leaves on then we do not take into account co2 uptake by imaginary leaves in carbon budget. rmlvgstp(i,j) should be zero because we set maintenance respiration from storage/imaginary leaves equal to zero. in loop 180

since no real leaves are on, make allocation fractions equal to zero.

convert net change in leaf, stem, and root biomass into u-mol co2/m2.sec for use in balcar subroutine

to avoid over/underflow problems set gleafmas, stemmass, and rootmass to zero if they get too small

calculate grid averaged value of C related to spatial expansion

Phenology part starts

the phenology subroutine determines leaf status for each pft and calculates leaf litter. the phenology subroutine uses soil temperature (tbar) and root temperature. however, since ctem doesn't make the distinction between canopy over ground, and canopy over snow sub-areas for phenology purposes (for example, leaf onset is not assumed to occur at different times over these sub-areas) we use average soil and root temperature in the phenology subroutine.

calculate average soil temperature and root temperature using values for canopy over ground and canopy over snow sub-areas, for each vegetation type.

Call the phenology subroutine, which determines the leaf growth status, calculates leaf litter, and converts green grass into brown.

while leaf litter is calculated in the phenology subroutine, stem and root turnover is calculated in the turnover subroutine.

Update green leaf biomass for trees and crops and brown leaf biomass for grasses

Update stem and root biomass for litter deductions

Update litter pool with leaf litter calculated in the phenology subroutine and stem and root litter calculated in the turnover subroutine. Also add the reproduction carbon directly to the litter pool

Call the mortaliy subroutine which calculates mortality due to reduced growth and aging. exogenous mortality due to fire and other disturbances and the subsequent litter that is generated is calculated in the disturb subroutine.

set maxage $>$0 in ctem_params.f90 to switch on mortality due to age and reduced growth. Mortality is linked to the competition parameterization and generates bare fraction.

Update leaf, stem, and root biomass pools to take into loss due to mortality, and put the litter into the litter pool. the mortality for green grasses doesn't generate litter, instead they turn brown.

call the disturbance subroutine which calculates mortality due to fire and other disturbances. the primary output from from disturbance subroutine is litter generated, c emissions due to fire and area burned, which may be used

to estimate change in fractional coverages.

disturbance is spatial and requires area of gcm grid cell and areas of different pfts present in a given grid cell. however, when ctem is operated at a point scale then it is assumed that the spatial scale is 1 hectare = 10,000 m2. the disturbance subroutine may be stopped from simulating any fire by specifying fire extingushing probability equal to 1.

Calculate nbp (net biome production) for each pft by taking into account C emission losses. The disturbance routine produces emissions due to fire and it also calculates emissions due to LUC. These LUC carbon emissions due to combustion associated with LUC are first estimated in LUC. This flux is spread out over the whole year and is therefore subtracted to get NBP of each pft as well as the grid averaged value of NBP. Also LUC related combustion flux is assumed to be spread uniformly over the grid cell and thus reduces NBP of each PFT

Calculate grid. averaged rate of carbon emissions due to fire in u-mol co2/m2.sec. convert all emission losses from kgc/m^2 emitted in 1 day to u-mol co2/m2.sec. calculate grid averaged carbon emission losses from litter.

calculate total litter fall from each component (leaves, stem, and root) from all causes (normal turnover, drought and cold stress for leaves, mortality, and disturbance) for use in balcar subroutine

units here are $kgc/m^2.day$

convert units to u-mol co2/m2.sec

calculate grid-average vegetation biomass, litter mass, and soil carbon mass, and litter fall rate

At this stage we have all required fluxes in u-mol co2/m2.sec and initial (loop 140 and 145) and updated sizes of all pools (in kgc/m^2). Now we call the balcar subroutine and make sure that C in leaves, stem, root, litter and soil C pool balances within a certain tolerance.

Finally find vegetation structural attributes which can be passed to the land surface scheme using leaf, stem, and root biomass.

Calculation of gavglai is moved from loop 1100 to here since ailcg is updated by bio2str

9.25 ctem_params.f90 File Reference

This module holds CTEM globally accessible parameters These parameters are used in all CTEM subroutines via use statements pointing to this module EXCEPT PHTSYN3.f which has the information passed in via arguments. This is a legacy thing.

Functions/Subroutines

- subroutine, public ctem_params::initpftpars (compete)

Variables

- real, parameter **ctem_params::zero** = 1.0e-20
- real, parameter **ctem_params::abszero** = 1e-12

  this one is for runclassctem.f and allocate.f
- real, parameter **ctem_params::pi** = 3.1415926535898d0
- real, parameter **ctem_params::earthrad** = 6371.22

  radius of earth, km
- real, parameter **ctem_params::km2tom2** = 1.0e+06

  changes from km^2 to m^2
- real, parameter **ctem_params::deltat** = 1.0

  CTEM's time step in days.
- integer, dimension(12) **ctem_params::monthdays** = [31,28,31,30,31,30,31,31,30,31,30,31]

  days in each month
- integer, dimension(13) **ctem_params::monthend** = [0,31,59,90,120,151,181,212,243,273,304,334,365]

calender day at end of each month

- integer, dimension(12) **ctem_params::mmday** = [16,46,75,106,136,167,197,228,259,289,320,350]

  mid-month day
- integer, parameter **ctem_params::lon** = 128

  specify gcm resolution for longitude
- integer, parameter **ctem_params::lat** = 64

  specify gcm resolution for latitude
- real, dimension(lat+1), parameter **ctem_params::edgelat** = [-90.0,-86.4802,-83.7047,-80.9193,-78.1313,-75.3422,-72.5527, -69.7628,-66.9727,-64.1825,-61.3922,-58.6018,-55.8114,-53.021, -50.2305,-47.44,-44.6495,-41.8589,-39.0684,-36.2778,-33.4872, -30.6967,-27.9061,-25.1155,-22.3249,-19.5343,-16.7437,-13.9531, -11.1625,-8.3718,-5.5812,-2.7906,0.0,2.7906,5.5812,8.3718,11.16245, 13.9531,16.7437,19.↩5343,22.3249,25.1155,27.9061,30.69665,33.4872, 36.2778,39.06835,41.8589,44.64945,47.43995,50.↩23045,53.02095, 55.8114,58.6018,61.3922,64.1825,66.9727,69.7628,72.55265,75.3422, 78.13125,80.↩91925,83.7047,86.48015,90.0]

  latitudes of the edges of the gcm grid cells for 128/x64 resolution
- integer, parameter **ctem_params::nlat** = 1
- integer, parameter **ctem_params::nmos** = 10

  Number of mosaic tiles.
- integer, parameter **ctem_params::ilg** = nlat∗nmos
- integer, parameter **ctem_params::nmon** = 12

  Number of months in a year.
- integer, parameter **ctem_params::ican** = 4

  Number of CLASS pfts.
- integer, parameter **ctem_params::ignd** = 20

  Number of soil layers.
- integer, parameter **ctem_params::icp1** = ican + 1
- integer, parameter **ctem_params::icc** = 9

  Number of CTEM pfts.
- integer, parameter **ctem_params::iccp1** = icc + 1
- integer, parameter **ctem_params::l2max** = 3
- integer, parameter **ctem_params::kk** = 12

  product of class pfts and l2max
- integer, parameter **ctem_params::numcrops** = 2

  number of crop pfts
- integer, parameter **ctem_params::numtreepfts** = 5

  number of tree pfts
- integer, parameter **ctem_params::numgrass** = 2

  number of grass pfts
- integer, parameter **ctem_params::nbs** = 4
- integer, dimension(kk), parameter **ctem_params::modelpft** = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0]

  Separation of pfts into level 1 (for class) and level 2 (for ctem) pfts.
- real, parameter **ctem_params::seed** = 0.001

  seed pft fraction, same as in competition mosaic mode, all tiles are given this as a minimum
- real, parameter **ctem_params::minbare** = 1.0e-5

  minimum bare fraction when running competition on to prevent numerical problems.
- real, parameter **ctem_params::c2dom** = 450.0

  gc / kg dry organic matter factor from carbon to dry organic matter value is from Li et al. 2012 biogeosci
- real, parameter **ctem_params::wtch4** = 16.044

  Molar mass of CH4 ($g mol^{-1}$)
- logical, dimension(icc), parameter **ctem_params::crop** = [.false.,.false.,.false.,.false.,.false.,.true.,.true.,.false.,.false.]

  simple crop matrix, define the number and position of the crops (NOTE: dimension icc)

- logical, dimension(icc), parameter **ctem_params::grass** = [.false.,.false.,.false.,.false.,.false.,.false.,.false.,.true.,.true.]

  simple grass matric, define the number and position of grass
- integer, dimension(numgrass), parameter **ctem_params::grass_ind** = [8, 9]

  index of the grass pfts (only 2 grass pfts at present)
- real **ctem_params::tolrance** = 0.0001d0

  our tolerance for balancing c budget in kg c/m2 in one day (differs when competition on or not)
- real, dimension(kk) **ctem_params::kn** = [0.50, 0.50, 0.00, 0.50, 0.50, 0.50, 0.40, 0.48, 0.00, 0.46, 0.44, 0.00]

  Canopy light/nitrogen extinction coefficient (kn -> CAREFUL: Separate set defined in PHTSYN3.f!)
- real, dimension(kk) **ctem_params::omega**

  omega, parameter used in allocation formulae
- real, dimension(kk) **ctem_params::epsilonl**

  Epsilon leaf, parameter used in allocation formulae.
- real, dimension(kk) **ctem_params::epsilons**

  Epsilon stem, parameter used in allocation formulae.
- real, dimension(kk) **ctem_params::epsilonr**

  Epsilon root, parameter used in allocation formulae.
- logical **ctem_params::consallo** = .false.

  Logical switch for using constant allocation factors (default value is false)
- real, dimension(kk) **ctem_params::rtsrmin** = [0.16, 0.16, 0.00, 0.16, 0.16, 0.32, 0.16, 0.16, 0.00, 0.50, 0.50, 0.00]

  Minimum root:shoot ratio mostly for support and stability.
- real, dimension(kk) **ctem_params::aldrlfon** = [1.00, 1.00, 0.00, 1.00, 1.00, 1.00, 1.00, 1.00, 0.00, 1.00, 1.00, 0.00]

  Allocation to leaves during leaf onset.
- real, dimension(kk) **ctem_params::caleaf** = [0.275, 0.300, 0.000, 0.200, 0.250, 0.250, 0.400, 0.400, 0.000, 0.450, 0.450, 0.000]
- real, dimension(kk) **ctem_params::castem** = [0.475, 0.450, 0.000, 0.370, 0.400, 0.400, 0.150, 0.150, 0.000, 0.000, 0.000, 0.000]
- real, dimension(kk) **ctem_params::caroot** = [0.250, 0.250, 0.000, 0.430, 0.350, 0.350, 0.450, 0.450, 0.000, 0.550, 0.550, 0.000]
- real, dimension(kk) **ctem_params::abar** = [4.70, 5.86, 0.00, 3.87, 3.46, 3.97, 3.97, 3.97, 0.00, 5.86, 4.92, 0.00]

  parameter determining average root profile
- real, dimension(kk) **ctem_params::avertmas** = [1.85, 1.45, 0.00, 2.45, 2.10, 2.10, 0.10, 0.10, 0.00, 0.70, 0.70, 0.00]

  average root biomass (kg c/m2) for ctem's 8 pfts used for estimating rooting profile
- real, dimension(kk) **ctem_params::alpha** = [0.80, 0.80, 0.00, 0.80, 0.80, 0.80, 0.80, 0.80, 0.00, 0.80, 0.80, 0.00]

  parameter determining how the roots grow
- real, dimension(kk) **ctem_params::prcnslai** = [7.5, 7.5, 0.0, 7.5, 7.5, 7.5, 7.5, 7.5, 0.0, 2.5, 2.5, 0.0]

  storage/imaginary lai is this percentage of maximum leaf area index that a given root+stem biomass can support
- real, dimension(kk) **ctem_params::minslai** = [0.3, 0.3, 0.0, 0.3, 0.3, 0.3, 0.2, 0.2, 0.0, 0.2, 0.2, 0.0]

  minimum storage lai. this is what the model uses as lai when growing vegetation for scratch. consider these as model seeds.
- real, dimension(kk) **ctem_params::mxrtdpth** = [3.00, 3.00, 0.00, 5.00, 5.00, 3.00, 2.00, 2.00, 0.00, 1.00, 1.00, 0.00]

  maximum rooting depth. this is used so that the rooting depths simulated by ctem's variable rooting depth parameterzation are constrained to realistic values visible and near ir albedos of the 9 ctem pfts
- real, dimension(kk) **ctem_params::albvis** = [3.00, 3.00, 0.00, 3.00, 5.00, 5.00, 5.50, 5.50, 0.00, 5.00, 6.00, 0.00]

  visible albedos of the 9 ctem pfts

- real, dimension(kk) **ctem_params::albnir** = [19.0, 19.0, 0.00, 23.0, 29.0, 29.0, 34.0, 34.0, 0.00, 30.0, 34.0, 0.00]

  near IR albedos of the 9 ctem pfts
- real, dimension(kk) **ctem_params::tcoldmin** = [-999.9, -999.9, 0.0, 2.5, -35.0, 4.0, -999.9, -999.9, 0.0, -999.9, -999.9, 0.0]

  minimum coldest month temperature
- real, dimension(kk) **ctem_params::tcoldmax** = [18.0, -28.0, 0.0, 999.9, 16.0, 900.0, 999.9, 999.9, 0.0, 999.9, 999.9, 0.0]

  maximum coldest month temperature
- real, dimension(kk) **ctem_params::twarmmax** = [99.9, 25.0, 0.0, 99.9, 99.9, 99.9, 99.9, 99.9, 0.0, 99.9, 99.9, 0.0]

  maximum warmest month temperature
- real, dimension(kk) **ctem_params::gdd5lmt** = [375.0, 600.0, 0.0, 1200.0, 300.0, 9.9, 9.9, 9.9, 0.0, 9.9, 9.9, 0.0]

  minimum gdd above 5 c required to exist
- real, dimension(kk) **ctem_params::aridlmt** = [9.9, 9.9, 0.0, 9.9, 9.9, 0.9, 9.9, 9.9, 0.0, 9.9, 9.9, 0.0]

  aridity index limit for broadleaf drought/dry deciduous trees
- real, dimension(kk) **ctem_params::dryseasonlmt** =[99.0, 99.9, 0.0, 99.9, 99.9, 5.5, 99.9, 99.9, 0.0, 99.9, 99.9, 0.0]

  minimum length of dry season for PFT to exist
- real, dimension(kk) **ctem_params::bio2sap** = [0.32, 0.20, 0.00, 0.08, 0.14, 0.13, 0.00, 0.00, 0.00, 0.20, 0.20, 0.00]

  multiplying factor for converting biomass density to sapling density
- real **ctem_params::bioclimrt** = 0.25

  mortality rate (1/year) for pfts that no longer exist within their pre-defined bioclimatic range
- real, dimension(kk) **ctem_params::grescoef** = [0.15, 0.15, 0.00, 0.15, 0.15, 0.15, 0.15, 0.15, 0.00, 0.15, 0.15, 0.00]

  Growth respiration coefficient.
- real, dimension(kk) **ctem_params::humicfac** = [0.42, 0.42, 0.00, 0.53, 0.48, 0.48, 0.10, 0.10, 0.00, 0.42, 0.42, 0.00]

  Humification factor - used for transferring carbon from litter into soil c pool.
- real, dimension(kk) **ctem_params::laimin**

  Minimum lai below which a pft doesn't expand.
- real, dimension(kk) **ctem_params::laimax**

  Maximum lai above which a pft always expands and lambdamax fraction of npp is used for expansion.
- real **ctem_params::lambdamax** = 0.10

  Max. fraction of npp that is allocated for reproduction/colonization.
- real **ctem_params::repro_fraction** = 0.10

  Fraction of NPP that is used to create reproductive tissues.
- real, dimension(2) **ctem_params::bmasthrs_fire** = [0.4, 1.2]

  min. and max. vegetation biomass thresholds to initiate fire, kgc/m^2
- real **ctem_params::extnmois_veg** = 0.3

  extinction moisture content for estimating vegetation fire likeliness due to soil moisture
- real **ctem_params::extnmois_duff** = 0.5

  extinction moisture content for estimating duff layer fire likeliness due to soil moisture
- real **ctem_params::lwrlthrs** = 0.25

  lower cloud-to-ground lightning threshold for fire likelihood flashes/km^ 2.year
- real **ctem_params::hgrlthrs** = 10.0

  higher cloud-to-ground lightning threshold for fire likelihood flashes/km^ 2.year
- real **ctem_params::parmlght** = 0.8

  parameter m (mean) and b of logistic distribution used for
  ∗∗Parmlght was increased to 0.8 to make it so areas with higher amounts of lightning have higher lterm. The saturation is still the same, but the increase is more gradual at low lightning density. JM

- real **ctem_params::parblght** = 0.1

  estimating fire likelihood due to lightning
- real **ctem_params::reparea** = 500.0

  typical area representing ctem's fire parameterization (km2)
- real **ctem_params::popdthrshld** = 300.

  threshold of population density (people/km2) [Kloster et al., biogeosci. 2010]
- real **ctem_params::alpha_fire**

  parameter alpha_fire and f0 used for estimating wind function for fire spread rate
- real **ctem_params::f0** = 0.05

  Fire spread rate in the absence of wind – Not used in CTEM v 2.0!
- real, dimension(kk) **ctem_params::standreplace** = [0.20, 0.20, 0.00, 0.50, 0.20, 0.15, 0.00, 0.00, 0.00, 0.25, 0.25, 0.00]

  pft prevalence for stand replacing fire events (based on resistance to fire damage, ie. cambial kill)(unitless)
- real, dimension(kk) **ctem_params::maxsprd** = [0.38, 0.38, 0.00, 0.28, 0.28, 0.28, 0.00, 0.00, 0.00, 0.51, 0.75, 0.00]

  max. fire spread rate, km/hr
- real, dimension(kk) **ctem_params::frco2glf** = [0.42, 0.42, 0.00, 0.42, 0.42, 0.42, 0.00, 0.00, 0.00, 0.48, 0.48, 0.00]

  fraction of green leaf biomass converted to gases due to combustion
- real, dimension(kk) **ctem_params::frco2blf** = [0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.54, 0.54, 0.00]

  fraction of brown leaf biomass converted to gases due to combustion
- real, dimension(kk) **ctem_params::frltrglf** = [0.20, 0.20, 0.00, 0.20, 0.20, 0.20, 0.00, 0.00, 0.00, 0.10, 0.10, 0.00]

  fraction of green leaf biomass becoming litter after combustion
- real, dimension(kk) **ctem_params::frltrblf** = [0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.06, 0.06, 0.00]

  fraction of brown leaf biomass becoming litter after combustion
- real, dimension(kk) **ctem_params::frco2stm** = [0.12, 0.12, 0.00, 0.12, 0.06, 0.06, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]

  fraction of stem biomass converted to gases due to combustion
- real, dimension(kk) **ctem_params::frltrstm** = [0.60, 0.60, 0.00, 0.60, 0.40, 0.40, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]

  fraction of stem biomass becoming litter after combustion
- real, dimension(kk) **ctem_params::frco2rt** = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

  fraction of root biomass converted to gases due to combustion
- real, dimension(kk) **ctem_params::frltrrt** = [0.10, 0.10, 0.00, 0.10, 0.10, 0.10, 0.00, 0.00, 0.00, 0.25, 0.25, 0.00]

  fraction of root biomass becoming litter after combustion
- real, dimension(kk) **ctem_params::frltrbrn** = [0.30, 0.30, 0.00, 0.36, 0.36, 0.36, 0.00, 0.00, 0.00, 0.42, 0.42, 0.00]

  fraction of litter burned during fire and emitted as gases
- real, dimension(kk) **ctem_params::emif_co2** = [1576.0, 1576.0, 0.00, 1604.0, 1576.0, 1654.0, 1576.0, 1654.0, 0.00, 1576.0, 1654.0, 0.00]

  pft-specific emission factors for CO2,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_co** = [106.0, 106.0, 0.00, 103.0, 106.0, 64.0, 106.0, 64.0, 0.00, 106.0, 64.0, 0.00]

  pft-specific emission factors for CO,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_ch4** = [4.8, 4.8, 0.0, 5.8, 4.8, 2.4, 4.8, 2.4, 0.0, 4.8, 2.4, 0.0]

  pft-specific emission factors for CH4,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_nmhc** = [5.7, 5.7, 0.0, 6.4, 5.7, 3.7, 5.7, 3.7, 0.0, 5.7, 3.7, 0.0]

  pft-specific emission factors for non-methane hydrocarbons,g species / (kg DOM)

- real, dimension(kk) **ctem_params::emif_h2** = [1.80, 1.80, 0.00, 2.54, 1.80, 0.98, 1.80, 0.98, 0.00, 1.80, 0.98, 0.00]

  pft-specific emission factors for H2,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_nox** = [3.24, 3.24, 0.00, 2.90, 3.24, 2.49, 3.24, 2.49, 0.00, 3.24, 2.49, 0.00]

  pft-specific emission factors for NOx,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_n2o** = [0.26, 0.26, 0.00, 0.23, 0.26, 0.20, 0.26, 0.20, 0.00, 0.26, 0.20, 0.00]

  pft-specific emission factors for N2O,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_pm25** = [12.7, 12.7, 0.0, 10.5, 12.7, 5.2, 12.7, 5.2, 0.0, 12.7, 5.2, 0.0]

  pft-specific emission factors for particles <2.5 micrometers in diameter,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_tpm** = [17.6, 17.6, 0.0, 14.7, 17.6, 8.5, 17.6, 8.5, 0.0, 17.6, 8.5, 0.0]

  pft-specific emission factors for total particulate matter,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_tc** = [8.3, 8.3, 0.0, 7.2, 8.3, 3.4, 8.3, 3.4, 0.0, 8.3, 3.4, 0.0]

  pft-specific emission factors for total carbon,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_oc** = [9.1, 9.1, 0.0, 6.7, 9.1, 3.2, 9.1, 3.2, 0.0, 9.1, 3.2, 0.0]

  pft-specific emission factors for organic carbon,g species / (kg DOM)
- real, dimension(kk) **ctem_params::emif_bc** = [0.56, 0.56, 0.00, 0.56, 0.56, 0.47, 0.56, 0.47, 0.00, 0.56, 0.47, 0.00]

  pft-specific emission factors for black carbon,g species / (kg DOM)
- real, dimension(kk) **ctem_params::bsratelt** = [0.4453, 0.5986, 0.0000, 0.6339, 0.7576, 0.6957, 0.6000, 0.6000, 0.0000, 0.5260, 0.5260, 0.0000]

  litter respiration rates at 15 c in in kg c/kg c.year
- real, dimension(kk) **ctem_params::bsratesc** = [0.0260, 0.0260, 0.0000, 0.0208, 0.0208, 0.0208, 0.0350, 0.0350, 0.0000, 0.0125, 0.0125, 0.0000]

  soil carbon respiration rates at 15 c in kg c/kg c.year
- real, dimension(4) **ctem_params::tanhq10** = [1.44, 0.56, 0.075, 46.0]

  Constants used in tanh formulation of respiration Q10 determination.
- real **ctem_params::alpha_hetres** = 0.7

  parameter for finding litter temperature as a weighted average of top soil layer temperature and root temperature
- real **ctem_params::bsratelt_g** = 0.5605

  bare ground litter respiration rate at 15 c in kg c/kg c.year
- real **ctem_params::bsratesc_g** = 0.02258

  bare ground soil c respiration rates at 15 c in kg c/kg c.year
- real **ctem_params::a** = 4.0

  parameter describing exponential soil carbon profile. used for estimating temperature of the carbon pool
- real, dimension(3) **ctem_params::combust** = [0.15, 0.30, 0.45]

  how much deforested/chopped off biomass is combusted (these absolutely must add to 1.0!)
- real, dimension(3) **ctem_params::paper** = [0.70, 0.70, 0.55]

  how much deforested/chopped off biomass goes into short term storage such as paper
- real, dimension(3) **ctem_params::furniture** = [0.15, 0.00, 0.00]

  how much deforested/chopped off biomass goes into long term storage such as furniture
- real, dimension(2) **ctem_params::bmasthrs** = [4.0, 1.0]

  biomass thresholds for determining if deforested area is a forest, a shrubland, or a bush kg c/m2
- real **ctem_params::tolrnce1** = 0.50

  kg c, tolerance of total c balance
- real, dimension(kk) **ctem_params::bsrtstem**
- real, dimension(kk) **ctem_params::bsrtroot**
- real **ctem_params::minlvfr** = 0.05

  Minimum live wood fraction.

- real **ctem_params::kmort1** = 0.3

  kmort1, parameter used in growth efficiency mortality formulation
- real, dimension(kk) **ctem_params::mxmortge**

  Maximum mortality when growth efficiency is zero (1/year)
- real, dimension(kk) **ctem_params::maxage**

  Maximum plant age. used to calculate intrinsic mortality rate. maximum age for crops is set to zero since they will be harvested anyway. grasses are treated the same way since the turnover time for grass leaves is 1 year and for roots is 2 year.
- real, dimension(kk) **ctem_params::eta** = [10.0, 30.8, 0.00, 31.0, 50.0, 30.0, 7.0, 7.0, 0.00, 3.0, 3.0, 0.00]

  eta and kappa, parameters for estimating min. stem+root biomass
- real, dimension(kk) **ctem_params::kappa** = [1.6, 1.6, 0.0, 1.6, 1.6, 1.6, 1.6, 1.6, 0.0, 1.2, 1.2, 0.0]

  required to support green leaf biomass. kappa is 1.6 for trees and crops, and 1.2 for grasses.
- real, dimension(kk) **ctem_params::lfespany** = [5.00, 1.00, 0.00, 1.50, 1.00, 1.00, 1.75, 1.75, 0.00, 1.00, 1.00, 0.00]

  Leaf life span (in years) for CTEM's 9 pfts.
- real, dimension(kk) **ctem_params::specsla** =[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

  CTEM can use user-specified specific leaf areas (SLA) if the following specified values are greater than zero.
- real **ctem_params::fracbofg** = 0.55

  Parameter used to estimate lai of brown leaves. We assume that SLA of brown leaves is this fraction of SLA of green leaves.
- real, dimension(kk) **ctem_params::cdlsrtmx** = [0.10, 0.30, 0.00, 0.30, 0.40, 0.15, 0.15, 0.15, 0.00, 0.15, 0.15, 0.00]

  Max. loss rate for cold stress for all 9 pfts, (1/day)
- real, dimension(kk) **ctem_params::drlsrtmx**

  Max. loss rate for drought stress for all 9 pfts, (1/day)
- real, dimension(kk) **ctem_params::drgta** = [3.0, 3.0, 0.0, 3.0, 3.0, 3.0, 3.0, 3.0, 0.0, 3.0, 3.0, 0.0]

  Parameter determining how fast soil dryness causes leaves to fall.
- real, dimension(kk) **ctem_params::colda** = [3.0, 3.0, 0.0, 3.0, 3.0, 3.0, 3.0, 3.0, 0.0, 3.0, 3.0, 0.0]

  Parameter determining how fast cold temperatures causes leaves to fall.
- real, dimension(kk) **ctem_params::lwrthrsh** = [-50.0, -5.0, 0.0, 5.0, 8.0, 5.0, 5.0, 5.0, 0.0, 0.1, 5.0, 0.0]

  Lower temperature threshold for ctem's 9 pfts. these are used to estimate cold stress related leaf loss rate (degree c)
- integer, dimension(kk) **ctem_params::dayschk** = [7, 7, 0, 7, 7, 7, 7, 7, 0, 7, 7, 0]

  Number of days over which to check if net photosynthetic rate is positive before initiating leaf onset.
- integer, dimension(2) **ctem_params::coldlmt** = [7 , 5]

  No. of days for which some temperature has to remain below a given threshold for initiating a process.
- real, dimension(2) **ctem_params::coldthrs** = [-5.0 , 8.0]
- real, dimension(kk) **ctem_params::harvthrs** = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 4.5, 3.5, 0.0, 0.0, 0.0, 0.0]

  LAI threshold for harvesting crops. values are zero for all pftsother than c3 and c4 crops.
- real, dimension(2) **ctem_params::flhrspan** = [17.0, 45.0]

  Harvest span (time in days over which crops are harvested, 15 days), and fall span (time in days over which bdl cold dcd plants shed their leaves, 30 days)
- real, dimension(kk) **ctem_params::thrprcnt** = [40.0, 40.0, 0.0, 40.0, 50.0, 50.0, 50.0, 50.0, 0.0, 40.0, 40.0, 0.0]

  Percentage of max. LAI that can be supported which is used as a threshold for determining leaf phenology status.
- real **ctem_params::roothrsh** = 8.0

  Root temperature threshold for initiating leaf onset for cold broadleaf deciduous pft, degrees celcius.
- real, dimension(kk) **ctem_params::stemlife** = [86.3, 86.3, 0.00, 80.5, 80.5, 75.8, 20.0, 20.0, 0.00, 0.00, 0.00, 0.00]

  Stemlife, turnover time scale for stem for different pfts.
- real, dimension(kk) **ctem_params::rootlife** = [13.8,13.2, 0.0, 12.7,10.9, 9.8, 3.0, 3.0, 0.0, 3.0, 3.0, 0.0]

  Rootlife, turnover time scale for root for different pfts.
- real **ctem_params::stmhrspn** = 17.0

Stem harvest span. same as crop harvest span. period in days over which crops are harvested.

- real **ctem_params::ratioch4** = 0.135

  methane to carbon dioxide flux scaling factor.
  Rita Wania's thesis suggests about 0.25, but we get a better agreement to outputs from the Walter's model if we use 0.16. Note that this scaling factor likely is temperature dependent, and increases with temperature, but it is difficult to know the function, so leave constant for now ratio is $molch_4$ *to* $molco_2$

- real **ctem_params::wtdryres** = 0.45

  ratio of wetland to upland respiration
  Use the heterotrophic respiration outputs for soil and litter as the ecosystem basis. These were summed as "hetrores". This respiration is for upland soils; we multiply by wtdryres as the ratio of wetland to upland respiration based on literature measurements: Dalva et al. 1997 found 0.5 factor; Segers 1998 found a 0.4 factor. use 0.45 here (unitless)

- real **ctem_params::factor2** = 0.015

  constant value for secondary (ch4wet2) methane emissions calculation

- real **ctem_params::lat_thrshld1** = 40.0

  Northern zone for wetland determination (degrees North)

- real **ctem_params::lat_thrshld2** = -35.0

  Boundary with southern zone for wetland determination (degrees North)

- real **ctem_params::soilw_thrshn** = 0.55

  Soil wetness threshold in the North zone.

- real **ctem_params::soilw_thrshe** = 0.80

  Soil wetness threshold in the Equatorial zone.

- real **ctem_params::soilw_thrshs** = 0.70

  Soil wetness threshold in the South zone.

9.25.1 Detailed Description

This module holds CTEM globally accessible parameters These parameters are used in all CTEM subroutines via use statements pointing to this module EXCEPT PHTSYN3.f which has the information passed in via arguments. This is a legacy thing.

The structure of this subroutine is variables that are common to competition/prescribe PFT fractions first, then the remaining variables are assigned different variables if competition is on, or not.

PFT parameters

Note the structure of vectors which clearly shows the CLASS PFTs (along rows) and CTEM sub-PFTs (along columns)

needle leaf	evg	dcd	—
broad leaf	evg	dcd-cld	dcd-dry
crops	c3	c4	—
grasses	c3	c4	—

9.26 ctem_statevars.f90 File Reference

this module contains the variable type structures:

Data Types

- type ctem_statevars::ctem_switches

  switches for running CTEM, read from the joboptions file

- type ctem_statevars::veg_rot

CTEM's 'rot' vars.

- type ctem_statevars::veg_gat

  CTEM's 'gat' vars.
- type ctem_statevars::class_moyr_output

  CLASS's monthly outputs.
- type ctem_statevars::ctem_gridavg

  CTEM's grid average variables.
- type ctem_statevars::ctem_tile_level

  CTEM's variables per tile.
- type ctem_statevars::ctem_monthly

  CTEM's variables monthly averaged (per pft)
- type ctem_statevars::ctem_gridavg_monthly

  CTEM's grid average monthly values.
- type ctem_statevars::ctem_tileavg_monthly

  CTEM's variables per tile monthly values.
- type ctem_statevars::ctem_annual

  CTEM's average annual values (per PFT)
- type ctem_statevars::ctem_gridavg_annual

  CTEM's grid average annual values.
- type ctem_statevars::ctem_tileavg_annual

  CTEM's variables per tile annual values.

Functions/Subroutines

- subroutine, public **ctem_statevars::initrowvars** ()
- subroutine, public **ctem_statevars::resetclassmon** (nltest)
- subroutine, public **ctem_statevars::resetclassyr** (nltest)
- subroutine, public **ctem_statevars::resetdaily** (nltest, nmtest)
- subroutine, public **ctem_statevars::resetmonthend** (nltest, nmtest)
- subroutine, public **ctem_statevars::resetyearend** (nltest, nmtest)
- subroutine, public **ctem_statevars::resetclassaccum** (nltest, nmtest)
- subroutine, public **ctem_statevars::resetgridavg** (nltest)
- subroutine, public **ctem_statevars::finddaylength** (solday, radl, daylength)

Variables

- type(ctem_switches), target, save **ctem_statevars::c_switch**
- type(veg_rot), target, save **ctem_statevars::vrot**
- type(veg_gat), target, save **ctem_statevars::vgat**
- type(class_moyr_output), target, save **ctem_statevars::class_out**
- type(ctem_gridavg), target, save **ctem_statevars::ctem_grd**
- type(ctem_tile_level), target, save **ctem_statevars::ctem_tile**
- type(ctem_monthly), target, save **ctem_statevars::ctem_mo**
- type(ctem_gridavg_monthly), target, save **ctem_statevars::ctem_grd_mo**
- type(ctem_tileavg_monthly), target, save **ctem_statevars::ctem_tile_mo**
- type(ctem_annual), target, save **ctem_statevars::ctem_yr**
- type(ctem_gridavg_annual), target, save **ctem_statevars::ctem_grd_yr**
- type(ctem_tileavg_annual), target, save **ctem_statevars::ctem_tile_yr**

9.26.1 Detailed Description

this module contains the variable type structures:

1. c_switch - switches for running CTEM, read from the joboptions file
2. vrot - CTEM's 'rot' vars
3. vgat - CTEM's 'gat' vars
4. class_out - CLASS's monthly outputs
5. ctem_grd - CTEM's grid average variables
6. ctem_tile - CTEM's variables per tile
7. ctem_mo - CTEM's variables monthly averaged (per pft)
8. ctem_grd_mo - CTEM's grid average monthly values
9. ctem_tile_mo - CTEM's variables per tile monthly values
10. ctem_yr - CTEM's average annual values (per PFT)
11. ctem_grd_yr - CTEM's grid average annual values
12. ctem_tile_yr - CTEM's variables per tile annual values

9.27 ctemg1.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM)

Functions/Subroutines

- subroutine **ctemg1** (gleafmasgat, bleafmasgat, stemmassgat, rootmassgat, fcancmxgat, zbtwgat, dlzwgat, sdepgat, ailcggat, ailcbgat, ailcgat, zolncgat, rmatcgat, rmatctemgat, slaigat, bmasveggat, cmasvegcgat, veghghtgat, rootdpthgat, alvsctmgat, alirctmgat, paicgat, slaicgat, faregat, ilmos, jlmos, iwmos, jwmos, nml, gleafmasrow, bleafmasrow, stemmassrow, rootmassrow, fcancmxrow, zbtwrow, dlzwrow, sdeprow, ailcgrow, ailcbrow, ailcrow, zolncrow, rmatcrow, rmatctemrow, slairow, bmasvegrow, cmasvegcrow, veghghtrow, rootdpthrow, alvsctmrow, alirctmrow, paicrow, slaicrow, FAREROT)

9.27.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM)

9.28 ctemg2.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM)

Functions/Subroutines

- subroutine **ctemg2** (fcancmxgat,rmatcgat,zolncgat,paicgat,ailcgat,ailcggat,cmasvegcgat,slaicgat,ailcgsgat,fcancsgat,fcancgat,r canresgat,sdepgat,ch4concgat,sandgat,claygat,orgmgat,anveggat,rmlveggat,tcanoaccgat_m, tbaraccgat_↩ m,uvaccgat_m,vvaccgat_m,mlightnggat,prbfrhucgat,extnprobgat, stdalngat,pfcancmxgat,nfcancmxgat,stemmassgat, rootmassgat, litrmassgat,gleafmasgat,bleafmasgat, soilcmasgat, ailcbgat,flhrlossgat,pandaysgat,lfstatusgat,

grwtheffgat,lystmmasgat,lyrotmasgat, tymaxlaigat, vgbiomasgat,gavgltmsgat,stmhrlosgat, bmasveggat,colddaysgat,rothrlosgat,alvsctmgat,alirctmgat,gavglaigat,nppgat,nepgat,hetroresgat, autoresgat,soilrespgat,rmgat,rggat,nbpveghghtgat,rootdpthgat,rmlgat,rmsgat,rmrgat,tltrleafgat,tltrstemgat,tltrrootgat, leaflitrgat, roottempgat,afrleafgat,afrstemgat,afrrosmfuncveggat, lucemcomgat,lucltringat,lucsocingat, nppveggat,dstcemls3gat, popdingat,faregat,gavgscmsgat, rmlvegaccgat, pftexistgat,rmsveggat,rmrveggat,rgveggat,vgbiomas_veggat,gppveggat,nepveggat,ailcmingat,ailcmaxgat,emit↩_co2gat,emit_cogat, emit_ch4gat,emit_nmhcgat,emit_h2gat,emit_noxgat, emit_n2ogat,emit_pm25gat,emit↩_tpmgat,emit_tcgat, emit_ocgat,emit_bcgat,btermgat,ltermgat,mtermgat, daylgat, dayl_maxgat,nbpveggat, hetroresveggat, autoresveggat, litresveggat,soilcresveggat, burnvegfgat, pstemmassgat, pgleafmassgat,ch4wet1gat, ch4wet2gat, slopefracgat, wetfrac_mongat,wetfdyngat, ch4dyn1gat, ch4dyn2gat, ch4soillsgat,twarmmgat,tcoldmgat,gdd5gat,aridity gat, srplsmongat,defctmongat, anndefctgat,annsrplsgat,annpcpgat,dry↩_season_lengthgat,

9.28.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM)

9.29 ctems1.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM)

Functions/Subroutines

- subroutine **ctems1** (gleafmasrow, bleafmasrow, stemmassrow, rootmassrow, fcancmxrow, zbtwrow, dlzwrow, sdeprow, ailcgrow, ailcbrow, ailcrow, zolncrow, rmatcrow, rmatctemrow, slairow, bmasvegrow, cmasvegcrow, veghghtrow, rootdpthrow, alvsctmrow, alirctmrow, paicrow, slaicrow, ilmos, jlmos, iwmos, jwmos, nml, gleafmasgat, bleafmasgat, stemmassgat, rootmassgat, fcancmxgat, zbtwgat, dlzwgat, sdepgat, ailcggat, ailcbgat, ailcgat, zolncgat, rmatcgat, rmatctemgat, slaigat, bmasveggat, cmasvegcgat, veghghtgat, rootdpthgat, alvsctmgat, alirctmgat, paicgat, slaicgat)

9.29.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM)

9.30 ctems2.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM)

Functions/Subroutines

- subroutine **ctems2** (fcancmxrow, rmatcrow, zolncrow, paicrow,ailcrow,ailcgrow,cmasvegcrow,slaicrow,ailcgsrow,fcancsrow,fcanccanresrow,sdeprow,ch4concrow,sandrow,clayrow,orgmrow,anvegrow,rmlvegrow,tcanoaccrow_m, tbaraccrow↩_m,uvaccrow_m,vvaccrow_m,prbfrhucgrd,extnprobgrd, pfcancmxrow,nfcancmxrow,stemmassrow, rootmassrow, litrmassrow,gleafmasrow,bleafmasrow, soilcmasrow, ailcbrow,flhrlossrow,pandaysrow,lfstatusrow, grwtheffrow,lystmmasrow,lyrotmasrow, tymaxlairow, vgbiomasrow,gavgltmsrow,stmhrlosrow, bmasvegrow,colddaysrow,rothrlosrow,alvsctmrow,alirctmrow,gavglairow,npprow,neprow,hetroresrow, autoresrow,soilresprow,rmrow,rgrow,nbprow,litresrow,socresrow,gpprow,dstcemlsrow,litrfallrow,humiftrsrow, veghghtrow,rootdpthrow,rmlrow,litrfallvegrow, humiftrsvegrow,rmsrow,rmrrow,tltrleafrow,tltrstemrow,tltrrootrow, leaflitrrow, roottemprow,afrleafrow,afrstemrow,afrrootrow,wtstatusrow,ltstatusrow,burnfracrow, smfuncvegrow, lucemcomrow,lucltrinrow,lucsocinrow, nppvegrow,dstcemls3row,farerow,gavgscmsrow, tcanoaccrow↩_out,rmlvegaccrow, rmsvegrow,rmrvegrow,rgvegrow,vgbiomas_vegrow, gppvegrow, nepvegrow, ailcminrow,

ailcmaxrow,fcanrow,pftexistrow,emit_co2row,emit_corow, emit_ch4row,emit_nmhcrow,emit_h2row,emit_↩
noxrow, emit_n2orow,emit_pm25row,emit_tpmrow,emit_tcrow, emit_ocrow,emit_bcrow,btermrow,ltermrow,mtermrow,nbpvegrow
autoresvegrow, litresvegrow,soilcresvegrow, burnvegfrow, pstemmassrow, pgleafmassrow,ch4wet1row,
ch4wet2row, wetfdynrow, ch4dyn1row,ch4dyn2row, ch4soillsrow,twarmmrow,tcoldmrow,gdd5row,aridityrow,
srplsmonrow,defctmonrow, anndefctrow,annsrplsrow,annpcprow,dry_season_lengthrow,

9.30.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM)

9.31 CWCALC.f File Reference

Purpose: Check for freezing or thawing of liquid or frozen water on the vegetation canopy, and adjust canopy temperature and intercepted water stores accordingly.

Functions/Subroutines

- subroutine cwcalc (TCAN, RAICAN, SNOCAN, FRAINC, FSNOWC, CHCAP, HMFC, HTCC, FI, CMASS, ILG, IL1, IL2, JL)

9.31.1 Detailed Description

Purpose: Check for freezing or thawing of liquid or frozen water on the vegetation canopy, and adjust canopy temperature and intercepted water stores accordingly.

9.31.2 Function/Subroutine Documentation

9.31.2.1 subroutine cwcalc (real, dimension (ilg) *TCAN,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension(ilg) *FRAINC,* real, dimension(ilg) *FSNOWC,* real, dimension (ilg) *CHCAP,* real, dimension (ilg) *HMFC,* real, dimension (ilg) *HTCC,* real, dimension (ilg) *FI,* real, dimension (ilg) *CMASS,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

tcan	Temperature of vegetation canopy $[K](T_c)$
raican	Intercepted liquid water stored on the canopy $[kgm^{-2}]$
snocan	Intercepted frozen water stored on the canopy $[kgm^{-2}]$
frainc	Fractional coverage of canopy by liquid water [ ]
fsnowc	Fractional coverage of canopy by frozen water [ ]
chcap	Heat capacity of vegetation canopy $[Jm^{-2}K^{-1}](C_c)$
hmfc	Energy associated with freezing or thawing of water in canopy interception stores $[Wm^{-2}]$
htcc	Internal energy change of canopy due to changes in temperature and/or mass $[Wm^{-2}](I_c)$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
cmass	Mass of vegetation canopy $[kgm^{-2}]$

The change of internal energy I_c of the vegetation canopy as a result of the phase change processes treated here is calculated as the difference in I_c between the beginning and end of the subroutine:

$$\Delta I_c = X_i \Delta[C_c T_c]/\Delta t$$

where C_c represents the volumetric heat capacity of the canopy, T_c its temperature, Δt the length of the time step, and X_i the fractional coverage of the subarea under consideration relative to the modelled area.

If there is liquid water stored on the canopy and the canopy temperature is less than 0 C, the available energy sink HFREZ is calculated from CHCAP and the difference between TCAN and 0 C, and compared with HCONV,

calculated as the energy sink required to freeze all of the liquid water on the canopy. If HFREZ $\leq$ HCONV, the amount of water that can be frozen is calculated using the latent heat of melting. The fractional coverages of frozen and liquid water FSNOWC and FRAINC and their masses SNOCAN and RAICAN are adjusted accordingly, TCAN is set to 0 C, and the amount of energy involved is subtracted from the internal energy HTCC and added to HMFC. Otherwise all of the intercepted liquid water is converted to frozen water, and the energy available for cooling the canopy is calculated as HCOOL = HFREZ – HCONV. This available energy is applied to decreasing the temperature of the canopy, using the specific heat of the canopy elements, and the amount of energy that was involved in the phase change is subtracted from HTCC and added to HMFC.

If there is frozen water stored on the canopy and the canopy temperature is greater than 0 C, the available energy for melting, HMELT, is calculated from CHCAP and the difference between TCAN and 0 C, and compared with HCONV, calculated as the energy required to melt all of the frozen water on the canopy. If HMELT $\leq$ HCONV, the amount of frozen water that can be melted is calculated using the latent heat of melting. The fractional coverages of frozen and liquid water FSNOWC and FRAINC and their masses SNOCAN and RAICAN are adjusted accordingly, TCAN is set to 0 C, and the amount of energy involved is subtracted from HTCC and added to HMFC. Otherwise, all of the intercepted frozen water is converted to liquid water, and the energy available for warming the canopy is calculated as HWARM = HMELT – HCONV. This available energy is applied to increasing the temperature of the canopy, using the specific heats of the canopy elements, and the amount of energy that was involved in the phase change is subtracted from HTCC and added to HMFC.

In the final cleanup, the canopy heat capacity is recomputed and the remaining internal energy calculations are completed.

9.32 disturb.f90 File Reference

Update fractional coverages of pfts to take into account the area burnt by fire. Adjust all pools with new densities in their new areas and increase bare fraction.

Functions/Subroutines

- subroutine, public disturbance_scheme::disturb (stemmass, rootmass, gleafmas, bleafmas,thliq,wiltsm,fieldsm,uwind,vwind,ligh litrmass,prbfrhuc, rmatctem, extnprob, popdon,il1,il2,sort, nol2pfts,grclarea,thice,popdin, lucemcom,dofire,currlat,iday,fsnow,isa

- subroutine, public disturbance_scheme::burntobare (il1, il2, nilg, sort, pvgbioms, pgavltms, pgavscms, fcancmx, burnvegf, stemmass, rootmass, gleafmas, bleafmas, litrmass, soilcmas, pstemmass, pgleafmass, nppveg)

9.32.1 Detailed Description

Update fractional coverages of pfts to take into account the area burnt by fire. Adjust all pools with new densities in their new areas and increase bare fraction.

And while we are doing this also run a small check to make sure grid averaged quantities do not get messed up. Central module for all disturbance scheme-related operations

9.33 DRCOEF.f File Reference

Purpose: CALCULATES DRAG COEFFICIENTS AND RELATED VARIABLES FOR CLASS.

Functions/Subroutines

- subroutine drcoef (CDM, CDH, RIB, CFLUX, QG, QA, ZOMIN, ZOHIN, CRIB, TVIRTG, TVIRTA, VA, FI, ITER, ILG, IL1, IL2)

9.33.1 Detailed Description

Purpose: CALCULATES DRAG COEFFICIENTS AND RELATED VARIABLES FOR CLASS.

9.33.2 Function/Subroutine Documentation

9.33.2.1 subroutine drcoef (real, dimension (ilg) *CDM,* real, dimension (ilg) *CDH,* real, dimension (ilg) *RIB,* real, dimension (ilg) *CFLUX,* real, dimension (ilg) *QG,* real, dimension (ilg) *QA,* real, dimension (ilg) *ZOMIN,* real, dimension (ilg) *ZOHIN,* real, dimension (ilg) *CRIB,* real, dimension (ilg) *TVIRTG,* real, dimension (ilg) *TVIRTA,* real, dimension (ilg) *VA,* real, dimension (ilg) *FI,* integer, dimension(ilg) *ITER,* integer *ILG,* integer *IL1,* integer *IL2*)

Parameters

cdm	STABILITY-DEPENDENT DRAG COEFFICIENT FOR MOMENTUM.
cdh	STABILITY-DEPENDENT DRAG COEFFICIENT FOR HEAT.
rib	BULK RICHARDSON NUMBER.
cflux	CD ∗ MOD(V), BOUNDED BY FREE-CONVECTIVE LIMIT.
zomin	ROUGHNESS HEIGHTS FOR MOMENTUM/HEAT NORMALIZED BY REFERENCE HEI↩ GHT.
zohin	ROUGHNESS HEIGHTS FOR MOMENTUM/HEAT NORMALIZED BY REFERENCE HEI↩ GHT.
crib	-RGAS∗SLTHKEF/(VA∗∗2), WHERE SLTHKEF=-LOG(MAX(SGJ(ILEV),SHJ(ILEV)))
tvirtg	"SURFACE" VIRTUAL TEMPERATURE.
tvirta	LOWEST LEVEL VIRTUAL TEMPERATURE.
va	AMPLITUDE OF LOWEST LEVEL WIND.
fi	FRACTION OF SURFACE TYPE BEING STUDIED.
qg	SATURATION SPECIFIC HUMIDITY AT GROUND TEMPERATURE.
qa	LOWEST LEVEL SPECIFIC HUMIDITY.
iter	INDEX ARRAY INDICATING IF POINT IS UNDERGOING FURTHER ITERATION OR NOT.

ZOM/ZOH: WORK ARRAYS USED FOR SCALING ZOMIN/ZOHIN ON STABLE SIDE, AS PART OF CALCULA↩ TION.

9.34 GATPREP.f File Reference

Purpose: Assign values to pointer vectors relating the location of elements on the "gathered" variable vectors to elements on the original two-dimensional arrays (latitude circle x mosaic tiles) for land grid cells.

Functions/Subroutines

- subroutine gatprep (ILMOS, JLMOS, IWMOS, JWMOS, NML, NMW, GCROW, FAREA, MOSID, NL, NM, ILG, IL1, IL2, IM)

9.34.1 Detailed Description

Purpose: Assign values to pointer vectors relating the location of elements on the "gathered" variable vectors to elements on the original two-dimensional arrays (latitude circle x mosaic tiles) for land grid cells.

9.34.2 Function/Subroutine Documentation

9.34.2.1 subroutine gatprep (integer, dimension (ilg) *ILMOS,* integer, dimension (ilg) *JLMOS,* integer, dimension (ilg) *IWMOS,* integer, dimension (ilg) *JWMOS,* integer *NML,* integer *NMW,* real, dimension (nl) *GCROW,* real, dimension (nl,nm) *FAREA,* integer, dimension (nl,nm) *MOSID,* integer *NL,* integer *NM,* integer *ILG,* integer *IL1,* integer *IL2,* integer *IM*)

Parameters

nml	Total number of mosaic tiles in land surface gather vectors
nmw	Total number of mosaic tiles in inland water gather vectors
nl	Hard-coded maximum number of grid cells
nm	Hard-coded maximum number of mosaic tiles
ilg	Hard-coded maximum number of elements in the gathered vectors
im	Maximum number of mosaic tiles within the grid cells in the array under consideration
ilmos	Index of grid cell corresponding to current element of gathered vector of land surface variables [ ]
jlmos	Index of mosaic tile corresponding to current element of gathered vector of land surface variables [ ]
iwmos	Index of grid cell corresponding to current element of gathered vector of inland water body variables [ ]
jwmos	Index of mosaic tile corresponding to current element of gathered vector of inland water body variables [ ]
gcrow	Real number identifier indicating whether the grid cell is land (-1.0), sea ice (+1.0), or ocean (0.0)
farea	Fractional coverage of mosaic tile on grid cell [ ]
mosid	Mosaic tile type identifier (1 for land, 0 for inland water)

A looping operation is performed over the latitude circle, or array of grid cells, under consideration. If the grid cell is a land one (GCROW = -1.0), an additional internal loop is performed over all the mosaic tiles present. For each mosaic tile, if its fractional coverage is greater than zero, then if the mosaic type identifier MOSID is equal to 1 (indicating land), the counter of total mosaic tiles in the land surface gather vectors, NML, is incremented by one, and the elements of the vectors ILMOS and JLMOS corresponding to NML are set to the indices of the current grid cell and mosaic tile respectively. If MOSID is equal to zero (indicating inland water), the counter of total mosaic tiles in the inland water gather vectors, NMW, is incremented by one, and the elements of the vectors IWMOS and JWMOS corresponding to NMW are set to the indices of the current grid cell and mosaic tile respectively.

9.35 GAUSSG.f File Reference

THIS ROUTINE CALCULATES THE ROOTS (F) OF THE ORDINARY LEGENDRE POLYNOMIALS OF ORDER NZERO. THE FIRST STEP IS TO MAKE AN INITIAL GUESS FOR EACH ROOT AND THEN TO USE THE O↩ RDINARY LEGENDRE ALGORITHM (ORDLEG) AND NEWTONS METHOD TO REFINE THE SOLUTION UNTIL THE CRITERION XLIM IS SATISFIED.

Functions/Subroutines

- subroutine **gaussg** (NZERO, F, WT, SIA, RAD, WOCS)

9.35.1 Detailed Description

THIS ROUTINE CALCULATES THE ROOTS (F) OF THE ORDINARY LEGENDRE POLYNOMIALS OF ORDER NZERO. THE FIRST STEP IS TO MAKE AN INITIAL GUESS FOR EACH ROOT AND THEN TO USE THE O↩ RDINARY LEGENDRE ALGORITHM (ORDLEG) AND NEWTONS METHOD TO REFINE THE SOLUTION UNTIL THE CRITERION XLIM IS SATISFIED.

F = COSINE OF COLATITUDE

WT = CORRESPONDING GAUSSIAN WEIGHT

SIA = SINE OF COLATITUDE

RAD = COLATITUDE IN RADIANS

WOCS = GAUSSIAN WEIGHT / COS(COLAT)∗∗2

9.36 GRALB.f File Reference

Purpose: Calculate visible and near-IR ground albedos.

Functions/Subroutines

- subroutine gralb (ALVSG, ALIRG, ALVSGC, ALIRGC, ALGWV, ALGWN, ALGDV, ALGDN, ALGWET, ALG↩ DRY, THLIQ, FSNOW, ALVSU, ALIRU, FCMXU, AGVDAT, AGIDAT, FG, ISAND, ILG, IG, IL1, IL2, JL, IALG, IGRALB)

9.36.1 Detailed Description

Purpose: Calculate visible and near-IR ground albedos.

9.36.2 Function/Subroutine Documentation

9.36.2.1 subroutine gralb (real, dimension (ilg) *ALVSG,* real, dimension (ilg) *ALIRG,* real, dimension (ilg) *ALVSGC,* real, dimension (ilg) *ALIRGC,* real, dimension (ilg) *ALGWV,* real, dimension (ilg) *ALGWN,* real, dimension (ilg) *ALGDV,* real, dimension (ilg) *ALGDN,* real, dimension(ilg) *ALGWET,* real, dimension(ilg) *ALGDRY,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg) *FSNOW,* real, dimension (ilg) *ALVSU,* real, dimension (ilg) *ALIRU,* real, dimension (ilg) *FCMXU,* real, dimension(ilg) *AGVDAT,* real, dimension(ilg) *AGIDAT,* real, dimension (ilg) *FG,* integer, dimension (ilg,ig) *ISAND,* integer *ILG,* integer *IG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IALG,* integer *IGRALB*)

Parameters

alvsg	Visible albedo of bare ground $[](\alpha_{g,VIS})$
alirg	Near-IR albedo of bare ground $[](\alpha_{g,NIR})$
alvsgc	Visible albedo of ground under vegetation canopy [ ]
alirgc	Near-IR albedo of ground under vegetation canopy [ ]
algwet	All-wave albedo of wet soil for modelled area $[](\alpha_{g,wet})$
algdry	All-wave albedo of dry soil for modelled area $[](\alpha_{g,dry})$
thliq	Volumetric liquid water content of soil layers $[m^3m^{-3}]$
alvsu	Visible albedo of urban part of modelled area $[](alpha_{u,VIS})$
aliru	Near-IR albedo of urban part of modelled area $[](alpha_{u,NIR})$
fcmxu	Fractional coverage of urban part of modelled area $[](X_u)$
agvdat	Assigned value of visible albedo of ground – optional [ ]
agidat	Assigned value of near-IR albedo of ground – optional [ ]
isand	Soil type flag based on sand content, assigned in subroutine CLASSB

If the ISAND flag for the surface soil layer is greater than zero (indicating mineral soil), the visible and near-IR open ground albedos, $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$, are calculated on the basis of the wet and dry ground albedos $\alpha_{g,wet}$ and $\alpha_{g,dry}$ which were assigned for the modelled area in CLASSB. Idso et al. (1975) found a correlation between the soil liquid moisture content in the top 10 cm of soil (represented in CLASS by that of the first soil layer, $\theta_{1,1}$) and the total surface albedo $\alpha_{g,T}$: for liquid water contents less than 0.22 m^3m^{-3}, $\alpha_{g,T}$ took on the value of $\alpha_{g,dry}$; for liquid water contents greater than 0.26 m^3m^{-3}, $\alpha_{g,T}$ took on the value of $\alpha_{g,wet}$. For values of $\theta_{1,1}$ between these two limits, a linear relationship is assumed:

$$[\alpha_{g,T} - \alpha_{g,dry}]/[\theta_{l,1} - 0.22] = [\alpha_{g,wet} - \alpha_{g,dry}]/[0.26 - 0.22]$$

Thus, in GRALB $\alpha_{g,T}$ is calculated as follows:

$$\alpha_{g,T} = \alpha_{g,dry}\ \theta_{l,1} \leq 0.22$$
$$\alpha_{g,T} = \theta_{l,1}[\alpha_{g,wet} - \alpha_{g,dry}]/0.04 - 5.50[\alpha_{g,wet} - \alpha_{g,dry}] + \alpha_{g,dry}\ 0.22 < \theta_{l,1} < 0.26$$
$$\alpha_{g,T} = \alpha g, wet\ 0.26 \leq \theta_{l,1}$$

The total albedo is partitioned into values for the visible and near-IR albedo by making use of the observation that in the case of mineral soils, the near-IR albedo is typically twice that of the visible albedo (e.g. Dickinson, 1983). Since

the partitioning of incoming shortwave radiation into visible and near-IR can be approximated as 50:50 on average, $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$ can be calculated as

$\alpha_{g,VIS} = 2.0\alpha_{g,T}/3.0$
$\alpha_{g,NIR} = 2.0\alpha_{g,VIS}$

Finally, a correction is applied to $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$ in order to account for the possible presence of urban areas over the modelled area. Visible and near-IR albedos are assigned for local urban areas, $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$, as part of the background data (see the section on "Data Requirements"). A weighted average is calculated from the fractional urban area X_u as:

$\alpha_{g,VIS} = X_u\alpha_{u,VIS} + [1.0 - X_u]\alpha_{g,VIS}$
$\alpha_{g,NIR} = X_u\alpha_{u,NIR} + [1.0 - X_u]\alpha_{g,NIR}$

If the soil on the modelled area is not mineral, i.e. if the ISAND flag is less than zero, $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$ are determined as follows:

If ISAND = -2, indicating organic soil, $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$ are assigned values of 0.05 and 0.30 respectively from the lookup tables in the block data subroutine CLASSBD, corresponding to average measured values reported in Comer et al. (2000).

If ISAND = -3, indicating rock at the surface, $\alpha_{g,T}$ is given a value of 0.27 from CLASSBD, based on typical literature values (e.g. Sellers, 1974), and this is partitioned into values of $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$ as above.

If ISAND = -4, indicating continental ice sheet or glacier, $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$ are assigned values of 0.95 and 073 from CLASSBD, reflecting values reported for Antarctica (e.g. Sellers, 1974).

The above calculations are all performed if the flag IALG is set to zero. If IALG is set to one, indicating that assigned ground albedos are to be used instead of calculated values, $\alpha_{g,VIS}$ and $\alpha_{g,NIR}$ are set to the assigned values AGVDAT and AGIDAT respectively.

Lastly, the ground values of visible and near-IR albedo under the vegetation canopy are currently set equal to the open values (this approach is currently under review).

9.37 GRDRAN.f File Reference

Purpose: Quantify movement of liquid water between soil layers under non-infiltrating conditions, in response to gravity and tension forces.

Functions/Subroutines

- subroutine grdran (IVEG, THLIQ, THICE, TBARW, FDT, TFDT, BASFLW, TBASFL, RUNOFF, TRUNOF, QFG, WLOST, FI, EVAP, R, ZPOND, DT, WEXCES, THLMAX, THTEST, THPOR, THLRET, THLMIN, BI, PSISAT, GRKSAT, THFC, DELZW, XDRAIN, ISAND, LZF, IGRN, IGRD, IGDR, IG, IGP1, IGP2, ILG, IL1, IL2, JL, N)

9.37.1 Detailed Description

Purpose: Quantify movement of liquid water between soil layers under non-infiltrating conditions, in response to gravity and tension forces.

9.37.2 Function/Subroutine Documentation

9.37.2.1 subroutine grdran (integer *IVEG,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *TBARW,* real, dimension (ilg,igp1) *FDT,* real, dimension (ilg,igp1) *TFDT,* real, dimension(ilg) *BASFLW,* real, dimension (ilg) *TBASFL,* real, dimension(ilg) *RUNOFF,* real, dimension (ilg) *TRUNOF,* real, dimension (ilg) *QFG,* real, dimension (ilg) *WLOST,* real, dimension (ilg) *FI,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *R,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *DT,* real, dimension(ilg) *WEXCES,* real, dimension(ilg,ig) *THLMAX,* real, dimension(ilg,ig) *THTEST,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLRET,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *BI,* real, dimension(ilg,ig) *PSISAT,* real, dimension(ilg,ig) *GRKSAT,* real, dimension (ilg,ig) *THFC,* real, dimension (ilg,ig) *DELZW,* real, dimension(ilg) *XDRAIN,* integer, dimension (ilg,ig) *ISAND,* integer, dimension (ilg) *LZF,* integer, dimension (ilg) *IGRN,* integer, dimension (ilg) *IGRD,* integer, dimension (ilg) *IGDR,* integer *IG,* integer *IGP1,* integer *IGP2,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

iveg	Subarea type flag
thliq	Volumetric liquid water content of soil layer $[m^3m^{-3}](\theta_l)$
thice	Volumetric frozen water content of soil layer $[m^3m^{-3}](\theta_i)$
tbarw	Temperature of water in soil layer [C]
fdt	Water flow at soil layer interfaces during current time step [m]
tfdt	Temperature of water flowing between soil layers [C]
basflw	Base flow from bottom of soil column [m]
tbasfl	Temperature of base flow from bottom of soil column [K]
runoff	Total runoff from soil column [m]
trunof	Temperature of total runoff from soil column [K]
qfg	Evaporation from soil surface (diagnostic) $[kgm^{-2}s^{-1}]$
wlost	Residual amount of water that cannot be supplied by surface stores $[kgm^{-2}]$
fi	Fractional coverage of subarea in question on modelled area [ ] (X_i)
evap	Evaporation rate from ground surface $[ms^{-1}]$
r	Rainfall rate at ground surface $[ms^{-1}]$
zpond	Depth of ponded water on soil surface [m]
dt	Time period over which water movement takes place [s]
igrn	Flag to indicate whether calculations in subroutine GRINFL are done
lzf	Index of soil layer in which wetting front is located
igrd	Flag to indicate whether calculations in this subroutine are to be done
igdr	Index of soil layer in which bedrock is encountered
thpor	Pore volume in soil layer $[m^3m^{-3}](\theta_p)$
thlret	Liquid water retention capacity for organic soil $[m^3m^{-3}](\theta_{l,ret})$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}](\theta_{l,min})$
bi	Clapp and Hornberger empirical "b" parameter [ ] (b)
psisat	Soil moisture suction at saturation $[m](\Psi_{sat})$
grksat	Hydraulic conductivity of soil at saturation $[ms^{-1}](K_{sat})$
thfc	Field capacity $[m^3m^{-3}]$
delzw	Permeable depth of soil layer $[m](\Delta_{zg,w})$
xdrain	Drainage index for water flow at bottom of soil profile [ ]
isand	Sand content flag

In loop 50, the flag IGRD is first set to 1 for all grid cells where the calculations in this subroutine are to be performed. The necessary conditions are: that the surface being modelled is not a glacier or ice sheet (ISAND $>$ -4); that the time period DT is greater than zero; that the infiltration calculations in subroutine GRINFL are not simultaneously being performed (IGRN = 0); and that the rainfall rate and the depth of ponded water are both vanishingly small. If any of these conditions is not met, IGRD is set to zero.

In loop 100, if the surface being modelled is not an ice sheet (ISAND = -4) and if the soil layer in question is not completely rock (ISAND = -3), the maximum possible water content of each soil layer, THLMAX, is calculated as the maximum of the available pore volume THPOR – THICE (where THPOR is the total pore volume and THICE is the ice content of the layer), the actual liquid water content THLIQ, and the minimum residual liquid water content THLMIN. The last two conditions are required because in the case of a saturated soil undergoing freezing, since water expands when frozen, the sum of the liquid and frozen volumetric water contents may be greater than the pore volume, and thus THLIQ or THLMIN may be greater than THPOR – THICE. An effective saturated hydraulic conductivity GRKSATF of the soil layer is also defined, applying an empirical correction for the presence of ice. This ice content factor, fice, is calculated from Zhao and Gray (1997) as:

$$f_{ice} = [1.0^{\smile} min((\theta_p - \theta_{l,min})/\theta_p, \theta_i/\theta_p)]^2$$

The pore volume of the soil layer is corrected for the presence of ice by defining the effective porevolume THPORF as equivalent to THLMAX.

In loops 150 and 200, the theoretical amounts of water FDT flowing across the soil layer boundaries are calculated. FDT is evaluated as the product of the flow rate F(z) at the given depth z, and the period of time (DT) over which the flow is occurring. At z=0, the water flux is simply equal to the soil surface evaporation rate. Within the soil, the flow rate F at a given depth z is obtained using equation 21 from Verseghy (1991):

$$F(z) = K(z)[-b\Psi(z)/\theta_l(z) \bullet d\theta_l/dz + 1]$$

where K(z) is the hydraulic conductivity and psi(z) is the soil moisture suction at depth z, and b is an empirical parameter developed by Clapp and Hornberger (1978). K(z) and $\Psi(z)$ are calculated following Clapp and Hornberger as

$K(z) = K_{sat}(\theta_l/\theta_p)^{(2b+3)}$ $\Psi(z) = \Psi_{sat}(\theta_l/\theta_p)^{(-b)}$

where K_{sat} and Ψ_{sat} are the values of K and Ψ respectively at saturation.

At the bottom of the permeable soil depth, in layer IGDR, if the liquid water content of the soil is greater than the field capacity, the vertical flow out of the bottom of the soil profile is calculated using a relation derived from Soulis et al. (2010):

Between soil layers, values of $K(z)$ and $\Psi(z)$ must be determined. This requires the estimation of soil properties at the layer interfaces. For θ_l, θ_p and $d(\theta_l)/dz$, slightly different approaches are followed if the permeable soil layer thickness DELZW increases with depth (the normal case), or if it decreases between one soil layer and the next (indicating the presence of an impermeable barrier at some depth in the second layer). In the first case, the pore volume THPBND, liquid water content THLBND, and liquid water gradient DTHLDZ are simply calculated as arithmetic averages of the values above and below the interface. This has the effect of weighting the interface values towards the upper layer values, roughly emulating a surfaceward exponential decay curve of liquid water content. In the second case, in order to avoid a spurious weighting towards the lower layer, the gradients of liquid water content and pore volume between the upper and lower layers are calculated as linear relations, and then solved for the values at the interface.

The Clapp and Hornberger b parameter at the interface is calculated as a simple average of the values in the upper and lower soil layers. The saturated hydraulic conductivity $K_{sat,bnd}$ is calculated as a harmonic mean, and the saturated soil moisture suction $\Psi_{sat,bnd}$ as a geometric mean, of the top and bottom layer values:

$K_{sat,bnd} = K_{sat,t}K_{sat,b}(\Delta z_{g,w,,t} + \Delta z_{g,w,,b})/(K_{sat,t}\Delta z_{g,w,b} + K_{sat,b}\Delta z_{g,w,t})$

$\Psi_{sat,bnd} = \Psi_{sat,t}^{\Delta zg,w,t/(\Delta zg,w,t+\Delta zg,w,b)}\Psi_{sat,b}^{\Delta zg,w,b/(\Delta zg,w,t+\Delta zg,w,b)}$

Finally, K(z), $\Psi(z)$ and F(z) at the interface are calculated from the equations given above. At the end of the loop, if set to zero behind the soil layer LZF containing the wetting front, and the flow rate at the bottom of LZF is constrained to be $\geq$ 0.

In the next several loops, checks are carried out to ascertain whether the theoretically determined flow rates at the soil layer interfaces can be supported by the water contents in the layers. At the beginning of loop 250, the soil surface evaporation rate is addressed. A trial liquid water content THTEST is calculated for the first soil layer, reflecting the removal of the evaporated water. If THTEST is less than the minimum soil water content, F(0) is set to zero and θ_l is set to $\theta_{l,min}$. The excess surface flux that was not met by the removed liquid water is converted to a frozen water content, and an attempt is made to remove it from the frozen water in the layer. (The energy involved in converting the required frozen water mass to liquid water is obtained by decreasing the water temperature of the layer.) The frozen water content of the layer is adjusted to reflect the removal of the required amount. If the demand exceeds the supply of frozen water available, the frozen water content is set to zero, the diagnostic evaporative flux QFG at the soil surface is adjusted, and the remaining water flux not able to be met by the first soil layer is assigned to the variable WLOST.

In the remainder of the 250 loop, checks are carried out for soil layers with liquid water contents already effectively equal to $\theta_{l,min}$. If the flux at the top of the layer is upward and that at the bottom of the layer is downward, both are set to zero. If both are downward, only the flux at the bottom is set to zero; if both are upward, only the flux at the top is set to zero. If the soil layer is an organic one and the liquid water content is less than the layer retention capacity THLRET and the flux at the bottom of the layer is downward, it is set to zero.

In the 300 loop, checks are carried out for soil layers with liquid water contents already effectively equal to THLMAX. If the flux at the top of the layer is downward and that at the bottom of the layer is upward, both are set to zero. If both are downward, then if the top flux is greater than the bottom flux, it is set equal to the bottom flux. If both are upward, then if magnitude of the bottom flux is greater than that of the top flux, it is set equal to the top flux.

In the 400 loop, for each soil layer THTEST is recalculated as the liquid water content resulting from the updated boundary fluxes, and is compared with a residual value THLTHR. For organic soil layers deeper than the first layer, THLTHR is set to the minimum of $\theta_{l,ret}$ and θ_l, since only evapotranspiration can cause the soil moisture to fall below the retention capacity. (The first layer is excepted because surface evaporation can drive its moisture content below $\theta_{l,ret}$.) For mineral soils, $\theta_{l,min}$ is set to THLMIN. If THTEST $<$ THLTHR, then if the flow at the bottom of the soil layer is downward, it is recalculated as the sum of the flow at the top of the layer plus the amount of water

that must be removed to make the liquid water content of the layer equal to THLTHR. If the flow at the bottom of the layer is upward, the flow at the top of the layer is recalculated as the sum of the flow at the bottom of the layer minus the amount of water that must be removed to make the liquid water content of the layer equal to THLTHR. THTEST of the current layer is then reset to THLTHR, and THTEST of the overlying and underlying layers (if any) are recalculated using the new interface fluxes.

In the 500 loop, for each soil layer THTEST is compared with THLMAX. If THTEST $>$ THLMAX, two temporary variables are defined: WLIMIT as the amount of water required to raise the liquid water content of the soil layer to THLMAX, and WEXCES as the amount of water representing the excess of THTEST over THLMAX. If the flux at the top of the layer is downward and the flux at the bottom of the layer is upward, then if the negative of the flux at the bottom of the layer is greater than WLIMIT, it is set equal to -WLIMIT and the flux at the top is set to zero; otherwise WEXCES is subtracted from the flux at the top. If both the flux at the top and the flux at the bottom are downward, then WEXCES is subtracted from the flux at the top. If both are upward, then WEXCES is added to the flux at the bottom, and a correction is applied to the flux at the bottom of the underlying layer. Finally, THTEST is recalculated for all the soil layers using the new interface fluxes.

In the first part of the 600 loop, a correction is performed in the unlikely event that as a result of the above adjustments, the flux at the bottom of the last permeable soil layer has become upward. If this occurs, all of the fluxes above are corrected by the inverse of this same amount. However, this will result in a small spurious upward water flux at the surface. Since the liquid water flows are now accounted for, an attempt is made, as in loop 250, to remove the required water from the frozen water store in the first layer. The excess that cannot be supplied from this source is assigned to WLOST.

In the second part of the 600 loop, the temperatures TFDT of the water fluxes at the top and bottom of the layers in the permeable soil profile are set equal to the temperatures of the water in the first and last layers respectively. The flow at the bottom of the profile and the temperature of the flow are used to update BASFLW and TBASFL, the baseflow from the current subarea and its temperature, and RUNOFF and TRUNOF, the total runoff from the grid cell in question and its temperature.

In the 700 loop, for each successive soil layer the temperature of the water flux at the bottom of the layer is first assigned. If the flux is downward, TFDT is set to the temperature of the water in the current layer; if it is upward, TFDT is set to the temperature of the water in the layer below. The temperature of the water in the current layer is then updated using the FDT and TFDT values at the top and bottom of the layer, and the liquid water content of the layer is set to THTEST.

9.38 GRINFL.f File Reference

Purpose: Quantify movement of liquid water between soil layers under conditions of infiltration.

Functions/Subroutines

- subroutine grinfl (IVEG, THLIQ, THICE, TBARW, BASFLW, TBASFL, RUNOFF, TRUNOF, ZFAV, LZFAV, T↩ HLINV, QFG, WLOST, FI, EVAP, R, TR, TPOND, ZPOND, DT, ZMAT, WMOVE, TMOVE, THLIQX, THICEX, TBARWX, DELZX, ZBOTX, FDT, TFDT, PSIF, THLINF, GRKINF, THLMAX, THTEST, ZRMDR, FDUMMY, T↩ DUMMY, THLDUM, THIDUM, TDUMW, TRMDR, ZF, FMAX, TUSED, RDUMMY, ZERO, WEXCES, FDTBND, WADD, TADD, WADJ, TIMPND, DZF, DTFLOW, THLNLZ, THLQLZ, DZDISP, WDISP, WABS, THPOR, T↩ HLRET, THLMIN, BI, PSISAT, GRKSAT, THLRAT, THFC, DELZW, ZBOTW, XDRAIN, DELZ, ISAND, IGRN, IGRD, IFILL, IZERO, LZF, NINF, IFIND, ITER, NEND, ISIMP, IGDR, IG, IGP1, IGP2, ILG, IL1, IL2, JL, N)

9.38.1 Detailed Description

Purpose: Quantify movement of liquid water between soil layers under conditions of infiltration.

9.38.2 Function/Subroutine Documentation

9.38.2.1 subroutine grinfl (integer *IVEG,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *TBARW,* real, dimension(ilg) *BASFLW,* real, dimension(ilg) *TBASFL,* real, dimension(ilg) *RUNOFF,* real, dimension(ilg) *TRUNOF,* real, dimension (ilg) *ZFAV,* integer, dimension (ilg) *LZFAV,* real, dimension(ilg) *THLINV,* real, dimension (ilg) *QFG,* real, dimension (ilg) *WLOST,* real, dimension (ilg) *FI,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg) *TPOND,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *DT,* real, dimension (ilg,igp2,igp1) *ZMAT,* real, dimension (ilg,igp2) *WMOVE,* real, dimension (ilg,igp2) *TMOVE,* real, dimension(ilg,igp1) *THLIQX,* real, dimension(ilg,igp1) *THICEX,* real, dimension(ilg,igp1) *TBARWX,* real, dimension (ilg,igp1) *DELZX,* real, dimension (ilg,igp1) *ZBOTX,* real, dimension (ilg,igp1) *FDT,* real, dimension (ilg,igp1) *TFDT,* real, dimension (ilg,igp1) *PSIF,* real, dimension(ilg,igp1) *THLINF,* real, dimension(ilg,igp1) *GRKINF,* real, dimension(ilg,ig) *THLMAX,* real, dimension(ilg,ig) *THTEST,* real, dimension (ilg,igp1) *ZRMDR,* real, dimension(ilg,igp1) *FDUMMY,* real, dimension(ilg,igp1) *TDUMMY,* real, dimension(ilg,ig) *THLDUM,* real, dimension(ilg,ig) *THIDUM,* real, dimension (ilg,ig) *TDUMW,* real, dimension (ilg) *TRMDR,* real, dimension (ilg) *ZF,* real, dimension (ilg) *FMAX,* real, dimension (ilg) *TUSED,* real, dimension(ilg) *RDUMMY,* real, dimension (ilg) *ZERO,* real, dimension(ilg) *WEXCES,* real, dimension(ilg) *FDTBND,* real, dimension (ilg) *WADD,* real, dimension (ilg) *TADD,* real, dimension (ilg) *WADJ,* real, dimension(ilg) *TIMPND,* real, dimension (ilg) *DZF,* real, dimension(ilg) *DTFLOW,* real, dimension(ilg) *THLNLZ,* real, dimension(ilg) *THLQLZ,* real, dimension(ilg) *DZDISP,* real, dimension (ilg) *WDISP,* real, dimension (ilg) *WABS,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLRET,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *BI,* real, dimension(ilg,ig) *PSISAT,* real, dimension(ilg,ig) *GRKSAT,* real, dimension(ilg,ig) *THLRAT,* real, dimension (ilg,ig) *THFC,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *ZBOTW,* real, dimension(ilg) *XDRAIN,* real, dimension(ig) *DELZ,* integer, dimension (ilg,ig) *ISAND,* integer, dimension (ilg) *IGRN,* integer, dimension (ilg) *IGRD,* integer, dimension (ilg) *IFILL,* integer, dimension (ilg) *IZERO,* integer, dimension (ilg) *LZF,* integer, dimension (ilg) *NINF,* integer, dimension (ilg) *IFIND,* integer, dimension (ilg) *ITER,* integer, dimension (ilg) *NEND,* integer, dimension (ilg) *ISIMP,* integer, dimension (ilg) *IGDR,* integer *IG,* integer *IGP1,* integer *IGP2,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

iveg	Subarea type flag
thliq	Volumetric liquid water content of soil layer $[m^3m^{-3}](\theta_l)$
thice	Volumetric frozen water content of soil layer $[m^3m^{-3}](\theta_i)$
tbarw	Temperature of water in soil layer [C]
basflw	Base flow from bottom of soil column [m]
tbasfl	Temperature of base flow from bottom of soil column [K]
runoff	Total runoff from soil column [m]
trunof	Temperature of total runoff from soil column [K]
qfg	Evaporation from soil surface (diagnostic) $[kgm^{-2}s^{-1}]$
wlost	Residual amount of water that cannot be supplied by surface stores $[kgm^{-2}]$
zfav	Average depth of wetting front over current time step [m]
thlinv	Liquid water content behind the wetting front $[m^3m^{-3}]$
lzfav	Soil layer index in which the average position of the wetting front occurs
fi	Fractional coverage of subarea in question on modelled area [ ]
evap	Evaporation rate from ground surface $[ms^{-1}]$
r	Rainfall rate at ground surface $[ms^{-1}]$
tr	Temperature of rainfall [C]
tpond	Temperature of ponded water [C]
zpond	Depth of ponded water on soil surface [m]
dt	Time period over which water movement takes place [s]
thpor	Pore volume in soil layer $[m^3m^{-3}](\theta_p)$
thlret	Liquid water retention capacity for organic soil $[m^3m^{-3}](\theta_{l,ret})$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}](theta_{l,min})$
bi	Clapp and Hornberger empirical "b" parameter [ ] (b)
psisat	Soil moisture suction at saturation $[m](\Psi_{sat})$
grksat	Hydraulic conductivity of soil at saturation $[ms^{-1}](K_{sat})$
thlrat	Fractional saturation of soil behind the wetting front $[](f_{inf})$
thfc	Field capacity $[m^3m^{-3}]$
delzw	Permeable depth of soil layer $[m](\Delta z_{g,w})$
zbotw	Depth to permeable bottom of soil layer [m]
xdrain	Drainage index for water flow at bottom of soil profile [ ]
delz	Overall thickness of soil layer [m]
isand	Sand content flag
igrn	Flag to indicate whether calculations in this subroutine are to be done
igrd	Flag to indicate whether calculations in subroutine GRDRAN are to be done
igdr	Index of soil layer in which bedrock is encountered

In loop 50, the flag IGRN is first set to 1 for all grid cells where the calculations in this subroutine are to be performed. The necessary conditions are: that the surface being modelled is not a glacier or ice sheet (ISAND $>$ -4); that the time period DT is greater than zero; and that either the rainfall rate is greater than zero or that ponded water exists on the surface. If any of these conditions is not met, IGRN is set to zero.

In loop 100, a series of local arrays THLIQX, THICEX, TBARWX, DELZX and ZBOTX are defined for use in the subsequent infiltration calculations, with a “y” dimension of IG+1, where IG is the number of soil layers. The entries from 1 to IG are set equal to the corresponding values in THLIQ, THICE, TBARW, DELZW and ZBOTW. An effective saturated hydraulic conductivity GRKSATF is calculated for each soil layer, applying an empirical correction for the presence of ice. This ice content factor, f_{ice}, is calculated from Zhao and Gray (1997) as:

$f_{ice} = [1.0^{\vee} min(1.0, \theta_i/\theta_p)]^2$

To further account for the presence of ice, a modified pore volume THPORF for the soil layer is calculated as the maximum of the available pore volume θ_p – θ_i (where θ_p is the total pore volume and θ_i is the ice content of the layer), the actual liquid water content θ_l, and the minimum residual liquid water content $\theta_{l,min}$. (The last two conditions are required because in the case of a saturated soil undergoing freezing, since water expands when frozen, the sum of the liquid and frozen volumetric water contents may be greater than the pore volume, and thus θ_l or $\theta_{l,min}$ may be greater than θ_p – θ_i.) Finally, the water content THLINF and the hydraulic conductivity GRKINF behind the wetting front are evaluated, following the analysis of Green and Ampt (1911) and treating the change

in soil moisture due to infiltration as a downward-propagating square wave. THLINF is calculated as the maximum of $f_{inf}(\theta_p \breve{} \theta_i)$, θ_l, and $\theta_{l,min}$, where f_{inf} represents the fractional saturation of the soil behind the wetting front, corresponding to a hydraulic conductivity of half the saturated value (this correction is applied in order to account for the fact that as infiltration occurs, a small amount of air is usually trapped in the soil). GRKINF is calculated from GRKSATF, THLINF and THPORF, using the classic Clapp and Hornberger (1978) equation

$K(z) = K_{sat}(\theta_l/\theta_p)^{(2b+3)}$

where K(z) is the hydraulic conductivity at a depth z, K_{sat} is the hydraulic conductivity at saturation, θ_p is the pore volume and b is an empirical coefficient.

In loop 150, values for the IG+1 entries in each of the above arrays are assigned. If the permeable thickness DELZW of the IG layer is less than its overall thickness DELZ (indicating the presence of bedrock) the IG+1 entries are set to 0; otherwise they are set to the values for the IG layer in the case of THLIQX, THICEX, TBARWX and THLINF, and for DELZX and ZBOTX they are set to large positive numbers. GRKINF is set to the value for the IG layer multiplied by the drainage parameter XDRAIN, which takes a value of 0 if the soil is underlain by an impermeable layer (as in a bog), and a value of 1 otherwise.

In loop 200, the soil water suction across the wetting front psi_f is calculated for each soil layer, using equation 25 in Verseghy (1991):

$\Psi_f = b[\Psi_{inf}K_{inf} - \Psi(z)K(z)]/[K_{inf}(b+3)]$

where Ψ_{inf} and K_{inf} are the soil moisture suction and the hydraulic conductivity behind the wetting front, and $\Psi(z)$ and $K(z)$ are the soil moisture suction and the hydraulic conductivity ahead of the wetting front. The soil moisture suction values are obtained using the classic Clapp and Hornberger (1978) equation:

$\Psi(z) = \Psi_{sat}(\theta_l/\theta_p)^{(-b)}$

where Ψ_{sat} is the soil moisture suction at saturation.

In loop 400, a test is carried out to determine whether saturated conditions are already present in the soil. Generally it is assumed that a period of unsaturated flow occurs initially, so the flag IFILL is set to 1, the depth of the wetting front ZF is set to zero, and the flag LZF indicating the index of the soil layer in which the wetting front occurs is set to 1. If a pond is present on the soil surface or if the hydraulic conductivity if the first soil layer is very small, it is deemed that saturated flow begins immediately, so IFILL is set to zero. Then each soil layer is checked in turn, to ascertain whether the liquid water content is greater than THLINF. If so, it is concluded that saturated flow is occurring in this layer; ZF is set to ZBOTW, the depth of the bottom of the soil layer, LZF is set to the index of the next layer, and IFILL is set to zero. For each layer found to be undergoing saturated flow, its water content is stored in the J+1 level of the water movement matrix WMOVE, and its water temperature is stored in the J+1 level of the matrix TMOVE. The counter NINF is set to the number of soil layers behind and including the one with the wetting front, plus 2.

A series of subroutines is called to complete the infiltration calculations. Subroutine WFILL treats the period of unsaturated flow up the point when saturated flow begins. Subroutine WFLOW treats the period of saturated flow. Subroutine WEND reassigns liquid water contents and temperatures at the conclusion of the infiltration period. T↩
HLIQ, THICE and TBARW are then set to the corresponding updated values of THLIQX, THICEX and TBARWX. The average depth of the wetting front in the soil layer in which it occurs, the index of the soil layer, and the moisture content behind the wetting front are stored in output variables ZFAV, LZFAV and THLINV respectively. Finally, if there is any time remaining in the current time step following the infiltration period, subroutine GRDRAN is called to treat the movement of soil water over the remainder of the time step.

9.39 hetres_mod.f90 File Reference

Heterotrophic Respiration Subroutine For Vegetated Fraction.

Functions/Subroutines

- subroutine, public heterotrophic_respiration::hetresg (litrmass, soilcmas, delzw,thpor,il1,il2,tbar,psisat, b,thliq,zbotw,thiceg,frac,isnow,isand,

- subroutine, public heterotrophic_respiration::hetresv (fcan,fct, litrmass, soilcmas,delzw,thpor, il1,il2,tbar,psisat,

b, thliq,roottemp,zbotw,sort,isand, thicec,

9.39.1 Detailed Description

Heterotrophic Respiration Subroutine For Vegetated Fraction.

Central module for all heterotrophic respiration-related operations

9.40 hetresg_old.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Heterotrophic Respiration Subroutine For Bare Fraction.

Functions/Subroutines

- subroutine hetresg (litrmass, soilcmas,il1,il2,tbar,thliq,sand,clay,zbotw,frac,isnow,isand,

9.40.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Heterotrophic Respiration Subroutine For Bare Fraction.

Heterotrophic respiration, $R_{\rm h}$ ($mol\,CO_2\,m^{-2}\,s^{-1}$), in CTEM is based on respiration from the litter (which includes contributions from the stem, leaf and root components), $R_{h,D}$, and soil carbon, $R_{h,H}$, pools,

$$R_{\rm h} = R_{h,D} + R_{h,H}.$$

Heterotrophic respiration is regulated by soil temperature and moisture and is calculated on a daily time step. The original heterotrophic respiration scheme is described in [7] while the modified parametrization used in CTEM v. 2.0 is detailed in [37] and is briefly described here. Respiration from the litter and soil carbon pools takes the following basic form

$$R_{{\rm h},i} = 2.64 \times 10^{-6}\,\varsigma_i C_i f_{15}(Q_{10}) f(\Psi)_i, i = {\rm D, H}.$$

The soil carbon and litter respiration depends on the amount of carbon in these components ($C_{\rm H}$ and $C_{\rm D}$; $kg\,C\,m^{-2}$) and on a PFT-dependent respiration rate specified at $15\,C$ ($\varsigma_{\rm H}$ and $\varsigma_{\rm D}$; $kg\,C\,(kg\,C)^{-1}\,yr^{-1}$; see also ctem_params.f90). The constant 2.64×10^{-6} converts units from $kg\,C\,m^{-2}\,yr^{-1}$ to $mol\,CO_2\,m^{-2}\,s^{-1}$.

The effect of soil moisture is accounted for via dependence on soil matric potential ($f(\Psi)$), described later. The temperature dependency of microbial soil respiration rates has been estimated by several different formulations, ranging from simple Q_{10} (exponential) to Arrhenius-type formulations (see review by [34]). In CTEM, soil temperature influences heterotrophic respiration through a temperature-dependent Q_{10} function ($f_{15}(Q_{10})$). The value of Q_{10} itself is assumed to be a function of temperature following a hyperbolic tan function:

$$Q_{10} = 1.44 + 0.56\,\tanh[0.075(46.0 - T_i)], i = {\rm D, H},$$

where $T_{\{D,H\}}$ is the temperature of either the litter or soil carbon pool (C), respectively. The parametrization is a compromise between the temperature-independent Q_{10} commonly found in many terrestrial ecosystem models [18] and the temperature-dependent Q_{10} of [26]. While a constant Q_{10} yields an indefinitely increasing respiration rate with increasing temperature, the formulation of [26] gives a continuously increasing Q_{10} under decreasing temperature, which leads to unreasonably high soil and litter carbon pools at high latitudes in CTEM. The CTEM parametrization avoids these issues with a Q_{10} value of about 2.0 for temperatures less than $20\,C$, while a decreasing value of Q_{10} at temperatures above $20\,C$ ensures that the respiration rate does not increase indefinitely.

The temperature of the litter pool is a weighted average of the temperature of the top soil layer (T_1) and the root temperature ($T_{\rm R}$) as litter consists of leaf, stem, and root litter ($T_{\rm D} = 0.7T_1 + 0.3T_{\rm R}$). The temperature of the soil carbon pool is calculated as the mean soil temperature in the rooting zone based upon the fraction of roots in each soil layer and their temperature. The carbon in each soil layer is not explicitly tracked but assumed to adopt an exponential distribution [25].

The response of heterotrophic respiration to soil moisture is formulated through soil matric potential (Ψ; MPa). While soil matric potential values are usually negative, the formulation uses absolute values to allow its logarithm to be taken. Absolute values of soil matric potential are high when soil is dry and low when it is wet. The primary premise of soil moisture control on heterotrophic respiration is that heterotrophic respiration is constrained both when the soils are dry (due to reduced microbial activity) and when they are wet (due to impeded oxygen supply to microbes) with optimum conditions in-between. The exception is the respiration from the litter component, which is assumed to be continually exposed to air, and thus never oxygen deprived, even when soil moisture content is high ($0.04 > |\Psi| \geq |\Psi_{sat}|$, where Ψ_{sat} is the soil matric potential at saturation). The soil moisture dependence thus varies between 0 and 1 with matric potential as follows:

for $0.04 > |\Psi| \geq |\Psi_{sat}|$

$$f(\Psi)_{\rm H} = 1 - 0.5\frac{\log(0.04) - \log|\Psi|}{\log(0.04) - \log|\Psi_{sat}|} f(\Psi)_D = 1;$$

for $0.06 \geq |\Psi| \geq 0.04$

$$f(\Psi)_{\{}D, H\} = 1;$$

for $100.0 \geq |\Psi| > 0.06$

$$f(\Psi)_{\{}D, H\} = 1 - 0.8\frac{\log|\Psi| - \log(0.06)}{\log(100) - \log(0.06)};$$

for $|\Psi| > 100.0$

$$f(\Psi)_{\{}D, H\} = 0.2.$$

Heterotrophic respiration for bare ground is treated separately in CTEM. The carbon contributions to the bare ground litter and soil carbon pools come via processes such as creation of bare ground due to fire, competition between PFTs and land use change. The heterotrophic respiration is sensitive to temperature and moisture in the same manner as vegetated areas using Eqs. (lithet})–(lastpsi}). The base respiration rates of $\varsigma_{D,bare}$ and $\varsigma_{H,bare}$ are set to 0.5605 and $0.02258\,kg\,C\,(kg\,C)^{-1}\,yr^{-1}$, respectively.

The amount of humidified litter, which is transferred from the litter to the soil carbon pool ($C_{\rm D\rightarrow H}$) is modelled as a fraction of litter respiration ($R_{h,D}$) as

$$C_{\rm D\rightarrow H} = \chi\, R_{h,D}$$

where χ (see also ctem_params.f90) is the PFT-dependent humification factor and varies between 0.4 and 0.5. For crops, χ is set to 0.1 to account for reduced transfer of humidified litter to the soil carbon pool which leads to loss in soil carbon when natural vegetation is converted to croplands. Over the bare ground fraction χ is set to 0.45.

With heterotrophic respiration known, net ecosystem productivity (NEP) is calculated as

$$NEP = G_{canopy} - R_{\rm m} - R_{\rm g} - R_{\rm h}.$$

9.40.2 Function/Subroutine Documentation

9.40.2.1 subroutine hetresg (real, dimension(ilg,icc+1) *litrmass,* real, dimension(ilg,icc+1) *soilcmas,* integer *il1,* integer *il2,* real, dimension(ilg,ignd) *tbar,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,ignd) *sand,* real, dimension(ilg,ignd) *clay,* real, dimension(ilg,ignd) *zbotw,* real, dimension(ilg) *frac,* integer *isnow,* integer, dimension(ilg,ignd) *isand*)

Parameters

il1	il1=1
il2	il2=ilg
isnow	integer telling if bare fraction is fg (0) or fgs (1), isnow is changed to isnow(ilg) in classt of class version higher than 3.4 for coupling with ctem
litrmass	litter mass for the 8 pfts + bare in kgc/m^2
soilcmas	soil carbon mass for the 8 pfts + bare in kgc/m^2

tbar	soil temperature, k
thliq	liquid soil moisture content in 3 soil layers
sand	percentage sand
zbotw	bottom of soil layers
clay	percentage clay
frac	fraction of ground (fg) or snow over ground (fgs)

Constants and parameters are located in ctem_params.f90

initialize required arrays to zero

initialization ends

estimate temperature of the litter and soil carbon pools.

over the bare fraction there is no live root. so we make the simplest assumption that litter temperature is same as temperature of the top soil layer.

we estimate the temperature of the soil c pool assuming that soil carbon over the bare fraction is distributed exponentially. note that bare fraction may contain dead roots from different pfts all of which may be distributed differently. for simplicity we do not track each pft's dead root biomass and assume that distribution of soil carbon over the bare fraction can be described by a single parameter.

make sure we don't use temperatures of 2nd and 3rd soil layers if they are specified bedrock via sand -3 flag

third layer bed rock

second layer bed rock

find moisture scalar for soil c decomposition

this is modelled as function of logarithm of matric potential. we find values for all soil layers, and then find an average value based on fraction of carbon present in each layer.

set to large number so that ltrmoscl becomes 0.2

if sand.eq.-3 or -4

make sure we don't use scmotrm of 2nd and 3rd soil layers if they are specified bedrock via sand -3 flag

find moisture scalar for litter decomposition

the difference between moisture scalar for litter and soil c is that the litter decomposition is not constrained by high soil moisture (assuming that litter is always exposed to air). in addition, we use moisture content of the top soil layer as a surrogate for litter moisture content. so we use only psi(i,1) calculated in loops 260 and 270 above.

use temperature of the litter and soil c pools, and their soil moisture scalars to find respiration rates from these pools

first find the q10 response function to scale base respiration rate from 15 c to current temperature, we do litter first

2.64 converts bsratelt_g from kg c/kg c.year to u-mol co2/kg c.s respiration from soil c pool

2.64 converts bsratesc_g from kg c/kg c.year to u-mol co2/kg c.s

9.41 hetresv_old.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Heterotrophic Respiration Subtoutine For Vegetated Fraction.

Functions/Subroutines

- subroutine hetresv (fcan,fct, litrmass, soilcmas,il1,il2,tbar,thliq,sand,clay, roottemp,zbotw,sort,isand,

9.41.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Heterotrophic Respiration Subtoutine For Vegetated Fraction.

9.41.2 Function/Subroutine Documentation

9.41.2.1 subroutine hetresv (real, dimension(ilg,icc) *fcan,* real, dimension(ilg) *fct,* real, dimension(ilg,icc+1) *litrmass,* real, dimension(ilg,icc+1) *soilcmas,* integer *il1,* integer *il2,* real, dimension(ilg,ignd) *tbar,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,ignd) *sand,* real, dimension(ilg,ignd) *clay,* real, dimension(ilg,icc) *roottemp,* real, dimension(ilg,ignd) *zbotw,* integer, dimension(icc) *sort,* integer, dimension(ilg,ignd) *isand*)

Parameters

il1	il1=1
il2	il2=ilg
sort	index for correspondence between 9 pfts and 12 values in the parameters vectors
fcan	fractional coverage of ctem's 9 pfts
fct	sum of all fcan, fcan & fct are not used at this time but could be used at some later stage
litrmass	litter mass for the 9 pfts + bare in kgc/m^2
tbar	soil temperature, k
soilcmas	soil carbon mass for the 9 pfts + bare in kgc/m^2
thliq	liquid soil moisture content in 3 soil layers
sand	percentage sand
clay	percentage clay
roottemp	root temperature as estimated in mainres subroutine
zbotw	bottom of soil layers

Constants and parameters are located in ctem_params.f90

parameters of the hyperbolic tan q10 formulation

initialize required arrays to zero

initialization ends

estimate temperature of the litter and soil carbon pools. litter temperature is weighted average of temperatue of top soil layer (where the stem and leaf litter sits) and root temperature, because litter pool is made of leaf, stem, and root litter.

estimation of soil carbon pool temperature is not straight forward. ideally soil c pool temperature should be set same as root temperature, since soil c profiles are similar to root profiles. but in the event when the roots die then we may run into trouble. so we find the temperature of the soil c pool assuming that soil carbon is exponentially distributed, just like roots. but rather than using the parameter of this exponential profile from our variable root distribution we use fixed vegetation-dependent parameters.

fraction of carbon in

soil layers

make sure we don't use temperatures of 2nd and 3rd soil layers if they are specified bedrock via sand -3 flag

third layer bed rock

second layer bed rock

find moisture scalar for soil c decomposition

this is modelled as function of logarithm of matric potential. we find values for all soil layers, and then find an average value based on fraction of carbon present in each layer. this makes moisture scalar a function of vegetation type.

set to large number so that ltrmoscl becomes 0.2

sand.eq.-3 or -4

make sure we don't use scmotrm of 2nd and 3rd soil layers if they are specified bedrock via sand -3 flag

third layer bed rock

second layer bed rock

find moisture scalar for litter decomposition

the difference between moisture scalar for litter and soil c is that the litter decomposition is not constrained by high soil moisture (assuming that litter is always exposed to air). in addition, we use moisture content of the top soil layer as a surrogate for litter moisture content. so we use only psi(i,1) calculated in loops 260 and 270 above.

use temperature of the litter and soil c pools, and their soil moisture scalars to find respiration rates from these pools

first find the q10 response function to scale base respiration rate from 15 c to current temperature, we do litter first

2.64 converts bsratelt from kg c/kg c.year to u-mol co2/kg c.s respiration from soil c pool

2.64 converts bsratesc from kg c/kg c.year to u-mol co2/kg c.s

9.42 ICEBAL.f File Reference

Purpose: Perform temperature stepping and surface runoff calculations over ice sheets.

Functions/Subroutines

- subroutine icebal (TBAR, TPOND, ZPOND, TSNOW, RHOSNO, ZSNOW, HCPSNO, ALBSNO, HMFG, H↩ TCS, HTC, WTRS, WTRG, GFLUX, RUNOFF, TRUNOF, OVRFLW, TOVRFL, ZPLIM, GGEO, FI, EVAP, R, TR, GZERO, G12, G23, HCP, QMELT, WSNOW, ZMAT, TMOVE, WMOVE, ZRMDR, TADD, ZMOVE, TBOT, DELZ, ISAND, ICONT, IWF, IG, IGP1, IGP2, ILG, IL1, IL2, JL, N)

9.42.1 Detailed Description

Purpose: Perform temperature stepping and surface runoff calculations over ice sheets.

9.42.2 Function/Subroutine Documentation

9.42.2.1 **subroutine icebal (real, dimension (ilg,ig) *TBAR,* real, dimension (ilg) *TPOND,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *HCPSNO,* real, dimension(ilg) *ALBSNO,* real, dimension (ilg,ig) *HMFG,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *WTRS,* real, dimension (ilg) *WTRG,* real, dimension (ilg,ig) *GFLUX,* real, dimension(ilg) *RUNOFF,* real, dimension(ilg) *TRUNOF,* real, dimension(ilg) *OVRFLW,* real, dimension(ilg) *TOVRFL,* real, dimension (ilg) *ZPLIM,* real, dimension (ilg) *GGEO,* real, dimension (ilg) *FI,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg) *GZERO,* real, dimension (ilg) *G12,* real, dimension (ilg) *G23,* real, dimension (ilg,ig) *HCP,* real, dimension (ilg) *QMELT,* real, dimension (ilg) *WSNOW,* real, dimension (ilg,igp2,igp1) *ZMAT,* real, dimension (ilg,igp2) *TMOVE,* real, dimension (ilg,igp2) *WMOVE,* real, dimension (ilg,igp1) *ZRMDR,* real, dimension (ilg) *TADD,* real, dimension (ilg) *ZMOVE,* real, dimension (ilg) *TBOT,* real, dimension (ig) *DELZ,* integer, dimension (ilg,ig) *ISAND,* integer, dimension (ilg) *ICONT,* integer *IWF,* integer *IG,* integer *IGP1,* integer *IGP2,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)**

Parameters

tbar	Temperature of ice layer $[C](T_{av}(\Delta z))$
hmfg	Energy associated with freezing or thawing of water in ice layer $[Wm^{-2}]$
htc	Internal energy change of ice layer due to conduction and/or change in mass $[Wm^{-2}]$ (lj)

gflux	Heat flow between ice layers $[Wm^{-2}](G(\Delta z))$
tpond	Temperature of ponded water [C]
zpond	Depth of ponded water [m]
tsnow	Temperature of the snow pack [C]
rhosno	Density of snow pack $[kgm^{-3}]$
zsnow	Depth of snow pack [m]
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}]$
albsno	Albedo of snow [ ]
htcs	Internal energy change of ice layer due to conduction and/or change in mass $[Wm^{-2}](I_j)$
wtrs	Water transferred into or out of the snow pack $[kgm^{-2}s^{-1}]$
wtrg	Water transferred into or out of the ice $[kgm^{-2}s^{-1}]$
runoff	Total runoff from ice column [m]
trunof	Temperature of total runoff from ice column [K]
ovrflw	Overland flow from top of ice column [m]
tovrfl	Temperature of overland flow from top of ice column [K]
fi	Fractional coverage of subarea in question on modelled area $[](X)$
evap	Evaporation rate from ice surface $[ms^{-1}]$
r	Rainfall rate at ice surface $[ms^{-1}]$
tr	Temperature of rainfall [C]
gzero	Heat flow into ice surface $[Wm^{-2}](G(0))$
g12	Heat flow between first and second ice layers $[Wm^{-2}](G(\Delta z1))$
g23	Heat flow between second and third ice layers $[Wm^{-2}](G(\Delta z2))$
qmelt	Energy available for melting of ice $[Wm^{-2}]$
wsnow	Liquid water content of snow pack $[kgm^{-2}]$
zplim	Limiting depth of ponded water [m]
ggeo	Geothermal heat flux at bottom of modelled ice profile $[Wm^{-2}]$
hcp	Heat capacity of ice layer $[Jm^{-3}K^{-1}]$
isand	Sand content flag
delz	Overall thickness of ice layer $[m](\Delta z)$

In the 100 loop, any rainfall or snowmelt R reaching the ice surface is added to the ponded water on the surface. The ponded water temperature is calculated as the weighted average of the existing pond and the rainfall or snowmelt added, and the change in internal energy HTC of the first ice layer is updated using the temperature of the added water.

If a full-scale hydrological modelling application is not being run, that is, if only vertical fluxes of energy and moisture are being modelled, the flag IWF will have been pre-set to zero. In this case, overland flow of water is treated using a simple approach: if the ponded depth of water on the soil surface ZPOND exceeds a pre-determined limiting value ZPLIM, the excess is assigned to overland flow. The total runoff from the ice sheet, RUNOFF, is incremented by the excess of the ponded water, and the overland flow for the whole grid cell OVRFLW is incremented by the product of the excess ponded water and the fractional area of the grid cell. The temperature of the overall runoff from the modelled area TRUNOF, and the temperature of the overland flow for the grid cell TOVRFL, are calculated as weighted averages over their previous values and the ponded water temperature TPOND. The internal energy change HTC of the first soil layer is adjusted for the amount of water lost, and ZPOND is set to ZPLIM.

If the temperature of the remaining ponded water is greater than 0 C, the sink of energy required to cool it to 0 C, HCOOL, is calculated and compared with the amount of energy required to warm the first ice layer to 0 C, HWARM. If HWARM $>$ HCOOL, the energy sink of the first layer is used to cool the ponded water to 0 C, and the layer temperature is updated accordingly. Otherwise, the ponded water temperature and the temperature of the first ice layer are both set to 0 C, and the excess energy source given by HCOOL-HWARM is added to the heat available for melting ice, QMELT.

In loop 125, if the temperature of the first ice layer is less than -2 C after the above operations (i.e. if it is not very close to 0 C), and if the ponded water depth is not vanishingly small, freezing of the ponded water can take place. The energy sink required to freeze all of the ponded water, HFREZ, is calculated and compared with HWARM, the amount of energy required to raise the temperature of the first ice layer to 0 C. If HWARM $>$ HFREZ, then HFREZ is converted into an equivalent temperature change using the heat capacity of ice, and added to the temperature of the first ice layer. HFREZ is also used to update HMFG, the diagnosed energy used for phase changes of water in the first ice layer. The internal energy of the first soil layer, HTC, is adjusted to account for the loss of the ponded

water, which is assumed to be added to the snow pack. The ponded water is converted into a frozen depth ZFREZ, which is used to update the internal energy of the snow pack, HTCS. If there is not a pre-existing snow pack, the snow albedo is set to the limiting low value of 0.50. The temperature and density of the snow pack are recalculated as weighted averages over the original values and the frozen amount that has been added. ZFREZ is added to the snow depth ZSNOW. The snow heat capacity is recalculated using the new value of the snow density. Finally, the diagnostic amounts of water transferred to the snow pack, WTRS, and from the ice, WTRG, are updated using ZFREZ.

In the 150 loop the heat fluxes between the ice layers are calculated for the optional multiple-layer configuration, in which the standard third layer, normally with a thickness of 3.75 m, can be subdivided into smaller layers and extended to a greater depth if desired. (The heat fluxes at the ice surface, and between the first and second and the second and third layers, were already calculated in the CLASST subroutines.) The remaining fluxes are calculated by using a simple linearization of the soil temperature profile. The expression for the ground heat flux at a depth z, G(z), which depends on the thermal conductivity $\lambda(z)$ and the temperature gradient, is written as:

$G(z) = \lambda(z) dT(z)/dz$

The linearized form is thus:

$G_j = 2\lambda_i (T_{j-1} \breve{} T_j)/(\Delta z_{j-1} + \Delta z_j)$

where G_j is the heat flux at the top of layer j, T_j and Δz_j refer to the temperature and thickness of the layer, and λ_i is the thermal conductivity of ice.

In the 200 loop a value is assigned to the temperature at the bottom of the ice profile, TBOT, and the temperatures for the first three ice layers are stepped ahead. If the standard three-layer configuration is being used, TBO↩ T is obtained by making use of the assumption (see documentation for subroutine TNPREP) that the variation of temperature T with depth z within each soil layer can be modelled using a quadratic equation:

$T(z) = 1/2az^2 + bz + c$

It can be shown that the temperature at the bottom of a given soil layer, $T(\Delta z)$, is related to the temperature at the top of the layer T(0) and the heat fluxes at the top and bottom of the layer, G(0) and $G(\Delta z)$, as follows:

$T(\Delta z) = T(0) - (\Delta z/2\lambda_i)[G(0) + G(\Delta z)]$

Making use of the continuity requirement that the heat flux and temperature at the bottom of a given layer must be equal to the heat flux and temperature at the top of the layer beneath it, an expression for the temperature at the bottom of the third ice layer can be obtained as a function of the temperature at the surface and the fluxes between the ice layers:

$T(\Delta z_3) = T(0) \breve{} G(\Delta z_2)[\Delta z_3 + \Delta z_2] + G(\Delta z_1)[\Delta z_2 + \Delta z_1] + G(0)\Delta z_{21}/2\lambda_i$

The surface temperature T(0) is obtained by integrating the equation for T(z) to obtain an expression for the average layer temperature $T_{av}(\Delta z)$, and then inverting this to solve for T(0):

$T(0) = T_{av}(\Delta z_1) + (\Delta z_1/3\lambda_i)[G(0) + 1/2G(\Delta z_1)]$

The third layer temperature is then updated using the geothermal flux GGEO. If the optional multiple- layer configuration is used, TBOT is simply set to the temperature of the lowest layer. In either the standard or the multiple-layer case, the first three layer temperatures are updated using the heat fluxes at the surface, between the first and second layers and between the second and third layers, which were determined in the CLASST subroutines. Finally, the latter fluxes are assigned to the appropriate levels in the diagnostic GFLUX vector.

In the 250 loop, the GFLUX values determined in the 150 loop are used to update the temperatures in the third and lower ice layers for the multiple-layer configuration. The calculations are bracketed by a determination of the change of internal energy I_j of the ice layers as a result of the heat fluxes, obtained as the difference in I_j between the beginning and end of the calculations:

$\Delta I_j = X_i \Delta[C_i \Delta z_j T_{av}(\Delta z_j)]/\Delta t$

where C_i is the heat capacity of ice, $Deltat$ is the length of the time step, and X_i is the fractional coverage of the subarea under consideration relative to the modelled area.

In the 300 loop, checks are carried out to determine whether any of the ice layer temperatures has overshot 0 C as a result of the calculations in the previous loop. If so, the excess energy is assigned to a temporary variable QADD and is also added to the total heat available for melting of the ice, QMELT. QADD is subtracted from the internal energy of the layer in questions and is added to the internal energy of the first layer, since melting is assumed to

proceed from the top downwards. The temperature of the layer is reset to 0 C. Finally, the first half of a calculation of the change of internal energy of each ice layer is performed, to be completed at the end of the subroutine.

In the next three loops, the ice layers are adjusted downward to account for the removal of mass at the surface by melting or sublimation. First, the temperature TMOVE of the ice into which the layer is moving is set to the temperature of the layer below it. TMOVE for the bottom layer is set to TBOT. A depth of ice ZMELT is calculated as the amount of ice for which QMELT is sufficient to both raise its temperature to 0 C (if necessary) and melt it. The temperature of the overall runoff and the overland flow are updated as averages of the original temperature and the meltwater temperature (assumed to be at the freezing point), weighted according to their respective amounts, and the overall runoff and overland flow are incremented by ZMELT (with ZMELT converted to an equivalent water depth). The energy used for the melting of ice is calculated from ZMELT and added to HMFG for the first layer, and the amount of energy used to raise the layer temperature to 0 C is added to HTC for the first layer. The total depth of downward adjustment of the ice layers, ZMOVE, is obtained as the sum of ZMELT and the sublimation depth, calculated from EVAP. This amount is added to the diagnostic variable WTRG. Finally, the new temperature of each ice layer is calculated over the layer thickness DELZ as the average of the original temperature weighted by DELZ-ZMOVE and TMOVE weighted by ZMOVE.

In the next loops, the ice layers are adjusted upward to account for addition of mass at the surface by conversion of snow to ice. Snow is converted to ice if the mass of the snow pack exceeds $100kgm^{-2}$, or if the density exceeds $900kgm^{-3}$ (approaching that of ice). In the first case the excess over and above $100kgm^{-2}$ is converted; in the second, the whole snow pack is converted. These calculations are performed in the 500 loop, bracketed by a calculation of the change in internal energy of the snow pack, HTCS. In both cases the first level of the ice level movement matrix WMOVE is set to the amount of snow that is converted, expressed as a depth of ice, and the first level of the temperature matrix TMOVE is set to the snow temperature. The amount of converted snow is added to the diagnostic variables WTRS and WTRG. The depth, density and heat capacity of the snow are recalculated. If the entire snow pack is being converted and the water content WSNOW was non-zero, WSNOW is subtracted from WTRS and added to WTRG, and is also added to the total runoff and the overland flow. The runoff and overland flow temperatures are updated accordingly. The snow temperature and water content are reset to zero. The amount of ice that is lost to the bottom of the profile is added to the total runoff, and the runoff temperature is updated accordingly.

In the remaining parts of the code, the actual adjustment of ice layer positions is performed. First the levels of the available depth matrix ZRMDR are set to the ice layer thicknesses; the matrix ZMAT is initialized to zero; and each level J of WMOVE and TMOVE from 2 to the bottom of the ice profile is set to the value of DELZ and TBAR respectively of the J-1 ice level. The ZMAT matrix represents the depth of each ice layer J that is occupied by ice from level K in the layer movement matrix WMOVE after the layer adjustments are complete. In the 700 loop, starting at the top of the ice profile, an attempt is made to assign each layer of WMOVE in turn to the K,J level of ZMAT. If the calculated value of ZMAT is greater than the available depth ZRMDR of the layer, ZMAT is set to ZRMDR, WMOVE is decremented by ZRMDR, and ZRMDR is set to zero. Otherwise the calculated value of ZMAT is accepted, ZRMDR is decremented by ZMAT, and WMOVE is set to zero.

Finally, the 900 loop is performed over each ice layer J to determine the new layer temperature. The temperature adjustment variable TADD is initialized to zero, and then incremented by the temperature TMOVE of each level K weighted by the corresponding K,J level of ZMAT, and finally by the original layer temperature weighted by the corresponding level of ZRMDR. The layer temperature is reset to TADD normalized by DELZ, and the updating of HTC, begun at the end of the 300 loop, is completed.

9.43 io_driver.f90 File Reference

Central module that handles all CTEM reading and writing of external files.

Functions/Subroutines

- subroutine, public io_driver::read_from_ctm (nltest, nmtest, FCANROT, FAREROT, RSMNROT, QA50ROT, VPDAROT, VPDBROT, PSGAROT, PSGBROT, DRNROT, SDEPROT, XSLPROT, GRKFROT, WFSFRO↩T, WFCIROT, MIDROT, SANDROT, CLAYROT, ORGMROT, TBARROT, THLQROT, THICROT, TCANRO↩T, TSNOROT, TPNDROT, ZPNDROT, RCANROT, SCANROT, SNOROT, ALBSROT, RHOSROT, GROROT, argbuff, onetile_perPFT)

- subroutine, public io_driver::write_ctm_rs (nltest, nmtest, FCANROT, argbuff)
- subroutine, public io_driver::create_outfiles (argbuff, title1, title2, title3, title4, title5, title6, name1, name2, name3, name4, name5, name6, place1, place2, place3, place4, place5, place6)
- subroutine, public io_driver::class_monthly_aw (IDAY, IYEAR, NCOUNT, NDAY, SBC, DELT, nltest, nmtest, ALVSROT, FAREROT, FSVHROW, ALIRROT, FSIHROW, GTROT, FSSROW, FDLROW, HFSROT, ROF↩ ROT, PREROW, QFSROT, QEVPROT, SNOROT, TAROW, WSNOROT, TBARROT, THLQROT, THICROT, TFREZ, QFCROT, QFGROT, QFNROT, QFCLROT, QFCFROT, FSGVROT, FSGSROT, FSGGROT, ACTL↩ YR, FTABLE)
- subroutine, public io_driver::class_annual_aw (IDAY, IYEAR, NCOUNT, NDAY, SBC, DELT, nltest, nmtest, ALVSROT, FAREROT, FSVHROW, ALIRROT, FSIHROW, GTROT, FSSROW, FDLROW, HFSROT, RO↩ FROT, PREROW, QFSROT, QEVPROT, TAROW, QFCROT, FSGVROT, FSGSROT, FSGGROT, ACTLYR, FTABLE, leapnow)
- subroutine, public io_driver::ctem_daily_aw (nltest, nmtest, iday, FAREROT, iyear, jdstd, jdsty, jdendd, jdendy, grclarea, onetile_perPFT)
- subroutine, public io_driver::ctem_monthly_aw (nltest, nmtest, iday, FAREROT, iyear, nday, onetile_perPFT)
- subroutine, public io_driver::ctem_annual_aw (nltest, nmtest, iday, FAREROT, iyear, onetile_perPFT, leapnow)
- subroutine, public io_driver::close_outfiles ()

9.43.1 Detailed Description

Central module that handles all CTEM reading and writing of external files.

9.44 landuse_change_mod.f90 File Reference

Central module for all land use change operations.

Functions/Subroutines

- subroutine, public landuse_change::initialize_luc (iyear, lucdat, nmtest, nltest, nol2pfts, cyclemet, cylucyr, lucyr, fcanrow, farerow, nfcancmxrow, pfcancmxrow, fcancmxrow, reach_eof, start_bare, compete, onetile_↩ perPFT)
- subroutine, public landuse_change::readin_luc (iyear, nmtest, nltest, lucyr, nfcancmxrow, pfcancmxrow, reach_eof, compete, onetile_perPFT)
- subroutine, public landuse_change::luc (il1,il2,nilg,nol2pfts,grclarea, pfcancmx, nfcancmx,iday,todfrac,yesfrac, interpol, compete,leapnow,

9.44.1 Detailed Description

Central module for all land use change operations.

9.45 mainres.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Maintenance Respiration Subroutine.

Functions/Subroutines

- subroutine mainres (fcan,fct,stemmass,rootmass,il1, il2, leapnow,tcan,tbar,rmatctem,sort, nol2pfts,isand,

9.45.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Maintenance Respiration Subroutine.

Autotrophic respiration ($mol\,CO_2\,m^{-2}\,s^{-1}$) is composed of maintenance, $R_{\rm m}$, and growth respirations, $R_{\rm g}$,

$$R_{\rm a} = R_{\rm m} + R_{\rm g}.$$

Maintenance respiration accounts for carbon consumed by processes that keep existing plant tissues alive and is a function of environmental stresses. Maintenance respiration is calculated on a half-hourly time step (with photosynthesis) for the leaves, R_{mL}, and at a daily time step for the stem, R_{mS}, and root, R_{mR}, components

$$R_{\rm m} = R_{mL} + R_{mS} + R_{mR}.$$

Maintenance respiration is generally strongly correlated with nitrogen content [45] [46]. The current version of CT↩ EM does not explicitly track nitrogen in its vegetation components. Therefore, we adopt the approach of [16] [15] in which the close relation between maximum catalytic capacity of Rubisco, $V_{\rm m}$, and leaf nitrogen content is used as a proxy to estimate leaf maintenance respiration,

$$R_{mL} = \varsigma_{\rm L} V_{\rm m}\, f_{25}(Q_10d, n) f_{PAR},$$

where $\varsigma_{\rm L}$ is set to 0.015 and 0.025 for C_3 and C_4 plants, respectively, f_{PAR} scales respiration from the leaf to the canopy level, similar to Eq. (G_canopy}), and the $f_{25}(Q_10d, n)$ function accounts for different temperature sensitivities of leaf respiration during day (d) and night (n). [43] and [55] suggest lower temperature sensitivity for leaf respiration during the day compared to night, and therefore we use values of $Q_10d = 1.3$ and $Q_10n = 2.0$ for day and night, respectively.

Maintenance respiration from the stem and root components is estimated based on PFT-specific base respiration rates ($\varsigma_{\rm S}$ and $\varsigma_{\rm R}$ specified at $15\,C, kg\,C\,(kg\,C)^{-1}\,yr^{-1}$; see also ctem_params.f90) that are modified to account for temperature response following a Q_{10} function. Maintenance respiration from stem and root components, $R_{m\{S,R\}}$, is calculated as

$$R_{{\rm m},i} = 2.64 \times 10^{-6} \varsigma_i l_{{\rm v},i} C_i f_{15}(Q_{10}), \quad i = {\rm S, R},$$

where $l_{v,i}$ is the live fraction of stem or root component, i.e. the sapwood, and C_i is the stem or root carbon mass ($kg\,C\,m^{-2}$). The constant 2.64×10^{-6} converts units from $kg\,C\,m^{-2}\,yr^{-1}$ to $mol\,CO_2\,m^{-2}\,s^{-1}$. The live sapwood fraction, $l_{{\rm v},i}$, for stem or root component is calculated following the CENTURY model [42] as

$$l_{{\rm v},i} = \max(0.05, \min[1.0, \exp^{-0.2835 C_i}]), \quad i = {\rm S, R}.$$

The Q_{10} value used in Eq. (r_msr}) is not assumed to be constant but modelled as a function of temperature following [50] as

$$Q_{10} = 3.22 - 0.046 \left(\frac{15.0 + T_{\{S,R\}}}{1.9} \right),$$

where $T_{\{S,R\}}$ is stem or root temperature (C). Stem temperature is assumed to be the same as air temperature while root temperature is based on the soil temperature weighted by the fraction of roots present in each soil layer [8]. The calculated Q_{10} value is additionally constrained to be between 1.5 and 4.0.

Growth respiration, $R_{\rm g}$ ($mol\,CO_2\,m^{-2}\,s^{-1}$), is estimated as a fraction ($\epsilon_{\rm g} = 0.15$) of the positive gross canopy photosynthetic rate after maintenance respiration has been accounted for

$$R_{\rm g} = \epsilon_{\rm g} \max[0, (G_{canopy} - R_{\rm m})].$$

Finally, net primary productivity (NPP) is calculated as

$$NPP = G_{canopy} - R_{\rm m} - R_{\rm g}.$$

9.45.2 Function/Subroutine Documentation

9.45.2.1 subroutine mainres (real, dimension(ilg,icc) *fcan,* real, dimension(ilg) *fct,* real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* integer *il1,* integer *il2,* logical *leapnow,* real, dimension(ilg) *tcan,* real, dimension(ilg,ignd) *tbar,* real, dimension(ilg,icc,ignd) *rmatctem,* integer, dimension(icc) *sort,* integer, dimension(ican) *nol2pfts,* integer, dimension(ilg,ignd) *isand*)

Parameters

il1	il1=1
il2	il2=ilg
sort	index for correspondence between 9 pfts and 12 values in the parameter vectors
nol2pfts	number of level 2 ctem pfts
isand	flag for bedrock or ice in a soil layer
leapnow	true if this year is a leap year. Only used if the switch 'leap' is true.
fcan	fractional coverage of ctem's 9 pfts over the given sub-area
fct	sum of all fcan fcan & fct are not used at this time but could be used at some later stage
stemmass	stem biomass for the 9 pfts in kgc/m^2
tcan	canopy temperature, k
tbar	soil temperature, k
rootmass	root biomass for the 9 pfts in kgc/m^2
rmatctem	fraction of roots in each layer for each pft

Constants and parameters are located in ctem_params.f90

set the following switch to .true. for using constant temperature indepedent q10 specified below

q10 - if using a constant temperature independent value, i.e. if consq10 is set to true

initialize required arrays to zero

initialization ends

based on root and stem biomass, find fraction which is live. for stem this would be the sapwood to total wood ratio.

fraction of roots for each vegetation type, for each soil layer, in each grid cell is given by rmatctem (grid cell, veg type, soil layer) which bio2str subroutine calculates. rmatctem can thus be used to find average root temperature for each plant functional type

we assume that stem temperature is same as canopy temperature tcan. using stem and root temperatures we can find their maintenance respirations rates

first find the q10 response function to scale base respiration rate from 15 c to current temperature, we do the stem first.

when finding temperature dependent q10, use temperature which is close to average of actual temperature and the temperature at which base rate is specified

This q10 value is then used with the base rate of respiration (commonly taken at some reference temperature (15 deg c), see Tjoelker et al. 2009 New Phytologist or Atkin et al. 2000 New Phyto for an example.). Long-term acclimation to temperature could be occuring see King et al. 2006 Nature SOM for a possible approach. JM.

convert kg c/m2.day -> u mol co2/m2.sec

root respiration

convert kg c/m2.day -> u mol co2/m2.sec

9.46 mortality.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Mortality Subroutine.

Functions/Subroutines

- subroutine mortalty (stemmass, rootmass,ailcg, gleafmas,bleafmas,il1,il2, leapnow,iday, sort,fcancmx,

9.46.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Mortality Subroutine.

Mortality

The PFT-dependent mortality rate (day^{-1}),

$$m_\alpha = m_{intr,\alpha} + m_{ge,\alpha} + m_{bioclim,\alpha} + m_{dist,\alpha},$$

reflects the net effect of four different processes: (1) intrinsic- or age-related mortality, m_{intr}, (2) growth or stress-related mortality, m_{ge}, (3) mortality associated with bioclimatic criteria, $m_{bioclim}$ and (4) mortality associated with disturbances, m_{dist}.

Intrinsic- or age-related mortality uses a PFT-specific maximum age, A_{max} (see also ctem_params.f90), to calculate an annual mortality rate such that only $1\,\%$ of tree PFTs exceed A_{max}, α. Intrinsic mortality accounts for processes, whose effect is not explicitly captured in the model including insect damage, hail, wind throw, etc.,

$$m_{intr,\alpha} = 1 - \exp(-4.605/A_{max,\alpha}).$$

Grasses and crops have $m_{intr} = 0$. The annual growth-related mortality m_{ge} is calculated using growth efficiency of a PFT over the course of the previous year following [14] and [49] as

$$m_{ge,\alpha} = \frac{m_{ge,max,\alpha}}{1 + k_m g_{e,\alpha}},$$

where $m_{ge,max}$ represents the PFT-specific maximum mortality rate when no growth occurs (see also ctem_↩ params.f90). k_m is a parameter set to $0.3\,m^2\,(g\,C)^{-1}$. g_e is the growth efficiency of the PFT ($g\,C\,m^{-2}$) calculated based on the maximum LAI ($L_{\alpha,max}$; $m^2\,m^{-2}$) and the increment in stem and root mass over the course of the previous year (ΔC_S and ΔC_R; $kg\,C\,m^{-2}$, respectively) [54]

$$g_{e,\alpha} = 1000 \frac{\max(0, (\Delta C_{S,\alpha} + \Delta C_{R,\alpha}))}{L_{\alpha,max}}.$$

Mortality associated with bioclimatic criteria, $m_{bioclim}$ ($0.25\,yr^{-1}$), is applied when climatic conditions in a grid cell become unfavourable for a PFT to exist and ensures that PFTs do not exist outside their bioclimatic envelopes, as explained in the next section.

The annual mortality rates for m_{intr}, m_{ge} and $m_{bioclim}$ are converted to daily rates and applied at the daily time step of the model, while m_{dist} is calculated by the fire module of the model based on daily area burned for each PFT as summarized in Appendix fire}. In practice, the $\frac{df_\alpha}{dt} = -m_{dist,\alpha} f_\alpha$ term of Eq. (compact}) is implemented right after area burnt is calculated.

9.46.2 Function/Subroutine Documentation

9.46.2.1 subroutine mortalty (real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* real, dimension(ilg,icc) *ailcg,* real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *bleafmas,* integer *il1,* integer *il2,* logical *leapnow,* integer *iday,* integer, dimension(icc) *sort,* real, dimension(ilg,icc) *fcancmx*)

Parameters

il1	il1=1
il2	il2=ilg
iday	day of the year
leapnow	true if this year is a leap year. Only used if the switch 'leap' is true.
sort	index for correspondence between ctem 9 pfts and size 12 of parameters vectors
stemmass	stem mass for each of the 9 ctem pfts, kgc/m^2
rootmass	root mass for each of the 9 ctem pfts, kgc/m^2
gleafmas	green leaf mass for each of the 9 ctem pfts, kgc/m^2
ailcg	green or live lai
bleafmas	brown leaf mass for each of the 9 ctem pfts, kgc/m^2

Constants and parameters are located in ctem_params.f90

initialize required arrays to zero

initialization ends

at the end of every year, i.e. when iday equals 365, we calculate growth related mortality. rather than using this number to kill plants at the end of every year, this mortality rate is applied gradually over the next year.

calculate growth related mortality using last year's growth efficiency or the new growth efficiency if day is 365 and growth efficiency estimate has been updated above.

convert (1/year) rate into (1/day) rate

calculate intrinsic mortality rate due to aging which implicity includes effects of frost, hail, wind throw etc. it is assumed that only 1% of the plants exceed maximum age (which is a pft-dependent parameter). to achieve this some fraction of the plants need to be killed every year.

convert (1/year) rate into (1/day) rate

now that we have both growth related and intrinsic mortality rates, lets combine these rates for every pft and estimate litter generated

9.47 mvidx.f File Reference

Functions/Subroutines

- integer function mvidx (V, N, X)

9.47.1 Function/Subroutine Documentation

9.47.1.1 integer function mvidx (real, dimension(n) *V*, integer *N*, real *X*)

- GIVEN A MONOTONIC VECTOR V OF LENGTH N AND A VALUE X,
- RETURN THE INDEX MVIDX SUCH THAT X IS BETWEEN
- V(MVIDX) AND V(MVIDX+1).
-
- V MUST BE MONOTONIC, EITHER INCREASING OF DECREASING.
- THERE IS NO CHECK ON WHETHER OR NOT THIS VECTOR IS
- MONOTONIC.
-

- THIS FUNCTION RETURNS 1 OR N-1 IF X IS OUT OF RANGE.
-
- INPUT:
-
- REAL V(N) ...MONITONIC VECTOR (INCREASING OR DECREASING)
- INTEGER N ...SIZE OF V
- REAL X ...SINGLE REAL VALUE
-
- OUTPUT:
-
- V(MVIDX) .LE/.GE. X .LE./.GE. V(MVIDX+1)

9.48 oldphen.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Phenology, Leaf Turnover & Mortality Subroutine.

Functions/Subroutines

- subroutine phenolgy (gleafmas, bleafmas,il1,il2,tbar,thliq,wiltsm,fieldsm,ta,anveg,iday,radl, roottemp,rmatctem, stemmass, rootmass,sort,nol2pfts,fcancmx,

9.48.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Phenology, Leaf Turnover & Mortality Subroutine.

The leaf phenology parametrization used in CTEM v. 1.0 is described in detail by [10]. Changes between version 1.0 and 2.0 are limited to parameter values and the parametrization is briefly described here. There are four different leaf phenological states in which vegetation can be at a given instant: (i) no leaves or dormant, (ii) maximum growth, (iii) normal growth and (iv) leaf fall or harvest. PFTs may go through only some, or all, of these phenological states depending on their deciduousness. A broadleaf cold deciduous tree, for example, transitions through all these four states in a year. In winter, the broadleaf cold deciduous trees are in the no leaves/dormant state; favourable climatic conditions in spring trigger leaf growth and the tree enters the maximum leaf growth state when all the NPP is allocated to leaves to accelerate leaf out; when the LAI reaches a threshold (described below) the tree enters the normal leaf growth state and NPP is also allocated to stem and root components; finally the arrival of autumn triggers leaf fall and the trees go into the leaf fall mode where no carbon is allocated to leaves (but it continues for roots and stems). When all the leaves have been shed, the trees go into the no leaves or dormant state again and the cycle is repeated the next year. The evergreen tree PFTs and the grass PFTs do not enter the leaf fall state and maintain a leaf canopy as long as environmental conditions are favourable. Although drought and cold stress cause accelerated leaf loss compared to the normal leaf turnover from these PFTs, they do not explicitly go into the leaf fall mode where the intent is to lose all leaves in a specified amount of time.

The leaf phenological state transitions are dependent upon environmental conditions. In particular, the transition from no leaves/dormant state to the maximum growth state is based on the carbon-gain approach. CTEM uses *virtual* leaves to assess favourable meteorological conditions for leaf out. The virtual leaves photosynthesize and respire in a manner similar to normal leaves except the carbon gain or loss is not taken into account in vegetation's carbon balance. A positive net leaf photosynthesis rate ($G_{canopy,net}$, Eq. Gnet}) for the virtual leaves over seven consecutive days indicates the arrival of favourable growth conditions and triggers leaf onset and the associated transition from the no leaves/dormant state to the maximum leaf growth state, when the entire positive NPP is allocated to leaves ($a_{fL} = 1$, $a_{fS} = a_{fR} = 0$). When LAI reaches LAI_{thrs} then the vegetation switches to the

normal growth mode and positive NPP is allocated to all three vegetation components – leaves, stem and roots ($a_{fL}, a_{fS}, a_{fR} > 0$). LAI_{thrs} is calculated as

$$LAI_{thrs} = L_{\rm f}\left[SLA\left(\frac{C_{\rm S}+C_{\rm R}}{\eta}\right)^{1/\kappa}\right].$$

The PFT-specific $L_{\rm f}$ term (see also ctem_params.f90) calculates LAI_{thrs} to be typically between 40 and 50\,{%} of the maximum LAI that a given amount of stem and root biomass can support (based on the terms in the square brackets and Eq. (propwoody}). SLA is the specific leaf area (Eq. sla}). This rule for transition from a maximum to a normal growth state is also used for evergreen tree PFTs and grass PFTs. Similar to LAI_{thrs}, the LAI of virtual leaves is $7.5\,\%$ of the maximum LAI a given amount of root and stem biomass can support for tree and crop PFTs and $2.5\,\%$ for grass PFTs. In addition, the LAI of virtual leaves is constrained to be, at least, $0.3\,m^2\,m^{-2}$ for tree PFTs and $0.2\,m^2\,m^{-2}$ for crop and grass PFTs.

The transition from the normal growth state to the leaf fall state is triggered by unfavourable environmental conditions and shorter day length. Broadleaf deciduous trees transition to the leaf fall state when either: (i) day length is less than $11\,h$ and the rooting zone temperature drops below $11.15\,C$ or (ii) when the rooting zone temperature drops below $8\,C$ regardless of the day length. Needleleaf deciduous tress begin leaf fall after seven consecutive days with daily mean air temperature below $-5\,C$. Leaf fall occurs over a period of 15 days. In the leaf fall state, the vegetation continues carbon allocation to its root and stem components, but not to leaves ($a_{fL} = 0$, $a_{fS} + a_{fR} = 1$). Evergreen trees and grasses do not enter the leaf fall state and neither do the broadleaf drought deciduous trees. The implication for the latter PFT is that if the climate changes and the dry season becomes shorter, then the trees will keep their leaves on for a longer period of time since broadleaf drought deciduous trees lose leaves due to soil moisture stress (described below).

The model vegetation is able to transition between the different leaf phenological states in response to changing conditions. For example, a leaf out in spring for broadleaf cold deciduous trees can be interrupted by a cold event when the vegetation goes into a leaf fall state until the return of more favourable conditions.

Leaf litter generation is caused by normal turnover of leaves ($\Omega_{\rm N}$, day^{-1}) and also due to cold ($\Omega_{\rm C}$, day^{-1}) and drought ($\Omega_{\rm D}$, day^{-1}) stress, both of which contribute to seasonality of LAI. For example, the leaf loss associated with drought and reduced photosynthesis during the dry season are the principal causes of the seasonality of LAI for the broadleaf drought deciduous tree PFT.

The conversion of leaf carbon to leaf litter ($D_{\rm L}$, $kg\,C\,m^{-2}\,day^{-1}$) is expressed as

$$D_{\rm L} = C_{\rm L}[1-\exp(-\Omega_{\rm N}-\Omega_{\rm C}-\Omega_{\rm D})],$$

where ($\Omega_{N,C,D}$, day^{-1}) are the leaf loss rates associated with normal turnover of leaves and the cold and drought stress. The rate of normal turnover of leaves is governed by PFT-specific leaf lifespan ($\tau_{\rm L}$, yr) as $\Omega_{\rm N} = 1/365\tau_{\rm L}$ (see also ctem_params.f90} for PFT specific values of $\tau_{\rm L}$). The leaf loss rate associated with cold stress ($\Omega_{\rm C}$) is calculated as

$$\Omega_{\rm C} = \Omega_{C,max} L_{cold}^3,$$

where $\Omega_{C,max}$ (day^{-1}, see also ctem_params.f90) is the maximum cold stress loss rate. L_{cold} is a scalar that varies between 0 and 1 as

$$L_{cold} = \{\,1\,,\quad T_{\rm a} < \left(T_{cold}^{leaf}-5\right)\,1-\frac{T_{\rm a}-\left(T_{cold}^{leaf}-5\right)}{5},\quad T_{cold}^{leaf} > T_{\rm a} > (T_{cold}^{leaf}-5)0,\quad T_{\rm a} > T_{cold}^{leaf},$$

where T_{cold}^{leaf} is a PFT-specific temperature threshold below which a PFT experiences damage to its leaves promoting leaf loss (see also ctem_params.f90) and $T_{\rm a}$ is the daily mean air temperature (C). The leaf loss rate due to drought stress is calculated in a similar manner

$$\Omega_{\rm D} = \Omega_{{\rm D},max}\,(1-\phi_{root})^3,$$

where $\Omega_{{\rm D},max}$ (day^{-1}, see also ctem_params.f90) is the maximum drought stress loss rate and ϕ_{root} (Eq. degsoilsat}) is the degree of soil saturation in the rooting zone.

9.48.2 Function/Subroutine Documentation

9.48.2.1 subroutine phenolgy (real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *bleafmas,* integer *il1,* integer *il2,* real, dimension(ilg,ignd) *tbar,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,ignd) *wiltsm,* real, dimension(ilg,ignd) *fieldsm,* real, dimension(ilg) *ta,* real, dimension(ilg,icc) *anveg,* integer *iday,* real, dimension(ilg) *radl,* real, dimension(ilg,icc) *roottemp,* real, dimension(ilg,icc,ignd) *rmatctem,* real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* integer, dimension(icc) *sort,* integer, dimension(ican) *nol2pfts,* real, dimension(ilg,icc) *fcancmx*)

Parameters

il1	il1=1
il2	il2=ilg
iday	day of year
sort	index for correspondence between 9 pfts and the 12 values in parameters vectors
nol2pfts	number of level 2 ctem pfts
gleafmas	green or live leaf mass in kgc/m^2, for the 9 pfts
bleafmas	brown or dead leaf mass in kgc/m^2, for the 9 pfts
ta	air temperature, k
tbar	soil temperature, k
thliq	liquid soil moisture content in 3 soil layers
anveg	net photosynthesis rate of ctem's pfts, umol co2/m2.s
roottemp	root temperature, which is a function of soil temperature of course, k.
rmatctem	fraction of roots in each soil layer for each pft
stemmass	stem mass for each of the 9 ctem pfts, kgc/m^2
rootmass	root mass for each of the 9 ctem pfts, kgc/m^2
fcancmx	max. fractional coverage of ctem's 9 pfts, but this can be modified by land-use change, and competition between pfts
fieldsm	field capacity soil moisture content both calculated in allocate subroutine
wiltsm	wilting point soil moisture content
radl	latitude in radians

Constants and parameters are located in ctem_params.f90

initialize required arrays to zero

initialization ends

convert green leaf mass into leaf area index using specific leaf area $(sla, m^2/kgc)$ estimated using leaf life span. see bio2str subroutine for more details.

also find threshold lai as a function of stem+root biomass which is used to determine leaf status

using green leaf area index (ailcg) determine the leaf status for each pft. loops 190 and 200 thus initialize lfstatus, if this this information is not passed specifically as an initialization quantity.

knowing lfstatus (after initialization above or using value from from previous time step) we decide if we stay in a given leaf mode or we move to some other mode.

we start with the "no leaves" mode

add one to pandays(i,j) if daily an is positive, otherwise set it to zero.

if in "no leaves" mode check if an has been positive over last dayschk(j) days to move into "max. growth" mode. if not we stay in "no leaves" mode. also set the chkmode(i,j) switch to 1.

find day length using day of year and latitude. this is to be used for initiating leaf offset for broad leaf dcd trees.

even if pandays criteria has been satisfied do not go into max. growth mode if environmental conditions are such that they will force leaf fall or harvest mode.

needle leaf dcd

broad leaf dcd cld & dry

crops

similar to the way we count no. of days when an is positive, we find no. of days when temperature is below -5 c. we need this to determine if we go into "leaf fall" mode for needle leaf dcd trees.

also estimate no. of days below 8 c. we use these days to decide if its cold enough to harvest crops.

if in "max growth" mode

if mode hasn't been checked and we are in "max. growth" mode, then check if we are above pft-dependent lai threshold. if lai is more then this threshold we move into "normal growth" mode, otherwise we stay in "max growth" mode so that leaves can grow at their max. climate-dependent rate

for dcd trees we also need to go into "leaf fall" mode directly from "max. growth" mode.

ndl dcd

bdl dcd cold

bdl dcd dry

if in "normal growth" mode

if in "normal growth" mode then go through every pft individually and follow set of rules to determine if we go into "fall/harvest" mode

needle leaf evg

needle leaf dcd

broad leaf evg

broad leaf dcd cold we use daylength and roottemp to initiate leaf offset

broad leaf dcd dry we still use daylength and roottemp to initiate leaf offset, for the pathological cases of dry dcd trees being further away from the equator then we can imagine. other wise leaf loss will occur due to drought anyway.

"normal growth" to "fall/harvest" transition for crops is based on specified lai. we harvest if lai of crops reaches a threshold. if lai doesn't reach this threshold (say due to a bad year) we harvest anyway if it starts getting cold, otherwise we don't harvest.

"normal growth" to "max. growth" transition for grasses

if in "fall/harvest" mode

grasses and evg trees do not come into this mode, because they want to stay green if possible. this mode is activated for dcd plants and crops. once in this mode dcd trees loose their leaves and crops are harvested. ndl dcd trees keep loosing their leaves at rate determined by cold stress, bdl dcd trees loose their leaves at a specified rate, and crops are harvested over a period of 15 days. dcd trees and crops stay in "leaf fall/harvest" model until all green leaves are gone at which time they switch into "no leaves" mode, and then wait for the climate to become favourable to go into "max. growth" mode

check that leaf status of all vegetation types in all grid cells has been updated

having decided leaf status for every pft, we now calculate normal leaf turnover, cold and drought stress mortality, and for bdl dcd plants we also calculate specified loss rate if they are in "leaf fall" mode, and for crops we calculate harvest loss, if they are in "harvest" mode.

all these loss calculations will yield leaf litter in kgc/m^2 for the given day for all pfts

normal leaf turn over

for drought stress related mortality we need field capacity and wilting point soil moisture contents, which we calculated in allocate subroutine

estimate (1-drought stress)

estimate drought stress term averaged over the rooting depth for each pft

using this drought stress term and our two vegetation-dependent parameters we find leaf loss rate associated with drought

drought related leaf loss rate

estimate leaf loss in kgc/m^2 due to drought stress

similar to drgtstrs we find coldstrs for each pft. we assume that max. cold stress related leaf loss occurs when temperature is 5 c or more below pft's threshold

using this cold stress term and our two vegetation-dependent parameters we find leaf loss rate associated with cold cold related leaf loss rate

estimate leaf loss in kgc/m^2 due to cold stress

now that we have all types of leaf losses (due to normal turnover, cold and drought stress, and fall/harvest) we take the losses for grasses and use those to turn live green grass into dead brown grass. we then find the leaf litter from the brown grass which will then go into the litter pool.

we assume life span of brown grass is 10% that of green grass but this is an adjustable parameter.

combine nrmlloss, drgtloss, and coldloss together to get total leaf litter generated, which is what we use to update gleafmass, except for grasses, for which we have alerady updated gleafmass.

9.49 ORDLEG.f File Reference

THIS ROUTINE IS A SUBSET OF BELOUSOVS ALGORITHM USED TO CALCULATE ORDINARY LEGENDRE POLYNOMIALS.

Functions/Subroutines

- subroutine **ordleg** (SX, COA, IR)

9.49.1 Detailed Description

THIS ROUTINE IS A SUBSET OF BELOUSOVS ALGORITHM USED TO CALCULATE ORDINARY LEGENDRE POLYNOMIALS.

SX = LEGENDRE POLYNOMIAL EVALUATED AT COA

COA = COSINE OF COLATITUDE

IR = WAVE NUMBER

9.50 phenolgy.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Phenology, Leaf Turnover & Mortality Subroutine.

Functions/Subroutines

- subroutine phenolgy (gleafmas, bleafmas,il1,il2,leapnow,tbar,thliq,wiltsm,fieldsm,ta,anveg,iday,radl, root-temp,rmatctem, stemmass, rootmass,sort,nol2pfts,fcancmx, isand,

9.50.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Phenology, Leaf Turnover & Mortality Subroutine.

The leaf phenology parametrization used in CTEM v. 1.0 is described in detail by [10]. Changes between version 1.0 and 2.0 are limited to parameter values and the parametrization is briefly described here. There are four different leaf phenological states in which vegetation can be at a given instant: (i) no leaves or dormant, (ii) maximum growth, (iii) normal growth and (iv) leaf fall or harvest. PFTs may go through only some, or all, of these phenological states depending on their deciduousness. A broadleaf cold deciduous tree, for example, transitions through all these four states in a year. In winter, the broadleaf cold deciduous trees are in the no leaves/dormant state; favourable climatic conditions in spring trigger leaf growth and the tree enters the maximum leaf growth state when all the NPP is allocated to leaves to accelerate leaf out; when the LAI reaches a threshold (described below) the tree enters the normal leaf growth state and NPP is also allocated to stem and root components; finally the arrival of autumn triggers leaf fall and the trees go into the leaf fall mode where no carbon is allocated to leaves (but it continues for roots and stems). When all the leaves have been shed, the trees go into the no leaves or dormant state again and the cycle is repeated the next year. The evergreen tree PFTs and the grass PFTs do not enter the leaf fall state and maintain a leaf canopy as long as environmental conditions are favourable. Although drought and cold stress cause accelerated leaf loss compared to the normal leaf turnover from these PFTs, they do not explicitly go into the leaf fall mode where the intent is to lose all leaves in a specified amount of time.

The leaf phenological state transitions are dependent upon environmental conditions. In particular, the transition from no leaves/dormant state to the maximum growth state is based on the carbon-gain approach. CTEM uses *virtual* leaves to assess favourable meteorological conditions for leaf out. The virtual leaves photosynthesize and respire in a manner similar to normal leaves except the carbon gain or loss is not taken into account in vegetation's carbon balance. A positive net leaf photosynthesis rate ($G_{canopy,net}$, Eq. Gnet}) for the virtual leaves over seven consecutive days indicates the arrival of favourable growth conditions and triggers leaf onset and the associated transition from the no leaves/dormant state to the maximum leaf growth state, when the entire positive NPP is allocated to leaves ($a_{fL} = 1$, $a_{fS} = a_{fR} = 0$). When LAI reaches LAI_{thrs} then the vegetation switches to the normal growth mode and positive NPP is allocated to all three vegetation components – leaves, stem and roots ($a_{fL}, a_{fS}, a_{fR} > 0$). LAI_{thrs} is calculated as

$$LAI_{thrs} = L_{\mathrm{f}} \left[SLA \left(\frac{C_{\mathrm{S}} + C_{\mathrm{R}}}{\eta} \right)^{1/\kappa} \right].$$

The PFT-specific L_{f} term (see also ctem_params.f90) calculates LAI_{thrs} to be typically between 40 and 50\,{%} of the maximum LAI that a given amount of stem and root biomass can support (based on the terms in the square brackets and Eq. (propwoody}). SLA is the specific leaf area (Eq. sla}). This rule for transition from a maximum to a normal growth state is also used for evergreen tree PFTs and grass PFTs. Similar to LAI_{thrs}, the LAI of virtual leaves is $7.5\,\%$ of the maximum LAI a given amount of root and stem biomass can support for tree and crop PFTs and $2.5\,\%$ for grass PFTs. In addition, the LAI of virtual leaves is constrained to be, at least, $0.3\,m^2\,m^{-2}$ for tree PFTs and $0.2\,m^2\,m^{-2}$ for crop and grass PFTs.

The transition from the normal growth state to the leaf fall state is triggered by unfavourable environmental conditions and shorter day length. Broadleaf deciduous trees transition to the leaf fall state when either: (i) day length is less than $11\,h$ and the rooting zone temperature drops below $11.15\,C$ or (ii) when the rooting zone temperature drops below $8\,C$ regardless of the day length. Needleleaf deciduous tress begin leaf fall after seven consecutive days with daily mean air temperature below $-5\,C$. Leaf fall occurs over a period of 15 days. In the leaf fall state, the vegetation continues carbon allocation to its root and stem components, but not to leaves ($a_{fL} = 0$, $a_{fS} + a_{fR} = 1$). Evergreen trees and grasses do not enter the leaf fall state and neither do the broadleaf drought deciduous trees. The implication for the latter PFT is that if the climate changes and the dry season becomes shorter, then the trees will keep their leaves on for a longer period of time since broadleaf drought deciduous trees lose leaves due to soil moisture stress (described below).

The model vegetation is able to transition between the different leaf phenological states in response to changing conditions. For example, a leaf out in spring for broadleaf cold deciduous trees can be interrupted by a cold event when the vegetation goes into a leaf fall state until the return of more favourable conditions.

Leaf litter generation is caused by normal turnover of leaves (Ω_{N}, day^{-1}) and also due to cold (Ω_{C}, day^{-1}) and drought (Ω_{D}, day^{-1}) stress, both of which contribute to seasonality of LAI. For example, the leaf loss associated with drought and reduced photosynthesis during the dry season are the principal causes of the seasonality of LAI for the broadleaf drought deciduous tree PFT.

The conversion of leaf carbon to leaf litter ($D_{\rm L}$, $kg\,C\,m^{-2}\,day^{-1}$) is expressed as

$$D_{\rm L} = C_{\rm L}[1 - \exp(-\Omega_{\rm N} - \Omega_{\rm C} - \Omega_{\rm D})],$$

where ($\Omega_{N,C,D}$, day^{-1}) are the leaf loss rates associated with normal turnover of leaves and the cold and drought stress. The rate of normal turnover of leaves is governed by PFT-specific leaf lifespan ($\tau_{\rm L}$, yr) as $\Omega_{\rm N} = 1/365\tau_{\rm L}$ (see also ctem_params.f90} for PFT specific values of $\tau_{\rm L}$). The leaf loss rate associated with cold stress ($\Omega_{\rm C}$) is calculated as

$$\Omega_{\rm C} = \Omega_{C,max} L_{cold}^3,$$

where $\Omega_{C,max}$ (day^{-1}, see also ctem_params.f90) is the maximum cold stress loss rate. L_{cold} is a scalar that varies between 0 and 1 as

$$L_{cold} = \{\,1\,, \quad T_{\rm a} < \left(T_{cold}^{leaf} - 5\right) 1 - \frac{T_{\rm a} - \left(T_{cold}^{leaf} - 5\right)}{5}, \quad T_{cold}^{leaf} > T_{\rm a} > (T_{cold}^{leaf} - 5)0, \quad T_{\rm a} > T_{cold}^{leaf},$$

where T_{cold}^{leaf} is a PFT-specific temperature threshold below which a PFT experiences damage to its leaves promoting leaf loss (see also ctem_params.f90) and $T_{\rm a}$ is the daily mean air temperature (C). The leaf loss rate due to drought stress is calculated in a similar manner

$$\Omega_{\rm D} = \Omega_{{\rm D},max}\,(1 - \phi_{root})^3,$$

where $\Omega_{{\rm D},max}$ (day^{-1}, see also ctem_params.f90) is the maximum drought stress loss rate and ϕ_{root} (Eq. degsoilsat}) is the degree of soil saturation in the rooting zone.

9.50.2 Function/Subroutine Documentation

9.50.2.1 subroutine phenolgy (real, dimension(ilg,icc) *gleafmas,* real, dimension(ilg,icc) *bleafmas,* integer *il1,* integer *il2,* logical *leapnow,* real, dimension(ilg,ignd) *tbar,* real, dimension(ilg,ignd) *thliq,* real, dimension(ilg,ignd) *wiltsm,* real, dimension(ilg,ignd) *fieldsm,* real, dimension(ilg) *ta,* real, dimension(ilg,icc) *anveg,* integer *iday,* real, dimension(ilg) *radl,* real, dimension(ilg,icc) *roottemp,* real, dimension(ilg,icc,ignd) *rmatctem,* real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* integer, dimension(icc) *sort,* integer, dimension(ican) *nol2pfts,* real, dimension(ilg,icc) *fcancmx,* integer, dimension(ilg,ignd) *isand*)

Parameters

il1	il1=1
il2	il2=ilg
iday	day of year
leapnow	true if this year is a leap year. Only used if the switch 'leap' is true.
sort	index for correspondence between 9 pfts and the 12 values in parameters vectors
nol2pfts	number of level 2 ctem pfts
gleafmas	green or live leaf mass in kgc/m^2, for the 9 pfts
bleafmas	brown or dead leaf mass in kgc/m^2, for the 9 pfts
ta	air temperature, k
tbar	soil temperature, k
thliq	liquid soil moisture content in 3 soil layers

anveg	net photosynthesis rate of ctem's pfts, umol co2/m2.s
roottemp	root temperature, which is a function of soil temperature of course, k.
rmatctem	fraction of roots in each soil layer for each pft
stemmass	stem mass for each of the 9 ctem pfts, kgc/m^2
rootmass	root mass for each of the 9 ctem pfts, kgc/m^2
fcancmx	max. fractional coverage of ctem's 9 pfts, but this can be modified by land-use change, and competition between pfts
fieldsm	field capacity soil moisture content both calculated in allocate subroutine
wiltsm	wilting point soil moisture content
radl	latitude in radians

Constants and parameters are located in ctem_params.f90

initialize required arrays to zero

initialization ends

convert green leaf mass into leaf area index using specific leaf area $(sla, m^2/kgc)$ estimated using leaf life span. see bio2str subroutine for more details.

also find threshold lai as a function of stem+root biomass which is used to determine leaf status

using green leaf area index (ailcg) determine the leaf status for each pft. loops 190 and 200 thus initialize lfstatus, if this this information is not passed specifically as an initialization quantity.

knowing lfstatus (after initialization above or using value from from previous time step) we decide if we stay in a given leaf mode or we move to some other mode.

we start with the "no leaves" mode

add one to pandays(i,j) if daily an is positive, otherwise set it to zero.

if in "no leaves" mode check if an has been positive over last dayschk(j) days to move into "max. growth" mode. if not we stay in "no leaves" mode. also set the chkmode(i,j) switch to 1.

find day length using day of year and latitude. this is to be used for initiating leaf offset for broad leaf dcd trees.

even if pandays criteria has been satisfied do not go into max. growth mode if environmental conditions are such that they will force leaf fall or harvest mode.

needle leaf dcd

broad leaf dcd cld & dry

crops

similar to the way we count no. of days when an is positive, we find no. of days when temperature is below -5 c. we need this to determine if we go into "leaf fall" mode for needle leaf dcd trees.

also estimate no. of days below 8 c. we use these days to decide if its cold enough to harvest crops.

if in "max growth" mode

if mode hasn't been checked and we are in "max. growth" mode, then check if we are above pft-dependent lai threshold. if lai is more then this threshold we move into "normal growth" mode, otherwise we stay in "max growth" mode so that leaves can grow at their max. climate-dependent rate

for dcd trees we also need to go into "leaf fall" mode directly from "max. growth" mode.

ndl dcd

bdl dcd cold

bdl dcd dry

if in "normal growth" mode

if in "normal growth" mode then go through every pft individually and follow set of rules to determine if we go into "fall/harvest" mode

needle leaf evg

needle leaf dcd

broad leaf evg

broad leaf dcd cold we use daylength and roottemp to initiate leaf offset

broad leaf dcd dry we still use daylength and roottemp to initiate leaf offset, for the pathological cases of dry dcd trees being further away from the equator then we can imagine. other wise leaf loss will occur due to drought anyway.

"normal growth" to "fall/harvest" transition for crops is based on specified lai. we harvest if lai of crops reaches a threshold. if lai doesn't reach this threshold (say due to a bad year) we harvest anyway if it starts getting cold, otherwise we don't harvest.

"normal growth" to "max. growth" transition for grasses

if in "fall/harvest" mode

grasses and evg trees do not come into this mode, because they want to stay green if possible. this mode is activated for dcd plants and crops. once in this mode dcd trees loose their leaves and crops are harvested. ndl dcd trees keep loosing their leaves at rate determined by cold stress, bdl dcd trees loose their leaves at a specified rate, and crops are harvested over a period of 15 days. dcd trees and crops stay in "leaf fall/harvest" model until all green leaves are gone at which time they switch into "no leaves" mode, and then wait for the climate to become favourable to go into "max. growth" mode

check that leaf status of all vegetation types in all grid cells has been updated

having decided leaf status for every pft, we now calculate normal leaf turnover, cold and drought stress mortality, and for bdl dcd plants we also calculate specified loss rate if they are in "leaf fall" mode, and for crops we calculate harvest loss, if they are in "harvest" mode.

all these loss calculations will yield leaf litter in kgc/m^2 for the given day for all pfts

normal leaf turn over

for drought stress related mortality we need field capacity and wilting point soil moisture contents, which we calculated in allocate subroutine

estimate drought stress term averaged over the rooting depth for each pft

```
using this drought stress term and our two vegetation-dependent
parameters we find leaf loss rate associated with drought
```

estimate leaf loss in kgc/m^2 due to drought stress

similar to drgtstrs we find coldstrs for each pft. we assume that max. cold stress related leaf loss occurs when temperature is 5 c or more below pft's threshold

using this cold stress term and our two vegetation-dependent parameters we find leaf loss rate associated with cold cold related leaf loss rate

estimate leaf loss in kgc/m^2 due to cold stress

now that we have all types of leaf losses (due to normal turnover, cold and drought stress, and fall/harvest) we take the losses for grasses and use those to turn live green grass into dead brown grass. we then find the leaf litter from the brown grass which will then go into the litter pool.

we assume life span of brown grass is 10% that of green grass but this is an adjustable parameter.

combine nrmlloss, drgtloss, and coldloss together to get total leaf litter generated, which is what we use to update gleafmass, except for grasses, for which we have alerady updated gleafmass.

9.51 PHTSYN3.f File Reference

CANADIAN TERRESTRIAL ECOSYSTEM MODEL (CTEM) PHOTOSYNTHESIS SUBROUTINE.

Functions/Subroutines

- subroutine phtsyn3 (AILCG,FCANC, TCAN, CO2CONC,PRESSG,FC,CFLUX,QA, QSWV,IC,THLIQ, ISAN↩ D,TA,RMAT,COSZS, XDIFFUS,ILG,IL1,IL2,IG,ICC,ISNOW, SLAI,THFC, THLW, FCANCMX,L2MAX, NOL2↩ PFTS,

9.51.1 Detailed Description

CANADIAN TERRESTRIAL ECOSYSTEM MODEL (CTEM) PHOTOSYNTHESIS SUBROUTINE.

Photosynthesis and canopy conductance

Net photosynthesis

All biogeochemical processes in CTEM are simulated at a daily time step except gross photosynthetic uptake and associated calculation of canopy conductance, which are simulated on a half hour time step with CLASS. The photosynthesis module of CTEM calculates the net canopy photosynthesis rate, which, together with atmospheric CO_2 concentration and vapour pressure or relative humidity, is used to calculate canopy conductance. This canopy conductance is then used by CLASS in its energy and water balance calculations.

The photosynthesis parametrization is based upon the approach of [20] and [16] [15] as implemented in SiB2 [47] and MOSES [17] with some minor modifications as described in [7]. [7] outlines four possible configurations for the model based on choice of a *big-leaf* or *two-leaf* (sunlight and shaded leaves) mode and stomatal conductance formulations based on either [12] or [31]. The [12] formulation uses relative humidity while [31] uses vapour pressure deficit in calculation of canopy conductance. While the model remains capable of all four possible configurations, in practice, the model is usually run using the big-leaf parametrization with the stomatal conductance formulation of [31], which is the configuration described here. The original description of the CTEM photosynthesis parametrization in [7] did not include discussion of all the PFTs simulated by CTEM, which we expand upon here and also include changes to the parametrization since version 1.0.

The gross leaf photosynthesis rate, G_o, depends upon the maximum assimilation rate allowed by the light (J_{e}), Rubisco (J_c) and transport capacity (J_s). The limitation placed on G_o by the amount of available light is calculated as ($mol\, CO_2\, m^{-2}\, s^{-1}$)

$$J_e = \{ \varepsilon\ (1-\nu) I \left[\frac{c_i - \Gamma}{c_i + 2\Gamma} \right], \quad C_3 plants \varepsilon\, (1-\nu) I, \quad C_4 plants,$$

where I is the incident photosynthetically active radiation (PAR; $mol\, photons\, m^{-2}\, s^{-1}$), ν is the leaf scattering coefficient, with values of 0.15 and 0.17 for C_3 and C_4 plants, respectively, and ε is the quantum efficiency ($mol\, CO_2\, (mol\, photons)^{-1}$; values of 0.08 and 0.04 are used for C_3 and C_4 plants, respectively). c_i is the partial pressure of CO_2 in the leaf interior (Pa) and Γ is the CO_2 compensation point (Pa) (described below).

The Rubisco enzyme limited photosynthesis rate, J_c, is given by

$$J_c = \{ V_m \left[\frac{c_i - \Gamma}{c_i + K_c(1 + O_a/K_o)} \right], \quad C_3 plants V_m, C_4 plants,$$

where V_m is the maximum catalytic capacity of Rubisco ($mol\, CO_2\, m^{-2}\, s^{-1}$), adjusted for temperature and soil moisture, as described below. K_o and K_c are the Michaelis–Menten constants for O_2 and CO_2, respectively. O_a is the partial pressure (Pa) of oxygen.

The transport capacity (J_s) limitation determines the maximum capacity to transport the products of photosynthesis for C_3 plants, while for C_4 plants it represents CO_2 limitation

$$J_s = \begin{cases} 0.5 V_m, C_3 plants \\ 2 \times 10^4\, V_m \frac{c_i}{p}, C_4 plants \end{cases}$$

where p is surface atmospheric pressure (Pa).

$V_{\rm m}$ is calculated as

$$V_{\rm m} = \frac{V_{max} f_{25}(2.0) S_{root}(\theta) \times 10^{-6}}{[1+\exp 0.3(T_{\rm c} - T_{high})][1+\exp 0.3(T_{low} - T_{\rm c})]},$$

where $T_{\rm c}$ is the canopy temperature (C) and T_{low} and T_{high} are PFT-dependent lower and upper temperature limits for photosynthesis (see also ctem_params.f90). f_{25} is the standard Q_{10} function at $25\,C$ ($(f_{25}(Q_{10}) = Q_{10}^{(0.1(T_{\rm c}-25))})$ and V_{max} is the PFT-dependent maximum rate of carboxylation by the enzyme Rubisco ($mol\,CO_2\,m^{-2}\,s^{-1}$; see also ctem_params.f90). The constant 10^{-6} converts V_{max} from units of $\mu mol\,CO_2\,m^{-2}\,s^{-1}$ to $mol\,CO_2\,m^{-2}\,s^{-1}$.

The influence of soil moisture stress is simulated via $S_{root}(\theta)$, which represents a soil moisture stress term formulated as

$$S_{root}(\theta) = \sum_{i=1}^{g} S(\theta_i) r_i,$$

$$S(\theta_i) = [1 - \{1 - \phi_i\}]^{\varrho},$$

where $S_{root}(\theta)$ is calculated by weighting $S(\theta_i)$ with the fraction of roots, r_i, in each soil layer i and ϱ is a PF↩ T-specific sensitivity to soil moisture stress (unitless; see also ctem_params.f90). ϕ_i is the degree of soil saturation (soil wetness) given by

$$\phi_i(\theta_i) = \max\left[0, \min\left(1, \frac{\theta_i - \theta_{i,wilt}}{\theta_{i,field} - \theta_{i,wilt}}\right)\right],$$

where θ_i is the volumetric soil moisture ($m^3 water\,(m^3 soil)^{-1}$) of the ith soil layer and $\theta_{i,field}$ and $\theta_{i,wilt}$ the soil moisture at field capacity and wilting point, respectively.

The CO_2 compensation point (Γ) is the CO_2 partial pressure where photosynthetic uptake equals the leaf respiratory losses (used in Eqs. J_e} and J_c}). Γ is zero for C_4 plants but is sensitive to oxygen partial pressure for C_3 plants as

$$\Gamma = \begin{cases} \frac{O_{\rm a}}{2\sigma}, C_3 plants \\ 0, C_4 plants, \end{cases}$$

where σ is the selectivity of Rubisco for CO_2 over O_2 (unitless), estimated by $\sigma = 2600 f_{25}(0.57)$. The CO_2 ($K_{\rm c}$) and O_2 ($K_{\rm o}$) Michaelis–Menten constants used in Eq. (J_c}) are determined via

$$K_{\rm c} = 30 f_{25}(2.1), K_o = 3 \times 10^4 f_{25}(1.2).$$

Given the light ($J_{\rm e}$), Rubsico ($J_{\rm c}$) and transportation capacity ($J_{\rm s}$) limiting rates, the leaf-level gross photosynthesis rate, $G_{\rm o}$ ($mol\,CO_2\,m^{-2}\,s^{-1}$), is then determined following a minimization based upon smallest roots of the following two quadratic equations

$$J_{\rm p} = \frac{(J_{\rm c}+J_{\rm e}) \pm \sqrt{(J_{\rm c}+J_{\rm e})^2 - 4\beta_1(J_{\rm c}+J_{\rm e})}}{2\beta_1}, G_{\rm o} = \frac{(J_{\rm p}+J_{\rm s}) \pm \sqrt{(J_{\rm p}+J_{\rm s})^2 - 4\beta_2(J_{\rm p}+J_{\rm s})}}{2\beta_2},$$

where β_1 is 0.95 and β_2 is 0.99. When soil moisture stress is occurring, both the $J_{\rm s}$ and $J_{\rm c}$ terms are reduced since the $V_{\rm m}$ term (Eq. V_m}) includes the effect of soil moisture stress through the $S(\theta)$ term and this reduces the leaf-level gross photosynthesis rate.

The current version of CTEM does not include nutrient constraints on photosynthesis and, as a result, increasing atmospheric CO_2 concentration leads to unconstrained increase in photosynthesis. In natural ecosystems, however, down regulation of photosynthesis occurs due to constraints imposed by availability of nitrogen, as well as phosphorus. To capture this effect, CTEM uses a nutrient limitation term, based on experimental plant growth studies, to down regulate the photosynthetic response to elevated CO_2 concentrations [5]. The parametrization, and its rationale, are fully described in [5] but the basic relations are summarized here. The leaf-level gross photosynthetic rate is scaled by the down-regulation term, $\Xi_{\rm N}$, to yield the nutrient limited leaf level gross photosynthetic rate as

$$G_{{\rm o},N-limited} = \Xi_{\rm N} G_{\rm o}, \Xi_{\rm N} = \frac{1+\gamma_{gd}\ln(c_{\rm a}/c_0)}{1+\gamma_g \ln(c_{\rm a}/c_0)},$$

where c_a is the atmospheric CO_2 concentration in ppm, c_0 is the pre-industrial CO_2 concentration ($285.0\,ppm$), γ_g is 0.95 [5]. A value of γ_{gd} lower than γ_g ensures that Ξ_N gradually decreases from its pre-industrial value of one as c_a increases to constrain the rate of increase of photosynthesis with rising atmospheric CO_2 concentrations. CTEM v. 2.0 uses a γ_{gd} value of 0.30 (unitless).

Finally, the leaf-level gross photosynthesis rate, $G_{o,N-limited}$ is scaled up to the canopy-level, G_{canopy}, by considering the exponential vertical profile of radiation along the depth of the canopy as

$$G_{canopy} = G_{o,N-limited} f_{PAR}, f_{PAR} = \frac{1}{k_n}(1 - \exp^{-k_n LAI}),$$

which yields the gross primary productivity (G_{canopy}, GPP). k_n is the extinction coefficient that describes the nitrogen and time-mean photosynthetically absorbed radiation (PAR) profile along the depth of the canopy (see also ctem_params.f90) [23] [21], and LAI ($m^2\,leaf\,(m^2\,ground)^{-1}$) is the leaf area index.

The net canopy photosynthetic rate, $G_{canopy,net}$ ($mol\,CO_2\,m^{-2}\,s^{-1}$), is calculated by subtracting canopy leaf maintenance respiration costs (R_{mL}; see Sect. maint}) as

$$G_{canopy,net} = G_{canopy} - R_{mL}.$$

Coupling of photosynthesis and canopy conductance

When using the [31] approach for photosynthesis–canopy conductance coupling, canopy conductance (g_c; $mol\,m^{-2}\,s^{-1}$) is expressed as a function of the net canopy photosynthesis rate, $G_{canopy,net}$, as

$$g_c = m\frac{G_{canopy,net}p}{(c_s - \Gamma)}\frac{1}{(1 + V/V_o)} + bLAI$$

where p is the surface atmospheric pressure (Pa), the parameter m is set to 9.0 for needle-leaved trees, 12.0 for other C_3 plants and 6.0 for C_4 plants, parameter b is assigned the values of 0.01 and 0.04 for C_3 and C_4 plants, respectively. V is the vapour pressure deficit (Pa) and the parameter V_o is set to $2000\,Pa$ for trees and $1500\,Pa$ for crops and grasses. The partial pressure of CO_2 at the leaf surface, c_s, is found via

$$c_s = c_{ap} - \frac{1.37 G_{canopy,net} p}{g_b}.$$

Here, c_{ap} is the atmospheric CO_2 partial pressure (Pa) and g_b is the aerodynamic conductance estimated by CLASS ($mol\,m^{-2}\,s^{-1}$). The intra-cellular CO_2 concentration required in Eqs. (J_e})–(J_s}) is calculated as

$$c_i = c_s - \frac{1.65 G_{canopy,net} p}{g_c}.$$

Since calculations of $G_{canopy,net}$ and c_i depend on each other, the photosynthesis-canopy conductance equations need to be solved iteratively. The initial value of c_i used in calculation of $G_{canopy,net}$ is the value from the previous time step or, in its absence, c_i is assumed to be $0.7 c_{ap}$.

Canopy (g_c) and aerodynamic (g_b) conductance used in above calculations are expressed in units of $mol\,CO_2\,m^{-2}\,s^{-1}$ but can be converted to the traditional units of $m\,s^{-1}$ as follows

$$g_c(m\,s^{-1}) = 0.0224\,\frac{T_c}{T_f}\,\frac{p_0}{p}\,g_c(mol\,m^{-2}\,s^{-1}),$$

where p_0 is the standard atmospheric pressure ($101\,325\,Pa$) and T_f is freezing temperature ($273.16\,K$).

```
1.  SINGLE-LEAF & TWO-LEAF COMBINED VERSION, CAN USE EITHER APPROACH
2.  CAN USE EITHER BWB OR LEUNING TYPE STOMATAL CONDUCTANCE FORMULATION
3.  ALSO, CAN USE SMOOTHED AVERAGE OF THE 3 LIMITING RATES, MIN. OF
    THE 3 LIMITING RATES, OR MIN. OF LIGHT AND RUBSICO RATES.

CLASS' 4 MAJOR VEGETATION TYPES ARE

1. NEEDLE LEAF OR TALL CONIFEROUS (C3, DECIDUOUS AND EVERGREEN)
2. BROAD LEAF (C3, DECD. AND EVRG.)
3. ARABLE & CROPS - (BOTH C3 AND C4)
```

```
 4. GRASSES, TUNDRA, ETC. (BOTH C3 AND C4)

 BUT FOR PHOTOSYNTHESIS WE NEED TO MAKE DISTINCTION BETWEEN C3 AND
 C4, AND DECIDUOUS AND EVERGREEN. SO THESE 4 VEGETATION TYPES GET
 CONVERTED INTO THE FOLLOWING 9

 1. NEEDLE LEAF EVERGREEN, C3
 2. NEEDLE LEAF DECIDUOUS, C3
 3. BROAD LEAF EVERGREEN, C3
 4. BROAD LEAF COLD DECIDUOUS, C3
 5. BROAD LEAF DRY DECIDUOUS, C3
 6. C3 CROP
 7. C4 CROP
 8. C3 GREEN GRASS
 9. C4 GREEN GRASS

 INPUTS

 NOL2MAX   - NUMBER OF LEVEL 2 CTEM PFTs
```

9.51.2 Function/Subroutine Documentation

9.51.2.1 subroutine phtsyn3 (real, dimension(ilg,icc) *AILCG,* real, dimension(ilg,icc) *FCANC,* real, dimension(ilg) *TCAN,* real, dimension(ilg) *CO2CONC,* real, dimension(ilg) *PRESSG,* real, dimension(ilg) *FC,* real, dimension(ilg) *CFLUX,* real, dimension(ilg) *QA,* real, dimension(ilg) *QSWV,* integer *IC,* real, dimension(ilg,ig) *THLIQ,* integer, dimension(ilg,ig) *ISAND,* real, dimension(ilg) *TA,* real, dimension(ilg,icc,ig) *RMAT,* real, dimension(ilg) *COSZS,* real, dimension(ilg) *XDIFFUS,* integer *ILG,* integer *IL1,* integer *IL2,* integer *IG,* integer *ICC,* integer *ISNOW,* real, dimension(ilg,icc) *SLAI,* real, dimension(ilg,ig) *THFC,* real, dimension(ilg,ig) *THLW,* real, dimension(ilg,icc) *FCANCMX,* integer *L2MAX,* integer, dimension(ic) *NOL2PFTS*)

Parameters

ilg	NO. OF GRID CELLS IN LATITUDE CIRCLE
ic	NO. OF CLASS VEGETATION TYPES, 4
il1	IL1=1
il2	IL2=ILG
ig	NO. OF SOIL LAYERS, 3
icc	NO. OF CTEM's PFTs, CURRENTLY 9
isnow	INTEGER (0 or 1) TELLING IF PHTSYN IS TO BE RUN OVER CANOPY OVER SNOW OR CANOPY OVER GROUND SUBAREA
l2max	MAX. NUMBER OF LEVEL 2 PFTs
fcanc	FRACTIONAL COVERAGE OF CTEM's 9 PFTs
ailcg	GREEN LEAF AREA INDEX FOR USE BY PHOTOSYNTHESIS, M^2/M^2
tcan	CANOPY TEMPERATURE, KELVIN
fc	SUM OF ALL FCANC OVER A GIVEN SUB-AREA
cflux	AERODYNAMIC CONDUCTANCE, M/S
slai	SCREEN LEVEL HUMIDITY IN KG/KG - STORAGE LAI. THIS LAI IS USED FOR PHTSYN EVEN IF ACTUAL LAI IS ZERO. ESTIMATE OF NET PHOTOSYNTHESIS BASED ON S↩ LAI IS USED FOR INITIATING LEAF ONSET. SEE PHENOLGY SUBROUTINE FOR MORE DETAILS.
co2conc	ATMOS. CO_2 IN PPM, AND THEN CONVERT IT TO PARTIAL PRESSURE, PASCALS, CO2A, FOR USE IN THIS SUBROUTINE
pressg	ATMOS. PRESSURE, PASCALS
qswv	ABSORBED VISIBLE PART OF SHORTWAVE RADIATION, W/M^2

ta	AIR TEMPERATURE IN KELVINS
rmat	FRACTION OF ROOTS IN EACH LAYER (grid cell, vegetation, layer)
thliq	LIQUID MOIS. CONTENT OF 3 SOIL LAYERS
thfc	SOIL FIELD CAPACITY.
thlw	SOIL WILT CAPACITY.
fcancmx	MAX. FRACTIONAL COVERAGES OF CTEM's 8 PFTs. THIS IS DIFFERENT FROM FCANC AND FCANCS (WHICH MAY VARY WITH SNOW DEPTH). FCANCMX DOESN'T CHANGE, UNLESS OF COURSE ITS CHANGED BY LAND USE CHANGE OR DYNAMIC VEGETATION.
coszs	COS OF ZENITH ANGLE
xdiffus	FRACTION OF DIFFUSED PAR
isand	SAND INDEX.

IF LAI IS LESS THAN SLAI THAN WE USE STORAGE LAI TO PHOTOSYNTHESIZE. HOWEVER, WE DO NOT USE THE STOMATAL RESISTANCE ESTIMATED IN THIS CASE, BECAUSE STORAGE LAI IS AN IMAGINARY LAI, AND WE SET STOMATAL RESISTANCE TO ITS MAX. NOTE THAT THE CONCEPT OF STORAGE/IMAGINARY LAI IS USED FOR PHENOLOGY PURPOSES AND THIS IMAGINARY LAI ACTS AS MODEL SEEDS.

SET MIN. AND MAX. VALUES FOR STOMATAL CONDUCTANCE. WE MAKE SURE THAT MAX. STOMATAL RESISTANCE IS AROUND 5000 S/M AND MIN. STOMATAL RESISTANCE IS 51 S/M.

IF WE ARE USING LEUNING TYPE PHOTOSYNTHESIS-STOMATAL CONDUCTANCE COUPLING WE NEED VAPOR PRESSURE DEFICIT AS WELL. CALCULATE THIS FROM THE RH AND AIR TEMPERATURE WE HAVE. WE FIND E_SAT, E, AND VPD IN PASCALS.

ESTIMATE PARTIAL PRESSURE OF CO_2 AND IPAR

CONVERT CO2CONC FROM PPM TO PASCALS

CHANGE PAR FROM W/M^2 TO MOL/M^2.S

SUNLIT PART GETS BOTH DIRECT AND DIFFUSED, WHILE THE SHADED PART GETS ONLY DIFFUSED

FOR TWO-LEAF MODEL FIND Kb AS A FUNCTION OF COSZS AND LEAF ANGLE DISTRIBUTION (VEGETATION DEPENDENT)

MAKE SURE -0.4 $<$ CHI $<$ 0.6

MAKE VALUES CLOSE TO ZERO EQUAL TO 0.01

ALSO FIND SUNLIT AND SHADED LAI

FOLLOWING FEW LINES TO MAKE SURE THAT ALL LEAVES ARE SHADED WHEN XDIFFUS EQUALS 1. NOT DOING SO GIVES ERRATIC RESULTS WHEN TWO LEAF OPTION IS USED

FIND FPAR - FACTOR FOR SCALING PHOTOSYNTHESIS TO CANOPY BASED ON ASSUMPTION THAT NITROGEN IS OPTIMALLY DISTRIBUTED. THE TWO-LEAF MODEL IS NOT THAT DIFFERENT FROM THE SINGLE-LEAF MODEL. ALL WE DO IS USE TWO SCALING FACTORS (I.E. SCALING FROM LEAF TO CANOPY) INSTEAD OF ONE, AND THUS PERFORM CALCULATIONS TWICE, AND IN THE END ADD CONDUCTANCE AND NET PHOTOSYNTHESIS FROM THE TWO LEAVES TO GET THE TOTAL.

IF ALL RADIATION IS DIFFUSED, THEN ALL LEAVES ARE SHADED, AND WE ADJUST FPARs ACCORDINGLY. WITHOUT THIS THE TWO LEAF MODELS MAY BEHAVE ERRATICALLY

FIND Vmax,canopy, THAT IS Vmax SCALED BY LAI FOR THE SINGLE LEAF MODEL

-———— Changing Vcmax seasonally ————————

Based on [13] and [1] there is good evidence for the Vcmax varying throughout the season for deciduous tree species. We are adopting a parameterization based upon their paper with some differences. We don't apply it to evergreens like they suggest. Their paper had only one evergreen species and other papers ([40]) don't seem to back that up. Grasses and crops are also not affected by the dayl. [1] seems to indicate that all PFTs except BDL-EVG tropical should vary intra-annually (see their figure 8).

The two leaf is assumed to be affect by the insolation seasonal cycle the same for each sun/shade leaf

-———— Changing Vcmax seasonally ————————///

FIND Vm,unstressed (DUE TO WATER) BUT STRESSED DUE TO TEMPERATURE

ASSUMING THAT SUNLIT AND SHADED TEMPERATURES ARE SAME

CALCULATE SOIL MOIS STRESS TO ACCOUNT FOR REDUCTION IN PHOTOSYN DUE TO LOW SOIL MOI↩ STURE, THREE STEPS HERE -> 1. FIND WILTING POINT AND FIELD CAPACITY SOIL MOIS. CONTENT FOR ALL THREE LAYERS.

1. USING THESE FIND THE SOIL MOISTURE STRESS TERM FOR ALL THREE LAYERS, AND 3. AV↩ ERAGE THIS SOIL MOISTURE STRESS TERM OVER THE 3 LAYERS USING FRACTION OF ROOTS PRESENT IN EACH LAYER FOR EACH PFT. NOTE THAT WHILE SOIL MOISTURE IS UNIFORM OVER AN ENTIRE GCM GRID CELL, THE SOIL MOISTURE STRESS FOR EACH PFT IS NOT BECAUSE OF DIFFERENCES IN ROOT DISTRIBUTION.

WILTING POINT CORRESPONDS TO MATRIC POTENTIAL OF 150 M FIELD CAPACITY CORRESPONDS TO HYDARULIC CONDUCTIVITY OF 0.10 MM/DAY -> 1.157x1E-09 M/S

USE SOIL MOISTURE FUNCTION TO MAKE Vm,unstressed -> Vm STRESSED

FIND TEMPERATURE DEPENDENT PARAMETER VALUES

FIND RUBISCO SPECIFICITY FOR CO_2 RELATIVE TO O_2 - SIGMA

FIND CO_2 COMPENSATION POINT USING RUBISCO SPECIFICITY - TGAMMA. KEEP IN MIND THAT CO_2 COMPENSATION POINT FOR C4 PLANTS IS ZERO, SO THE FOLLOWING VALUE IS RELEVANT FOR C3 PLANTS ONLY

ESTIMATE MICHELIS-MENTON CONSTANTS FOR CO_2 (Kc) and O_2 (Ko) TO BE USED LATER FOR ESTIM↩ ATING RUBISCO LIMITED PHOTOSYNTHETIC RATE

CHOOSE A VALUE OF INTERCELLULAR CO_2 CONCENTRATION (CO_2i) IF STARTING FOR THE FIRST TI↩ ME, OR USE VALUE FROM THE PREVIOUS TIME STEP

ESTIMATE RUBISCO LIMITED PHOTOSYNTHETIC RATE

ESTIMATE PHOTOSYNTHETIC RATE LIMITED BY AVAILABLE LIGHT

ESTIMATE PHOTOSYNTHETIC RATE LIMITED BY TRANSPORT CAPACITY

INCLUDE NUTRIENT LIMITATION EFFECT BY DOWN-REGULATING PHOTOSYNTHESIS N_EFFECT DECR↩ EASES FROM 1.0 AS CO_2 INCREASES ABOVE 288 PPM.

LIMIT N_EFFECT TO MAX OF 1.0 SO THAT NO UP-REGULATION OCCURS

FIND THE SMOOTHED AVERAGE OF THREE PHOTOSYNTHETIC RATES JC, JE, AND JS USING COLLATZ'S TWO QUADRATIC EQUATIONS, OR FIND THE MIN. OF THIS TWO RATES OR FIND MIN. OF JC AND JE.

DOWN-REGULATE PHOTOSYNTHESIS FOR C3 PLANTS

ESTIMATE LEAF MAINTENANCE RESPIRATION RATES AND NET PHOTOSYNTHETIC RATE. THIS NET P↩ HOSYNTHETIC RATE IS /M^2 OF VEGETATED LAND.

RECENT STUDIES SHOW RmL IS LESS TEMPERATURE SENSITIVE THAN PHOTOSYNTHESIS DURING DAY, THAT'S WHY A SMALL Q10 VALUE IS USED DURING DAY.

FIND CO_2 CONCENTRATION AT LEAF SURFACE FOR ALL VEGETATION TYPES. ALTHOUGH WE ARE F↩ INDING CO_2 CONC AT THE LEAF SURFACE SEPARATELY FOR ALL VEGETATION TYPES, THE BIG ASS↩ UMPTION HERE IS THAT THE AERODYNAMIC CONDUCTANCE IS SAME OVER ALL VEGETATION TYPES. CLASS FINDS AERODYNAMIC RESISTANCE OVER ALL THE 4 SUB-AREAS, BUT NOT FOR DIFFERENT V↩ EGETATION TYPES WITHIN A SUB-AREA. ALSO CHANGE AERODYNAMIC CONDUCTANCE, CFLUX, FROM M/S TO $MOL/M^2/S$

FIND STOMATAL CONDUCTANCE AS PER BALL-WOODROW-BERRY FORMULATION USED BY COLLATZ ET AL. OR USE THE LEUNING TYPE FORMULATION WHICH USES VPD INSTEAD OF RH

IF LIGHT IS TOO LESS MAKE PARAMETER BB VERY SMALL

FIND THE INTERCELLULAR CO_2 CONCENTRATION BASED ON ESTIMATED VALUE OF GC

SEE IF WE HAVE PERFORMED THE REQUIRED NO. OF ITERATIONS, IF NOT THEN WE GO BACK AND DO

ANOTHER ITERATION

WHEN REQUIRED NO. OF ITERATIONS HAVE BEEN PERFORMED THEN FIND STOMATAL CONDUCTANC↩ ES FOR ALL VEGETATION TYPES IN M/S AND THEN USE CONDUCTANCES TO FIND RESISTANCES. GCTU IMPLIES GC IN TRADITIONAL UNITS OF M/S

DON'T WANT TO REDUCE RESISTANCE AT NIGHT TO LESS THAN OUR MAX. VALUE OF AROUND 5000 S/M

IF USING STORAGE LAI THEN WE SET STOMATAL RESISTANCE TO ITS MAXIMUM VALUE.

AND FINALLY TAKE WEIGHTED AVERAGE OF RC_VEG BASED ON FRACTIONAL COVERAGE OF OUR 4 VEGETATION TYPES

CONVERT AN_VEG AND RML_VEG TO u-MOL CO2/M2.SEC

9.52 read_from_job_options.f90 File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Joboptions Read-In Subroutine.

Functions/Subroutines

- subroutine read_from_job_options (argbuff, transient_run, trans_startyr, ctemloop, ctem_on, ncyear, lnduseon, spinfast, cyclemet, nummetcylyrs, metcylyrst, co2on, setco2conc, ch4on, setch4conc, popdon, popcycleyr, parallelrun, dofire, dowetlands, obswetf, compete, inibioclim, start_bare, rsfile, start_from_rs, leap, jmosty, idisp, izref, islfd, ipcp, itc, itcg, itg, iwf, ipai, ihgt, ialc, ials, ialg, isnoalb, igralb, jhhstd, jhhendd, jdstd, jdendd, jhhsty, jhhendy, jdsty, jdendy)

9.52.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Joboptions Read-In Subroutine.

EXAMPLES:

To set up a transient run that does a cycling of the climate at the start:

In the example below the model will run from 1851 - 2012 using a climate dataset that spans 1901 - 2012. The LUC, POPD, CO2 files all span 1850 - 2012. The climate dataset will cycle over 25 years of climate (NUMMETCYLYRS; from 1901 - 1925) twice (CTEMLOOP) for the period 1851 - 1900 while the CO2, POPD, and LUC all run from 1851 - 1900 in their respective files (Each file will search for 1851 at the start so skipping 1850). Once 1901 is hit the MET file no longer cycles and simply runs, like the other input files, until it reaches its end (year 2012).

(only the relevant switches are shown below)
transient_run = .true.
trans_startyr = 1851
CTEMLOOP = 2 , <– note this is set to the number of loops that NUMMETCYLYRS must make to match up with the other datasets.
NCYEAR = 112 ,
LNDUSEON = .TRUE. ,
CYCLEMET = .TRUE. , <– note this is set to TRUE.
NUMMETCYLYRS = 25 , <– note this times ctemloop should allow the datasets to match up (e.g. 1850 + 2∗25 yrs ends in 1900)
METCYLYRST = 1901 ,
CO2ON = .TRUE. ,
POPDON = .TRUE. ,
OBSWETF = .false. ,

If you are doing methane then this would be true like the CO2 switches

If you want a transient run that does not spin over climate at the start, you need to change from above the following:

(only the relevant switches are shown below)

trans_startyr = 2000, <– whatever year you want to start at
CYCLEMET = .FALSE. , <– note this is set to FALSE.

9.52.2 Function/Subroutine Documentation

**9.52.2.1 subroutine read_from_job_options (character(80), intent(out) *argbuff,* logical, intent(out) *transient_run,* integer, intent(out) *trans_startyr,* integer, intent(out) *ctemloop,* logical, intent(out) *ctem_on,* integer, intent(out) *ncyear,* logical, intent(out) *lnduseon,* integer, intent(out) *spinfast,* logical, intent(out) *cyclemet,* integer, intent(out) *nummetcylyrs,* integer, intent(out) *metcylyrst,* logical, intent(out) *co2on,* real, intent(out) *setco2conc,* logical, intent(out) *ch4on,* real, intent(out) *setch4conc,* logical, intent(out) *popdon,* integer, intent(out) *popcycleyr,* logical, intent(out) *parallelrun,* logical, intent(out) *dofire,* logical, intent(out) *dowetlands,* logical, intent(out) *obswetf,* logical, intent(out) *compete,* logical, intent(out) *inibioclim,* logical, intent(out) *start_bare,* logical, intent(out) *rsfile,* logical, intent(out) *start_from_rs,* logical, intent(out) *leap,* integer, intent(out) *jmosty,* integer, intent(out) *idisp,* integer, intent(out) *izref,* integer, intent(out) *islfd,* integer, intent(out) *ipcp,* integer, intent(out) *itc,* integer, intent(out) *itcg,* integer, intent(out) *itg,* integer, intent(out) *iwf,* integer, intent(out) *ipai,* integer, intent(out) *ihgt,* integer, intent(out) *ialc,* integer, intent(out) *ials,* integer, intent(out) *ialg,* integer, intent(out) *isnoalb,* integer, intent(out) *igralb,* integer, intent(out) *jhhstd,* integer, intent(out) *jhhendd,* integer, intent(out) *jdstd,* integer, intent(out) *jdendd,* integer, intent(out) *jhhsty,* integer, intent(out) *jhhendy,* integer, intent(out) *jdsty,* integer, intent(out) *jdendy*)

Parameters

out	*transient_run*	true if the run is a transient run. With this flag set set to .true., you can cycle over nummetcyclyrs of climate a ctemloop number of times then continue on through the climate the reason for this is to allow a transient from 1850 on, but only having the climate from 1901 while having the other inputs from 1850. See the bottom of this subroutine for an example of how to set this correctly.
out	*trans_startyr*	the year you want the transient run to start (e.g. 1850). If you are not doing a transient run, set to a negative value (like -9999)
out	*ctemloop*	no. of times the .met file is to be read. this option is useful to see how ctem's c pools equilibrate when driven with same climate data over and over again.
out	*ctem_on*	set this to true for using ctem simulated dynamic lai and canopy mass, else class simulated specified lai and canopy mass are used. with this switch on, all ctem subroutines are run.
out	*ncyear*	no. of years in the .met file.
out	*lnduseon*	set this to 1 if land use change is to be implimented by reading in the fractions of 9 ctem pfts from a file. keep in mind that once on, luc read-in is also influenced by the cyclemet and popcycleyr switches
out	*spinfast*	set this to a higher number up to 10 to spin up soil carbon pool faster
out	*cyclemet*	to cycle over only a fixed number of years (nummetcylyrs) starting at a certain year (metcylyrst) if cyclemet, then put co2on = false and set an appopriate setco2conc, also if popdon is true, it will choose the popn and luc data for year metcylyrst and cycle on that.
out	*nummetcylyrs*	years of the climate file to spin up on repeatedly ignored if cyclemet is false
out	*metcylyrst*	climate year to start the spin up on ignored if cyclemet is false
out	*co2on*	use co2 time series, set to false if cyclemet is true
out	*setco2conc*	set the value of atmospheric co2 if co2on is false. (ppmv)
out	*ch4on*	use CH4 time series, set to false if cyclemet is true the CO2 timeseries is in the same input file as the CO2 one.
out	*setch4conc*	set the value of atmospheric CH4 if ch4on is false. (ppmv)
out	*popdon*	if set true use population density data to calculate fire extinguishing probability and probability of fire due to human causes, or if false, read directly from .ctm file

out	*popcycleyr*	popd and luc year to cycle on when cyclemet is true, set to -9999 to cycle on metcylyrst for both popd and luc. if cyclemet is false this defaults to -9999, which will then cause the model to cycle on whatever is the first year in the popd and luc datasets
out	*parallelrun*	set this to be true if model is run in parallel mode for multiple grid cells, output is limited to monthly & yearly grid-mean only. else the run is in stand alone mode, in which output includes half-hourly and daily and mosaic-mean as well.
out	*dofire*	if true the fire/disturbance subroutine will be used.
out	*dowetlands*	if true the ch4wetland subroutine will be used.
out	*obswetf*	if true the observed wetland fraction will be used.
out	*compete*	set this to true if competition between pfts is to be implimented
out	*inibioclim*	set this to true if competition between pfts is to be implimented and you have the mean climate values in the ctm files.
out	*start_bare*	set this to true if competition is true, and if you wish to start from bare ground. if this is set to false, the ini and ctm file info will be used to set up the run. NOTE: This still keeps the crop fractions (while setting all pools to zero)
out	*rsfile*	set this to true if restart files (.ini_rs and .ctm_rs) are written at the end of each year. these files are necessary for checking whether the model reaches equilibrium after running for a certain years. set this to false if restart files are not needed (known how many years the model will run)
out	*start_from_rs*	if true, this option copies the _RS INI and CTM files to be the .INI and .CTM files and then starts the run as per normal. it is handy when spinning up so you don't have to do a complicated copying of the RS files to restart from them. NOTE! This will not work on hadar or spica, instead you have to manually move the files and set this to .false.
out	*leap*	set to true if all/some leap years in the .MET file have data for 366 days also accounts for leap years in .MET when cycling over meteorology (cyclemet)
out	*jmosty*	Year to start writing out the monthly output files. If you want to write monthly outputs right from the start then put in a negative number (like -9999), if you never want to have monthly outputs put a large positive number (like 9999). This is given in the same timescale as IYEAR
out	*idisp*	if idisp=0, vegetation displacement heights are ignored, because the atmospheric model considers these to be part of the "terrain". if idisp=1, vegetation displacement heights are calculated.
out	*izref*	if izref=1, the bottom of the atmospheric model is taken to lie at the ground surface. if izref=2, the bottom of the atmospheric model is taken to lie at the local roughness height.
out	*islfd*	if islfd=0, drcoef is called for surface stability corrections and the original gcm set of screen-level diagnostic calculations is done. if islfd=1, drcoef is called for surface stability corrections and sldiag is called for screen-level diagnostic calculations. if islfd=2, flxsurfz is called for surface stability corrections and diasurf is called for screen-level diagnostic calculations.
out	*ipcp*	if ipcp=1, the rainfall-snowfall cutoff is taken to lie at 0 c. if ipcp=2, a linear partitioning of precipitation betweeen rainfall and snowfall is done between 0 c and 2 c. if ipcp=3, rainfall and snowfall are partitioned according to a polynomial curve between 0 c and 6 c.
out	*iwf*	if iwf=0, only overland flow and baseflow are modelled, and the ground surface slope is not modelled. if iwf=n ($0<n<4$), the watflood calculations of overland flow and interflow are performed; interflow is drawn from the top n soil layers.
out	*itc*	itc, itcg and itg are switches to choose the iteration scheme to be used in calculating the canopy or ground surface temperature respectively. if the switch is set to 1, a bisection method is used; if to 2, the newton-raphson method is used.

out	*itcg*	itc, itcg and itg are switches to choose the iteration scheme to be used in calculating the canopy or ground surface temperature respectively. if the switch is set to 1, a bisection method is used; if to 2, the newton-raphson method is used.
out	*itg*	itc, itcg and itg are switches to choose the iteration scheme to be used in calculating the canopy or ground surface temperature respectively. if the switch is set to 1, a bisection method is used; if to 2, the newton-raphson method is used.
out	*ipai*	if ipai, ihgt, ialc, ials and ialg are zero, the values of plant area index, vegetation height, canopy albedo, snow albedo and soil albedo respectively calculated by class are used. if any of these switches is set to 1, the value of the corresponding parameter calculated by class is overridden by a user-supplied input value.
out	*ihgt*	if ipai, ihgt, ialc, ials and ialg are zero, the values of plant area index, vegetation height, canopy albedo, snow albedo and soil albedo respectively calculated by class are used. if any of these switches is set to 1, the value of the corresponding parameter calculated by class is overridden by a user-supplied input value.
out	*ialc*	if ipai, ihgt, ialc, ials and ialg are zero, the values of plant area index, vegetation height, canopy albedo, snow albedo and soil albedo respectively calculated by class are used. if any of these switches is set to 1, the value of the corresponding parameter calculated by class is overridden by a user-supplied input value.
out	*ials*	if ipai, ihgt, ialc, ials and ialg are zero, the values of plant area index, vegetation height, canopy albedo, snow albedo and soil albedo respectively calculated by class are used. if any of these switches is set to 1, the value of the corresponding parameter calculated by class is overridden by a user-supplied input value.
out	*ialg*	if ipai, ihgt, ialc, ials and ialg are zero, the values of plant area index, vegetation height, canopy albedo, snow albedo and soil albedo respectively calculated by class are used. if any of these switches is set to 1, the value of the corresponding parameter calculated by class is overridden by a user-supplied input value.
out	*jhhstd*	day of the year to start writing the half-hourly output
out	*jhhendd*	day of the year to stop writing the half-hourly output
out	*jdstd*	day of the year to start writing the daily output
out	*jdendd*	day of the year to stop writing the daily output
out	*jhhsty*	simulation year (iyear) to start writing the half-hourly output
out	*jhhendy*	simulation year (iyear) to stop writing the half-hourly output
out	*jdsty*	simulation year (iyear) to start writing the daily output
out	*jdendy*	simulation year (iyear) to stop writing the daily output
out	*isnoalb*	if isnoalb is set to 0, the original two-band snow albedo algorithms are used. if it is set to 1, the new four-band routines are used.
out	*igralb*	if igralb is set to 0, the wet and dry soil albedos are calculated on the basis of soil texture. if it is set to 1, they are assigned values based on the ncar clm soil "colour" dataset.

9.53 runclass36ctem.f File Reference

Principle driver program to run CLASS in stand-alone mode using specified boundary conditions and atmospheric forcing, coupled to CTEM.

Functions/Subroutines

- program **runclass36ctem**

- integer function **strlen** (ST)

9.53.1 Detailed Description

Principle driver program to run CLASS in stand-alone mode using specified boundary conditions and atmospheric forcing, coupled to CTEM.

Overview

This driver program initializes the run, reads in CLASS input files, manages the run and the coupling between CLASS and CTEM, writes the CLASS sub-monthly outputs, and closes the run.

9.54 SCREENRH.f File Reference

CALCULATES SCREEN RELATIVE HUMIDITY BASED ON INPUT SCREEN TEMPERATURE, SCREEN SPE↩ CIFIC HUMIDITY AND SURFACE PRESSURE. THE FORMULAE USED HERE ARE CONSISTENT WITH THAT USED ELSEWHERE IN THE GCM PHYSICS.

Functions/Subroutines

- subroutine **screenrh** (SRH, ST, SQ, PRESSG, FMASK, ILG, IL1, IL2)

9.54.1 Detailed Description

CALCULATES SCREEN RELATIVE HUMIDITY BASED ON INPUT SCREEN TEMPERATURE, SCREEN SPE↩ CIFIC HUMIDITY AND SURFACE PRESSURE. THE FORMULAE USED HERE ARE CONSISTENT WITH THAT USED ELSEWHERE IN THE GCM PHYSICS.

9.55 SLDIAG.f File Reference

CALCULATES NEAR SURFACE OUTPUT VARIABLES.

Functions/Subroutines

- subroutine sldiag (SUT, SVT, STT, SQT, CDM, CDH, UA, VA, TA, QA, T0, Q0, Z0M, Z0E, F, ZA, ZU, ZT, ILG, IL1, IL2, JL)

9.55.1 Detailed Description

CALCULATES NEAR SURFACE OUTPUT VARIABLES.

9.55.2 Function/Subroutine Documentation

9.55.2.1 subroutine sldiag (real, dimension(ilg) *SUT,* real, dimension(ilg) *SVT,* real, dimension(ilg) *STT,* real, dimension(ilg) *SQT,* real, dimension(ilg) *CDM,* real, dimension(ilg) *CDH,* real, dimension(ilg) *UA,* real, dimension(ilg) *VA,* real, dimension(ilg) *TA,* real, dimension(ilg) *QA,* real, dimension(ilg) *T0,* real, dimension(ilg) *Q0,* real, dimension(ilg) *Z0M,* real, dimension(ilg) *Z0E,* real, dimension(ilg) *F,* real, dimension(ilg) *ZA,* real, dimension(ilg) *ZU,* real, dimension(ilg) *ZT,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

ilg	NUMBER OF POINTS TO BE TREATED
sut	U COMPONENT OF THE WIND AT ZU
svt	V COMPONENT OF THE WIND AT ZU
stt	TEMPERATURE AT ZT
sqt	SPECIFIC HUMIDITY AT ZT
cdm	DRAG COEFFICIENT
cdh	TRASFER COEFFICIENT FOR HEAT AND MOISTURE
ua	U COMPONENT OF THE WIND AT ZA
va	V COMPONENT OF THE WIND AT ZA
ta	POTENTIAL TEMPERATURE AT ZA
qa	SPECIFIC HUMIDITY AT REFERENCE HEIGHT
z0m	ROUGHNESS LENGTH FOR MOMENTUM
z0e	ROUGHNESS LENGTH FOR HEAT AND MOISTURE
f	FRACTION OF GRID POINT AFFECTED BY PROCESS
t0	TEMPERATURE AT BOTTOM OF SURFACE LAYER
q0	SPECIFIC HUMIDITY AT BOTTOM OF SURFACE LAYER
za	TOP OF SURFACE LAYER
zu	HEIGHT OF OUTPUT WIND
zt	HEIGHT OF OUTPUT TEMPERATURE AND HUMIDITY

9.56 SNINFL.f File Reference

Purpose: Address infiltration of rain and meltwater into snow pack, and snow ripening.

Functions/Subroutines

- subroutine sninfl (R, TR, ZSNOW, TSNOW, RHOSNO, HCPSNO, WSNOW, HTCS, HMFN, PCPG, ROFN, FI, ILG, IL1, IL2, JL)

9.56.1 Detailed Description

Purpose: Address infiltration of rain and meltwater into snow pack, and snow ripening.

9.56.2 Function/Subroutine Documentation

9.56.2.1 subroutine sninfl (real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension(ilg) *HCPSNO,* real, dimension (ilg) *WSNOW,* real, dimension (ilg) *HTCS,* real, dimension (ilg) *HMFN,* real, dimension (ilg) *PCPG,* real, dimension (ilg) *ROFN,* real, dimension (ilg) *FI,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

r	Rainfall rate incident on snow pack $[ms^{-1}]$
tr	Temperature of rainfall [C]
zsnow	Depth of snow pack $[m](z_g)$
tsnow	Temperature of the snow pack $[C](T_s)$

rhosno	Density of snow pack $[kgm^{-3}](\rho_s)$
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}](C_s)$
wsnow	Liquid water content of snow pack $[kgm^{-2}](w_s)$
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}](I_s)$
hmfn	Energy associated with freezing or thawing of water in the snow pack $[Wm^{-2}]$
pcpg	Precipitation incident on ground $[kgm^{-2}s^{-1}]$
rofn	Runoff reaching the ground surface from the bottom of the snow pack $[kgm^{-2}s^{-1}]$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$

The rainfall rate, i.e. the liquid water precipitation rate incident on the snow pack from the atmosphere, from canopy drip and/or from melting of the top of the snow pack, may cause warming and/or melting of the snow pack as a whole. The overall change of internal energy of the snow pack as a result of the rainfall added to it, I_s or HTCS, is calculated as the difference in Is between the beginning and end of the subroutine:

$$\Delta I_s = X_i \Delta[C_s z_s T_s]/Deltat$$

where C_s represents the volumetric heat capacity of the snow pack, T_s its temperature, Δ the length of the time step, and X_i the fractional coverage of the subarea under consideration relative to the modelled area.

Four diagnostic variables are evaluated at the outset. HRCOOL, the energy sink required to cool the whole rainfall amount to 0 C, is calculated from the rainfall rate and temperature, and the heat capacity of water. HRFREZ, the energy sink required to freeze all the rainfall, is obtained from the latent heat of melting, the density of water and the rainfall rate. HSNWRM, the energy required to warm the whole snow pack to 0 C, is calculated from the temperature, heat capacity and depth of the snow pack. HSNMLT, the energy required to melt all the snow, is obtained from the latent heat of melting and the heat capacity and depth of the snow pack.

If HRCOOL $\geq$ (HSNWRM + HSNMLT), the energy contributed by the temperature of the rainfall is sufficient to warm to 0 C and melt the whole snow pack. HRCOOL is recalculated as the difference between HRCOOL and (HSNWRM + HSNMLT), and the snow depth is converted to a water depth ZMELT. The energy used to melt the snow is added to the diagnostic variables HMFN, representing the energy associated with water phase changes in the snow pack, and HTCS. The new temperature of the rainfall is calculated by applying HRCOOL over the new rainfall rate reaching the soil, which now includes the original rainfall rate, the melted snow pack and the liquid water that was contained in the snow pack. The depth, temperature, density, heat capacity and liquid water content of the snow pack are set to zero.

If HRCOOL $\geq$ HSNWRM but HRCOOL $<$ (HSNWRM + HSNMLT), the energy contributed by the temperature of the rainfall is sufficient to warm the whole snowpack to 0 C but not to melt all of it. HSNMLT is therefore recalculated as HRCOOL - HSNWRM, and used to determine a melted depth of the snowpack ZMELT, which is subtracted from the snow depth ZSNOW. The energy used to melt this depth of snow is added to HMFN and HTCS. The total water now available for retention in the snowpack, WAVAIL, is obtained as the sum of the mass of melted water and the mass of water originally retained in the snow pack, WSNOW. This amount is compared to the water retention capacity of the snow pack, calculated from the maximum retention percentage by weight, WSNCAP (currently set to 4%). If WAVAIL is greater than the water retention capacity, WSNOW is set to the capacity value and the excess is reassigned to ZMELT. Otherwise WSNOW is set to WAVAIL and ZMELT is set to zero. The temperature of the snow and the temperature TR of the rainfall reaching the ground surface are each set to 0 C, HCPSNO is recalculated, and ZMELT is added to the rainfall rate R.

If HSNWRM $\geq$ (HRCOOL + HRFREZ), the energy sink of the snow pack is sufficient to cool to 0 C and freeze all of the rainfall. HSNWRM is recalculated as the difference between HSNWRM and (HRCOOL + HRFREZ). The energy used in the freezing, HRFREZ, is added to HMFN and HTCS. The rainfall is applied to increasing the density of the snow pack RHOSNO; if the new density is greater than the density of ice, RHOSNO is reset to the ice density and ZSNOW is recalculated. HCPSNO is also recalculated, and the new snow temperature is obtained from HSNWRM, HCPSNO and ZSNOW. R and TR are set to zero.

If HSNWRM $>$ HRCOOL and HSNWRM $<$ (HRCOOL + HRFREZ), the energy sink of the snow pack is sufficient to cool the rainfall to 0 C, but not to freeze all of it. HRFREZ is therefore recalculated as HSNWRM – HRCOOL, and used to determine a depth of rain to be frozen, ZFREZ. The energy used in the freezing is added to HMFN and HTCS. The frozen rainfall is applied to increasing the density of the snow pack as above; if the calculated density exceeds that of ice, RHOSNO and ZSNOW are recalculated. The water available for retention in the snow pack, WAVAIL, is obtained as the sum of the unfrozen rainfall and WSNOW, and compared to the water retention capacity of the snow pack. If WAVAIL is greater than the water retention capacity, WSNOW is set to the capacity value and WAVAIL is recalculated. Otherwise WSNOW is set to WAVAIL and WAVAIL is set to zero. The heat capacity of the

snow is recalculated, R is calculated from WAVAIL, and TR and TSNOW are set to zero.

Finally, the calculation of the change in internal energy is completed, and the rainfall rate leaving the bottom of the snow pack and reaching the soil is added to the diagnostic variables PCPG and ROFN.

9.57 SNOADD.f File Reference

Purpose: Add snow incident on the ground surface to the snow pack.

Functions/Subroutines

- subroutine snoadd (ALBSNO, TSNOW, RHOSNO, ZSNOW, HCPSNO, HTCS, FI, S, TS, RHOSNI, WSNOW, ILG, IL1, IL2, JL)

9.57.1 Detailed Description

Purpose: Add snow incident on the ground surface to the snow pack.

9.57.2 Function/Subroutine Documentation

9.57.2.1 subroutine snoadd (real, dimension(ilg) *ALBSNO,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *HCPSNO,* real, dimension (ilg) *HTCS,* real, dimension (ilg) *FI,* real, dimension (ilg) *S,* real, dimension (ilg) *TS,* real, dimension(ilg) *RHOSNI,* real, dimension (ilg) *WSNOW,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

albsno	Albedo of snow [ ]
tsnow	Temperature of the snow pack $[C](T_s)$
rhosno	Density of snow pack $[kgm^{-3}](\rho_s)$
zsnow	Depth of snow pack $[m](z_s)$
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}](C_s)$
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}](I_s)$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
s	Snowfall rate incident on snow pack $[ms^{-1}]$
ts	Temperature of snowfall [C]
rhosni	Density of fresh snow $[kgm^{-3}]$
wsnow	Liquid water content of snow pack $[kgm^{-2}](w_s)$

The change of internal energy HTCS of the snow pack as a result of the snowfall added to it is calculated as the difference in Is between the beginning and end of the subroutine:

$$\Delta I_s = X_i \Delta [C_s z_s T_s] / \Delta t$$

where C_s represents the volumetric heat capacity of the snow pack, T_s its temperature, Δ the length of the time step, and X_i the fractional coverage of the subarea under consideration relative to the modelled area.

The amount of snow incident at the given time step, SNOFAL, is calculated from S and the timestep length DELT. If SNOFAL $\geq$ 0.1 mm, the snow albedo is set to the fresh snow value of 0.84. Otherwise, if the snow is falling on bare ground, its initial albedo is set to the old snow value of 0.50. The heat capacity of the precipitating snow, HCPSNP, is calculated from the fresh snow density RHOSNI and the heat capacity and density of ice. The new temperature of the snow pack is calculated as a weighted average of its old temperature, weighted by the snow depth ZSNOW and heat capacity HCPSNO, and the snowfall temperature, weighted by SNOFAL and HCPSNP. The new density of snow is calculated as a weighted average of the original density RHOSNO and RHOSNI, and the new snow depth is calculated as ZSNOW + SNOFAL. Finally, the new heat capacity of the snow pack is obtained from the heat capacities of ice and water C_i and C_w, the snow, ice and water densities ρ_s rho_i, and ρ_w, and the water content and depth of the snow pack w_s and z_s, as:

$C_s = C_i[\rho_s/\rho_i] + C_w w_s/[\rho_w z_s]$

9.58 SNOALBA.f File Reference

Purpose: Diagnose snowpack visible and near-IR albedos given the all-wave albedo at the current time step. Calculate snowpack transmissivity for shortwave radiation.

Functions/Subroutines

- subroutine snoalba (ALVSSN, ALIRSN, ALVSSC, ALIRSC, ALBSNO, TRSNOWC, ALSNO, TRSNOWG, F↩ SDB, FSFB, RHOSNO, REFSN, BCSN, SNO, CSZ, ZSNOW, FSNOW, ASVDAT, ASIDAT, ALVSG, ALIRG, ILG, IG, IL1, IL2, JL, IALS, NBS, ISNOALB)

9.58.1 Detailed Description

Purpose: Diagnose snowpack visible and near-IR albedos given the all-wave albedo at the current time step. Calculate snowpack transmissivity for shortwave radiation.

9.58.2 Function/Subroutine Documentation

**9.58.2.1 subroutine snoalba (real, dimension(ilg) *ALVSSN,* real, dimension(ilg) *ALIRSN,* real, dimension(ilg) *ALVSSC,* real, dimension(ilg) *ALIRSC,* real, dimension(ilg) *ALBSNO,* real, dimension(ilg) *TRSNOWC,* real, dimension (ilg,nbs) *ALSNO,* real, dimension(ilg,nbs) *TRSNOWG,* real, dimension(ilg,nbs) *FSDB,* real, dimension(ilg,nbs) *FSFB,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *REFSN,* real, dimension (ilg) *BCSN,* real, dimension (ilg) *SNO,* real, dimension (ilg) *CSZ,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *FSNOW,* real, dimension(ilg) *ASVDAT,* real, dimension(ilg) *ASIDAT,* real, dimension (ilg) *ALVSG,* real, dimension (ilg) *ALIRG,* integer *ILG,* integer *IG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IALS,* integer *NBS,* integer *ISNOALB*)

Parameters

alvssn	Visible albedo of snow pack on bare ground $[](alpha_{s,VIS})$
alirsn	Near-IR albedo of snow pack on bare ground $[](alpha_{s,NIR})$
alvssc	Visible albedo of snow on ground under vegetation canopy [ ]
alirsc	Near-IR albedo of snow on ground under vegetation canopy [ ]
trsnowc	Transmissivity of snow to shortwave radiation $[](\tau_s)$
albsno	All-wave albedo of snow pack $[](\alpha_{s,T})$
zsnow	Depth of snow $[m](z_s)$
fsnow	Fractional coverage of snow on grid cell [ ]
asvdat	Assigned value of visible albedo of snow pack – optional [ ]
asidat	Assigned value of near-IR albedo of snow pack – optional [ ]

In subroutine SNOALBW, called at the end of CLASSW, the change of total snow albedo over the current time step is calculated using an empirical exponential decay function, which has different coefficients depending on whether the snow is dry or melting. In this subroutine, the visible and near-IR components of the snow albedo are diagnosed from the total albedo. According to the literature (Aguado, 1985; Robinson and Kukla, 1984; Dirmhirn and Eaton, 1975), the following represent typical snow albedos for fresh snow, old dry snow and melting snow:

	Total albedo	Visible albedo	Near-IR albedo
Fresh snow	0.84	0.95	0.73
Old dry snow	0.70	0.84	0.56
Melting snow	0.50	0.62	0.38

The same decay function is assumed to apply to all three albedo ranges, so the relative location of the visible and near-IR albedos, $\alpha_{s,VIS}$ and $\alpha_{s,NIR}$, on the decay curve will be analogous to that of the total albedo, $\alpha_{s,T}$. Thus, for dry snow:

$[\alpha_{s,VIS}-0.84]/[0.95-0.84] = [\alpha_{s,T}-0.70]/[0.84-0.70]$ $[\alpha_{s,NIR}-0.56]/[0.73-0.56] = [\alpha_{s,T}-0.70]/[0.84-0.70]$

or, simplifying:

$\alpha_{s,VIS} = 0.79[\alpha_{s,T} - 0.70] + 0.84$ $\alpha_{s,NIR} = 1.21[\alpha_{s,T} - 0.70] + 0.56$

For melting snow:

$[\alpha_{s,VIS}-0.62]/[0.95-0.62] = [\alpha_{s,T}-0.50]/[0.84-0.50]$ $[\alpha_{s,NIR}-0.38]/[0.73-0.38] = [\alpha_{s,T}-0.50]/[0.84-0.50]$

or, simplifying:

$\alpha_{s,VIS} = 0.97[\alpha_{s,T} - 0.50] + 0.62$ $\alpha_{s,NIR} = 1.03[\alpha_{s,T} - 0.50] + 0.38$

The above calculations are performed if the flag IALS is set to zero. If IALS is set to one, indicating that assigned snow albedos are to be used instead of calculated values, $\alpha_{s,VIS}$ and $\alpha_{s,NIR}$ are set to the assigned values AS↩ VDAT and ASIDAT respectively. The sub-canopy values of visible and near-IR albedo are currently set equal to the open snowpack values (this is expected to change if a canopy litterfall parametrization is developed).

The transmissivity of snow τ_s is calculated from the snow depth ZSNOW using Beer's law, with an empirical extinction coefficient of $25.0m^{-1}$ derived from the literature (Grenfell and Maykut, 1977; Thomas, 1963):

$\tau_s = exp[-25.0z_s]$

Convert the units of the snow grain size and BC mixing ratio Snow grain size from meters to microns and BC from $kgBC/m^3$ to ng BC/kg SNOW

9.59 SNOALBW.f File Reference

Purpose: Calculate decrease in snow albedo and increase in density due to aging.

Functions/Subroutines

- subroutine snoalbw (ALBSNO, RHOSNO, ZSNOW, HCPSNO, TSNOW, FI, S, RMELT, WSNOW, RHOMAX, ISAND, ILG, IG, IL1, IL2, JL)

9.59.1 Detailed Description

Purpose: Calculate decrease in snow albedo and increase in density due to aging.

9.59.2 Function/Subroutine Documentation

9.59.2.1 subroutine snoalbw (real, dimension(ilg) *ALBSNO,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *HCPSNO,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *FI,* real, dimension (ilg) *S,* real, dimension (ilg) *RMELT,* real, dimension (ilg) *WSNOW,* real, dimension(ilg) *RHOMAX,* integer, dimension (ilg,ig) *ISAND,* integer *ILG,* integer *IG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

albsno	Albedo of snow $[](\alpha_s)$
rhosno	Density of snow pack $[kgm^{-3}](\rho_s)$
zsnow	Depth of snow pack $[m](z_s)$
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}]$

tsnow	Temperature of the snow pack $[C]$
fi	Fractional coverage of subarea in question on modelled area [ ]
s	Snowfall rate $[ms^{-1}]$
rmelt	Melt rate at top of snow pack $[ms^{-1}]$
wsnow	Liquid water content of snow pack $[kgm^{-2}]$
isand	Sand content flag
rhomax	Maximum density of snow pack $[kgm^{-3}](\rho_{s,max})$

The albedo and density of snow are modelled using empirical exponential decay functions. In the absence of any fresh snowfall the snow albedo α_s is assumed to decrease exponentially with time from a fresh snow value of 0.84 to a background old snow value $\alpha_{s,old}$ using an expression based on data given in Aguado (1985), Robinson and Kukla (1984) and Dirmhirn and Eaton (1975):

$$\alpha_s(t+1) = [\alpha_s(t) - \alpha_{s,old}]exp[-0.01\Delta t/3600] + \alpha_{s,old}$$

where Δt is the length of the time step. If the melt rate RMELT at the top of the snow pack is non- negligible or if the temperature of the snow is close to 0 C, $\alpha_{s,old}$ is assigned a value of 0.50; otherwise $\alpha_{s,old}$ is assigned a value of 0.70.

The maximum snow density $\rho_{s,max}$ is estimated as a function of snow depth z_s, after Tabler et al. (1990):

$$\rho_{s,max} = A_s - [204.70/z_s][1.0 - exp(-z_s/0.673)]$$

The empirical constant A_s is assigned a value of 450.0 for cold snow packs, and 700.0 for snow packs near the melting point, following Brown et al. (2006).

The density of snow ρ_s increases exponentially with time from its fresh snow value to the background old snow density calculated above, according to an expression analogous to that for albedo, derived from the field measurements of Longley (1960) and Gold (1958):

$$\rho_s(t+1) = [\rho_s(t) - \rho_{s,max}]exp[-0.01\Delta t/3600] + \rho s, max$$

The snow depth and heat capacity are adjusted (see notes on subroutine SNOVAP), and a check is performed with a call to abort if for unphysical albedo values are encountered.

9.60 SNOVAP.f File Reference

Purpose: Sublimation calculations for the snow pack on the ground.

Functions/Subroutines

- subroutine snovap (RHOSNO, ZSNOW, HCPSNO, TSNOW, EVAP, QFN, QFG, HTCS, WLOST, TRUNOF, RUNOFF, TOVRFL, OVRFLW, FI, R, S, RHOSNI, WSNOW, ILG, IL1, IL2, JL)

9.60.1 Detailed Description

Purpose: Sublimation calculations for the snow pack on the ground.

9.60.2 Function/Subroutine Documentation

9.60.2.1 subroutine snovap (real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *HCPSNO,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *QFN,* real, dimension (ilg) *QFG,* real, dimension (ilg) *HTCS,* real, dimension (ilg) *WLOST,* real, dimension(ilg) *TRUNOF,* real, dimension(ilg) *RUNOFF,* real, dimension(ilg) *TOVRFL,* real, dimension(ilg) *OVRFLW,* real, dimension (ilg) *FI,* real, dimension (ilg) *R,* real, dimension (ilg) *S,* real, dimension(ilg) *RHOSNI,* real, dimension (ilg) *WSNOW,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

rhosno	Density of snow pack $[kgm^{-3}](\rho_s)$
zsnow	Depth of snow pack $[m](z_g)$
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}](C_s)$
tsnow	Temperature of the snow pack $[C](T_s)$
evap	Sublimation rate from snow surface at start of subroutine $[ms^{-1}]$
qfn	Sublimation from snow pack $[kgm^{-2}s^{-1}]$
qfg	Evaporation from ground $[kgm^{-2}s^{-1}]$
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}](I_s)$
wlost	Residual amount of water that cannot be supplied by surface stores $[kgm^{-2}]$
trunof	Temperature of total runoff [K]
runoff	Total runoff $[ms^{-1}]$
tovrfl	Temperature of overland flow [K]
ovrflw	Overland flow from top of soil column $[ms^{-1}]$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
r	Rainfall rate incident on snow pack $[ms^{-1}]$
s	Snowfall rate incident on snow pack $[kgm^{-2}s^{-1}]$
rhosni	Density of fresh snow $[kgm^{-3}]$
wsnow	Liquid water content of snow pack $[kgm^{-2}](w_s)$

These calculations are done if a snowpack is present and there is no rainfall or snowfall occurring. The change of internal energy I_s of the snow pack as a result of the change in its mass is calculated as the difference in I_s between the beginning and end of the subroutine:

$$\Delta I_s = X_i \Delta [C_s z_s T_s] / \Delta t$$

where C_s represents the volumetric heat capacity of the snow pack, T_s its temperature, Δt the length of the time step, and X_i the fractional coverage of the subarea under consideration relative to the modelled area.

If the sublimation rate EVAP over the snow pack is negative (downward), the deposited depth of snow ZADD is calculated from EVAP by converting it from a liquid water flux to a fresh snow depth using RHOSNI, the fresh snow density. The snowpack density is updated as a weighted average of the original snow density RHOSNO and RH↩ OSNI. The new snow depth is calculated as the sum of the old snow depth ZSNOW and ZADD. The new volumetric heat capacity of the snow pack is obtained from the heat capacities of ice and water C_i and C_w, the snow, ice and water densities ρ_s ρ_i, and ρ_w, and the water content and depth of the snow pack w_s and z_s, as:

$$C_s = C_i [\rho_s / \rho_i] + C_w w_s / [\rho_w z_s]$$

If the sublimation rate is positive, the depth of the snow pack ZLOST that is sublimated over the time step is calculated from EVAP using RHOSNO. If ZLOST $\leq$ ZSNOW, the snow depth is reduced and HCPSNO is recalculated. Otherwise the deficit amount ZREM is calculated from ZLOST – ZSNOW and converted to a depth of water. This amount is further converted to an evaporation rate by applying a correction factor of $(L_m + L_v)/L_v$, where L_m is the latent heat of melting and L_v is the latent heat of vaporization (to account for the fact that the energy is now being used to evaporate water instead of sublimate snow). This necessarily leads to a small discrepancy between the overall vapour flux for the subarea that was originally calculated in CLASST, and the actual change of water storage in the subarea, and therefore this discrepancy is added to the housekeeping variable WLOST for use in the water balance checks done later in CHKWAT. If there was liquid water in the snow pack, WSNOW, it is assigned to overall runoff RUNOFF, and to overland flow OVRFLW. The resulting temperatures of the runoff and overland flow, TRUNOF and TOVRFL, are recalculated as weighted averages using the original runoff amounts and temperatures, and the original snow temperature TSNOW for WSNOW. The snow depth, heat capacity, temperature and water content are all set to zero. Finally, since ZREM now becomes soil evaporation rather than snow sublimation, the diagnostic variables QFN and QFG, representing the vapour flux from snow and soil respectively, are adjusted to reflect this.

9.61 soil_ch4uptake.f90 File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Soil Methane Oxidation Subroutine.

Functions/Subroutines

- subroutine soil_ch4uptake (IL1, IL2, tbar, THP, BI, THLQ, THIC, PSIS, GRAV, FCAN, obswetf, wetfdyn, wetfracgrd, isand, RHOW, RHOICE, atm_CH4, CH4_soills)

9.61.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Soil Methane Oxidation Subroutine.

Coded up based on [19].

9.61.2 Function/Subroutine Documentation

9.61.2.1 subroutine soil_ch4uptake (integer, intent(in) *IL1,* integer, intent(in) *IL2,* real, dimension(ilg,ignd), intent(in) *tbar,* real, dimension(ilg,ignd), intent(in) *THP,* real, dimension(ilg,ignd), intent(in) *BI,* real, dimension(ilg,ignd), intent(in) *THLQ,* real, dimension(ilg,ignd), intent(in) *THIC,* real, dimension(ilg,ignd), intent(in) *PSIS,* real, intent(in) *GRAV,* real, dimension(ilg,ican), intent(in) *FCAN,* logical, intent(in) *obswetf,* real, dimension(ilg), intent(in) *wetfdyn,* real, dimension(nlat), intent(in) *wetfracgrd,* integer, dimension(ilg,ignd), intent(in) *isand,* real, intent(in) *RHOW,* real, intent(in) *RHOICE,* real, dimension(ilg), intent(in) *atm_CH4,* real, dimension(ilg), intent(out) *CH4_soills*)

Parameters

in	*tbar*	Temperature of soil layers (K) - daily average
in	*thp*	Total porosity (cm^3cm^{-3}) - daily average
in	*bi*	Clapp and Hornberger b-term (-)
in	*thlq*	Fractional water content (-) - daily average
in	*thic*	Fractional ice content (-) - daily average
in	*psis*	Soil moisture suction at saturation (m)
in	*fcan*	Fractional coverage of vegetation (-)
in	*wetfracgrd*	Prescribed fraction of wetlands in a grid cell
in	*wetfdyn*	Dynamic gridcell wetland fraction determined using slope and soil moisture
in	*atm_ch4*	Atmospheric CH_4 concentration at the soil surface (ppmv)
in	*grav*	Acceleration due to gravity (ms^{-1})
in	*obswetf*	Switch, if true then use the prescribed wetland cover
in	*rhow*	Density of water (kgm^{-3})
in	*rhoice*	Density of ice (kgm^{-3})
in	*isand*	flag for soil/bedrock/ice/glacier
out	*ch4_soills*	Methane uptake into the soil column $(mgCH_4m^{-2}s^{-1})$

Begin

The soil oxidation methane sink is assumed to only operate in the first model soil layer, thus we only consider that layer here.

not soil so move on.

Convert tbar to Tsoil (from K to deg C)

Find the diffusion coefficient in soil (D_soil)

First the temperature factor, G_T:

Find the air filled porosity, THP_air:

Note: THP_air can fall to $<$ 0 after snow melt

The BI (Clapp and Hornberger b-term) is already calculated by CLASS as: BI = 15.9 $*$ f_clay + 2.91, thus we use that value.

G_soil is the influence of the soil texture, moisture, and porosity:

The diffusion coefficient of CH4 in soil is then:

Determine the first-order oxidation rate constant (k_oxidr)

First find the temperature term, r_T (FLAG note that Charles' original code does not have the high temp limit!)

all other temps ($<$-10 and $>$=43.3)

Next find the term based on soil moisture (suction)

Find the soil water potential for the uppermost layer need the absolute value.

Convert units from m to kPa

0.2 MPa in paper (NOTE: In Charles's code this is $\leq$, but is $<$ in paper)

0.2 and 100 Mpa in paper

psi $>$ 100 MPa.

Find the flux correction for croplands

Find the flux correction due to wetlands

Use the prescribed wetland fractions

use the dynamically determined wetland area

Find the surface flux (CH4_soills) for each tile, then for each gridcell

Convert from mg CH4 m^-2 s^-1 to umol CH4 m^-2 s^-1

9.62 STORVAR.f File Reference

Functions/Subroutines

- subroutine storvar (NF, G, NTIME, NAME, ILEV, ILG, ILAT, MAXX)

9.62.1 Function/Subroutine Documentation

9.62.1.1 subroutine storvar (integer *NF,* real, dimension(10088) *G,* integer *NTIME,* character *NAME,* integer *ILEV,* integer *ILG,* integer *ILAT,* integer *MAXX*)

- DETERMINE THE MACHINE TYPE (PASSED TO LOW-LEVEL I/O ROUTINES).

9.63 SUBCAN.f File Reference

Purpose: Assess water flux elements at the ground surface under the vegetation canopy.

Functions/Subroutines

- subroutine subcan (IWATER, R, TR, S, TS, RHOSNI, EVAPG, QFN, QFG, PCPN, PCPG, FI, ILG, IL1, IL2, JL)

9.63.1 Detailed Description

Purpose: Assess water flux elements at the ground surface under the vegetation canopy.

9.63.2 Function/Subroutine Documentation

9.63.2.1 subroutine subcan (integer *IWATER,* real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg) *S,* real, dimension (ilg) *TS,* real, dimension(ilg) *RHOSNI,* real, dimension (ilg) *EVAPG,* real, dimension (ilg) *QFN,* real, dimension (ilg) *QFG,* real, dimension (ilg) *PCPN,* real, dimension (ilg) *PCPG,* real, dimension (ilg) *FI,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

r	Rainfall rate incident on ground $[ms^{-1}]$
tr	Temperature of rainfall [C]
s	Snowfall rate incident on ground $[ms^{-1}]$
ts	Temperature of snowfall [C]
rhosni	Density of fresh snow $[kgm^{-3}]$
evapg	Evaporation rate from surface $[ms^{-1}]$
qfn	Sublimation from snow pack $[kgm^{-2}s^{-1}]$
qfg	Evaporation from ground $[kgm^{-2}s^{-1}]$
pcpn	Precipitation incident on ground $[kgm^{-2}s^{-1}]$
pcpg	Precipitation incident on ground $[kgm^{-2}s^{-1}]$
fi	Fractional coverage of subarea in question on modelled area [ ]

This subroutine starts with the precipitation rate under the canopy (a result of throughfall and unloading) and calculates the resulting overall evaporation or deposition rates.

For IWATER = 2 (snow on the ground under the canopy), the water vapour flux EVAPG at the ground surface is in the first instance assumed to be sublimation. Thus the first step is to compare it to the snowfall rate. The sum of the snowfall rate and the evaporation rate, SADD, is calculated as S – EVAPG, with EVAPG converted from a liquid water flux (the standard output from TSOLVC) to a snow flux. If SADD is greater than zero (indicating a downward flux) the snowfall rate is set to SADD and EVAPG is set to zero. Otherwise EVAPG is set to -SADD (converted back to a liquid water flux), and S and TS are set to zero.

After this section, any remaining evaporative flux is compared to the rainfall rate. The sum RADD is calculated as R – EVAPG. If RADD is greater than zero, the rainfall rate is set to RADD and EVAPG is set to zero. Otherwise EVAPG is set to –RADD, and R and TR are set to zero.

Analogous calculations are done for IWATER = 1 (bare ground under the canopy). In this case EVAPG is assumed in the first instance to be evaporation or condensation. Thus the first step is to compare it to the rainfall rate, and the same steps are followed as in the paragraph above. Afterwards, any remaining vapour flux is compared to the snowfall rate. If SADD is positive (downward), EVAPG, which is now considered to be absorbed into the snowfall rate, is subtracted from the ground vapour flux QFG and added to the snow vapour flux QFN. If SADD is negative (upward), S, which has now been absorbed into the evaporative flux, is subtracted from the snow precipitation flux PCPN and added to the ground precipitation flux PCPG.

9.64 TFREEZ.f File Reference

Purpose: Address freezing of water ponded on ground surface.

Functions/Subroutines

- subroutine tfreez (ZPOND, TPOND, ZSNOW, TSNOW, ALBSNO, RHOSNO, HCPSNO, GZERO, HMFG, H↩ TCS, HTC, WTRS, WTRG, FI, QFREZ, WSNOW, TA, TBAR, ISAND, IG, ILG, IL1, IL2, JL)

9.64.1 Detailed Description

Purpose: Address freezing of water ponded on ground surface.

9.64.2 Function/Subroutine Documentation

9.64.2.1 subroutine tfreez (real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TPOND,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *ALBSNO,* real, dimension(ilg) *RHOSNO,* real, dimension(ilg) *HCPSNO,* real, dimension (ilg) *GZERO,* real, dimension (ilg,ig) *HMFG,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *WTRS,* real, dimension (ilg) *WTRG,* real, dimension (ilg) *FI,* real, dimension (ilg) *QFREZ,* real, dimension (ilg) *WSNOW,* real, dimension (ilg) *TA,* real, dimension (ilg,ig) *TBAR,* integer, dimension (ilg,ig) *ISAND,* integer *IG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

zpond	Depth of ponded water $[m](z_p)$
tpond	Temperature of ponded water $[C](T_p)$
zsnow	Depth of snow pack $[m](z_g)$
tsnow	Temperature of the snow pack $[C](T_s)$
albsno	Albedo of snow [ ]
rhosno	Density of snow pack $[kgm^{-3}](\rho_s)$
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}](C_s)$
gzero	Heat flow into soil surface $[Wm^{-2}]$
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}]$ (ls)
wtrs	Water transferred into or out of the snow pack $[kgm^{-2}s^{-1}]$
wtrg	Water transferred into or out of the soil $[kgm^{-2}s^{-1}]$
hmfg	Energy associated with phase change of water in soil layers $[Wm^{-2}]$
htc	Internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}]$ (lg)
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
qfrez	Energy sink for freezing of water at the ground surface $[Wm^{-2}]$
wsnow	Liquid water content of snow pack $[kgm^{-2}](w_s)$
ta	Air temperature [K]
tbar	Temperature of soil layer $[C](T_g)$
isand	Sand content flag

Freezing of water ponded on the ground surface occurs if an energy sink QFREZ is produced as a result of the solution of the surface energy balance, or if the pond temperature at the beginning of the subroutine has been projected to be below 0 C. The change of internal energy I in the snow and first soil layer (which for the purposes of diagnostic calculations includes the ponded water) as a result of these processes is calculated as the difference in I between the beginning and end of the subroutine:

$\Delta I_s = X_i \Delta(C_s T_s z_s)/\Delta t \; \Delta I_g = X_i \Delta(C_w T_p z_p)/\Delta t$

where the C terms represent volumetric heat capacities, the T terms temperatures, and the z terms depths of the snow pack and the ponded water respectively, Δt is the length of the time step, and X_i the fractional coverage of the subarea under consideration relative to the modelled area.

The energy sink HADD to be applied to the ponded water is calculated from QFREZ and the pond temperature T↩ POND (if it is below 0 C). Two diagnostic variables, HCOOL and HCONV, are calculated as the energy sink required to cool the ponded water to 0 C, and that required both to cool and to freeze the ponded water, respectively. If HADD $\leq$ HCOOL, the available energy sink is only sufficient to decrease the temperature of the ponded water. This decrease is applied, and the energy used is added to the internal energy HTC of the first soil layer.

If HADD $>$ HCOOL but HADD $\leq$ HCONV, the available energy sink is sufficient to decrease the ponded water temperature to 0 C and to freeze part of it. The energy used in freezing is calculated as HADD – HCOOL, and is used to calculate the depth of frozen water ZFREZ, which is then subtracted from the ponded water depth ZPOND. HCOOL is added to HTC, and ZFREZ is converted from a depth of water to a depth of ice and added to the snow pack. If there is not a pre-existing snow pack, the snow albedo is set to the background value for old snow, 0.50. The temperature, density and heat capacity of the snow pack are recalculated.

If HADD $>$ HCONV, the available energy sink is sufficient to cool and freeze the whole depth of ponded water, and also to decrease the temperature of the ice thus formed. The ponded water depth is converted to a depth of ice ZFREZ, and HCOOL is added to HTC. In order to avoid unphysical temperature decreases caused by the length of the time step and the small ponding depth, a limit is set on the allowable decrease. The theoretical temperature of the newly formed ice, TTEST, is calculated from HADD – HCONV. If there is a pre-existing snow pack, the limiting temperature TLIM is set to the minimum of the snow pack temperature TSNOW, the first soil layer temperature, and 0 C; otherwise it is set to the minimum of the air temperature, the first soil layer temperature and 0 C. If TTEST $<$ TLIM, the new ice is assigned TLIM as its temperature in the recalculation of TSNOW, the excess heat sink HE↩ XCES is calculated from HADD and TLIM and assigned to the ground heat flux GZERO, and HADD – HEXCES is used to update HTC. Otherwise the new ice is assigned TTEST as its temperature in the recalculation of TSNOW, and HADD is used to update HTC. If there is not a pre-existing snow pack, the snow albedo is set to the background value. The density, depth and heat capacity of the snow pack are recalculated, and the pond depth and temperature are set to zero.

At the end of the subroutine, the internal energy calculations are completed, and ZFREZ is used to update the

diagnostic variable HMFG describing the energy associated with phase changes of water in soil layers, and also the diagnostic variables WTRS and WTRG describing transfers of water into or out of the snow and soil respectively. Finally, the initial step in the calculation of the internal energy change for the ponded water over the following subroutines GRINFL and GRDRAN is performed (the calculation is completed in subroutine TMCALC).

9.65 TMCALC.f File Reference

Purpose: Calculate overland flow; step ahead pond and soil layer temperatures, and check for freezing of the pond and freezing or thawing of liquid or frozen water in the soil layers. Adjust pond temperature, soil layer temperatures and water stores accordingly.

Functions/Subroutines

- subroutine tmcalc (TBAR, THLIQ, THICE, HCP, TPOND, ZPOND, TSNOW, ZSNOW, ALBSNO, RHOSNO, HCPSNO, TBASE, OVRFLW, TOVRFL, RUNOFF, TRUNOF, HMFG, HTC, HTCS, WTRS, WTRG, FI, TBA↩ RW, GZERO, G12, G23, GGEO, TA, WSNOW, TCTOP, TCBOT, GFLUX, ZPLIM, THPOR, THLMIN, HCPS, DELZW, DELZZ, DELZ, ISAND, IWF, IG, ILG, IL1, IL2, JL, N)

9.65.1 Detailed Description

Purpose: Calculate overland flow; step ahead pond and soil layer temperatures, and check for freezing of the pond and freezing or thawing of liquid or frozen water in the soil layers. Adjust pond temperature, soil layer temperatures and water stores accordingly.

9.65.2 Function/Subroutine Documentation

9.65.2.1 subroutine tmcalc (real, dimension (ilg,ig) *TBAR,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *HCP,* real, dimension (ilg) *TPOND,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *ALBSNO,* real, dimension(ilg) *RHOSNO,* real, dimension(ilg) *HCPSNO,* real, dimension (ilg) *TBASE,* real, dimension(ilg) *OVRFLW,* real, dimension(ilg) *TOVRFL,* real, dimension(ilg) *RUNOFF,* real, dimension(ilg) *TRUNOF,* real, dimension (ilg,ig) *HMFG,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *HTCS,* real, dimension (ilg) *WTRS,* real, dimension (ilg) *WTRG,* real, dimension (ilg) *FI,* real, dimension (ilg,ig) *TBARW,* real, dimension (ilg) *GZERO,* real, dimension (ilg) *G12,* real, dimension (ilg) *G23,* real, dimension (ilg) *GGEO,* real, dimension (ilg) *TA,* real, dimension (ilg) *WSNOW,* real, dimension (ilg,ig) *TCTOP,* real, dimension (ilg,ig) *TCBOT,* real, dimension (ilg,ig) *GFLUX,* real, dimension (ilg) *ZPLIM,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *HCPS,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *DELZZ,* real, dimension (ig) *DELZ,* integer, dimension (ilg,ig) *ISAND,* integer *IWF,* integer *IG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

tbar	Temperature of soil layer $[C](T_g)$
thliq	Volumetric liquid water content of soil layer $(\theta_l)[m^3m^{-3}]$
thice	Volumetric frozen water content of soil layer $(\theta_i)[m^3m^{-3}]$
hcp	Heat capacity of soil layer $[Jm^{-3}K^{-1}](C_g)$
hmfg	Energy associated with freezing or thawing of water in soil layer $[Wm^{-2}]$
htc	Internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}](I_g)$
tpond	Temperature of ponded water $[C](T_p)$
zpond	Depth of ponded water $[m](z_p)$

tsnow	Temperature of the snow pack $[C](T_s)$
zsnow	Depth of snow pack $[m](z_g)$
albsno	Albedo of snow [ ]
rhosno	Density of snow pack $(\rho_s)[kgm^{-3}]$
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}](C_s)$
tbase	Temperature of bedrock in third soil layer, if only three layers are being modelled [K]
ovrflw	Overland flow from top of soil column [m]
tovrfl	Temperature of overland flow from top of soil column [K]
runoff	Total runoff from soil column [m]
trunof	Temperature of total runoff from soil column [K]
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}](I_s)$
wtrs	Water transferred into or out of the snow pack $[kgm^{-2}s^{-1}]$
wtrg	Water transferred into or out of the soil $[kgm^{-2}s^{-1}]$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
tbarw	Temperature of water in soil layer [C]
gzero	Heat flow into soil surface $[Wm^{-2}]$
g12	Heat flow between first and second soil layers $[Wm^{-2}]$
g23	Heat flow between second and third soil layers $[Wm^{-2}]$
ggeo	Geothermal heat flux at bottom of soil profile $[Wm^{-2}]$
ta	Air temperature [K]
wsnow	Liquid water content of snow pack $[kgm^{-2}](w_s)$
tctop	Thermal conductivity of soil at top of soil layer $[Wm^{-1}K^{-1}]$
tcbot	Thermal conductivity of soil at bottom of soil layer $[Wm^{-1}K^{-1}]$
zplim	Limiting depth of ponded water [m]
thpor	Pore volume in soil layer $(\theta_p)[m^3m^{-3}]$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $(\theta_r)[m^3m^{-3}]$
hcps	Heat capacity of soil material $[Jm^{-3}K^{-1}](C_m)$
delzw	Permeable thickness of soil layer $(\Delta z_{g,w})[m]$
delzz	Soil layer thicknesses to bottom of permeable depth for standard three-layer configuration, or to bottom of thermal depth for multiple layers $(\Delta z_{g,z})[m]$
delz	Overall thickness of soil layer [m]
isand	Sand content flag
gflux	Heat flow between soil layers $[Wm^{-2}]$

If a full-scale hydrological modelling application is not being run, that is, if only vertical fluxes of energy and moisture are being modelled, the flag IWF is pre-set to zero. In this case, overland flow of water is treated using a simple approach: if the ponded depth of water on the soil surface ZPOND exceeds a limiting value ZPLIM (which varies by land surface type), the excess is assigned to overland flow. These calculations are carried out in loop 100, for all modelled areas except ice sheets (which are handled in subroutine ICEBAL). The overall runoff from the modelled area in question, RUNOFF, is incremented by the excess ponded water, and the overland flow for the whole grid cell OVRFLW is incremented by the product of the excess ponded water and the fractional area of the grid cell. The temperature of the overall runoff from the modelled area TRUNOF, and the temperature of the overland flow for the grid cell TOVRFL, are calculated as weighted averages over their previous values and the ponded water temperature TPOND. Finally, ZPOND is set to ZPLIM.

In loop 200, the calculation of the change in internal energy HTC within the soil layers due to movement of soil water (addressed in the preceding subroutines GRINFL and GRDRAN) is completed. (The initial step was performed at the end of subroutine TWCALC.) The volumetric heat capacity of the soil layers HCP is recalculated as a weighted average over the volumetric contents of liquid water, frozen water and soil particles. The temperature TBAR of each soil layer is recalculated as a weighted average of the liquid water temperature TBARW resulting from soil water movement, and the previous soil layer temperature. As noted in the documentation for subroutine TWCA↩ LC., TBARW and HCP apply only over the permeable depth of the soil layer, DELZW, whereas TBAR applies over the whole depth, DELZZ, and is associated with HCPSND, the volumetric heat capacity assigned to rock, in the interval DELZZ- DELZW. (For the default three-layer version of CLASS, DELZZ=DELZW in the third soil layer; see the documentation of loop 550 below.)

In loop 300 the calculation of the change of internal energy within the first soil layer due to changes in the surface ponded water is completed (for diagnostic purposes, the first level of HTC includes the internal energy of both the ponded water and the first soil layer). (The initial step of this calculation was performed at the end of subroutine

TFREEZ.) The flow of heat between the bottom of the ponded water and the top of the first soil layer, GP1, is calculated by assuming a linear variation with depth of the heat flux between the top of the pond, GZERO, and between the first and second soil layers, G12. Since the heat flux G(z) varies directly with the temperature gradient dT(z)/dz, it can be seen that this approach is consistent with the assumption made in the CLASST subroutines that T(z) is a quadratic function of depth (and therefore that its first derivative is a linear function of depth). The temperature of the ponded water is stepped ahead using GZERO and GP1, and GZERO is reset to GP1 for use in the later soil temperature calculations.

In the 400 loop, a check is performed to ascertain whether the ponded water temperature has fallen below 0 C as a result of the above temperature change. If so, calculations are carried out analogous to those done in TFREEZ. A recalculation of the available energy in the snow pack, HTCS, is initiated. The available energy sink HADD is calculated from TPOND and ZPOND, and TPOND is set to 0 C. The amount of energy required to freeze the whole depth of ponded water is calculated as HCONV. An update of the first level value of HTC, reflecting the freezing of ponded water, is initiated. If HADD $<$ HCONV, the available energy sink is sufficient to freeze only part of the ponded water. This depth ZFREZ is calculated from HADD, and subtracted from ZPOND. ZFREZ is then converted from a depth of water to a depth of ice, and is reassigned as part of the snow pack. The snow temperature, density, heat capacity and depth are recalculated as weighted averages of the pre-existing snow properties and those of the frozen water. If there was no snow on the ground to begin with, the snow albedo is initialized to its minimum background value of 0.50.

If HADD $>$ HCONV, the available energy sink is sufficient to freeze the whole depth of ponded water and then to decrease the frozen water temperature to below 0 C. ZFREZ is evaluated as ZPOND converted to a depth of ice, and the remaining energy sink HADD-HCONV is used to calculated a test temperature TTEST of the newly formed ice. In order to avoid unphysical overshoots of the frozen water temperature, a limiting temperature TLIM is determined as the minimum of the snow temperature and the first level soil temperature if there is pre-existing snow, and as the minimum of the air temperature and the first level soil temperature if not. If TTEST $<$ TLIM, the excess energy sink required to decrease the frozen water temperature from TLIM to TTEST is calculated and added to GZERO, and the new frozen water temperature is assigned as TLIM; otherwise it is assigned as TTEST. The energy sink used to cool the frozen water from 0 C to TLIM or TTEST is decremented from the first level of HTC, and the snow temperature is recalculated as the weighted average of the pre-existing snow temperature and the frozen water temperature. If there was no pre-existing snow, the snow albedo is initialized to the background value of 0.50. The density, heat capacity and depth of the snow are recalculated as above. Finally, the recalculations of the internal energy diagnostic variables HTC and HTCS are completed; and ZFREZ is used to update the diagnostic variable HMFG describing the energy associated with phase changes of water in soil layers, and the diagnostic variables WTRS and WTRG describing transfers of water into or out of the snow and soil respectively.

In the 500, 550 and 600 loops, the heat fluxes between soil layers and the updated temperatures for the soil layers are calculated according to the discretization approach that has been selected for the model run. It is assumed that the first two layer thicknesses are always set to 0.10 and 0.25 m respectively, but two possible options exist for the treatment of deeper layers. In the operational, "three-layer" configuration, a single third soil layer is modelled of thickness 3.75 m, with a user-specified permeable thickness DELZW. TBAR of the third layer is taken to apply to this permeable thickness, whereas a separate bedrock temperature, TBASE, is carried for the impermeable thickness, represented by DELZ-DELZW. This strategy is adopted so that temperature variations caused by melting and freezing of water are confined to the permeable thickness. In the "multi-layer" configuration, there can be any number of deeper layers, of thicknesses specified by the user (with the proviso that the thicknesses not be small enough to lead to numerical instability in the forward explicit time-stepping scheme). In this configuration it is deemed unnecessary to carry a separate calculation of TBASE.

In the CLASST subroutines, the heat fluxes at the top of the first soil layer, and between the first and second and the second and third soil layers, were determined. In loop 500, the fluxes between the remaining soil layers are calculated for the multi-layer configuration, and assigned to the heat flux vector GFLUX. Since the temperature variation is increasingly damped with depth, a simple linearization of the temperature profile is used. The expression for the ground heat flux at depth z, G(z), which depends on the thermal conductivity $\lambda(z)$ and the temperature gradient, is given as: $G(z) = \lambda(z)dT(z)/dz$ The linearized form is written as: $G_j = (\lambda_{j-1} + \lambda_j)(T_{j-1} - T_j)/(\Delta z_{j-1} - \Delta z_j)$ where G j is the heat flux at the top of layer j, and λ_j , T_j and Δz_j refer to the thermal conductivity, temperature and heat capacity of the layer.

In the 550 loop, the first and second soil layer temperatures are stepped ahead using GZERO, G12 and G23. (Recall that the calculated heat capacity HCP applies to the permeable thickness DELZW, and the heat capacity of rock HCPSND applies to the rest of the layer thickness, DELZZ-DELZW.) If the three- layer configuration is being used, then a check is carried out to ascertain whether the permeable thickness DELZZ of the third layer is greater

than zero. If so, and if DELZZ is not effectively equal to DELZ, the heat flow G3B at the interface between the permeable thickness and the underlying bedrock is calculated using the linearized heat flow equation given above; TBAR is updated using the difference between G23 and G3B, and TBASE is updated using the difference between G3B and GGEO, the geothermal heat flux at the bottom of the soil profile. If DELZZ is equal to DELZ, the whole layer is permeable and TBAR is updated using the difference between G23 and GGEO. If DELZZ is zero, the whole layer is impermeable, and TBASE is updated using the difference between G23 and GGEO. HTC for the third layer is updated with the GGEO value. Finally, if the multi-layer configuration is being used, TBAR of the third layer is updated using G23, the flux at the top of the layer. For diagnostic purposes, the first three levels of the GFLUX vector are assigned as GZERO, G12 and G23 respectively.

In the 600 loop, the remaining soil layer temperature calculations are done for layers 3 and deeper, for the multi-layer configuration. At the beginning and end of the loop an updated calculation of HTC for the layer in question is initiated and completed respectively. For the third layer, TBAR is updated using the heat flux at the bottom of the layer; for the last layer, TBAR is updated using the difference between GFLUX at the top of the layer and GGEO at the bottom. For the intermediate layers, TBAR is calculated using the difference between the GFLUX values at the top and bottom of the layer.

In the 700 loop, checks are carried out to determine whether, as a result of the forward stepping of the temperature, TBAR has fallen below 0 C while liquid water still exists in the layer, or risen above 0 C while frozen water still exists in the layer. If either occurs, adjustments to the water content are performed analogous to those done in subroutine TWCALC. Again, at the beginning and end of the loop an updated calculation of HTC for each layer in turn is initiated and completed respectively.

If the soil layer temperature is less than 0 C and the volumetric liquid water content THLIQ of the layer is greater than the residual water content THLMIN, the water content THFREZ that can be frozen by the available energy sink is calculated from TBAR and the weighted average of HCP and HCPSND. If THFREZ $\leq$ THLIQ - THLMIN, all of the available energy sink is used to freeze part of the liquid water content in the permeable part of the soil layer. The amount of energy involved is subtracted from HTC and added to HMFG. THFREZ is subtracted from THLIQ, converted to an ice volume and added to THICE. HCP is recalculated, and the layer temperature is set to 0 C. Otherwise, all of the liquid water content of the layer above THLMIN is converted to frozen water, and HMFG and HTC are recalculated to reflect this. HCP is recomputed, and the remaining energy sink HADD is applied to decreasing the temperature of the soil layer.

If the soil layer temperature is greater than 0 C and the volumetric ice content THICE of the layer is greater than zero, the ice content THMELT that can be melted by the available energy is calculated from TBAR and the weighted average of HCP and HCPSND. If THMELT $\leq$ THICE, all of the available energy is used to melt part of the frozen water content of the permeable part of the layer. The amount of energy involved is subtracted from HTC and added to HMFG. THMELT is subtracted from THICE, converted to a liquid water volume and added to THLIQ; HCP is recalculated and the layer temperature is set to 0 C. Otherwise, all of the frozen water content of the layer is converted to liquid water, and HMFG and HTC are recalculated to reflect this. HCP is recomputed, and the remaining energy HADD is applied to increasing the temperature of the soil layer.

9.66 TMELT.f File Reference

Purpose: Address melting of the snow pack.

Functions/Subroutines

- subroutine tmelt (ZSNOW, TSNOW, QMELT, R, TR, GZERO, RALB, HMFN, HTCS, HTC, FI, HCPSNO, R↩ HOSNO, WSNOW, ISAND, IG, ILG, IL1, IL2, JL)

9.66.1 Detailed Description

Purpose: Address melting of the snow pack.

9.66.2 Function/Subroutine Documentation

9.66.2.1 subroutine tmelt (real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *QMELT,* real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg) *GZERO,* real, dimension (ilg) *RALB,* real, dimension (ilg) *HMFN,* real, dimension (ilg) *HTCS,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *FI,* real, dimension(ilg) *HCPSNO,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *WSNOW,* integer, dimension (ilg,ig) *ISAND,* integer *IG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

htc	Internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}]$
zsnow	Depth of snow pack [m]
tsnow	Temperature of the snow pack [C]
qmelt	Energy available for melting of snow $[Wm^{-2}]$
r	Rainfall rate $[ms^{-1}]$
tr	Temperature of rainfall [C]
gzero	Heat flow into soil surface $[Wm^{-2}]$
ralb	Rainfall rate saved for snow albedo calculations $[ms^{-1}]$
hmfn	Energy associated with freezing or thawing of water in the snow pack $[Wm^{-2}]$
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}]$
fi	Fractional coverage of subarea in question on modelled area [ ]
hcpsno	Heat capacity of snow pack $[Jm^{-3}K^{-1}]$
rhosno	Density of snow pack $[kgm^{-3}]$
wsnow	Liquid water content of snow pack $[kgm^{-2}]$
isand	Sand content flag

Melting of the snow pack occurs if a source of available energy QMELT is produced as a result of the solution of the surface energy balance, or if the snow pack temperature is projected to go above 0 C in the current time step (the available energy thus produced is added to QMELT in subroutine TSPOST). The change in internal energy in the snow pack is calculated at the beginning and end of the subroutine, and stored in diagnostic variable HTCS (see notes on subroutine SNOADD).

The calculations in the 100 loop are performed if QFREZ and the snow depth ZSNOW are both greater than zero. The available energy HADD to be applied to the snow pack is calculated from QMELT. The amount of energy required to raise the snow pack temperature to 0 C and melt it completely is calculated as HCONV. If HADD $\leq$ HCONV, the depth of snow ZMELT that is warmed to 0 C and melted is calculated from HADD. (It is assumed that melting of an upper layer of snow can occur even if the lower part of the snow pack is still below 0 C.) The amount of water generated by melting the snow, RMELTS, is calculated from ZMELT, and the temperature of the meltwater TRMELT is set to 0 C. ZMELT is subtracted from ZSNOW, the heat capacity of the snow is recalculated, and HTCS is corrected for the amount of heat used to warm the removed portion of the snow pack.

If HADD $>$ HCONV, the amount of available energy is sufficient to warm and melt the whole snow pack, with some energy left over. The amount of water generated by melting the snow, RMELTS, is calculated from ZSNOW, and the total amount of water reaching the soil, RMELT, is obtained by adding the liquid water content of the snow pack, WSNOW, to RMELTS. HADD is recalculated as HADD – HCONV, and used to calculate TRMELT. The snow depth, heat capacity, temperature and water content are set to zero, and HTCS is corrected for the amount of heat that was used to warm the snow pack to 0 C.

After the IF block, the diagnostic variable HMFN describing melting or freezing of water in the snow pack is updated using RMELTS, the temperature of the rainfall rate reaching the soil is updated using TRMELT, and RMELT is added to the rainfall rate R. QMELT is set to zero, and a flag variable RALB, used later in subroutine SNOALBW, is set to the rainfall rate reaching the ground.

In the 200 loop, a check is performed to see whether QMELT is still greater than zero and the modelled area is not an ice sheet (ISAND $>$ -4). In this case QMELT is added to the ground heat flux GZERO, and the internal energy diagnostics HTCS and HTC for the snow and soil respectively are corrected. The flag variable RALB is evaluated as above.

9.67 TNPOST.f File Reference

Purpose: Soil heat flux calculations and cleanup after surface energy budget calculations.

Functions/Subroutines

- subroutine tnpost (TBARPR, G12, G23, TPOND, GZERO, QFREZG, GCONST, GCOEFF, TBAR, TCTOP, TCBOT, HCP, ZPOND, TSURF, TBASE, TBAR1P, A1, A2, B1, B2, C2, FI, IWATER, ISAND, DELZ, DELZW, ILG, IL1, IL2, JL, IG)

9.67.1 Detailed Description

Purpose: Soil heat flux calculations and cleanup after surface energy budget calculations.

9.67.2 Function/Subroutine Documentation

9.67.2.1 subroutine tnpost (real, dimension(ilg,ig) *TBARPR,* real, dimension (ilg) *G12,* real, dimension (ilg) *G23,* real, dimension (ilg) *TPOND,* real, dimension (ilg) *GZERO,* real, dimension(ilg) *QFREZG,* real, dimension(ilg) *GCONST,* real, dimension(ilg) *GCOEFF,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg,ig) *TCTOP,* real, dimension (ilg,ig) *TCBOT,* real, dimension (ilg,ig) *HCP,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TSURF,* real, dimension (ilg) *TBASE,* real, dimension(ilg) *TBAR1P,* real, dimension (ilg) *A1,* real, dimension (ilg) *A2,* real, dimension (ilg) *B1,* real, dimension (ilg) *B2,* real, dimension (ilg) *C2,* real, dimension (ilg) *FI,* integer, dimension(ilg) *IWATER,* integer, dimension (ilg,ig) *ISAND,* real, dimension (ig) *DELZ,* real, dimension (ilg,ig) *DELZW,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IG*)

Parameters

tbarpr	Temperatures of soil layers for subarea [C]
g12	Heat conduction between first and second soil layers $[Wm^{-2}](G(\Delta z))$
g23	Heat conduction between second and third soil layers $[Wm^{-2}]$
tpond	Temperature of ponded water $[C](T_p)$
gzero	Heat conduction into soil surface $[Wm^{-2}](G(0))$
qfrezg	Energy sink to be applied to freezing of ponded water $[Wm^{-2}]$
tbar	Temperatures of soil layers, averaged over modelled area [K]
tctop	Thermal conductivity of soil at top of layer $[Wm^{-1}K^{-1}](\lambda)$
tcbot	Thermal conductivity of soil at bottom of layer $[Wm^{-1}K^{-1}](\lambda)$
hcp	Heat capacity of soil layer $[Jm^{-3}K^{-1}]$
delzw	Permeable thickness of soil layer [m]
zpond	Depth of ponded water on surface $[m](\Delta z_p)$
tsurf	Ground surface temperature [K]
tbase	Temperature of bedrock in third soil layer [K]
tbar1p	Lumped temperature of ponded water and first soil layer $[K](T_{1p})$
a1	Work array used in calculation of GCONST and GCOEFF
a2	Work array used in calculation of GCONST and GCOEFF
b1	Work array used in calculation of GCONST and GCOEFF
b2	Work array used in calculation of GCONST and GCOEFF
c2	Work array used in calculation of GCONST and GCOEFF
fi	Fractional coverage of subarea in question on modelled area [ ]
gconst	Intercept used in equation relating ground surface heat flux to surface temperature $[Wm^{-2}]$
gcoeff	Multiplier used in equation relating ground surface heat flux to surface temperature $[Wm^{-2}K^{-1}]$
iwater	Flag indicating condition of surface (dry, water-covered or snow-covered)
isand	Sand content flag
delz	Overall thickness of soil layer $[m](\Delta z)$

In the 100 loop, the heat flux into the ground (without adjustments that may have been applied owing to phase changes of water at the surface or partitioning the residual of the surface energy balance among the surface flux terms) is calculated from the ground surface temperature TSURF, using the GCOEFF and GCONST terms (see documentation of subroutine TNPREP). This flux is then used to back-calculate the heat conduction between the first and second, and the second and third soil layers.

If the depth of water ponded on the surface is greater than zero, the total thickness of the lumped layer consisting of the first soil layer and the ponded water, Δz_{1p}, is calculated as the sum of the two depths. The temperature of the ponded water over the subarea in question is disaggregated from the temperature of the lumped layer, T_{1p}, by making use of the calculated heat fluxes at the top and bottom of the layer, and the assumption (discussed in the documentation of subroutine TNPREP) that that the variation of temperature T with depth z within a soil layer can be modelled using a quadratic equation:

$$T(z) = 1/2T''(0)z^2 + T'(0)z + T(0)$$

Integrating this equation separately over the ponded water depth Δz_p and over the thickness of the lumped layer produces expressions for the ponded water temperature T_p and the temperature of the lumped layer. Making use of the fact that

$T''(0) = [T'(\Delta z) - T'(0)]/\Delta z$

where Δz is a depth interval, and

$G(z) = -\lambda(z)T'(z)$ where λ represents the thermal conductivity, an expression for T_p can be derived:

$T_p = [G(0)/\lambda(0)\check{}G(\Delta z_{1p})/\lambda(\Delta z_{1p})][\Delta z_p^2 - \Delta z_{1p}^2]/6\Delta z_{1p}\check{}G(0)[\Delta z_p - \Delta z_{1p}]/2\lambda(0)\check{}T_{1P}$

The temperature TBARPR of the first soil layer over the subarea in question can then be obtained by disaggregating the ponded water temperature from the lumped layer temperature using the respective heat capacities of the ponded water and the soil layer. (The heat capacity of the soil is determined as the weighted average of HCP over the permeable thickness DELZW, and the heat capacity of rock, HCPSND, over the impermeable thickness, DELZ-↩ DELZW.) Both the ponded water temperature and the soil layer temperature are converted to C. Lastly, if there is ponded water on the surface (IWATER = 1) and QFREZG, the surface energy available for phase change of water, is positive (indicating an energy source), or if there is snow on the ground (IWATER = 2) and QFREZG is negative (indicating an energy sink), or if there is no liquid or frozen water on the surface (IWATER = 0), QFREZG is added to the heat flux into the ground, GZERO, and then reset to zero.

In loop 200, the subarea soil layer temperatures TBARPR are set for the remaining soil layers. In all cases the temperature of the layer is set to that for the modelled area, TBAR, converted to C, except in the case of the third soil layer if the standard three-layer configuration is being modelled (with a very thick third soil layer of 3.75 m). In this case TBARPR and the layer heat capacity HCP are considered to apply to the permeable depth DELZW of the layer, and the bedrock temperature TBASE and the rock heat capacity HCPSND apply to the remainder, DELZ-D↩ ELZW. The disaggregation of TBARPR from TBAR and TBASE is carried out on this basis. TBARPR for this layer is also converted to C.

9.68 TNPREP.f File Reference

Purpose: Calculate coefficients for solution of heat conduction into soil.

Functions/Subroutines

- subroutine tnprep (A1, A2, B1, B2, C2, GDENOM, GCOEFF, GCONST, CPHCHG, IWATER, TBAR, TCTOP, TCBOT, FI, ZPOND, TBAR1P, DELZ, TCSNOW, ZSNOW, ISAND, ILG, IL1, IL2, JL, IG)

9.68.1 Detailed Description

Purpose: Calculate coefficients for solution of heat conduction into soil.

9.68.2 Function/Subroutine Documentation

9.68.2.1 subroutine tnprep (real, dimension (ilg) *A1,* real, dimension (ilg) *A2,* real, dimension (ilg) *B1,* real, dimension (ilg) *B2,* real, dimension (ilg) *C2,* real, dimension(ilg) *GDENOM,* real, dimension(ilg) *GCOEFF,* real, dimension(ilg) *GCONST,* real, dimension(ilg) *CPHCHG,* integer, dimension(ilg) *IWATER,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg,ig) *TCTOP,* real, dimension (ilg,ig) *TCBOT,* real, dimension (ilg) *FI,* real, dimension (ilg) *ZPOND,* real, dimension(ilg) *TBAR1P,* real, dimension (ig) *DELZ,* real, dimension(ilg) *TCSNOW,* real, dimension (ilg) *ZSNOW,* integer, dimension (ilg,ig) *ISAND,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IG*)

Parameters

gdenom	Work array used in calculation of GCONST and GCOEFF
gcoeff	Multiplier used in equation relating ground surface heat flux to surface temperature $[Wm^{-2}K^{-1}]$
gconst	Intercept used in equation relating ground surface heat flux to surface temperature $[Wm^{-2}]$
cphchg	Latent heat of sublimation $[Jkg^{-1}]$
iwater	Flag indicating condition of surface (dry, water-covered or snow-covered)
tbar	Temperatures of soil layers, averaged over modelled area [K]
tctop	Thermal conductivity of soil at top of layer $[Wm^{-1}K^{-1}](\lambda_t)$
tcbot	Thermal conductivity of soil at bottom of layer $[Wm^{-1}K^{-1}](\lambda_b)$
fi	Fractional coverage of subarea in question on modelled area [ ]
zpond	Depth of ponded water on surface [m]
tbar1p	Lumped temperature of ponded water and first soil layer [K]
tcsnow	Thermal conductivity of snow $[Wm^{-1}K^{-1}]$
zsnow	Depth of snow pack [m]
isand	Sand content flag
delz	Overall thickness of soil layer $[m](\Delta_z)$

In this subroutine, coefficients are derived for an equation relating the heat flux at the ground surface to the ground surface temperature, using the average temperatures and the thermal conductivities of the underlying first three soil layers. It is assumed that the variation of temperature T with depth z within each soil layer can be modelled by using a quadratic equation:

$T(z) = (1/2)az^2 + bz + c$

By substituting 0 for z in the above equation and in the expressions for its first and second derivatives, it can be shown that $a = T''(0)$, $b = T'(0)$, and $c = T(0)$. The term $T''(0)$ can be evaluated from the expression for the first derivative evaluated at the bottom of the soil layer, $T(\Delta z)$:

$T''(0) = [T'(\Delta z) - T'(0)]/\Delta z$

The temperature gradient $T'(0)$ at the top of each layer is related to the heat flux G(0) through the thermal conductivity λ_t; and the temperature gradient and heat flux at the bottom of the layer, $G(\Delta z)$ and $T(\Delta z)$, are similarly related through the bottom thermal conductivity λ_b:

$G(0) = -\lambda_t T'(0)\ G(\Delta z) = -\lambda_b T'(\Delta z)$

The average soil layer temperature, $T_{av}(\Delta z)$, can be obtained by integrating the resulting equation for T(z) between 0 and Δz. Making use of all of the above expressions, recognizing that the heat fluxes and temperatures at the bottoms of layers 1 and 2 must equal the heat fluxes and temperatures at the tops of layers 2 and 3 respectively, and neglecting as a first approximation the heat flux at the bottom of the third layer, a linear equation can be derived relating G(0) to T(0) at the soil surface, where the slope and intercept of the equation are functions only of the average temperatures, thicknesses, and top and bottom thermal conductivities of the three soil layers.

In the subroutine loop, first the depth corresponding to TBAR1P (the lumped temperature of the first soil layer and the ponded water) is calculated, as the sum of the first soil layer thickness and the ponded water depth. If the ponded water depth is not vanishingly small, the surface water flag IWATER is set to 1; otherwise it is set to 0 for soils and 2 for ice sheets (indicated by ISAND = -4). If IWATER = 2, indicating a frozen water surface, the latent heat of vaporization, CPHCHG, is set to the value for sublimation (by adding the latent heat of melting to the latent heat of vaporization). If there is a snow pack present, the thermal conductivity at the top of the ground surface is calculated as the harmonic mean of the thermal conductivity at the top of the first soil layer and that of the snow pack. Finally, a series of work arrays is evaluated and is used to calculate the slope and intercept, GCOEFF and GCONST, of the equation relating G(0) to T(0) at the ground surface.

9.69 TPREP.f File Reference

Purpose: Initialize subarea variables and calculate various parameters for surface energy budget calculations.

Functions/Subroutines

- subroutine tprep (THLIQC, THLIQG, THICEC, THICEG, TBARC, TBARG, TBARCS, TBARGS, HCPC, HCPG, TCTOPC, TCBOTC, TCTOPG, TCBOTG, HCPSCS, HCPSGS, TCSNOW, TSNOCS, TSNOGS, WSNOCS, WSNOGS, RHOSCS, RHOSGS, TCANO, TCANS, CEVAP, IEVAP, TBAR1P, WTABLE, ZERO, EVAPC, EVAPCG, EVAPG, EVAPCS, EVPCSG, EVAPGS, GSNOWC, GSNOWG, GZEROC, GZEROG, GZROCS, GZROGS, QMELTC, QMELTG, EVAP, GSNOW, TPONDC, TPONDG, TPNDCS, TPNDGS, QSENSC, QSENSG, QEVAPC, QEVAPG, TACCO, QACCO, TACCS, QACCS, ILMOX, UEX, HBLX, ILMO, UE, HBL, ST, SU, SV, SQ, SRH, CDH, CDM, QSENS, QEVAP, QLWAVG, FSGV, FSGS, FSGG, FLGV, FLGS, FLGG, HFSC, HFSS, HFSG, HEVC, HEVS, HEVG, HMFC, HMFN, QFCF, QFCL, EVPPOT, ACOND, DRAG, THLIQ, THICE, TBAR, ZPOND, TPOND, THPOR, THLMIN, THLRET, THFC, HCPS, TCS, TA, RHOSNO, TSNOW, ZSNOW, WSNOW, TCAN, FC, FCS, DELZ, DELZW, ZBOTW, ISAND, ILG, IL1, IL2, JL, IG, FVEG, TCSATU, TCSATF, FTEMP, FTEMPX, FVAP, FVAPX, RIB, RIBX)

9.69.1 Detailed Description

Purpose: Initialize subarea variables and calculate various parameters for surface energy budget calculations.

9.69.2 Function/Subroutine Documentation

9.69.2.1 subroutine tprep (real, dimension(ilg,ig) *THLIQC,* real, dimension(ilg,ig) *THLIQG,* real, dimension(ilg,ig) *THICEC,* real, dimension(ilg,ig) *THICEG,* real, dimension (ilg,ig) *TBARC,* real, dimension (ilg,ig) *TBARG,* real, dimension(ilg,ig) *TBARCS,* real, dimension(ilg,ig) *TBARGS,* real, dimension (ilg,ig) *HCPC,* real, dimension (ilg,ig) *HCPG,* real, dimension(ilg,ig) *TCTOPC,* real, dimension(ilg,ig) *TCBOTC,* real, dimension(ilg,ig) *TCTOPG,* real, dimension(ilg,ig) *TCBOTG,* real, dimension(ilg) *HCPSCS,* real, dimension(ilg) *HCPSGS,* real, dimension(ilg) *TCSNOW,* real, dimension(ilg) *TSNOCS,* real, dimension(ilg) *TSNOGS,* real, dimension(ilg) *WSNOCS,* real, dimension(ilg) *WSNOGS,* real, dimension(ilg) *RHOSCS,* real, dimension(ilg) *RHOSGS,* real, dimension (ilg) *TCANO,* real, dimension (ilg) *TCANS,* real, dimension (ilg) *CEVAP,* integer, dimension (ilg) *IEVAP,* real, dimension(ilg) *TBAR1P,* real, dimension(ilg) *WTABLE,* real, dimension (ilg) *ZERO,* real, dimension (ilg) *EVAPC,* real, dimension(ilg) *EVAPCG,* real, dimension (ilg) *EVAPG,* real, dimension(ilg) *EVAPCS,* real, dimension(ilg) *EVPCSG,* real, dimension(ilg) *EVAPGS,* real, dimension(ilg) *GSNOWC,* real, dimension(ilg) *GSNOWG,* real, dimension(ilg) *GZEROC,* real, dimension(ilg) *GZEROG,* real, dimension(ilg) *GZROCS,* real, dimension(ilg) *GZROGS,* real, dimension(ilg) *QMELTC,* real, dimension(ilg) *QMELTG,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *GSNOW,* real, dimension(ilg) *TPONDC,* real, dimension(ilg) *TPONDG,* real, dimension(ilg) *TPNDCS,* real, dimension(ilg) *TPNDGS,* real, dimension(ilg) *QSENSC,* real, dimension(ilg) *QSENSG,* real, dimension(ilg) *QEVAPC,* real, dimension(ilg) *QEVAPG,* real, dimension (ilg) *TACCO,* real, dimension (ilg) *QACCO,* real, dimension (ilg) *TACCS,* real, dimension (ilg) *QACCS,* real, dimension (ilg) *ILMOX,* real, dimension (ilg) *UEX,* real, dimension (ilg) *HBLX,* real, dimension (ilg) *ILMO,* real, dimension (ilg) *UE,* real, dimension (ilg) *HBL,* real, dimension (ilg) *ST,* real, dimension (ilg) *SU,* real, dimension (ilg) *SV,* real, dimension (ilg) *SQ,* real, dimension (ilg) *SRH,* real, dimension (ilg) *CDH,* real, dimension (ilg) *CDM,* real, dimension (ilg) *QSENS,* real, dimension (ilg) *QEVAP,* real, dimension(ilg) *QLWAVG,* real, dimension (ilg) *FSGV,* real, dimension (ilg) *FSGS,* real, dimension (ilg) *FSGG,* real, dimension (ilg) *FLGV,* real, dimension (ilg) *FLGS,* real, dimension (ilg) *FLGG,* real, dimension (ilg) *HFSC,* real, dimension (ilg) *HFSS,* real, dimension (ilg) *HFSG,* real, dimension (ilg) *HEVC,* real, dimension (ilg) *HEVS,* real, dimension (ilg) *HEVG,* real, dimension (ilg) *HMFC,* real, dimension (ilg) *HMFN,* real, dimension (ilg) *QFCF,* real, dimension (ilg) *QFCL,* real, dimension(ilg) *EVPPOT,* real, dimension (ilg) *ACOND,* real, dimension (ilg) *DRAG,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *TBAR,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TPOND,* real, dimension(ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLMIN,* real, dimension(ilg,ig) *THLRET,* real, dimension (ilg,ig) *THFC,* real, dimension (ilg,ig) *HCPS,* real, dimension (ilg,ig) *TCS,* real, dimension (ilg) *TA,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *WSNOW,* real, dimension (ilg) *TCAN,* real, dimension (ilg) *FC,* real, dimension (ilg) *FCS,* real, dimension(ig) *DELZ,* real, dimension(ilg,ig) *DELZW,* real, dimension(ilg,ig) *ZBOTW,* integer, dimension (ilg,ig) *ISAND,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IG,* real, dimension (ilg) *FVEG,* real, dimension(ilg) *TCSATU,* real, dimension(ilg) *TCSATF,* real, dimension (ilg) *FTEMP,* real, dimension(ilg) *FTEMPX,* real, dimension (ilg) *FVAP,* real, dimension (ilg) *FVAPX,* real, dimension (ilg) *RIB,* real, dimension (ilg) *RIBX*)

Parameters

tbarc	Subarea temperatures of soil layers [C]
tbarg	Subarea temperatures of soil layers [C]
tbarcs	Subarea temperatures of soil layers [C]
tbargs	Subarea temperatures of soil layers [C]
thliqc	Liquid water content of soil layers under vegetation $[m^3m^{-3}]$
thliqg	Liquid water content of soil layers in bare areas $[m^3m^{-3}]$
thicec	Frozen water content of soil layers under vegetation $[m^3m^{-3}]$
thiceg	Frozen water content of soil layers in bare areas $[m^3m^{-3}]$
hcpc	Heat capacity of soil layers under vegetation $[Jm^{-3}K^{-1}](C_g)$
hcpg	Heat capacity of soil layers in bare areas $[Jm^{-3}K^1](Cg)$
tctopc	Thermal conductivity of soil at top of layer in canopy-covered subareas $[Wm^{-1}K^{-1}](\lambda)$
tcbotc	Thermal conductivity of soil at bottom of layer in canopy-covered subareas $[Wm^{-1}K^{-1}](\lambda)$
tctopg	Thermal conductivity of soil at top of layer in bare ground subareas $[Wm^{-1}K^{-1}](\lambda)$
tcbotg	Thermal conductivity of soil at bottom of layer in bare ground subareas $[Wm^{-1}K^{-1}](\lambda)$
hcpscs	Heat capacity of snow pack under vegetation canopy $[Jm^{-3}K^1](C_s)$
hcpsgs	Heat capacity of snow pack in bare areas $[Jm^{-3}K^1](C_s)$
tcsnow	Thermal conductivity of snow $[Wm^{-1}K^{-1}]$
tsnogs	Temperature of snow pack in bare areas [K]
tsnocs	Temperature of snow pack under vegetation canopy [K]
wsnocs	Liquid water content of snow pack under vegetation $[kgm^{-2}]$
wsnogs	Liquid water content of snow pack in bare areas $[kgm^{-2}]$
rhoscs	Density of snow pack under vegetation canopy $[kgm^{-3}]$
rhosgs	Density of snow pack in bare areas $[kgm^{-3}]$
tcano	Temperature of canopy over ground [K]
tcans	Temperature of canopy over snow [K]
cevap	Soil evaporation efficiency coefficient $[](\beta)$
tbar1p	Lumped temperature of ponded water and first soil layer [K]
wtable	Depth of water table in soil $[m](z_{wt})$
zero	Dummy vector containing all zeros
tpondc	Subarea temperature of surface ponded water [C]
tpondg	Subarea temperature of surface ponded water [C]
tpndcs	Subarea temperature of surface ponded water [C]
tpndgs	Subarea temperature of surface ponded water [C]
ievap	Flag indicating whether soil evaporation is occurring or not
evapc	Evaporation from vegetation over ground $[ms^{-1}]$
evapcg	Evaporation from ground under vegetation $[ms^{-1}]$
evapg	Evaporation from bare ground $[ms^{-1}]$
evapcs	Evaporation from vegetation over snow $[ms^{-1}]$
evpcsg	Evaporation from snow under vegetation $[ms^{-1}]$
evapgs	Evaporation from snow on bare ground $[ms^{-1}]$
gsnowc	Heat flux at top of snow pack under canopy $[Wm^{-2}]$
gsnowg	Heat flux at top of snow pack over bare ground $[Wm^{-2}]$

gzeroc	Subarea heat flux at soil surface $[Wm^{-2}]$
gzerog	Subarea heat flux at soil surface $[Wm^{-2}]$
gzrocs	Subarea heat flux at soil surface $[Wm^{-2}]$
gzrogs	Subarea heat flux at soil surface $[Wm^{-2}]$
qmeltc	Heat to be used for melting snow under canopy $[Wm^{-2}]$
qmeltg	Heat to be used for melting snow on bare ground $[Wm^{-2}]$
evap	Diagnosed total surface water vapour flux over modelled area $[kgm^{-2}s^{-1}]$
qsensc	Sensible heat flux from vegetation canopy over subarea $[Wm^{-2}]$
qsensg	Sensible heat flux from ground over subarea $[Wm^{-2}]$
qevapc	Latent heat flux from vegetation canopy over subarea $[Wm^{-2}]$
qevapg	Latent heat flux from ground over subarea $[Wm^{-2}]$
tacco	Temperature of air within vegetation canopy space over bare ground [K]
qacco	Specific humidity of air within vegetation canopy space over bare ground $[kgkg^{-1}]$
taccs	Temperature of air within vegetation canopy space over snow [K]
qaccs	Specific humidity of air within vegetation canopy space over snow $[kgkg^{-1}]$
st	Diagnosed screen-level air temperature [K]
su	Diagnosed anemometer-level zonal wind $[ms^{-1}]$
sv	Diagnosed anemometer-level meridional wind $[ms^{-1}]$
sq	Diagnosed screen-level specific humidity $[kgkg^{-1}]$
cdh	Surface drag coefficient for heat [ ]
cdm	Surface drag coefficient for momentum [ ]
qsens	Diagnosed total surface sensible heat flux over modelled area $[Wm^{-2}]$
qevap	Diagnosed total surface latent heat flux over modelled area $[Wm^{-2}]$
qlwavg	Upwelling longwave radiation over modelled area $[Wm^{-2}]$
fsgv	Diagnosed net shortwave radiation on vegetation canopy $[Wm^{-2}]$
fsgs	Diagnosed net shortwave radiation at snow surface $[Wm^{-2}]$
fsgg	Diagnosed net shortwave radiation at soil surface $[Wm^{-2}]$
flgv	Diagnosed net longwave radiation on vegetation canopy $[Wm^{-2}]$
flgs	Diagnosed net longwave radiation at snow surface $[Wm^{-2}]$
flgg	Diagnosed net longwave radiation at soil surface $[Wm^{-2}]$
hfsc	Diagnosed sensible heat flux on vegetation canopy $[Wm^{-2}]$
hfss	Diagnosed sensible heat flux at snow surface $[Wm^{-2}]$
hfsg	Diagnosed sensible heat flux at soil surface $[Wm^{-2}]$
hevc	Diagnosed latent heat flux on vegetation canopy $[Wm^{-2}]$
hevs	Diagnosed latent heat flux at snow surface $[Wm^{-2}]$
hevg	Diagnosed latent heat flux at soil surface $[Wm^{-2}]$
hmfc	Diagnosed energy associated with phase change of water on vegetation $[Wm^{-2}]$
hmfn	Diagnosed energy associated with phase change of water in snow pack $[Wm^{-2}]$
evppot	Diagnosed potential evapotranspiration $[kgm^{-2}s^{-1}]$
acond	Diagnosed product of drag coefficient and wind speed over modelled area $[ms^{-1}]$
drag	Surface drag coefficient under neutral stability [ ]
ilmo	Surface drag coefficient under neutral stability [ ]
ue	Friction velocity of air $[ms^{-1}]$
hbl	Height of the atmospheric boundary layer [m]
ilmox	Inverse of Monin-Obukhov roughness length over each subarea $[m^{-1}]$
uex	Friction velocity of air over each subarea $[ms^{-1}]$
hblx	Height of the atmospheric boundary layer over each subarea [m]

thliq	Volumetric liquid water content of soil layers $[m^3m^{-3}](\theta_l)$
thice	Volumetric frozen water content of soil layers $[m^3m^{-3}](\theta_i)$
tbar	Temperature of soil layers [K]
zpond	Depth of ponded water on surface [m]
tpond	Temperature of ponded water [K]
ta	Air temperature at reference height [K]
rhosno	Density of snow $[kgm^{-3}](\rho_s)$
tsnow	Snowpack temperature [K]
zsnow	Depth of snow pack $[m](z_s)$
wsnow	Liquid water content of snow pack $[kgm^{-2}](w_s)$
tcan	Vegetation canopy temperature [K]
fc	Fractional coverage of canopy over bare ground for modelled area [ ]
fcs	Fractional coverage of canopy over snow for modelled area [ ]
thpor	Pore volume in soil layer $[m^3m^{-3}](\theta_p)$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}]$
thlret	Liquid water retention capacity for organic soil $[m^3m^{-3}](\theta_{ret})$
thfc	Field capacity $[m^3m^{-3}](theta_{fc})$
hcps	Heat capacity of soil material $[Jm^{-3}K^{-1}](C_m)$
tcs	Thermal conductivity of soil particles $[Wm^{-1}K^{-1}](\theta_s)$
delzw	Permeable thickness of soil layer $[m](\Delta z_w)$
zbotw	Depth to permeable bottom of soil layer $[m](z_{b,w})$
delz	Overall thickness of soil layer [m]
isand	Sand content flag

In the first two loops, various subarea arrays and internal CLASST variables are initialized. The initial temperatures of the vegetation canopy above snow and above bare ground (TCANS and TCANO) are set to the temperature of the vegetation over the whole modelled area (TCAN) if TCAN is not effectively 0 K (the value it is assigned if vegetation is not present). Otherwise, the canopy temperatures are initialized to the air temperature TA.

In loop 200 the soil surface evaporation flag IEVAP and the evaporation efficiency coefficient CEVAP are assigned. If the liquid water content of the first soil layer is effectively equal to the minimum water content THLMIN, IEVAP and CEVAP are set to zero. If the liquid water content of the first soil layer is greater than the field capacity THFC, IEVAP and CEVAP are set to unity. Otherwise, IEVAP is set to 1 and CEVAP (or β as it is typically symbolized in the literature) is calculated using a relation presented by Lee and Pielke (1992):

$\beta = 0.25[1 \breve{} cos(\theta_l \pi / \theta_{fc})]^2$

where θ_l is the liquid water content of the first soil layer and θ_{fc} is its field capacity.

In loop 300 the volumetric heat capacities Cg of the soil layers under a bare surface (HCPG) and under vegetation (HCPC) are calculated, from their respective liquid and frozen water contents θ_l and θ_i:

$C_g = C_w\theta_l + C_i\theta_i + C_m(1 - \theta_p)$

where C_m is the heat capacity of the soil matter and θ_p is the pore volume. (The heat capacity of air is neglected.)

In loop 400, the thermal properties of the snow pack under the vegetation canopy and over bare soil are assigned on the basis of the properties of the snow pack over the whole modelled area. The heat capacity of the snow pack Cs is calculated from the volume fractions of snow particles and liquid water in the pack. The former is obtained from the ratio of the densities of the snow pack and ice, and the latter from the ratio of the liquid water content, normalized by the snow depth, and the density of water:

$C_s = C_i[\rho_s/\rho_i] + C_w w_s/[\rho_w z_s]$

The thermal conductivity of snow λ_s is obtained from the snow density using an empirical relationship derived by Sturm et al. (1997):

$\lambda_s = 3.233x10^{-6}\rho_s^2 \breve{} 1.01x10^{-3}\rho_s + 0.138\rho_s \geq 156.0$ = 0.234 x 10^{-3} + 0.023 $<$ 156.0

In loop 500, the thermal conductivities of the soil layers are assigned. If the ISAND flag for the first soil layer is -4 (indicating glacier or ice sheet), or if the ISAND flag is -3 (indicating rock), then literature values for glacier ice or sand particles respectively are assigned. If the ISAND flag is equal to -2, indicating organic soil, the depth of the water table z_{wt} is first calculated. This is taken to lie within the first layer, counting from the bottom of the soil profile, in which the soil water content is larger than the retention capacity θ_{ret}. The water table depth is deduced

by assuming that the soil is saturated below the water table, and that the water content is at the retention capacity above it. Thus, if $\theta_l + \theta_i = \theta_p$ for the soil layer, the water table is located at the top of the soil layer; if $\theta_l + \theta_i = \theta_{ret}$, it is located at the permeable bottom of the soil layer; and if $\theta_l + \theta_i$ is between these two values, its location is given by:

$z_{wt} = z_{b,w} - \Delta z_w[(\theta_l + \theta_i - \theta_{ret})/(\theta_p - \theta_{ret})]$

where Δz_w is the permeable thickness of the soil layer.

The thermal conductivities of organic and mineral soils are calculated following the analysis of Côté and Konrad (2005). They model the soil thermal conductivity λ using the concept of a relative thermal conductivity λ_r which has a value of 0 for dry soils and 1 at saturation:

$\lambda = [\lambda_{sat}{}^{\smile}\lambda_{dry}]\lambda_r + \lambda_{dry}$

The relative thermal conductivity is obtained from the degree of saturation (the water content divided by the pore volume) S_r, using the following generalized relationship:

$\lambda_r = \kappa S_r/[1 + (\kappa - 1)S_r]$

The empirical coefficient kappa takes the following values:

Unfrozen coarse mineral soils: $\kappa = 4.0$
Frozen coarse mineral soils: $\kappa = 1.2$
Unfrozen fine mineral soils: $\kappa = 1.9$
Frozen fine mineral soils: $\kappa = 0.85$
Unfrozen organic soils: $\kappa = 0.6$
Frozen organic soils: $\kappa = 0.25$

The dry thermal conductivity $lambda_{dry}$ is calculated using an empirical relationship based on the pore volume θ_p, with different coefficients for mineral and organic soils:

$\lambda_{dry} = 0.75exp(-2.76\theta_p)(mineral)\ \lambda_{dry} = 0.30exp(-2.0\theta_p)(organic)$

The saturated thermal conductivity λ_{sat} is calculated by Cote and Konrad as a geometric mean of the conductivities of the soil components. However, other researchers (e.g. Zhang et al., 2008) have found the linear averaging used by de Vries (1963) to be more generally accurate:

$lambda_{sat} = lambda_w\theta_p + \lambda_s(1 - \theta_p)(unfrozen)\ lambda_{sat} = lambda_i\theta_p + \lambda_s(1 - \theta_p)(frozen)$

where λ_w is the thermal conductivity of water, λ_i is that of ice and λ_s is that of the soil particles.

In the 500 loop, thermal conductivities are calculated for the top and bottom of each soil layer. The degree of saturation SATRAT is calculated as the sum of liquid and frozen water contents, θ_w and θ_i, divided by the pore volume. In organic soils, if the liquid water content of the soil layer is above the retention capacity θ_{ret}, θ_w at the top of the soil layer is assumed to be equal to θ_{re} and S_r at the bottom of the layer is assumed to be 1. The relative liquid and frozen water contents, THLSAT and THISAT, are calculated from θ_w and θ_i normalized by $\theta_w + \theta_i$. The dry thermal conductivity, and the saturated thermal conductivity for unfrozen and frozen conditions, are evaluated using the equations above. The unfrozen and frozen relative thermal conductivity, TCRATU and TCRATF, are obtained from SATRAT and the appropriate values of the empirical coefficient κ. For mineral soils, κ is obtained as a weighted average over the percent sand content (ISAND converted to a real value) and the percentage of fine material (assumed to be 100-ISAND). The unfrozen and frozen soil thermal conductivities, TCSOLU and T↩ CSOLF, are then calculated from TCRATU, TCRATF, and the dry and saturated thermal conductivities; and the actual thermal conductivity of the soil, TCSOIL, is determined as the average of TCSOLU and TCSOLF, weighted according to the relative liquid and frozen water contents THLSAT and THISAT. If the permeable thickness of the layer, DELZW, is greater than zero, the thermal conductivity at the top of the layer is set to TCSOIL; otherwise it is set to the rock value, TCSAND. If DELZW is less than the thermal thickness of the layer DELZ, the thermal conductivity at the bottom of the layer is set to TCSAND; otherwise it is set to TCSOIL. (In the case of organic soils in the latter branch, if θ_w was greater than θ_{ret}, the thermal conductivity at the bottom of the layer is set to the average of the saturated unfrozen and frozen values, weighted by THLSAT and THISAT.) Finally, if there is ponded water present on the soil surface, the thermal conductivity at the top of the first soil layer is treated as varying linearly from the calculated soil thermal conductivity if the pond depth ZPOND is zero, to the thermal conductivity of water if ZPOND $\geq 10^{-2}m$.

Finally, in loop 600, a variable TBAR1P is evaluated, representing the weighted average value of the first layer soil temperature and the ponded water, if any. (The heat capacity of the soil is determined as the weighted average

of HCPG over the permeable thickness DELZW, and the heat capacity of rock, HCPSND, over the impermeable thickness, DELZ-DELZW.)

9.70 TRIGL.f File Reference

THE ARGUMENT LIST IS THE SAME AS FOR GAUSSG. GAUSSG FILLS ONLY THE N HEM ORDERED N TO S. THIS ROUTINE MAKES THE ARRAYS GLOBAL AND ORDERED FROM S TO N.

Functions/Subroutines

- subroutine trigl (ILATH, SR, WR, CR, RADR, WOSQ)

9.70.1 Detailed Description

THE ARGUMENT LIST IS THE SAME AS FOR GAUSSG. GAUSSG FILLS ONLY THE N HEM ORDERED N TO S. THIS ROUTINE MAKES THE ARRAYS GLOBAL AND ORDERED FROM S TO N.

9.70.2 Function/Subroutine Documentation

9.70.2.1 subroutine trigl (*ILATH,* real∗8, dimension(lat) *SR,* real∗8, dimension(lat) *WR,* real∗8, dimension(lat) *CR,* real∗8, dimension(lat) *RADR,* real∗8, dimension(lat) *WOSQ*)

Parameters

sr	SIN(LAT), ANTISYMMETRIC
wr	GAUSSIAN WEIGHTS, SYMMETRIC ABOUT THE EQUATOR
cr	COS(LAT), SYMMETRIC ABOUT THE EQUATOR
radr	LATITUDE IN RADIANS
wosq	WR/(SR∗∗2), SYMMETRIC ABOUT THE EQUATOR

9.71 TSOLVC.f File Reference

Purpose: Solution of surface energy balance for vegetated subareas.

Functions/Subroutines

- subroutine tsolvc (ISNOW, FI, QSWNET, QSWNC, QSWNG, QLWOUT, QLWOC, QLWOG, QTRANS, QSENS, QSENSC, QSENSG, QEVAP, QEVAPC, QEVAPG, EVAPC, EVAPG, EVAP, TCAN, QCAN, TZERO, QZERO, GZERO, QMELTC, QMELTG, RAICAN, SNOCAN, CDH, CDM, RIB, TAC, QAC, CFLUX, FTEMP, FVAP, ILMO, UE, H, QFCF, QFCL, HTCC, QSWINV, QSWINI, QLWIN, TPOTA, TA, QA, VA, VAC, PADRY, RHOAIR, ALVISC, ALNIRC, ALVISG, ALNIRG, TRVISC, TRNIRC, FSVF, CRIB, CPHCHC, CPHCHG, CEVAP, TADP, TVIRTA, RC, RBCOEF, ZOSCLH, ZOSCLM, ZRSLFH, ZRSLFM, ZOH, ZOM, FCOR, GCONST, GCOEFF, TGND, TRSNOW, FSNOWC, FRAINC, CHCAP, CMASS, PCPR, FROOT, THLMIN, DELZW, RHOSNO, ZSNOW, IWATER, IEVAP, ITERCT, ISLFD, ITC, ITCG, ILG, IL1, IL2, JL, N, TSTEP, TVIRTC, TVIRTG, EVBETA, XEVAP, EVPWET, Q0SAT, RA, RB, RAGINV, RBINV, RBTINV, RBCINV, TVRTAC, TPOTG, RESID, TCANO, WZERO, XEVAPM, DCFLXM, WC, DRAGIN, CFLUXM, CFLX, IEVAPC, TRTOP, QSTOR, CFSENS, CFEVAP, QSGADD, A, B, LZZ0, LZZ0T, FM, FH, ITER, NITER, KF1, KF2, AILCG, FCANC, CO2CONC, RMATCTEM, THLIQ, THFC, THLW, ISAND, IG, COSZS, PRESSG, XDIFFUS, ICTEM, IC, CO2I1, CO2I2, ctem_on, SLAI, FCANCMX, L2MAX, NOL2PFTS, CFLUXV, ANVEG, RMLVEG, LFSTATUS, DAYL, DAYL_MAX)

9.71.1 Detailed Description

Purpose: Solution of surface energy balance for vegetated subareas.

9.71.2 Function/Subroutine Documentation

9.71.2.1 subroutine tsolvc (integer *ISNOW,* real, dimension (ilg) *FI,* real, dimension(ilg) *QSWNET,* real, dimension (ilg) *QSWNC,* real, dimension (ilg) *QSWNG,* real, dimension(ilg) *QLWOUT,* real, dimension (ilg) *QLWOC,* real, dimension (ilg) *QLWOG,* real, dimension(ilg) *QTRANS,* real, dimension (ilg) *QSENS,* real, dimension(ilg) *QSENSC,* real, dimension(ilg) *QSENSG,* real, dimension (ilg) *QEVAP,* real, dimension(ilg) *QEVAPC,* real, dimension(ilg) *QEVAPG,* real, dimension (ilg) *EVAPC,* real, dimension (ilg) *EVAPG,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *TCAN,* real, dimension (ilg) *QCAN,* real, dimension (ilg) *TZERO,* real, dimension (ilg) *QZERO,* real, dimension (ilg) *GZERO,* real, dimension(ilg) *QMELTC,* real, dimension(ilg) *QMELTG,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension (ilg) *CDH,* real, dimension (ilg) *CDM,* real, dimension (ilg) *RIB,* real, dimension (ilg) *TAC,* real, dimension (ilg) *QAC,* real, dimension (ilg) *CFLUX,* real, dimension (ilg) *FTEMP,* real, dimension (ilg) *FVAP,* real, dimension (ilg) *ILMO,* real, dimension (ilg) *UE,* real, dimension (ilg) *H,* real, dimension (ilg) *QFCF,* real, dimension (ilg) *QFCL,* real, dimension (ilg) *HTCC,* real, dimension(ilg) *QSWINV,* real, dimension(ilg) *QSWINI,* real, dimension (ilg) *QLWIN,* real, dimension (ilg) *TPOTA,* real, dimension (ilg) *TA,* real, dimension (ilg) *QA,* real, dimension (ilg) *VA,* real, dimension (ilg) *VAC,* real, dimension (ilg) *PADRY,* real, dimension(ilg) *RHOAIR,* real, dimension(ilg) *ALVISC,* real, dimension(ilg) *ALNIRC,* real, dimension(ilg) *ALVISG,* real, dimension(ilg) *ALNIRG,* real, dimension(ilg) *TRVISC,* real, dimension(ilg) *TRNIRC,* real, dimension (ilg) *FSVF,* real, dimension (ilg) *CRIB,* real, dimension(ilg) *CPHCHC,* real, dimension(ilg) *CPHCHG,* real, dimension (ilg) *CEVAP,* real, dimension (ilg) *TADP,* real, dimension(ilg) *TVIRTA,* real, dimension (ilg) *RC,* real, dimension(ilg) *RBCOEF,* real, dimension(ilg) *ZOSCLH,* real, dimension(ilg) *ZOSCLM,* real, dimension(ilg) *ZRSLFH,* real, dimension(ilg) *ZRSLFM,* real, dimension (ilg) *ZOH,* real, dimension (ilg) *ZOM,* real, dimension (ilg) *FCOR,* real, dimension(ilg) *GCONST,* real, dimension(ilg) *GCOEFF,* real, dimension (ilg) *TGND,* real, dimension(ilg) *TRSNOW,* real, dimension(ilg) *FSNOWC,* real, dimension(ilg) *FRAINC,* real, dimension (ilg) *CHCAP,* real, dimension (ilg) *CMASS,* real, dimension (ilg) *PCPR,* real, dimension (ilg,ig) *FROOT,* real, dimension(ilg,ig) *THLMIN,* real, dimension(ilg,ig) *DELZW,* real, dimension(ilg) *RHOSNO,* real, dimension (ilg) *ZSNOW,* integer, dimension(ilg) *IWATER,* integer, dimension (ilg) *IEVAP,* integer, dimension(ilg,6,50) *ITERCT,* integer *ISLFD,* integer *ITC,* integer *ITCG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N,* real, dimension (ilg) *TSTEP,* real, dimension(ilg) *TVIRTC,* real, dimension(ilg) *TVIRTG,* real, dimension(ilg) *EVBETA,* real, dimension (ilg) *XEVAP,* real, dimension(ilg) *EVPWET,* real, dimension (ilg) *Q0SAT,* real, dimension (ilg) *RA,* real, dimension (ilg) *RB,* real, dimension(ilg) *RAGINV,* real, dimension (ilg) *RBINV,* real, dimension(ilg) *RBTINV,* real, dimension(ilg) *RBCINV,* real, dimension(ilg) *TVRTAC,* real, dimension (ilg) *TPOTG,* real, dimension (ilg) *RESID,* real, dimension (ilg) *TCANO,* real, dimension (ilg) *WZERO,* real, dimension(ilg) *XEVAPM,* real, dimension(ilg) *DCFLXM,* real, dimension (ilg) *WC,* real, dimension(ilg) *DRAGIN,* real, dimension(ilg) *CFLUXM,* real, dimension (ilg) *CFLX,* integer, dimension(ilg) *IEVAPC,* real, dimension (ilg) *TRTOP,* real, dimension (ilg) *QSTOR,* real, dimension(ilg) *CFSENS,* real, dimension(ilg) *CFEVAP,* real, dimension(ilg) *QSGADD,* real, dimension (ilg) *A,* real, dimension (ilg) *B,* real, dimension (ilg) *LZZ0,* real, dimension (ilg) *LZZ0T,* real, dimension (ilg) *FM,* real, dimension (ilg) *FH,* integer, dimension (ilg) *ITER,* integer, dimension (ilg) *NITER,* integer, dimension (ilg) *KF1,* integer, dimension (ilg) *KF2,* real, dimension(ilg,ictem) *AILCG,* real, dimension(ilg,ictem) *FCANC,* real, dimension(ilg) *CO2CONC,* real, dimension(ilg,ictem,ig) *RMATCTEM,* real, dimension(ilg,ig) *THLIQ,* real, dimension(ilg,ig) *THFC,* real, dimension(ilg,ig) *THLW,* integer, dimension(ilg,ig) *ISAND,* integer *IG,* real, dimension(ilg) *COSZS,* real, dimension(ilg) *PRESSG,* real, dimension(ilg) *XDIFFUS,* integer *ICTEM,* integer *IC,* real, dimension(ilg,ictem) *CO2I1,* real, dimension(ilg,ictem) *CO2I2,* logical *ctem_on,* real, dimension(ilg,ictem) *SLAI,* real, dimension(ilg,ictem) *FCANCMX,* integer *L2MAX,* integer, dimension(ic) *NOL2PFTS,* real, dimension(ilg) *CFLUXV,* real, dimension(ilg,ictem) *ANVEG,* real, dimension(ilg,ictem) *RMLVEG,* integer, dimension(ilg,ictem) *LFSTATUS,* real, dimension(ilg) *DAYL,* real, dimension(ilg) *DAYL_MAX*)

Parameters

isnow	Flag indicating presence or absence of snow
qswnet	Total net shortwave radiation of canopy and underlying surface $[Wm^{-2}]$
qswnc	Net shortwave radiation on vegetation canopy $[Wm^{-2}](K_{*c})$
qswng	Net shortwave radiation at underlying surface $[Wm^{-2}](K_{*g})$

qlwout	Upwelling longwave radiation from canopy and underlying surface $[Wm^{-2}]$
qlwoc	Upwelling longwave radiation from vegetation canopy $[Wm^{-2}](L\uparrow_c, L\downarrow_c)$
qlwog	Upwelling longwave radiation from underlying surface $[Wm^{-2}](L\uparrow_g)$
qtrans	Shortwave radiation transmitted into surface $[Wm^{-2}]$
qsens	Sensible heat flux from canopy and underlying surface $[Wm^{-2}](Q_H)$
qsensc	Sensible heat flux from vegetation canopy $[Wm^{-2}](Q_{H,c})$
qsensg	Sensible heat flux from underlying surface $[Wm^{-2}](Q_{H,g})$
qevap	Latent heat flux from canopy and underlying surface $[Wm^{-2}](Q_E)$
qevapc	Latent heat flux from vegetation canopy $[Wm^{-2}](Q_{E,c})$
qevapg	Latent heat flux from underlying surface $[Wm^{-2}](Q_{E,g})$
evapc	Evaporation rate from vegetation $[kgm^{-2}s^{-1}](E_c)$
evapg	Evaporation rate from underlying surface $[kgm^{-2}s^{-1}](E(0))$
tcan	Vegetation canopy temperature $[K](T_c)$
qcan	Saturated specific humidity at canopy temperature $[kgkg^{-1}](q_c)$
tzero	Temperature at surface $[K](T(0))$
qzero	Specific humidity at surface $[kgkg^{-1}](q(0))$
gzero	Heat flux into surface $[Wm^{-2}](G(0))$
qmeltc	Heat available for melting snow or freezing water on the vegetation $[Wm^{-2}]$
qmeltg	Heat available for melting snow or freezing water on the underlying surface $[Wm^{-2}]$
raican	Intercepted liquid water stored on vegetation canopy $[kgm^{-2}]$
snocan	Intercepted frozen water stored on vegetation canopy $[kgm^{-2}]$
cdh	Surface drag coefficient for heat [ ]
cdm	Surface drag coefficient for momentum [ ]
rib	Bulk Richardson number at surface [ ]
tac	Temperature of air within vegetation canopy $[K](T_{ac})$
qac	Specific humidity of air within vegetation canopy space $[kgkg^{-1}](q_{ac})$
cflux	Product of surface drag coefficient and wind speed $[ms^{-1}]$
ftemp	Product of surface-air temperature gradient, drag coefficient and wind speed $[Kms^{-1}]$
fvap	Product of surface-air humidity gradient, drag coefficient and wind speed $[kgkg^{-1}ms^{-1}]$
ilmo	Inverse of Monin-Obukhov roughness length (m^{-1})
ue	Friction velocity of air $[ms^{-1}]$
h	Height of the atmospheric boundary layer [m]
qfcf	Sublimation from frozen water on vegetation $[kgm^{-2}s^{-1}]$
qfcl	Evaporation from liquid water on vegetation $[kgm^{-2}s^{-1}]$
htcc	Internal energy change of canopy due to changes in temperature and/or mass $[Wm^{-2}]$
evap	Diagnosed total surface water vapour flux over modelled area $[kgm^{-2}s^{-1}]$
fi	Fractional coverage of subarea in question on modelled area [ ]
qswinv	Visible radiation incident on horizontal surface $[Wm^{-2}](K\downarrow)$
qswini	Near-infrared radiation incident on horizontal surface $[Wm^{-2}](K\downarrow)$
qlwin	Downwelling longwave radiation at bottom of atmosphere $[Wm^{-2}](L\downarrow)$
tpota	Potential temperature of air at reference height $[K](T_{a,pot})$
ta	Air temperature at reference height [K]
qa	Specific humidity at reference height $[kgkg^{-1}](q_a)$
va	Wind speed at reference height $[ms^{-1}]$
vac	Wind speed within vegetation canopy space $[ms^{-1}](v_{ac})$
padry	Partial pressure of dry air $[Pa](p_{dry})$
rhoair	Density of air $[kgm^{-3}](\rho_a)$

alvisc	Visible albedo of vegetation canopy $[](\alpha_c)$
alnirc	Near-IR albedo of vegetation canopy $[](\alpha_c)$
alvisg	Visible albedo of underlying surface $[](\alpha_g)$
alnirg	Near-IR albedo of underlying surface $[](\alpha_g)$
trvisc	Visible transmissivity of vegetation canopy $[](\tau_c)$
trnirc	Near-IR transmissivity of vegetation canopy $[](\tau_c)$
fsvf	Sky view factor of ground underlying canopy $[](\chi)$
crib	Richardson number coefficient $[K^{-1}]$
cphchc	Latent heat of vaporization on vegetation canopy $[Jkg^{-1}](L_v)$
cphchg	Latent heat of vaporization on underlying ground $[Jkg^{-1}](L_v)$
cevap	Soil evaporation efficiency coefficient [ ]
tadp	Dew point of air at reference height [K]
tvirta	Virtual potential temperature of air at reference height $[K](T_{a,v})$
rc	Stomatal resistance of vegetation $[sm^{-1}](r_c)$
rbcoef	Parameter for calculation of leaf boundary resistance
zosclh	Ratio of roughness length for heat to reference height for temperature and humidity [ ]
zosclm	Ratio of roughness length for momentum to reference height for wind speed [ ]
zrslfh	Difference between reference height for temperature and humidity and height at which extrapolated wind speed goes to zero [m]
zrslfm	Difference between reference height for wind speed and height at which extrapolated wind speed goes to zero [m]
zoh	Surface roughness length for heat [m]
zom	Surface roughness length for momentum $[m](z_{0,m})$
fcor	Coriolis parameter $[s_{-1}]$
gconst	Intercept used in equation relating surface heat flux to surface temperature $[Wm^{-2}]$
gcoeff	Multiplier used in equation relating surface heat flux to surface temperature $[Wm^{-2}K^{-1}]$
tgnd	Starting point for surface temperature iteration [K]
trsnow	Short-wave transmissivity of snow pack [ ]
fsnowc	Fractional coverage of canopy by frozen water $[](F_s)$
frainc	Fractional coverage of canopy by liquid water $[](F_l)$
chcap	Heat capacity of vegetation canopy $[Jm^{-2}K^{-1}](C_c)$
cmass	Mass of vegetation canopy $[kgm^{-2}]$
pcpr	Surface precipitation rate $[kgm^{-2}s^{-1}]$
iwater	Flag indicating condition of surface (dry, water-covered or snow-covered)
ievap	Flag indicating whether surface evaporation is occurring or not output variable
iterct	Counter of number of iterations required to solve energy balance for four subareas

For the subcanopy surface temperature iteration, two alternative schemes are offered: the bisection method (selected if the flag ITCG = 1) and the Newton-Raphson method (selected if ITCG = 2). In the first case, the maximum number of iterations ITERMX is set to 12, and in the second case it is set to 5. An optional windless transfer coefficient EZERO is made available, which can be used, following the recommendations of Brown et al. (2006), to prevent the sensible heat flux over snow packs from becoming vanishingly small under highly stable conditions. If the snow cover flag ISNOW is zero (indicating bare ground), EZERO is set to zero; if ISNOW=1, EZERO is set to $2.0Wm^{-2}K^{-1}$. The surface transfer coefficient under conditions of free convection, RAGCO, is set to $1.9x10^{-3}$ (see the calculation of RAGINV below).

In the 50 loop, some preliminary calculations are done. The shortwave transmissivity at the surface, TRTOP, is set to zero in the absence of a snow pack, and to the transmissivity of snow, TRSNOW, otherwise. The net shortwave radiation at the surface, QSWNG, is calculated as the sum of the net visible and net near-infrared shortwave radiation. Each of these is obtained as: $K_{*g} = K \downarrow \tau_c[1 - \alpha_g]$ where $K_{*g}isthenetradiationatthesurface$,K $istheincomingshortwaveradiationabovethecanopy$,$ is the canopy transmissivity and α_g is the surface albedo. This average value is corrected for the amount of radiation transmitted into the surface, QTRANS, obtained using TRTOP. The net shortwave radiation for the vegetation canopy, QSWNC, is calculated as the sum of the net visible and net near-infrared shortwave radiation. Each of these is determined as: $K_{*c} = K \downarrow [1 - \alpha_c] - K_{*g}$ where $K_{*c}isthenetradiationonthecanopyand$$ is the canopy albedo. If the canopy temperature is essentially 0 K, indicating that a canopy was not present in the previous time step but has now appeared, the canopy temperature is initialized to the potential temperature of the air, TPOTA. The outgoing longwave radiation emitted upward $(L \uparrow_c)$ or downward $(L \downarrow_c)$ from the canopy is calculated using the standard Stefan-Boltzmann equation: $L \uparrow_c = L \downarrow_c = \sigma T_c^4$

where σ is the Stefan-Boltzmann constant and T_c is the canopy temperature.

Virtual temperature is defined as temperature adjusted for the reduction in air density due to the presence of water vapour. This is applied in order to enable the use of the equation of state for dry air. The virtual temperature can be approximated by multiplying the actual temperature by a factor [1 + 0.61 q], where q is the specific humidity of the air in question. Thus, the virtual temperature of air at the vegetation canopy temperature, $T_{c,v}$, is obtained as: $T_{c,v} = T_c[1 + 0.61q_c]$ where q_c is the saturated specific humidity at the canopy temperature. This is determined from the saturation mixing ratio at the canopy temperature, w_c : $q_c = w_c/[1 + w_c]$ The saturation mixing ratio is a function of the saturation vapour pressure e_c at the canopy temperature: $w_c = 0.622e_c/(p_{dry})$ where p_{dry} is the partial pressure of dry air. A standard empirical equation for the saturation vapour pressure dependence on the temperature T is used:

$e_{sat} = 611.0exp[17.269(T - T_f)/(T - 35.86)]T \geq T_f$ $e_{sat} = 611.0exp[21.874(T - T_f)/(T - 7.66)]T < T_f$

where T_f is the freezing point. The virtual temperature of the air in the canopy space, $T_{ac,v}$, is likewise calculated from the canopy air temperature T_{ac} and the specific humidity in the canopy air space, q_{ac} , as

$T_{ac,v} = T_{ac}[1 + 0.61q_{ac}]$

If the Newton-Raphson method is being used for the canopy temperature iteration (pre-selected by setting the flag ITC to 2), the temperature of the air in the canopy space is approximated as the canopy temperature, and the specific humidity in the canopy air space is approximated as the specific humidity of the air above the canopy.

If there is intercepted snow on the vegetation, the latent heat associated with water flux from the canopy, CPHCHC, is set to the latent heat of sublimation (the sum of the heat of vaporization and the heat of melting). Otherwise the latent heat is set to that of vaporization alone. The leaf boundary layer resistance RB, and its inverse RBINV, are calculated using the wind speed in the canopy air space, VAC, and a coefficient RBCOEF evaluated in subroutine APREP. This coefficient is formulated after Bartlett (2004), who developed an expression for the inverse of the leaf boundary resistance, $1/r_b$, drawing on the analysis of Bonan (1996) and McNaughton and van den Hurk (1995), of the form:

$1/r_b = v_{ac}^{1/2}\sigma f_i\gamma_i\Lambda_i^{1/2}/0.75[1 - exp(-0.75\Lambda_i^{1/2})]$

where v_{ac} is the wind speed in the canopy air space, f_i is the fractional coverage of each vegetation type i over the subarea in question, Λ_i is its leaf area index, and γ_i is a vegetation-dependent parameter which incorporates the effects of leaf dimension and sheltering. The initial value of the surface temperature TZERO is set to TGND, which contains the value of TZERO from the previous time step, and the initial temperature of the canopy, TCANO, is set to the current canopy temperature. The first step in the iteration sequence, TSTEP, is set to 1.0 K. The flag ITER is set to 1 for each element of the set of modelled areas, indicating that its surface temperature has not yet been found. The iteration counter NITER is initialized to 1 for each element. Initial values are assigned to other variables.

(At this point in the code, if CLASS is being run in coupled mode with CTEM, the CTEM subroutine PHTSYN3 is called. These lines are commented out for uncoupled runs.)

The 100 continuation line marks the beginning of the surface temperature iteration sequence. First the flag NUMIT (indicating whether there are still locations at the end of the current iteration step for which the surface temperature has not yet been found) is set to zero. Loop 125 is then performed over the vector of modelled areas. If ITER=1, the saturated specific humidity at the ground surface temperature, $q(0)_{sat}$, is calculated using the same method as that outlined above for the saturated specific humidity at the canopy temperature. If there is a snow cover or ponded water present on the surface (IWATER $>$ 0), the surface evaporation efficiency EVBETA is set to 1 and the surface specific humidity q(0) is set to $q(0)_{sat}$. Otherwise EVBETA is set to CEVAP, the value obtained in subroutine TPREP on the basis of ambient conditions, and q(0) is calculated from $q(0)_{sat}$ by making use of the definition of the surface evaporation efficiency β:

$\beta = [q(0) - q_{ac}]/[q(0)_{sat} - q_{ac}]$

which is inverted to obtain an expression for q(0). If $q(0) > q_{ac}$ and the evaporation flag IEVAP has been set to zero, EVBETA is reset to zero and q(0) is reset to q_{ac}.

Next, the potential temperature of air at the ground surface temperature, $T(0)_{pot}$, is calculated relative to the sum of the displacement height d and the roughness length for momentum $z_{0,m}$, the height at which the canopy temperature is assumed to apply. (This is also the height corresponding to the potential temperature of the air at the reference height above the canopy). $T(0)_{pot}$ is found using the dry adiabatic lapse rate, $dT/dz = -g/c_p$, where g is the acceleration due to gravity and c_p is the specific heat at constant pressure. Since the displacement height d is assumed to lie at 0.7 of the canopy height, and the roughness length for momentum is assumed to lie at 0.1 of

the canopy height, their sum can be obtained as $8.0z_{0,m}$. Thus, $T(0)_{pot} = T(0) - 8.0z_{0,m}g/c_p$

The virtual potential temperature at the surface is obtained using the same expression as above: $T(0)_v = T(0)_{pot}[1 + 0.61q(0)]$

Since wind speeds under the canopy are expected to be relatively small, it is assumed that under stable or neutral conditions, turbulent fluxes are negligible. If the virtual potential temperature of the surface is more than 1 K greater than that of the canopy air, free convection conditions are assumed. Townsend's (1964) equation for the surface-air transfer coefficient, or the inverse of the surface resistance $r_{a,,g}$, is used in a form derived from the analysis of Deardorff (1972): $1/r_{a,,g} = 1.9x10^{-3}[T(0)_v - T_{ac,v}]^{1/3}$

The first derivative of the transfer coefficient with respect to the surface temperature is calculated for use with the Newton-Raphson iteration scheme: $d(1/r_{a,,g})/dT = 1.9x10^{-3}[T(0)_v - T_{ac,v}]^{-2/3}/3$

If the virtual potential temperature of the surface is between 0.001 and 1 K greater than that of the canopy air, a simple diffusion relation is assumed: $1/r_{a,,g} = 1.9x10^{-3}[T(0)_v - T_{ac,v}]$ Thus, $d(1/r_{a,,g})/dT = 1.9x10^{-3}$

The remaining terms of the surface energy balance are now evaluated. The energy balance equation is expressed as: $K_{*g}+L_{*g}-Q_{H,g}-Q_{E,g}-G(0) = 0$ where K_{*g} is the net surface shortwave radiation, L_{*g} is the net longwave radiation, $Q_{H,g}$ is the sensible heat flux, $Q_{E,g}$ is the latent heat flux, and G(0) is the conduction into the surface. K_{*g} was evaluated earlier in loop 50. L_{*g} is calculated as the difference between the downwelling radiation $L\downarrow_g$ at the surface and the upwelling radiation $L\uparrow_g$. The downwelling radiation incident on the surface is determined from the downwelling sky radiation above the canopy $L\downarrow$ and the downwelling radiation from the canopy itself, $L\downarrow_c$, weighted according to the sky view factor χ: $L\downarrow_g = \chi L\downarrow + [1-\chi]L\downarrow_c$ The upwelling radiation is obtained using the Stefan-Boltzmann equation: $L\uparrow_g = \sigma T(0)^4$

The sensible heat flux is given by $Q_{H,g} = \rho_a c_p[T(0)_{pot} - T_{a,c}]/r_{a,,g}$ where ρ_a is the density of the air and c_p is its specific heat. (The windless transfer coefficient defined at the beginning of the subroutine is currently not used under vegetation canopies.) The evaporation rate at the surface, E(0), is calculated as $E(0) = \rho_a[q(0) - q_{a,c}]/r_{a,,g}$

Q_E is obtained by multiplying E(0) by the latent heat of vaporization at the surface. The heat flux into the surface G(0) is determined as a linear function of T(0) (see documentation for subroutines TNPREP and TSPREP). It can be seen that each of the terms of the surface energy balance is a function of a single unknown, T(0) or TZERO. The residual RESID of the energy balance is now evaluated on the basis of the current estimation for TZERO. If the absolute value of RESID is less than $5.0Wm^{-2}$, or if the absolute value of the iteration step TSTEP most recently used is less than 0.01 K, the surface temperature is deemed to have been found and ITER is set to 0. If the iteration counter NITER is equal to the maximum number and ITER is still 1, ITER is set to -1.

In the following section, the iteration sequence is moved ahead a step. If ITCG = 1, the calculations for the bisection method of solution in loop 150 are performed, over each of the modelled areas for which ITER = 1. If NITER = 1 (indicating that this is the first step in the iteration), then if RESID $>$ 0 (indicating that the current value for TZERO had undershot the correct value), TZERO is incremented by 1 K; otherwise it is decremented by 1 K. If this is not the first step in the iteration, then if RESID $>$0 and TSTEP $<$ 0 (indicating that TZERO has undershot the correct value and the last temperature increment had been a negative one) or if RESID $<$ 0 and TSTEP $>$ 0 (indicating that TZERO has overshot the correct value and the last temperature increment had been a positive one), TSTEP is divided in half and its sign changed. TSTEP is then added to TZERO. The iteration counter NITER and the flag NUMIT are each incremented by one. Finally, if NUMIT $>$ 0, the iteration cycle is repeated from line 100 on.

If ITCG = 2, the calculations for the Newton-Raphson method of iteration in loop 175 are performed, over each of the modelled areas for which ITER = 1. In this approach, the value x_{n+1} used at each iteration step is obtained from the value x_n at the previous step as follows: $x_{n+1} = x_n - f(x_n)/f'(x_n)$

Identifying x n with TZERO and $f(x_n)$ with the surface energy balance equation, it can be seen that the second term on the right-hand side corresponds to TSTEP; the numerator is equal to RESID and the denominator to the first derivative of the energy balance equation evaluated at TZERO, which in turn is equal to the sum of the derivatives of the individual terms: f[d(L)/dT = -4 T(0)^3 f]f[d(Q_{H,g})/dT = c_p {1/r_{a,,g} + [T(0)_{pot} - T_{ac}] d(1/r_{a,,g})/dT} f]f[d(Q_{E,g})/dT = L_v {1/r_{a,,g} dq(0)/dT+ [q(0) - q_{ac}] d(1/r_{a,,g})/dT} f] and dG(0)/dT is equal to the coefficient multiplying TZERO in the equation for G(0). (L_v is the latent heat of vaporization at the surface.) At the end of the calculations the iteration counter NITER and the flag NUMIT are each incremented by one, and upon exiting the loop, if NUMIT $>$ 0, the iteration cycle is repeated from line 100 on.

After the iteration has been completed, if the Newton-Raphson method has been used, a check is carried out in loop 200 to ascertain whether convergence has not been reached (i.e. whether ITER = -1) for any location. In such cases it is assumed that conditions of near-neutral stability at the surface are the cause of the difficulty in finding a

solution. A trial value of TZERO is calculated using the virtual potential temperature of the canopy. If the absolute value of RESID is $> 15Wm^{-2}$, TZERO is set to this trial value. The values of q(0) and the components of the surface energy balance are recalculated as above, except that $Q_{H,g}$ and $Q_{E,g}$ are assumed as a first approximation to be zero. RESID is determined on this basis, and is then partitioned between $Q_{H,g}$ and $Q_{E,g}$ on the basis of the Bowen ratio B, the ratio of $Q_{H,g}$ over $Q_{E,g}$. Setting the residual R equal to $Q_{H,g} + Q_{E,g}$, and substituting for $Q_{H,g}$ using $B = Q_{H,g}/Q_{E,g}$, results in: $Q_{E,g} = R/(1 + B)$ $Q_{H,g}$ is then obtained as $R - Q_{E,g}$, the residual is reset to zero, and E(0) is recalculated from $Q_{E,g}$.

At this point a check is performed for unphysical values of the surface temperature, i.e. for values greater than 100 C or less than -100 C. If such values are encountered, an error message is printed and a call to abort is carried out.

Finally, clean-up calculations are performed in loop 250. A check is carried out to ensure that TZERO is not less than 0 C if ponded water is present on the surface (IWATER = 1) or greater than 0 C if snow is present on the surface (IWATER = 2). If either is the case, TZERO is reset to the freezing point, and q(0), $T(0)_{pot}$ and $T(0)_v$ are re-evaluated. The components of the surface energy balance are recalculated using the above equations. The residual of the energy balance equation is assigned to the energy associated with phase change of water at the surface, QMELTG, and RESID is set to zero.

In the last part of the loop, some final adjustments are made to a few other variables. If the evaporation flux is vanishingly small, it is added to RESID and reset to zero. If both RESID and $Q_{E,g}$ are not small, and if the precipitation rate is vanishingly small, RESID is added to Q E,g ; otherwise RESID is added to $Q_{H,g}$. Lastly, the iteration counter ITERCT is updated for the level corresponding to the subarea type and the value of NITER.

In the 300 loop, preliminary calculations are done in preparation for the canopy temperature iteration. The sensible heat flux from the surface is treated differently if ITC = 1 (bisection method of solution) and ITC = 2 (Newton-↩ Raphson method of solution). In the first instance the surface sensible heat flux is applied to heating the air in the canopy space; in the second, the canopy and the air space within it are treated as one aggregated mass, and the sensible heat flux from below is assumed to be added to its energy balance. Thus, if ITC = 1, the sensible heat flux that is added to the canopy, QSGADD, is set to 0. If ITC = 2, it is set to $Q_{H,g}$; and T_{ac} is set to T_c , q_{ac} to q_c , and $T_{ac,v}$ to $T_{c,v}$. The flag ITER is set to 1 for each element of the set of modelled areas, indicating that its surface temperature has not yet been found. The iteration counter NITER is initialized to 1 for each element. The first step in the iteration sequence, TSTEP, is set to 1.0 K. Initial values are assigned to other variables. After exiting the loop, the maximum number of iterations ITERMX is set to 12 if ITC = 1, and to 5 if ITC = 2.

The 400 continuation line marks the beginning of the canopy temperature iteration sequence. First the flags N↩ IT (indicating whether there are still locations at the beginning of the current iteration step for which the surface temperature has not yet been found) and NUMIT (indicating whether there are still locations at the end of the current iteration step for which the surface temperature has not yet been found) are set to zero. Loop 450 is then performed over the set of modelled areas. If ITER=1, NIT is incremented by one; if ITC = 1, the vegetation virtual temperature $T_{c,v}$ is recalculated.

If NIT > 0, a subroutine is called to evaluate the stability-corrected surface drag coefficients for heat and momentum. The subroutine selection is made on the basis of the flag ISLFD. If ISLFD=1, indicating that the calculations are to be consistent with CCCma conventions, subroutine DRCOEF is called; if ISLFD=2, indicating that the calculations are to be consistent with RPN conventions, subroutine FLXSURFZ is called.

Next, canopy parameters and turbulent transfer coefficients are calculated prior to evaluating the terms of the surface energy balance equation. If ITC = 1, the analysis of Garratt (1992) is followed. The sensible and latent heat fluxes from the canopy air to the overlying atmosphere, Q_H and Q_E , are obtained as the sums of the sensible and latent heat fluxes from the canopy to the canopy air, $Q_{H,c}$ and $Q_{E,c}$, and from the underlying surface to the canopy air, $Q_{H,g}$ and $Q_{E,g}$:

$$Q_H = Q_{H,c} + Q_{H,g}$$

$$Q_E = Q_{E,c} + Q_{E,g}$$

where

$$Q_H = \rho_a c_p [T_{a,c} - T_{a,pot},]/r_a$$

$$Q_{H,c} = \rho_a c_p [T_c - T_{a,c},]/r_b$$

and

$$Q_E = L_v \rho_a [q_{a,c} - q_{a,}]/r_a$$

$Q_{E,c} = L_v \rho_a [q_{,c} - q_{a,c,}]/(r_b + r_c)$

The equations for the sensible and latent heat fluxes from the surface were presented above. In these expressions, $T_{a,pot} and$q_a $are the potential temperature and specific humidity of the air overlying the canopy, is the aerodynamic resistance to turbulent transfer between the canopy air and the overlying air, and is the stomatal resistance$ _a .$) Thus,$T_{a,c} andq_{a,c} $can be evaluated as${a,c} = [T_{a,pot} /r_a + T_c /r_b + T(0)_{pot} /r_{a„g}]/[1/r_a + 1/r_b + 1/r_{a„g}] {a,c} = [q_a /r_a + q_c /(r_b + r_c) + q(0)/r_{a„g}]/[1/r_a + 1/(r_b + r_c) + 1/r_{a„g}] $If the water vapour flux is towards the canopy leaves, or if the canopy is snow - covered or rain - covered, is zero. If the water vapour flux is away from the canopy leaves and the fractional snow or water coverage on the canopy, or$F_l /(r_b + r_c) $adjusted for the presence of intercepted snow or water, (XEVAP in the code) is calculated as a weighted averag$ _s > 0 $, the canopy must be at a temperature of 0C or less, and so it is deduced that no transpiration can be occurring. Thus,$ = (F_s + F_l)/r_b $. Otherwise, is calculated on the assumption that the resistance is equal to$r_b $over the water - covered area and$ + r_c $over the rest :$ = F_l /r_b + [1 - F_l]/[r_b + r_c]

In the 450 loop, XEVAP is first set as a trial value to RBINV (neglecting stomatal resistance), and QAC is calculated using this value. If $q_{a,c} < q_c$, indicating that the vapour flux is away from the canopy leaves, XEVAP is recalculated as above and QAC is re-evaluated. Otherwise, if $F_s > 0$, XEVAP is scaled by F_s , since it is assumed that snow on the canopy will be at a lower temperature than the canopy itself, and deposition via sublimation will occur preferentially onto it. $T_{a,c}$ and $T_{ac,v}$ are calculated as above, and the canopy-air turbulent transfer coefficients for sensible and latent heat flux, CFSENS and CFEVAP, are set to RBINV and XEVAP respectively.

If ITC = 2, the canopy parameters and turbulent transfer coefficients are calculated, as noted above, on the basis of the assumption that the vegetation canopy and the air space within it can be treated as one aggregated mass, with a single representative temperature. Thus, the resistances r_a and r_b are considered as acting in series upon the sensible and latent heat fluxes between the canopy and the overlying air:

$Q_{H,c} = \rho_a c_p [T_{,c} - T_{a,pot,}]/(r_a + r_b)$

$Q_{E,c} = L_v \rho_a [q_{,c} - q_{a,}]/(r_a + r_b + r_c)$

In loop 500 the term $1/(r_a + r_b)$ is calculated from the variables CFLUX (the inverse of r_a) and RBINV (the inverse of r_b), and assigned to the temporary variable CFLX. If the incoming visible shortwave radiation QSWINV is greater than or equal to $25 W m^{-2}$, the calculated value of CFLX is retained; if QSWINV is zero, it is reset to CFLUX; and between these two limits it varies linearly between the two.

Thus the effect of the calculated leaf boundary resistance is suppressed during conditions of zero or low solar heating. (This is done to avoid unrealistically low calculated turbulent fluxes at night, which can lead to anomalously low canopy temperatures.) The overall aerodynamic resistance $r_A = r_a + r_b$ is obtained as 1/CFLX and assigned to the variable RA. As with ITC = 1, if $q_a < q_c$, XEVAP is recalculated as above (except that r_A is substituted for r_b). If $F_s > 0$, XEVAP is again scaled by F_s .

In the approach used here, the specific humidity of the aggregated canopy $q_{0,c}$ is not assumed to be equal to the saturated specific humidity at the canopy temperature, but is rather determined using X_E . If the two methods of calculating $Q_{E,c}$ are assumed to be analogous:

$Q_{E,c} = L_v \rho_a X_E [q_{,c} - q_{a,}]$

$Q_{E,c} = L_v \rho_a [q_{,0,c} - q_{a,}]/r_A$

then solving for $q_{0,c}$ leads to the expression

$q_{0,c} = r_A X_E q_c + [1 - r_A X_E] q_a$

In the second part of the 500 loop the saturated specific humidity of the canopy is calculated as before and adjusted using the above equation to obtain the specific humidity of the aggregated canopy. This is then used to calculate $T_{c,v}$. The canopy-air turbulent transfer coefficients for sensible and latent heat flux, CFSENS and CFEVAP, are both set to CFLX, and for calculation purposes in the following loop $T_{a,c}$ is set to the potential temperature of the air above the canopy, and $q_{a,c}$ to the specific humidity of the air above the canopy.

In the 525 loop, the terms of the canopy energy balance are evaluated. The energy balance is expressed as:

$K_{*c} + L_{*c} - Q_{H,c} + Q_{H,g+} - Q_{E,c} - \Delta Q_{S,c} = 0$

(The term $Q_{H,g+}$ corresponds to the variable QSGADD discussed above; the term $\Delta Q_{S,c}$ represents the change of energy storage in the canopy.) The net shortwave radiation K_{*c} was evaluated earlier in the 50 loop. The net longwave radiation is obtained as:

$L_{*c} = (1 - \chi)[L\downarrow + L\uparrow_g - 2L\downarrow_c]$

$Q_{H,c}$ is calculated as above. If there is intercepted liquid or frozen water on the canopy, or if the vapour flux is downward, or if the stomatal resistance is less than a limiting high value of $5000sm^{-1}$, the canopy vapour flux E_c (that is, $Q_{E,c}/L_v$) is calculated as above and the flag IEVAPC is set to 1; otherwise E_c and IEVAPC are both set to zero and q_c is set to q_a . If the water vapour flux is towards the canopy and the canopy temperature is greater than the air dew point temperature, the flux is set to zero. If there is intercepted water on the canopy, a limiting evaporation flux EVPWET is calculated as the rate required to sublimate all of the intercepted snow if $F_s > 0$, or all of the intercepted rain otherwise. If the canopy is more than half covered by intercepted water and the calculated canopy vapour flux is greater than EVPWET, it is reset to EVPWET and IEVAPC is set to zero. $Q_{E,c}$ is calculated from E_c and $\Delta Q_{S,c}$ is obtained as

$\Delta Q_{S,c} = C_c[T_c - T_{c,o}]/\Delta t$

where C_c is the canopy heat capacity, $T_{c,o}$ is the canopy temperature from the previous time step and Δt is the length of the time step. The residual RESID of the energy balance is evaluated on the basis of the current estimation for the canopy temperature TCAN. If the absolute value of RESID is less than $5.0Wm^{-2}$, or if the absolute value of the iteration step TSTEP most recently used is less than 0.01 K, the surface temperature is deemed to have been found and ITER is set to 0. If the iteration counter NITER is equal to the maximum number and ITER is still 1, ITER is set to -1.

In the following section, the iteration sequence is moved ahead a step. If ITC = 1, the calculations for the bisection method of solution in loop 550 are performed, over each of the modelled areas for which ITER = 1. If NITER = 1 (indicating that this is the first step in the iteration), then if RESID $>$ 0 (indicating that the current value for TCAN had undershot the correct value), TCAN is incremented by 1 K; otherwise it is decremented by 1 K. If this is not the first step in the iteration, then if RESID $>$0 and TSTEP $<$ 0 (indicating that TCAN has undershot the correct value and the last temperature increment had been a negative one) or if RESID $<$ 0 and TSTEP $>$ 0 (indicating that TCAN has overshot the correct value and the last temperature increment had been a positive one), TSTEP is divided in half and its sign changed. TSTEP is then added to TCAN. If TCAN is vanishingly close to 0 C, it is reset to that value. The iteration counter NITER and the flag NUMIT are each incremented by one. Finally, if NUMIT $>$ 0, the iteration cycle is repeated from line 400 on.

If ITC = 2, the calculations for the Newton-Raphson method of iteration in loop 575 are performed, over each of the modelled areas for which ITER = 1. As outlined above, in this approach the value x_{n+1} used at each iteration step is obtained from the value x_n at the previous step as follows:

$x_{n+1} = x_n - f(x_n)/f'(x_n)$

Identifying x_n with TCAN and $f(x_n)$ with the surface energy balance equation, it can be seen that the second term on the right-hand side corresponds to TSTEP; the numerator is equal to RESID and the denominator to the first derivative of the energy balance equation evaluated at TCAN, which in turn is equal to the sum of the derivatives of the individual terms:

$d(L_{*c})/dT = -8\sigma T_c^3(1 - \chi)$

$d(Q_{H,c})/dT = \rho_a c_p 1/r_A + [T_c - T_{a,pot}]d(1/r_A)/dT$

$d(Q_{E,c})/dT = L_v \rho_a X_E dq_c/dT + [q_c - q_a]dX_E/dT$

$d\Delta Q_{S,c}/dT = C_c/\Delta t$

The term $d(1/r_A)/dT$ is represented by the variable DCFLXM, which is approximated as the difference between CFLX and its value for the previous iteration, CFLUXM, divided by TSTEP. The term dX_E/dT is represented by the variable DXEVAP, which is approximated as the difference between XEVAP and its value for the previous iteration, XEVAPM, divided by TSTEP. The calculated value of TSTEP obtained from the above calculations is constrained to be between -10 and 5 K to dampen any spurious oscillations, and is then added to TCAN. If the resulting value of TCAN is vanishingly close to 0 C, it is reset to that value. At the end of the calculations the iteration counter NITER and the flag NUMIT are each incremented by one. The values of $T_{a,c}$, $q_{a,c}$ and $T_{ac,v}$ are reset to T_c , q_c and $T_{c,v}$ respectively. Upon exiting the loop, if NUMIT $>$ 0, the iteration cycle is repeated from line 400 on.

After the iteration has been completed, if the Newton-Raphson method has been used, a check is carried out in loop 600 to ascertain whether convergence has not been reached (i.e. whether ITER = -1) for any location. In such cases it is assumed that conditions of near-neutral stability at the surface are the cause of the difficulty in finding a solution. The flags NUMIT and IEVAPC are set to zero, and a trial value of TCAN is calculated using the virtual potential temperature of the air and the canopy specific hu-

midity. If the absolute value of RESID is $> 100Wm^{-2}$, TCAN is set to this trial value. The values of $q_{0,c} and the components of the surface energy balance are recalculated as above, except that$ {H,c} and Q_{E,c} $are assumed as a first approximation to be zero. RESID is determined on this basis, and is then assigned to$ {H,c} and {E,c} $. If RESID > 0,$ Q_{E,c} $is set to this value; otherwise it is equally divided between$ Q_{H,c} and Q_{E,c} $. The residual is then reset to zero, and E(0) and$ {v,c}

After loop 600, calls to DRCOEF or FLXSURFZ are performed to re-evaluate the surface turbulent transfer coefficients for any locations for which the fluxes were modified in the previous loop, i.e. for any locations for which IEVAPC was set to 1. After this a check is performed for unphysical values of the canopy temperature, i.e. for values greater than 100 C or less than -100 C. If such values are encountered, an error message is printed and a call to abort is carried out.

Next a check is carried out to determine whether freezing or melting of intercepted water has occurred over the current time step. If this is the case, adjustments are required to the canopy temperature, the intercepted liquid and frozen water amounts and fractional coverages, and to C_c and $\Delta Q_{S,c}$. In loop 650, if there is liquid water stored on the canopy and the canopy temperature is less than 0 C, the first half of the adjustment to $\Delta Q_{S,c}$ is performed, and the flags ITER and NIT are set to 1. The available energy sink HFREZ is calculated from CHCAP and the difference between TCAN and 0 C, and compared with HCONV, calculated as the energy sink required to freeze all of the liquid water on the canopy. If HFREZ $\leq$ HCONV, the amount of water that can be frozen is calculated using the latent heat of melting. The fractional coverages of frozen and liquid water FSNOWC and FRAINC and their masses SNOCAN and RAICAN are adjusted accordingly, TCAN is set to 0 C, and the amount of energy involved is stored in the diagnostic variable QMELTC. Otherwise all of the intercepted liquid water is converted to frozen water, and the energy available for cooling the canopy is calculated as HCOOL = HFREZ - HCONV. This available energy is applied to decreasing the temperature of the canopy, using the specific heat of the canopy elements, and the amount of energy that was involved in the phase change is stored in the diagnostic variable QMELTC. In both cases q_c and $T_{c,v}$ are recalculated, and at the end of the loop C_c and $\Delta Q_{S,c}$ are re-evaluated.

In loop 675, if there is frozen water stored on the canopy and the canopy temperature is greater than 0 C, the first half of the adjustment to $\Delta Q_{S,c}$ is performed, and the flags ITER and NIT are set to 1. The available energy for melting, HMELT, is calculated from CHCAP and the difference between TCAN and 0 C, and compared with HCONV, calculated as the energy required to melt all of the frozen water on the canopy. If HMELT $\leq$ HCONV, the amount of frozen water that can be melted is calculated using the latent heat of melting. The fractional coverages of frozen and liquid water FSNOWC and FRAINC and their masses SNOCAN and RAICAN are adjusted accordingly, TCAN is set to 0 C, and the amount of energy involved is stored in the diagnostic variable QMELTC. Otherwise, all of the intercepted frozen water is converted to liquid water, and the energy available for warming the canopy is calculated as HWARM = HMELT - HCONV. This available energy is applied to increasing the temperature of the canopy, using the specific heats of the canopy elements, and the amount of energy that was involved in the phase change is stored in the diagnostic variable QMELTC. In both cases q_c and $T_{c,v}$ are recalculated, and at the end of the loop C_c and $\Delta Q_{S,c}$ are re-evaluated.

For the locations over which melting or freezing of water on the canopy has occurred, the surface fluxes must now be recalculated to reflect the modified canopy temperature and humidity. If NIT $>$ 0, first DRCOEF or FLXSURFZ is called to re-evaluate the surface turbulent transfer coefficients over all locations where ITER has been set to 1. Loops 700 and 750 repeat the calculations done in loops 475 and 500, to obtain the surface transfer coefficients. Loop 800 repeats the calculations of the surface fluxes done in loop 525.

At the end of the subroutine, some final diagnostic and clean-up calculations are performed. If the evaporation rate from the canopy, EVAPC, is very small, it is added to the residual of the canopy energy balance equation, RESID, and then reset to zero. The overall residual is added to the sensible heat flux from the canopy. The energy balance of the surface underlying the canopy is then re-evaluated to take into account the new value of the canopy longwave radiation. If the surface temperature is close to 0 C, the residual of the equation is assigned to QME↩ LTG, representing the energy associated with melting or freezing of water at the surface; otherwise it is assigned to the ground heat flux. If the water vapour flux is towards the canopy, the water sublimated or condensed is assigned to interception storage: to SNOCAN if SNOCAN $>$ 0 (for consistency with the definition of CPHCHC), and to RAICAN otherwise. The diagnostic water vapour flux variables QFCF and QFCL, and the diagnostic change of canopy heat storage HTCC, are updated accordingly, and the canopy heat capacity CHCAP is recalculated; EVAPC is added to the diagnostic variable EVAP and then reset to zero. Finally, the net shortwave radiation, outgoing longwave radiation, sensible and latent heat fluxes are calculated for the whole canopy-ground surface ensemble; the evaporation rates are converted to ms^{-1} ; and the iteration counter ITERCT is updated for the level corresponding to the subarea type and the value of NITER.

9.72 TSOLVE.f File Reference

Purpose: Solution of surface energy balance for non-vegetated subareas.

Functions/Subroutines

- subroutine tsolve (ISNOW, FI, QSWNET, QLWOUT, QTRANS, QSENS, QEVAP, EVAP, TZERO, QZERO, GZERO, QMELT, CDH, CDM, RIB, CFLUX, FTEMP, FVAP, ILMO, UE, H, QLWIN, TPOTA, QA, VA, PADRY, RHOAIR, ALVISG, ALNIRG, CRIB, CPHCH, CEVAP, TVIRTA, ZOSCLH, ZOSCLM, ZRSLFH, ZRSLFM, ZOH, ZOM, FCOR, GCONST, GCOEFF, TSTART, PCPR, TRSNOWG, FSSB, ALSNO, THLIQ, THLMIN, DELZW, RHOSNO, ZSNOW, IWATER, IEVAP, ITERCT, ISAND, ISLFD, ITG, ILG, IG, IL1, IL2, JL, NBS, ISNOALB, TSTEP, TVIRTS, EVBETA, Q0SAT, RESID, DCFLXM, CFLUXM, WZERO, TRTOP, A, B, LZZ0, LZZ0T, FM, FH, ITER, NITER, JEVAP, KF)

9.72.1 Detailed Description

Purpose: Solution of surface energy balance for non-vegetated subareas.

9.72.2 Function/Subroutine Documentation

9.72.2.1 **subroutine tsolve (integer *ISNOW,* real, dimension (ilg) *FI,* real, dimension(ilg) *QSWNET,* real, dimension(ilg) *QLWOUT,* real, dimension(ilg) *QTRANS,* real, dimension (ilg) *QSENS,* real, dimension (ilg) *QEVAP,* real, dimension (ilg) *EVAP,* real, dimension (ilg) *TZERO,* real, dimension (ilg) *QZERO,* real, dimension (ilg) *GZERO,* real, dimension (ilg) *QMELT,* real, dimension (ilg) *CDH,* real, dimension (ilg) *CDM,* real, dimension (ilg) *RIB,* real, dimension (ilg) *CFLUX,* real, dimension (ilg) *FTEMP,* real, dimension (ilg) *FVAP,* real, dimension (ilg) *ILMO,* real, dimension (ilg) *UE,* real, dimension (ilg) *H,* real, dimension (ilg) *QLWIN,* real, dimension (ilg) *TPOTA,* real, dimension (ilg) *QA,* real, dimension (ilg) *VA,* real, dimension (ilg) *PADRY,* real, dimension(ilg) *RHOAIR,* real, dimension(ilg) *ALVISG,* real, dimension(ilg) *ALNIRG,* real, dimension (ilg) *CRIB,* real, dimension (ilg) *CPHCH,* real, dimension (ilg) *CEVAP,* real, dimension(ilg) *TVIRTA,* real, dimension(ilg) *ZOSCLH,* real, dimension(ilg) *ZOSCLM,* real, dimension(ilg) *ZRSLFH,* real, dimension(ilg) *ZRSLFM,* real, dimension (ilg) *ZOH,* real, dimension (ilg) *ZOM,* real, dimension (ilg) *FCOR,* real, dimension(ilg) *GCONST,* real, dimension(ilg) *GCOEFF,* real, dimension(ilg) *TSTART,* real, dimension (ilg) *PCPR,* real, dimension(ilg,nbs) *TRSNOWG,* real, dimension(ilg,nbs) *FSSB,* real, dimension(ilg,nbs) *ALSNO,* real, dimension(ilg,ig) *THLIQ,* real, dimension(ilg,ig) *THLMIN,* real, dimension(ilg,ig) *DELZW,* real, dimension(ilg) *RHOSNO,* real, dimension(ilg) *ZSNOW,* integer, dimension(ilg) *IWATER,* integer, dimension (ilg) *IEVAP,* integer, dimension(ilg,6,50) *ITERCT,* integer, dimension(ilg,ig) *ISAND,* integer *ISLFD,* integer *ITG,* integer *ILG,* integer *IG,* integer *IL1,* integer *IL2,* integer *JL,* integer *NBS,* integer *ISNOALB,* real, dimension (ilg) *TSTEP,* real, dimension(ilg) *TVIRTS,* real, dimension(ilg) *EVBETA,* real, dimension (ilg) *Q0SAT,* real, dimension (ilg) *RESID,* real, dimension(ilg) *DCFLXM,* real, dimension(ilg) *CFLUXM,* real, dimension (ilg) *WZERO,* real, dimension (ilg,nbs) *TRTOP,* real, dimension (ilg) *A,* real, dimension (ilg) *B,* real, dimension (ilg) *LZZ0,* real, dimension (ilg) *LZZ0T,* real, dimension (ilg) *FM,* real, dimension (ilg) *FH,* integer, dimension (ilg) *ITER,* integer, dimension (ilg) *NITER,* integer, dimension (ilg) *JEVAP,* integer, dimension (ilg) *KF*)

Parameters

isnow	Flag indicating presence or absence of snow
qswnet	Net shortwave radiation at surface $[Wm^{-2}]$
qlwout	Upwelling longwave radiation at surface $[Wm^{-2}]$ (L)
qtrans	Shortwave radiation transmitted into surface $[Wm^{-2}]$
qsens	Sensible heat flux from surface $[Wm^{-2}](Q_H)$
qevap	Latent heat flux from surface $[Wm^{-2}](Q_E)$

evap	Evaporation rate at surface $[kgm^{-2}s^{-1}](E(0))$
tzero	Temperature at surface $[K](T(0))$
qzero	Specific humidity at surface $[kgkg^{-1}](q(0))$
gzero	Heat flux into surface $[Wm^{-2}](G(0))$
qmelt	Heat available for melting snow or freezing water at the surface $[Wm^{-2}]$
cdh	Surface drag coefficient for heat $[](C_{DH})$
cdm	Surface drag coefficient for momentum [ ]
rib	Bulk Richardson number at surface [ ]
cflux	Product of surface drag coefficient and wind speed $[ms^{-1}]$
ftemp	Product of surface-air temperature gradient, drag coefficient and wind speed $[Kms^{-1}]$
fvap	Product of surface-air humidity gradient, drag coefficient and wind speed $[kgkg^{-1}ms^{-1}]$
ilmo	Inverse of Monin-Obukhov roughness length $(m-1]$
ue	Friction velocity of air $[ms^{-1}]$
h	Height of the atmospheric boundary layer [m]
fi	Fractional coverage of subarea in question on modelled area [ ]
qlwin	Downwelling longwave radiation at bottom of atmosphere $[Wm^{-2}]$
tpota	Potential temperature of air at reference height $[K](T_{a,pot})$
qa	Specific humidity at reference height $[kgkg^{-1}](q_a)$
va	Wind speed at reference height $[ms^{-1}](v_a)$
padry	Partial pressure of dry air $[Pa](p_{dry})$
rhoair	Density of air $[kgm^{-3}](\rho_a)$
alvisg	Visible albedo of ground surface [ ]
alnirg	Near-IR albedo of ground surface [ ]
crib	Richardson number coefficient $[K^{-1}]$
cphch	Latent heat of vaporization at surface $[Jkg^{-1}]$
cevap	Soil evaporation efficiency coefficient $[](\beta)$
tvirta	Virtual potential temperature of air at reference height [K]
zosclh	Ratio of roughness length for heat to reference height for temperature and humidity [ ]
zosclm	Ratio of roughness length for momentum to reference height for wind speed [ ]
zrslfh	Difference between reference height for temperature and humidity and height at which extrapolated wind speed goes to zero [m]
zrslfm	Difference between reference height for wind speed and height at which extrapolated wind speed goes to zero [m]
zoh	Surface roughness length for heat [m]
zom	Surface roughness length for momentum [m]
gconst	Intercept used in equation relating surface heat flux to surface temperature $[Wm^{-2}]$
gcoeff	Multiplier used in equation relating surface heat flux to surface temperature $[Wm^{-2}K^{-1}]$
tstart	Starting point for surface temperature iteration [K]
fcor	Coriolis parameter $[s^{-1}]$
pcpr	Surface precipitation rate $[kgm^{-2}s^{-1}]$
iwater	Flag indicating condition of surface (dry, water-covered or snow-covered)
ievap	Flag indicating whether surface evaporation is occurring or not
iterct	Counter of number of iterations required to solve energy balance for four subareas
isand	Sand content flag

For the surface temperature iteration, two alternative schemes are offered: the bisection method (selected if the flag ITG = 1) and the Newton-Raphson method (selected if ITG = 2). In the first case, the maximum number of iterations ITERMX is set to 12, and in the second case it is set to 5. An optional windless transfer coefficient EZERO is available, which can be used, following the recommendations of Brown et al. (2006), to prevent the sensible heat flux over snow packs from becoming vanishingly small under highly stable conditions. If the snow cover flag ISNOW is zero (indicating bare ground), EZERO is set to zero; if ISNOW=1, EZERO is set to $2.0Wm^{-2}K^{-1}$.

FLAG: this comment below likely needs updating!

In the beginning loop, some preliminary calculations are done. The shortwave transmissivity at the surface, T↩ RTOP, is set to zero in the absence of a snow pack, and to the transmissivity of snow TRSNOW otherwise. The net shortwave radiation at the surface, QSWNET, is calculated as the sum of the incoming visible and near-infrared shortwave radiation, weighted according to one minus their respective albedos. This average value is corrected for the amount of radiation transmitted into the surface, obtained using TRTOP. The initial value of the surface

temperature TZERO is set to TSTART, which contains the value of TZERO from the previous time step, and the first step in the iteration sequence, TSTEP, is set to 1.0 K. The flag ITER is set to 1 for each element of the set of modelled areas, indicating that its surface temperature has not yet been found. The iteration counter NITER is initialized to 1 for each element. Initial values are assigned to several other variables.

The 100 continuation line marks the beginning of the surface temperature iteration sequence. First the flags NIT (indicating that there are still locations at the beginning of the current iteration step for which the surface temperature has not yet been found) and NUMIT (indicating that there are still locations at the end of the current iteration step for which the surface temperature has not yet been found) are set to zero. Loop 150 is then performed over the set of modelled areas. If ITER=1, NIT is incremented by one, and the initial value of the surface transfer coefficient CFLUXM for this iteration pass is set to its value from the previous pass. The virtual temperature at the surface, $T(0)_v$, is obtained using the standard expression (see documentation for subroutine CLASST):

$$T(0)_v = T(0)[1 + 0.61q(0)]$$

where T(0) is the surface temperature and q(0) is the specific humidity at the surface. The surface humidity can be obtained from the saturated specific humidity $q(0)_{sat}$ by making use of the definition of the surface evaporation efficiency β:

$$\beta = [q(0)\breve{}q_a]/[q(0)_{sat} - q_a]$$

where q_a is the specific humidity of the air at the reference height. This expression is inverted to obtain an expression for q(0). The saturated specific humidity $q(0)_{sat}$ is determined from the mixing ratio at saturation, $w(0)_{sat}$:

$$q(0)_{sat} = w(0)_{sat}/[1 + w(0)_{sat}]$$

The saturation mixing ratio is a function of the saturation vapour pressure $e(0)_{sat}$ at the surface:

$$w(0)_{sat} = 0.622e(0)_{sat}/(p_{dry})$$

where p_{dry} is the partial pressure of dry air. A standard empirical equation for the saturation vapour pressure dependence on the temperature T is used:

$$e_{sat} = 611.0exp[17.269(T\breve{}T_f)/(T\breve{}35.86)]T \geq T_f \; e_{sat} = 611.0exp[21.874(T\breve{}T_f)/(T\breve{}7.66)]T < T_f$$

where T_f is the freezing point. If there is a snow cover or ponded water present on the surface (IWATER > 0), the surface evaporation efficiency EVBETA is set to 1 and q(0) is set to $q(0)_{sat}$. Otherwise EVBETA is set to CEVAP, the value obtained in subroutine TPREP on the basis of ambient conditions, and q(0) is calculated as above. If $q(0) > q_a$ and the evaporation flag IEVAP has been set to zero, EVBETA is reset to zero and q(0) is reset to q_a. Finally, $T(0)_v$ is determined using the equation above.

If NIT > 0, a subroutine is called to evaluate the stability- corrected surface drag coefficients for heat and momentum. The subroutine selection is made on the basis of the flag ISLFD. If ISLFD=1, indicating that the calculations are to be consistent with CCCma conventions, subroutine DRCOEF is called; if ISLFD=2, indicating that the calculations are to be consistent with RPN conventions, subroutine FLXSURFZ is called.

In loop 175, the terms of the surface energy balance are evaluated. The energy balance equation is written as:

$$K_* + L_* - Q_H\breve{}Q_E\breve{}G(0) = 0$$

where K_* is the net shortwave radiation, L_* is the net longwave radiation, Q_H is the sensible heat flux, Q_E is the latent heat flux, and G(0) is the conduction into the surface. K_* was evaluated earlier in loop 50. L_* is obtained as the difference between the downwelling radiation $L\downarrow$ and the upwelling radiation $L\uparrow$, which in turn is determined using the Stefan- Boltzmann equation:

$$L\uparrow = \sigma T(0)^4$$

where σ is the Stefan-Boltzmann constant. (It is assumed that natural surfaces, because of their radiative complexity, act as effective black bodies, so that their emissivity can be taken to be 1.) The sensible heat flux is given by

$$Q_H = [\rho_a c_p C_{DH} v_a + \epsilon_0][T(0)\breve{}T_{a,pot}]$$

where ρ_a is the density of the air, c_p is its specific heat, C_{DH} is the surface drag coefficient for heat, and v_a and $T_{a,pot}$ are the wind speed and potential temperature respectively at the reference height. (Note in the code that the variable CFLUX, evaluated in subroutine DRCOEF or FLXSURFZ, represents the product of C_{DH} and v_a.) The windless transfer coefficient ϵ_0, evaluated at the beginning of the subroutine, is used only under stable conditions, i.e. if $T(0) < T_{a,pot}$. The evaporation rate at the surface, E(0), is calculated as

$$E(0) = \rho_a C_{DH} v_a [Q(0)\breve{}q_a]$$

Q_E is obtained by multiplying E(0) by the latent heat of vaporization at the surface. The ground heat flux G(0) is determined as a linear function of T(0) (see documentation for subroutines TNPREP and TSPREP). It can be seen that each of the terms of the surface energy balance is a function of a single unknown, T(0) or TZERO. The residual RESID of the energy balance is now evaluated on the basis of the current estimation for TZERO. If the absolute value of RESID is less than $5.0Wm^{-2}$, or if the absolute value of the iteration step TSTEP most recently used is less than 0.01 K, the surface temperature is deemed to have been found and ITER is set to 0. If the iteration counter NITER is equal to the maximum number and ITER is still 1, ITER is set to -1.

In the following section, the iteration sequence is moved ahead a step. If ITG = 1, then if NIT $>$ 0 and if ITER for the array element in question is 1, the calculations for the bisection method of solution are performed. If NITER = 1 (indicating that this is the first step in the iteration), then if RESID $>$ 0 (indicating that the current value for TZERO had undershot the correct value), TZERO is incremented by 1 K; otherwise it is decremented by 1 K. If this is not the first step in the iteration, then if RESID $>$0 and TSTEP $<$ 0 (indicating that TZERO has undershot the correct value and the last temperature increment had been a negative one) or if RESID $<$ 0 and TSTEP $>$ 0 (indicating that TZERO has overshot the correct value and the last temperature increment had been a positive one), TSTEP is divided in half and its sign changed. TSTEP is then added to TZERO. The iteration counter NITER and the flag NUMIT are each incremented by one. The next loop contains an optional set of print statements that are executed if ITER=-1, that is if a solution for the surface temperature has not been found within the prescribed maximum number of iteration steps. Finally, if NUMIT $>$ 0, the iteration cycle is repeated from line 100 on.

If ITG = 2, then if NIT $>$ 0 and if ITER for the array element in question is 1, the calculations for the Newton-Raphson method of iteration are performed. In this approach, the value x_{n+1} used at each iteration step is obtained from the value x_n at the previous step as follows:

$x_{n+1} = x_n \breve{} f(x_n)/f'(x_n)$

Identifying x_n with TZERO and $f(x_n)$ with the surface energy balance equation, it can be seen that the second term on the right-hand side corresponds to TSTEP; the numerator is equal to RESID and the denominator to the first derivative of the energy balance equation evaluated at TZERO, which in turn is equal to the sum of the derivatives of the individual terms:

$d(L\uparrow)/dT = -4\sigma T(0)^3$

$d(Q_H)/dT = \rho_a c_p C_{DH} v_a + [T(0) \breve{} T_{a,pot}] d(C_{DH} v_a)/dT$

$d(Q_E)/dT = L_v \rho_a C_{DH} v_a dq(0)/dT + [q(0) \breve{} q_a] d(C_{DH} v_a)/dT$

and dG(0)/dT is equal to the coefficient multiplying TZERO in the equation for G(0). (L_v is the latent heat of vaporization at the surface.) At the end of the calculations the iteration counter NITER and the flag NUMIT are each incremented by one, and upon exiting the loop, if NUMIT $>$ 0, the iteration cycle is repeated from line 100 on.

After the iteration has been completed, NUMIT is reset to zero and a check is carried out to ascertain whether convergence has not been reached (i.e. whether ITER = -1) for any location. In such cases it is assumed that conditions of near-neutral stability at the surface are the cause of the difficulty in finding a solution. A trial value of TZERO is calculated using the virtual potential temperature of the air. If RESID $> 50Wm^{-2}$, TZERO is set to this trial value. The values of q(0) and the components of the surface energy balance are recalculated as above, except that Q_H and Q_E are assumed as a first approximation to be zero. RESID is determined on this basis. If RESID is positive, it is assigned to Q_E; otherwise RESID is divided equally between Q_H and Q_E, except in the case of an absolutely dry surface, in which case Q_E is set to zero and Q_H to RESID. RESID is reset to zero, E(0) is obtained from Q_E, and $T(0)_v$ is recalculated. The flag JEVAP for the location is set to 1, and NUMIT is incremented by 1. If NUMIT $>$ 0 at the end of this loop, DRCOEF or FLXSURF are called again, and their calculations are performed for any location where JEVAP is 1, to ensure consistency with the new surface temperature and humidity.

At this point a check is performed for unphysical values of the surface temperature, i.e. for values greater than 100 C or less than -100 C. If such values are encountered, an error message is printed and a call to abort is carried out.

Finally, a check is performed to ensure that TZERO is not less than 0 C if ponded water is present on the surface (IWATER = 1) or greater than 0 C if snow is present on the surface (IWATER = 2), or greater than zero if the surface is an ice sheet (ISAND = -4). If any of these cases is true, TZERO is reset to the freezing point, and q(0) and $T(0)_v$ are recalculated. DRCOEF or FLXSURFZ are called, and their calculations are performed for all locations meeting these criteria. The components of the surface energy balance are recalculated; the residual amount is assigned to the energy associated with phase change of water at the surface, QMELT, and RESID is set to zero. If QMELT $<$ 0, i.e. if there is freezing taking place, the evaporation flux QEVAP is added to it and then reset to zero (since if ponded water is freezing, it is unavailable for evaporation).

In the last half of the loop, some final adjustments are made to a few variables. If the evaporation flux is vanishingly small, it is added to RESID and reset to zero. If an anomalous case has arisen in which QMELT $<$ 0 over a snow-covered surface or QMELT $>$ 0 over a snow-free surface, QMELT is added to the heat flux into the surface and then reset to zero. Any remaining residual flux is added to Q_H. The shortwave radiation transmitted into the surface is added back to the net shortwave radiation for diagnostic purposes. The surface vapour flux is converted into units of ms^{-1}. Lastly, the iteration counter ITERCT is updated for the level corresponding to the subarea type and the value of NITER.

9.73 TSPOST.f File Reference

Purpose: Snow temperature calculations and cleanup after surface energy budget calculations.

Functions/Subroutines

- subroutine tspost (GSNOW, TSNOW, WSNOW, RHOSNO, QMELTG, GZERO, TSNBOT, HTCS, HMFN, G↩ CONSTS, GCOEFFS, GCONST, GCOEFF, TBAR, TSURF, ZSNOW, TCSNOW, HCPSNO, QTRANS, FI, D↩ ELZ, ILG, IL1, IL2, JL, IG)

9.73.1 Detailed Description

Purpose: Snow temperature calculations and cleanup after surface energy budget calculations.

9.73.2 Function/Subroutine Documentation

9.73.2.1 subroutine tspost (real, dimension (ilg) *GSNOW,* real, dimension (ilg) *TSNOW,* real, dimension (ilg) *WSNOW,* real, dimension(ilg) *RHOSNO,* real, dimension(ilg) *QMELTG,* real, dimension (ilg) *GZERO,* real, dimension(ilg) *TSNBOT,* real, dimension (ilg) *HTCS,* real, dimension (ilg) *HMFN,* real, dimension(ilg) *GCONSTS,* real, dimension(ilg) *GCOEFFS,* real, dimension(ilg) *GCONST,* real, dimension(ilg) *GCOEFF,* real, dimension(ilg,ig) *TBAR,* real, dimension (ilg) *TSURF,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *TCSNOW,* real, dimension(ilg) *HCPSNO,* real, dimension(ilg) *QTRANS,* real, dimension (ilg) *FI,* real, dimension (ig) *DELZ,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IG*)

Parameters

gzero	Heat conduction into soil surface $[Wm^{-2}](G(\Delta z_s))$
tsnbot	Temperature at bottom of snow pack [K]
gsnow	Heat conduction into surface of snow pack $[Wm^{-2}](G(0))$
tsnow	Snowpack temperature $[K/C](T_s)$
wsnow	Liquid water content of snow pack $[kgm^{-2}](w_s)$
rhosno	Density of snow $[kgm^{-3}](\rho_s)$
qmeltg	Available energy to be applied to melting of snow $[Wm^{-2}]$
htcs	Internal energy change of snow pack due to conduction and/or change in mass $[Wm^{-2}](I_s)$
hmfn	Energy associated with phase change of water in snow pack $[Wm^{-2}]$
tsurf	Snow surface temperature [K]
zsnow	Depth of snow pack $[m](\Delta z_s)$
tcsnow	Thermal conductivity of snow $[Wm^{-1}K^{-1}]$
hcpsno	Heat capacity of snow $[Jm^{-3}K^{1}](C_s)$

qtrans	Shortwave radiation transmitted through the snow pack $[Wm^{-2}]$
gconst	Intercept used in equation relating snow surface heat flux to snow surface temperature $[Wm^{-2}]$
gcoeff	Multiplier used in equation relating snow surface heat flux to snow surface temperature $[Wm^{-2}K^{-1}]$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
gconsts	Intercept used in equation relating snow surface heat flux to snow surface temperature $[Wm^{-2}]$
gcoeffs	Multiplier used in equation relating snow surface heat flux to snow surface temperature $[Wm^{-2}K^{-1}]$
tbar	Temperatures of soil layers, averaged over modelled area [K]
delz	Overall thickness of soil layer [m]

In the 100 loop, the heat flux into the snow surface (without adjustments that may have been applied relating to partitioning of the residual of the surface energy balance among the surface flux terms) is calculated from the snow surface temperature TSURF, using the GCOEFFS and GCONSTS terms (see documentation of subroutine TSPREP). The temperature at the bottom of the snow pack, TSNBOT, is then calculated. Currently TSNBOT is determined as a simple average of the temperatures of the snow and the first soil layer, weighted according to their respective depths (and constrained to be $\leq$ 0 C), but this is under review. The heat flux into the soil surface is then evaluated from TSNBOT and the GCOEFF and GCONST terms (see documentation of subroutine TNPREP). If the energy to be applied to the melting of snow, QMELTG, is negative (indicating an energy sink), QMELTG is added to the heat flux into the ground, GZERO, and reset to zero. The temperature of the snow pack is then stepped forward using the heat fluxes at the top and bottom of the snow pack, G(0) and $G(\Delta z_s)$:

$$\Delta T_s = [G(0) - G(\Delta z_s)]\Delta t/(C_s \Delta z_s)$$

where C_s is the snow heat capacity, Δt the time step and Δz_s the snow depth. If the new snow temperature is greater than zero, the excess amount of heat is calculated and added to QMELTG and subtracted from GSNOW, and TSNOW is reset to 0 C. Finally, the shortwave radiation transmitted through the snow pack, QTRANS, is added to GZERO.

In the 200 loop, since liquid water is assumed only to exist in the snow pack if it is at 0 C, a check is carried out to determine whether the liquid water content WSNOW $>$ 0 at the same time as the snow temperature TSNOW $<$ 0. If so, the change of internal energy I_s of the snow pack as a result of this phase change is calculated as the difference in I_s between the beginning and end of the loop:

$$\Delta I_s = X_i \Delta [C_s T_s]/\Delta t$$

where X_i represents the fractional coverage of the subarea under consideration relative to the modelled area. The total energy sink HADD available to freeze liquid water in the snow pack is calculated from TSNOW, and the amount of energy HCONV required to freeze all the available water is calculated from WSNOW. If HADD $<$ HCONV, only part of WSNOW is frozen; this amount WFREZ is calculated from HADD and subtracted from WSNOW, the snow temperature is reset to 0 C, the frozen water is used to update the snow density, and the snow heat capacity is recalculated:

$$C_s = C_i[\rho_s/\rho_i] + C_w w_s/[\rho_w \Delta z_s]$$

where C_i and C_w are the heat capacities of ice and water respectively, w_s is the snow water content and ρ_s, ρ_i and ρ_w are the densities of snow, ice and water respectively. If HADD $>$ HCONV, the available energy sink is sufficient to freeze all of WSNOW. HADD is recalculated as HADD – HCONV, WFREZ is set to WSNOW and added to the snow density, WSNOW is set to zero, the snow heat capacity is recalculated and HADD is used to determine a new value of TSNOW. Finally, WFREZ is used to update the diagnostic variables HMFN describing phase changes of water in the snow pack, and the change in internal energy HTCS.

9.74 TSPREP.f File Reference

Calculate coefficients for solution of snow pack heat conduction.

Functions/Subroutines

- subroutine tsprep (GCOEFFS, GCONSTS, CPHCHG, IWATER, FI, ZSNOW, TSNOW, TCSNOW, ILG, IL1, IL2, JL)

9.74.1 Detailed Description

Calculate coefficients for solution of snow pack heat conduction.

9.74.2 Function/Subroutine Documentation

9.74.2.1 subroutine tsprep (real, dimension(ilg) *GCOEFFS,* real, dimension(ilg) *GCONSTS,* real, dimension(ilg) *CPHCHG,* integer, dimension(ilg) *IWATER,* real, dimension (ilg) *FI,* real, dimension (ilg) *ZSNOW,* real, dimension (ilg) *TSNOW,* real, dimension(ilg) *TCSNOW,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

gcoeffs	Multiplier used in equation relating snow surface heat flux to snow surface temperature $[Wm^{-2}K^{-1}]$
gconsts	Intercept used in equation relating snow surface heat flux to snow surface temperature $[Wm^{-2}]$
cphchg	Latent heat of sublimation $[Jkg^{-1}]$
iwater	Flag indicating condition of surface (dry, water-covered or snow-covered)
fi	Fractional coverage of subarea in question on modelled area [ ]
zsnow	Depth of snow pack $[m](\Delta z_s)$
tsnow	Snowpack temperature $[K](T_s(\Delta z_s))$
tcsnow	Thermal conductivity of snow $[Wm^{-1}K^{-1}]$

In this subroutine, coefficients are derived for an equation relating the heat flux at the snow surface to the snow surface temperature. It is assumed that the variation of temperature T with depth z in the snow pack can be modelled by using a quadratic equation:

$$T(z) = (1/2)az^2 + bz + c$$

By substituting 0 for z in the above equation and in the expressions for its first and second derivatives, it can be shown that a = T"(0), b = T'(0), and c = T(0). The term T"(0) can be evaluated from the expression for the first derivative evaluated at the bottom of the snow pack, $T(\Delta z_s)$:

$$T"(0) = [T'(\Delta z_s) - T'(0)]/\Delta z_s$$

The temperature gradient T'(0) at the snow surface is related to the surface heat flux G(0) by the snow thermal conductivity λ_s:

$$G(0) = -\lambda_s T'(0)$$

The average snow temperature, $T_s(\Delta z_s)$, can be obtained by integrating the resulting equation for T(z) between 0 and Δz_s. Making use of all of the above expressions, and assuming as a first approximation that the heat flux at the bottom of the snow pack is zero, a linear equation can be derived relating G(0) to T(0):

$$G(0) = 3\lambda_s/\Delta z_s[T(0) - T_s(\Delta z_s)]$$

Just four calculations are performed in this subroutine. The slope and intercept of the G(0) vs. T(0) relation, G↩ COEFF and GCONST, are evaluated as $3\lambda_s/\Delta z_s$ and $-3\lambda_s T_s(\Delta z_s)/\Delta z_s$ respectively; the flag IWATER is set to 2, indicating a snow surface; and the latent heat of vaporization at the surface, CPHCHG, is set to the value for sublimation (by adding the latent heat of melting to the latent heat of vaporization).

- CALCULATE COEFFICIENTS.

9.75 turnover.f File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Stem And Root Turnover Subroutine.

Functions/Subroutines

- subroutine turnover (stemmass, rootmass,lfstatus,ailcg,il1,il2,leapnow,sort, nol2pfts,fcancmx,

9.75.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Stem And Root Turnover Subroutine.

The turnover of stem and root components is modelled via their PFT-dependent specified lifetimes. The litter generation ($kg\,C\,m^{-2}\,day^{-1}$) associated with turnover of stem ($D_{\rm S}$) and root ($D_{\rm R}$) components is calculated based on the amount of biomass in the respective components ($C_{\rm S}, C_{\rm R}$; $kg\,C\,m^{-2}$) and their respective turnover timescales ($\tau_{\rm S}$ and $\tau_{\rm R}$; yr; see also ctem_params.f90) as

$$D_i = C_i \left[1 - \exp\left(-\frac{1}{365\,\tau_i}\right)\right], \quad i = S, R.$$

9.75.2 Function/Subroutine Documentation

9.75.2.1 subroutine turnover (real, dimension(ilg,icc) *stemmass,* real, dimension(ilg,icc) *rootmass,* integer, dimension(ilg,icc) *lfstatus,* real, dimension(ilg,icc) *ailcg,* integer *il1,* integer *il2,* logical *leapnow,* integer, dimension(icc) *sort,* integer, dimension(ican) *nol2pfts,* real, dimension(ilg,icc) *fcancmx*)

Parameters

il1	il1=1
il2	il2=ilg
lfstatus	leaf status. an integer indicating if leaves are in "max. growth", "normal growth", "fall/harvest", or "no leaves" mode. see phenolgy subroutine for more details.
leapnow	true if this year is a leap year. Only used if the switch 'leap' is true.
sort	index for correspondence between 9 ctem pfts and size 12 of parameter vectors
nol2pfts	number of level 2 ctem pfts
stemmass	stem mass for each of the 9 ctem pfts, kgc/m^2
rootmass	root mass for each of the 9 ctem pfts, kgc/m^2
ailcg	green or live lai
fcancmx	max. fractional coverage of ctem's 9 pfts, but this can be modified by land-use change, and competition between pfts

Constants and parameters are located in ctem_params.f90

initialize required arrays to zero

initialization ends

calculate normal stem and root litter using the amount of stem and root biomass and their turnover time scales.

if crops are in harvest mode then we start harvesting stem as well. if stem has already been harvested then we set the stem harvest loss equal to zero. the roots of the crop die in a similar way.

stem/root harvest/death for crops

add stem and root litter from all sources

9.76 TWCALC.f File Reference

Check for freezing or thawing of liquid or frozen water in the soil layers, and adjust layer temperatures and water stores accordingly.

Functions/Subroutines

- subroutine twcalc (TBAR, THLIQ, THICE, HCP, TBARW, HMFG, HTC, FI, EVAP, THPOR, THLMIN, HCPS, DELZW, DELZZ, ISAND, IG, ILG, IL1, IL2, JL)

9.76.1 Detailed Description

Check for freezing or thawing of liquid or frozen water in the soil layers, and adjust layer temperatures and water stores accordingly.

9.76.2 Function/Subroutine Documentation

9.76.2.1 subroutine twcalc (real, dimension (ilg,ig) *TBAR,* real, dimension (ilg,ig) *THLIQ,* real, dimension (ilg,ig) *THICE,* real, dimension (ilg,ig) *HCP,* real, dimension (ilg,ig) *TBARW,* real, dimension (ilg,ig) *HMFG,* real, dimension (ilg,ig) *HTC,* real, dimension (ilg) *FI,* real, dimension (ilg) *EVAP,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *HCPS,* real, dimension (ilg,ig) *DELZW,* real, dimension (ilg,ig) *DELZZ,* integer, dimension (ilg,ig) *ISAND,* integer *IG,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL*)

Parameters

tbar	Temperature of soil layer $[C](T_g)$
thliq	Volumetric liquid water content of soil layer $[m^3m^{-3}](\theta_l)$
thice	Volumetric frozen water content of soil layer $[m^3m^{-3}](\theta_i)$
hcp	Heat capacity of soil layer $[Jm^{-3}K^{-1}](C_g)$
tbarw	Temperature of water in soil layer [C]
hmfg	Energy associated with freezing or thawing of water in soil layer $[Wm^{-2}]$
htc	Internal energy change of soil layer due to conduction and/or change in mass $[Wm^{-2}](I_g)$
fi	Fractional coverage of subarea in question on modelled area $[](X_i)$
evap	Calculated evaporation rate from soil surface $[ms^{-1}]$
thpor	Pore volume in soil layer $[mm](\theta_p)$
thlmin	Residual soil liquid water content remaining after freezing or evaporation $[m^3m^{-3}](\theta_r)$
hcps	Heat capacity of soil material $[Jm^{-3}K^{-1}](C_m)$
delzw	Permeable thickness of soil layer $[m](\Delta z_{g,w})$
delzz	Soil layer thicknesses to bottom of permeable depth for standard three-layer configuration, or to bottom of thermal depth for multiple layers $[m](\Delta z_{g,z})$
isand	Sand content flag

The adjustments of soil layer temperature and water content in this routine are done over the whole soil profile in the case of multiple soil layers (see the section on assignment of background data), but only to the bottom of the permeable depth in the case of the standard three-layer configuration (0.10, 0.25 and 3.75 m). This is because if the permeable depth lies within the thick third soil layer, it is recognized as desirable to apply the temperature changes only to that upper part of the layer in which the phase change is occurring, in order to avoid systematic damping of the temperature response of the layer. Thus the local array DELZZ (set in subroutine WPREP) is used here instead of DELZ when referring to the total thermal thickness of the soil layer, where DELZZ=DELZW for the third soil layer when the three-layer configuration is being used, but DELZZ=DELZ for all other cases.

The heat capacity C_g of the permeable part $\Delta z_{g,w}$ of the soil layer under consideration is calculated here and in various other places in the subroutine as the weighted average of the heat capacities of the liquid water content θ_l, the frozen water content θ_i, and the soil material (taken to apply to the volume fraction not occupied by the pore volume θ_p). The heat capacity of air in the pores is neglected:

$$C_g = C_w\theta_l + C_i\theta_i + C_m(1 - \theta_p)$$

Over the impermeable portion of the layer, the heat capacity of rock C_r is assumed to apply. Thus an effective heat capacity $C_{g,e}$ (in units of $Jm^{-2}K^{-1}$) over the soil layer in question, $\Delta z_{g,z}$, can be calculated as:

$C_{g,e} = C_g \Delta z_{g,w} + C_r(\Delta z_{g,z} - \Delta z_{g,w})$

The change of internal energy I in the soil layers as a result of freezing and thawing is calculated as the difference in I between the beginning and end of the subroutine:

$\Delta I_j = X_i \Delta(C_{g,e} T_g)/\Delta t$

where T_g is the temperature of the layer, Δt the length of the time step, and X_i the fractional coverage of the subarea under consideration relative to the modelled area.

If the soil layer temperature is less than 0 C and the volumetric liquid water content θ_l of the layer is greater than the residual water content θ_f, the water content THFREZ that can be frozen by the available energy sink is calculated from C_e and T_g. The volumetric water content THEVAP of the first layer that is required to satisfy the surface evaporative flux is determined. For each layer, if THLIQ is found to exceed THLMIN + THEVAP, THFREZ is compared to the available water. If THFREZ $\leq$ THLIQ – THLMIN – THEVAP, all of the available energy sink is used to freeze part of the liquid water content in the permeable part of the soil layer, the amount of energy involved is subtracted from HTC and added to HMFG, C_g is recalculated and the layer temperature is set to 0 C. Otherwise, all of the liquid water content of the layer above THLMIN + THEVAP is converted to frozen water, and HMFG and HTC are recalculated to reflect this. Then C_g is recomputed, and the remaining energy sink is applied to decreasing the temperature of the soil layer (both the permeable and impermeable portions) using C_e.

If the soil layer temperature is greater than 0 C and the volumetric ice content θ_i of the layer is greater than zero, the ice content THMELT that can be melted by the available energy is calculated from C_e and T_g. For each layer, if THMELT $\leq$ THICE, all of the available energy is used to melt part of the frozen water content of the permeable part of the layer, the amount of energy involved is subtracted from HTC and added to HMFG, C_g is recalculated and the layer temperature is set to 0 C. Otherwise, all of the frozen water content of the layer is converted to liquid water, and HMFG and HTC are recalculated to reflect this. Then C_g is recomputed, and the remaining energy is applied to increasing the temperature of the soil layer (both the permeable and impermeable portions) using C_e.

In the final cleanup, the internal energy calculations for this subroutine are completed, and the first half of a new set of internal energy calculations is done to span the subroutines treating ground water movement, which will be completed in subroutine TMCALC. Lastly, TBARW, the liquid water temperature of each soil layer, is assigned using TBAR.

9.77 WEND.f File Reference

Purpose: Recalculate liquid water content of soil layers after infiltration, and evaluate baseflow.

Functions/Subroutines

- subroutine wend (THLIQX, THICEX, TBARWX, ZPOND, TPOND, BASFLW, TBASFL, RUNOFF, TRUNOF, FI, WMOVE, TMOVE, LZF, NINF, TRMDR, THLINF, DELZX, ZMAT, ZRMDR, FDTBND, WADD, TADD, FDT, TFDT, THLMAX, THTEST, THLDUM, THIDUM, TDUMW, TUSED, RDUMMY, ZERO, WEXCES, XDRAIN, T↩ HPOR, THLRET, THLMIN, BI, PSISAT, GRKSAT, THFC, DELZW, ISAND, IGRN, IGRD, IGDR, IZERO, IVEG, IG, IGP1, IGP2, ILG, IL1, IL2, JL, N)

9.77.1 Detailed Description

Purpose: Recalculate liquid water content of soil layers after infiltration, and evaluate baseflow.

9.77.2 Function/Subroutine Documentation

9.77.2.1 subroutine wend (real, dimension(ilg,igp1) *THLIQX,* real, dimension(ilg,igp1) *THICEX,* real, dimension(ilg,igp1) *TBARWX,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *TPOND,* real, dimension(ilg) *BASFLW,* real, dimension(ilg) *TBASFL,* real, dimension(ilg) *RUNOFF,* real, dimension(ilg) *TRUNOF,* real, dimension (ilg) *FI,* real, dimension (ilg,igp2) *WMOVE,* real, dimension (ilg,igp2) *TMOVE,* integer, dimension (ilg) *LZF,* integer, dimension (ilg) *NINF,* real, dimension (ilg) *TRMDR,* real, dimension(ilg,igp1) *THLINF,* real, dimension (ilg,igp1) *DELZX,* real, dimension (ilg,igp2,igp1) *ZMAT,* real, dimension (ilg,igp1) *ZRMDR,* real, dimension(ilg) *FDTBND,* real, dimension (ilg) *WADD,* real, dimension (ilg) *TADD,* real, dimension (ilg,igp1) *FDT,* real, dimension (ilg,igp1) *TFDT,* real, dimension(ilg,ig) *THLMAX,* real, dimension(ilg,ig) *THTEST,* real, dimension(ilg,ig) *THLDUM,* real, dimension(ilg,ig) *THIDUM,* real, dimension (ilg,ig) *TDUMW,* real, dimension (ilg) *TUSED,* real, dimension(ilg) *RDUMMY,* real, dimension (ilg) *ZERO,* real, dimension(ilg) *WEXCES,* real, dimension(ilg) *XDRAIN,* real, dimension (ilg,ig) *THPOR,* real, dimension(ilg,ig) *THLRET,* real, dimension(ilg,ig) *THLMIN,* real, dimension (ilg,ig) *BI,* real, dimension(ilg,ig) *PSISAT,* real, dimension(ilg,ig) *GRKSAT,* real, dimension (ilg,ig) *THFC,* real, dimension (ilg,ig) *DELZW,* integer, dimension (ilg,ig) *ISAND,* integer, dimension (ilg) *IGRN,* integer, dimension (ilg) *IGRD,* integer, dimension (ilg) *IGDR,* integer, dimension (ilg) *IZERO,* integer *IVEG,* integer *IG,* integer *IGP1,* integer *IGP2,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

thliqx	Volumetric liquid water content of soil layer $[m^3m^{-3}]$
thicex	Volumetric frozen water content of soil layer $[m^3m^{-3}]$
tbarwx	Temperature of water in soil layer [C]
zpond	Depth of ponded water [m]
tpond	Temperature of ponded water [C]
basflw	Base flow from bottom of soil column $[kgm^{-2}]$
tbasfl	Temperature of base flow from bottom of soil column [K]
runoff	Total runoff from soil column [m]
trunof	Temperature of total runoff from soil column [K]
wmove	Water movement matrix $[m^3m^{-2}]$
tmove	Temperature matrix associated with ground water movement [C]
thlinf	Volumetric liquid water content behind the wetting front $[m^3m^{-3}]$
fi	Fractional coverage of subarea in question on modelled area [ ]
trmdr	Time remaining in current time step [s]
delzx	Permeable depth of soil layer [m]

At levels in the soil profile lower than the bottom of the wetting front, redistribution of soil liquid water proceeds in response to normal gravity and suction forces. Subroutine GRDRAN is called to calculate these redistributions. The time period TUSED that is passed to GRDRAN is set in the 100 loop to the time period over which infiltration was occurring during the current time step, except if the wetting front has passed the bottom of the lowest soil layer, in which case TUSED is set to zero (since the flows calculated by GRDRAN are not required). GRDRAN is called using dummy variables THLDUM, THIDUM and TDUMW, which are set in loop 125 to the liquid water content, the frozen water content and the water temperature of the soil layers respectively.

After GRDRAN has been called, the maximum value of the water movement index NINF is set to the number of soil layers plus 1. The values in the matrix ZRMDR, representing for each soil layer the depth that has not been affected by infiltration, are initialized to the soil permeable layer thicknesses DELZX. The water flows FDT coming out of GRDRAN are set to zero at the soil layer interfaces above the wetting front. For the layer containing the wetting front, if the flow at the bottom of the layer is upward, it is set to zero (to avoid possible overflows in liquid water content).

The values in the three-dimensional matrix ZMAT are initialized to zero. This matrix contains the depth of each soil layer J (including the dummy soil layer below the lowest layer) that is filled by water from level K in the water movement matrix WMOVE. In WMOVE, the first level contains the amount of water that has infiltrated at the surface during the time period in question, and each successive level K contains the amount of water in soil layer K-1 that has been displaced during the infiltration, down to the soil layer containing the wetting front. Thus, the number of levels in WMOVE that are used in the current infiltration calculations, NINF, is equal to LZF+1, where LZF is the index of the soil layer containing the wetting front; or to IGP1, the number of soil layers IG plus 1, if LZF is greater than IG, i.e. if the wetting front has penetrated below the bottom of the lowest soil layer into the underlying dummy layer. In the 400 loop, starting at the top of the soil profile, an attempt is made to assign each layer of WMOVE, converted into a depth by using the volumetric water content THLINF behind the wetting front for the soil layer, in turn to the K,J level of ZMAT. If the calculated value of ZMAT is greater than the available depth ZRMDR of the layer, ZMAT is set to ZRMDR, WMOVE is decremented by ZRMDR converted back to a water amount, and ZRMDR is set to zero. Otherwise the calculated value of ZMAT is accepted, ZRMDR is decremented by ZMAT, and WMOVE is set to zero. At the end of these calculations, any remaining residual amounts in the WMOVE matrix are assigned to ponded water, and the ponded water temperature is updated accordingly.

As a result of the above processes, the liquid water content and temperature of each soil layer (excluding the bottom, dummy layer) will be a combined result of infiltration processes (WADD, TADD), redistribution processes (WDRA, TDRA), and water in the layer that has remained unaffected (WREM, TREM). For each layer, WADD is calculated by summing over the respective ZMAT values corresponding to that layer multiplied by THLINF, and TADD by summing over the ZMAT and THLINF values multiplied by the respective TMOVE values. WREM is obtained as the product of the original water content THLIQX multiplied by ZRMDR, and TREM as the product of the water temperature TBARWX, THLIQX and ZRMDR. For the soil layer containing the wetting front, a check is carried out to determine whether the liquid moisture content resulting from the infiltration and drainage processes, THINFL, is less than the residual liquid moisture content THLMIN. If so, the flow FDT at the bottom of the layer is recalculated as the value required to keep THLIQX at THLMIN. WDRA is obtained from the difference between the water fluxes FDT at the top and bottom of the layer, supplied by GRDRAN, and TDRA is obtained from the water fluxes FDT and their corresponding temperatures TFDT. Finally, THLIQX is calculated as the sum of WADD, WREM and WDRA

normalized by DELZX, and TBARWX as the sum of TADD, TREM and TDRA normalized by the product of THLIQX and DELZX.

Lastly, the base flow BASFLW at the bottom of the soil profile and its temperature TBASFL are calculated. If the wetting front is located in the dummy soil layer below the soil profile, BASFLW is obtained by summing over the ZMAT values for the IGP1 level, multiplied by the dummy layer THLINF value and the fractional coverage FI of the modelled subarea. TBASFL is similarly obtained as the weighted average of the original TBASFL and the values of TMOVE corresponding to the ZMAT values. The overall subarea runoff RUNOFF and its temperature TRUNOF are calculated in the same manner without the FI weightings. Otherwise, the baseflow and total runoff are obtained from the value of FDT at the bottom of the IGDR layer, and their temperatures from the values of TFDT and FDT.

9.78 wetland_methane.f90 File Reference

Canadian Terrestrial Ecosystem Model (CTEM) Wetland and wetland methane subroutine.

Functions/Subroutines

- subroutine wetland_methane (hetrores, il1, il2, ta, wetfrac,npp, tbar, thliqg, currlat,sand,slopefrac,

9.78.1 Detailed Description

Canadian Terrestrial Ecosystem Model (CTEM) Wetland and wetland methane subroutine.

9.78.2 Function/Subroutine Documentation

9.78.2.1 subroutine wetland_methane (real, dimension(ilg), intent(in) *hetrores,* integer, intent(in) *il1,* integer, intent(in) *il2,* real, dimension(ilg), intent(in) *ta,* real, dimension(ilg), intent(in) *wetfrac,* real, dimension(ilg), intent(in) *npp,* real, dimension(ilg,ignd), intent(in) *tbar,* real, dimension(ilg,ignd), intent(in) *thliqg,* real, dimension(ilg), intent(in) *currlat,* real, dimension(ilg,ignd), intent(in) *sand,* real, dimension(ilg,8), intent(in) *slopefrac*)

Parameters

in	*il1*	il1=1
in	*il2*	il2=ilg
in	*hetrores*	heterotrophic respiration from main ctem program calculated as sum of litres + socres
in	*ta*	air temperature, k
in	*wetfrac*	prescribed fraction of wetlands in a grid cell
in	*npp*	grid-averaged npp from ctem (u-mol co2/m2.s)
in	*currlat*	centre latitude of grid cells in degrees
in	*slopefrac*	prescribed fraction of wetlands based on slope only(0.025, 0.05, 0.1, 0.15, 0.20, 0.25, 0.3 and 0.35 percent slope thresholds)
in	*tbar*	temperature of soil layers
in	*thliqg*	liquid soil moisture content (fraction)
in	*sand*	percentage sand in soil layers

Constants and parameters are located in ctem_params.f90

initialize required arrays to zero

initialization ends

Estimate the methane flux from wetlands for each grid cell scaling by the wetland fraction in a grid cell and set the methane flux to zero when screen temperature (ta) is below or at freezing this is consistent with recent flux measurements by the university of manitoba at churchill, manitoba

if (obswetf) then ! Use the read-in wetland locations as the CH4 producing area

else ! dynamically find the wetland locations

9.79 WFILL.f File Reference

Purpose: Evaluate infiltration of water into soil under unsaturated conditions.

Functions/Subroutines

- subroutine wfill (WMOVE, TMOVE, LZF, NINF, ZF, TRMDR, R, TR, PSIF, GRKINF, THLINF, THLIQX, TBARWX, DELZX, ZBOTX, DZF, TIMPND, WADJ, WADD, IFILL, IFIND, IG, IGP1, IGP2, ILG, IL1, IL2, JL, N)

9.79.1 Detailed Description

Purpose: Evaluate infiltration of water into soil under unsaturated conditions.

9.79.2 Function/Subroutine Documentation

9.79.2.1 subroutine wfill (real, dimension (ilg,igp2) *WMOVE,* real, dimension (ilg,igp2) *TMOVE,* integer, dimension (ilg) *LZF,* integer, dimension (ilg) *NINF,* real, dimension (ilg) *ZF,* real, dimension (ilg) *TRMDR,* real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg,igp1) *PSIF,* real, dimension(ilg,igp1) *GRKINF,* real, dimension(ilg,igp1) *THLINF,* real, dimension(ilg,igp1) *THLIQX,* real, dimension(ilg,igp1) *TBARWX,* real, dimension (ilg,igp1) *DELZX,* real, dimension (ilg,igp1) *ZBOTX,* real, dimension (ilg) *DZF,* real, dimension (ilg) *TIMPND,* real, dimension (ilg) *WADJ,* real, dimension (ilg) *WADD,* integer, dimension (ilg) *IFILL,* integer, dimension (ilg) *IFIND,* integer *IG,* integer *IGP1,* integer *IGP2,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

wmove	Water movement matrix $[m^3m^{-2}]$
tmove	Temperature matrix associated with ground water movement [C}
lzf	Index of soil layer in which wetting front is located
ninf	Number of levels involved in water movement
zf	Depth of the wetting front [m]
trmdr	Remainder of time step after unsaturated infiltration ceases [s]
r	Rainfall rate at ground surface $[ms^{-1}]$
tr	Temperature of rainfall [C]
psif	Soil water suction across the wetting front [m]
grkinf	Hydraulic conductivity of soil behind the wetting front $[ms^{-1}]$
thlinf	Volumetric liquid water content behind the wetting front $[m^3m^{-3}]$
thliqx	Volumetric liquid water content of soil layer $[m^3m^{-3}]$
tbarwx	Temperature of water in soil layer [C]

delzx	Permeable depth of soil layer $[m](\Delta z_{g,w})$
zbotx	Depth of bottom of soil layer [m]
ifill	Flag indicating whether unsaturated infiltration is occurring

The infiltration rate F_{inf} under conditions of a constant water supply can be expressed, e.g. in Mein and Larson (1973), as

F_{inf} =K_{inf} $[(\Psi_f + z_f)/z_f]$

where K_{inf} is the hydraulic conductivity of the soil behind the wetting front, Ψ_f is the soil moisture suction across the wetting front, and z_f is the depth of the wetting front. It can be seen that F_{inf} decreases with increasing z_f to an asymptotic value of K_{inf}. Thus, if the rainfall rate r is less than K_{inf}, the actual infiltration rate is limited by r, i.e. $F_{inf} = r$. Otherwise, F_{inf} will be equal to R until the right-hand side of the above equation becomes less than r, after which point the above equation applies and ponding of excess water begins on the surface. The depth of the wetting front at this time t_p can be calculated by setting F_{inf} equal to r in the above equation and solving for z_f. This results in:

$z_f = \Psi_f/[r/K_{inf}\breve{}1]$

The amount of water added to the soil up to the time of ponding is $t_p r$, or $z_f(\theta_{inf}\breve{}\theta_l)$, where θ_l and θ_{inf} are respectively the liquid water content of the soil before and after the wetting front has passed. Setting these two equal and solving for t_p results in

$t_p = z_f(\theta_{inf}\breve{}\theta_l)/r$

In the 100 loop, a check is done for each successive soil layer to compare the infiltration rate in the layer with the rainfall rate. If $K_{inf} < R$, a test calculation is performed to determine where the depth of the wetting front would theoretically occur at the ponding time t_p. If the calculated value of z_f is less than the depth of the top of the soil layer, z_f is set to the depth of the top of the layer; if z_f falls within the soil layer, that value of z_f is accepted. In both cases, the index LZF is set to the index of the layer, and the flag IFIND, indicating that z_f, has been successfully located, is set to 1. If the infiltration rate in the soil layer is greater than the rainfall rate, z_f is provisionally set to the bottom of the current layer, and LZF to the index of the next layer. IFIND remains zero. If the infiltration rate in the layer is vanishingly small, z_f is set to the depth of the top of the current layer, LZF to the index of the overlying layer, and IFIND to 1.

If LZF is greater than 1, some adjustment to the equation for t_p above is required to account for the fact that the values of θ_{inf} and θ_l in the layer containing the wetting front may differ from those in the overlying layers. The equation for t_p above can be rewritten as

$t_p = [z_f[\theta_{inf}(z_f)\breve{}\theta_l(z_f)] + w_{adj}]/r$

where w_{adj} is calculated as

$LZF - 1$

$w_{adj} = \Sigma[(\theta_{inf.i}\breve{}\theta_{l,i})\breve{}(\theta_{inf}(z_f)\breve{}\theta_l(z_f))]\Delta z_{g,w}$

$i = 1$

The adjusting volume WADJ is calculated in loop 200, and the time to ponding TIMPND in loop 250. If TIMPND is greater than the amount of time remaining in the current time step TRMDR, then unsaturated infiltration is deemed to be occurring over the entire time step. In this case, the amount of water infiltrating over the time step is assigned to the first level of the water movement matrix WMOVE and to the accounting variable WADD, and the temperature of the infiltrating water is assigned to the first level of the matrix TMOVE.

In loop 300 WADD is partitioned over the soil profile by comparing in turn the liquid water content of each soil layer with the calculated liquid water content behind the wetting front THLINF, and decrementing WADD layer by layer until a layer is reached in which the remainder of WADD is insufficient to raise the liquid water content to THLINF. If this condition is reached, LZF is set to the index of the soil layer; the depth of the wetting front DZF within the layer, obtained as WADD/(THLINF-THLIQX), is added to the depth of the bottom of the overlying layer to obtain ZF.

In loop 400, the water content in each soil layer J existing above ZF is assigned to the J+1 level of the water movement matrix WMOVE, and the respective water temperatures are assigned to TMOVE.

If TIMPND $<$ TRMDR, the amount of water infiltrating between the start of the time step and TIMPND is again assigned to the first level of the water movement matrix WMOVE, and the temperature of the infiltrating water is assigned to the first level of the matrix TMOVE. The depth DZF of the wetting front within the layer containing it is

calculated by subtracting the depth of the bottom of the overlying layer from ZF.

In loop 500, the water content in each soil layer J existing above ZF is assigned to the J+1 level of the water movement matrix WMOVE, and the respective water temperatures are assigned to TMOVE.

Finally, the time remaining in the current time step after the period of unsaturated infiltration is recalculated, and the counter NINF is set to LZF+1.

9.80 WFLOW.f File Reference

Evaluates infiltration of water into soil under saturated conditions.

Functions/Subroutines

- subroutine wflow (WMOVE, TMOVE, LZF, NINF, TRMDR, TPOND, ZPOND, R, TR, EVAP, PSIF, GRKINF, THLINF, THLIQX, TBARWX, DELZX, ZBOTX, FMAX, ZF, DZF, DTFLOW, THLNLZ, THLQLZ, DZDISP, WDISP, WABS, ITER, NEND, ISIMP, IGRN, IG, IGP1, IGP2, ILG, IL1, IL2, JL, N)

9.80.1 Detailed Description

Evaluates infiltration of water into soil under saturated conditions.

9.80.2 Function/Subroutine Documentation

9.80.2.1 subroutine wflow (real, dimension (ilg,igp2) *WMOVE,* real, dimension (ilg,igp2) *TMOVE,* integer, dimension (ilg) *LZF,* integer, dimension (ilg) *NINF,* real, dimension (ilg) *TRMDR,* real, dimension (ilg) *TPOND,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *R,* real, dimension (ilg) *TR,* real, dimension (ilg) *EVAP,* real, dimension (ilg,igp1) *PSIF,* real, dimension (ilg,igp1) *GRKINF,* real, dimension (ilg,igp1) *THLINF,* real, dimension (ilg,igp1) *THLIQX,* real, dimension (ilg,igp1) *TBARWX,* real, dimension (ilg,igp1) *DELZX,* real, dimension (ilg,igp1) *ZBOTX,* real, dimension (ilg) *FMAX,* real, dimension (ilg) *ZF,* real, dimension (ilg) *DZF,* real, dimension(ilg) *DTFLOW,* real, dimension(ilg) *THLNLZ,* real, dimension(ilg) *THLQLZ,* real, dimension(ilg) *DZDISP,* real, dimension (ilg) *WDISP,* real, dimension (ilg) *WABS,* integer, dimension (ilg) *ITER,* integer, dimension (ilg) *NEND,* integer, dimension (ilg) *ISIMP,* integer, dimension (ilg) *IGRN,* integer *IG,* integer *IGP1,* integer *IGP2,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *N*)

Parameters

wmove	Water movement matrix $[m^3m^{-2}]$
tmove	Temperature matrix associated with ground water movement [C]
lzf	Index of soil layer in which wetting front is located
ninf	Number of levels involved in water movement
trmdr	Time remaining in current time step [s]
tpond	Temperature of ponded water [C]
zpond	Depth of ponded water $[m](z_p)$
r	Rainfall rate at ground surface $[ms^{-1}]$
tr	Temperature of rainfall [C]
evap	Surface evaporation rate $[ms^{-1}]$
psif	Soil water suction across the wetting front $[m](\Psi_f)$
grkinf	Hydraulic conductivity of soil behind the wetting front $[ms^{-1}](K)$
thlinf	Volumetric liquid water content behind the wetting front $[m^3m^{-3}]$

thliqx	Volumetric liquid water content of soil layer $[m^3m^{-3}]$
tbarwx	Temperature of water in soil layer [C]
delzx	Permeable depth of soil layer $[m](\Delta z_{z,w})$
zbotx	Depth of bottom of soil layer [m]
fmax	Maximum infiltration rate, defined as minimum value of GRKINF
zf	Depth of the wetting front $[m](z_f)$
igrn	Flag to indicate whether infiltration is occurring

General calculations performed: The infiltration rate F_{inf} under saturated conditions is calculated as

$$F_{inf} = K_{inf}[(\Psi_f + z_f + z_p)/z_f]$$

where K_{inf} is the hydraulic conductivity of the soil behind the wetting front, Ψ_f is the soil moisture suction across the wetting front, z_f is the depth of the wetting front, and z_p is the depth of water ponded on the surface. Since Ψ_f will vary by soil layer, and z_f and z_p will vary with time, the period of infiltration is divided into two-minute segments. A maximum iteration flag NEND is defined as the number of iteration segments plus 100, as a safeguard against runaway iterations. Since the surface infiltration rate will be limited by the lowest infiltration rate encountered, a maximum infiltration rate FMAX is defined as the minimum value of GRKINF, the hydraulic conductivity behind the wetting front, in all of the soil layers above and including that containing the wetting front.

In loop 200, a check is performed to determine whether the liquid water content of the current soil layer, THLIQX, equals or exceeds that behind the wetting front, THLINF. If so, the depth of the wetting front is reset to the depth of the soil layer. The water content of the soil layer is assigned to the level of the water movement matrix WMOVE corresponding to the soil layer index plus 1, and the temperature of the water in the layer is assigned to the same level of the matrix TMOVE. FMAX is recalculated, and the flags LZF and NINF are each incremented by 1. The flag ISIMP is set to 1, indicating that the following calculations are to be bypassed for this iteration.

If ISIMP is not 1, the time period DTFLOW applying to the current iteration loop is set to two minutes or to the remainder of the time step, whichever is less. If GRKINF for the current soil layer is not vanishingly small, the flag ISIMP is set to -2. Otherwise, it is deemed that infiltration is suppressed, and the temperature and depth of water ponded on the surface are simply updated. The temperature of the ponded water is calculated as the weighted average of the current pond temperature and the rainfall added to it. The ponded water remaining after rainfall and evaporation have taken place is calculated as ZPTEST. If ZPTEST is less than zero, it is deduced that evaporation must have removed the ponded water before the end of the current iteration loop. If this is the case, the time period of the current iteration loop is recalculated as the amount of time required for the difference between the evaporation and the rainfall rates to consume the ponded water, and the pond depth ZPOND is set to zero. Otherwise, ZPOND is set to ZPTEST. Finally, ISIMP is set to -1.

The 400 loop addresses the infiltration process under saturated conditions. Such infiltration is modelled as “piston flow”. First the current infiltration rate FINF is calculated using the equation given above. If the wetting front has passed the bottom of the lowest soil layer, PSIF Cis neglected. If FINF is greater than the rainfall rate R, FINF is set equal to R; if FINF is greater than CFMAX, FINF is set equal to FMAX. If the wetting front has not passed the bottom of the lowest soil layer, the change in depth of the wetting front DZF over the current time interval is calculated from the amount of infiltrating water WINF and the volumetric water content behind the wetting front, THLINF. The amount of soil water WDISP displaced by this movement of the wetting front is calculated from DZF and THLIQX, and the depth to which this water penetrates, DZDISP, is calculated as WDISP/(THLINF – THLIQX). The amount of soil water WABS entrained by the movement of WDISP itself is calculated as the product of DZDISP and THLIQX. The change in depth of the wetting front, behind which the liquid water content of the soil layer is THLINF, is now the sum of the depth represented by infiltration of water at the surface, DZF, and the depth represented by displacement of the pre-existing soil water, DZDISP. If this change in depth causes the wetting front to move beyond the bottom of the current soil layer, DTFLOW is recalculated as the amount of time required for the composite wetting front to reach the bottom of the soil layer, and DZF, WDISP, DZDISP and WABS are likewise recalculated. As in the case for ISIMP = -1, the temperature of the ponded water on the surface is calculated as the weighted average of the current pond temperature and the rainfall added to it. The ponded water remaining after rainfall, infiltration and evaporation have taken place is calculated as ZPTEST. If ZPTEST is less than zero, it is deduced that infiltration and evaporation must have removed the ponded water before the end of the current iteration loop. If this is the case, the time period of the current iteration loop is recalculated as the amount of time required for the infiltration and evaporation minus the rainfall to consume the ponded water. The pond depth ZPOND is set to zero; if the wetting front has not passed the bottom of the lowest soil layer, DZF, WDISP, DZDISP and WABS are recalculated. Otherwise, ZPOND is set to ZPTEST. Finally, the first layer of the water movement matrix WMOVE is incremented with the amount of the infiltrated water, and the first layer of the matrix TMOVE with the temperature of the infiltrated water. The last layer

of WMOVE is incremented by WDISP+WABS, and the depth of the wetting front by DZF+DZDISP.

In the 500 loop, if ISIMP $<$ 0 (i.e. water movement has occurred), the time remaining in the current time step is recalculated. If the wetting front is at a soil layer boundary, if the time remaining is non-zero, if the wetting front is still within the modelled soil column, and if the calculated water content behind the wetting front is greater than zero, FMAX is updated, and LZF and NINF are incremented by 1. The NINF level of the TMOVE matrix is set to the water temperature of the new soil layer.

At the end of the iteration pass, checks are done in the 600 loop to ascertain whether the number of iteration passes is still less than NEND, whether either ponded water still exists or rain is still falling, and whether there is still time remaining in the current time step. If these conditions are all fulfilled, the counter NPNTS representing the number of points in the current vector of mosaic tiles for which infiltration is still occurring is incremented by 1. Otherwise, the iteration flag for the current tile is changed from 1 to 0, signaling the end of the saturated infiltration calculations for that tile.

9.81 WPREP.f File Reference

Purpose: Initialize subarea variables for surface water budget calculations, and perform preliminary calculations for diagnostic variables.

Functions/Subroutines

- subroutine wprep (THLQCO, THLQGO, THLQCS, THLQGS, THICCO, THICGO, THICCS, THICGS, HCPCO, HCPGO, HCPCS, HCPGS, GRKSC, GRKSG, GRKSCS, GRKSGS, SPCC, SPCG, SPCCS, SPCGS, TSPCC, TSPCG, TSPCCS, TSPCGS, RPCC, RPCG, RPCCS, RPCGS, TRPCC, TRPCG, TRPCCS, TRPCGS, EV↩PIC, EVPIG, EVPICS, EVPIGS, ZPONDC, ZPONDG, ZPNDCS, ZPNDGS, XSNOWC, XSNOWG, XSNOCS, XSNOGS, ZSNOWC, ZSNOWG, ZSNOCS, ZSNOGS, ALBSC, ALBSG, ALBSCS, ALBSGS, RHOSC, RH↩OSG, HCPSC, HCPSG, HCPSCS, HCPSGS, RUNFC, RUNFG, RUNFCS, RUNFGS, TRUNFC, TRUNFG, TRNFCS, TRNFGS, TBASC, TBASG, TBASCS, TBASGS, GFLXC, GFLXG, GFLXCS, GFLXGS, SUBLC, SUBLCS, WLOSTC, WLOSTG, WLSTCS, WLSTGS, RAC, RACS, SNC, SNCS, TSNOWC, TSNOWG, OV↩RFLW, SUBFLW, BASFLW, TOVRFL, TSUBFL, TBASFL, PCFC, PCLC, PCPN, PCPG, QFCF, QFCL, QFN, QFG, QFC, HMFG, ROVG, ROFC, ROFN, TRUNOF, THLIQX, THICEX, THLDUM, THIDUM, DT, RDUMMY, ZERO, IZERO, DELZZ, FC, FG, FCS, FGS, THLIQC, THLIQG, THICEC, THICEG, HCPC, HCPG, TBARC, TBARG, TBARCS, TBARGS, TBASE, TSURX, FSVF, FSVFS, RAICAN, SNOCAN, RAICNS, SNOCNS, E↩VAPC, EVAPCG, EVAPG, EVAPCS, EVPCSG, EVAPGS, RPCP, TRPCP, SPCP, TSPCP, RHOSNI, ZPOND, ZSNOW, ALBSNO, WSNOCS, WSNOGS, RHOSCS, RHOSGS, THPOR, HCPS, GRKSAT, ISAND, DELZW, DELZ, ILG, IL1, IL2, JL, IG, IGP1, NLANDCS, NLANDGS, NLANDC, NLANDG, RADD, SADD)

9.81.1 Detailed Description

Purpose: Initialize subarea variables for surface water budget calculations, and perform preliminary calculations for diagnostic variables.

9.81.2 Function/Subroutine Documentation

9.81.2.1 subroutine wprep (real, dimension(ilg,ig) *THLQCO,* real, dimension(ilg,ig) *THLQGO,* real, dimension(ilg,ig) *THLQCS,* real, dimension(ilg,ig) *THLQGS,* real, dimension(ilg,ig) *THICCO,* real, dimension(ilg,ig) *THICGO,* real, dimension(ilg,ig) *THICCS,* real, dimension(ilg,ig) *THICGS,* real, dimension (ilg,ig) *HCPCO,* real, dimension (ilg,ig) *HCPGO,* real, dimension (ilg,ig) *HCPCS,* real, dimension (ilg,ig) *HCPGS,* real, dimension (ilg,ig) *GRKSC,* real, dimension (ilg,ig) *GRKSG,* real, dimension(ilg,ig) *GRKSCS,* real, dimension(ilg,ig) *GRKSGS,* real, dimension (ilg) *SPCC,* real, dimension (ilg) *SPCG,* real, dimension (ilg) *SPCCS,* real, dimension (ilg) *SPCGS,* real, dimension (ilg) *TSPCC,* real, dimension (ilg) *TSPCG,* real, dimension(ilg) *TSPCCS,* real, dimension(ilg) *TSPCGS,* real, dimension (ilg) *RPCC,* real, dimension (ilg) *RPCG,* real, dimension (ilg) *RPCCS,* real, dimension (ilg) *RPCGS,* real, dimension (ilg) *TRPCC,* real, dimension (ilg) *TRPCG,* real, dimension(ilg) *TRPCCS,* real, dimension(ilg) *TRPCGS,* real, dimension (ilg) *EVPIC,* real, dimension (ilg) *EVPIG,* real, dimension(ilg) *EVPICS,* real, dimension(ilg) *EVPIGS,* real, dimension(ilg) *ZPONDC,* real, dimension(ilg) *ZPONDG,* real, dimension(ilg) *ZPNDCS,* real, dimension(ilg) *ZPNDGS,* real, dimension(ilg) *XSNOWC,* real, dimension(ilg) *XSNOWG,* real, dimension(ilg) *XSNOCS,* real, dimension(ilg) *XSNOGS,* real, dimension(ilg) *ZSNOWC,* real, dimension(ilg) *ZSNOWG,* real, dimension(ilg) *ZSNOCS,* real, dimension(ilg) *ZSNOGS,* real, dimension (ilg) *ALBSC,* real, dimension (ilg) *ALBSG,* real, dimension(ilg) *ALBSCS,* real, dimension(ilg) *ALBSGS,* real, dimension (ilg) *RHOSC,* real, dimension (ilg) *RHOSG,* real, dimension (ilg) *HCPSC,* real, dimension (ilg) *HCPSG,* real, dimension(ilg) *HCPSCS,* real, dimension(ilg) *HCPSGS,* real, dimension (ilg) *RUNFC,* real, dimension (ilg) *RUNFG,* real, dimension(ilg) *RUNFCS,* real, dimension(ilg) *RUNFGS,* real, dimension(ilg) *TRUNFC,* real, dimension(ilg) *TRUNFG,* real, dimension(ilg) *TRNFCS,* real, dimension(ilg) *TRNFGS,* real, dimension (ilg) *TBASC,* real, dimension (ilg) *TBASG,* real, dimension(ilg) *TBASCS,* real, dimension(ilg) *TBASGS,* real, dimension (ilg,ig) *GFLXC,* real, dimension (ilg,ig) *GFLXG,* real, dimension(ilg,ig) *GFLXCS,* real, dimension(ilg,ig) *GFLXGS,* real, dimension (ilg) *SUBLC,* real, dimension(ilg) *SUBLCS,* real, dimension(ilg) *WLOSTC,* real, dimension(ilg) *WLOSTG,* real, dimension(ilg) *WLSTCS,* real, dimension(ilg) *WLSTGS,* real, dimension (ilg) *RAC,* real, dimension (ilg) *RACS,* real, dimension (ilg) *SNC,* real, dimension (ilg) *SNCS,* real, dimension(ilg) *TSNOWC,* real, dimension(ilg) *TSNOWG,* real, dimension(ilg) *OVRFLW,* real, dimension(ilg) *SUBFLW,* real, dimension(ilg) *BASFLW,* real, dimension(ilg) *TOVRFL,* real, dimension(ilg) *TSUBFL,* real, dimension(ilg) *TBASFL,* real, dimension (ilg) *PCFC,* real, dimension (ilg) *PCLC,* real, dimension (ilg) *PCPN,* real, dimension (ilg) *PCPG,* real, dimension (ilg) *QFCF,* real, dimension (ilg) *QFCL,* real, dimension (ilg) *QFN,* real, dimension (ilg) *QFG,* real, dimension (ilg,ig) *QFC,* real, dimension (ilg,ig) *HMFG,* real, dimension (ilg) *ROVG,* real, dimension (ilg) *ROFC,* real, dimension (ilg) *ROFN,* real, dimension(ilg) *TRUNOF,* real, dimension(ilg,igp1) *THLIQX,* real, dimension(ilg,igp1) *THICEX,* real, dimension(ilg,ig) *THLDUM,* real, dimension(ilg,ig) *THIDUM,* real, dimension (ilg) *DT,* real, dimension(ilg) *RDUMMY,* real, dimension (ilg) *ZERO,* integer, dimension (ilg) *IZERO,* real, dimension (ilg,ig) *DELZZ,* real, dimension (ilg) *FC,* real, dimension (ilg) *FG,* real, dimension (ilg) *FCS,* real, dimension (ilg) *FGS,* real, dimension(ilg,ig) *THLIQC,* real, dimension(ilg,ig) *THLIQG,* real, dimension(ilg,ig) *THICEC,* real, dimension(ilg,ig) *THICEG,* real, dimension (ilg,ig) *HCPC,* real, dimension (ilg,ig) *HCPG,* real, dimension (ilg,ig) *TBARC,* real, dimension (ilg,ig) *TBARG,* real, dimension(ilg,ig) *TBARCS,* real, dimension(ilg,ig) *TBARGS,* real, dimension (ilg) *TBASE,* real, dimension(ilg,4) *TSURX,* real, dimension (ilg) *FSVF,* real, dimension (ilg) *FSVFS,* real, dimension(ilg) *RAICAN,* real, dimension(ilg) *SNOCAN,* real, dimension(ilg) *RAICNS,* real, dimension(ilg) *SNOCNS,* real, dimension (ilg) *EVAPC,* real, dimension(ilg) *EVAPCG,* real, dimension (ilg) *EVAPG,* real, dimension(ilg) *EVAPCS,* real, dimension(ilg) *EVPCSG,* real, dimension(ilg) *EVAPGS,* real, dimension (ilg) *RPCP,* real, dimension (ilg) *TRPCP,* real, dimension (ilg) *SPCP,* real, dimension (ilg) *TSPCP,* real, dimension(ilg) *RHOSNI,* real, dimension (ilg) *ZPOND,* real, dimension (ilg) *ZSNOW,* real, dimension(ilg) *ALBSNO,* real, dimension(ilg) *WSNOCS,* real, dimension(ilg) *WSNOGS,* real, dimension(ilg) *RHOSCS,* real, dimension(ilg) *RHOSGS,* real, dimension (ilg,ig) *THPOR,* real, dimension (ilg,ig) *HCPS,* real, dimension(ilg,ig) *GRKSAT,* integer, dimension (ilg,ig) *ISAND,* real, dimension (ilg,ig) *DELZW,* real, dimension (ig) *DELZ,* integer *ILG,* integer *IL1,* integer *IL2,* integer *JL,* integer *IG,* integer *IGP1,* integer *NLANDCS,* integer *NLANDGS,* integer *NLANDC,* integer *NLANDG,* real, dimension (ilg) *RADD,* real, dimension (ilg) *SADD*)

Parameters

thlqco	Subarea volumetric liquid water content of soil layers $[m^3m^{-3}]$
thlqgo	Subarea volumetric liquid water content of soil layers $[m^3m^{-3}]$
thlqcs	Subarea volumetric liquid water content of soil layers $[m^3m^{-3}]$
thlqgs	Subarea volumetric liquid water content of soil layers $[m^3m^{-3}]$
thicco	Subarea volumetric frozen water content of soil layers $[m^3m^{-3}](\theta_i)$
thicgo	Subarea volumetric frozen water content of soil layers $[m^3m^{-3}](\theta_i)$
thiccs	Subarea volumetric frozen water content of soil layers $[m^3m^{-3}](\theta_i)$
thicgs	Subarea volumetric frozen water content of soil layers $[m^3m^{-3}](\theta_i)$
hcpco	Subarea heat capacity of soil layers $[Jm^{-3}K^{-1}]$
hcpgo	Subarea heat capacity of soil layers $[Jm^{-3}K^{-1}]$
hcpcs	Subarea heat capacity of soil layers $[Jm^{-3}K^{-1}]$
hcpgs	Subarea heat capacity of soil layers $[Jm^{-3}K^{-1}]$
grksc	Subarea saturated hydraulic conductivity $[ms^{-1}](K_{sat})$
grksg	Subarea saturated hydraulic conductivity $[ms^{-1}](K_{sat})$
grkscs	Subarea saturated hydraulic conductivity $[ms^{-1}](K_{sat})$
grksgs	Subarea saturated hydraulic conductivity $[ms^{-1}](K_{sat})$
gflxc	Subarea heat flux between soil layers $[Wm^{-2}]$
gflxg	Subarea heat flux between soil layers $[Wm^{-2}]$
gflxcs	Subarea heat flux between soil layers $[Wm^{-2}]$
gflxgs	Subarea heat flux between soil layers $[Wm^{-2}]$
thldum	Internal CLASSW dummy variable for soil frozen water $[m^3m^{-3}]$
thidum	Internal CLASSW dummy variable for soil frozen water $[m^3m^{-3}]$
qfc	Water removed from soil layers by transpiration $[kgm^{-2}s^{-1}]$
hmfg	Energy associated with phase change of water in soil layers $[Wm^{-2}]$
thliqx	Internal CLASSW work array for soil frozen water $[m^3m^{-3}]$
thicex	Internal CLASSW work array for soil frozen water $[m^3m^{-3}]$
spcc	Subarea snowfall rate $[ms^{-1}]$
spcg	Subarea snowfall rate $[ms^{-1}]$
spccs	Subarea snowfall rate $[ms^{-1}]$
spcgs	Subarea snowfall rate $[ms^{-1}]$
tspcc	Subarea snowfall temperature [K/C]
tspcg	Subarea snowfall temperature [K/C]
tspccs	Subarea snowfall temperature [K/C]
tspcgs	Subarea snowfall temperature [K/C]
rpcc	Subarea rainfall rate $[ms^{-1}]$
rpcg	Subarea rainfall rate $[ms^{-1}]$
rpccs	Subarea rainfall rate $[ms^{-1}]$
rpcgs	Subarea rainfall rate $[ms^{-1}]$
trpcc	Subarea rainfall temperature [K/C]
trpcg	Subarea rainfall temperature [K/C]
trpccs	Subarea rainfall temperature [K/C]
trpcgs	Subarea rainfall temperature [K/C]
evpic	Subarea evapotranspiration rate going into CLASSW $[ms^{-1}]$
evpig	Subarea evapotranspiration rate going into CLASSW $[ms^{-1}]$

evpics	Subarea evapotranspiration rate going into CLASSW $[ms^{-1}]$
evpigs	Subarea evapotranspiration rate going into CLASSW $[ms^{-1}]$
zpondc	Subarea depth of surface ponded water [m]
zpondg	Subarea depth of surface ponded water [m]
zpndcs	Subarea depth of surface ponded water [m]
zpndgs	Subarea depth of surface ponded water [m]
xsnowc	Subarea fractional snow coverage [ ]
xsnowg	Subarea fractional snow coverage [ ]
xsnocs	Subarea fractional snow coverage [ ]
xsnogs	Subarea fractional snow coverage [ ]
zsnowc	Subarea depth of snow pack
zsnowg	Subarea depth of snow pack
zsnocs	Subarea depth of snow pack
zsnogs	Subarea depth of snow pack
albsc	Subarea snow albedo [ ]
albsg	Subarea snow albedo [ ]
albscs	Subarea snow albedo [ ]
albsgs	Subarea snow albedo [ ]
rhosc	Subarea snow density $[kgm^{-3}]$
rhosg	Subarea snow density $[kgm^{-3}]$
hcpsc	Subarea heat capacity of snow $[Jm^{-3}K^{-1}](C_s)$
hcpsg	Subarea heat capacity of snow $[Jm^{-3}K^{-1}](C_s)$
hcpscs	Subarea heat capacity of snow $[Jm^{-3}K^{-1}](C_s)$
hcpsgs	Subarea heat capacity of snow $[Jm^{-3}K^{-1}](C_s)$
runfc	Subarea total runoff [m]
runfg	Subarea total runoff [m]
runfcs	Subarea total runoff [m]
runfgs	Subarea total runoff [m]
trunfc	Subarea total runoff temperature [K]
trunfg	Subarea total runoff temperature [K]
trnfcs	Subarea total runoff temperature [K]
trnfgs	Subarea total runoff temperature [K]
tbasc	Subarea temperature of bedrock in third soil layer [C]
tbasg	Subarea temperature of bedrock in third soil layer [C]
tbascs	Subarea temperature of bedrock in third soil layer [C]
tbasgs	Subarea temperature of bedrock in third soil layer [C]
sublc	Subarea sublimation rate from vegetation $[ms^{-1}]$
sublcs	Subarea sublimation rate from vegetation $[ms^{-1}]$
wlostc	Subarea residual water not met by surface stores $[kgm^{-2}]$
wlostg	Subarea residual water not met by surface stores $[kgm^{-2}]$
wlstcs	Subarea residual water not met by surface stores $[kgm^{-2}]$
wlstgs	Subarea residual water not met by surface stores $[kgm^{-2}]$
rac	Subarea liquid water on canopy going into CLASSW $[kgm^{-2}]$
racs	Subarea liquid water on canopy going into CLASSW $[kgm^{-2}]$
snc	Subarea frozen water on canopy going into CLASSW $[kgm^{-2}]$
sncs	Subarea frozen water on canopy going into CLASSW $[kgm^{-2}]$
tsnowc	Subarea snowpack temperature [K]

tsnowg	Subarea snowpack temperature [K]
ovrflw	Overland flow from top of soil column [m]
subflw	Interflow from sides of soil column [m]
basflw	Base flow from bottom of soil column [m]
tovrfl	Temperature of overland flow from top of soil column [K]
tsubfl	Temperature of interflow from sides of soil column [K]
tbasfl	Temperature of base flow from bottom of soil column [K]
pcfc	Frozen precipitation intercepted by vegetation $[kgm^{-2}s^{-1}]$
pclc	Liquid precipitation intercepted by vegetation $[kgm^{-2}s^{-1}]$
pcpn	Precipitation incident on snow pack $[kgm^{-2}s^{-1}]$
pcpg	Precipitation incident on ground $[kgm^{-2}s^{-1}]$
qfcf	Sublimation from frozen water on vegetation $[kgm^{-2}s^{-1}]$
qfcl	Evaporation from liquid water on vegetation $[kgm^{-2}s^{-1}]$
qfn	Sublimation from snow pack $[kgm^{-2}s^{-1}]$
qfg	Evaporation from ground $[kgm^{-2}s^{-1}]$
rovg	Liquid/frozen water runoff from vegetation to ground surface $[kgm^{-2}s^{-1}]$
rofc	Liquid/frozen water runoff from vegetation $[kgm^{-2}s^{-1}]$
rofn	Liquid water runoff from snow pack $[kgm^{-2}s^{-1}]$
trunof	Temperature of total runoff [K]
dt	Time stepping variable used in GRINFL/GRDRAN [s]
rdummy	Dummy variable
zero	Zero vector used in several subroutines [ ]
izero	Zero integer flag used in GRINFL
fc	Subarea fractional coverage of modelled area [ ]
fg	Subarea fractional coverage of modelled area [ ]
fcs	Subarea fractional coverage of modelled area [ ]
fgs	Subarea fractional coverage of modelled area [ ]
fsvf	Sky view factor of ground under vegetation canopy [ ]
fsvfs	Sky view factor of snow under vegetation canopy [ ]
raican	Intercepted liquid water stored on canopy over ground $[kgm^{-2}]$
snocan	Intercepted frozen water stored on canopy over ground $[kgm^{-2}]$
raicns	Intercepted liquid water stored on canopy over snow $[kgm^{-2}]$
snocns	Intercepted frozen water stored on canopy over snow $[kgm^{-2}]$
evapc	Evaporation from vegetation over ground $[ms^{-1}]$
evapcg	Evaporation from ground under vegetation $[ms^{-1}]$
evapg	Evaporation from bare ground $[ms^{-1}]$
evapcs	Evaporation from vegetation over snow $[ms^{-1}]$
evpcsg	Evaporation from snow under vegetation $[ms^{-1}]$
evapgs	Evaporation from snow on bare ground $[ms^{-1}]$
rpcp	Rainfall rate over modelled area $[ms^{-1}]$
spcp	Snowfall rate over modelled area $[ms^{-1}]$
tspcp	Snowfall temperature over modelled area [C]
rhosni	Density of fresh snow $[kgm^{-3}]$
zpond	Depth of ponded water on surface [m]
zsnow	Depth of snow pack $[m](z_s)$
albsno	Albedo of snow [ ]
wsnocs	Liquid water content of snow pack under vegetation $[kgm^{-2}](w_s)$

wsnogs	Liquid water content of snow pack in bare areas $[kgm^{-2}](w_s)$
rhoscs	Density of snow under vegetation $[kgm^{-3}](\rho_s)$
rhosgs	Density of snow in bare areas $[kgm^{-3}](\rho_s)$
tbase	Temperature of bedrock in third soil layer [K]
tsurx	Ground surface temperature over subarea [K]
thliqc	Liquid water content of soil layers under vegetation $[m^3m^{-3}]$
thliqg	Liquid water content of soil layers in bare areas $[m^3m^{-3}]$
thicec	Frozen water content of soil layers under vegetation $[m^3m^{-3}]$
thiceg	Frozen water content of soil layers in bare areas $[m^3m^{-3}]$
tbarc	Subarea temperatures of soil layers $[C](T_g)$
tbarg	Subarea temperatures of soil layers $[C](T_g)$
tbarcs	Subarea temperatures of soil layers $[C](T_g)$
tbargs	Subarea temperatures of soil layers $[C](T_g)$
hcpc	Heat capacity of soil layers under vegetation $[Jm^{-3}K^{-1}]$
thpor	Pore volume in soil layer $[m^3m^{-3}](\theta_p)$
hcps	Heat capacity of soil material $[Jm^{-3}K^{-1}]$
grksat	Saturated hydraulic conductivity of soil layers $[ms^{-1}]$
delzz	Soil layer depth variable used in TWCALC/TMCALC [m]
delzw	Permeable thickness of soil layer [m]
delz	Overall thickness of soil layer [m]
isand	Sand content flag

In the first three loops, various subarea arrays and internal CLASSW variables are initialized.

At the end of the 100 loop, a preliminary calculation of the precipitation diagnostics over the four subareas is carried out, as follows:

The rainfall incident on the vegetation, PCLC, is summed over the vegetated subareas FC and FCS, minus the fraction that falls through the gaps in the canopy (denoted by the sky view factors FSVF and FSVFS respectively). The rainfall incident on the snowpack PCPN is the sum of the rainfall on the snow- covered bare area FGS, and that on the snow under the gaps in the canopy in subarea FCS. The rainfall incident on bare ground, PCPG, is the sum of the rainfall on the snow-free bare area FG, and that on the ground under the gaps in the canopy in subarea FC. The snowfall incident on the vegetation, PCFC, as in the case of rainfall, is summed over the vegetation subareas FC and FCS, minus the fraction that falls through the gaps in the canopy. The remaining amount is assigned to snowfall incident on the snow pack, PCPN.

In loops 200 to 550, each of the four subareas is addressed in turn. Additional variables are initialized, and an empirical correction is applied to the saturated hydraulic conductivity in each of the soil layers over the subarea, to account for the viscosity of water at the layer temperature (Dingman, 2002):

$$K'_{sat} = (1.7915 * 10^{-3})K_{sat}/[(2.0319 * 10^{-4}) + (1.5883 * 10^{-3})exp(-T_g^{0.9}/22.0)]$$

Next, a preliminary calculation of the water vapour flux diagnostics for each subarea is carried out. Over the vegetated subareas, first the canopy evaporative flux is assigned to sublimation if there is snow present on the canopy (since it is assumed that any water present exists within or underneath the snow). Then, if the sublimation rate is upward, it is assigned to the diagnostic variable QFCF; if it is downward, the portion of the flux corresponding to the canopy-covered area is assigned to QFCF and that corresponding to the gap area to QFN. Similarly, if the evaporation rate is upward it is assigned to the diagnostic variable QFCL; if it is downward, the flux corresponding to the canopy-covered area is assigned to QFCL and that corresponding to the gap area to QFN over subarea FCS and to QFG over subarea FC. Over the non-vegetated subareas, the evaporation rates are assigned to QFN for subarea GS, and to QFG for subarea G.

For the purposes of the subsequent water balance calculations done in the other CLASSW subroutines, the subarea snowfall is lumped together with any simultaneously occurring sublimation, and the subarea rainfall with any simultaneously occurring evaporation. Depending on whether the sum of the snowfall and the sublimation, and the sum of the rainfall and the evaporation, are positive (downward) or negative (upward), corrections are applied to the appropriate diagnostic variables, and the snowfall/rainfall are set to the net flux if downward and the sublimation/evaporation are set to the net flux if upward. The smaller of the two fluxes in the sums are set to zero.

Finally, ponded water and snow pack physical characteristics are set, including the snow heat capacity, which is calculated from the heat capacities of ice and water C_i and C_w, the snow, ice and water densities ρ_s ρ_i, and ρ_w, and the water content and depth of the snow pack w_s and z_s, as follows:

$C_s = C_i[\rho_s/\rho_i] + C_w w_s/[\rho_w z_s]$

9.82 XIT.f File Reference

Purpose: Print the name of the subroutine and an error code when an error condition is encountered.

Functions/Subroutines

- subroutine xit (NAME, N)

9.82.1 Detailed Description

Purpose: Print the name of the subroutine and an error code when an error condition is encountered.

9.82.2 Function/Subroutine Documentation

9.82.2.1 subroutine xit (character∗(∗) *NAME, N*)

Parameters

name	Name of the subroutine in which the error was found N: error code

In CLASS, this subroutine is called when a test of ambient values of selected variables is performed and an abnormal condition is encountered. The name of the subroutine in which the condition arose is passed in, and is printed together with an error code, flagging the location of the error in the subroutine. A call to abort is then executed.

Bibliography

[1] Paul B Alton. Retrieval of seasonal rubisco-limited photosynthetic capacity at global FLUXNET sites from hyperspectral satellite remote sensing: Impact on carbon modelling. *Agric. For. Meteorol.*, 232:74–88, 15 January 2017. 221

[2] M. O. Andreae and P. Merlet. Emission of trace gases and aerosols from biomass burning. *Global Biogeochem. Cycles*, 15(4):955–966, 2001. 38

[3] V Arora and G Boer. Achieving coexistence: Comment on "modelling rainforest diversity: The role of competition" by bampfylde et al.(2005). *Ecol. Modell.*, 2006. 1, 160, 161, 163

[4] V K Arora and G J Boer. Uncertainties in the 20th century carbon budget associated with land use change. *Glob. Chang. Biol.*, 16(12):3327–3348, 2010. 62

[5] V. K. Arora, G. J. Boer, J. R. Christian, C. L. Curry, K. L. Denman, K. Zahariev, G. M. Flato, J. F. Scinocca, W. J. Merryfield, and W. G. Lee. The effect of terrestrial photosynthesis down regulation on the Twentieth-Century carbon budget simulated with the CCCma earth system model. *J. Clim.*, 22(22):6066–6088, 2009. 218, 219

[6] V. K. Arora, J. F. Scinocca, G. J. Boer, J. R. Christian, K. L. Denman, G. M. Flato, V. V. Kharin, W. G. Lee, and W. J. Merryfield. Carbon emission limits required to satisfy future representative concentration pathways of greenhouse gases. *Geophys. Res. Lett.*, 38(5):L05805, 2011. 1

[7] Vivek K. Arora. Simulating energy and carbon fluxes over winter wheat using coupled land surface and terrestrial ecosystem models. *Agric. For. Meteorol.*, 118(1-2):21–47, August 2003. 1, 44, 195, 217

[8] Vivek K. Arora and George J. Boer. A representation of variable root distribution in dynamic vegetation models. *Earth Interact.*, 7(6):1–19, 2003. 1, 106, 204

[9] Vivek K Arora and George J Boer. Fire as an interactive component of dynamic vegetation models. *J. Geophys. Res.*, 110(G2):G02008, 1 December 2005. 1, 35

[10] Vivek K. Arora and George J. Boer. A parameterization of leaf phenology for the terrestrial ecosystem component of climate models. *Glob. Chang. Biol.*, 11(1):39–59, 2005. 1, 95, 208, 213

[11] Vivek K. Arora and George J. Boer. Simulating competition and coexistence between plant functional types in a dynamic vegetation model. *Earth Interact.*, 10(10):1–30, 2006. 1, 160, 161

[12] Timothy J Ball, Ian E Woodrow, and Joseph A Berry. A model predicting stomatal conductance and its contribution to the control of photosynthesis under different environmental conditions. In *Progress in Photosynthesis Research*, pages 221–224. Springer Netherlands, 1 January 1987. 217

[13] William L Bauerle, Ram Oren, Danielle A Way, Song S Qian, Paul C Stoy, Peter E Thornton, Joseph D Bowden, Forrest M Hoffman, and Robert F Reynolds. Photoperiodic regulation of the seasonal pattern of photosynthetic capacity and the implications for carbon cycling. *Proc. Natl. Acad. Sci. U. S. A.*, 109(22):8612–8617, 29 May 2012. 221

[14] I Colin Prentice, Martin T Sykes, and Wolfgang Cramer. A simulation model for the transient effects of climate change on forest landscapes. *Ecol. Modell.*, 65(1-2):51–70, January 1993. 206

[15] GJ Collatz, M Ribas-Carbo, and JA Berry. Coupled Photosynthesis-Stomatal conductance model for leaves of C 4 plants. *Funct. Plant Biol.*, 19(5):519–538, 1 January 1992. 204, 217

[16] G.james Collatz, J.timothy Ball, Cyril Grivet, and Joseph A Berry. Physiological and environmental regulation of stomatal conductance, photosynthesis and transpiration: a model that includes a laminar boundary layer. *Agric. For. Meteorol.*, 54(2–4):107–136, April 1991. 204, 217

[17] P M Cox, R A Betts, C B Bunton, R. L. H. Essery, P R Rowntree, and J Smith. The impact of new land surface physics on the GCM simulation of climate and climate sensitivity. *Clim. Dyn.*, 15(3):183–203, 1 March 1999. 217

[18] PM Cox. Description of the TRIFFID dynamic global vegetation model. Technical Report 24, Hadley Centre, 2001. 44, 195

[19] Charles L Curry. Modeling the soil consumption of atmospheric methane at the global scale. *Global Biogeochem. Cycles*, 21(4):GB4012, 1 December 2007. 235

[20] G D Farquhar, S von Caemmerer, and J A Berry. A biochemical model of photosynthetic CO2 assimilation in leaves of C3 species. *Planta*, 149(1):78–90, 1 June 1980. 217

[21] C H Field and H A Mooney. Photosynthesis–nitrogen relationship in wild plants. In T J Givnish, editor, *On the economy of plant form and function*, pages 25–55. agris.fao.org, 1986. 219

[22] R P Guyette, R M Muzika, and D C Dey. Dynamics of an anthropogenic fire regime. *Ecosystems*, 5(5):472–486, 1 August 2002. 35

[23] Torsten Ingestad and Ann-britt Lund. Theory and techniques for steady state mineral nutrition and growth of plants. *Scand. J. For. Res.*, 1(1-4):439–453, 1986. 219

[24] R B Jackson, J Canadell, J R Ehleringer, H A Mooney, O E Sala, and E D Schulze. A global analysis of root distributions for terrestrial biomes. *Oecologia*, 108(3):389–411, 1 November 1996. 106

[25] Esteban G. Jobbágy and Robert B. Jackson. The vertical distribution of soil organic carbon and its relation to climate and vegetation. *Ecol. Appl.*, 10(2):423–436, 2000. 44, 195

[26] Miko U.f. Kirschbaum. The temperature dependence of soil organic matter decomposition, and the effect of global warming on soil organic C storage. *Soil Biol. Biochem.*, 27(6):753–760, June 1995. 44, 195

[27] Kees Klein Goldewijk, Arthur Beusen, and Peter Janssen. Long-term dynamic modeling of global population and built-up area in a spatially explicit way: HYDE 3.1. *Holocene*, 20(4):565–573, 1 June 2010. 36

[28] S. Kloster, N. M. Mahowald, J. T. Randerson, and P. J. Lawrence. The impacts of climate, land use, and demography on fires during the 21st century simulated by CLM-CN. *Biogeosciences*, 9(1):509–525, 26 January 2012. 35

[29] S. Kloster, N. M. Mahowald, J. T. Randerson, P. E. Thornton, F. M. Hoffman, S. Levis, P. J. Lawrence, J. J. Feddema, K. W. Oleson, and D. M. Lawrence. Fire dynamics during the 20th century simulated by the community land model. *Biogeosciences*, 7(6):1877–1902, 11 June 2010. 35, 36, 37

[30] Christopher J. Kucharik, Jonathan A. Foley, Christine Delire, Veronica A. Fisher, Michael T. Coe, John D. Lenters, Christine Young-Molling, Navin Ramankutty, John M. Norman, and Stith T. Gower. Testing the performance of a dynamic global ecosystem model: Water balance, carbon balance, and vegetation structure. *Global Biogeochem. Cycles*, 14(3):795–825, 2000. 35

[31] R Leuning. A critical appraisal of a combined stomatal-photosynthesis model for C3 plants. *Plant Cell Environ.*, 1995. 217, 219

[32] F Li, X D Zeng, and S Levis. A process-based fire parameterization of intermediate complexity in a dynamic global vegetation model. *Biogeosciences*, 9(7):2761–2780, 30 July 2012. 35, 36, 37, 38

[33] R. Li and V. K. Arora. Effect of mosaic representation of vegetation in land surface schemes on simulated energy and carbon balances. *Biogeosciences*, 9(1):593–605, 31 January 2012. 1

[34] J Lloyd and J A Taylor. On the temperature dependence of soil respiration. *Funct. Ecol.*, 8(3):315–323, 1 June 1994. 44, 195

[35] Mkb Ludeke, F W Badeck, R D Otto, C Hager, S Donges, and others. The frankfurt biosphere model: a global process-oriented model of seasonal and long-term COz exchange between terrestrial ecosystems and the atmosphere. . . . . *Climate Research*, 4:143–166, 1994. 96

[36] J R Melton and V K Arora. Sub-grid scale representation of vegetation in global land surface schemes: implications for estimation of the terrestrial carbon sink. *Biogeosciences*, 11(4):1021–1036, 21 February 2014. 1

[37] J R Melton and V K Arora. Sub-grid scale representation of vegetation in global land surface schemes: implications for estimation of the terrestrial carbon sink. *Biogeosciences*, 11(4):1021–1036, 21 February 2014. 44, 195

[38] J R Melton and V K Arora. Competition between plant functional types in the canadian terrestrial ecosystem model (CTEM) v. 2.0. *Geoscientific Model Development*, 27 January 2016. 1, 10

[39] M. Migliavacca, A. Dosio, S. Kloster, D. S. Ward, A. Camia, R. Houborg, T. Houston Durrant, N. Khabarov, A. A. Krasovskii, J. San Miguel-Ayanz, and A. Cescatti. Modeling burned area in europe with the community land model. *Journal of Geophysical Research: Biogeosciences*, 118(1):265–279, 2013. 35

[40] Yoshiyuki Miyazawa and Kihachiro Kikuzawa. Physiological basis of seasonal trend in leaf photosynthesis of five evergreen broad-leaved species in a temperate deciduous forest. *Tree Physiol.*, 26(2):249–256, February 2006. 221

[41] P R Moorcroft, G C Hurtt, and S W Pacala. A method for scaling vegetation dynamics: The ecosystem demography model (ED). *Ecol. Monogr.*, 71(4):557–586, 2001. 35

[42] WJ Parton, A Haxeltine, P Thornton, R Anne, and Melannie Hartman. Ecosystem sensitivity to land-surface models and leaf area index. *Glob. Planet. Change*, 13(1–4):89–98, June 1996. 204

[43] Thijs L Pons and Rob A M Welschen. Midday depression of net photosynthesis in the tropical rainforest tree eperua grandiflora: contributions of stomatal and internal conductances, respiration and rubisco functioning. *Tree Physiol.*, 23(14):937–947, 4 October 2003. 204

[44] Colin Price and David Rind. What determines the cloud-to-ground lightning fraction in thunderstorms? *Geophys. Res. Lett.*, 20(6):463–466, 1993. 35

[45] P B Reich, M B Walters, M G Tjoelker, D Vanderklein, and C Buschena. Photosynthesis and respiration rates depend on leaf and root morphology and nitrogen concentration in nine boreal tree species differing in relative growth rate. *Funct. Ecol.*, 12(3):395–405, 1998. 204

[46] Michael G. Ryan. Effects of climate change on plant respiration. *Ecol. Appl.*, 1(2):157–167, 1991. 204

[47] Piers J. Sellers, Compton J. Tucker, G. James Collatz, Sietse O. Los, Christopher O. Justice, Donald A. Dazlich, and David A. Randall. A revised land surface parameterization (SiB2) for atmospheric GCMS. part II: The generation of global fields of terrestrial biophysical parameters from satellite data. *J. Clim.*, 9(4):706–737, 1996. 217

[48] Evan Siemann and William E Rogers. Changes in light and nitrogen availability under pioneer trees may indirectly facilitate tree invasions of grasslands. *J. Ecol.*, 91(6):923–931, 2003. 161

[49] S. Sitch, B. Smith, I. C. Prentice, A. Arneth, A. Bondeau, W. Cramer, J. O. Kaplan, S. Levis, W. Lucht, M. T. Sykes, K. Thonicke, and S. Venevsky. Evaluation of ecosystem dynamics, plant geography and terrestrial carbon cycling in the LPJ dynamic global vegetation model. *Glob. Chang. Biol.*, 9(2):161–185, 2003. 206

[50] Mark G Tjoelker, Jacek Oleksyn, and Peter B Reich. Modelling respiration of vegetation: evidence for a general temperature-dependent Q10. *Glob. Chang. Biol.*, 7(2):223–230, 2001. 204

[51] D Verseghy. *CLASS – The Canadian land surface scheme*. Climate Research Division, Science and Technology Branch, Environment Canada, 2012. 1

[52] Vito Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560, October 1926. 160

[53] Audrey Wang, David T Price, and Vivek Arora. Estimating changes in global vegetation cover (1850–2100) for use in climate models. *Global Biogeochem. Cycles*, 20(3):GB3028, 1 September 2006. 1, 62

[54] R H Waring. Estimating forest growth and efficiency in relation to canopy leaf area. *Adv. Ecol. Res.*, 1983. 206

[55] Liukang Xu and Dennis D Baldocchi. Seasonal trends in photosynthetic parameters and stomatal conductance of blue oak (quercus douglasii) under prolonged summer drought and high temperature. *Tree Physiol.*, 23(13):865–877, 9 September 2003. 204

Index