The Community Noah-MP Land Surface Modeling System Technical Description Version 5.0

Cenlin He
Prasanth Valayamkunnath
Michael Barlage
Fei Chen
David Gochis
Ryan Cabell
Tim Schneider
Roy Rasmussen
Guo-Yue Niu
Zong-Liang Yang
Dev Niyogi
Michael Ek

NCAR Technical Notes
NCAR/TN-575+STR

National Center for Atmospheric Research
P. O. Box 3000
Boulder, Colorado
80307-3000
www.ucar.edu

NCAR | National Center for Atmospheric Research
UCAR | Atmospheric Research

National Science Foundation
NCAR IS SPONSORED BY THE NSF

NCAR TECHNICAL NOTES

http://library.ucar.edu/research/publish-technote

The Technical Notes series provides an outlet for a variety of NCAR Manuscripts that contribute in specialized ways to the body of scientific knowledge but that are not yet at a point of a formal journal, monograph or book publication. Reports in this series are issued by the NCAR scientific divisions, serviced by OpenSky and operated through the NCAR Library. Designation symbols for the series include:

EDD – Engineering, Design, or Development Reports
Equipment descriptions, test results, instrumentation, and operating and maintenance manuals.

IA – Instructional Aids
Instruction manuals, bibliographies, film supplements, and other research or instructional aids.

PPR – Program Progress Reports
Field program reports, interim and working reports, survey reports, and plans for experiments.

PROC – Proceedings
Documentation or symposia, colloquia, conferences, workshops, and lectures. (Distribution maybe limited to attendees).

STR – Scientific and Technical Reports
Data compilations, theoretical and numerical investigations, and experimental results.

The National Center for Atmospheric Research (NCAR) is operated by the nonprofit University Corporation for Atmospheric Research (UCAR) under the sponsorship of the National Science Foundation. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

National Center for Atmospheric Research
P. O. Box 3000
Boulder, Colorado 80307-3000

NCAR/TN-575+STR
NCAR Technical Note

2023-3

The Community Noah-MP Land Surface Modeling System Technical Description Version 5.0

Cenlin He, Fei Chen, David Gochis, Ryan Cabell, Tim Schneider, Roy Rasmussen, Michael Ek
Research Applications Laboratory, NCAR, Boulder, Colorado, USA

Prasanth Valayamkunnath
Research Applications Laboratory, NCAR, Boulder, Colorado, USA
Indian Institute of Science Education and Research, Thiruvananthapuram, India

Michael Barlage
NOAA Environmental Modeling Center, College Park, Maryland, USA

Guo-Yue Niu
University of Arizona, Tucson, Arizona, USA

Zong-Liang Yang, Dev Niyogi
University of Texas Austin, Austin, Texas, USA

NCAR Research Applications Laboratory

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
P. O. Box 3000
BOULDER, COLORADO 80307-3000
ISSN Print Edition 2153-2397
ISSN Electronic Edition 2153-2400

The Community Noah-MP Land Surface Modeling System

Technical Description

Version 5.0

Originated: March 7, 2023

Cenlin He[1], Prasanth Valayamkunnath[1,5], Michael Barlage[2], Fei Chen[1],
David Gochis[1], Ryan Cabell[1], Tim Schneider[1], Roy Rasmussen[1], Guo-Yue Niu[3],
Zong-Liang Yang[4], Dev Niyogi[4], Michael Ek[1]

1. Research Applications Laboratory, National Center for Atmospheric Research, USA
2. NOAA Environmental Modeling Center, USA
3. University of Arizona, USA
4. University of Texas Austin, USA
5. Indian Institute of Science Education and Research, Thiruvananthapuram, India

Citation: He, C., P. Valayamkunnath, M. Barlage, F. Chen, D. Gochis, R. Cabell, T. Schneider, R. Rasmussen, G.-Y. Niu, Z.-L. Yang, D. Niyogi, and M. Ek (2023): The Community Noah-MP Land Surface Modeling System Technical Description Version 5.0. NCAR Technical Note NCAR/TN-575+STR, doi:10.5065/ew8g-yr95

National Center for Atmospheric Research P. O. Box 3000, Boulder, Colorado 80307-300

Preface

This Technical Description document presents the details of modernized/refactored community Noah-MP land surface modeling system architecture and physics released in Version 5.0 in March 2023.

Acknowledgements

The funding support for modernizing/refactoring the community Noah-MP LSM has been provided by:

The NCAR Water System Program

The United States Geological Survey (USGS)

The National Oceanic and Atmospheric Administration (NOAA)

Table of Contents

1. Introduction

The overarching goal of this technical description documentation is to describe the new modeling structure of the modernized/refactored community Noah land surface model (LSM) with Multiple-Physics (hereafter Noah-MP), its physics parameterizations, numerical implementation, coding conventions, software architecture and operating instructions, as released in Version 5.0. The open-source community Noah-MP LSM is maintained and supported by the National Center for Atmospheric Research (NCAR) (https://github.com/NCAR/noahmp). Noah-MP is designed to simulate land surface energy and water balance (both uncoupled and coupled with an atmospheric model) at various spatial scales spanning from point scale locally to ~100-km resolution globally, and temporal scales spanning from sub-daily to decadal time scales. Noah-MP can accurately estimate soil moisture (both liquid water and ice content), soil temperature, skin temperature, snow depth (or snowpack density), snow water equivalent, canopy water content, and surface energy, water, and CO_2 fluxes (if dynamic vegetation physics is activated). Furthermore, with the optional physics schemes of Noah-MP, it can simulate crop growth (both seasonal cycles and yield), irrigation water requirements, subsurface tile drainage, groundwater dynamics, and urban processes. Scientific reasoning and evaluation of these parameterizations can be found in the referenced scientific papers of this documentation. Various community modeling efforts, like the High-Resolution Land Data Assimilation System (HRLDAS), the Weather Research and Forecasting (WRF) model, the Model for Prediction Across Scales (MPAS), the NOAA Unified Forecasting System (UFS), the WRF-Hydro/National Water Model (NWM), and NASA Land Information System (LIS), employs Noah-MP for land surface turbulent flux and momentum flux estimations.

1.1 Model History

The Noah LSM, which was originally named Oregon State University (OSU) LSM, was developed by coupling the diurnally dependent Penman potential evaporation approach of Mahrt and Ek (1984), the multilayer soil model of Mahrt and Pan (1984), and a simple canopy model of Pan and Mahrt (1987) and Ek and Mahrt (1991). Beginning in 1990, and accelerating after 1993 under the sponsorship from the GEWEX/GCIP Program Office of NOAA/OGP via collaboration with numerous GCIP Principal Investigators (PIs), Kenneth Mitchell, Fei Chen and Michael Ek at the Environmental Modeling Center (EMC) of the National Centers for Environmental Prediction (NCEP) joined with the NWS Office of Hydrology (OH) and the NESDIS Office of Research and Applications (ORA), the project started to pursue and refine a modern-era LSM suitable for use in NCEP operational weather and climate prediction models. The OSU LSM was evaluated and further developed to include Jarvis canopy resistance formulation, subgrid-scale variability of soil moisture, surface runoff, seasonal vegetation cycle (Chen et al. 1996, Schaake et al. 1996), a new atmospheric surface-layer parameterization (Chen et al. 1997), a physically-based parameterization of frozen soil and a new snow accumulation/ablation scheme (Koren et al. 1999), and a new iterative procedure to represent the snow evaporation/sublimation and melting process (Chen and Dudhia 2001). As a result of several test and evaluation efforts in the 1990s which showed good performance by the OSU LSM, EMC chose the OSU LSM for its implementation in the operational NCEP mesoscale Eta weather model (Chen et al., 1996, 1997; Mitchell et al., 1999, 2000; Ek et al., 2003). The enhanced OSU LSM was, along with comprehensive vegetation and soil datasets, implemented in the NCAR fifth-generation Mesoscale Model (MM5, Chen and Dudhia 2001), and started its journey as a community LSM.

In 2000, the OSU LSM was named as NOAH to represent a collaboration among the following groups:

- **NCEP/EMC**: NCEP Environmental Modeling Center (EMC) (Mitchell, Chen, Ek, Lin, Marshall, Janjic, Manikin, Lohmann, Grunmann, Pan)
- **OSU**: Oregon State University (Mahrt, Pan, Ek, Kim, Rusher)
- **AFWA**: Air Force Weather Agency (Moore, Mitchell, Gayno) and **AFRL**: Air Force Research Lab (Mitchell, Hahn, Chang, Yang)
- **HL**: NWS Hydrology Lab (Schaake, Koren, Duan)

Shortly after, the acronym NOAH was renamed as Noah to embrace the nature of a true community model. In 2003, NCEP and NCAR joined forces to unify the development of the Noah LSM to create an unified Noah LSM (so named Noah V.3) to be implemented in the WRF and in NCEP weather and climate models. Various versions of the unified Noah LSM were maintained in the official release of the WRF model and at https://ral.ucar.edu/solutions/products/unified-wrf-noah-lsm.

Under the support of the NOAA CPPA Program, a project led by Zong-Liang Yang (with Co-Is: Guo-Yue Niu, Fei Chen, David Gochis, and Kenneth Mitchell) created the Noah with multiple-physics (Noah-MP) to augment the modeling capabilities of Noah by enhancing its representation of dynamic vegetation, radiation, hydrology, and snow physics, and establishing the multi-physics modeling framework (Niu et al. 2011, Yang et al. 2011). Noah-MP was implemented in WRF in 2014 (Barlage et al. 2015) and its various community versions have been maintained at the NCAR Github: https://github.com/NCAR/noahmp.

1.2 Noah-MP Version 5.0

The previous Noah-MP code (prior to Version 5.0) was written in a single Fortran source file containing all model physics with complex coding and data structures that are not in a unified/consistent format. As such, the code is hard to modularize, and difficult for users and developers to extract existing physics modules for testing or use. These issues limit the application of Noah-MP. Furthermore, a lengthy code is error prone and challenging to read, modify, and debug. These issues would limit the application of Noah-MP. The modernized/refactored Noah-MP (Version 5.0) resolves these issues by adopting advanced Fortran data and code structures and modularization of physical processes. Specifically, the current version (v5.0) of Noah-MP is a newly refactored code to improve the modularity, interoperability, applicability, and understanding of process-level physics and variables, and to ease its use, further developments, and contribution by the community at large.

In this refactored version, the model code has been re-organized into different individual Fortran files that include one module for each single physical process/scheme. All the one-dimensional column Noah-MP variables (state, flux and parameters) are grouped into a high level 'noahmp_type' user defined data type (Figure 1). The sub-types of 'noahmp_type' are 'forcing_type', 'config_type', 'water_type', 'energy_type', and 'biochem_type'. Each sub-type under 'noahmp_type' is further divided into 'state_type', 'flux_type', and 'parameter_type'. The data/variable initialization and declaration structures have also been optimized along with the new Noah-MP data types. The original names of subroutines and variables have been changed to an easy understandable form. The main Noah-MP module file is renamed from "module_sf_noahmplsm.F90" to "NoahmpMainMod.F90" and the main column Noah-MP subroutine 'SFLX' is renamed as "NoahmpMain". The one-dimensional column Noah-MP source code, which was previously written in a single Fortran90 file, is separated into process-level

Fortran modules. For instance, the energy related processes are called from "EnergyMainMod.F90" using the subroutine "EnergyMain". A similar approach is adopted to represent other processes as well. See Figures 2-5 for the detailed calling trees of the refactored Noah-MP model processes. Each main process is further divided into process level modules and will be discussed in detail later in this document. Overall, the modernized community Noah-MP model will allow a more efficient process for future earth system model developments and applications.

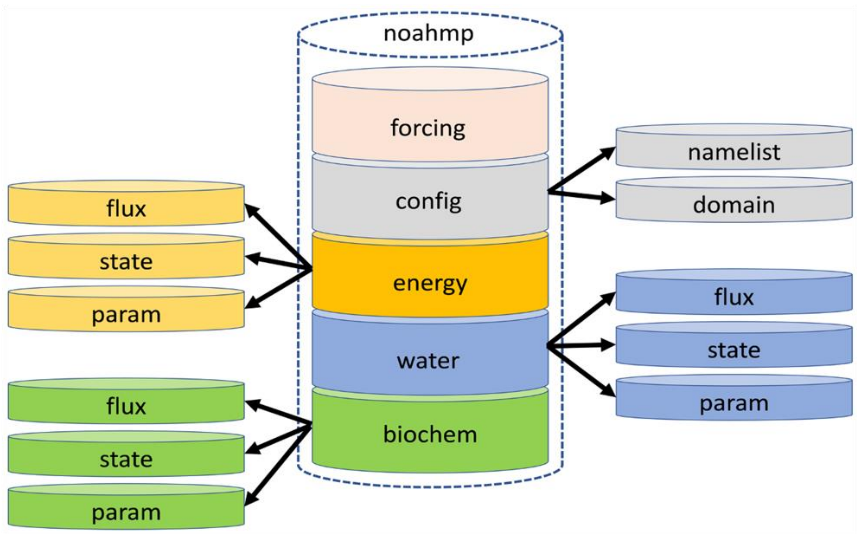


Figure 1. The structure of "noahmp" data types used in the Noah-MP model.

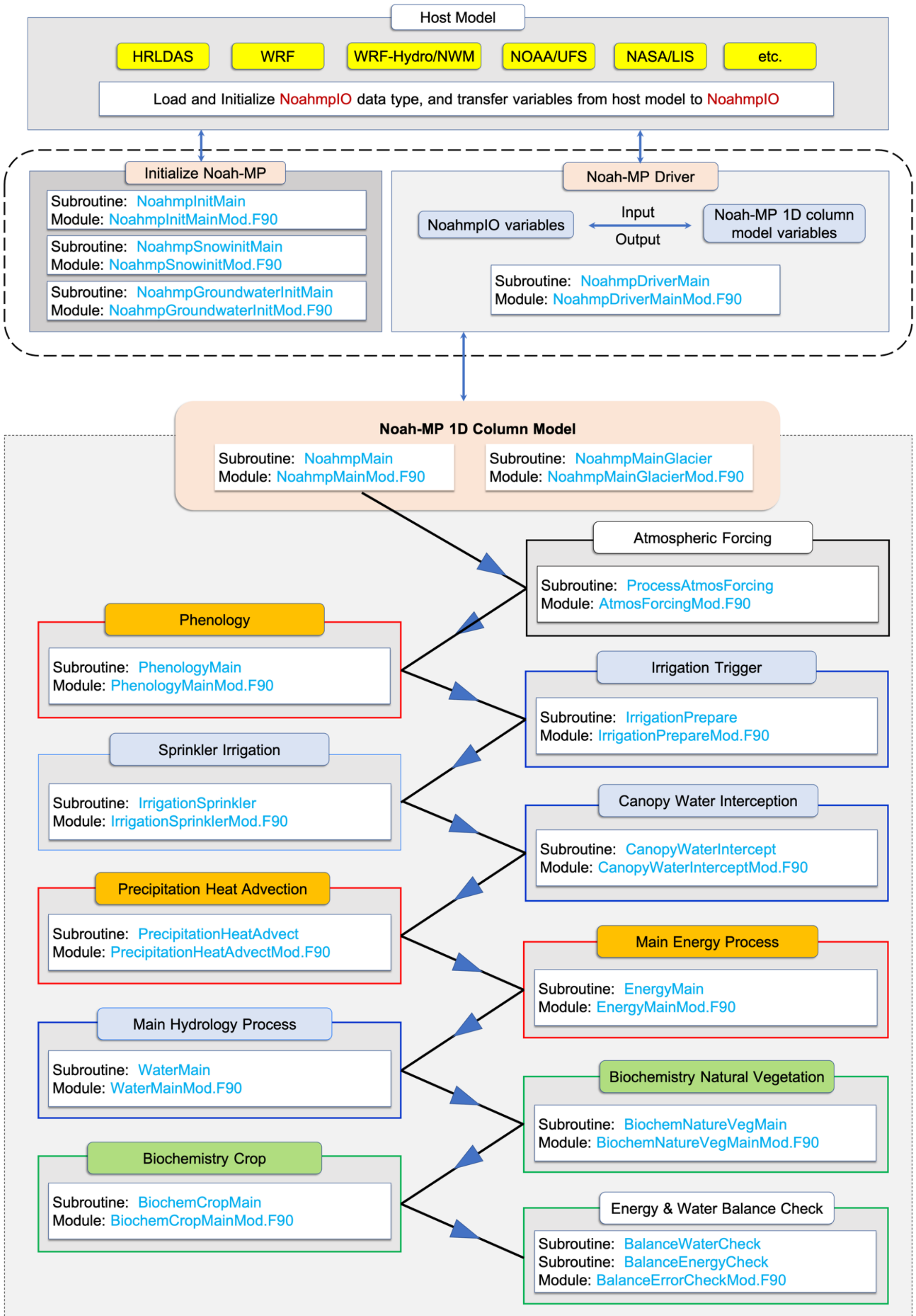

Host Model
HRLDAS
WRF
WRF-Hydro/NWM
NOAA/UFS
NASA/LIS
etc.
Load and Initialize NoahmpIO data type, and transfer variables from host model to NoahmpIO
Initialize Noah-MP
Subroutine: NoahmpInitMain
Module: NoahmpInitMainMod.F90
Subroutine: NoahmpSnowinitMain
Module: NoahmpSnowinitMod.F90
Subroutine: NoahmpGroundwaterInitMain
Module: NoahmpGroundwaterInitMod.F90
Noah-MP Driver
NoahmpIO variables
Input
Output
Noah-MP 1D column model variables
Subroutine: NoahmpDriverMain
Module: NoahmpDriverMainMod.F90
Noah-MP 1D Column Model
Subroutine: NoahmpMain
Module: NoahmpMainMod.F90
Subroutine: NoahmpMainGlacier
Module: NoahmpMainGlacierMod.F90
Atmospheric Forcing
Subroutine: ProcessAtmosForcing
Module: AtmosForcingMod.F90
Phenology
Subroutine: PhenologyMain
Module: PhenologyMainMod.F90
Irrigation Trigger
Subroutine: IrrigationPrepare
Module: IrrigationPrepareMod.F90
Sprinkler Irrigation
Subroutine: IrrigationSprinkler
Module: IrrigationSprinklerMod.F90
Canopy Water Interception
Subroutine: CanopyWaterIntercept
Module: CanopyWaterInterceptMod.F90
Precipitation Heat Advection
Subroutine: PrecipitationHeatAdvect
Module: PrecipitationHeatAdvectMod.F90
Main Energy Process
Subroutine: EnergyMain
Module: EnergyMainMod.F90
Main Hydrology Process
Subroutine: WaterMain
Module: WaterMainMod.F90
Biochemistry Natural Vegetation
Subroutine: BiochemNatureVegMain
Module: BiochemNatureVegMainMod.F90
Biochemistry Crop
Subroutine: BiochemCropMain
Module: BiochemCropMainMod.F90
Energy & Water Balance Check
Subroutine: BalanceWaterCheck
Subroutine: BalanceEnergyCheck
Module: BalanceErrorCheckMod.F90

Figure 2. The Noah-MP physical process calling tree. Blue boxes indicate hydrological processes, orange boxes indicate energy processes, and green boxes indicate biochemistry processes in the Noah-MP model. Direction of arrows indicates process calling sequence and information flow. Note that the 1-D glacier column model has similar structures as the main non-glacier column model, except that the vegetation-related processes are removed and soil is replaced by glacier ice.

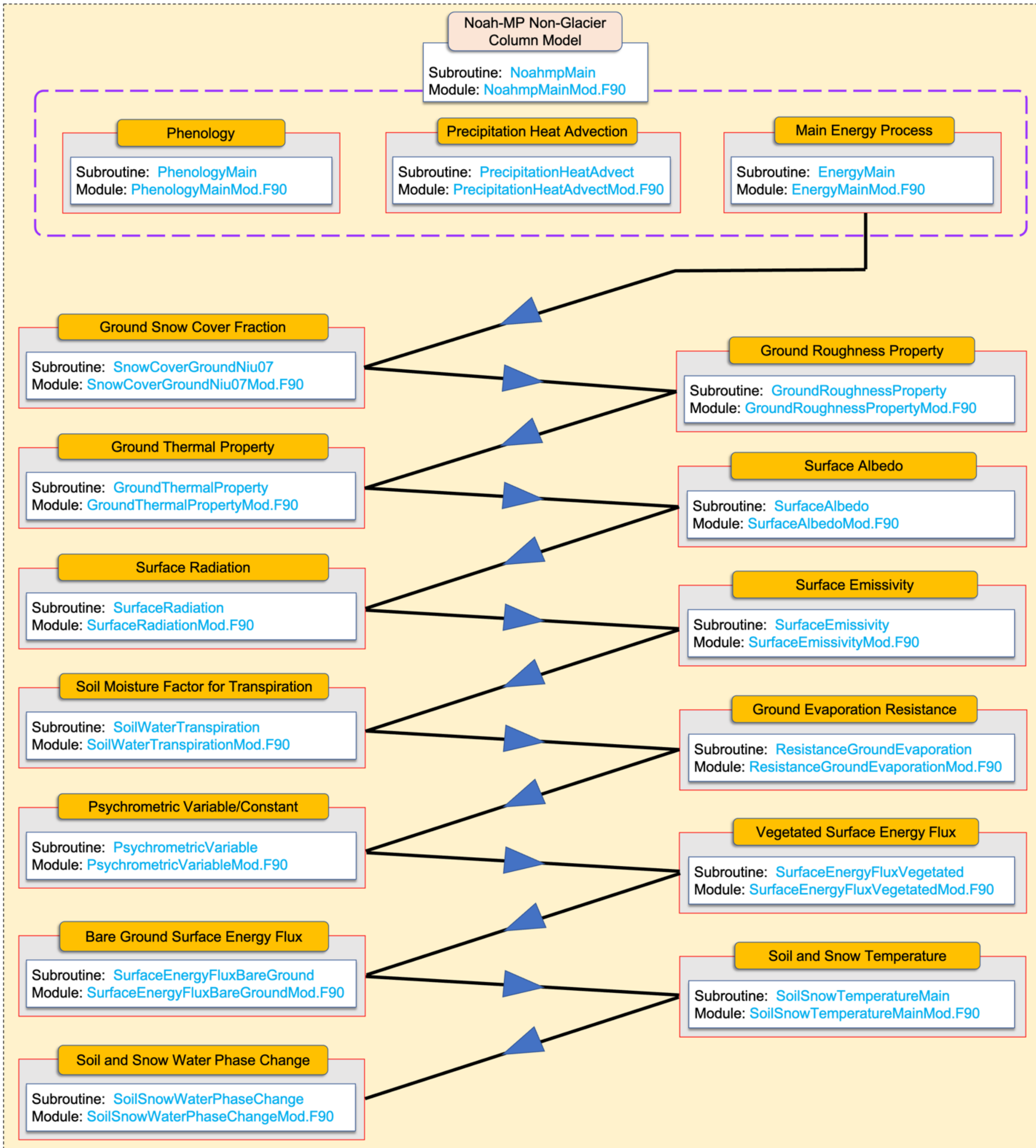
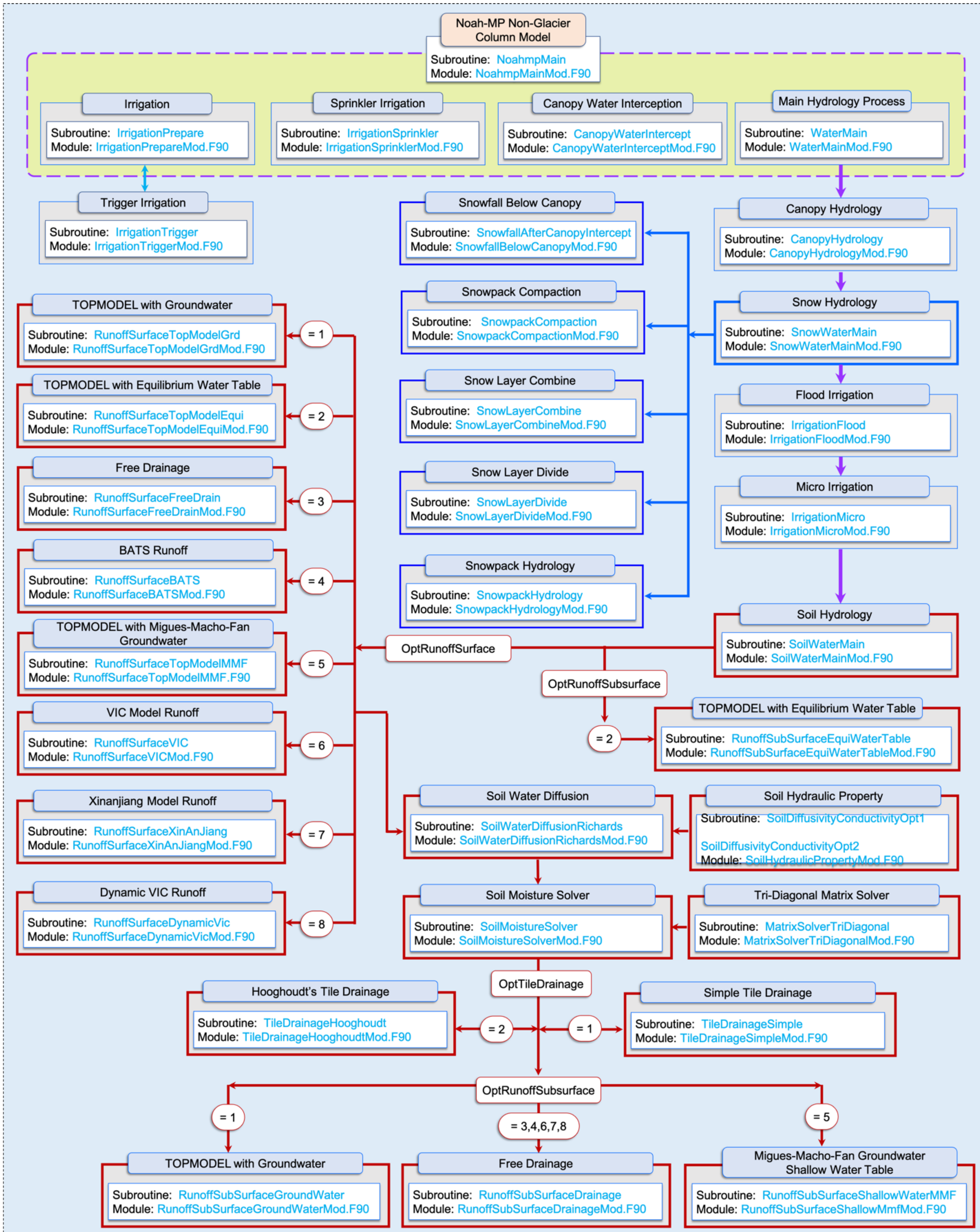


Figure 3. The Noah-MP energy process calling tree for the non-glacier column model. Note that the glacier column model has similar structures except that the vegetation-related processes are removed and soil is replaced by glacier ice.



Noah-MP Non-Glacier Column Model
Subroutine: NoahmpMain
Module: NoahmpMainMod.F90
Irrigation
Subroutine: IrrigationPrepare
Module: IrrigationPrepareMod.F90
Sprinkler Irrigation
Subroutine: IrrigationSprinkler
Module: IrrigationSprinklerMod.F90
Canopy Water Interception
Subroutine: CanopyWaterIntercept
Module: CanopyWaterInterceptMod.F90
Main Hydrology Process
Subroutine: WaterMain
Module: WaterMainMod.F90
Trigger Irrigation
Subroutine: IrrigationTrigger
Module: IrrigationTriggerMod.F90
Snowfall Below Canopy
Subroutine: SnowfallAfterCanopyIntercept
Module: SnowfallBelowCanopyMod.F90
Canopy Hydrology
Subroutine: CanopyHydrology
Module: CanopyHydrologyMod.F90
TOPMODEL with Groundwater
Subroutine: RunoffSurfaceTopModelGrd
Module: RunoffSurfaceTopModelGrdMod.F90
= 1
Snowpack Compaction
Subroutine: SnowpackCompaction
Module: SnowpackCompactionMod.F90
Snow Hydrology
Subroutine: SnowWaterMain
Module: SnowWaterMainMod.F90
TOPMODEL with Equilibrium Water Table
Subroutine: RunoffSurfaceTopModelEqui
Module: RunoffSurfaceTopModelEquiMod.F90
= 2
Snow Layer Combine
Subroutine: SnowLayerCombine
Module: SnowLayerCombineMod.F90
Flood Irrigation
Subroutine: IrrigationFlood
Module: IrrigationFloodMod.F90
Free Drainage
Subroutine: RunoffSurfaceFreeDrain
Module: RunoffSurfaceFreeDrainMod.F90
= 3
Snow Layer Divide
Subroutine: SnowLayerDivide
Module: SnowLayerDivideMod.F90
Micro Irrigation
Subroutine: IrrigationMicro
Module: IrrigationMicroMod.F90
BATS Runoff
Subroutine: RunoffSurfaceBATS
Module: RunoffSurfaceBATSMod.F90
= 4
Snowpack Hydrology
Subroutine: SnowpackHydrology
Module: SnowpackHydrologyMod.F90
TOPMODEL with Migues-Macho-Fan Groundwater
Subroutine: RunoffSurfaceTopModelMMF
Module: RunoffSurfaceTopModelMMF.F90
= 5
OptRunoffSurface
Soil Hydrology
Subroutine: SoilWaterMain
Module: SoilWaterMainMod.F90
OptRunoffSubsurface
VIC Model Runoff
Subroutine: RunoffSurfaceVIC
Module: RunoffSurfaceVICMod.F90
= 6
= 2
TOPMODEL with Equilibrium Water Table
Subroutine: RunoffSubSurfaceEquiWaterTable
Module: RunoffSubSurfaceEquiWaterTableMod.F90
Xinanjiang Model Runoff
Subroutine: RunoffSurfaceXinAnJiang
Module: RunoffSurfaceXinAnJiangMod.F90
= 7
Soil Water Diffusion
Subroutine: SoilWaterDiffusionRichards
Module: SoilWaterDiffusionRichardsMod.F90
Soil Hydraulic Property
Subroutine: SoilDiffusivityConductivityOpt1
SoilDiffusivityConductivityOpt2
Module: SoilHydraulicPropertyMod.F90
Dynamic VIC Runoff
Subroutine: RunoffSurfaceDynamicVic
Module: RunoffSurfaceDynamicVicMod.F90
= 8
Soil Moisture Solver
Subroutine: SoilMoistureSolver
Module: SoilMoistureSolverMod.F90
Tri-Diagonal Matrix Solver
Subroutine: MatrixSolverTriDiagonal
Module: MatrixSolverTriDiagonalMod.F90
OptTileDrainage
Hooghoudt's Tile Drainage
Subroutine: TileDrainageHooghoudt
Module: TileDrainageHooghoudtMod.F90
= 2
= 1
Simple Tile Drainage
Subroutine: TileDrainageSimple
Module: TileDrainageSimpleMod.F90
OptRunoffSubsurface
= 1
= 3,4,6,7,8
= 5
TOPMODEL with Groundwater
Subroutine: RunoffSubSurfaceGroundWater
Module: RunoffSubSurfaceGroundWaterMod.F90
Free Drainage
Subroutine: RunoffSubSurfaceDrainage
Module: RunoffSubSurfaceDrainageMod.F90
Migues-Macho-Fan Groundwater Shallow Water Table
Subroutine: RunoffSubSurfaceShallowWaterMMF
Module: RunoffSubSurfaceShallowMmfMod.F90

Figure 4. The Noah-MP water process calling tree for the non-glacier column model. Note that the glacier column model has similar structures except that it only includes the snowpack processes and soil is replaced by glacier ice.

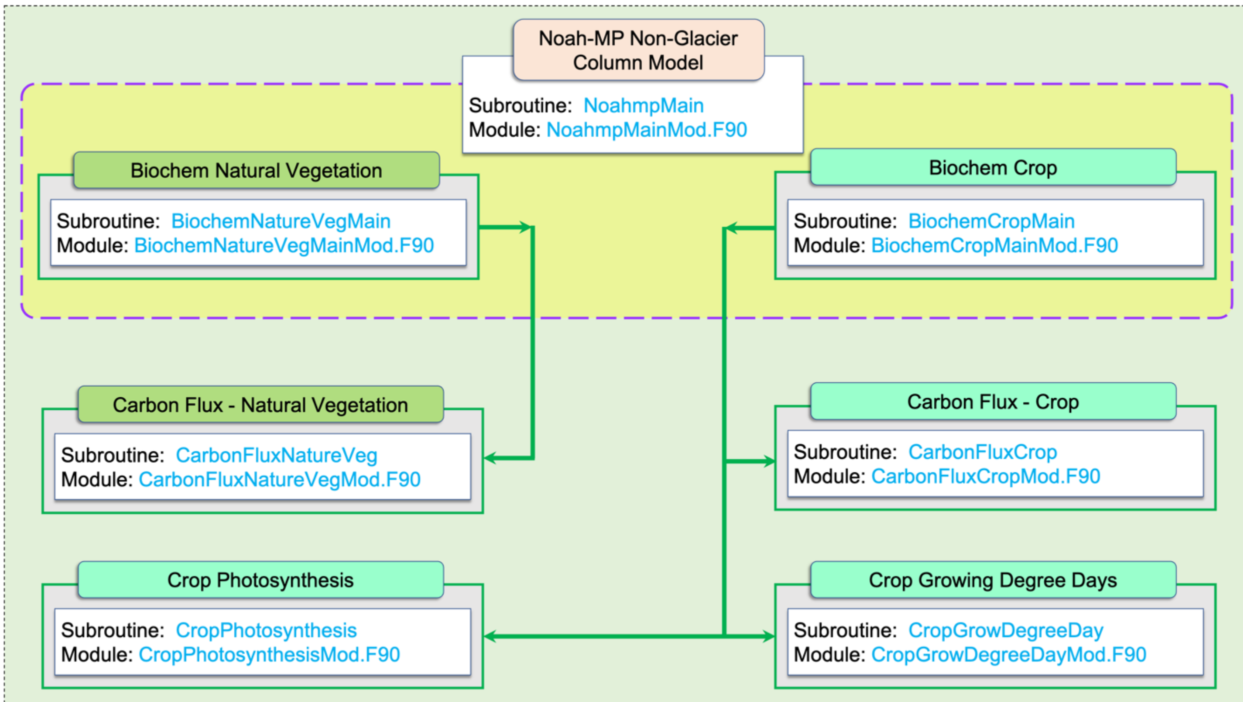


Figure 5. The Noah-MP biochemistry process calling tree. Note that currently Noah-MP only includes carbon processes.

2. Noah-MP Namelist Options for Physical Schemes

Noah-MP Physics	Option	Notes (* indicates the default option)
OptDynamicVeg options for dynamic vegetation	1	off (use table LeafAreaIndex; use VegFrac = VegFracGreen from input) (Niu et al., 2011; Yang et al., 2011)
	2	on (together with OptStomataResistance = 1) (Dickinson et al., 1998; Niu and Yang, 2003)
	3	off (use table LeafAreaIndex; calculate VegFrac)
	4*	off (use table LeafAreaIndex; use maximum vegetation fraction)
	5	on (use maximum vegetation fraction)
	6	on (use VegFrac = VegFracGreen from input)
	7	off (use input LeafAreaIndex; use VegFrac = VegFracGreen from input)
	8	off (use input LeafAreaIndex; calculate VegFrac)
	9	off (use input LeafAreaIndex; use maximum vegetation fraction)
OptRainSnowPartition options for partitioning precipitation into rainfall & snowfall	1*	Jordan (1991) scheme
	2	BATS: when TemperatureAirRefHeight < freezing point+2.2 (Yang and Dickinson, 1996)
	3	TemperatureAirRefHeight < freezing point (Niu et al., 2011)
	4	Use WRF microphysics output (Barlage et al., 2015)
	5	Use wetbulb temperature (Wang et al., 2019)
OptSoilWaterTranspiration options for soil moisture factor for stomatal resistance & ET	1*	Noah (soil moisture) (Ek et al., 2003)
	2	CLM (matric potential) (Oleson et al., 2004)
	3	SSiB (matric potential) (Xue et al., 1991)
OptGroundResistanceEvap options for ground resistent to evaporation/sublimation	1*	Sakaguchi and Zeng (2009) scheme
	2	Sellers (1992) scheme
	3	adjusted Sellers (1992) for wet soil
	4	Sakaguchi and Zeng (2009) for non-snow; rsurf = rsurf_snow for snow (set in NoahmpTable.TBL)
OptSurfaceDrag options for surface layer drag/exchange coefficient	1*	Monin-Obukhov (M-O) Similarity Theory (Brutsaert, 1982)
	2	original Noah (Chen et al. 1997)
OptStomataResistance options for canopy stomatal resistance	1*	Ball-Berry scheme (Ball et al., 1987; Bonan, 1996)
	2	Jarvis scheme (Jarvis, 1976)
OptSnowAlbedo options for ground snow surface albedo	1*	BATS snow albedo (Dickinson et al., 1993)
	2	CLASS snow albedo (Verseghy, 1991)

OptCanopyRadiationTransfer options for canopy radiation transfer	1	modified two-stream (gap=F(solar angle,3D structure, etc)<1-VegFrac) (Niu and Yang, 2004)
	2	two-stream applied to grid-cell (gap=0) (Niu et al., 2011)
	3*	two-stream applied to vegetated fraction (gap=1-VegFrac) (Dickinson, 1983; Sellers, 1985)
OptSnowSoilTempTime options for snow/soil temperature time scheme (only layer 1)	1*	semi-implicit; flux top boundary condition (Niu et al., 2011)
	2	full implicit (original Noah); temperature top boundary condition (Ek et al., 2003)
	3	same as 1, but snow cover for skin temperature calculation (Niu et al., 2011)
OptSnowThermConduct options for snow thermal conductivity	1*	Stieglitz scheme (Yen,1965)
	2	Anderson (1976) scheme
	3	Constant (Niu et al., 2011)
	4	Verseghy (1991) scheme
	5	Douvill scheme (Yen, 1981)
OptSoilTemperatureBottom options for lower boundary condition of soil temperature	1	zero heat flux from bottom (DepthSoilTempBottom & TemperatureSoilBottom not used) (Niu et al., 2011)
	2*	TemperatureSoilBottom at DepthSoilTempBottom (8m) read from a file (original Noah) (Ek et al., 2003)
OptSoilSupercoolWater options for soil supercooled liquid water	1*	No iteration (Niu and Yang, 2006)
	2	Koren's iteration (Koren et al., 1999)
OptRunoffSurface options for surface runoff	1	TOPMODEL with groundwater (Niu et al., 2007)
	2	TOPMODEL with an equilibrium water table (Niu et al., 2005)
	3*	Original Noah free drainage (Schaake et al., 1996)
	4	BATS surface and subsurface runoff (Yang and Dickinson, 1996)
	5	Miguez-Macho&Fan groundwater scheme (Fan et al., 2007; Miguez-Macho et al. 2007)
	6	Variable Infiltration Capacity Model surface runoff scheme (Liang et al., 1994)
	7	Xinanjiang Infiltration and surface runoff scheme (Jayawardena and Zhou, 2000)
	8	Dynamic VIC surface runoff scheme (Liang and Xie, 2003)
OptRunoffSubsurface options for drainage & subsurface runoff	1~8	similar to runoff option, separated from original Noah-MP runoff option, currently tested & recommended the same option# as surface runoff (default)
OptSoilPermeabilityFrozen options for frozen soil permeability	1*	linear effects, more permeable (Niu and Yang, 2006)
	2	nonlinear effects, less permeable (Koren et al., 1999)

OptDynVicInfiltration options for infiltration in dynamic VIC runoff scheme	1*	Philip scheme (Liang and Xie, 2003)
	2	Green-Ampt scheme (Liang and Xie, 2003)
	3	Smith-Parlange scheme (Liang and Xie, 2003)
OptTileDrainage options for tile drainage currently only tested & calibrated to work with runoff option=3	0*	No tile drainage
	1	on (simple scheme) (Valayamkunnath et al., 2022)
	2	on (Hooghoudt's scheme) (Valayamkunnath et al., 2022)
OptIrrigation options for irrigation	0*	No irrigation
	1	Irrigation on (Valayamkunnath et al., 2021)
	2	irrigation trigger based on crop season planting and harvesting dates (Valayamkunnath et al., 2021)
	3	irrigation trigger based on LeafAreaIndex threshold (Valayamkunnath et al., 2021)
OptIrrigationMethod options for irrigation method, only works when OptIrrigation > 0	0*	method based on geo_em fractions
	1	sprinkler method (Valayamkunnath et al., 2021)
	2	micro/drip irrigation (Valayamkunnath et al., 2021)
	3	surface flooding (Valayamkunnath et al., 2021)
OptCropModel options for crop model	0*	No crop model
	1	Liu et al. (2016) crop scheme
OptSoilProperty options for defining soil properties	1*	use input dominant soil texture
	2	use input soil texture that varies with depth
	3	use soil composition (sand, clay, orgm) and pedotransfer function
	4	use input soil properties
OptPedotransfer options for pedotransfer functions, only works when OptSoilProperty=3	1*	Saxton and Rawls (2006) scheme
OptGlacierTreatment options for glacier treatment	1*	include phase change of glacier ice
	2	Glacier ice treatment more like original Noah

3. Noah-MP Physical Processes

A detailed description of Noah-MP physics parametrization is presented in this chapter. At the interface between land surface and atmosphere, each Noah-MP land grid is divided into two fractions (or tiles); namely, vegetated and bare soil fractions. Noah-MP independently estimates biogeophysical and biogeochemical processes for vegetated and bare soil fractions at each model grid. Special attention is devoted to resolving energy and water balance components over glacier points. Noah-MP calculates canopy water interception and canopy evaporation before partitioning precipitation into surface runoff and infiltration. Multiple saturation-excess runoff and infiltration-excess runoff physics are implemented into Noah-MP for surface runoff and infiltration estimation. Unsaturated water flow through the soil column and soil moisture are simulated using one-dimensional Richard's equation. An optional two-dimensional groundwater physics is also implemented into Noah-MP. However, the default configuration of Noah-MP considers bottom-open soil column and flow through the column bottom as groundwater recharge. Physics are incorporated into Noah-MP to calculate bare soil evaporation and plant transpiration, which are represented in the model as the functions of soil moisture and other properties considered as parameters.

Noah-MP resolves energy balance components separately for vegetated and bare soil fractions. Vegetation phenology, estimated by the phenology module of Noah-MP, is used to calculate vegetation fraction and leaf area index in each model grid. The energy module of Noah-MP calculates snow cover fraction, surface roughness, ground thermal properties, albedo, surface radiation, surface emissivity, etc. before solving energy flux balance for vegetated and bare soil fractions of a land model grid. The energy module also solves snow and soil thermodynamics and soil and snow water phase changes at each model time step. In Noah-MP the weighted mean surface states and fluxes are estimated at each model grid cell based on the vegetated and bare soil fractions and transferred into the atmospheric model and output files.

Noah-MP considers separate biogeochemistry (currently only carbon) modules for natural vegetation and agriculture crops. Biogeochemistry modules estimate vegetation growth dynamics and crop yields by allocating carbon to various vegetation carbon pools (leaf, stem, wood and root) and soil carbon pools (fast and slow). The Net Primary Productivity is calculated after allocating carbon loss to the environment through respiration.

Different biogeophysical and biogeochemical processes represented in the Noah-MP land surface model include (see also Figures 6-8):

1. Surface characterization including land type heterogeneity and ecosystem structure
2. Absorption, reflection, and transmittance of solar radiation
3. Absorption and emission of longwave radiation
4. Momentum, sensible heat (ground and canopy), and latent heat (ground evaporation, canopy evaporation, transpiration) fluxes
5. Heat transfer in soil and snow, including phase change
6. Canopy hydrology (interception, throughfall, and drip)
7. Soil hydrology (surface runoff, infiltration, redistribution of water within the column, sub-surface drainage, groundwater)
8. Snow hydrology (snow accumulation and melt, compaction, water transfer between snow layers)
9. Stomatal physiology, photosynthesis, plant respiration and mortality
10. Plant/soil hydraulics
11. Glacier processes
12. Vegetation phenology and carbon allocation
13. Soil and litter carbon decomposition
14. Crop dynamics, irrigation, and tile drainage

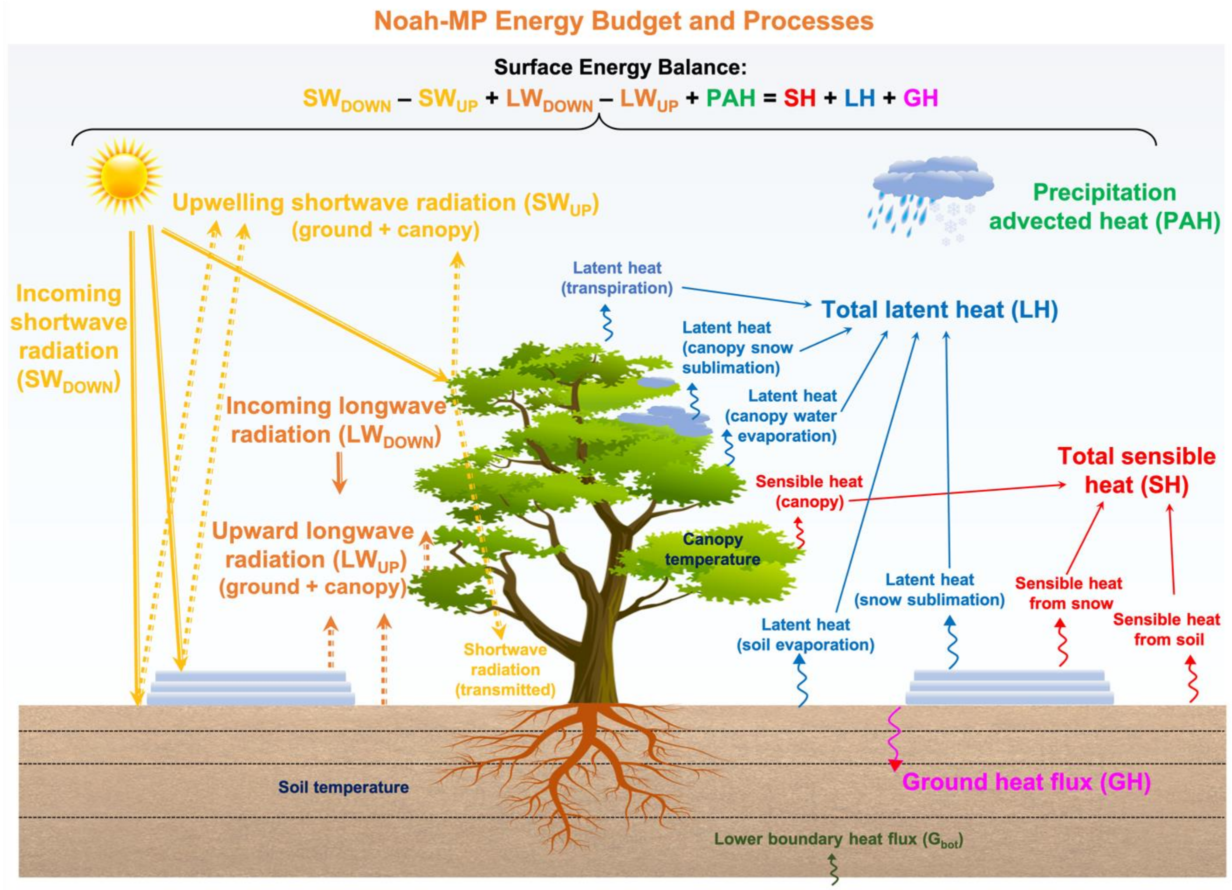


Figure 6. Schematic diagram of energy budget and processes represented in Noah-MP.

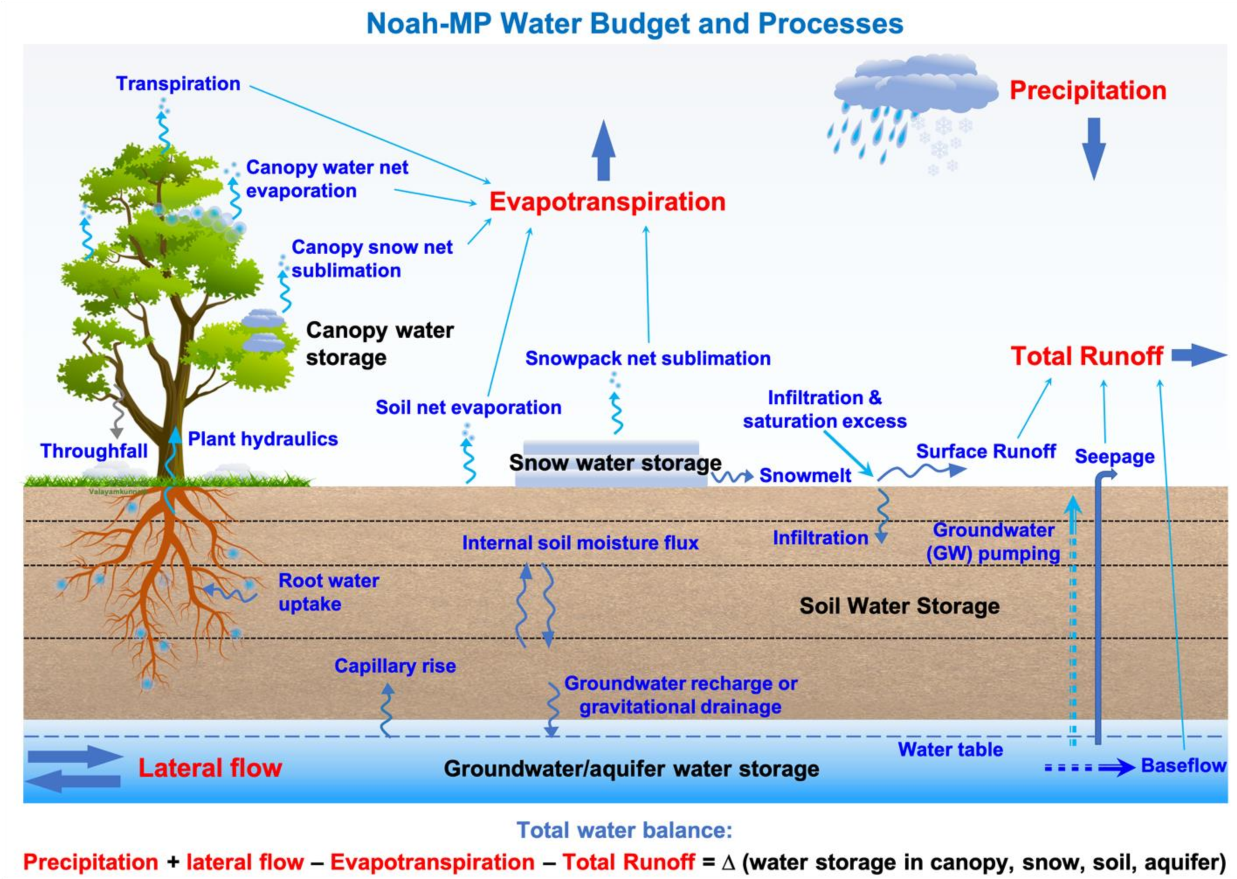


Figure 7. Schematic diagram of water budget and processes represented in Noah-MP.

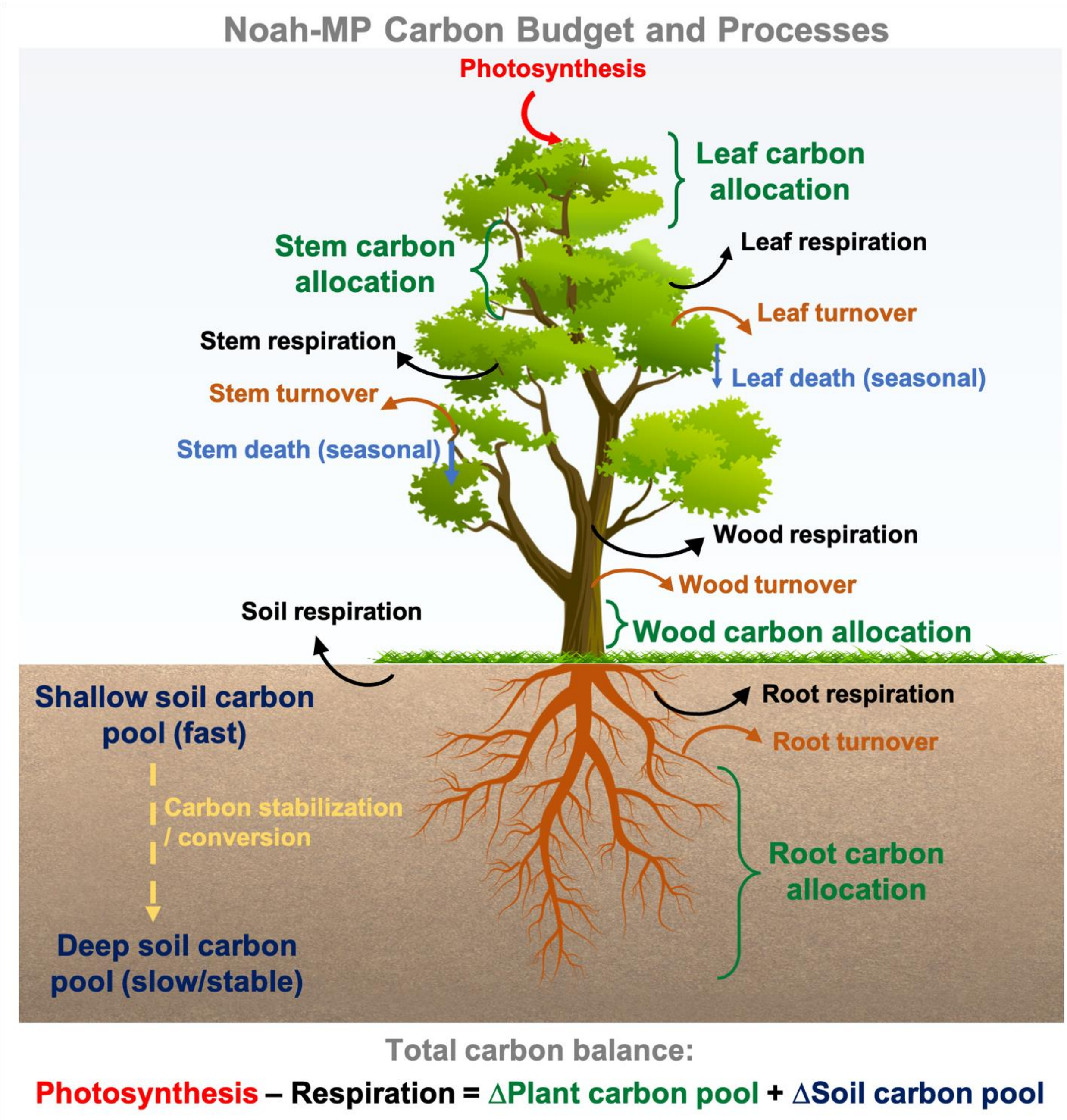


Figure 8. Schematic diagram of carbon budget and processes represented in Noah-MP.

3.1 Atmospheric Forcing Processing

Description:

The purpose of the Noah-MP atmospheric forcing module is to process the input atmospheric forcing variables (surface temperature, precipitation, surface pressure, wind, specific humidity, downward shortwave and longwave radiation) for use in Noah-MP model physics.

Relevant code modules:

Module: AtmosForcingMod.F90

Subroutines: ProcessAtmosForcing

Relevant Noah-MP namelist options:

Physical Processes	Options	Notes (* indicates the default option)
OptRainSnowPartition	1*	Jordan (1991) scheme
	2	BATS: when TemperatureAirRefHeight < freezing point+2.2
options for partitioning	3	TemperatureAirRefHeight < freezing point
precipitation into rainfall	4	Use WRF microphysics output
& snowfall	5	Use wetbulb temperature (Wang et al., 2019)

Physics:

Several key surface variables are computed based on the input forcing, including surface potential temperature, vapor pressure, air density, downward direct and diffuse shortwave radiation for visible and near-infrared (NIR) bands, rainfall, and snowfall.

1. Surface potential temperature (θ_{sfc}, [K]):

$$\theta_{sfc} = T_{sfc}\left(\frac{P_{ref}}{P_{sfc}}\right)^{\frac{R_d}{C_p}} \quad \text{(Eq. 3.1.1)}$$

where T_{sfc} [K] is the surface air temperature (forcing input), P_{ref} [Pa] is the air pressure at reference level, P_{sfc} [Pa] is the surface air pressure (forcing input), R_d [J/kg/K] is the gas constant for dry air, and C_p [J/kg/K] is the specific heat capacity of dry air.

2. Surface air vapor pressure (e_{air}, [Pa]):

$$e_{air} = q_{sfc} \times \frac{P_{sfc}}{(0.622+0.378 \times q_{sfc})} \quad \text{(Eq. 3.1.2)}$$

where q_{sfc} [kg/kg] is the specific humidity (forcing input).

3. Dry air density (ρ_{air}, [kg/m^3]):

$$\rho_{air} = \frac{(P_{sfc} - 0.378 \times e_{air})}{R_d \times T_{sfc}} \quad \text{(Eq. 3.1.3)}$$

4. Downward solar radiation:

The diffuse and direct radiation fractions are assumed to be 30% and 70%, respectively to partition the total downward solar radiation. There are 2 solar wavelength bands (visible and NIR) accounted for in Noah-MP, which are assumed to be 50% and 50%, respectively. These fractions for partitioning total downward solar radiation are currently hard-coded, but will be moved to the Noah-MP parameter table for easier tuning in the future.

5. Fractional area receiving precipitation:

The total precipitation forcing includes precipitation from deep convection, shallow convection, and non-convection. The total precipitation from input forcing is further partitioned to two types: convective and large-scale (non-convective) precipitation. If OptRainSnowPartition = 4, Noah-MP uses the explicit convective and non-convective precipitation from atmospheric forcing; otherwise, Noah-MP assumes 10% of total precipitation to be convective precipitation ($P_{r,conv}$, [mm/s]) and 90% to be large-scale precipitation ($P_{r,nonc}$, [mm/s]).

Then, the fractional area that receives precipitation (F_{prcp}) is computed as follows (Niu et al. 2005):

$$F_{prcp} = \frac{(P_{r,conv} + P_{r,nonc})}{(10 \times P_{r,conv} + P_{r,nonc})} \quad \text{(Eq. 3.1.4)}$$

The F_{prcp} is further used in the canopy interception process.

6. Snowfall and rainfall partitioned from total precipitation:

There are five different options to partition total precipitation into rainfall and snowfall based on the frozen precipitation fraction ($F_{snowfall}$)

If OptRainSnowPartition == 1 (Jordan, 1991 scheme):

$$F_{snowfall} = \begin{cases} 0.0, & T_{sfc} > T_{frz} + 2.5 \\ 0.6, & T_{frz} + 2.0 < T_{sfc} \le T_{frz} + 2.5 \\ 1.0 - (-54.632 + 0.2 \times T_{sfc}), & T_{frz} - 0.5 < T_{sfc} \le T_{frz} + 2 \\ 1.0, & T_{sfc} \le T_{frz} - 0.5 \end{cases} \quad \text{(Eq. 3.1.5)}$$

where T_{frz} is the freezing point temperature (273.15 K).

If OptRainSnowPartition == 2 (BATS scheme):

$$F_{snowfall} = \begin{cases} 0.0, & T_{sfc} \ge T_{frz} + 2.2 \\ 1.0, & T_{sfc} < T_{frz} + 2.2 \end{cases} \quad \text{(Eq. 3.1.6)}$$

If OptRainSnowPartition == 3 (Simple scheme):

$$F_{snowfall} = \begin{cases} 0.0, & T_{sfc} \ge T_{frz} \\ 1.0, & T_{sfc} < T_{frz} \end{cases} \quad \text{(Eq. 3.1.7)}$$

If OptRainSnowPartition == 4 (Use WRF microphysics):

The total frozen precipitation ($P_{r,frz}$ = snow+graupel+hail from input forcing) is computed first and then

$$F_{snowfall} = \begin{cases} \frac{P_{r,frz}}{P_{r,nonc}}, & \text{if } P_{r,nonc} > 0 \text{ and } P_{r,frz} > 0 \\ 0.0, & \text{if } P_{r,nonc} \le 0 \text{ or } P_{r,frz} \le 0 \end{cases} \quad \text{(Eq. 3.1.8)}$$

If OptRainSnowPartition == 5 (Wet-bulb scheme, Wang et al., 2019):

$$F_{snowfall} = \frac{1}{(1 + a \times e^{b \times (T_{wetb} + c)})} \quad \text{(Eq. 3.1.9)}$$

where a (=6.99×10^{-5}), b (=2.0, [1/K]), and c (=3.97, [K]) are parameterization parameters, and T_{wetb} is the wet-bulb temperature computed iteratively from the following equation:

$$T_{wetb} = T_{sfc} - \frac{(e_{sat}(T_{wetb}) - e_{air})}{\gamma} \quad \text{(Eq. 3.1.10)}$$

where γ [Pa/K] is the psychrometric constant and e_{sat} is the saturated water vapor pressure at T_{wetb}.

Finally, snowfall and rainfall are calculated from as shown below:

$$P_{snow} = (P_{r,conv} + P_{r,nonc}) \times F_{snowfall} \quad \text{(Eq. 3.1.11)}$$

$$P_{rain} = (P_{r,conv} + P_{r,nonc}) \times (1 - F_{snowfall}) \quad \text{(Eq. 3.1.12)}$$

7. Fresh snowfall density ($\rho_{snowfall}$, [kg/m^3]):

The snowfall density is also computed based on Hedstrom and Pomeory (1998) for use in canopy-snow interception and canopy hydrology.

$$\rho_{snowfall} = \min(\rho_{snowfall,max}\ , 67.92 + 51.25 \times e^{\frac{(T_{sfc}-T_{frz})}{2.59}}) \qquad \text{(Eq. 3.1.13)}$$

where $\rho_{snowfall,max}$ is the maximum fresh snowfall density (currently set to 120.0 kg/m^3) in the Noah-MP parameter table.

If OptRainSnowPartition == 4, the mean snowfall density is further updated by accounting for graupel and hail as follows:

$$\rho_{snowfall} = \rho_{snowfall} \times \frac{P_{snow}}{P_{r,frz}} + \rho_{graupel} \times \frac{P_{graupel}}{P_{r,frz}} + \rho_{hail} \times \frac{P_{hail}}{P_{r,frz}} \qquad \text{(Eq. 3.1.14)}$$

where $P_{graupel}$ [mm/s] and P_{hail} [mm/s] are the graupel and hail rates, respectively. $\rho_{graupel}$ [kg/m^3] and ρ_{hail} [kg/m^3] are the densities of graupel and hail, respectively.

3.2 Initialization of Key State Variables

Description:

These two Noah-MP initialization modules are to initialize key state variables, including soil and snow layer thickness, and root-zone soil temperature, and initial water storage.

Relevant code modules:

Module: GeneralInitMod.F90

Subroutines: GeneralInit

Module: BalanceErrorCheckMod.F90

Subroutines: BalanceWaterInit

Physics:

1. **Initialize snow and soil *ith* layer thickness** (D_{zsnso}, [m]):

$$D_{zsnso}(i) = \begin{cases} -Z_{snso}(i)\ , & i = -N_{sno} + 1 \\ Z_{snso}(i-1) - Z_{snso}(i)\ , & i = -N_{sno} + 2, \dots, N_{soil} \end{cases} \qquad \text{(Eq. 3.2.1)}$$

where $Z_{snso}(i)$ [m] is the depth (negative) of the i^{th} layer bottom of snow and soil from snow surface, N_{sno} is the actual number of snow layers, and N_{soil} is the total number of soil layers.

2. Initialize root-zone mean temperature (T_{root}, [K]) as a thickness-weighted soil temperature (T_{soil}, [K]):

$$T_{root} = \sum_{i=1}^{N_{root}} (T_{soil} \times \frac{D_{zsnso}(i)}{-Z_{soil}(i=N_{root})}) \quad \text{(Eq. 3.2.2)}$$

where $Z_{soil}(i)$ [m] is the depth (negative) of the soil layer bottom and N_{root} is the number of soil layers in the root zone.

3. Initialize water storage before any Noah-MP processes (W_{beg}, [mm]):

$$W_{beg} = W_{ice,can} + W_{liq,can} + W_{snow} + W_{aquifer} + W_{soil} \quad \text{(Eq. 3.2.3)}$$

$$W_{soil} = \sum_{i=1}^{N_{soil}} [\theta_{soil}(i) \times D_{zsnso}(i) \times 1000] \quad \text{(Eq. 3.2.4)}$$

where $W_{ice,can}$ [mm] is the canopy ice, $W_{liq,can}$ [mm] is the canopy liquid water, W_{snow} [mm] is the snow water equivalent (SWE), $W_{aquifer}$ [mm] is the aquifer water storage, W_{soil} [mm] is the soil column water storage, and θ_{soil} [m^3/m^3] is the soil moisture for each soil layer.

3.3 Phenology

Description:

The purpose of the Noah-MP phenology module is to estimate vegetation phenology parameters such as leaf area index (LAI), steam area index (SAI), vegetation fraction and crop growing season. When vegetation is buried by snow effective LAI and SAI are estimated.

Relevant code modules:

Module: PhenologyMainMod.F90

Subroutines: PhenologyMain

Relevant Noah-MP namelist options:

Physical Processes	Options	Notes (* indicates the default option)
OptDynamicVeg	1	off (use table LeafAreaIndex; use VegFrac = VegFracGreen from input)
	2	on (together with OptStomataResistance = 1)
	3	off (use table LeafAreaIndex; calculate VegFrac)
options for dynamic vegetation	4*	off (use table LeafAreaIndex; use maximum vegetation fraction)
	5	on (use maximum vegetation fraction)

	6	on (use VegFrac = VegFracGreen from input)
	7	off (use input LeafAreaIndex; use VegFrac = VegFracGreen from input)
	8	off (use input LeafAreaIndex; calculate VegFrac)
	9	off (use input LeafAreaIndex; use maximum vegetation fraction)

Physics:

1. General LAI and SAI treatments:

Algorithms of Noah-MP Phenology module are activated using the Noah-MP physics option 'OptDynamicVeg'. When OptDynamicVeg = 1, 3, or 4, the subroutine 'PhenologyMain' estimates LAI and SAI at model time steps from monthly values provided in the Noah-MP parameter table (NoahmpTable.TBL). If OptDynamicVeg = 7, 8, or 9, then LAI and SAI are estimated from time series inputs of LAI [m^2/m^2] and SAI [m^2/m^2] derived from remote sensing or ground-based measurements.

SAI Check:

$$SAI = \begin{cases} max(\,0.05,\ 0.10 \times LAI\,)\ , & when\ LAI\ is\ input \\ 0.0 & , when\ LAI < 0.05 \\ 0.0 & , when\ SAI < 0.05 \end{cases} \qquad \text{(Eq. 3.3.1)}$$

LAI Check:

$$LAI = \begin{cases} 0.0\ , & when\ LAI < 0.05 \\ 0.0\ , & when\ SAI = 0.0 \end{cases} \qquad \text{(Eq. 3.3.2)}$$

Both LAI and SAI are set to 0.0 for barren land, water bodies, glaciers, and urban areas.

2. Vegetation buried by snow:

Thickness of canopy buried by snow, d_b [m] is given by

$$d_b = min\{\max(h_{snow} - h_{v,b}, 0.0), (h_{v,t} - h_{v,b})\} \qquad \text{(Eq. 3.3.3)}$$

where, h_{snow} [m] is snow height, $h_{v,t}$ [m] and $h_{v,b}$ [m] are the height of canopy top and bottom, respectively.

The vegetation fraction buried by snow (f_b) is given by

$$f_b = \frac{d_b}{max\{10^{-6}, (h_{v,t} - h_{v,b})\}} \qquad \text{(Eq. 3.3.4)}$$

Note that when $0 < h_{v,t} \le 1.0$, a critical snow depth, $h_{snow,c}$ [m], at which the short vegetation is fully covered by snow, is calculated as

$$h_{snow,c} = h_{v,t} \times e^{\left(\frac{-h_{snow}}{0.2}\right)} \quad \text{(Eq. 3.3.5)}$$

Then f_b is calculated by

$$f_b = \frac{min(h_{snow},\ h_{snow,c})}{h_{snow,c}} \quad \text{(Eq. 3.3.6)}$$

When the vegetation is buried by snow, the effective LAI (E_{LAI}, [m^2/m^2]) and SAI (E_{SAI}, [m^2/m^2]) are calculated as

$$E_{LAI} = LAI \ \times (1.0 - f_b) \quad \text{(Eq. 3.3.7)}$$

$$E_{SAI} = SAI \ \times (1.0 - f_b) \quad \text{(Eq. 3.3.8)}$$

3. Crop growing season:

The Phenology module determines whether it is Growing Season (I_{GS} =1) or not (I_{GS}=0) by

$$I_{GS} = \begin{cases} 1, & when\ T_v > T_{min}\ and\ \ 2 < Growth\ Stage < 7\ and\ \ CropType > 0 \\ 0, & otherwise \end{cases} \quad \text{(Eq. 3.3.9)}$$

where T_v is the temperature of vegetation [K] and T_{min} is the minimum temperature [K] for photosynthesis for each vegetation type.

4. Calculation of Vegetation fraction:

If OptDynamicVeg = 1, 6, or 7, then the vegetation fraction (f_{veg}) is read from input. When OptDynamicVeg = 2, 3, or 8, f_{veg} is estimated from LAI and SAI as

$$f_{veg} = 1.0 - e^{(\,-0.52\,\times\,(LAI+SAI)\,)} \quad \text{(Eq. 3.3.10)}$$

f_{veg} is assumed as the input annual maximum vegetation fraction for OptDynamicVeg = 4, 5, or 9. If the crop model is activated and $CropType > 0$, then f_{veg} is assumed as the input annual maximum vegetation fraction. For barren and glacier lands, f_{veg} is set to 0.0.

5. Flag for dynamic vegetation and crop model:

The Phenology module of Noah-MP can decide whether the dynamic vegetation or crop model are active or not based on following criteria,

- If OptDynamicVeg = 2, 5, or 6, then dynamic vegetation is activated (i.e., FlagDynamicVeg = True)

- If OptCropModel > 0 and $CropType > 0$, then the Noah-MP crop model is activated and dynamic vegetation is turned off (i.e., FlagDynamicCrop = True; FlagDynamicVeg = False).

Notes for dynamic irrigation trigger and sprinkler irrigation:

The dynamic irrigation trigger process and sprinkler irrigation process are described below in the dynamic irrigation subsection within the Hydrology section. The sprinkler irrigated water process is done before canopy water interception and hence will account for the impacts of canopy water interception and canopy hydrology process (described later).

3.4 Canopy Water Interception

Description:

This is the Noah-MP module for Canopy water processes for snow and rain interception. The subsequent hydrological process for intercepted water is done in CanopyHydrologyMod.F90

Relevant code modules:

Module: CanopyWaterInterceptMod.F90

Subroutine: CanopyWaterIntercept

Physics:

1. Canopy liquid water processes:

(1) Maximum liquid water held by canopy ($W_{liq,can,max}$, [mm]):

$$W_{liq,can,max} = W_{liq,can,holdcap} \times (E_{LAI} + E_{SAI}) \quad \text{(Eq. 3.4.1)}$$

where $W_{liq,can,holdcap}$ [mm] is the canopy holding capacity for liquid water as a Noah-MP table parameter depending on vegetation type, E_{LAI} [m^2/m^2] is the effective leaf area index after burying by snow, and E_{SAI} [m^2/m^2] is the effective stem area index after burying by snow. Both E_{LAI} and E_{LAI} are computed in the PhenologyMainMod.F90.

(2) Canopy interception for liquid water ($Q_{liq,int}$, [mm/s]):

If $(E_{LAI} + E_{SAI})$ > 0, then

$$Q_{liq,int} = F_{veg} \times P_{rain} \times F_{prcp} \quad \text{(Eq. 3.4.2)}$$

where F_{veg} is the vegetation fraction (computed in PhenologyMainMod.F90), P_{rain} [mm/s] is the rain rate after rain-snow partitioning (computed in AtmosForcingMod.F90), and F_{prcp} is the fractional area receiving precipitation (computed in AtmosForcingMod.F90). $Q_{liq,int}$ has a minimum value of zero and is capped at a threshold as shown below:

$$Q_{liq,int} = \begin{cases} Q_{liq,int} \ , & Q_{liq,int} < \frac{(W_{liq,can,max} - W_{liq,can})}{\Delta t} \times (1 - e^{\frac{-P_{rain} \times \Delta t}{W_{liq,can,max}}}) \\ \frac{(W_{liq,can,max} - W_{liq,can})}{\Delta t} \times \left(1 - e^{\frac{-P_{rain} \times \Delta t}{W_{liq,can,max}}}\right), & Q_{liq,int} > \frac{(W_{liq,can,max} - W_{liq,can})}{\Delta t} \times (1 - e^{\frac{-P_{rain} \times \Delta t}{W_{liq,can,max}}}) \end{cases}$$

(Eq. 3.4.3)

$$Q_{liq,int} = \begin{cases} Q_{liq,int} \ , & Q_{liq,int} > 0.0 \\ 0.0 \ , & Q_{liq,int} < 0.0 \end{cases} \quad \text{(Eq. 3.4.4)}$$

where $W_{liq,can}$ [mm] is the canopy liquid water and Δt [s] is the Noah-MP main model time step.

If $(E_{LAI} + E_{SAI}) \leq 0$, then

$$Q_{liq,int} = 0 \quad \text{(Eq. 3.4.5)}$$

(3) Canopy dripping for liquid water ($Q_{liq,drip}$, [mm/s]):

If $(E_{LAI} + E_{SAI})$ > 0, then

$$Q_{liq,drip} = F_{veg} \times P_{rain} - Q_{liq,int} \quad \text{(Eq. 3.4.6)}$$

If $(E_{LAI} + E_{SAI}) \leq 0$, then

$$Q_{liq,drip} = 0 \quad \text{(Eq. 3.4.7)}$$

(4) Canopy throughfall for liquid water ($Q_{liq,thro}$, [mm/s]):

If $(E_{LAI} + E_{SAI})$ > 0, then

$$Q_{liq,thro} = (1 - F_{veg}) \times P_{rain} \quad \text{(Eq. 3.4.8)}$$

If $(E_{LAI} + E_{SAI}) \leq 0$, then

$$Q_{liq,thro} = P_{rain} \quad \text{(Eq. 3.4.9)}$$

(5) Canopy liquid water update:

If $(E_{LAI} + E_{SAI}) > 0$, then

$$W_{liq,can} = \begin{cases} W_{liq,can} + Q_{liq,int} \times \Delta t \ , & W_{liq,can} > 0.0 \\ 0.0 \ , & W_{liq,can} < 0.0 \end{cases} \quad \text{(Eq. 3.4.10)}$$

If $W_{liq,can} > 0$ from the previous time step but the canopy is fully buried by snow at this time step ($(E_{LAI} + E_{SAI}) \leq 0$), then $W_{liq,can}$ is reset to 0 and the canopy water is then added to $Q_{liq,drip}$.

2. Canopy ice/snow processes:

(1) Maximum ice/snow held by canopy ($W_{ice,can,max}$, [mm]):

$$W_{ice,can,max} = 6.6 \times (0.27 + \frac{46.0}{\rho_{snowfall}}) \times (E_{LAI} + E_{SAI}) \quad \text{(Eq. 3.4.11)}$$

where $\rho_{snowfall}$ [kg/m^3] is the bulk snowfall density (computed in AtmosForcingMod.F90).

(2) Canopy interception for ice/snow ($Q_{ice,int}$, [mm/s]):

If $(E_{LAI} + E_{SAI}) > 0$, then

$$Q_{ice,int} = F_{veg} \times P_{snow} \times F_{prcp} \quad \text{(Eq. 3.4.12)}$$

where P_{snow} [mm/s] is the snowfall rate after rain-snow partitioning (computed in AtmosForcingMod.F90).

$$Q_{ice,int} = \begin{cases} Q_{ice,int} \ , & Q_{ice,int} < \frac{(W_{ice,can,max} - W_{ice,can})}{\Delta t} \times (1 - e^{\frac{-P_{snow} \times \Delta t}{W_{ice,can,max}}}) \\ \frac{(W_{ice,can,max} - W_{ice,can})}{\Delta t} \times (1 - e^{\frac{-P_{snow} \times \Delta t}{W_{ice,can,max}}}), & Q_{liq,int} > \frac{(W_{ice,can,max} - W_{ice,can})}{\Delta t} \times (1 - e^{\frac{-P_{snow} \times \Delta t}{W_{ice,can,max}}}) \end{cases}$$

(Eq. 3.4.13)

$$Q_{ice,int} = \begin{cases} Q_{ice,int} \ , & Q_{ice,int} > 0.0 \\ 0.0 \ , & Q_{ice,int} < 0.0 \end{cases} \quad \text{(Eq. 3.4.14)}$$

where $W_{ice,can}$ [mm] is the canopy ice and Δt [s] is the Noah-MP main model time step.

If $(E_{LAI} + E_{SAI}) \leq 0$, then

$$Q_{ice,int} = 0 \quad \text{(Eq. 3.4.15)}$$

(3) Canopy dripping for ice/snow ($Q_{ice,drip}$, [mm/s]):

If $(E_{LAI} + E_{SAI}) > 0$, then

$$F_{drip,temp} = \max(0.0,\ \frac{(T_{can}-270.15)}{1.87\times10^5}) \qquad \text{(Eq. 3.4.16)}$$

$$F_{drip,wind} = \frac{\sqrt{(U^2+V^2)}}{1.56\times10^5} \qquad \text{(Eq. 3.4.17)}$$

$$Q_{ice,unload} = \max(0.0, W_{ice,can}) \times (F_{drip,temp} + F_{drip,wind}) \qquad \text{(Eq. 3.4.18)}$$

$$Q_{ice,unload} = \min(Q_{ice,unload}, \frac{W_{ice,can}}{\Delta t} + Q_{ice,int}) \qquad \text{(Eq. 3.4.19)}$$

$$Q_{ice,drip} = \left(F_{veg} \times P_{snow} - Q_{ice,int}\right) + Q_{ice,unload} \qquad \text{(Eq. 3.4.20)}$$

where $F_{drip,temp}$ [1/s] is the ice dripping temperature factor, $F_{drip,wind}$ [1/s] is the frictional velocity representing the ice dripping wind factor, T_{can} [K] is the canopy temperature, U and V are the wind speeds [m/s] in the x and y directions, and $Q_{ice,unload}$ [mm/s] is the canopy ice unloading rate.

If $(E_{LAI} + E_{SAI}) \leq 0$, then

$$Q_{ice,drip} = 0 \qquad \text{(Eq. 3.4.21)}$$

(4) Canopy throughfall for ice/snow ($Q_{ice,thro}$, [mm/s]):

If $(E_{LAI} + E_{SAI}) > 0$, then

$$Q_{ice,thro} = (1 - F_{veg}) \times P_{snow} \qquad \text{(Eq. 3.4.22)}$$

If $(E_{LAI} + E_{SAI}) \leq 0$, then

$$Q_{ice,thro} = P_{snow} \qquad \text{(Eq. 3.4.23)}$$

(5) Canopy ice/snow update:

If $(E_{LAI} + E_{SAI}) > 0$, then

$$W_{ice,can} = \begin{cases} W_{ice,can} + (Q_{ice,int} - Q_{ice,unload}) \times \Delta t\ , & W_{ice,can} > 0.0 \\ 0.0\ , & W_{ice,can} < 0.0 \end{cases} \qquad \text{(Eq. 3.4.24)}$$

If $W_{ice,can} > 0$ from the previous time step but the canopy is fully buried by snow at this time step ($(E_{LAI} + E_{SAI}) \leq 0$), then $W_{ice,can}$ is reset to 0 and the canopy water is then added to $Q_{ice,drip}$.

3. Canopy wetted fraction:

The wet fraction is dependent on the actual canopy ice or liquid water as a fraction of the maximum canopy interception capacity for snow or rain.

$$F_{wet} = \begin{cases} \frac{\max(0.0, W_{ice,can})}{\max(W_{ice,can,max}, 10^{-6})}, & W_{ice,can} > 0.0 \\ \frac{\max(0.0, W_{liq,can})}{\max(W_{liq,can,max}, 10^{-6})}, & W_{ice,can} \leq 0.0 \end{cases} \quad \text{(Eq. 3.4.25)}$$

$$F_{wet} = (\min(1.0, F_{wet}))^{0.667} \quad \text{(Eq. 3.4.26)}$$

where F_{wet} is the wet fraction of canopy.

4. Snow and rain below canopy:

$$Q_{rain} = Q_{liq,thro} + Q_{liq,drip} \quad \text{(Eq. 3.4.27)}$$

$$Q_{snow} = Q_{ice,thro} + Q_{ice,drip} \quad \text{(Eq. 3.4.28)}$$

$$\Delta h_{snow} = \frac{Q_{snow}}{\rho_{snowfall}} \quad \text{(Eq. 3.4.29)}$$

where Q_{rain} [mm/s] is the rain rate below canopy on the ground, Q_{snow} [mm/s] is the snowfall rate below canopy on the ground, $\rho_{snowfall}$ [kg/m^3] is the fresh snowfall density computed in AtmosForcingMod.F90, and Δh_{snow} [m/s] is the snow depth increasing rate due to snowfall. If the land type is water/lake and ground temperature is above freezing temperature, then there is no snow on the ground and hence Q_{snow} and Δh_{snow} are reset to 0.

3.5 Precipitation Heat Advection

Description:

This module is to estimate heat flux advected from precipitation to both canopy and ground.

Relevant code modules:

Module: PrecipitationHeatAdvectMod.F90

Subroutines: PrecipitationHeatAdvect

Physics:

1. Heat advection from rainfall:

$$H_{rain,air2can} = F_{veg} \times P_{rain} \times \frac{C_{wat}}{1000} \times (T_{sfc} - T_{can}) \quad \text{(Eq. 3.5.1)}$$

$$H_{rain,can2grd} = Q_{liq,drip} \times \frac{C_{wat}}{1000} \times (T_{can} - T_{grd}) \quad \text{(Eq. 3.5.2)}$$

$$H_{rain,air2grd} = Q_{liq,thro} \times \frac{C_{wat}}{1000} \times (T_{sfc} - T_{grd}) \quad \text{(Eq. 3.5.3)}$$

where $H_{rain,air2can}$ [W/m^2] is the heat advection from air to canopy due to rain, $H_{rain,can2grd}$ [W/m^2] is the heat advection from canopy to ground due to rain dripping, and $H_{rain,air2grd}$ [W/m^2] is the heat advection from air to ground due to rain throughfall. C_{wat} [J/m^3/K] is the volumetric heat capacity of water defined as a constant in ConstantDefineMod.F90. T_{sfc}, T_{can}, and T_{grd} are the temperature for surface (assumed to equal to precipitation temperature), canopy, and ground. $Q_{liq,drip}$ [mm/s] and $Q_{liq,thro}$ [mm/s] are the rain dripping and throughfall rates (computed from CanopyWaterInterceptMod.F90).

2. Heat advection from snowfall:

$$H_{snow,air2can} = F_{veg} \times P_{rain} \times \frac{C_{ice}}{1000} \times (T_{sfc} - T_{can}) \quad \text{(Eq. 3.5.4)}$$

$$H_{snow,can2grd} = Q_{ice,drip} \times \frac{C_{ice}}{1000} \times (T_{can} - T_{grd}) \quad \text{(Eq. 3.5.5)}$$

$$H_{snow,air2grd} = Q_{ice,thro} \times \frac{C_{ice}}{1000} \times (T_{sfc} - T_{grd}) \quad \text{(Eq. 3.5.6)}$$

where $H_{snow,air2can}$ [W/m^2] is the heat advection from air to canopy due to snowfall, $H_{snow,can2grd}$ [W/m^2] is the heat advection from canopy to ground due to snow unloading, and $H_{snow,air2grd}$ [W/m^2] is the heat advection from air to ground due to snow throughfall. C_{ice} [J/m^3/K] is the volumetric heat capacity of ice defined as a constant in ConstantDefineMod.F90. $Q_{ice,drip}$ [mm/s] and $Q_{ice,thro}$ [mm/s] are the snow dripping and throughfall rates (computed from CanopyWaterInterceptMod.F90).

3. Net heat advection:

The heat advection from rainfall and snowfall are merged together first:

$$H_{pr,air2can} = H_{rain,air2can} + H_{snow,air2can} \quad \text{(Eq. 3.5.7)}$$

$$H_{pr,can2grd} = H_{rain,can2grd} + H_{snow,can2grd} \quad \text{(Eq. 3.5.8)}$$

$$H_{pr,air2grd} = H_{rain,air2grd} + H_{snow,air2grd} \quad \text{(Eq. 3.5.9)}$$

Then:

$$H_{pr,can} = H_{pr,air2can} - H_{pr,can2grd} \quad \text{(Eq. 3.5.10)}$$

$$H_{pr,veggrd} = H_{pr,can2grd} \quad \text{(Eq. 3.5.11)}$$

$$H_{pr,baregrd} = H_{pr,air2grd} \quad \text{(Eq. 3.5.12)}$$

where $H_{pr,can}$ [W/m^2] is the net precipitation heat advected flux for canopy, $H_{pr,veggrd}$ [W/m^2] is the net precipitation heat advected flux for below-canopy ground, $H_{pr,baregrd}$ [W/m^2] is the net precipitation heat advected flux for bare ground.

The $H_{pr,veggrd}$ and $H_{pr,baregrd}$ are further divided by F_{veg} and (1-F_{veg}), respectively, to account for vegetation fraction impact, since $H_{pr,veggrd}$ and $H_{pr,baregrd}$ will be multiplied by the F_{veg} weighting factors later. If $F_{veg} \leq 0$ (canopy buried by snow), then $H_{pr,veggrd}$ will be added to $H_{pr,baregrd}$ and will be reset to 0. Currently, these net precipitation advection fluxes (which are typically very small) are limited to -20~20 W/m2 for the model stability consideration.

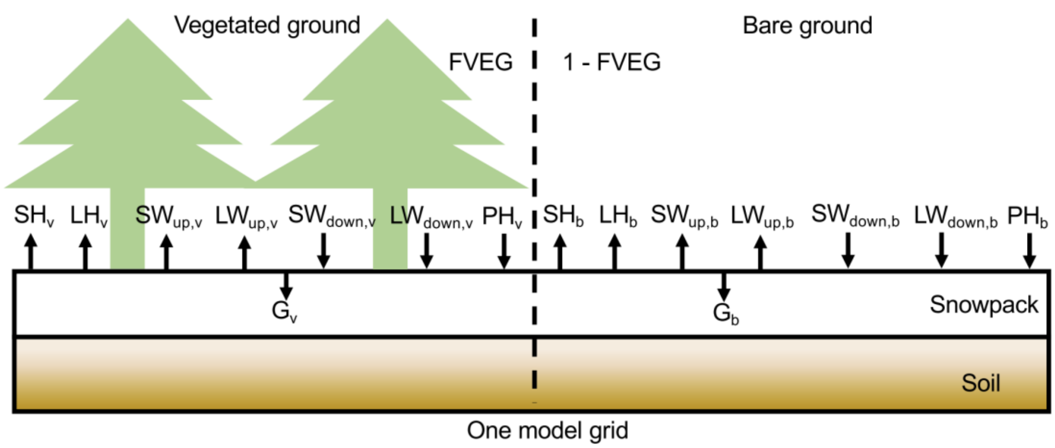
3.6 Energy

Description:

The Noah-MP energy module calculates land surface energy balance, radiation transfer and turbulent transfer. Noah-MP uses a 'tile' approach to compute turbulent fluxes in vegetated fractions and bare fractions separately and then sum them up weighted by fraction. For the radiation transfer, Noah-MP employs traditional and modified two-stream physics options. The modified two-stream scheme assumes tree crowns are evenly distributed within the grid cell with 100% vegetation fraction, but with gaps. The 'tile' approach is not used in radiation transfer because it overlaps too much shadow. The flow diagram of the Noah-MP energy module is provided below.

```
----------------------------------------------------------------------------------------------
turbulence transfer : 'tile' approach to compute energy fluxes in vegetated fraction and
                       bare fraction separately and then sum them up weighted by fraction
                   --------------------------------------
                  / O  O  O  O  O  O  O  O  /          /
                 /  |  |  |  |  |  |  |  | /          /
                / O  O  O  O  O  O  O  O  /          /
               /  |  |  |tile1|  |  |  | /  tile2   /
              / O  O  O  O  O  O  O  O  /  bare    /
             /  |  |  | vegetated |  | /          /
            / O  O  O  O  O  O  O  O  /          /
           /  |  |  |  |  |  |  |  | /          /
          --------------------------------------
----------------------------------------------------------------------------------------------
radiation transfer : modified two-stream (Yang and Friedl, 2003, JGR; Niu ang Yang, 2004, JGR)
                   --------------------------------------  two-stream treats leaves as
                  /   O   O   O   O   O   O   O   O    /  cloud over the entire grid-cell,
                 /    |   |   |   |   |   |   |   |   / while the modified two-stream
                /   O   O   O   O   O   O   O   O    / aggregates cloudy leaves into
               /    |   |   |   |   |   |   |   |   / tree crowns with gaps (as shown in
              /   O   O   O   O   O   O   O   O    / the left figure). We assume these
             /    |   |   |   |   |   |   |   |   / tree crowns are evenly distributed
            /   O   O   O   O   O   O   O   O    / within the gridcell with 100% veg
           /    |   |   |   |   |   |   |   |   / fraction, but with gaps. The 'tile'
          -------------------------------------- approach overlaps too much shadows.
----------------------------------------------------------------------------------------------
```

Figure 9. Demonstration of the approaches for subpixel turbulence and radiative transfer processes for vegetated and bared areas.



$$\text{netRad} + \text{PH} = \text{SH} + \text{LH} + \text{G}$$

$$\text{netRad} = \text{SW}_{\text{down}} - \text{SW}_{\text{up}} + \text{LW}_{\text{down}} - \text{LW}_{\text{up}}$$

Figure 10. Demonstration of surface energy balance treatments in Noah-MP. Energy fluxes over bare ("b" in the subscript) and vegetated ("v" in the subscript) parts of a model grid with a vegetation fraction (F_{VEG}), including shortwave (SW) and longwave (LW) radiation, latent heat (LH), sensible heat (SH), heat into snowpack and soil (G) to drive melting, and heat advected to the surface by precipitation (PH) due to temperature differences between the surface and the air. Adapted from He et al. (2021).

Relevant code modules:

Module: EnergyMainMod.F90

Subroutines: EnergyMain

Physics:

3.6.1 Prepare Key Quantities

Description:

This part is to prepare some key quantities used in energy flux calculations in later sections.

1. Wind speed:

Wind speed at reference height (U_{ref} [m/s]) is given as

$$U_{ref} = \sqrt{u^2 + v^2} \geq 1 \quad \text{(Eq. 3.6.1)}$$

where, u [m/s] is the eastward component of wind speed and v [m/s] is the northward component of wind speed.

2. Vegetated or non-vegetated Check:

The effective vegetation area index (E_{VAI}, [m^2/m^2]) is calculated as

$$E_{VAI} = E_{LAI} + E_{LAI} \quad \text{(Eq. 3.6.2)}$$

If $E_{VAI} > 0.0$, then the model grid is vegetated, else it is not vegetated.

3.6.2 Ground Snow Cover Fraction

Description:

This Noah-MP module is to compute ground snow cover fraction based on Niu and Yang (2007) scheme. A recent study has also implemented and tested additional snow cover parameterizations (Jiang et al., 2020) in Noah-MP, which will be integrated into the community Noah-MP version in the future.

Relevant code modules:

Module: SnowCoverGroundNiu07Mod.F90

Subroutines: SnowCoverGroundNiu07

Physics:

Bulk density of snow (ρ_{snow}, [kg/m^3]) is calculated as

$$\rho_{snow} = \frac{m_{snow}}{h_{snow}} \quad \text{(Eq. 3.6.3)}$$

where m_{snow} [kg/m^2] is the snow mass or snow water equivalent and h_{snow} [m] is the snow height.

The melting factor for snow cover fraction, f_{melt} is given as

$$f_{melt} = \left(\frac{\rho_{snow}}{\rho_{new}}\right)^m \quad \text{(Eq. 3.6.4)}$$

where, ρ_{new} [kg/m^3] is fresh snow density, which is assumed to be 100 kg/m^3 (Niu and Yang, 2007). m is the melting factor determining the curves in melting season, and is adjustable depending on scale (generally a larger value for a larger scale, specified in NoahmpTable.TBL). It can be calibrated against observed snow cover fraction or surface albedo. Ground Snow Cover Fraction or fractional area of the grid cell covered by snow, f_{snow} is calculated as

$$f_{snow} = tanh\left(\frac{h_{snow}}{F_{sno} \times f_{melt}}\right) \quad \text{(Eq. 3.6.5)}$$

where F_{sno} is the snow cover factor parameter (specified in NoahmpTable.TBL).

3.6.3 Ground Roughness Property

Description:

This part is to compute the ground roughness length, displacement height, and surface reference height.

Relevant code modules:

Module: GroundRoughnessPropertyMod.F90

Subroutines: GroundRoughnessProperty

Physics:

1. Ground roughness length for momentum ($z_{0m,g}$ [m]) is given as

$$z_{0m,g} = \begin{cases} z_{0,lake} \times (1 - f_{snow}) + (f_{snow} \times z_{0,snow}) \ , \ for\ lake\ when\ T_g \leq T_{frz} \\ z_{0,lake} \ , \ for\ lake\ when\ T_g > T_{frz} \\ z_{0,soil} \times (1 - f_{snow}) + (f_{snow} \times z_{0,snow}) \ , \ for\ soil \end{cases} \quad \text{(Eq. 3.6.6)}$$

where, $z_{0,lake}$ [m] is the lake surface roughness length, $z_{0,snow}$ [m] is the snow surface roughness length, $z_{0,soil}$ [m] is the soil surface roughness length. T_g and T_{frz} [K] are ground temperature and freezing temperature, respectively.

2. Roughness length for vegetated and non-vegetated ground is given by

$$z_{0m} = \begin{cases} z_{0m,vt} \ , \ for\ vegetated\ surface \\ z_{0m,g} \ , \ for\ non-vegetated\ surface \end{cases} \quad \text{(Eq. 3.6.7)}$$

where $z_{0m,vt}$ [m] is roughness length for momentum determined by vegetation type. Its values are provided in the Noah-MP parameter table (NoahmpTable.TBL).

The zero-plane displacement height (d_0, [m]) is calculated as

$$d_0 = \begin{cases} h_{snow}, \ for\ \ h_{snow} > 0.65 \times h_{can} \\ 0.65 \times h_{can}, \ for\ vegetated\ with\ h_{snow} \leq 0.65 \times h_{can} \end{cases} \quad \text{(Eq. 3.6.8)}$$

where h_{can} [m] is the top of canopy layer.

The reference height (z_{ref}, [m]) is calculated as

$$z_{ref} = max(d_0,\ h_{can}) + z_a' \quad \text{(Eq. 3.6.9)}$$

where z_a' [m] is the reference height above the surface zero plane as received from the input.

When $h_{snow} > z_a$, then the reference height is calculated as

$$z_{ref} = h_{snow} + z_a' \quad \text{(Eq. 3.6.10)}$$

3.6.4 Ground Thermal Property

Description:

This Noah-MP module is to estimate thermal conductivity and heat capacity of snow and soil.

Relevant code modules:

Module: GroundThermalPropertyMod.F90

Subroutines: GroundThermalProperty

Physics:

1. Snow thermal properties:

Relevant code modules:

Submodule: SnowThermalPropertyMod.F90

Subroutines: SnowThermalProperty

Relevant Noah-MP namelist options:

Physical Processes	Options	Notes (* indicates the default option)
OptSnowThermConduct options for snow thermal conductivity	1*	Stieglitz(yen,1965) scheme
	2	Anderson, 1976 scheme
	3	constant
	4	Verseghy (1991) scheme
	5	Douvill(Yen, 1981) scheme

Thermal properties of snow are computed in SnowThermalPropertyMod.F90. Partial volume of ice in a snow layer (the fraction of ice volume to snow volume, $\theta_{ice,sno}$) is given by

$$\theta_{ice,sno}(i) = min\left\{1.0, \frac{W_{ice,sno}(i)}{\Delta z(i) \times \rho_{ice}}\right\} \quad \text{(Eq. 3.6.11)}$$

where i is the layer index ranging from 0 to $I_{n,sno} + 1$, and $I_{n,sno}$ is the snow layer indicator defined as (see detail in snow hydrology section below):

$$I_{n,sno} = \begin{cases} -3, & 3\ layer\ snow \\ -2, & 2\ layer\ snow \\ -1, & 1\ layer\ snow \\ 0, & 0\ layer\ snow \end{cases} \quad \text{(Eq. 3.6.12)}$$

$W_{ice,sno}(i)$ [mm] is ice content in the i^{th} snow layer, Δz [m] is the thickness of snow/soil layers (m), and ρ_{ice} [kg/m^3] is the density of ice (defined in ConstantDefineMod.F90). The effective porosity of snow ($\theta_{e,sno}$) is calculated as

$$\theta_{e,sno}(i) = 1 - \theta_{ice,sno}(i) \quad \text{(Eq. 3.6.13)}$$

Partial volume of liquid water in snow layer ($\theta_{liq,sno}$) is computed as

$$\theta_{liq,sno}(i) = \min\left\{\theta_{e,sno}(i), \frac{W_{liq,sno}(i)}{\Delta z(i) \times \rho_{liq}}\right\} \quad \text{(Eq. 3.6.14)}$$

where $W_{liq,sno}(i)$ [mm] is liquid water content in the i^{th} snow layer and ρ_{liq} [kg/m^3] is the density of water (defined in ConstantDefineMod.F90).

The bulk density of snow (ρ_{snow}, [kg/m^3]) is given as

$$\rho_{snow}(i) = \frac{W_{ice,sno}(i) + W_{liq,sno}(i)}{\Delta z(i)} \quad \text{(Eq. 3.6.15)}$$

The volumetric specific heat capacity of snow ($C_{h,snow}$, [J/m^3/K]) is calculated by

$$C_{h,snow} = C_{h,ice} \times \theta_{ice,sno} + C_{h,wat} \times \theta_{liq,sno} \quad \text{(Eq. 3.6.16)}$$

where $C_{h,ice}$ [J/m^3/K] and $C_{h,wat}$ [J/m^3/K] are the volumetric specific heat capacity of ice and water, respectively (defined in ConstantDefineMod.F90).

Based on the physics option OptSnowThermConduct, the thermal conductivity of snow (k_{snow}, [W/m/K]) is calculated as

$$k_{snow} = \begin{cases} 3.2217 \times 10^{-6} \times \rho_{snow}^2, & OptSnowThermConduct = 1 \\ 2 \times 10^{-2} + 2.5 \times 10^{-6} \times \rho_{snow}^2, & OptSnowThermConduct = 2 \\ 0.35, & OptSnowThermConduct = 3 \\ 2.576 \times 10^{-6} \times \rho_{snow}^2 + 0.074, & OptSnowThermConduct = 4 \\ 2.22 \times \left(\frac{\rho_{snow}}{1000}\right)^{1.88}, & OptSnowThermConduct = 5 \end{cases} \quad \text{(Eq. 3.6.17)}$$

2. Soil thermal properties:

Relevant code modules:

Submodule: SoilThermalPropertyMod.F90

Subroutines: SoilThermalProperty

Thermal properties of soil are estimated in SoilThermalPropertyMod.F90. Partial volume of soil ice (i.e., soil ice content, $W_{ice,soil}$, [m^3/m^3]) in the i^{th} soil layer is given by

$$W_{ice,soil}(i) = \theta_{soil}(i) - W_{ice,soil}(i) \quad \text{(Eq. 3.6.18)}$$

Volumetric specific heat capacity of soil ($C_{h,soil}$, [J/m^3/K]) is calculated as

$$C_{h,soil} = W_{liq,soil} \times C_{h,wat} + \left(1 - \theta_{soil,max}\right) \times C_{soil,solid} + \left(\theta_{soil,max} - \theta_{soil}\right) \times C_{air} + W_{ice,soil} \times C_{h,ice} \quad \text{(Eq. 3.6.19)}$$

where $\theta_{soil,max}$ [m^3/m^3] is the saturated soil moisture (specified in NoahmpTable.TBL) and $C_{soil,solid}$ [J/m^3/K] is the volumetric specific heat capacity of soil solid (non-ice) components (specified in NoahmpTable.TBL).

The soil saturation ratio (S_r) is given by

$$S_r = \frac{\theta_{soil}}{\theta_{soil,max}} \quad \text{(Eq. 3.6.20)}$$

Thermal conductivity of soil solids (k_{sld}, [W/m/K]) is calculated as

$$k_{sld} = k_{qtz}^{f_{qtz}} \times k_{oth}^{1-f_{qtz}} \quad \text{(Eq. 3.6.21)}$$

where k_{qtz} [W/m/K] is thermal conductivity for quartz (k_{qtz}= 7.7, currently hard-coded). k_{oth} [W/m/K] is thermal conductivity for the other soil component (k_{oth}= 2.0, currently hard-coded). f_{qtz} is the soil quartz content which depends on soil type and is provided in the 'NoahmpTable.TBL'.

The unfrozen fraction (f_{uf}) of soil is computed as

$$f_{uf} = \frac{W_{liq,soil}}{\theta_{soil}} \quad \text{(Eq. 3.6.22)}$$

Moisture in the soil layer is completely frozen if f_{uf} = 0.0 and it is 100% liquid if f_{uf} = 1.0 .

Unfrozen soil volume of saturation ($\theta_{uf,sat}$) is calculated as

$$\theta_{uf,sat} = f_{uf} \times \theta_{soil,max} \quad \text{(Eq. 3.6.23)}$$

Saturated soil thermal conductivity ($k_{soil,sat}$, [W/m/K]) is given by

$$k_{soil,sat} = k_{sld}^{(1-\theta_{soil,max})} \times k_{ice}^{(\theta_{soil,max}-\theta_{uf,sat})} \times k_{wat}^{\theta_{uf,sat}} \quad \text{(Eq. 3.6.24)}$$

where k_{ice} [W/m/K] is thermal conductivity of ice (defined in ConstantDefineMod.F90). k_{wat} [W/m/K] is the thermal conductivity of water (defined in ConstantDefineMod.F90).

Dry density of soil ($\rho_{soil,dry}$, [kg/m^3]) is computed as

$$\rho_{soil,dry} = \left(1 - \theta_{soil,max}\right) \times 2700 \quad \text{(Eq. 3.6.25)}$$

Dry thermal conductivity of soil ($k_{soil,dry}$, [W/m/K]) is given by

$$k_{soil,dry} = \frac{0.135 \times \rho_{soil,dry} + 64}{2700 - 0.947 \times \rho_{soil,dry}} \quad \text{(Eq. 3.6.26)}$$

The Kersten number (k_e) as a function of the saturation S_r and the phase of water, uses the "fine" formula, valid for soils containing at least 5% of particles with diameter less than 2.0×10^{-6} m (Peters-Lidard et al., 1998). It is calculated by

$$k_e = \begin{cases} S_r, & For\ frozen\ soil\ (W_{liq,soil} + 0.0005 < \theta_{soil}) \\ \ln(S_r) + 1.0, & For\ unfrozen\ soil\ and\ S_r > 0.1 \\ 0.0, & For\ unfrozen\ soil\ and\ S_r \le 0.1 \end{cases} \quad \text{(Eq. 3.6.27)}$$

The final thermal conductivity of soil is computed as:

$$k_{soil} = k_e \times \left(k_{soil,sat} - k_{soil,dry}\right) + k_{soil,dry} \quad \text{(Eq. 3.6.28)}$$

If the land type is urban, then $k_{soil} = 3.24$ [W/m/K].

3. Lake thermal properties:

Relevant code modules:

Module: GroundThermalPropertyMod.F90

Subroutines: GroundThermalProperty

This section is incorporated in GroundThermalPropertyMod.F90. The lake heat capacity ($C_{h,lake}$, [J/m^3/K]) is calculated as

$$C_{h,lake} = \begin{cases} C_{h,wat}, & unfrozen \\ C_{h,ice}, & frozen \end{cases} \quad \text{(Eq. 3.6.29)}$$

The thermal conductivity of lake (k_{lake}, [W/m/K]) is given as

$$k_{lake} = \begin{cases} k_{wat}, & unfrozen \\ k_{ice}, & frozen \end{cases} \quad \text{(Eq. 3.6.30)}$$

A phase change factor (f_{phase}) used for melting/freezing of snow or soil:

$$f_{phase} = \frac{\Delta t}{C_{h,snow/soil} \times \Delta z} \quad \text{(Eq. 3.6.31)}$$

For snowpack without an explicit layer ($I_{n,sno}$ =0), the thermal conductivity of the snow and soil interface (k_{int}, [W/m/K]) is estimated as

$$k_{int} = \frac{k_{soil} \times \Delta z + k_{snow} \times z_{snow}}{\Delta z + z_{snow}} \quad \text{(Eq. 3.6.32)}$$

where k_{soil} [W/m/K] is the thermal conductivity calculated for layer 1 (top soil layer). Snow thermal conductivity k_{snow} [W/m/K] is set to 0.35. z_{snow} [m] is snow height.

3.6.5 Surface Albedo

Description:

The purpose of this Noah-MP module is to compute the surface (including canopy) albedo, fluxes (per unit incoming direct and diffuse radiation) reflected, transmitted, and absorbed by vegetation, and sunlit fraction of the canopy. These processes are only done if there is sunlight.

Relevant code modules:

Module: SurfaceAlbedoMod.F90

Subroutines: SurfaceAlbedo

Physics:

1. Prepare key quantities for surface albedo calculations:

The effective Vegetation Area Index (E_{VAI}) is given as

$$E_{VAI} = E_{LAI} + E_{SAI} \quad \text{(Eq. 3.6.33)}$$

The weighted E_{LAI} ($E_{LAI,wt}$) and E_{SAI} ($E_{SAI,wt}$) with E_{VAI} is calculated as

$$E_{LAI,wt} = \frac{E_{LAI}}{\max(10^{-6}, E_{VAI})} \quad \text{(Eq. 3.6.34)}$$

$$E_{SAI,wt} = \frac{E_{SAI}}{\max(10^{-6}, E_{VAI})} \quad \text{(Eq. 3.6.35)}$$

Weighted reflectance/transmittance by E_{LAI} and E_{SAI} is calculated as

$$\rho_{v,b} = \rho_{l,b} \times E_{LAI,wt} + \rho_{s,b} \times E_{SAI,wt} \quad \text{(Eq. 3.6.36)}$$

$$\tau_{v,b} = \tau_{l,b} \times E_{LAI,wt} + \tau_{s,b} \times E_{SAI,wt} \quad \text{(Eq. 3.6.37)}$$

where $\rho_{l,b}$, $\rho_{s,b}$ and $\rho_{v,b}$ are reflectance of leaf, stem, and vegetation, respectively, for a given waveband b. $\tau_{l,b}$, $\tau_{s,b}$ and $\tau_{v,b}$ are transmittance of leaf, stem, and vegetation, respectively, for a given waveband b. The leaf and stem reflectance and transmittance are set as parameters in NoahmpTable.TBL.

2. Snow age (only used by BATS snow albedo scheme):

Relevant code modules:

Submodule: SnowAgingBatsMod.F90

Subroutines: SnowAgingBats

Snow age based on BATS snow albedo scheme is calculated as follows:

Snow aging time factor ($\Delta t_{f,snow}$) is computed as:

$$\Delta t_{f,snow} = \frac{\Delta t}{\tau_0} \quad \text{(Eq. 3.6.38)}$$

where τ_0 is the snow age factor (specified in NoahmpTable.TBL) and Δt [s] is the model timestep.

The effect of snow grain growth due to vapor diffusion (A_1) is computed as:

$$A_1 = e^{f_{A1} \times \left[\frac{1}{T_{frz}} - \frac{1}{T_g}\right]} \quad \text{(Eq. 3.6.39)}$$

where T_{frz} and T_g [K] are the freezing and ground temperatures, respectively. f_{A1} is the vapor diffusion parameter (specified in NoahmpTable.TBL).

The effect of grain growth from freezing of melt water (A_2) is computed as

$$A_2 = e^{\min(0, f_{A2} \times f_{A1} \times \left[\frac{1}{T_{frz}} - \frac{1}{T_g}\right])} \quad \text{(Eq. 3.6.40)}$$

where f_{A2} is the extra freezing growth parameter (specified in NoahmpTable.TBL).

The effect of grain growth due to soot (A_3) is computed as

$$A_3 = f_{A3} \quad \text{(Eq. 3.6.41)}$$

where f_{A3} is the soot-induced snow aging parameter (specified in NoahmpTable.TBL).

The total snow aging effect (A_t) is then computed as:

$$A_t = A_1 + A_2 + A_3 \quad \text{(Eq. 3.6.42)}$$

Fresh snowfall factor ($f_{fr,snow}$) is calculated as

$$f_{fr,snow} = \frac{\max(0.0, m_{snow} - m_{snow,p})}{m_{snow,mx}} \quad \text{(Eq. 3.6.43)}$$

where m_{snow} [mm] is the snow water equivalent, $m_{snow,p}$ [mm] is the snow water equivalent from previous time step, and $m_{snow,mx}$ [mm] is the new snow mass to fully cover old snow, which is provided in the 'NoahmpTable.TBL'.

Non-dimensional age of snow (τ_{ss}) is defined as

$$\tau_{ss} = \max(0.0, \tau_{ss} + (A_t \times \Delta t_{f,snow}) \times (1.0 - f_{fr,snow})) \quad \text{(Eq. 3.6.44)}$$

Snow age factor (S_a) is calculated as

$$S_a = \frac{\tau_{ss}}{\tau_{ss}+1} \quad \text{(Eq. 3.6.45)}$$

3. Snow Albedo:

Relevant code modules:

Submodule: SnowAlbedoBatsMod.F90

Subroutines: SnowAlbedoBats

Submodule: SnowAlbedoClassMod.F90

Subroutines: SnowAlbedoClass

Relevant Noah-MP namelist options:

Physical Processes	Options	Notes (* indicates the default option)
OptSnowAlbedo options for ground snow surface albedo	1*	BATS snow albedo scheme
	2	CLASS snow albedo scheme

(1) BATS Scheme (OptSnowAlbedo=1):

This part computes snow albedo based on BATS scheme (Yang et al.,1997).

The corrected zenith angle (Z_c) is calculated as

$$Z_c = \max(0, \frac{\left(1.0+\frac{1}{cosZ_b}\right)}{1.0+(2.0\times cosZ_b \times cosZ)} - \frac{1}{cosZ_b}) \quad \text{(Eq. 3.6.46)}$$

where $cosZ_b$ is the zenith angle adjustment for snow albedo estimation by BATS (specified in NoahmpTable.TBL). $cosZ$ is the cosine solar zenith angle.

Snow albedo for diffuse solar radiation (visible spectrum), $\alpha_{s,i,Vis}$, is calculated as

$$\alpha_{s,i,Vis} = \alpha_{s,Vis} \times \left[1.0 - \left(S_{a,b,Vis} \times S_a\right)\right] \quad \text{(Eq. 3.6.47)}$$

where $\alpha_{s,Vis}$ is the new snow visible albedo (specified in NoahmpTable.TBL). $S_{a,b,Vis}$ is the BATS age-related visible band parameter (specified in NoahmpTable.TBL), and S_a is the snow age factor computed in SnowAgingBatsMod.F90 above.

Snow albedo for diffuse solar radiation (near-infrared spectrum), $\alpha_{s,i,NIR}$, is calculated as

$$\alpha_{s,i,NIR} = \alpha_{s,NIR} \times \left[1.0 - \left(S_{a,b,NIR} \times S_a\right)\right] \quad \text{(Eq. 3.6.48)}$$

where $\alpha_{s,NIR}$ is the new snow near-infrared albedo (specified in NoahmpTable.TBL). $S_{a,b,NIR}$ is the BATS age-related NIR band parameter (specified in NoahmpTable.TBL).

Snow albedo for direct solar radiation (visible spectrum), $\alpha_{s,d,Vis}$ is calculated as

$$\alpha_{s,d,Vis} = \alpha_{s,i,Vis} + f_{cosZ,Vis} \times Z_c \times [1.0 - \alpha_{s,i,Vis}] \qquad \text{(Eq. 3.6.49)}$$

where $f_{cosZ,Vis}$ is the $cosZ$ factor for direct visible snow albedo (specified in NoahmpTable.TBL).

Snow albedo for direct solar radiation (near-infrared spectrum), $\alpha_{s,d,Vis}$ is calculated as

$$\alpha_{s,d,NIR} = \alpha_{s,i,NIR} + f_{cosZ,NIR} \times Z_c \times [1.0 - \alpha_{s,i,NIR}] \qquad \text{(Eq. 3.6.50)}$$

where, $f_{cosZ,NIR}$ is the $cosZ$ factor for direct near-infrared snow albedo (specified in NoahmpTable.TBL).

(2) CLASS Scheme (OptSnowAlbedo=2):

This part computes snow albedo based on CLASS scheme (Verseghy, 1991).

The snow albedo (α_s) based on the CLASS scheme is calculated as

$$\alpha_s = \alpha_{Ref,CLASS} + (\alpha_{s,p} - \alpha_{Ref,CLASS}) \times e^{\frac{-0.01 \times \Delta t}{S_{a,c}}} \qquad \text{(Eq. 3.6.51)}$$

where $\alpha_{Ref,CLASS}$ is the reference snow albedo in the CLASS scheme (specified in NoahmpTable.TBL). $\alpha_{s,p}$ is the snow albedo from previous time step. $S_{a,c}$ [s] is the snow aging e-folding time in the CLASS albedo scheme,

When the snowfall rate at the ground (Q_{snow} [mm/s]) is larger than zero, then:

$$\alpha_s = \alpha_s + min\left(Q_{snow}, \frac{m_{snow,mx}}{\Delta t}\right) \times (\alpha_{s,CLASS} - \alpha_s) \qquad \text{(Eq. 3.6.52)}$$

where $\alpha_{s,CLASS}$ is fresh snow albedo in the CLASS scheme (specified in NoahmpTable.TBL).

Note that there are some recent developments enhancing the snow albedo parameters using in-situ observations (Abolafia-Rosenzweig et al., 2022b) and coupling the snow radiative transfer (SNICAR) model with an older Noah-MP version (Wang et al., 2020, 2022), which will be integrated into the community Noah-MP version in the future.

3. Ground Albedo:

Relevant code modules:

Submodule: GroundAlbedoMod.F90

Subroutines: GroundAlbedo

Noah-MP computes ground albedo based on soil and snow albedo as follows:

$$C_w = 0.11 - 0.4 \times \theta_{soil}(1) \qquad \text{(Eq. 3.6.53)}$$

where C_w is the correction factor for soil albedo because of soil water which can change the soil albedo across a relatively large range. $\theta_{soil}(1)$ [m^3/m^3] is soil moisture for the top soil layer.

(1) For soil:

Soil albedo for direct solar radiation, α_{dir} is calculated as

$$\alpha_{dir} = \begin{cases} \alpha_{sat} + C_w, \ if \ \alpha_{sat} + C_w < \alpha_{dry} \\ \alpha_{dry}, \ if \ \alpha_{sat} + C_w > \alpha_{dry} \end{cases} \qquad \text{(Eq. 3.6.54)}$$

where α_{sat} and α_{dry} are saturated soil albedo and dry soil albedo, respectively. Based on the above equation, higher soil water content corresponds to lower soil albedo, which results from the higher heat capacity of water than that of soil and the transparency of water.

Soil albedo for diffuse solar radiation, α_{dif} is calculated as

$$\alpha_{dfs} = \alpha_{dir} \qquad \text{(Eq. 3.6.55)}$$

(2) For lake:

If $T_g > T_{frz}$ (unfrozen lake):

$$\alpha_{dir} = \frac{0.06}{(\max(0.01, cosZ))^{1.7}} + 0.15 \qquad \text{(Eq. 3.6.56)}$$

$$\alpha_{dfs} = 0.06 \qquad \text{(Eq. 3.6.57)}$$

If $T_g \le T_{frz}$ (frozen lake):

$$\alpha_{dir} = \alpha_{lake,frz} \qquad \text{(Eq. 3.6.58)}$$

$$\alpha_{dfs} = \alpha_{dir} \qquad \text{(Eq. 3.6.59)}$$

where T_g [K] is temperature of the ground, T_{frz} [K] is the freezing temperature, and $\alpha_{lake,frz}$ is the frozen lake surface albedo (specified in NoahmpTable.TBL).

By combining the snow albedo, ground albedo can be estimated as:

$$\alpha_{g,dir} = \alpha_{dir} \times (1.0 - f_{snow}) + \alpha_{s,d} \times f_{snow} \qquad \text{(Eq. 3.6.60)}$$

$$\alpha_{g,dfs} = \alpha_{dfs} \times (1.0 - f_{snow}) + \alpha_{s,i} \times f_{snow} \qquad \text{(Eq. 3.6.61)}$$

where $\alpha_{g,d}$ and $\alpha_{g,i}$ are albedos of ground for direct and diffuse radiation. $\alpha_{s,d}$ and $\alpha_{s,i}$ are albedos of snow for direct and diffuse radiation computed above.

4. Two-Stream Canopy Radiation Transfer Scheme:

Relevant code modules:

Submodule: CanopyRadiationTwoStreamMod.F90

Subroutines: CanopyRadiationTwoStream

Relevant Noah-MP namelist options:

Physical Processes	Options	Notes (* indicates the default option)
OptCanopyRadiationTransfer options for canopy radiation transfer	1	modified two-stream (gap=F(solar angle,3D structure, etc) < 1-VegFrac)
	2	two-stream applied to grid-cell (gap=0)
	3*	two-stream applied to vegetated fraction (gap=1-VegFrac)

Noah-MP uses three versions of two-stream canopy radiative transfer schemes based on the two-stream approximation of Dickinson (1983) and Sellers (1985) to calculate fluxes absorbed by vegetation, reflected by vegetation, and transmitted through vegetation for unit incoming direct or diffuse flux given an underlying surface with known albedo.

The more recent modified version of the two-stream canopy radiative transfer scheme (OptCanopyRadiationTransfer = 1) works by inducing the total canopy gap probability, P_c, which equals to the sum of the between-crown gap probability, P_{bc}, and the within-crown gap probability, P_{wc}:

$$P_{bc} = e^{\frac{-\rho_t \times \pi \times R^2}{\cos(\theta')}} \qquad \text{(Eq. 3.6.62)}$$

where R [m] is the tree crown radius, ρ_t and θ' are calculated as follows:

$$\rho_t = -\frac{\log(max\{1.0 - f_{veg},\ 0.01\})}{\pi \times R^2} \qquad \text{(Eq. 3.6.63)}$$

$$\theta' = tan^{-1}\left[\frac{H_{top} - H_{bot}}{2R} \times tan(\theta)\right] \qquad \text{(Eq. 3.6.64)}$$

Here θ is the solar zenith angle limited to a minimum of Z and 89.5°. H_{top} [m] and H_{bot} [m] are height to canopy top and bottom.

$$P_{wc} = (1 - P_{bc}) \times e^{\frac{-0.5 \times F_a \times H_d}{\cos(\theta)}} \qquad \text{(Eq. 3.6.65)}$$

where F_a [m^{-1}] is the foliage volume density and H_d [m] is the crown depth.

$$H_d = H_{top} - H_{bot} \quad \text{(Eq. 3.6.66)}$$

$$F_a = \frac{E_{LAI}}{1.33 \times \pi \times R^3 \times \frac{H_d}{2R} \times \rho_t} \quad \text{(Eq. 3.6.67)}$$

where E_{LAI} is the effective vegetation (leaf + stem) area index.

Gap fraction (K_{open}) and total canopy gap probability (P_c) for diffuse light is given by

$$K_{open} = \int_0^{\frac{\pi}{2}} P_{bc} \times \sin(2\theta) d\theta \quad \text{(Eq. 3.6.68)}$$

$$P_c = min \begin{cases} 1 - f_{veg} \\ P_{bc} + P_{wc} \end{cases} \quad \text{(Eq. 3.6.69)}$$

If OptCanopyRadiationTransfer = 1:

$$P_c = min \begin{cases} 1 - f_{veg} \\ P_{bc} + P_{wc} \end{cases} \quad \text{(Eq. 3.6.70)}$$

$$K_{open} = 0.05 \quad \text{(Eq. 3.6.71)}$$

If OptCanopyRadiationTransfer = 2:

$$P_c = 0.0 \quad \text{(Eq. 3.6.72)}$$

$$K_{open} = 0.0 \quad \text{(Eq. 3.6.73)}$$

If OptCanopyRadiationTransfer = 3:

$$P_c = 1 - f_{veg} \quad \text{(Eq. 3.6.74)}$$

$$K_{open} = 1 - f_{veg} \quad \text{(Eq. 3.6.75)}$$

The flux absorbed by vegetation (f_{ab}) is calculated as,

$$f_{ab} = 1 - f_{re} - (1 - \alpha_d) \times f_{td} - (1 - \alpha_i) \times f_{ti} \quad \text{(Eq. 3.6.76)}$$

Where α_d is direct albedo of underlying surface and α_i is the diffuse albedo of underlying surface.

f_{re} is the total reflected flux by vegetation and ground computed as:

$$f_{re} = \begin{cases} \left[\frac{h_1}{\sigma} + h_2 + h_3\right] \times (1 - P_c) + \alpha_d \times P_c \quad , \ for\ direct\ beam \\ (h_7 + h_8) \times \left(1 - K_{open}\right) + \alpha_i \times K_{open}, \ for\ diffuse\ beam \end{cases} \quad \text{(Eq. 3.6.77)}$$

The downward direct flux below vegetation (f_{td}) is calculated as,

$$f_{td} = \begin{cases} S_2 \times (1 - P_c) + P_c, & for\ direct\ beam \\ 0, & for\ diffuse\ beam \end{cases} \quad \text{(Eq. 3.6.78)}$$

The downward diffuse flux below vegetation (f_{ti}) is calculated as

$$f_{ti} = \begin{cases} \left[\frac{h_4 \times S_2}{\sigma} + h_5 \times S_1 + \frac{h_6}{S_1}\right] \times (1 - P_c) & , \ for\ direct\ beam \\ \left[h_9 \times S_1 + \frac{h_{10}}{S_1}\right]\left(1 - K_{open}\right) + K_{open} & , \ for\ diffuse\ beam \end{cases} \quad \text{(Eq. 3.6.79)}$$

where S_1 and S_2 are expressed as:

$$S_1 = e^{-H \times E_{LAI}} \quad \text{(Eq. 3.6.80)}$$

$$S_2 = e^{-\tau_{dir} \times E_{LAI}} \quad \text{(Eq. 3.6.81)}$$

where τ_{dir} is optical depth of direct beam per unit leaf area, and H can be calculated as following:

$$H = \frac{\sqrt{t_1}}{\tau_{avd}} \quad \text{(Eq. 3.6.82)}$$

where τ_{avd} is average diffuse optical depth. In the above equations, h_1, h_2, ..., h_{10} are intermediate variables used to simplify the equations, and they are expressed as the following:

$$h_1 = t_0 \times \Omega \times \beta_d \times (b - t_0) - \Omega^2 \times \beta_i \times t_0 \times (1 - \beta_d) \quad \text{(Eq. 3.6.83)}$$

$$h_2 = \frac{\frac{t_2 \times t_6}{S_1} - (b - \tau_{avd} \times H) \times t_7}{d_1} \quad \text{(Eq. 3.6.84)}$$

$$h_3 = \frac{t_3 \times t_6 \times S_1 - (b - \tau_{avd} \times H) \times t_7}{d_1} \quad \text{(Eq. 3.6.85)}$$

$$h_4 = -t_0 \times \Omega \times (1 - \beta_d) \times (b + t_0) - \Omega^2 \times \beta_i \times \beta_d \times t_0 \quad \text{(Eq. 3.6.86)}$$

$$h_5 = \frac{\frac{t_4 \times t_8}{S_1} + t_9}{d_2} \quad \text{(Eq. 3.6.87)}$$

$$h_6 = \frac{t_5 \times t_8 \times S_1 + t_9}{d_2} \quad \text{(Eq. 3.6.88)}$$

$$h_7 = \frac{\Omega \times \beta_i \times t_2}{d_1 \times S_1} \quad \text{(Eq. 3.6.89)}$$

$$h_8 = \frac{\Omega \times \beta_i \times t_3 \times S_1}{d_1} \quad \text{(Eq. 3.6.90)}$$

$$h_9 = \frac{t_4}{d_2 \times S_1} \quad \text{(Eq. 3.6.91)}$$

$$h_{10} = \frac{-t_5 \times S_1}{d_2} \quad \text{(Eq. 3.6.92)}$$

In the above equations, t_1, t_2, …, t_9 are intermediate variables used to simplify the equations, and they are express as the following:

$$t_0 = \tau_{avd} \times \tau_{dir} \quad \text{(Eq. 3.6.93)}$$

$$t_1 = b^2 - c^2 \quad \text{(Eq. 3.6.94)}$$

$$t_2 = u_1 - \tau_{avd} \times H \quad \text{(Eq. 3.6.95)}$$

$$t_3 = u_1 + \tau_{avd} \times H \quad \text{(Eq. 3.6.96)}$$

$$t_4 = u_2 - \tau_{avd} \times H \quad \text{(Eq. 3.6.97)}$$

$$t_5 = u_2 + \tau_{avd} \times H \quad \text{(Eq. 3.6.98)}$$

$$t_6 = t_0 \times \Omega \times \beta_d - \frac{h_1 \times (b + t_0)}{\sigma} \quad \text{(Eq. 3.6.99)}$$

$$t_7 = \left[t_0 \times \Omega \times \beta_d - \Omega \times \beta_i - \frac{h_1}{\sigma} \times (u_1 + t_0)\right] \times S_2 \quad \text{(Eq. 3.6.100)}$$

$$t_8 = \frac{h_4}{\sigma} \quad \text{(Eq. 3.6.101)}$$

$$t_9 = [u_3 - t_8 \times (u_2 + t_0)] \times S_2 \quad \text{(Eq. 3.6.102)}$$

where β_d is the upscatter parameter for direct beam radiation and β_i is the upscatter parameter for diffuse radiation.

The fraction of intercepted radiation that is scattered (Ω) can be expressed as,

$$\Omega = \begin{cases} \Omega_L & , \; when \; T_v > T_{frz} \\ (1 - f_{wet}) \times \Omega_L + f_{wet} \times \Omega_s, & when \; T_v \leq T_{frz} \end{cases} \quad \text{(Eq. 3.6.103)}$$

where f_{wet} is the fraction of leaf and stem area index that is wetted, Ω_L and Ω_s are the fraction of intercepted radiation that is scattered by leaves and stem, respectively. T_v [K] is the vegetation temperature. T_{frz} [K] is the freezing point temperature.

b, c, d_1 and d_2 are intermediate variables and expressed as the following,

$$b = 1 - \Omega + \Omega \times \beta_i \quad \text{(Eq. 3.6.104)}$$

$$c = \Omega \times \beta_i \quad \text{(Eq. 3.6.105)}$$

$$d_1 = \frac{(b + \tau_{avd}) \times t_2}{S_1} - (b - \tau_{avd} \times H) \times t_3 \times S_1 \quad \text{(Eq. 3.6.106)}$$

$$d_2 = \frac{t_4}{S_1} - t_5 \times S_1 \quad \text{(Eq. 3.6.107)}$$

where u_1, u_2, and u_3 are expressed as,

$$u_1 = b - \frac{c}{\alpha} \quad \text{(Eq. 3.6.108)}$$

$$u_2 = b - c \times \alpha \quad \text{(Eq. 3.6.109)}$$

$$u_3 = t_0 \times \Omega \times (1 - \beta_d) + c \times \alpha \quad \text{(Eq. 3.6.110)}$$

where α is albedo of underlying surface,

$$\alpha = \begin{cases} \alpha_d, \; for \; direct \; beam \\ \alpha_i, \; for \; diffuse \; beam \end{cases} \quad \text{(Eq. 3.6.111)}$$

and β_i and β_d are expressed as,

$$\beta_d = \begin{cases} \beta_{dl} & , \; when \; T_v > T_{frz} \\ \frac{(1 - f_{wet}) \times \Omega_L \times \beta_{dl} + f_{wet} \times \Omega_s \times \beta_{ds}}{\Omega} & , \; when \; T_v \leq T_{frz} \end{cases} \quad \text{(Eq. 3.6.112)}$$

$$\beta_i = \begin{cases} \beta_{il} & , \ when\ T_v > T_{frz} \\ \frac{(1-f_{wet})\times\Omega_L\times\beta_{dl}+f_{wet}\times\Omega_s\times\beta_{ds}}{\Omega}, & when\ T_v \le T_{frz} \end{cases}$$ (Eq. 3.6.113)

For leaves,

$$\beta_{dl} = \frac{[1+\tau_{avd}\times\tau_{dir}]\times A}{\Omega_L\times\tau_{avd}\times\tau_{dir}}$$ (Eq. 3.6.114)

$$\beta_{il} = \frac{\rho\times\left[1+\frac{X_L}{2}\right]^2}{\Omega_L}$$ (Eq. 3.6.115)

where A is single scattering albedo expressed as,

$$A = \frac{0.5\times(\rho+\tau)\times\left[\frac{\varphi_1+\varphi_2\times cos\theta'}{\varphi_1+2\times\varphi_2\times cos\theta'}\right]\times\left[1-\frac{\varphi_1\times cos\theta'}{(\varphi_1+2\times\varphi_2\times cos\theta')}\times log\left(\frac{\varphi_1\times(1+cos\theta')+2\times\varphi_2\times cos\theta'}{\varphi_1\times cos\theta'}\right)\right]}{(\varphi_1+2\times\varphi_2\times cos\theta')}$$

(Eq. 3.6.116)

$$\tau_{avd} = \frac{1-\frac{\varphi_1}{\varphi_2}\times log\left(\frac{\varphi_1+\varphi_2}{\varphi_1}\right)}{\varphi_2}$$ (Eq. 3.6.117)

$$\tau_{dir} = \frac{\varphi_1}{cos\theta'} + \varphi_2$$ (Eq. 3.6.118)

$$\varphi_1 = 0.5 - 0.633\times X_L - 0.33\times {X_L}^2$$ (Eq. 3.6.119)

$$\varphi_2 = 0.877\times(1-2\times\varphi_1)$$ (Eq. 3.6.120)

$$X_L = \begin{cases} x, & if\ x > 0.01 \\ 0.01, & if\ x \le 0.01 \end{cases}$$ (Eq. 3.6.121)

where x ranges from -0.4 to 0.6 (specified in NoahmpTable.TBL).

5. Other radiation diagnostic quantities:

Sunlit fraction of canopy (f_{sun}):

$$f_{sun} = \frac{1-e^{(-\sigma_{ext}\times E_{VAI})}}{max(\sigma_{ext}\times E_{VAI},10^{-6})}$$ (Eq. 3.6.122)

$$\sigma_{ext} = \frac{\varphi_1+\varphi_2\times cos\theta}{cos\theta}\times\sqrt{1-\rho-\tau}$$ (Eq. 3.6.123)

where ρ is the vegetation reflectance and τ is the vegetation transmittance. θ is the solar zenith angle. $\varphi_1 + \varphi_2 \times cos\theta$ is the projected vegetation area in solar radiation. All of these are computed from the two-stream canopy radiative transfer scheme above

The shaded fraction of the canopy (f_{shd}) is computed as

$$f_{shd} = 1 - f_{sun}$$ (Eq. 3.6.124)

$$E_{LAI,sun} = E_{LAI}\times f_{sun}$$ (Eq. 3.6.125)

$$E_{LAI,shd} = E_{LAI}\times f_{shd}$$ (Eq. 3.6.126)

3.6.6 Surface Radiation

Description:

The purpose of this Noah-MP module is to compute surface (ground and vegetation) radiative fluxes (absorption and reflection).

Relevant code modules:

Module: SurfaceRadiationMod.F90

Subroutines: SurfaceRadiation

Physics:

Direct radiation absorbed by canopy, C_{ad} [W/m^2] is given by

$$C_{ad} = S_d \times f_{abd} \qquad \text{(Eq. 3.6.127)}$$

Diffuse radiation absorbed by canopy, C_{ai} [W/m^2] is given by

$$C_{ai} = S_i \times f_{abi} \qquad \text{(Eq. 3.6.128)}$$

Solar radiation absorbed by vegetation, $S_{abs,veg}$ [W/m^2] is given by

$$S_{abs,veg} = C_{ad} + C_{ai} \qquad \text{(Eq. 3.6.129)}$$

In the above equations, S_d and S_i [W/m^2] are the incoming direct and diffuse solar radiation, respectively. f_{abd} is radiation flux absorbed by the vegetation per unit direct flux, and f_{abi} is radiation flux absorbed by the vegetation per unit diffuse flux. Both are computed from the two-stream canopy radiative transfer scheme above.

Transmitted solar fluxes incident on ground is given by following equations:

$$T_d = S_d \times F_{dd} \qquad \text{(Eq. 3.6.130)}$$

$$T_i = S_d \times F_{id} + S_i \times F_{ii} \qquad \text{(Eq. 3.6.131)}$$

where T_d and T_i [W/m^2] are the transmitted direct and diffuse solar radiation, respectively. F_{dd} is the downward direct flux per unit direct flux below vegetation, F_{id} is the downward diffuse flux per unit direct flux below vegetation, and F_{ii} is the downward diffuse flux per unit diffuse flux below vegetation. All three of these quantities are computed from the two-stream canopy radiative transfer scheme above.

Solar radiation absorbed by ground, $S_{abs,grd}$ [W/m^2] is computed as,

$$S_{abs,grd} = T_d \times (1 - \alpha_{dir}) + T_i \times (1 - \alpha_{dfs}) \quad \text{(Eq. 3.6.132)}$$

The total solar radiation absorbed by surface, $S_{abs,sfc}$ [W/m^2] is computed as,

$$S_{abs,sfc} = S_{abs,veg} + S_{abs,grd} \quad \text{(Eq. 3.6.133)}$$

Partition visible canopy absorption to sunlit and shaded fractions to get average absorbed parameter for sunlit and shaded leaves.

$$f_{LAI} = \frac{E_{LAI}}{E_{VAI}} \quad \text{(Eq. 3.6.134)}$$

where f_{LAI} is the leaf area fraction of canopy, E_{LAI} and E_{VAI} are the effective leaf and vegetation area indices, respectively computed from previous sections.

Average absorbed photosynthetically active radiation ($S_{PAR,sun}$, [W/m^2]) for sunlit leaves is calculated as

$$S_{PAR,sun} = \begin{cases} (C_{ad} + C_{ai} \times f_{sun}) \times \frac{f_{LAI}}{E_{LAI,sun}} & f_{sun} > 0 \\ 0 & f_{sun} \leq 0 \end{cases} \quad \text{(Eq. 3.6.135)}$$

Average absorbed photosynthetically active radiation ($S_{PAR,shd}$, [W/m^2]) for shaded leaves is calculated as

$$S_{PAR,shd} = \begin{cases} C_{ai} \times f_{shd} \times \frac{f_{LAI}}{E_{LAI,shd}} & f_{sun} > 0 \\ (C_{ad} + C_{ai}) \times \frac{f_{LAI}}{E_{LAI,shd}} & f_{sun} \leq 0 \end{cases} \quad \text{(Eq. 3.6.136)}$$

Now, the reflected solar radiation (F_{sr}, [W/m^2]) will be computed as

$$R_{nir} = \alpha_{d,nir} \times S_{d,nir} + \alpha_{i,nir} \times S_{i,nir} \quad \text{(Eq. 3.6.137)}$$

$$R_{vis} = \alpha_{d,vis} \times S_{d,vis} + \alpha_{i,vis} \times S_{i,vis} \quad \text{(Eq. 3.6.138)}$$

$$F_{sr} = R_{nir} + R_{vis} \quad \text{(Eq. 3.6.139)}$$

where R_{nir} and R_{vis} are the total surface reflected near-infrared and visible radiations. Subscripts 'd' and 'i' represent direct and diffuse radiations, respectively. α is the surface albedo computed from the canopy radiative transfer scheme above.

Reflected solar radiation of vegetation (F_{srv}, [W/m^2]) and ground (F_{srg}, [W/m^2]) are:

$$F_{sr,v} = F_{d,rv,vis} \times S_{d,vis} + F_{i,rv,vis} \times S_{i,vis} + F_{d,rv,nir} \times S_{d,nir} + F_{i,rv,nir} \times S_{i,nir} \quad \text{(Eq. 3.6.140)}$$

$$F_{sr,g} = F_{d,rg,vis} \times S_{d,vis} + F_{i,rg,vis} \times S_{i,vis} + F_{d,rg,nir} \times S_{d,nir} + F_{i,rg,nir} \times S_{i,nir} \quad \text{(Eq. 3.6.141)}$$

where $F_{d/i,rv,vis/nir}$ and $F_{d/i,rg,vis/nir}$ are all computed from the canopy radiative transfer scheme above. Subscripts 'v' and 'g' are for vegetation and ground, respectively. Subscripts 'd' and 'i' represent direct and diffuse radiations, respectively. Subscript 'r' means reflected radiation.

3.6.7 Surface Longwave Emissivity

Description:

This part is to compute ground, vegetation, and total surface longwave emissivity.

Relevant code modules:

Module: SurfaceEmissivityMod.F90

Subroutines: SurfaceEmissivity

Physics:

1. Vegetation/canopy longwave emissivity (ε_{veg}):

$$\varepsilon_{veg} = 1.0 - e^{-\frac{E_{LAI}+E_{SAI}}{1.0}} \quad \text{(Eq. 3.6.142)}$$

where E_{LAI} and E_{SAI} are the effective leaf and steam area indices after accounting for snow bury, which are computed in Phenology section (PhenologyMainMod.F90).

2. Ground longwave emissivity (ε_{grd}):

If it is ice surface:

$$\varepsilon_{grd} = \varepsilon_{ice} \times (1 - f_{snow}) + \varepsilon_{snow} \times f_{snow} \quad \text{(Eq. 3.6.143)}$$

where f_{snow} is the snow cover fraction computed from SnowCoverGroundNiu07Mod.F90, ε_{ice} and ε_{snow} are the ice and snow emissivity, respectively, specified as table parameters in NoahmpTable.TBL.

If it is soil/lake surface:

$$\varepsilon_{grd} = \varepsilon_{soil/lake} \times (1 - f_{snow}) + \varepsilon_{snow} \times f_{snow} \quad \text{(Eq. 3.6.144)}$$

where $\varepsilon_{soil/lake}$ is the emissivity of soil/lake specified as table parameters in NoahmpTable.TBL.

3. Net surface longwave emissivity (ε_{sfc}):

$$\varepsilon_{sfc} = f_{veg} \times \left[\varepsilon_{grd} \times \left(1 - \varepsilon_{veg}\right) + \varepsilon_{veg} + \varepsilon_{veg} \times \left(1 - \varepsilon_{veg}\right) \times \left(1 - \varepsilon_{grd}\right)\right] + \left(1 - f_{veg}\right) \times \varepsilon_{grd}$$

(Eq. 3.6.145)

where $\varepsilon_{grd} \times \left(1 - \varepsilon_{veg}\right)$ indicates the ground longwave emission going through canopy, $\varepsilon_{veg} \times \left(1 - \varepsilon_{veg}\right) \times \left(1 - \varepsilon_{grd}\right)$ indicates the downward canopy emission that are reflected by ground and further go through canopy.

3.6.8 Soil Moisture Transpiration Factor

Description:

This part is to compute soil water (moisture) transpiration factor based on the soil hydraulics that will be used for stomata resistance and evapotranspiration calculations. This computation is only active for soil point/pixel. There are some recent studies developing new big-tree plant hydraulics (Li et al., 2021) and dynamic root optimization (Wang et al. 2018) with explicit representation of plant water storage (Niu et al., 2020), which will be integrated into the community Noah-MP version in the future.

Relevant code modules:

Module: SoilWaterTranspirationMod.F90

Subroutines: SoilWaterTranspiration

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSoilWaterTranspiration	1*	Noah (soil moisture)
	2	CLM (matric potential)
options for soil moisture factor for stomatal resistance & ET	3	SSiB (matric potential)

Physics:

Noah-MP uses a soil moisture factor (β_{tr}) as a linear constraint to stomatal resistance, and in turn transpiration. β_{tr} is computed as the average of the soil wetness factor (w), weighted by root ratio (r):

$$\beta_{tr} = \sum_{i=1}^{N_{root}} (r_i \times w_i) \qquad \text{(Eq. 3.6.146)}$$

$$r_i = \frac{D_{zsnso}(i)}{-Z_{soil}(N_{root})} \qquad \text{(Eq. 3.6.147)}$$

where N_{root} is the number of soil layers containing roots, and i is the soil layer, D_{zsnso} [m] is the soil layer thickness, and Z_{soil} [m] is the depth (negative) of the soil layer bottom. w is computed using three different soil hydraulics schemes: (a) Noah-type (OptSoilWaterTranspiration=1), (b) CLM-type (OptSoilWaterTranspiration=2), and (c) SSiB-type (OptSoilWaterTranspiration=3):

$$w_i = \begin{cases} \frac{W_{liq,soil}(i) - \theta_{soil,wilt}(i)}{\theta_{soil,ref}(i) - \theta_{soil,wilt}(i)} & (Noah) \\ \frac{\psi_{soil,wilt} - \psi_{soil}(i)}{\psi_{soil,wilt} + \psi_{soil,sat}(i)} & (CLM) \\ 1 - e^{-5.8 \times ln\left(\frac{\psi_{soil,wilt}}{\psi_{soil}(i)}\right)} & (SSiB) \end{cases} \qquad \text{(Eq. 3.6.148)}$$

$$w_i = \min(1.0, \max(0, w_i)) \qquad \text{(Eq. 3.6.149)}$$

$$\psi_{soil}(i) = \max(\psi_{soil,wilt}, \; -\psi_{soil,sat}(i) \times \max(0.01, \left[\frac{W_{liq,soil}(i)}{\theta_{soil,max}}\right]^{-B_{exp}(i)}) \qquad \text{(Eq. 3.6.150)}$$

where $W_{liq,soil}$ [m^3/m^3] is the soil liquid water content, $\theta_{soil,wilt}$ [m^3/m^3] is the soil moisture at wilting point (set in NoahmpTable.TBL), $\theta_{soil,ref}$ [m^3/m^3] is the reference soil moisture (set in NoahmpTable.TBL), $\theta_{soil,max}$ [m^3/m^3] is the saturated soil moisture (set in NoahmpTable.TBL), $\psi_{soil,wilt}$ [m] is the soil matric potential at wilting point (set in NoahmpTable.TBL), $\psi_{soil,sat}$ [m] is the saturated soil matric potential (set in NoahmpTable.TBL), ψ_{soil} [m] is the soil matric potential, and B_{exp} is the soil B exponential parameter (set in NoahmpTable.TBL).

Note that these traditional soil hydraulics schemes ignore the role of plant traits in controlling transpiration which contributes to significant uncertainty in simulated transpiration and terrestrial water storage from Noah-MP, particularly under water limited conditions (Niu et al., 2020; Li et al., 2021). Explicit treatment of plant hydraulics is needed in the future.

3.6.9 Soil Surface Resistance to Ground Evaporation

Description:

This part is to compute soil surface resistance to ground evaporation/sublimation. It represents the resistance imposed by the molecular diffusion in the soil surface (as opposed to aerodynamic resistance computed elsewhere in the model).

Relevant code modules:

Module: ResistanceGroundEvaporationMod.F90

Subroutines: ResistanceGroundEvaporation

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptGroundResistanceEvap options for ground resistance to evaporation/sublimation	1*	Sakaguchi and Zeng, 2009
	2	Sellers (1992)
	3	adjusted Sellers 1992 for wet soil
	4	option 1 for non-snow; rsurf = rsurf_snow for snow (set in table)

Physics:

The soil evaporation factor (B_{evap}) used for OptGroundResistanceEvap = 2 and 3 is computed as:

$$B_{evap} = \max(0, \frac{W_{liq,soil}(1)}{\theta_{soil,max}(1)}) \qquad \text{(Eq. 3.6.151)}$$

where $W_{liq,soil}$ [m^3/m^3] is the soil liquid water content and $\theta_{soil,max}$ [m^3/m^3] is the saturated soil moisture (set in NoahmpTable.TBL).

For lake points/pixels, the surface resistance to ground evaporation/sublimation ($R_{grd,evap}$, [s/m]) is set to 1.0, and the relative humidity in surface soil/snow air space ($RH_{grd,air}$) is set to 1.0.

For soil points/pixels,

(1) If OptGroundResistanceEvap = 1 (Sakaguchi and Zeng, 2009), the surface resistance to ground evaporation/sublimation ($R_{grd,evap}$, [s/m]) is computed as:

$$R_{grd,evap} = \frac{Z_{soil,dry}}{D_{vap,red}} \qquad \text{(Eq. 3.6.152)}$$

The dry soil thickness ($Z_{soil,dry}$, [m]) is computed as:

$$Z_{soil,dry} = -Z_{soil}(1) \times \frac{e^{\left[1-\min(1,\frac{W_{liq,soil}(1)}{\theta_{soil,max}(1)}\right]^{R_{s,exp}}}-1}{2.71828-1} \quad \text{(Eq. 3.6.153)}$$

The reduced vapor diffusivity ($D_{vap,red}$, [m^2/s]) is computed as:

$$D_{vap,red} = 2.2 \times 10^{-5} \times \theta_{soil,max}(1) \times \theta_{soil,max}(1) \times \left(1 - \frac{\theta_{soil,wilt}(1)}{\theta_{soil,max}(1)}\right)^{2+\frac{3}{B_{exp}(1)}} \quad \text{(Eq. 3.6.154)}$$

where $\theta_{soil,wilt}$ [m^3/m^3] is the soil moisture at wilting point (set in NoahmpTable.TBL), and B_{exp} is the soil B exponential parameter (set in NoahmpTable.TBL).

(2) If OptGroundResistanceEvap = 2 (Sellers (1992) original), the $R_{grd,evap}$ [s/m] is computed as:

$$R_{grd,evap} = f_{snow} \times 1.0 + (1 - f_{snow}) \times e^{8.25-4.225 \times B_{evap}} \quad \text{(Eq. 3.6.155)}$$

(3) If OptGroundResistanceEvap = 3 (Sellers (1992) adjusted to decrease $R_{grd,evap}$ for wet soil), the $R_{grd,evap}$ [s/m] is computed as:

$$R_{grd,evap} = f_{snow} \times 1.0 + (1 - f_{snow}) \times e^{8.25-6.0 \times B_{evap}} \quad \text{(Eq. 3.6.156)}$$

(4) If OptGroundResistanceEvap = 4 (Sakaguchi and Zeng, 2009 adjusted by f_{snow} weighting), the $R_{grd,evap0}$ [s/m] for non-snow part is computed the same as OptGroundResistanceEvap = 1:

$$R_{grd,evap0} = \frac{Z_{soil,dry}}{D_{vap,red}} \quad \text{(Eq. 3.6.157)}$$

Then, the total $R_{grd,evap}$ is further computed with snow cover weights as:

$$R_{grd,evap} = \frac{1}{f_{snow} \times \frac{1}{R_{sno,evap}} + (1-f_{snow}) \times \frac{1}{\max(0.001, R_{grd,evap0})}} \quad \text{(Eq. 3.6.158)}$$

where $R_{sno,evap}$ [s/m] is the snow surface resistance to ground sublimation (set in NoahmpTable.TBL).

For all OptGroundResistanceEvap schemes,

if $W_{liq,soil}(1) < 0.01$ and $h_{snow} = 0$, then $R_{grd,evap} = 10^6$.

if it is urban points/pixels and $h_{snow} = 0$, then $R_{grd,evap} = 10^6$.

The relative humidity in surface soil/snow air space ($RH_{grd,air}$) is computed as:

$$RH_{grd,air} = f_{snow} \times 1.0 + (1 - f_{snow}) \times e^{\frac{\psi_{soil}(1) \times g}{R_w \times T_g}} \quad \text{(Eq. 3.6.159)}$$

$$\psi_{soil}(1) = -\psi_{soil,sat}(1) \times \max(0.01, \left[\frac{W_{liq,soil}(1)}{\theta_{soil,max}}\right]^{-B_{exp}(1)} \quad \text{(Eq. 3.6.160)}$$

where $\psi_{soil,sat}$ [m] is the saturated soil matric potential (set in NoahmpTable.TBL), ψ_{soil} [m] is the soil matric potential, and B_{exp} is the soil B exponential parameter (set in NoahmpTable.TBL). g [m/s^2] is the gravity acceleration (defined in ConstantDefineMod.F90), R_w [J/kg/K] is the gas constant for water vapor (defined in ConstantDefineMod.F90), and T_g [K] is the ground temperature.

3.6.10 Psychrometric Variables for Canopy and Ground

Description:

This part is to compute psychrometric variables for canopy and ground.

Relevant code modules:

Module: PsychrometricVariableMod.F90

Subroutines: PsychrometricVariable

Physics:

1. For canopy, the latent heat of vaporization ($C_{LH,can,vap}$, [J/kg]) is:

$$C_{LH,can,vap} = \begin{cases} C_{LH,evap} & T_{can} > T_{frz} \\ C_{LH,subl} & T_{can} \leq T_{frz} \end{cases} \quad \text{(Eq. 3.6.161)}$$

where $C_{LH,evap}$ [J/kg] and $C_{LH,subl}$ [J/kg] are the specific latent heat of evaporation and sublimation, respectively (defined in ConstantDefineMod.F90). T_{can} [K] and T_{frz} [K] are the canopy and freezing temperatures, respectively.

The psychrometric constant for the canopy (γ_{can}, [Pa/K]) is:

$$\gamma_{can} = \frac{C_{p,air} \times P_{sfc}}{0.622 \times C_{LH,can,vap}} \quad \text{(Eq. 3.6.162)}$$

where $C_{p,air}$ [J/kg/K] is the air heat capacity (defined in ConstantDefineMod.F90) and P_{sfc} [Pa] is the surface air pressure.

2. For ground, the latent heat of vaporization ($C_{LH,grd,vap}$, [J/kg]) is:

$$C_{LH,grd,vap} = \begin{cases} C_{LH,evap} & T_{grd} > T_{frz} \\ C_{LH,subl} & T_{grd} \leq T_{frz} \end{cases} \quad \text{(Eq. 3.6.163)}$$

where T_{grd} [K] is the ground temperature.

The psychrometric constant for the ground (γ_{grd}, [Pa/K]) is:

$$\gamma_{grd} = \frac{C_{p,air} \times P_{sfc}}{0.622 \times C_{LH,grd,vap}} \quad \text{(Eq. 3.6.164)}$$

3.6.11 Surface Fluxes and Temperature of Vegetated Ground

Description:

This part is to compute surface energy fluxes and update temperature for the vegetated ground portion of the pixel. The Newton-Raphson iteration is used to solve for vegetation and ground temperature.

Relevant code modules:

Module: SurfaceEnergyFluxBareGroundMod.F90

Subroutines: SurfaceEnergyFluxBareGround

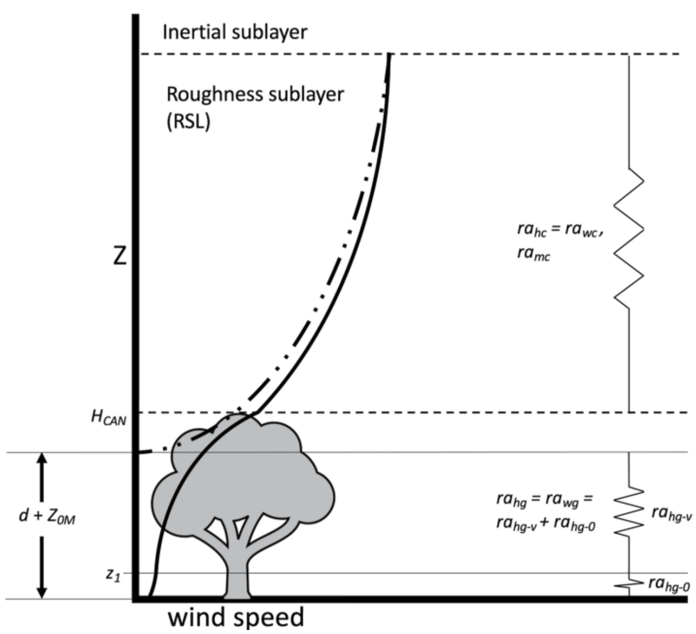


Figure 11. Conceptual example of the extrapolated Monin-Obukhov similarity theory (MOST) wind profile (dashed line) and an "observed" wind profile (solid line). The extrapolated MOST

profile becomes zero at the zero-plane displacement ($d_{0,sfc}$) plus roughness length for momentum (z_{0m}) height. Noah-MP assumes a similar profile as the solid line, which is characterized as a logarithmic profile above the canopy ($>d_{0,sfc} + z_{0m}$), an exponential profile within the canopy ($d_{0,sfc} + z_{0m} \sim z_1$), and again a logarithmic profile below the canopy ($< z_1$). Adapted from Abolafia-Rosenzweig et al. (2021).

Physics:

1. The overall energy balance (vegetated ground) is:

Canopy level:

$$S_{W,abs,can} + H_{pr,can} = L_{W,net,can} + H_{S,can} + H_{L,can} + H_{TR} + \Delta H_{s,can} \qquad \text{(Eq. 3.6.165)}$$

Below-canopy ground level:

$$S_{W,abs,grd} + H_{pr,veg,grd} = L_{W,net,grd} + H_{S,grd} + H_{L,grd} + G_H \qquad \text{(Eq. 3.6.166)}$$

where $S_{W,abs,can/grd}$ [W/m^2] and $L_{W,net,can/grd}$ [W/m^2] are the absorbed shortwave and net longwave radiative fluxes for canopy/ground, respectively. $H_{pr,veg,grd}$ [W/m^2] and $H_{pr,can}$ [W/m^2] are the net precipitation heat flux advected to below-canopy ground and canopy, respectively (computed in the PrecipitationHeatAdvecMod.F90). $H_{S,can/grd}$ [W/m^2] and $H_{L,can/grd}$ [W/m^2] are the canopy/ground sensible and latent heat fluxes (positive values indicate upward heat fluxes from the canopy/ground to the air), respectively. G_H [W/m^2] is the below-canopy ground heat flux (positive value indicates downward heat flux from the soil/snow surface into underlying snow/soil layers). H_{TR} [W/m^2] is the latent heat from plant transpiration. $\Delta H_{s,can}$ [W/m^2] is the canopy heat storage change.

Sensible heat (H_S, [W/m^2]) and latent heat (H_L, [W/m^2]) fluxes are computed based on the bulk transfer relationships following Garratt (1992):

$$H_S = \rho_a \times C_h \times C_p \times U \times (\theta_s - \theta_a) \qquad \text{(Eq. 3.6.167)}$$

$$H_L = \rho_a \times C_w \times C_{LH} \times U \times (q_s - q_a) \qquad \text{(Eq. 3.6.168)}$$

where ρ_a [kg/m^3] is the air density, $C_{p,air}$ [J/kg/K] is the air heat capacity, C_{LH} [J/kg] is the specific latent heat of water vaporization, and U [m/s] is the wind speed. θ_s [K] and θ_a [K] are the air potential temperatures at the surface and in the air, respectively. q_s [kg/kg] and q_a [kg/kg] are the specific humidity at the surface and in the air, respectively. C_h and C_w are the surface exchange coefficients for heat and moisture, respectively. Noah-MP assumes that C_w is equal to C_h (Chen and Zhang, 2009). Based on the Monin-Obukhov (M-O) similarity theory (Brutsaert, 1982), C_h is formulated as

$$C_h = \frac{\kappa^2}{\left[\ln\left(\frac{Z-d_0}{z_{0m}}\right)-\varphi_m\left(\frac{Z-d_0}{L_{MO}}\right)\right]\left[\ln\left(\frac{Z-d_0}{z_{0h}}\right)-\varphi_h\left(\frac{Z-d_0}{L_{MO}}\right)\right]} \quad \text{(Eq. 3.6.169)}$$

Where κ is the von Karman constant, Z is the height above the ground, d_0 [m] is the zero-displacement height, and L_{MO} [m] is the M-O length. φ_m and φ_h are the stability functions for momentum and heat transfer, respectively. z_{0m} [m] and z_{0h} [m] are the surface roughness length for momentum and heat, respectively.

2. Initialize and prepare key variables:

(1) Limit leaf area index to a reasonable range:

$$E_{VAI,l} = \min(6.0, E_{VAI}) \quad \text{(Eq. 3.6.170)}$$

$$E_{LAI,sun} = \min(6.0, LAI_{sun}) \quad \text{(Eq. 3.6.171)}$$

$$E_{LAI,shd} = \min(6.0, LAI_{shd}) \quad \text{(Eq. 3.6.172)}$$

where E_{VAI} [m^2/m^2] is the effective vegetation area index computed in the earlier section. LAI_{sun} [m^2/m^2] and LAI_{shd} [m^2/m^2] are sunlit and shaded leaf area index, respectively, computed in the canopy radiative transfer section.

(2) Saturation vapor pressure and derivative with respect to ground temperature are computed below.

Relevant code modules:

Module: VaporPressureSaturationMod.F90

Subroutines: VaporPressureSaturation

The saturation water vapor pressure ($e_{s,Tg}$, [Pa]) at ground temperature (T_g, [K]) is:

$$e_{s,Tg} = \begin{cases} e_{sw,Tg} & T_g > T_{frz} \\ e_{si,Tg} & T_g \le T_{frz} \end{cases} \quad \text{(Eq. 3.6.173)}$$

where $e_{sw,Tg}$ [Pa] and $e_{si,Tg}$ [Pa] are the saturation vapor pressure for water and ice, respectively, computed in VaporPressureSaturationMod.F90.

The derivative of saturation water vapor pressure ($\frac{d(e_{s,Tg})}{d(T_g)}$, [Pa/K]) at ground temperature is:

$$\frac{d(e_{s,Tg})}{d(T_g)} = \begin{cases} \frac{d(e_{sw,Tg})}{d(T_g)} & T_g > T_{frz} \\ \frac{d(e_{si,Tg})}{d(T_g)} & T_g \le T_{frz} \end{cases} \quad \text{(Eq. 3.6.174)}$$

where $\frac{d(e_{sw,Tg})}{d(T_g)}$ [Pa/K] and $\frac{d(e_{si,Tg})}{d(T_g)}$ [Pa/K] are the derivative of saturation water vapor pressure for water and ice, respectively, computed in VaporPressureSaturationMod.F90.

To compute the saturation vapor pressure and corresponding derivative w.r.t. temperature for both water and ice at certain temperature, the VaporPressureSaturation module uses polynomial empirical parameterizations:

$$e_{sw,T} = 100 \times (A_0 + T \times (A_1 + T \times (A_2 + T \times (A_3 + T \times (A_4 + T \times (A_5 + T \times A_6))))))$$

(Eq. 3.6.175)

$$e_{si,T} = 100 \times (B_0 + T \times (B_1 + T \times (B_2 + T \times (B_3 + T \times (B_4 + T \times (B_5 + T \times B_6))))))$$

(Eq. 3.6.176)

$$\frac{d(e_{sw,T})}{d(T)} = 100 \times (C_0 + T \times (C_1 + T \times (C_2 + T \times (C_3 + T \times (C_4 + T \times (C_5 + T \times C_6))))))$$

(Eq. 3.6.177)

$$\frac{d(e_{si,T})}{d(T)} = 100 \times (D_0 + T \times (D_1 + T \times (D_2 + T \times (D_3 + T \times (D_4 + T \times (D_5 + T \times D_6))))))$$

(Eq. 3.6.178)

where A_i, B_i, C_i, D_i are polynomial coefficients.

(3) The surface specific humidity ($q_{veg,sfc}$, [kg/kg]) is initially computed as:

$$q_{veg,sfc} = \frac{0.622 \times e_{air}}{P_{sfc} - 0.378 \times e_{air}} \quad \text{(Eq. 3.6.179)}$$

where e_{air} [Pa] is the surface air vapor pressure computed in the atmospheric forcing section.

(4) Canopy height (h_{can}, [m]): $h_{can} = h_{veg,top}$, where $h_{veg,top}$ [m] is the canopy top specified in NoahmpTable.TBL.

(5) Wind speed at the top of canopy layer (U_{can}, [m/s]) is computed based on a logarithm decay:

$$U_{can} = U_{ref} \times \frac{ln\left(\frac{h_{can} - d_{0,sfc} + z_{0m}}{z_{0m}}\right)}{ln\left(\frac{z_{ref}}{z_{0m}}\right)} \quad \text{(Eq. 3.6.180)}$$

where U_{ref} [m/s] is the wind speed at the reference height, z_{0m}, $d_{0,sfc}$, and z_{ref} [m] are the surface roughness length for momentum, surface zero displacement height, and reference height above the ground, respectively, computed in the ground roughness property section above. If $h_{can} \leq d_{0,sfc}$, the model will report error and stop.

(6) Prepare longwave radiation coefficients:

The coefficient for longwave radiation over vegetated ground ($C_{IR,veg}$) as a function of temperature is:

$$C_{IR,veg} = (2 - \varepsilon_{veg} \times (1 - \varepsilon_{grd})) \times \varepsilon_{veg} \times S_B \quad \text{(Eq. 3.6.181)}$$

where ε_{grd} and ε_{veg} are the ground and canopy longwave emissivity computed above and S_B is the Stefan-Boltzmann constant defined in ConstantDefineMod.F90.

The total canopy absorption of longwave radiation over vegetated ground ($A_{IR,veg}$) is:

$$A_{IR,veg} = -\varepsilon_{veg} \times \left(1 + \left(1 - \varepsilon_{veg}\right) \times \left(1 - \varepsilon_{grd}\right)\right) \times LW_{down} - \varepsilon_{veg} \times \varepsilon_{grd} \times S_B \times {T_g}^4$$

(Eq. 3.6.182)

where the first term of the right-hand side indicates the canopy absorbed longwave radiation of downward atmospheric longwave forcing and that passes through canopy and further reflected by the ground, and the second term indicates the canopy absorbed longwave radiation emitted by below-canopy ground.

3. Surface roughness length for heat flux:

The surface ($z_{0h,sfc}$, [m]) and below-canopy ground ($z_{0h,grd}$, [m]) roughness length for heat fluxes are:

$$z_{0h,sfc} = z_{0m,sfc} \quad \text{(Eq. 3.6.183)}$$

$$z_{0h,grd} = z_{0m,grd} \quad \text{(Eq. 3.6.184)}$$

where $z_{0m,sfc}$ and $z_{0m,grd}$ [m] are the surface and below-canopy ground roughness lengths for momentum computed in the ground roughness property section, which are the theoretical heights at which wind speed is zero. By default, Noah-MP assumes z_{0h} to be equal to z_{0m}. However, previous studies pointed out that z_{0m} and z_{0h} are different due to different mechanisms and resistances controlling momentum and heat transfer (Zilitinkevich, 1995; Chen and Zhang, 2009). Thus, an enhanced version of the z_{0h} formulation based on Zilitinkevich (1995) and Chen and Zhang (2009) is as follows (currently deactivated in Noah-MP) for non-1st iteration:

$$z_{0h} = z_{0m} \times \mathrm{e}^{(-\kappa \times C_{zil} \times \sqrt{R_e})} \qquad \text{(Eq. 3.6.185)}$$

$$C_{zil} = 10^{-0.4 \times h} \qquad \text{(Eq. 3.6.186)}$$

where R_e is the roughness Reynolds number, C_{zil} is the empirical parameter represented as a function of the canopy height (h, [m]).

4. Compute aerodynamic resistances and exchange coefficients between reference height and the $d_{0,sfc} + z_{0m,sfc}$ height level (above canopy):

Relevant code modules:

Module: ResistanceAboveCanopyMostMod.F90

Subroutines: ResistanceAboveCanopyMOST

Module: ResistanceAboveCanopyChen97Mod.F90

Subroutines: ResistanceAboveCanopyChen97

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSurfaceDrag options for surface layer drag/exchange coefficient	1*	Monin-Obukhov (M-O) Similarity Theory (MOST)
	2	original Noah (Chen et al. 1997)

Based on the surface roughness length (computed above) and some other quantities as input, the surface drag schemes: OptSurfaceDrag = 1 for the Monin-Obukhov (M-O) Similarity Theory (MOST) and OptSurfaceDrag = 2 for the Chen et al. 1997 scheme (original Noah), are used to compute aerodynamic resistance for sensible heat ($R_{h,can}$, [s/m]) and momentum ($R_{m,can}$, [s/m]) between canopy air and atmosphere, as well as the corresponding momentum drag coefficients ($C_{m,can}$) and sensible heat exchange coefficient ($C_{h,can}$). They are connected by:

$$R_{h,can} = \max(1, \frac{1}{C_{h,can} \times U_{ref}}) \qquad \text{(Eq. 3.6.187)}$$

$$R_{m,can} = \max(1, \frac{1}{C_{m,can} \times U_{ref}}) \qquad \text{(Eq. 3.6.188)}$$

where U_{ref} [m/s] is the wind speed at reference height (computed in the earlier section).

The aerodynamic resistance for water vapor ($R_{w,can}$, [s/m]) is assumed to be the same as sensible heat:

$$R_{w,can} = R_{h,can} \quad \text{(Eq. 3.6.189)}$$

The Monin-Obukhov (M-O) theory and Chen et al. (1997) schemes are introduced in detail below. Note that there is a recent study developing a unified turbulence parameterization throughout the canopy and roughness sublayer (Abolafia-Rosenzweig et al., 2021) in Noah-MP, which will be integrated into the community Noah-MP version in the future.

5. Compute aerodynamic resistances between the $d_{0,sfc} + z_{0m,sfc}$ and $z_{0m,grd}$ height level (below canopy) and leaf boundary layer:

Relevant code modules:

Module: ResistanceLeafToGroundMod.F90

Subroutines: ResistanceLeafToGround

(1) For below-canopy aerodynamic resistances:

The calculation of below-canopy aerodynamic resistances is based on the M-O similarity theory. The M-O stability parameter ($\zeta_{MO,grd}$) and M-O characteristic length ($L_{MO,grd}$) are both initialized to 0 at the 1st iteration. Starting from the 2nd iteration,

$$F_{MO,tmp} = \max(10^{-6}, \frac{\kappa \times g}{T_{air,can}} \times \frac{H_{tmp,grd}}{\rho_{air} \times C_{p,air}}) \quad \text{(Eq. 3.6.190)}$$

$$L_{MO,grd} = -\frac{{u_*}^3}{F_{MO,tmp}} \quad \text{(Eq. 3.6.191)}$$

$$\zeta_{MO,grd} = \min(1.0, \frac{d_{0,sfc} - z_{0m,grd}}{L_{MO,grd}}) \quad \text{(Eq. 3.6.192)}$$

where κ is the von Karman constant, g [m/s^2] is the gravity acceleration, $T_{air,can}$ [K] is the canopy air temperature, ρ_{air} [kg/m^3] is the air density, $C_{p,air}$ [J/kg/K] is the heat capacity of dry air, u_* [m/s] is the friction velocity computed from surface drag schemes (M-O or Chen et al. 1997 schemes), and $H_{tmp,grd}$ is a temporary heat flux variable updated in a later subsection in each iteration.

The M-O sensible heat stability correction factor ($\Psi_{h,MO,grd}$) at the current iteration is then computed as:

$$\Psi_{h,MO,grd} = \begin{cases} \left[1 - 15 \times \zeta_{MO,grd}\right]^{-0.25} & Z_{MO,grd} < 0 \\ 1 + 4.7 \times \zeta_{MO,grd} & Z_{MO,grd} \geq 0 \end{cases} \quad \text{(Eq. 3.6.193)}$$

For the 1st iteration, $\Psi_{h,MO,grd}$ is the value from the computation above. Starting from the 2nd iteration, $\Psi_{h,MO,grd}$ is the averaged value of the above calculated new $\Psi_{h,MO,grd}$ and the old $\Psi_{h,MO,grd}$ calculated from the previous iteration.

Then, the canopy wind extinction coefficient ($C_{wind,ext,can}$) is computed as:

$$C_{wind,ext,can} = \sqrt{C_{W,ext,can} \times E_{VAI,l} \times h_{can} \times \Psi_{h,MO,grd}} \quad \text{(Eq. 3.6.194)}$$

where $C_{W,ext,can}$ is the canopy wind absorption parameter dependent on vegetation types (specified in NoahmpTable.TBL). $E_{VAI,l}$ is the effective vegetation area index computed above. h_{can} is the canopy height computed above.

Then, a temporary below-canopy heat transfer aerodynamic resistance factor ($F_{RA,h,tmp}$) is computed based on a logarithm wind extinction:

$$F_{RA,h,tmp} = h_{can} \times \frac{e^{C_{wind,ext,can}}}{C_{wind,ext,can}} \times \left[e^{\frac{-C_{wind,ext,can} \times z_{0h,grd}}{h_{can}}} - e^{\frac{-C_{wind,ext,can} \times (z_{0h,sfc} + d_{0,sfc})}{h_{can}}} \right]$$

(Eq. 3.6.195)

Finally, the below-canopy (between the $d_{0,sfc} + z_{0m,sfc}$ and $z_{0m,grd}$ height level) aerodynamic resistances for momentum ($R_{m,grd}$, [s/m]) and sensible and latent heat ($R_{h,grd}$ and $R_{w,grd}$, [s/m]) are computed as:

$$R_{m,grd} = 0 \quad \text{(Eq. 3.6.196)}$$

$$R_{h,grd} = \frac{F_{RA,h,tmp}}{C_{h,turb}} \quad \text{(Eq. 3.6.197)}$$

$$R_{w,grd} = R_{h,grd} \quad \text{(Eq. 3.6.198)}$$

$$C_{h,turb} = \max(10^{-6}, \kappa \times u_* \times (h_{can} - d_{0,sfc})) \quad \text{(Eq. 3.6.199)}$$

where $C_{h,turb}$ is the turbulent transfer coefficient for sensible heat.

(2) For leaf boundary layer resistance ($R_{leaf,bdy}$, [s/m]):

$$R_{leaf,bdy} = \frac{50 \times C_{wind,ext,can}}{1 - e^{-C_{wind,ext,can}/2}} \times \sqrt{\frac{D_{leaf}}{U_{can}}} \quad \text{(Eq. 3.6.200)}$$

where D_{leaf} [m] is the characteristic leaf dimension (specified in NoahmpTable.TBL).

The $R_{leaf,bdy}$ is further limited to a reasonable range (5~50; hard-coded):

$$R_{leaf,bdy} = \min(\max(R_{leaf,bdy}, 5), 50) \quad \text{(Eq. 3.6.201)}$$

6. Compute saturated vapor pressure and derivative w.r.t. vegetation temperature:

Relevant code modules:

Module: VaporPressureSaturationMod.F90

Subroutines: VaporPressureSaturation

The saturation water vapor pressure ($e_{s,Tv}$, [Pa]) at vegetation temperature (T_v, [K]) is:

$$e_{s,Tv} = \begin{cases} e_{sw,Tv} & T_v > T_{frz} \\ e_{si,Tv} & T_v \le T_{frz} \end{cases} \quad \text{(Eq. 3.6.202)}$$

where $e_{sw,Tv}$ [Pa] and $e_{si,Tv}$ [Pa] are the saturation vapor pressure for water and ice, respectively, at vegetation temperature computed in VaporPressureSaturationMod.F90.

The derivative of saturation water vapor pressure ($\frac{d(e_{s,Tv})}{d(T_v)}$, [Pa/K]) at vegetation temperature is:

$$\frac{d(e_{s,Tv})}{d(T_v)} = \begin{cases} \frac{d(e_{sw,Tv})}{d(T_v)} & T_v > T_{frz} \\ \frac{d(e_{si,Tv})}{d(T_v)} & T_v \le T_{frz} \end{cases} \quad \text{(Eq. 3.6.203)}$$

where $\frac{d(e_{sw,Tv})}{d(T_v)}$ [Pa/K] and $\frac{d(e_{si,Tv})}{d(T_v)}$ [Pa/K] are the derivative of saturation water vapor pressure for water and ice, respectively, computed in VaporPressureSaturationMod.F90.

To compute the saturation vapor pressure and corresponding derivative w.r.t. temperature for both water and ice at certain temperature, the VaporPressureSaturation module uses polynomial empirical parameterizations:

$$e_{sw,T} = 100 \times (A_0 + T \times (A_1 + T \times (A_2 + T \times (A_3 + T \times (A_4 + T \times (A_5 + T \times A_6)))))) \quad \text{(Eq. 3.6.204)}$$

$$e_{si,T} = 100 \times (B_0 + T \times (B_1 + T \times (B_2 + T \times (B_3 + T \times (B_4 + T \times (B_5 + T \times B_6)))))) \quad \text{(Eq. 3.6.205)}$$

$$\frac{d(e_{sw,T})}{d(T)} = 100 \times (C_0 + T \times (C_1 + T \times (C_2 + T \times (C_3 + T \times (C_4 + T \times (C_5 + T \times C_6)))))) \quad \text{(Eq. 3.6.206)}$$

$$\frac{d(e_{si,T})}{d(T)} = 100 \times (D_0 + T \times (D_1 + T \times (D_2 + T \times (D_3 + T \times (D_4 + T \times (D_5 + T \times D_6)))))) \quad \text{(Eq. 3.6.207)}$$

where A_i, B_i, C_i, D_i are polynomial coefficients.

7. Compute leaf stomatal resistance:

Relevant code modules:

Module: ResistanceCanopyStomataBallBerryMod.F90

Subroutines: ResistanceCanopyStomataBallBerry

Module: ResistanceCanopyStomataJarvisMod.F90

Subroutines: ResistanceCanopyStomataJarvis

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptStomataResistance options for canopy stomatal resistance	1*	Ball-Berry scheme
	2	Jarvis scheme

There are two schemes to compute stomatal resistances, namely Ball-Berry (OptStomataResistance = 1) and Jarvis (OptStomataResistance = 2). Note that the stomatal resistance calculation is only done at the 1st iteration with the energy flux and temperature stability iteration loop. The scheme is called twice, one for sunlit condition and the other for shaded condition. The treatments of sunlit and shaded conditions are similar but with different input solar radiation conditions.

(1) The Ball-Berry stomatal resistance scheme (OptStomataResistance = 1):

The leaf stomatal resistance ($R_{stomata}$, [s/m]) is first initialized to its maximum value and the photosynthesis rate (P_{SN}, [umol CO_2/m^2/s]) is initialized to 0. This is because the calculation of these two quantities are only done for photosynthesis active radiation (S_{par} > 0 under sunlit or shaded conditions. Thus, for this initialization:

$$R_{stomata} = \frac{C_{F,stom}}{K_{leaf,min}} \quad \text{(Eq. 3.6.208)}$$

$$P_{SN} = 0 \quad \text{(Eq. 3.6.209)}$$

where $K_{leaf,min}$ [umol/m^2/s] is the minimum leaf conductance (specified in NoahmpTable.TBL) and $C_{F,stom}$ is a unit conversion factor defined to convert from [s m^2/umol] to [s/m]:

$$C_{F,stom} = \frac{P_{sfc} \times 10^6}{8.314 \times T_{sfc}} \quad \text{(Eq. 3.6.210)}$$

where P_{sfc} [Pa] and T_{sfc} [K] are the surface pressure and temperature, respectively.

For $S_{par} > 0$, first several intermediate quantities are computed:

The foliage nitrogen fraction ($f_{n,leaf}$) is:

$$f_{n,leaf} = \min(1, \frac{N_{leaf}}{\max(10^{-6}, N_{leaf,max})}) \quad \text{(Eq. 3.6.211)}$$

where N_{leaf} is the foliage nitrogen concentration and $N_{leaf,max}$ is the maximum foliage nitrogen concentration (specified in NoahmpTable.TBL).

The temperature deviation from freezing point ($T_{v,dif}$, [K]) is:

$$T_{v,dif} = T_v - T_{frz} \quad \text{(Eq. 3.6.212)}$$

The absorb photosynthetic photon flux (S_{PPF}, [umol photons/m^2/s]) is:

$$S_{PPF} = 4.6 \times S_{par} \quad \text{(Eq. 3.6.213)}$$

where S_{par} equals $S_{par,sun}$ for sunlit case and $S_{par,shd}$ for shaded case. $S_{par,sun}$ and $S_{par,shd}$ [W/m^2] are photosynthesis active radiation for sunlit and shaded leaves, respectively, computed in SurfaceRadiationMod.F90.

The electron transport (J_e, [umol CO_2/m^2/s]) is:

$$J_e = S_{PPF} \times Q_{E,25} \quad \text{(Eq. 3.6.214)}$$

where $Q_{E,25}$ [umol CO_2 / umol photon] is the photolysis quantum efficiency at 25 degC (specified in NoahmpTable.TBL).

The CO_2 Michaelis-Menten constant (K_C, [Pa]) is:

$$K_C = K_{C,25} \times K_{C,10,chg}^{\frac{T_C - 25}{10}} \quad \text{(Eq. 3.6.215)}$$

where $K_{C,25}$ [Pa] is the CO_2 Michaelis-Menten constant at 25 degC (specified in NoahmpTable.TBL), $K_{C,10,chg}$ is the change in CO_2 Michaelis-Menten constant for every 10-deg C temperature change (specified in NoahmpTable.TBL), and T_c [degC] is the vegetation (canopy) temperature in unit of degC.

The O_2 Michaelis-Menten constant (K_O, [Pa]) is:

$$K_O = K_{O,25} \times K_{O,10,chg}^{\frac{T_C - 25}{10}} \quad \text{(Eq. 3.6.216)}$$

where $K_{O,25}$ [Pa] is the O_2 Michaelis-Menten constant at 25 degC (specified in NoahmpTable.TBL), $K_{O,10,chg}$ is the change in O_2 Michaelis-Menten constant for every 10-deg C temperature change (specified in NoahmpTable.TBL).

The factor (F_{wc}, [Pa]) for Rubisco limited photosynthesis is:

$$F_{wc} = K_C \times (1 + \frac{P_{O2}}{K_O}) \qquad \text{(Eq. 3.6.217)}$$

where P_{O2} [Pa] is the atmospheric O_2 partial pressure.

The CO_2 compensation point ($P_{c,CO2}$, [Pa]) is:

$$P_{c,CO2} = 0.5 \times \frac{K_C}{K_O} \times P_{O2} \times 0.21 \qquad \text{(Eq. 3.6.218)}$$

The maximum rate of carbonylation ($V_{C,max}$, [umol CO_2/m^2/s]) is:

$$V_{C,max} = V_{C,max,25} \times f_{n,leaf} \times \beta_{tr} \times \frac{V_{C,max,10,chg}^{\frac{T_C-25}{10}}}{1+e^{\left[\frac{-2.2\times10^5+710\times(T_C+273.16)}{8.314\times(T_C+273.16)}\right]}} \qquad \text{(Eq. 3.6.219)}$$

where $V_{C,max,25}$ [umol CO_2/m^2/s] maximum rate of carbonylation at 25 degC (specified in NoahmpTable.TBL), $f_{n,leaf}$ is the foliage nitrogen fraction computed above, β_{tr} is the soil moisture factor computed in the earlier section, and $V_{C,max,10,chg}$ is the change in maximum rate of carbonylation for every 10-deg C temperature change (specified in NoahmpTable.TBL).

The first guess of the internal CO_2 partial pressure ($P_{CO2,init}$, [Pa]) is:

$$P_{CO2,init} = 0.7 \times P_{CO2} \times C_{3,psn} + 0.4 \times P_{CO2} \times (1 - C_{3,psn}) \qquad \text{(Eq. 3.6.220)}$$

where P_{CO2} [Pa] is the atmospheric CO_2 partial pressure, and $C_{3,psn}$ is the photosynthetic pathway (0 for C4 plant and 1 for C3 plant; specified in NoahmpTable.TBL).

The converted leaf boundary resistance ($R'_{leaf,bdy}$, [s m^2/umol]) is:

$$R'_{leaf,bdy} = \frac{R_{leaf,bdy}}{C_{F,stom}} \qquad \text{(Eq. 3.6.221)}$$

where $R_{leaf,bdy}$ [s/m] is the leaf boundary resistance computed in the earlier subsection, and $C_{F,stom}$ is the conversion factor defined above.

The constrained canopy air vapor pressure ($e_{can,cons}$) is (without constraint, the model may crash):

$$e_{can,cons} = \max(0.25 \times e_{s,Tv} \times C_{3,psn} + 0.4 \times e_{s,Tv} \times (1 - C_{3,psn}), \min(e_{can,air}, e_{s,Tv})) \qquad \text{(Eq. 3.6.222)}$$

where $e_{can,air}$ [Pa] is the canopy air vapor pressure, $e_{s,Tv}$ [Pa] is the saturated vapor pressure at vegetation temperature.

Then, iteratively calculate stomatal resistance and related quantities (currently 3 iterations):

The light limited photosynthesis (W_J, [umol CO_2/m^2/s]) is:

$$W_J = \frac{\max(P_{CO2,init} - P_{c,CO2},\ 0) \times J_e}{P_{CO2,init} + 2 \times P_{c,CO2}} \times C_{3,psn} + J_e \times (1 - C_{3,psn}) \qquad \text{(Eq. 3.6.223)}$$

The Rubisco limited photosynthesis (W_c, [umol CO_2/m^2/s]) is:

$$W_c = \frac{\max(P_{CO2,init} - P_{c,CO2},\ 0) \times V_{C,max}}{P_{CO2,init} + F_{wc}} \times C_{3,psn} + V_{C,max} \times (1 - C_{3,psn}) \qquad \text{(Eq. 3.6.224)}$$

The export limited photosynthesis (W_e, [umol CO_2/m^2/s]) is:

$$W_e = 0.5 \times V_{C,max} \times C_{3,psn} + 4000 \times V_{C,max} \times \frac{P_{CO2,init}}{P_{sfc}} \times (1 - C_{3,psn}) \qquad \text{(Eq. 3.6.225)}$$

Then, the photosynthesis rate (P_{SN}, [umol CO_2/m^2/s]) is

$$P_{SN} = \min(W_J, W_c, W_e) \times I_{GS} \qquad \text{(Eq. 3.6.226)}$$

where I_{GS} is the indicator for growing season (1 for growing season and 0 for not) computed in the phenology section.

The CO_2 pressure at leaf surface ($C_{CO2,leaf}$, [Pa]) is:

$$C_{CO2,leaf} = \max(P_{CO2} - 1.37 \times R'_{leaf,bdy} \times P_{sfc} \times P_{SN}, 10^{-6}) \qquad \text{(Eq. 3.6.227)}$$

To update leaf stomatal resistance, several intermediate quantities are computed:

$$A = \frac{F_{mp} \times P_{SN} \times P_{sfc} \times e_{can,cons}}{C_{CO2,leaf} \times e_{s,Tv}} + K_{leaf,min} \qquad \text{(Eq. 3.6.228)}$$

$$B = \left(\frac{F_{mp} \times P_{SN} \times P_{sfc}}{C_{CO2,leaf}} + K_{leaf,min}\right) \times R'_{leaf,bdy} - 1 \qquad \text{(Eq. 3.6.229)}$$

$$C = -R'_{leaf,bdy} \qquad \text{(Eq. 3.6.230)}$$

$$Q = \begin{cases} -0.5 \times \left[B + \sqrt{B^2 - 4 \times A \times C}\right] & B \geq 0 \\ -0.5 \times \left[B - \sqrt{B^2 - 4 \times A \times C}\right] & B < 0 \end{cases} \qquad \text{(Eq. 3.6.231)}$$

where P_{sfc} [Pa] is the surface pressure, $K_{leaf,min}$ [umol/m^2/s] is the minimum leaf conductance (specified in NoahmpTable.TBL), F_{mp} is the slope of conductance-to-photosynthesis relationship (specified in NoahmpTable.TBL).

Then, the leaf stomatal resistance ($R_{stomata}$, [s/m]) is updated as:

$$R_{stomata} = \max\left(\frac{Q}{A}, \frac{C}{Q}\right) \qquad \text{(Eq. 3.6.232)}$$

The initial CO_2 partial pressure is updated for the next iteration:

$$P_{CO2,init} = \max(0, C_{CO2,leaf} - P_{SN} \times P_{sfc} \times 1.65 \times R_{stomata}) \qquad \text{(Eq. 3.6.233)}$$

Finally, after the above iterations, the $R_{stomata}$ is converted from the unit of [s m^2/umol] back to the original unit [s/m] as:

$$R_{stomata} = R_{stomata} \times C_{F,stom} \qquad \text{(Eq. 3.6.234)}$$

$R_{stomata}$ represents $R_{stomata,sun}$ for sunlit case and $R_{stomata,shd}$ for shaded case.

The P_{SN} computed after the above iteration is the final photosynthesis rate, where P_{SN} represents $P_{SN,sun}$ for sunlit case and $P_{SN,shd}$ for shaded case.

(2) The Jarvis stomatal resistance scheme (OptStomataResistance = 2):

This Jarvis scheme (Jarvis, 1976; Chen et al., 1996) is to calculate canopy resistance which depends on incoming solar radiation air temperature, atmospheric water vapor pressure deficit at the lowest model level, and soil moisture (preferably unfrozen soil moisture rather than total).

First, the specific humidity ($q_{can,air}$, [kg/kg]) and mixing ratio ($w_{can,air}$, [kg/kg]) are computed as:

$$q_{can,air} = \frac{0.622 \times e_{can,air}}{P_{sfc} - 0.378 \times e_{can,air}} \qquad \text{(Eq. 3.6.235)}$$

$$w_{can,air} = \frac{q_{can,air}}{1 - q_{can,air}} \qquad \text{(Eq. 3.6.236)}$$

where $e_{can,air}$ [Pa] is the canopy air vapor pressure, P_{sfc} [Pa] is the surface pressure.

The contribution factor ($F_{rs,solar}$) of incoming solar radiation to stomatal resistance is:

$$F_{rs,solar} = \max\left(0.0001, \frac{\frac{2 \times S_{par}}{F_{rad,str}} + \frac{R_{s,min}}{R_{s,max}}}{1 + \frac{2 \times S_{par}}{F_{rad,str}}}\right) \qquad \text{(Eq. 3.6.237)}$$

where S_{par} equals $S_{par,sun}$ for sunlit case and $S_{par,shd}$ for shaded case. $S_{par,sun}$ and $S_{par,shd}$ [W/m^2] are photosynthesis active radiation for sunlit and shaded leaves, respectively, computed in SurfaceRadiationMod.F90. $F_{rad,str}$ is the radiation stress factor (specified in NoahmpTable.TBL). $R_{s,min}$ and $R_{s,max}$ [s/m] are the minimum and maximum leaf stomatal resistances, respectively (specified in NoahmpTable.TBL).

The contribution factor ($F_{rs,temp}$) of air temperature to stomatal resistance is:

$$F_{rs,temp} = \max\left(0.0001, 1 - 0.0016 \times \left(T_{opt} - T_v\right)^2\right) \qquad \text{(Eq. 3.6.238)}$$

where T_{opt} [K] is the optimum transpiration air temperature (specified in NoahmpTable.TBL) and T_v [K] is the vegetation (canopy) temperature.

The contribution factor ($F_{rs,vpd}$) of vapor pressure deficit to stomatal resistance is:

$$F_{rs,vpd} = \max(0.01, \frac{1}{1+F_{vpd,str}\times\max(0, w_{can,air,sat} - w_{can,air})}) \quad \text{(Eq. 3.6.239)}$$

where $F_{vpd,str}$ is the vapor pressure deficit stress factor (specified in NoahmpTable.TBL) and $w_{can,air,sat}$ [kg/kg] is the saturated mixing ratio computed as a function of temperature and pressure in HumiditySaturationMod.F90.

Finally, the total leaf stomatal resistance ($R_{stomata}$, [s/m]) is computed as:

$$R_{stomata} = \frac{R_{s,min}}{F_{rs,solar}\times F_{rs,temp}\times F_{rs,vpd}\times\beta_{tr}} \quad \text{(Eq. 3.6.240)}$$

where β_{tr} is the soil moisture factor computed in the earlier section. $R_{stomata}$ represents $R_{stomata,sun}$ for sunlit case and $R_{stomata,shd}$ for shaded case.

The photosynthesis rate is not computed by this Jarvis scheme in Noah-MP currently.

8. Prepare canopy heat flux coefficients:

(1) The coefficient for longwave radiation over vegetated ground ($C_{IR,veg}$) and the multiplier for longwave radiation over vegetated ground ($A_{IR,veg}$) are already prepared above.

(2) For sensible heat:

The sensible heat conductance ($C_{SH,cond,air}$, [m/s]) from canopy air to atmosphere at reference height is:

$$C_{SH,cond,air} = \frac{1}{R_{h,can}} \quad \text{(Eq. 3.6.241)}$$

where $R_{h,can}$ [s/m] is the aerodynamic resistance for sensible heat computed from the surface drag scheme above.

The sensible heat conductance ($C_{SH,cond,leaf}$, [m/s]) from leaf surface to canopy air is:

$$C_{SH,cond,leaf} = \frac{2\times E_{VAI,l}}{R_{leaf,bdy}} \quad \text{(Eq. 3.6.242)}$$

where $E_{VAI,l}$ is the effective vegetation area index computed above and $R_{leaf,bdy}$ [s/m] is the bulk leaf boundary resistance.

The sensible heat conductance ($C_{SH,cond,grd}$, [m/s]) from canopy air to below-canopy ground is:

$$C_{SH,cond,grd} = \frac{1}{R_{h,grd}} \quad \text{(Eq. 3.6.243)}$$

where $R_{h,grd}$ [s/m] is the aerodynamic resistance for below-canopy sensible heat exchange computed above.

Thus, the total sensible heat conductance ($C_{SH,cond,veg}$, [m/s]) related to vegetation energy flux is:

$$C_{SH,cond,veg} = C_{SH,cond,air} + C_{SH,cond,leaf} + C_{SH,cond,grd} \quad \text{(Eq. 3.6.244)}$$

Then two temporary canopy air temperature factors related to sensible heat ($T_{SH,can,air}$ and $F_{SH,can,air}$, [K]) are defined:

$$T_{SH,can,air} = \frac{C_{SH,cond,air} \times T_{sfc} + C_{SH,cond,grd} \times T_g}{C_{SH,cond,veg}} \quad \text{(Eq. 3.6.245)}$$

$$F_{SH,can,air} = \frac{C_{SH,cond,leaf}}{C_{SH,cond,veg}} \quad \text{(Eq. 3.6.246)}$$

The coefficient ($C_{SH,can}$) for canopy sensible heat flux is:

$$C_{SH,can} = (1 - F_{SH,can,air}) \times \rho_{air} \times C_{p,air} \times C_{SH,cond,leaf} \quad \text{(Eq. 3.6.247)}$$

where ρ_{air} [kg/m^3] is the air density and $C_{p,air}$ [J/kg/K] is the heat capacity of dry air.

(3) For latent heat:

The latent heat conductance ($C_{LH,cond,air}$, [m/s]) from canopy air to atmosphere at reference height is:

$$C_{LH,cond,air} = \frac{1}{R_{w,can}} \quad \text{(Eq. 3.6.248)}$$

where $R_{w,can}$ [s/m] is the aerodynamic resistance for latent heat computed from the surface drag scheme above.

The latent heat conductance ($C_{LH,cond,leaf}$, [m/s]) from wet leaf surface to canopy air is:

$$C_{LH,cond,leaf} = \frac{f_{wet} \times E_{VAI,l}}{R_{leaf,bdy}} \quad \text{(Eq. 3.6.249)}$$

Where f_{wet} is the wetted fraction of canopy computed in the phenology section. $E_{VAI,l}$ is the effective vegetation area index computed above and $R_{leaf,bdy}$ [s/m] is the bulk leaf boundary resistance.

The latent heat conductance ($C_{LH,cond,tr}$, [m/s]) due to plant transpiration is:

$$C_{LH,cond,tr} = (1 - f_{wet}) \times \left[\frac{E_{LAI,sun}}{R_{leaf,bdy} + R_{stomata,sun}} + \frac{E_{LAI,shd}}{R_{leaf,bdy} + R_{stomata,shd}}\right] \quad \text{(Eq. 3.6.250)}$$

where $E_{LAI,sun}$ and $E_{LAI,shd}$ are the effective sunlit and shaded leaf area index computed above, respectively. $R_{stomata,sun}$ and $R_{stomata,shd}$ [s/m] are the sunlit and shaded leaf stomatal resistance, respectively.

The latent heat conductance ($C_{LH,cond,grd}$, [m/s]) from canopy air to below-canopy ground is:

$$C_{LH,cond,grd} = \frac{1}{R_{w,grd} + R_{grd,evap}} \quad \text{(Eq. 3.6.251)}$$

where $R_{w,grd}$ [s/m] is the aerodynamic resistance for below-canopy latent heat exchange computed above. $R_{grd,evap}$ [s/m] is the surface resistance to ground evaporation/sublimation computed in ResistanceGroundEvaporationMod.F90.

Thus, the total latent heat conductance ($C_{LH,cond,veg}$, [m/s]) related to vegetation energy flux is:

$$C_{LH,cond,veg} = C_{LH,cond,air} + C_{LH,cond,leaf} + C_{LH,cond,grd} + C_{LH,cond,tr} \quad \text{(Eq. 3.6.252)}$$

Then two temporary canopy vapor pressure factors related to latent heat ($e_{LH,can,air}$ and $F_{LH,can,air}$, [K]) are defined:

$$e_{LH,can,air} = \frac{C_{LH,cond,air} \times e_{air} + C_{LH,cond,grd} \times e_{s,Tg}}{C_{LH,cond,veg}} \quad \text{(Eq. 3.6.253)}$$

$$F_{LH,can,air} = \frac{C_{LH,cond,leaf} + C_{LH,cond,tr}}{C_{LH,cond,veg}} \quad \text{(Eq. 3.6.254)}$$

where e_{air} [Pa] is the surface air vapor pressure computed in the atmospheric forcing section and $e_{s,Tg}$ [Pa] is the saturation water vapor pressure at ground temperature (T_g, [K]).

The coefficients for canopy evaporative ($C_{LH,can,evap}$) and transpiration ($C_{LH,can,tr}$) latent heat fluxes are:

$$C_{LH,can,evap} = (1 - F_{LH,can,air}) \times C_{LH,cond,leaf} \times \frac{\rho_{air} \times C_{p,air}}{\gamma_{can}} \quad \text{(Eq. 3.6.255)}$$

$$C_{LH,can,tr} = (1 - F_{LH,can,air}) \times C_{LH,cond,tr} \times \frac{\rho_{air} \times C_{p,air}}{\gamma_{can}} \quad \text{(Eq. 3.6.256)}$$

where ρ_{air} [kg/m^3] is the air density and $C_{p,air}$ [J/kg/K] is the heat capacity of dry air. γ_{can} [Pa/K] is the psychrometric constant for the ground computed in PsychrometricVariableMod.F90.

9. Compute canopy fluxes and temperature change:

(1) The canopy air temperature ($T_{can,air}$, [K]) and vapor pressure ($e_{can,air}$, [Pa]) are computed as:

$$T_{can,air} = T_{SH,can,air} + F_{SH,can,air} \times T_v \quad \text{(Eq. 3.6.257)}$$

$$e_{can,air} = e_{LH,can,air} + F_{LH,can,air} \times e_{s,Tv} \quad \text{(Eq. 3.6.258)}$$

where $e_{s,Tv}$ [Pa] is the saturation water vapor pressure at vegetation temperature (T_v, [K]).

(2) The net canopy longwave radiation ($L_{net,can}$, [W/m^2]; positive for upward direction) is

$$L_{net,can} = f_{veg} \times \left[A_{IR,veg} + C_{IR,veg} \times {T_v}^4\right] \quad \text{(Eq. 3.6.259)}$$

where f_{veg} is the vegetation fraction. $A_{IR,veg}$ and $C_{IR,veg}$ are the absorption and coefficient for vegetation longwave radiation computed above.

(3) The canopy sensible heat flux ($H_{S,can}$, [W/m^2], positive for upward direction) is computed:

$$H_{S,can} = f_{veg} \times \rho_{air} \times C_{p,air} \times C_{SH,cond,leaf} \times (T_v - T_{can,air}) \quad \text{(Eq. 3.6.260)}$$

(4) The canopy evaporation latent heat ($H_{L,can,evap}$, [W/m^2], positive for upward direction) is:

$$H_{L,can,evap} = f_{veg} \times \frac{\rho_{air} \times C_{p,air}}{\gamma_{can}} \times C_{LH,cond,leaf} \times (e_{s,Tv} - e_{can,air}) \quad \text{(Eq. 3.6.261)}$$

To further constrain leaf evaporation:

$$H_{L,can,evap} = \begin{cases} \min\left(H_{L,can,evap}, W_{liq,can} \times \frac{C_{LH,can,vap}}{\Delta t}\right) & T_v > T_{frz} \\ \min\left(H_{L,can,evap}, W_{ice,can} \times \frac{C_{LH,can,vap}}{\Delta t}\right) & T_v \le T_{frz} \end{cases} \quad \text{(Eq. 3.6.262)}$$

where $W_{liq,can}$ [mm] and $W_{ice,can}$ [mm] are the canopy liquid and ice water content, respectively. Δt [s] is the model time step. $C_{LH,can,vap}$ [J/kg] is the latent heat of vaporization for canopy computed in PsychrometricVariableMod.F90.

(5) The canopy transpiration latent heat ($LH_{can,tr}$, [W/m^2], positive for upward direction) is:

$$H_{L,can,tr} = f_{veg} \times \frac{\rho_{air} \times C_{p,air}}{\gamma_{can}} \times C_{LH,cond,tr} \times (e_{s,Tv} - e_{can,air}) \quad \text{(Eq. 3.6.263)}$$

(6) The canopy heat storage change ($\Delta H_{s,can}$, [W/m^2]) is computed through canopy heat capacity ($C_{p,can}$, [J/m^2/K]):

$$C_{p,can} = C_{biom} \times E_{VAI,l} \times C_{p,water} + \frac{W_{liq,can}}{\rho_{water}} \times C_{p,water} + \frac{W_{ice,can}}{\rho_{ice}} \times C_{p,ice} \quad \text{(Eq. 3.6.264)}$$

where C_{biom} [m] is the canopy biomass heat capacity factor (set in NoahmpTable.TBL), $E_{VAI,l}$ is the effective vegetation area index, $C_{p,water}$ [J/m^3/K] and $C_{p,ice}$ [J/m^3/K] are the specific heat capacity of water and ice, respectively.

(7) The change of canopy temperature (ΔT_v, [K]) due to energy balance is computed as:

$$\Delta T_v = \frac{S_{can,abs} - L_{net,can} - H_{S,can} - H_{L,can,evap} - H_{L,can,tr} + H_{pr,can}}{f_{veg} \times \left[4 \times C_{IR,veg} \times {T_v}^3 + C_{SH,can} + (C_{LH,can,evap} + C_{LH,can,tr}) \times \frac{d(e_{s,Tv})}{d(T_v)}\right] + \frac{C_{p,can}}{\Delta t}}$$

(Eq. 3.6.265)

where $S_{can,abs}$ [W/m^2] is the total canopy absorbed shortwave radiation computed in the radiation section. $H_{pr,can}$ [W/m^2] is the precipitation heat flux advected to canopy computed in the PrecipitationHeatAdvectMod.F90. $\frac{d(e_{s,Tv})}{d(T_v)}$ [Pa/K] is derivative of saturation water vapor pressure at vegetation temperature computed above.

(8) Update different canopy heat fluxes and vegetation temperature:

$$L_{net,can} = L_{net,can} + f_{veg} \times 4 \times C_{IR,veg} \times {T_v}^3 \times \Delta T_v \quad \text{(Eq. 3.6.266)}$$

$$H_{S,can} = H_{S,can} + f_{veg} \times C_{SH,can} \times \Delta T_v \quad \text{(Eq. 3.6.267)}$$

$$H_{L,can,evap} = H_{L,can,evap} + f_{veg} \times C_{LH,can,evap} \times \frac{d(e_{s,Tv})}{d(T_v)} \times \Delta T_v \quad \text{(Eq. 3.6.268)}$$

$$H_{L,can,tr} = H_{L,can,tr} + f_{veg} \times C_{LH,can,tr} \times \frac{d(e_{s,Tv})}{d(T_v)} \times \Delta T_v \quad \text{(Eq. 3.6.269)}$$

$$\Delta H_{s,can} = \frac{C_{p,can}}{\Delta t} \times \Delta T_v \quad \text{(Eq. 3.6.270)}$$

$$T_v = T_v + \Delta T_v \quad \text{(Eq. 3.6.271)}$$

$$H_{tmp} = \rho_{air} \times C_{p,air} \times \frac{(T_{can,air} - T_{sfc})}{R_{h,can}} \quad \text{(Eq. 3.6.272)}$$

$$H_{tmp,grd} = \rho_{air} \times C_{p,air} \times \frac{(T_g - T_{can,air})}{R_{h,grd}} \quad \text{(Eq. 3.6.273)}$$

where H_{tmp} and $H_{tmp,grd}$ are temporary heat flux variables used by the M-O theory calculation above (ResistanceAboveCanopyMostMod.F90) and below canopy (ResistanceLeafToGroundMod.F90) and in each iteration.

Then, the canopy air specific humidity ($q_{can,air}$, [kg/kg]) is then updated as:

$$q_{can,air} = \frac{0.622 \times e_{can,air}}{P_{sfc} - 0.378 \times e_{s,Tg} \times e_{can,air}} \quad \text{(Eq. 3.6.274)}$$

10. Iterations over the calculations for the above steps 3-9 to achieve convergence of canopy temperature:

Currently, Noah-MP does 20 iterations (hard-coded) or until $|\Delta T_v| \leq 0.01$ to go through the above 3^{rd}-9^{th} steps: 3. Surface roughness length for heat flux; 4. Compute aerodynamic resistances and exchange coefficients between reference height and the $d_{0,sfc} + z_{0m,sfc}$ height level (above canopy); 5. Compute aerodynamic resistances between the $d_{0,sfc} + z_{0m,sfc}$ and $z_{0m,grd}$ height level (below canopy) and leaf boundary layer; 6. Compute saturated vapor pressure and derivative w.r.t. vegetation temperature; 7. Compute leaf stomatal resistance; 8. Prepare canopy heat flux coefficients; 9. Compute canopy fluxes and temperature change.

11. Prepare below-canopy heat flux coefficients:

The ground absorption of longwave radiation ($A_{IR,grd}$) is computed as:

$$A_{IR,grd} = -\varepsilon_{grd} \times (1 - \varepsilon_{veg}) \times L_{W,down} - \varepsilon_{grd} \times \varepsilon_{veg} \times S_B \times {T_v}^4 \quad \text{(Eq. 3.6.275)}$$

where the first term of the right-hand side indicates the ground absorbed longwave radiation of downward atmospheric longwave forcing ($L_{W,down}$, [W/m^2]) that passes through canopy, and the second term indicates the ground absorbed longwave radiation emitted by canopy.

The coefficient for longwave radiation for below-canopy ground ($C_{IR,grd}$) is:

$$C_{IR,grd} = \varepsilon_{grd} \times S_B \quad \text{(Eq. 3.6.276)}$$

where ε_{grd} and ε_{veg} are the ground and canopy longwave emissivity computed above, respectively, and S_B is the Stefan-Boltzmann constant defined in ConstantDefineMod.F90.

The coefficient for sensible heat ($C_{SH,grd}$) is:

$$C_{SH,grd} = \frac{\rho_{air} \times C_{p,air}}{R_{h,grd}} \quad \text{(Eq. 3.6.277)}$$

The coefficient for latent heat ($C_{LH,grd}$) is:

$$C_{LH,grd} = \frac{\rho_{air} \times C_{p,air}}{\gamma_{grd} \times (R_{w,grd} + R_{grd,evap})} \quad \text{(Eq. 3.6.278)}$$

The coefficient for ground heat diffusion flux ($C_{GH,grd}$) is:

$$C_{GH,grd} = 2 \times \frac{K_{heat,snso}(I_{n,sno}+1)}{D_{zsnso}(I_{n,sno}+1)} \quad \text{(Eq. 3.6.279)}$$

where $K_{heat,snso}$ [W/m/K] is the soil/snow thermal conductivity and D_{zsnso} [m] is the soil/snow layer thickness. $I_{n,sno}$ is the snow layer indicator defined as (see detail in snow hydrology section below):

$$I_{n,sno} = \begin{cases} -3, & 3\ layer\ snow \\ -2, & 2\ layer\ snow \\ -1, & 1\ layer\ snow \\ 0, & 0\ layer\ snow \end{cases} \quad \text{(Eq. 3.6.280)}$$

12. Compute saturated vapor pressure and derivative w.r.t. below-canopy ground temperature:

Relevant code modules:

Module: VaporPressureSaturationMod.F90

Subroutines: VaporPressureSaturation

The saturation water vapor pressure ($e_{s,Tg}$, [Pa]) at ground temperature (T_g, [K]) is:

$$e_{s,Tg} = \begin{cases} e_{sw,Tg} & T_g > T_{frz} \\ e_{si,Tg} & T_g \le T_{frz} \end{cases} \quad \text{(Eq. 3.6.281)}$$

where $e_{sw,Tg}$ [Pa] and $e_{si,Tg}$ [Pa] are the saturation vapor pressure for water and ice, respectively, computed in VaporPressureSaturationMod.F90.

The derivative of saturation water vapor pressure ($\frac{d(e_{s,Tg})}{d(T_g)}$, [Pa/K]) at ground temperature is:

$$\frac{d(e_{s,Tg})}{d(T_g)} = \begin{cases} \frac{d(e_{sw,Tg})}{d(T_g)} & T_g > T_{frz} \\ \frac{d(e_{si,Tg})}{d(T_g)} & T_g \leq T_{frz} \end{cases} \qquad \text{(Eq. 3.6.282)}$$

where $\frac{d(e_{sw,Tg})}{d(T_g)}$ [Pa/K] and $\frac{d(e_{si,Tg})}{d(T_g)}$ [Pa/K] are the derivative of saturation water vapor pressure for water and ice, respectively, computed in VaporPressureSaturationMod.F90.

To compute the saturation vapor pressure and corresponding derivative w.r.t. temperature for both water and ice at certain temperature, the VaporPressureSaturation module uses polynomial empirical parameterizations:

$$e_{sw,T} = 100 \times (A_0 + T \times (A_1 + T \times (A_2 + T \times (A_3 + T \times (A_4 + T \times (A_5 + T \times A_6))))))$$

(Eq. 3.6.283)

$$e_{si,T} = 100 \times (B_0 + T \times (B_1 + T \times (B_2 + T \times (B_3 + T \times (B_4 + T \times (B_5 + T \times B_6))))))$$

(Eq. 3.6.284)

$$\frac{d(e_{sw,T})}{d(T)} = 100 \times (C_0 + T \times (C_1 + T \times (C_2 + T \times (C_3 + T \times (C_4 + T \times (C_5 + T \times C_6))))))$$

(Eq. 3.6.285)

$$\frac{d(e_{si,T})}{d(T)} = 100 \times (D_0 + T \times (D_1 + T \times (D_2 + T \times (D_3 + T \times (D_4 + T \times (D_5 + T \times D_6))))))$$

(Eq. 3.6.286)

where A_i, B_i, C_i, D_i are polynomial coefficients.

13. Compute below-canopy ground fluxes and temperature change:

(1) The net below-canopy ground longwave radiation ($L_{net,grd}$, [W/m^2]; positive for upward direction) is

$$L_{net,grd} = C_{IR,grd} \times {T_g}^4 + A_{IR,grd} \qquad \text{(Eq. 3.6.287)}$$

(2) The below-canopy ground sensible heat flux ($H_{S,grd}$, [W/m^2], positive for upward direction) is computed as:

$$H_{S,grd} = C_{SH,grd} \times (T_g - T_{can,air}) \qquad \text{(Eq. 3.6.288)}$$

(3) The below-canopy ground latent heat flux ($H_{L,grd}$, [W/m^2], positive for upward direction) is computed as:

$$H_{L,grd} = C_{LH,grd} \times (e_{s,Tg} \times RH - e_{can,air}) \quad \text{(Eq. 3.6.289)}$$

where $e_{can,air}$ [Pa] is the canopy air vapor pressure and RH is the surface relative humidity.

(4) The ground heat flux ($G_{H,grd}$, [W/m^2], positive for downward direction) is computed as:

$$G_{H,grd} = C_{GH,grd} \times (T_g - T_{snso}(I_{n,sno} + 1)) \quad \text{(Eq. 3.6.290)}$$

where $T_{snso}(I_{n,sno} + 1)$ is the first or top snow/soil layer temperature.

(5) The change of below-canopy ground temperature (ΔT_g, [K]) due to energy balance is computed as:

$$\Delta T_g = \frac{S_{grd,abs} - L_{net,grd} - H_{S,grd} - H_{L,grd} - G_{H,grd} + H_{pr,veg,grd}}{4 \times C_{IR,grd} \times {T_g}^3 + C_{SH,grd} + C_{LH,grd} \times \frac{d(e_{s,Tg})}{d(T_g)} + C_{GH,grd}} \quad \text{(Eq. 3.6.291)}$$

where $S_{grd,abs}$ [W/m^2] is the below-canopy ground absorbed shortwave radiation as computed in the radiation section. $H_{pr,veg,grd}$ [W/m^2] is the precipitation heat flux advected from canopy to below-canopy ground.

(6) Update different heat fluxes and ground temperature:

$$L_{net,grd} = L_{net,grd} + 4 \times C_{IR,grd} \times {T_g}^3 \times \Delta T_g \quad \text{(Eq. 3.6.292)}$$

$$H_{S,grd} = H_{S,grd} + C_{SH,grd} \times \Delta T_g \quad \text{(Eq. 3.6.293)}$$

$$H_{L,grd} = H_{L,grd} + C_{LH,grd} \times \frac{d(e_{s,Tg})}{d(T_g)} \times \Delta T_g \quad \text{(Eq. 3.6.294)}$$

$$G_{H,grd} = G_{H,grd} + C_{GH,grd} \times \Delta T_g \quad \text{(Eq. 3.6.295)}$$

$$T_g = T_g + \Delta T_g \quad \text{(Eq. 3.6.296)}$$

14. Iterations over steps 12-13 to achieve convergence of ground temperature:

Currently, Noah-MP does 5 iterations (hard-coded) to go through the above 12th-13th steps: 12. Compute saturated vapor pressure and derivative w.r.t. below-canopy ground temperature, 13. Compute below-canopy ground fluxes and temperature change.

15. Adjust below-canopy ground temperature based on snow conditions:

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSnowSoilTempTime	1*	semi-implicit; flux top boundary condition

options for snow/soil temperature time scheme (only layer 1)	2	full implicit (original Noah); temperature top boundary condition
	3	same as 1, but snow cover for skin temperature calculation (generally improves snow)

For OptSnowSoilTempTime = 1 or 3, if there is snowpack ($h_{snow} > 0.05\,m$) and ground temperature is above freezing point ($T_g > T_{frz}$), then:

$$T_g = \begin{cases} T_{frz} & \text{OptSnowSoilTempTime} = 1 \\ T_{frz} \times f_{snow} + T_g \times (1 - f_{snow}) & \text{OptSnowSoilTempTime} = 3 \end{cases} \quad \text{(Eq. 3.6.297)}$$

Then the heat fluxes are re-evaluated based on the adjusted ground temperature as:

$$L_{net,grd} = C_{IR,grd} \times {T_g}^4 + A_{IR,grd} \quad \text{(Eq. 3.6.298)}$$

$$H_{S,grd} = C_{SH,grd} \times (T_g - T_{can,air}) \quad \text{(Eq. 3.6.299)}$$

$$H_{L,grd} = C_{LH,grd} \times (e_{s,Tg} \times RH - e_{can,air}) \quad \text{(Eq. 3.6.300)}$$

$$G_{H,grd} = S_{grd,abs} + H_{pr,veg,grd} - L_{net,grd} - H_{S,grd} - H_{L,grd} \quad \text{(Eq. 3.6.301)}$$

16. Compute above-canopy wind stresses:

The wind stresses ($\tau_{x/y,veg}$, [N/m^2]) at x- (east-west) and y- (north-south) directions are:

$$\tau_{x,veg} = -\rho_{air} \times C_{m,can} \times U_{ref} \times U_x \quad \text{(Eq. 3.6.302)}$$

$$\tau_{y,veg} = -\rho_{air} \times C_{m,can} \times U_{ref} \times U_y \quad \text{(Eq. 3.6.303)}$$

where ρ_{air} [kg/m^3] is the air density, $C_{m,can}$ is the above-canopy momentum drag coefficient, U_{ref} [m/s] is the wind speed at reference height, and $U_{x/y}$ [m/s] is the wind speed at x- or y-direction at reference height.

17. Compute 2-m air temperature for vegetated ground as diagnosis:

For both OptSurfaceDrag = 1 and 2, the 2-m sensible heat exchange coefficient/conductance ($C_{h,2m,veg}$, [m/s]) for vegetated ground is computed as:

$$C_{h,2m,veg} = \frac{u_* \times \kappa}{ln\left(\frac{z_{0h,sfc}+2}{z_{0h,sfc}}\right) - \Psi_{h,2m}} \quad \text{(Eq. 3.6.304)}$$

where u_* [m/s] is the friction velocity computed from surface drag schemes (M-O or Chen et al. 1997 schemes). κ is the von Karman constant, $z_{0h,sfc}$ [m] is the surface roughness length for heat flux, and $\Psi_{h,2m}$ is the 2-m sensible heat stability correction factor (computed in M-O scheme or 0 for Chen et al. 1997 scheme in this equation). See below for details of M-O and Chen et al. 1997 schemes.

The latent heat exchange coefficient/conductance is assumed to be the same as $C_{h,2m,veg}$.

If $C_{h,2m,veg} < 10^{-5}$, the 2-m temperature ($T_{2m,veg}$, [K]) and specific humidity ($q_{2m,veg}$, [kg/kg]) are assumed to the same as the canopy air values:

$$T_{2m,veg} = T_{can,air} \quad \text{(Eq. 3.6.305)}$$

$$q_{2m,veg} = q_{can,air} \quad \text{(Eq. 3.6.306)}$$

If $C_{h,2m,veg} \geq 10^{-5}$, the 2-m temperature and specific humidity are computed as:

$$T_{2m,veg} = T_{can,air} - \frac{H_{S,grd} + \frac{H_{S,can}}{fveg}}{\rho_{air} \times C_{p,air}} \times \frac{1}{C_{h,2m,veg}} \quad \text{(Eq. 3.6.307)}$$

$$q_{2m,veg} = q_{can,air} - \frac{H_{L,grd} + \frac{H_{L,can,evap} + H_{L,can,tr}}{fveg}}{\rho_{air} \times C_{LH,can,vap}} \times \frac{1}{C_{h,2m,veg}} \quad \text{(Eq. 3.6.308)}$$

3.6.12 Surface Fluxes and Temperature of Bare Ground

Description:

This part is to compute surface energy fluxes and update temperature for the bare ground portion of the pixel. The Newton-Raphson iteration is used to solve for ground temperature.

Relevant code modules:

Module: SurfaceEnergyFluxBareGroundMod.F90

Subroutines: SurfaceEnergyFluxBareGround

Physics:

1. The overall ground level energy balance (bare ground) is:

$$S_{W,down} - S_{W,up} + L_{W,down} - L_{W,up} + H_{pr,bare,grd} = H_S + H_L + G_H \quad \text{(Eq. 3.6.309)}$$

where S_W and L_W [W/m^2] are the shortwave and longwave radiative fluxes, respectively. $H_{pr,bare,grd}$ [W/m^2] is the net precipitation heat flux advected to the bare ground (computed in the PrecipitationHeatAdvecMod.F90 in earlier sections). H_S and H_L [W/m^2] are the ground sensible and latent heat fluxes (positive values indicate upward heat fluxes from the ground to the air), respectively. G_H [W/m^2] is the ground heat flux (positive value indicates downward heat flux from the soil/snow surface into underlying snow/soil layers).

Sensible heat (H_S, [W/m^2]) and latent heat (H_L, [W/m^2]) fluxes are computed based on the bulk transfer relationships following Garratt (1992):

$$H_S = \rho_a \times C_h \times C_p \times U \times (\theta_s - \theta_a) \quad \text{(Eq. 3.6.310)}$$

$$H_L = \rho_a \times C_w \times C_{LH} \times U \times (q_s - q_a) \quad \text{(Eq. 3.6.311)}$$

where ρ_a [kg/m^3] is the air density, $C_{p,air}$ [J/kg/K] is the air heat capacity, C_{LH} [J/kg] is the specific latent heat of water vaporization, and U [m/s] is the wind speed. θ_s [K] and θ_a [K] are the air potential temperatures at the surface and in the air, respectively. q_s [kg/kg] and q_a [kg/kg] are the specific humidity at the surface and in the air, respectively. C_h and C_w are the surface exchange coefficients for heat and moisture, respectively. Noah-MP assumes that C_w is equal to C_h (Chen and Zhang, 2009). Based on the Monin-Obukhov (M-O) similarity theory (Brutsaert, 1982), C_h is formulated as

$$C_h = \frac{\kappa^2}{\left[\ln\left(\frac{Z-d_0}{z_{0m}}\right) - \varphi_m\left(\frac{Z-d_0}{L_{MO}}\right)\right]\left[\ln\left(\frac{Z-d_0}{z_{0h}}\right) - \varphi_h\left(\frac{Z-d_0}{L_{MO}}\right)\right]} \quad \text{(Eq. 3.6.312)}$$

where κ is the von Karman constant, Z is the height above the ground, d_0 [m] is the zero-displacement height, and L_{MO} [m] is the M-O length. φ_m and φ_h are the stability functions for momentum and heat transfer, respectively. z_{0m} [m] and z_{0h} [m] are the surface roughness length for momentum and heat, respectively.

2. **Prepare heat flux coefficients:**

The coefficient for longwave radiation (C_{IR}) is:

$$C_{IR} = \varepsilon_{grd} \times S_B \quad \text{(Eq. 3.6.313)}$$

where ε_{grd} is the ground longwave emissivity computed above and S_B is the Stefan-Boltzmann constant defined in ConstantDefineMod.F90.

The coefficient for ground heat diffusion flux (C_{GH}) is:

$$C_{GH} = 2 \times \frac{K_{heat,snso}(I_{n,sno}+1)}{D_{zsnso}(I_{n,sno}+1)} \quad \text{(Eq. 3.6.314)}$$

where $K_{heat,snso}$ [W/m/K] is the soil/snow thermal conductivity and D_{zsnso} [m] is the soil/snow layer thickness. $I_{n,sno}$ is the snow layer indicator defined as (see detail in snow hydrology section below):

$$I_{n,sno} = \begin{cases} -3, & 3\ layer\ snow \\ -2, & 2\ layer\ snow \\ -1, & 1\ layer\ snow \\ 0, & 0\ layer\ snow \end{cases} \quad \text{(Eq. 3.6.315)}$$

3. Ground roughness length for heat flux (Z_{0h}, [m]):

$$Z_{0h,grd} = Z_{0m,grd} \quad \text{(Eq. 3.6.316)}$$

where $Z_{0m,grd}$ [m] is the ground roughness length for momentum computed in an earlier section, which is the theoretical height at which wind speed is zero. By default, Noah-MP assumes Z_{0h} to be equal to Z_{0m}. However, previous studies pointed out that Z_{0m} and Z_{0h} are different due to different mechanisms and resistances controlling momentum and heat transfer (Zilitinkevich, 1995; Chen and Zhang, 2009). Thus, an enhanced version of the Z_{0h} formulation based on Zilitinkevich (1995) and Chen and Zhang (2009) is as follows (currently deactivated in Noah-MP) for non-1st iteration:

$$Z_{0h} = Z_{0m} \times \mathrm{e}^{(-\kappa \times C_{zil} \times \sqrt{R_e})} \quad \text{(Eq. 3.6.317)}$$

$$C_{zil} = 10^{-0.4 \times h} \quad \text{(Eq. 3.6.318)}$$

where R_e is the roughness Reynolds number, C_{zil} is the empirical parameter represented as a function of the canopy height (h, [m]).

4. Compute aerodynamic resistances and exchange coefficients between reference height and the $Z_{0m,grd}$ height level:

Relevant code modules:

Module: ResistanceBareGroundMostMod.F90

Subroutines: ResistanceBareGroundMOST

Module: ResistanceBareGroundChen97Mod.F90

Subroutines: ResistanceBareGroundChen97

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSurfaceDrag	1*	Monin-Obukhov (M-O) Similarity Theory (MOST)
options for surface layer drag/exchange coefficient	2	original Noah (Chen et al. 1997)

Based on the ground roughness length (computed above) and some other quantities as input, the surface drag schemes: OptSurfaceDrag = 1 for the Monin-Obukhov (M-O) Similarity Theory (MOST) and OptSurfaceDrag = 2 for the Chen et al. 1997 scheme (original Noah), are used then to compute aerodynamic resistance for sensible heat ($R_{h,bare}$, [s/m]) and momentum ($R_{m,bare}$, [s/m]) and the corresponding momentum drag coefficients ($C_{m,bare}$) and sensible heat exchange coefficient ($C_{h,bare}$). They are connected by:

$$R_{h,bare} = \max(1, \frac{1}{C_{h,bare} \times U_{ref}}) \quad \text{(Eq. 3.6.319)}$$

$$R_{m,bare} = \max(1, \frac{1}{C_{m,bare} \times U_{ref}}) \quad \text{(Eq. 3.6.320)}$$

where U_{ref} [m/s] is the wind speed at reference height (computed in the earlier section).

The aerodynamic resistance for water vapor ($R_{w,bare}$, [s/m]) is assumed to be the same as sensible heat:

$$R_{w,bare} = R_{h,bare} \quad \text{(Eq. 3.6.321)}$$

The Monin-Obukhov (M-O) theory and Chen et al. (1997) schemes are introduced in detail below.

5. Compute saturated vapor pressure and derivative w.r.t. ground temperature:

Relevant code modules:

Module: VaporPressureSaturationMod.F90

Subroutines: VaporPressureSaturation

The saturation water vapor pressure ($e_{s,Tg}$, [Pa]) at ground temperature (T_g, [K]) is:

$$e_{s,Tg} = \begin{cases} e_{sw,Tg} & T_g > T_{frz} \\ e_{si,Tg} & T_g \le T_{frz} \end{cases} \quad \text{(Eq. 3.6.322)}$$

where $e_{sw,Tg}$ [Pa] and $e_{si,Tg}$ [Pa] are the saturation vapor pressure for water and ice, respectively, computed in VaporPressureSaturationMod.F90.

The derivative of saturation water vapor pressure ($\frac{d(e_{s,Tg})}{d(T_g)}$, [Pa/K]) at ground temperature is:

$$\frac{d(e_{s,Tg})}{d(T_g)} = \begin{cases} \frac{d(e_{sw,Tg})}{d(T_g)} & T_g > T_{frz} \\ \frac{d(e_{si,Tg})}{d(T_g)} & T_g \leq T_{frz} \end{cases} \quad \text{(Eq. 3.6.323)}$$

where $\frac{d(e_{sw,Tg})}{d(T_g)}$ [Pa/K] and $\frac{d(e_{si,Tg})}{d(T_g)}$ [Pa/K] are the derivative of saturation water vapor pressure for water and ice, respectively, computed in VaporPressureSaturationMod.F90.

To compute the saturation vapor pressure and corresponding derivative w.r.t. temperature for both water and ice at certain temperature, the VaporPressureSaturation module uses polynomial empirical parameterizations:

$$e_{sw,T} = 100 \times (A_0 + T \times (A_1 + T \times (A_2 + T \times (A_3 + T \times (A_4 + T \times (A_5 + T \times A_6)))))) \quad \text{(Eq. 3.6.324)}$$

$$e_{si,T} = 100 \times (B_0 + T \times (B_1 + T \times (B_2 + T \times (B_3 + T \times (B_4 + T \times (B_5 + T \times B_6)))))) \quad \text{(Eq. 3.6.325)}$$

$$\frac{d(e_{sw,T})}{d(T)} = 100 \times (C_0 + T \times (C_1 + T \times (C_2 + T \times (C_3 + T \times (C_4 + T \times (C_5 + T \times C_6)))))) \quad \text{(Eq. 3.6.326)}$$

$$\frac{d(e_{si,T})}{d(T)} = 100 \times (D_0 + T \times (D_1 + T \times (D_2 + T \times (D_3 + T \times (D_4 + T \times (D_5 + T \times D_6)))))) \quad \text{(Eq. 3.6.327)}$$

where A_i, B_i, C_i, D_i are polynomial coefficients.

6. Compute ground fluxes and temperature change:

(1) The coefficient for sensible heat (C_{SH}) is:

$$C_{SH} = \frac{\rho_{air} \times C_{p,air}}{R_{h,bare}} \quad \text{(Eq. 3.6.328)}$$

where ρ_{air} [kg/m^3] is the air density and $C_{p,air}$ [J/kg/K] is the heat capacity of dry air.

The coefficient for latent heat (C_{LH}) is:

$$C_{LH} = \frac{\rho_{air} \times C_{p,air}}{\gamma_{grd} \times (R_{h,bare} + R_{grd,evap})} \quad \text{(Eq. 3.6.329)}$$

where $R_{grd,evap}$ [s/m] is the surface resistance to ground evaporation/sublimation computed in ResistanceGroundEvaporationMod.F90 in the earlier section. γ_{grd} [Pa/K] is the psychrometric constant for the ground computed in PsychrometricVariableMod.F90 in the earlier section.

(2) The net ground longwave radiation ($L_{net,bare}$, [W/m^2]; positive for upward direction) is

$$L_{net,bare} = C_{IR} \times {T_g}^4 - \varepsilon_{grd} \times L_{W,down} \quad \text{(Eq. 3.6.330)}$$

where $L_{W,down}$ [W/m^2] is the downward longwave radiation from atmospheric forcing, C_{IR} is the longwave radiation coefficient computed above, ε_{grd} is the longwave emissivity (i.e., absorptivity), and T_g [K] is the ground temperature. The first term of the right-hand side of the equation is the emitted longwave radiation and the second term of the right-hand side of the equation is the absorbed longwave radiation.

(3) The sensible heat flux ($H_{S,bare}$, [W/m^2], positive for upward direction) is computed as:

$$H_{S,bare} = C_{SH} \times (T_g - T_{air}) \quad \text{(Eq. 3.6.331)}$$

where T_{air} [K] is the surface air temperature.

(4) The latent heat flux ($H_{L,bare}$, [W/m^2], positive for upward direction) is computed as:

$$H_{L,bare} = C_{LH} \times (e_{s,Tg} \times RH - e_{air}) \quad \text{(Eq. 3.6.332)}$$

where e_{air} [Pa] is the surface air vapor pressure and *RH* is the surface relative humidity.

(5) The ground heat flux ($G_{H,bare}$, [W/m^2], positive for downward direction) is computed as:

$$G_{H,bare} = C_{GH} \times (T_g - T_{snso}(I_{n,sno} + 1)) \quad \text{(Eq. 3.6.333)}$$

where C_{GH} is the coefficient for ground heat flux computed above and $T_{snso}(I_{n,sno} + 1)$ is the first or top snow/soil layer temperature.

(6) The change of ground temperature (ΔT_g, [K]) due to energy balance is computed as:

$$\Delta T_g = \frac{S_{bare,abs} - L_{net,bare} - H_{S,bare} - H_{L,bare} - G_{H,bare} + H_{pr,bare}}{4 \times C_{IR} \times {T_g}^3 + C_{SH} + C_{LH} \times \frac{d(e_{s,Tg})}{d(T_g)} + C_{GH}} \quad \text{(Eq. 3.6.334)}$$

where $S_{bare,abs}$ [W/m^2] is the total ground absorbed shortwave radiation computed in the radiation section.

(7) Update different heat fluxes and ground temperature:

$$L_{net,bare} = L_{net,bare} + 4 \times C_{IR} \times {T_g}^3 \times \Delta T_g \quad \text{(Eq. 3.6.335)}$$

$$H_{S,bare} = H_{S,bare} + C_{SH} \times \Delta T_g \quad \text{(Eq. 3.6.336)}$$

$$H_{L,bare} = H_{L,bare} + C_{LH} \times \frac{d(e_{s,Tg})}{d(T_g)} \times \Delta T_g \quad \text{(Eq. 3.6.337)}$$

$$G_{H,bare} = G_{H,bare} + C_{GH} \times \Delta T_g \quad \text{(Eq. 3.6.338)}$$

$$T_g = T_g + \Delta T_g \quad \text{(Eq. 3.6.339)}$$

$$H_{tmp} = C_{SH} \times \left(T_g - T_{air}\right) \quad \text{(Eq. 3.6.340)}$$

where H_{tmp} is a temporary heat flux variable used by the M-O theory calculation (OptSurfaceDrag = 1) in each iteration.

Then, the saturation vapor pressure and corresponding derivative w.r.t. temperature for the updated ground temperature are also updated by VaporPressureSaturationMod.F90 as described above. Then,

$$e_{s,Tg} = \begin{cases} e_{sw,Tg} & T_g > T_{frz} \\ e_{si,Tg} & T_g \leq T_{frz} \end{cases} \quad \text{(Eq. 3.6.341)}$$

The ground specific humidity (q_{grd}, [kg/kg]) is then updated as:

$$q_{grd} = \frac{0.622 \times e_{s,Tg} \times RH}{P_{sfc} - 0.378 \times e_{s,Tg} \times RH} \quad \text{(Eq. 3.6.342)}$$

The surface moisture flux ($Q_{flx,sfc}$, [kg/m^3]) is computed as:

$$Q_{flx,sfc} = (q_{grd} - q_{air,sfc}) \times C_{LH} \times \frac{\gamma_{grd}}{CP_{air}} \quad \text{(Eq. 3.6.343)}$$

where $q_{air,sfc}$ [kg/kg] is the surface air specific humidity from atmospheric forcing.

7. Iterations over the calculations for 3-6 to achieve convergence of ground temperature:

Currently, Noah-MP does 5 iterations (hard-coded) to go through the above 3rd-6th steps: 3. Ground roughness length for heat flux, 4. Compute aerodynamic resistances and heat exchange coefficients, 5. Compute saturated vapor pressure and derivative w.r.t. ground temperature, 6. Compute ground fluxes and temperature change.

8. Adjust ground temperature based on snow conditions:

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSnowSoilTempTime options for snow/soil temperature time scheme (only layer 1)	1*	semi-implicit; flux top boundary condition
	2	full implicit (original Noah); temperature top boundary condition
	3	same as 1, but snow cover for skin temperature calculation (generally improves snow)

For OptSnowSoilTempTime = 1 or 3, if there is snowpack ($h_{snow} > 0.05\,m$) and ground temperature is above freezing point ($T_g > T_{frz}$), then:

$$T_g = \begin{cases} T_{frz} & \text{OptSnowSoilTempTime} = 1 \\ T_{frz} \times f_{snow} + T_g \times (1 - f_{snow}) & \text{OptSnowSoilTempTime} = 3 \end{cases} \quad \text{(Eq. 3.6.344)}$$

Then the heat fluxes are re-evaluated based on the adjusted ground temperature:

$$L_{net,bare} = C_{IR} \times {T_g}^4 - \varepsilon_{grd} \times L_{W,down} \quad \text{(Eq. 3.6.345)}$$

$$H_{S,bare} = C_{SH} \times (T_g - T_{air}) \quad \text{(Eq. 3.6.346)}$$

$$H_{L,bare} = C_{LH} \times (e_{s,Tg} \times RH - e_{air}) \quad \text{(Eq. 3.6.347)}$$

$$G_{H,bare} = S_{bare,abs} + H_{pr,bare} - L_{net,bare} - H_{S,bare} - H_{L,bare} \quad \text{(Eq. 3.6.348)}$$

9. Compute wind stresses:

The wind stresses ($\tau_{x/y,bare}$, [N/m2]) at x- (east-west) and y- (north-south) directions are:

$$\tau_{x,bare} = -\rho_{air} \times C_{m,bare} \times U_{ref} \times U_x \quad \text{(Eq. 3.6.349)}$$

$$\tau_{y,bare} = -\rho_{air} \times C_{m,bare} \times U_{ref} \times U_y \quad \text{(Eq. 3.6.350)}$$

where ρ_{air} [kg/m^3] is the air density, $C_{m,bare}$ is the momentum drag coefficient, U_{ref} [m/s] is the wind speed at reference height, and $U_{x/y}$ [m/s] is the wind speed at x- or y-direction at reference height.

10. Compute 2-m air temperature as diagnosis:

For both OptSurfaceDrag = 1 and 2, the 2-m sensible heat exchange coefficient/conductance ($C_{h,2m,bare}$, [m/s]) for bare ground is computed as:

$$C_{h,2m,bare} = \frac{u_* \times \kappa}{ln\left(\frac{Z_{0h}+2}{Z_{0h}}\right) - \Psi_{h,2m}} \quad \text{(Eq. 3.6.351)}$$

where u_* [m/s] is the friction velocity computed from surface drag schemes (M-O or Chen et al. 1997 schemes). κ is the von Karman constant, Z_{0h} [m] is the ground roughness length for heat flux, and $\Psi_{h,2m}$ is the 2-m sensible heat stability correction factor (computed in M-O scheme or 0 for Chen et al. 1997 scheme in this equation). See below for details of M-O and Chen et al. 1997 schemes.

The latent heat exchange coefficient/conductance is assumed to be the same as $C_{h,2m,bare}$.

If $C_{h,2m,bare} < 10^{-5}$, the 2-m temperature ($T_{2m,bare}$, [K]) and specific humidity ($q_{2m,bare}$, [kg/kg]) are assumed to the same as the ground values:

$$T_{2m,bare} = T_{g,bare} \quad \text{(Eq. 3.6.352)}$$

$$q_{2m,bare} = q_{grd} \quad \text{(Eq. 3.6.353)}$$

If $C_{h,2m,bare} \geq 10^{-5}$, the 2-m temperature and specific humidity are computed as:

$$T_{2m,bare} = T_{g,bare} - \frac{H_{S,bare}}{\rho_{air} \times C_{p,air}} \times \frac{1}{C_{h,2m,bare}} \quad \text{(Eq. 3.6.354)}$$

$$q_{2m,bare} = q_{grd} - \frac{H_{L,bare}}{\rho_{air} \times C_{LH,grd,vap}} \times \left(\frac{1}{C_{h,2m,bare}} + R_{grd,evap}\right) \quad \text{(Eq. 3.6.355)}$$

where $C_{LH,grd,vap}$ [J/kg] is the ground latent heat of vaporization computed in PsychrometricVariableMod.F90.

For urban pixels,

$$q_{2m,bare} = q_{grd} \quad \text{(Eq. 3.6.356)}$$

3.6.13 Surface Drag Calculation

Description:

This part is to compute momentum and heat flux exchange/drag coefficients and aerodynamic resistances based on the Monin-Obukhov (M-O) Similarity Theory (MOST) or for the Chen et al. 1997 scheme (original Noah).

Relevant code modules:

Module: ResistanceBareGroundMostMod.F90

Subroutines: ResistanceBareGroundMOST

Module: ResistanceBareGroundChen97Mod.F90

Subroutines: ResistanceBareGroundChen97

Module: ResistanceAboveCanopyMostMod.F90

Subroutines: ResistanceAboveCanopyMOST

Module: ResistanceAboveCanopyChen97Mod.F90

Subroutines: ResistanceAboveCanopyChen97

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSurfaceDrag options for surface layer drag/exchange coefficient	1*	Monin-Obukhov (M-O) Similarity Theory (MOST)
	2	original Noah (Chen et al. 1997)

Physics:

1. Monin-Obukhov (M-O) Similarity Theory (OptSurfaceDrag = 1):

First, check if the reference height (z_{ref}, [m]) is greater than zero plane displacement (d_0, [m]). If $z_{ref} \leq d_0$, then the model will report an error and stop.

Then, a few intermediate temporary quantities used to compute momentum and heat exchange coefficient are computed:

$$F_{CM,ln} = ln\left[\frac{z_{ref}-d_0}{z_{0m}}\right] \quad \text{(Eq. 3.6.357)}$$

$$F_{CH,ln} = ln\left[\frac{z_{ref}-d_0}{z_{0h}}\right] \quad \text{(Eq. 3.6.358)}$$

$$F_{CM,2m,ln} = ln\left[\frac{2+z_{0m}}{z_{0m}}\right] \quad \text{(Eq. 3.6.359)}$$

$$F_{CH,2m,ln} = ln\left[\frac{2+z_{0h}}{z_{0h}}\right] \quad \text{(Eq. 3.6.360)}$$

The friction velocity (u_*), M-O turbulent stability parameter (ζ_{MO} and $\zeta_{MO,2m}$) and M-O characteristic length (L_{MO}) are initialized to 0 for the first iteration and then are computed as follows for the non-first iterations:

$$T_{vir,tmp} = (1 + 0.61 \times q_{sfc}) \times T_{sfc} \quad \text{(Eq. 3.6.361)}$$

$$F_{MO,tmp} = \max(10^{-6}, \frac{\kappa \times g}{T_{vir,tmp}} \times \frac{H_{tmp}}{\rho_{air} \times C_{p,air}}) \quad \text{(Eq. 3.6.362)}$$

$$L_{MO} = -\frac{{u_*}^3}{F_{MO,tmp}} \quad \text{(Eq. 3.6.363)}$$

$$\zeta_{MO} = \min(1.0, \frac{z_{ref}-d_0}{L_{MO}}) \quad \text{(Eq. 3.6.364)}$$

$$\zeta_{MO,2m} = \min(1.0, \frac{2+z_{0h}}{L_{MO}}) \quad \text{(Eq. 3.6.365)}$$

If ζ_{MO} changes sign from the last iteration to the current iteration, then the count variable ($N_{\zeta,sign}$) is increased by 1: $N_{\zeta,sign} = N_{\zeta,sign} + 1$. If $N_{\zeta,sign} \geq 2$, then the ζ_{MO} and M-O stability correction factor for momentum ($\Psi_{m,MO}$) and sensible heat ($\Psi_{h,MO}$) are all re-set to 0.

The M-O stability correction factors ($\Psi_{m,MO}$, $\Psi_{h,MO}$) weighted by the prior iteration are computed as:

$$\chi_{MO} = (1 - 16 \times \zeta_{MO})^{0.25} \quad \text{(Eq. 3.6.366)}$$

$$\Psi_{m,MO} = \begin{cases} 2 \times ln\left(\frac{1+\chi_{MO}}{2}\right) + ln\left(\frac{1+{\chi_{MO}}^2}{2}\right) - 2 \times \arctan(\chi_{MO}) + \frac{\pi}{2} & \zeta_{MO} < 0 \\ -5 \times \zeta_{MO} & 0 \le \zeta_{MO} \le 1 \end{cases}$$

(Eq. 3.6.367)

$$\Psi_{h,MO} = \begin{cases} 2 \times ln\left(\frac{1+{\chi_{MO}}^2}{2}\right) & \zeta_{MO} < 0 \\ -5 \times \zeta_{MO} & 0 \le \zeta_{MO} \le 1 \end{cases} \quad \text{(Eq. 3.6.368)}$$

The corresponding 2-m stability correction factors are computed similarly:

$$\chi_{MO,2m} = (1 - 16 \times \zeta_{MO,2m})^{0.25} \quad \text{(Eq. 3.6.369)}$$

$$\Psi_{m,MO,2m} = \begin{cases} 2 \times ln\left(\frac{1+\chi_{MO,2m}}{2}\right) + ln\left(\frac{1+{\chi_{MO,2m}}^2}{2}\right) - 2 \times \arctan(\chi_{MO,2m}) + \frac{\pi}{2} & \zeta_{MO,2m} < 0 \\ -5 \times \zeta_{MO,2m} & 0 \le \zeta_{MO,2m} \le 1 \end{cases}$$

(Eq. 3.6.370)

$$\Psi_{h,MO,2m} = \begin{cases} 2 \times ln\left(\frac{1+{\chi_{MO,2m}}^2}{2}\right) & \zeta_{MO,2m} < 0 \\ -5 \times \zeta_{MO,2m} & 0 \le \zeta_{MO,2m} \le 1 \end{cases} \quad \text{(Eq. 3.6.371)}$$

For the 1st iteration, $\Psi_{h,MO}$ and $\Psi_{m,MO}$ are the values from the computation above, while starting from the 2nd iteration, $\Psi_{h,MO}$ and $\Psi_{m,MO}$ are the averaged value of the above calculated new values and the old values calculated from the previous iteration. Weighting the M-O stability factors from the previous iteration helps to avoid flip-flops between iterations.

The M-O stability factors are further constrained as:

$$\Psi_{m,MO} = \min(\Psi_{m,MO}, 0.9 \times F_{CM,ln}) \quad \text{(Eq. 3.6.372)}$$

$$\Psi_{h,MO} = \min(\Psi_{h,MO}, 0.9 \times F_{CH,ln}) \quad \text{(Eq. 3.6.373)}$$

$$\Psi_{m,MO,2m} = \min(\Psi_{m,MO,2m}, 0.9 \times F_{CM,2m,ln}) \quad \text{(Eq. 3.6.374)}$$

$$\Psi_{h,MO,2m} = \min(\Psi_{h,MO,2m}, 0.9 \times F_{CH,2m,ln}) \quad \text{(Eq. 3.6.375)}$$

Then, the exchange coefficients for momentum (C_m) and heat flux (C_h) are computed as:

$$C_m = \frac{\kappa^2}{[\max(10^{-6},\ F_{CM,ln} - \Psi_{m,MO})]^2} \quad \text{(Eq. 3.6.376)}$$

$$C_h = \frac{\kappa^2}{[\max(10^{-6},\ F_{CM,ln} - \Psi_{m,MO})] \times [\max(10^{-6},\ F_{CH,ln} - \Psi_{h,MO})]} \quad \text{(Eq. 3.6.377)}$$

where κ is the von Karman constant.

Finally, the friction velocity (u_*, [m/s]) is computed as:

$$u_* = U_{ref} \times \sqrt{C_m} = \frac{U_{ref} \times \kappa}{\max(10^{-6},\ F_{CM,ln} - \Psi_{m,MO})} \quad \text{(Eq. 3.6.378)}$$

$$C_{h,2m} = \frac{u_* \times \kappa}{\max(10^{-6},\ ln\left[\frac{2+z_{0h}}{z_{0h}}\right] - \Psi_{h,MO,2m})} \quad \text{(Eq. 3.6.379)}$$

where U_{ref} [m/s] is the wind speed at reference height.

2. Chen et al. 1997 scheme (original Noah; OptSurfaceDrag = 2):

For the initial iteration (1st iteration):

The friction velocity at horizontal (u_*, [m/s]) and vertical (w_*, [m/s]) directions are adjusted based on the Beljaars correction (Beljaars and Viterbo, 1998):

$$w_*{}^2 = \begin{cases} 0 & C_h \times (\theta_a - T_{sfc}) = 0 \\ 1.44 \times \left|\beta \times g \times h_{pbl} \times C_h \times (\theta_a - T_{sfc})\right|^{2/3} & C_h \times (\theta_a - T_{sfc}) \neq 0 \end{cases} \quad \text{(Eq. 3.6.380)}$$

$$u_* = \max(0.07, \sqrt{C_m \times \sqrt{w_*{}^2 + \max(U_{ref}{}^2, 10^{-4})}}) \quad \text{(Eq. 3.6.381)}$$

where θ_a [K] and T_{sfc} [K] are the potential and surface temperature, respectively. C_h and C_m are the exchange coefficients for momentum and heat flux from the prior iteration. β is a parameter (=1/270 hard-coded). g [m/s2] is the gravity acceleration. h_{pbl} [m] is the planetary boundary layer depth (=1000 m, hard-coded). U_{ref} [m/s] is the wind speed at reference height.

The inverse of M-O characteristic length for the jth iteration ($\frac{1}{L_{MO,j}}$, j=1 here) is:

$$\frac{1}{L_{MO,j}} = \frac{\kappa \times \beta \times g \times C_h \times (\theta_a - T_{sfc})}{u_*{}^3} \quad \text{(Eq. 3.6.382)}$$

where κ is the von Karman constant.

For all iterations:

Then, the Zilitinkevitch approach is used to adjust surface roughness length for heat (z_{0h}):

$$z_{0h} = \max(10^{-6}, z_{0m} \times e^{(-\kappa \times C_{zil} \times \sqrt{R_e})} \quad \text{(Eq. 3.6.383)}$$

$$R_e = \frac{u_* \times z_{0m}}{\upsilon} \quad \text{(Eq. 3.6.384)}$$

where R_e is the roughness Reynolds number, υ is the kinematic viscosity, C_{zil} is the empirical Zilitinkevitch parameter.

Then, a few intermediate temporary quantities used to compute momentum and heat exchange coefficient are computed:

$$F_{CM,ln} = ln\left[\frac{z_{ref} + z_{0m}}{z_{0m}}\right] \quad \text{(Eq. 3.6.385)}$$

$$F_{CH,ln} = ln\left[\frac{z_{ref}+z_{0h}}{z_{0h}}\right] \quad \text{(Eq. 3.6.386)}$$

Then, the M-O turbulent stability parameter for the j[th] iteration (ζ_j) is further updated as:

$$\zeta_{h,j} = \max\left(\frac{z_{ref}+z_{0h}}{L_{MO,j-1}}, \zeta_{min}\right) \quad , \quad \zeta_{min} = -5 \quad \text{(Eq. 3.6.387)}$$

$$\frac{1}{L_{MO,j}} = \frac{\zeta_{h,j}}{z_{ref}+z_{0h}} \quad \text{(Eq. 3.6.388)}$$

$$\zeta_{m,j} = \frac{z_{ref}+z_{0m}}{L_{MO,j}} \quad \text{(Eq. 3.6.389)}$$

$$\zeta_{m,j}' = \frac{z_{0m}}{L_{MO,j}} \quad \text{(Eq. 3.6.390)}$$

$$\zeta_{h,j}' = \frac{z_{0h}}{L_{MO,j}} \quad \text{(Eq. 3.6.391)}$$

If $\zeta \geq 0$, it is further constrained by a maximum value ($\zeta_{max} = 1$) as follows:

$$\zeta_{h,j} = \min(\zeta_{h,j}, \zeta_{max}) \quad \text{(Eq. 3.6.392)}$$

$$\zeta_{m,j} = \min(\zeta_{m,j}, \zeta_{max}) \quad \text{(Eq. 3.6.393)}$$

$$\zeta_{m,j}' = \min\left(\zeta_{m,j}', \zeta_{max} \times \frac{z_{0m}}{z_{ref}+z_{0m}}\right) \quad \text{(Eq. 3.6.394)}$$

$$\zeta_{h,j}' = \min\left(\zeta_{h,j}', \zeta_{max} \times \frac{z_{0h}}{z_{ref}+z_{0h}}\right) \quad \text{(Eq. 3.6.395)}$$

There are two options for computing the stability correction factors ($\Psi_{m,MO}$, $\Psi_{h,MO}$): the Paulson's surface function (default in Noah-MP) and the Lobocki's surface function.

If using the Paulson's surface function,

$$\chi_m = (1 - 16 \times \zeta_{m,j})^{0.25} \quad \text{(Eq. 3.6.396)}$$

$$\chi_h = (1 - 16 \times \zeta_{h,j})^{0.25} \quad \text{(Eq. 3.6.397)}$$

$$\chi_m' = (1 - 16 \times \zeta_{m,j}')^{0.25} \quad \text{(Eq. 3.6.398)}$$

$$\chi_h' = (1 - 16 \times \zeta_{h,j}')^{0.25} \quad \text{(Eq. 3.6.399)}$$

$$\Psi_m(\zeta) = \begin{cases} 2 \times ln\left(\frac{1+\chi}{2}\right) + ln\left(\frac{1+\chi^2}{2}\right) - 2 \times \arctan(\chi) + \frac{\pi}{2} & -5 \leq \zeta < 0 \\ -5 \times \zeta & 0 \leq \zeta \leq 1 \end{cases} \quad \text{(Eq. 3.6.400)}$$

$$\Psi_h(\zeta) = \begin{cases} 2 \times ln\left(\frac{1+\chi^2}{2}\right) & -5 \leq \zeta < 0 \\ -5 \times \zeta & 0 \leq \zeta \leq 1 \end{cases} \quad \text{(Eq. 3.6.401)}$$

where ζ can be $\zeta_{m,j}$, $\zeta_{h,j}$, $\zeta_{m,j}'$, or $\zeta_{h,j}'$ in the calculation of C_h and C_m below and hence χ in the above equation will be the corresponding $\chi_{m,j}$, $\chi_{h,j}$, $\chi_{m,j}'$, or $\chi_{h,j}'$ in $\Psi_m(\zeta)$ and $\Psi_h(\zeta)$ in the calculation of C_h and C_m below.

If using the the Lobocki's surface function,

$$\Psi_m(\zeta) = \begin{cases} -0.96 \times \ln(1 - 4.5 \times \zeta) & -5 \le \zeta < 0 \\ \frac{\zeta}{R_{ic}} - 2.076 \times \left[1 - \frac{1}{1+\zeta}\right] & 0 \le \zeta \le 1 \end{cases} \quad \text{(Eq. 3.6.402)}$$

$$\Psi_h(\zeta) = \begin{cases} -0.96 \times \ln(1 - 4.5 \times \zeta) & -5 \le \zeta < 0 \\ \frac{\zeta \times R_{ic}}{R_{FC}{}^2 \times \phi_u} - 2.076 \times \left[1 - \frac{1}{1+\zeta}\right] & 0 \le \zeta \le 1 \end{cases} \quad \text{(Eq. 3.6.403)}$$

where R_{ic} (=0.183) is the critical gradient Richardson number, R_{FC} (=0.191) is the critical flux Richardson number, and ϕ_u (=0.8) is the dimensionless velocity gradient for neutral conditions.

Update u_* using the Beljaars correction:

$$u_* = \max(0.07, \sqrt{C_m \times \sqrt{w_*{}^2 + \max(U_{ref}{}^2, 10^{-4})}}) \quad \text{(Eq. 3.6.404)}$$

Update z_{0h} using the Zilitinkevitch approach:

$$R_e = \frac{u_* \times z_{0m}}{v} \quad \text{(Eq. 3.6.405)}$$

$$z_{0h} = \max(10^{-6}, z_{0m} \times \mathrm{e}^{(-\kappa \times C_{zil} \times \sqrt{R_e})} \quad \text{(Eq. 3.6.406)}$$

Then, the exchange coefficients for momentum (C_m) and heat flux (C_h) are computed as:

$$C_m = \max(\frac{0.001}{z_{ref}}, \frac{\kappa \times u_*}{\max(10^{-6}, \Psi_m(\zeta_{m,j}) - \Psi_m(\zeta_{m,j}{}') + F_{CM,ln})}) \quad \text{(Eq. 3.6.407)}$$

$$C_h = \max(\frac{0.001}{z_{ref}}, \frac{\kappa \times u_*}{\max(10^{-6}, \Psi_h(\zeta_{h,j}) - \Psi_h(\zeta_{h,j}{}') + F_{CH,ln})}) \quad \text{(Eq. 3.6.408)}$$

where the specific formulation of $\Psi_m(\zeta)$ and $\Psi_h(\zeta)$ are based on the chosen surface function (Paulson's or Lobocki's) above.

Further update w_* based on the Beljaars correction:

$$w_*{}^2 = \begin{cases} 0 & C_h \times (\theta_a - T_{sfc}) = 0 \\ 1.44 \times \left|\beta \times g \times h_{pbl} \times C_h \times (\theta_a - T_{sfc})\right|^{2/3} & C_h \times (\theta_a - T_{sfc}) \ne 0 \end{cases} \quad \text{(Eq. 3.6.409)}$$

Then update the inverse of M-O characteristic length ($\frac{1}{L_{MO,j}{}'}$):

$$\frac{1}{L_{MO,j}{}'} = \frac{\kappa \times \beta \times g \times C_h \times (\theta_a - T_{sfc})}{u_*{}^3} \quad \text{(Eq. 3.6.410)}$$

$$\frac{1}{L_{MO,j}} = 0.15 \times \frac{1}{L_{MO,j}} + 0.85 \times \frac{1}{L_{MO,j}{}'} \quad \text{(Eq. 3.6.411)}$$

where the M-O characteristic length is updated using a weighted (15% vs 85%) average of the old value ($L_{MO,j}$) from prior iteration and the new value ($L_{MO,j}{}'$) from the current iteration.

3.6.14 Grid-level Mean Energy Flux and Temperature

Description:

This part is to compute the grid-level mean energy and temperature quantities by averaging over vegetated and bare portion of each pixel. This is because Noah-MP currently uses a tiled method to model vegetated and bare subpixel areas.

Relevant code modules:

Module: EnergyMainMod.F90

Subroutines: EnergyMain

Physics:

1. For pixels with non-zero vegetation cover,

(1) At the ground level, the grid mean energy or temperature quantities ($M_{grd,grid}$) are computed as:

$$M_{grd,grid} = f_{veg} \times M_{veg,grd} + (1 - f_{veg}) \times M_{bare,grd} \qquad \text{(Eq. 3.6.412)}$$

where f_{veg} is the vegetation fraction computed in PhenologyMainMod.F90. $M_{veg,grd}$ and $M_{bare,grd}$ are the quantities for vegetated (below-canopy) and bare portions of the pixel, respectively computed from the energy processes above.

(2) At the surface (canopy) level, the grid mean energy or temperature state variables ($M_{sfc,grid}$) are computed as:

$$M_{sfc,grid} = f_{veg} \times M_{veg} + (1 - f_{veg}) \times M_{bare} \qquad \text{(Eq. 3.6.413)}$$

where M_{veg} and M_{bare} are the quantities for vegetated (canopy-level) and bare portions of the pixel, respectively computed from the energy processes above.

The surface-level grid mean energy flux variables ($F_{sfc,grid}$) are computed as:

$$F_{sfc,grid} = M_{grd,grid} + M_{veg,can} \qquad \text{(Eq. 3.6.414)}$$

where $M_{grd,grid}$ is the mean ground level flux and $M_{veg,can}$ is the canopy-level flux for vegetated portion of the pixel.

2. For pixels without any vegetation fraction, the pixel is essentially bare ground and the grid-level mean energy or temperature quantities will be the same as the values for bare portion of the pixel:

$$M_{sfc,grid} = M_{bare} = M_{bare,grd} \quad \text{(Eq. 3.6.415)}$$

The canopy-related quantities will then be assigned as 0.

3.6.15 Surface Longwave Emission and Radiative Temperature

Description:

This part is to compute surface emitted longwave radiation and radiative/skin temperature.

Relevant code modules:

Module: EnergyMainMod.F90

Subroutines: EnergyMain

Physics:

1. The total emitted longwave radiation ($L_{W,sfc,emit}$, [W/m^2]) at the surface is computed as:

$$L_{W,sfc,emit} = L_{W,down} + L_{W,sfc,net} \quad \text{(Eq. 3.6.416)}$$

where $L_{W,down}$ [W/m^2] is the downward longwave radiation from atmospheric forcing and $L_{W,sfc,net}$ [W/m^2] is the net surface longwave radiation computed through energy processes above. The $L_{W,sfc,emit}$ includes the originally surface emitted longwave radiation and the reflected portion of the incoming longwave radiation. If the $L_{W,sfc,emit}$ is less than 0, the model will report the error and stop.

2. The radiative/skin temperature (T_{rad}, [K]) is computed from the surface longwave radiation and Stefan-Boltzmann law:

$$T_{rad} = \left[\frac{L_{W,sfc,emit} - (1-\varepsilon_{sfc}) \times L_{W,down}}{\varepsilon_{sfc} \times S_B}\right]^{0.25} \quad \text{(Eq. 3.6.417)}$$

where ε_{sfc} is the net surface longwave emissivity computed above and S_B is the Stefan-Boltzmann constant defined in ConstantDefineMod.F90.

3.6.16 Total Photosynthesis Radiation

Description:

This part is to compute the total photosynthesis-related radiation quantities.

Relevant code modules:

Module: EnergyMainMod.F90

Subroutines: EnergyMain

Physics:

1. The total photosynthesis active radiation ($S_{par,tot}$, [W/m^2]) is computed as:

$$S_{PAR,tot} = S_{PAR,sun} \times E_{LAI,sun} + S_{PAR,shd} \times E_{LAI,shd} \quad \text{(Eq. 3.6.418)}$$

where $S_{PAR,sun}$ [W/m^2] and $S_{PAR,shd}$ [W/m^2] are photosynthesis active radiation for sunlit and shaded leaves, respectively, computed in SurfaceRadiationMod.F90. $E_{LAI,sun}$ and $E_{LAI,shd}$ are the effective leaf area index for sunlit and shaded leaves, respectively, computed in SurfaceAlbedoMod.F90.

2. The total photosynthesis rate ($P_{SN,tot}$, [umol CO_2/m^2/s]) is computed as:

$$P_{SN,tot} = P_{SN,sun} \times E_{LAI,sun} + P_{SN,shd} \times E_{LAI,shd} \quad \text{(Eq. 3.6.419)}$$

where $P_{SN,sun}$ [W/m^2] and $P_{SN,shd}$ [W/m^2] are photosynthesis rates for sunlit and shaded leaves, respectively, computed in the Ball-Berry stomatal resistance scheme. Note that currently the Jarvis stomatal resistance scheme in Noah-MP does not compute photosynthesis rate and hence cannot be used together with the dynamic vegetation and crop model.

3.6.17 Snow and Soil Temperature Solver

Description:

This part is to compute snow and soil layer temperature based on energy flux calculations in previous sections.

Relevant code modules:

Module: SoilSnowTemperatureMainMod.F90

Subroutines: SoilSnowTemperatureMain

Physics:

1. Currently, Noah-MP does not account for solar radiation penetrating (φ_{snso}, [W/m2]) through snowpack and soil water, which needs more work in the future. Thus, this part of energy will not be included in snow and soil temperature calculations.

2. Snow and soil thermal diffusion calculation (SoilSnowThermalDiffusionMod.F90):
This module is to calculate the right hand side of the time tendency term of the snow and soil thermal diffusion equation. Currently, snow and soil layers are coupled in solving the equations. This module also computes/prepares the matrix coefficients for the tri-diagonal matrix of the implicit time scheme.

Relevant code modules:

Module: SoilSnowThermalDiffusionMod.F90

Subroutines: SoilSnowThermalDiffusion

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSnowSoilTempTime options for snow/soil temperature time scheme (only layer 1)	1*	semi-implicit; flux top boundary condition
	2	full implicit (original Noah); temperature top boundary condition
	3	same as 1, but snow cover for skin temperature calculation (generally improves snow)
OptSoilTemperatureBottom options for lower boundary condition of soil temperature	1	zero heat flux from bottom (DepthSoilTempBottom & TemperatureSoilBottom not used)
	2*	TemperatureSoilBottom at DepthSoilTempBottom (8m) read from a file (original Noah)

(1) Calculate the time tendency term of thermal diffusion equation:

First, the soil and snow thermal conductivity ($K_{heat,snso}$, [W/m/K]) and heat capacity ($C_{heat,snso}$, [J/m^3/K]) are used as input, which are computed in previous sections.

Then, the excess energy flux in the diffusion equations ($E_{heat,diff}$, [W/m^2]) is computed as:

$$E_{heat,diff}(i) = \begin{cases} K_{heat,snso}(i) \times D_h(i) - G_{snso} - \varphi_{snso}(i) & i = I_{n,sno} + 1 \\ K_{heat,snso}(i) \times D_h(i) - K_{heat,snso}(i-1) \times D_h(i-1) - \varphi_{snso}(i) & I_{n,sno} + 1 < i < N_{soil} \\ F_{bot} - K_{heat,snso}(i-1) \times D_h(i-1) - \varphi_{snso}(i) & i = N_{soil} \end{cases}$$

(Eq. 3.6.420)

where G_{snso} [W/m^2] is the ground heat flux going into snow/soil layers from air above. φ_{snso} is the solar radiation penetrating snow/soil layers (currently is 0). $I_{n,sno}$ is the snow layer indicator defined as (see detail in snow hydrology section below):

$$I_{n,sno} = \begin{cases} -3, & 3\ layer\ snow \\ -2, & 2\ layer\ snow \\ -1, & 1\ layer\ snow \\ 0, & 0\ layer\ snow \end{cases}$$

(Eq. 3.6.421)

$D_h(i)$ [K/m] is the temperature gradient (derivative) between two neighboring snow/soil layers defined as:

$$D_h(i) = \begin{cases} 2 \times \dfrac{T_{snso}(i) - T_{snso}(i+1)}{-Z_{snso}(i+1)} & i = I_{n,sno} + 1 \\ 2 \times \dfrac{T_{snso}(i) - T_{snso}(i+1)}{Z_{snso}(i-1) - Z_{snso}(i+1)} & I_{n,sno} + 1 < i < N_{soil} \\ \dfrac{T_{snso}(N_{soil}) - T_{bot}}{0.5 \times \left(Z_{snso}(N_{soil}-1) + Z_{snso}(N_{soil})\right) - Z_{bot}} & i = N_{soil} \end{cases}$$

(Eq. 3.6.422)

where T_{bot} [K] and Z_{bot} [m] are temperature and depth of deep soil bottom provided by the input dataset as forcing. Z_{snso} [m] and T_{snso} [K] are the snow and soil layer depth and temperature, respectively.

F_{bot} [W/m^2] is the heat flux from deep soil bottom defined as:

$$F_{bot} = \begin{cases} 0 & \text{OptSoilTemperatureBottom} = 1 \\ -K_{heat,snso}(N_{soil}) \times D_h(N_{soil}) & \text{OptSoilTemperatureBottom} = 2 \end{cases}$$

(Eq. 3.6.423)

(2) Compute (prepare) the matrix coefficients for the tri-diagonal matrix:

The left-hand side of the matrix coefficients for the tri-diagonal matrix are classified to the three groups (A_I, B_I, C_I; [1/s]):

For the top snow layer (i = $I_{n,sno}$ + 1):

$$D_{DZ}(i) = \frac{2}{-Z_{snso}(i+1)} \quad \text{(Eq. 3.6.424)}$$

$$A_I(i) = 0.0 \quad \text{(Eq. 3.6.425)}$$

$$C_I(i) = -K_{heat,snso}(i) \times \frac{D_{DZ}(i)}{-Z_{snso}(i) \times C_{heat,snso}(i)} \quad \text{(Eq. 3.6.426)}$$

$$B_I(i) = \begin{cases} -C_I(i) & \text{OptSnowSoilTempTime} = 1,3 \\ -C_I(i) + \dfrac{K_{heat,snso}(i)}{0.5 \times Z_{snso}(i) \times Z_{snso}(i) \times C_{heat,snso}(i)} & \text{OptSnowSoilTempTime} = 2 \end{cases}$$

(Eq. 3.6.427)

For the middle snow or soil layers ($I_{n,sno}$ + 1 < i < N_{soil}):

$$D_{DZ}(i) = \frac{2}{Z_{snso}(i-1) - Z_{snso}(i+1)} \quad \text{(Eq. 3.6.428)}$$

$$\begin{cases} A_I(i) = -K_{heat,snso}(i-1) \times \dfrac{D_{DZ}(i-1)}{(Z_{snso}(i-1) - Z_{snso}(i)) \times C_{heat,snso}(i)} \\ B_I(i) = -(A_I(i) + C_I(i)) \\ C_I(i) = -K_{heat,snso}(i) \times \dfrac{D_{DZ}(i)}{(Z_{snso}(i-1) - Z_{snso}(i)) \times C_{heat,snso}(i)} \end{cases} \quad \text{(Eq. 3.6.429)}$$

For the bottom soil layer (i = N_{soil}):

$$\begin{cases} A_I(i) = -K_{heat,snso}(i-1) \times \dfrac{D_{DZ}(i-1)}{(Z_{snso}(i-1) - Z_{snso}(i)) \times C_{heat,snso}(i)} \\ B_I(i) = -(A_I(i) + C_I(i)) \\ C_I(i) = 0.0 \end{cases} \quad \text{(Eq. 3.6.430)}$$

For all snow and soil layers, the right-hand side (R_{HS}, [1/s]) of the matrix coefficients for the tri-diagonal matrix is calculated as:

$$R_{HS}(i) = \begin{cases} \dfrac{E_{heat,diff}(i)}{Z_{snso}(i) \times C_{heat,snso}(i)} & i = I_{n,sno} + 1 \\ \dfrac{E_{heat,diff}(i)}{(Z_{snso}(i) - Z_{snso}(i-1)) \times C_{heat,snso}(i)} & I_{n,sno} + 1 < i \le N_{soil} \end{cases} \quad \text{(Eq. 3.6.431)}$$

The A_I [1/s], B_I [1/s], C_I [1/s], and R_{HS} [1/s] are used in the following temperature tri-diagonal matrix solver in SoilSnowTemperatureSolverMod.F90.

3. Snow and soil temperature solver (SoilSnowTemperatureSolverMod.F90):

This module is to compute snow and soil layer temperatures using a tri-diagonal matrix solution dependent on the output from SoilSnowThermalDiffusionMod.F90 above.

Relevant code modules:

Module: SoilSnowTemperatureSolverMod.F90

Subroutines: SoilSnowTemperatureSolver

First, update the matrix coefficients based on model timestep (Δt, [s]) for snow and soil layers:

$$\begin{cases} R_{HS}(i) = R_{HS}(i) \times \Delta t \\ A_I(i) = A_I(i) \times \Delta t \\ B_I(i) = 1 + B_I(i) \times \Delta t \\ C_I(i) = C_I(i) \times \Delta t. \end{cases} \qquad \text{(Eq. 3.6.432)}$$

Then, these matrix coefficients are input to the tri-diagonal matrix solver in MatrixSolverTriDiagonalMod.F90 (see below for details), which updates $C_I(i)$.

The snow and soil layer temperature (T_{snso}, [K]) is then updated based on $C_I(i)$:

$$T_{snso}(i) = T_{snso}(i) + C_I(i)\,, \qquad I_{n,sno} + 1 \le i \le N_{soil} \qquad \text{(Eq. 3.6.433)}$$

4. Tri-diagonal matrix solver in MatrixSolverTriDiagonalMod.F90:

Relevant code modules:

Module: MatrixSolverTriDiagonalMod.F90

Subroutine: MatrixSolverTriDiagonal

```
INVERT (SOLVE) THE TRI-DIAGONAL MATRIX PROBLEM SHOWN BELOW:
###                                            ### ###   ###    ###   ###
#B(1), C(1),  0  ,  0  ,  0  ,   . . . ,    0   # #       #    #       #
#A(2), B(2), C(2),  0  ,  0  ,   . . . ,    0   # #       #    #       #
# 0  , A(3), B(3), C(3),  0  ,   . . . ,    0   # #       #    # D(3) #
# 0  ,  0  , A(4), B(4), C(4),   . . . ,    0   # # P(4) #    # D(4) #
# 0  ,  0  ,  0  , A(5), B(5),   . . . ,    0   # # P(5) #    # D(5) #
# .                                         .   # #  .   # = #   .  #
# .                                         .   # #  .   #   #   .  #
# .                                         .   # #  .   #   #   .  #
# 0  , . . . , 0 , A(M-2), B(M-2), C(M-2),   0   # #P(M-2)#   #D(M-2)#
# 0  , . . . , 0 ,   0   , A(M-1), B(M-1), C(M-1)# #P(M-1)#   #D(M-1)#
# 0  , . . . , 0 ,   0   ,   0   ,  A(M) ,  B(M) # # P(M) #   # D(M) #
###                                            ### ###   ###    ###   ###
```

The matrix coefficients (A, B, C, D) are inputs (A_I, B_I, C_I, and R_{HS}) coming from the calculations above in SoilSnowThermalDiffusionMod.F90 and SoilSnowTemperatureSolverMod.F90. To solve (invert) the tri-diagonal matrix:

First, initialize equation coefficient:

$$C(N_{soil}) = 0.0 \quad \text{(Eq. 3.6.434)}$$

$$P\left(N_{top}\right) = \frac{-C\left(N_{top}\right)}{B\left(N_{top}\right)} \quad \text{(Eq. 3.6.435)}$$

where N_{top} (= $I_{n,sno} + 1$) is the top snow layer.

Then, solve the equation coefficients for the top snow layer:

$$\Delta\left(N_{top}\right) = \frac{D\left(N_{top}\right)}{B\left(N_{top}\right)} \quad \text{(Eq. 3.6.436)}$$

Solve the equation coefficients for other snow/soil layers i from $I_{n,sno} + 2$ through N_{soil}:

$$P(i) = -C(i) \times \left[\frac{1}{B(i)+A(i)\times P(i-1)}\right] \quad \text{(Eq. 3.6.437)}$$

$$\Delta(i) = [D(i) - A(i) \times \Delta(i-1)] \times \left[\frac{1}{B(i)+A(i)\times P(i-1)}\right] \quad \text{(Eq. 3.6.438)}$$

Set P to Δ for the lowest soil layer:

$$P(N_{soil}) = \Delta(N_{soil}) \quad \text{(Eq. 3.6.439)}$$

Adjust P for snow/soil layers i from $I_{n,sno} + 2$ through N_{soil}:

$$ii = N_{soil} - i + \left(N_{top} - 1\right) + 1 \quad \text{(Eq. 3.6.440)}$$

$$P(ii) = P(ii) \times P(ii+1) + \Delta(N_{soil}) \quad \text{(Eq. 3.6.441)}$$

Here P is the output variable that is further used to update soil and snow temperature as shown above.

3.6.18 Surface Temperature Adjustments for Snow Condition

Description:

This part is to adjust surface temperature based on snow conditions for OptSnowSoilTempTime = 2. For OptSnowSoilTempTime = 1 or 3, the adjustments are done in the surface energy flux and temperature calculations for vegetated and bare grounds above.

Relevant code modules:

Module: EnergyMainMod.F90

Subroutines: EnergyMain

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSnowSoilTempTime options for snow/soil temperature time scheme (only layer 1)	1*	semi-implicit; flux top boundary condition
	2	full implicit (original Noah); temperature top boundary condition
	3	same as 1, but snow cover for skin temperature calculation (generally improves snow)

Physics:

For OptSnowSoilTempTime = 2, if there is snowpack ($h_{snow} > 0.05\ m$) and ground temperature is above freezing point ($T_{g,grid} > T_{frz}$), then the ground temperature for both vegetated and bare portion of the pixel are reset to the freezing point:

$$T_{g,veg} = T_{frz} \quad \text{(Eq. 3.6.442)}$$

$$T_{g,bare} = T_{frz} \quad \text{(Eq. 3.6.443)}$$

If the pixel/grid has non-zero vegetation fraction ($f_{veg} > 0$), then:

$$T_{g,grid} = f_{veg} \times T_{g,veg} + (1 - f_{veg}) \times T_{g,bare} \quad \text{(Eq. 3.6.444)}$$

$$T_{sfc,grid} = f_{veg} \times T_{veg,can} + (1 - f_{veg}) \times T_{g,bare} \quad \text{(Eq. 3.6.445)}$$

where $T_{veg,can}$ [K] is the canopy temperature for the vegetated portion of the pixel/grid.

If the pixel/grid has no vegetation fraction ($f_{veg} = 0$), then:

$$T_{g,grid} = T_{g,bare} \quad \text{(Eq. 3.6.446)}$$

$$T_{sfc,grid} = T_{g,bare} \quad \text{(Eq. 3.6.447)}$$

3.6.19 Snow and Soil Water Phase Change

Description:

This part is to compute the phase change (melting/freezing) of snow water and soil water based on snow and soil layer temperature computed in previous sections.

Relevant code modules:

Module: SoilSnowWaterPhaseChangeMod.F90

Subroutines: SoilSnowWaterPhaseChange

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSoilSupercoolWater options for soil supercooled liquid water	1*	no iteration (Niu and Yang, 2006)
	2	Koren's iteration (Koren et al., 1999)

Physics:

1. Compute soil supercooled water for soil point/pixel:

(1) if OptSoilSupercoolWater = 1 (No iteration, Niu and Yang 2006 scheme):

Relevant code modules:

Module: SoilWaterSupercoolNiu06Mod.F90

Subroutines: SoilWaterSupercoolNiu06

When $T_{frz} > T_{soil}$, the frozen soil water matric potential ($\psi_{soil,frz}$, [m]) is firstly computed as:

$$\psi_{soil,frz} = C_{LH,fus} \times \frac{T_{frz} - T_{soil}}{g \times T_{soil}} \quad \text{(Eq. 3.6.448)}$$

where $C_{LH,fus}$ [J/kg] is the specific latent heat of fusion and g [m/s^2] is the gravity acceleration (defined in ConstantDefineMod.F90). T_{frz} and T_{soil} are the freezing and soil temperatures, respectively.

Then, the supercooled soil liquid water content ($W_{liq,supercool}$, [m^3/m^3]) is computed as:

$$W_{liq,supercool} = \theta_{soil,max} \times \left[\frac{\psi_{soil,frz}}{\psi_{soil,sat}}\right]^{\frac{-1}{B_{exp}}} \quad \text{(Eq. 3.6.449)}$$

where $\theta_{soil,max}$ [m^3/m^3] is the maximum (saturated) soil moisture (set in the NoahmpTable.TBL), B_{exp} is the soil exponential coefficient depending on soil texture (set in NoahmpTable.TBL), and $\psi_{soil,sat}$ [m] is the matric potential of saturated soil (set in NoahmpTable.TBL).

The amount of supercooled liquid water content ($M_{liq,supercool}$, [mm]) is computed as:

$$M_{liq,supercool} = W_{liq,supercool} \times D_{z,soil} \times 1000 \quad \text{(Eq. 3.6.450)}$$

where $D_{z,soil}$ [m] is the soil layer thickness.

(2) if OptSoilSupercoolWater = 2 (Koren's iteration, Koren et al. 1999 scheme):

Relevant code modules:

Module: SoilWaterSupercoolKoren99Mod.F90

Subroutines: SoilWaterSupercoolKoren99

This scheme uses Newton-type iteration to solve the nonlinear implicit equation given in Koren et al. (1999). A new version of this scheme has been created to accelerate the Newton iteration by first taking log of the equation cited above and then using the explicit 1-step solution option for special case of parameter ($C_k = 0$, see below), which reduces the original implicit equation to a simpler explicit form known as "Flerchinger Equation".

If $T_{frz} > T_{soil} > T_{frz} - 0.001$ (soil temperature not significantly below freezing):

$$W_{liq,supercool} = \theta_{soil} \quad \text{(Eq. 3.6.451)}$$

where θ_{soil} [m^3/m^3] is the soil moisture.

If $T_{soil} \leq T_{frz} - 0.001$ (soil temperature significantly below freezing):

While $C_k \neq 0$, the soil ice content ($W_{ice,soil}$, [m^3/m^3]) is computed as:

$$W_{ice,soil} = \theta_{soil} - W_{liq,soil} \quad \text{(Eq. 3.6.452)}$$

where $W_{liq,soil}$ [m^3/m^3] is the soil liquid water content.

$$W_{ice,soil} = \begin{cases} \theta_{soil} - 0.02 & W_{ice,soil} > \theta_{soil} - 0.02 \\ 0 & W_{ice,soil} < 0 \end{cases} \quad \text{(Eq. 3.6.453)}$$

Here $W_{ice,soil}$ is the initial guess to start the iteration.

$$k_i = ln\left[\frac{g \times \psi_{soil,sat}}{C_{LH,fus}} \times (1 + C_k \times W_{ice,soil})^2 \times \left(\frac{\theta_{soil,max}}{\theta_{soil} - W_{ice,soil}}\right)^{B_{exp}}\right] - ln\left[\frac{T_{frz} - T_{soil}}{T_{soil}}\right]$$

(Eq. 3.6.454)

$$D_{nom} = 2 \times \frac{C_k}{1 + C_k \times W_{ice,soil}} + \frac{B_{exp}}{\theta_{soil} - W_{ice,soil}} \quad \text{(Eq. 3.6.455)}$$

$$W_{ice,soil,tmp} = W_{ice,soil} - \frac{k_i}{D_{nom}} \quad \text{(Eq. 3.6.456)}$$

$$W_{ice,soil,tmp} = \begin{cases} \theta_{soil} - 0.02 & W_{ice,soil,tmp} > \theta_{soil} - 0.02 \\ 0 & W_{ice,soil,tmp} < 0 \end{cases} \quad \text{(Eq. 3.6.457)}$$

Iteration ends when soil ice content converges (i.e., $|W_{ice,soil,tmp} - W_{ice,soil}| < 0.005$), and the supercooled liquid water content will be:

$$W_{liq,supercool} = \theta_{soil} - W_{ice,soil,tmp} \quad \text{(Eq. 3.6.458)}$$

Otherwise (not converged), the iterations continue:

$$W_{ice,soil} = W_{ice,soil,tmp} \quad \text{(Eq. 3.6.459)}$$

to start a new iteration and re-do the above calculations and update $W_{ice,soil,tmp}$.

If more than 10 iterations, the explicit method (Flerchinger Equation, $C_k = 0$ approximation) will be used:

$$F_k = \theta_{soil,max} \times \left[\frac{C_{LH,fus}}{-\psi_{soil,sat} \times g} \times \frac{T_{soil} - T_{frz}}{T_{soil}}\right]^{\frac{-1}{B_{exp}}} \quad \text{(Eq. 3.6.460)}$$

$$W_{liq,supercool} = \min(F_k, \theta_{soil}) \quad \text{(Eq. 3.6.461)}$$

The amount of supercooled liquid water content ($M_{liq,supercool}$, [mm]) is computed as:

$$M_{liq,supercool} = W_{liq,supercool} \times D_{z,soil} \times 1000 \quad \text{(Eq. 3.6.462)}$$

where $D_{z,soil}$ [m] is the soil layer thickness.

2. Determine snow and soil melting/refreezing state:

First, define a phase change indicator (I_{phase}): 0-none;1-melt;2-refreeze.

(1) If snow/soil layer ice content is larger than 0 and snow/soil layer temperature is higher than freezing point, then the snow/soil layer ice is melting ($I_{phase} = 1$).

(2) If snow/soil layer liquid water content is greater than supercooled water content (note: snow supercooled water is zero), and snow/soil layer temperature is lower than freezing point, then ice is refreezing ($I_{phase} = 2$).

(3) If snow exists (SWE > 0) but its thickness is not enough to create an explicit snow layer ($I_{n,sno} = 0$, see also snow hydrology section below for details), and the first soil layer temperature is higher than freezing point, snow is melting ($I_{phase} = 1$).

3. Compute energy surplus and loss for melting and refreezing:

If $I_{phase} > 0$ (melting or refreezing active), the energy residual (surplus or loss) ($H_{M,phase}$, [W/m^2]) is computed for each snow/soil layer (i) as:

$$H_{M,phase}(i) = \frac{T_{snso}(i) - T_{frz}}{f_{phase}(i)} \quad \text{(Eq. 3.6.463)}$$

where f_{phase} is the phase change factor computed in GroundThermalPropertyMod.F90.

Then, the temperature is set to freezing point:

$$T_{snso}(i) = T_{frz} \quad \text{(Eq. 3.6.464)}$$

If $I_{phase}(i) = 1$ and $H_{M,phase}(i) < 0$, then

$$H_{M,phase}(i) = 0 \quad \text{(Eq. 3.6.465)}$$

$$I_{phase}(i) = 0 \quad \text{(Eq. 3.6.466)}$$

If $I_{phase}(i) = 2$ and $H_{M,phase}(i) > 0$, then

$$H_{M,phase}(i) = 0 \quad \text{(Eq. 3.6.467)}$$

$$I_{phase}(i) = 0 \quad \text{(Eq. 3.6.468)}$$

The amount of phase-change water (ΔW_{phase}, [kg/m^2]) is:

$$\Delta W_{phase}(i) = \frac{H_{M,phase}(i) \times \Delta t}{C_{LH,fus}} \quad \text{(Eq. 3.6.469)}$$

4. Compute rate of melting for snow without explicit layer (need more work):

If snow exists (SWE > 0) but its thickness is not enough to create an explicit snow layer ($I_{n,sno} = 0$, see also snow hydrology section below for details), and $\Delta W_{phase}(1) > 0$ (i.e., melting), then:

$$W_{snow,new} = \max(0, W_{snow,old} - \Delta W_{phase}(1)) \quad \text{(Eq. 3.6.470)}$$

$$h_{snow,new} = \max(0, h_{snow,old} \times \frac{W_{snow,new}}{W_{snow,old}}) \quad \text{(Eq. 3.6.471)}$$

where W_{snow} [mm] is the snow water equivalent (SWE) amount.

Snow height (h_{snow}, [m]) is limited to allow a reasonable snowpack density (50~500 kg/m^3):

$$h_{snow,new} = \min(\max\left(h_{snow,new}, \frac{W_{snow,new}}{500}\right), \frac{W_{snow,new}}{50}) \quad \text{(Eq. 3.6.472)}$$

Then, the updated energy residual ($H_{M,phase,res}$, [W/m^2]) is:

$$H_{M,phase,res} = H_{M,phase}(1) - C_{LH,fus} \times \frac{W_{snow,old} - W_{snow,new}}{\Delta t} \quad \text{(Eq. 3.6.473)}$$

If $H_{M,phase,res} > 0$ (still energy left for further melting), then:

$$\Delta W_{phase}(1) = \frac{H_{M,phase,res} \times \Delta t}{C_{LH,fus}} \quad \text{(Eq. 3.6.474)}$$

$$H_{M,phase}(1) = H_{M,phase,res} \quad \text{(Eq. 3.6.475)}$$

Otherwise,

$$\Delta W_{phase}(1) = 0 \quad \text{(Eq. 3.6.476)}$$

$$H_{M,phase}(1) = 0 \quad \text{(Eq. 3.6.477)}$$

The snow melting rate ($Q_{melt,sno}$, [mm/s]) is:

$$Q_{melt,sno} = \max(0, \frac{W_{snow,old} - W_{snow,new}}{\Delta t}) \quad \text{(Eq. 3.6.478)}$$

The total latent heat energy used for phase change ($E_{phase,snso}$, [W/m^2]):

$$E_{phase,snso} = C_{LH,fus} \times Q_{melt,sno} \quad \text{(Eq. 3.6.479)}$$

The melted snow here is assigned to a surface ponding term ($W_{ponding}$, [mm]) for later hydrology processes:

$$W_{ponding} = W_{snow,old} - W_{snow,new} \quad \text{(Eq. 3.6.480)}$$

5. Compute rate of melting and freezing for snow ($I_{n,sno} < 0$) and soil:

For each snow/soil layer, if melting or freezing is active ($I_{phase}(i) > 0$ and $|H_{M,phase}(i)| > 0$):

(1) If $\Delta W_{phase}(i) > 0$ (melting), then the snow/soil ice amount ($W_{ice,snso}$, [mm]) is updated as:

$$W_{ice,snso,new}(i) = \max(0, W_{ice,snso,old}(i) - \Delta W_{phase}(i)) \quad \text{(Eq. 3.6.481)}$$

The updated energy residual ($H_{M,phase,res}$, [W/m^2]) is:

$$H_{M,phase,res} = H_{M,phase}(i) - C_{LH,fus} \times \frac{W_{ice,snso,old}(i) - W_{ice,snso,new}(i)}{\Delta t} \quad \text{(Eq. 3.6.482)}$$

(2) If $\Delta W_{phase}(i) < 0$ (freezing), then:

For snow layers ($i \leq 0$),

$$W_{ice,snso,new}(i) = \min(W_{ice,snso,old}(i) + W_{liq,snso,old}(i), W_{ice,snso,old}(i) - \Delta W_{phase}(i)) \quad \text{(Eq. 3.6.483)}$$

For soil layers ($i > 0$),

If $W_{ice,snso,old}(i) + W_{liq,snso,old}(i) < M_{liq,supercool}$, then

$$W_{ice,snso,new}(i) = 0 \quad \text{(Eq. 3.6.484)}$$

If $W_{ice,snso,old}(i) + W_{liq,snso,old}(i) \geq M_{liq,supercool}$, then:

$$W_{ice,snso,new}(i) = \max(0, \ \min(W_{ice,snso,old}(i) + W_{liq,snso,old}(i) - M_{liq,supercool}, W_{ice,snso,old}(i) - \Delta W_{phase}(i))) \quad \text{(Eq. 3.6.485)}$$

The updated energy residual ($H_{M,phase,res}$, [W/m^2]) is:

$$H_{M,phase,res} = H_{M,phase}(i) - C_{LH,fus} \times \frac{W_{ice,snso,old}(i) - W_{ice,snso,new}(i)}{\Delta t} \quad \text{(Eq. 3.6.486)}$$

(3) Then, the snow/soil liquid water amount ($W_{liq,snso}$, [mm]) is updated as:

$$W_{liq,snso,new}(i) = \max(0, W_{ice,snso,old}(i) + W_{liq,snso,old}(i) - W_{ice,snso,new}(i)) \quad \text{(Eq. 3.6.487)}$$

(4) if there is still energy residual remaining (i.e., $|H_{M,phase,res}| > 0$), then further adjust snow/soil ice content and temperature (T_{snso}, [K]) as follows:

$$T_{snso}(i) = T_{snso}(i) + f_{phase}(i) \times H_{M,phase,res} \quad \text{(Eq. 3.6.488)}$$

For snow layers ($i \leq 0$),

If both snow ice and liquid exist (i.e., $W_{liq,snso,new}(i) \times W_{ice,snso,new}(i) > 0$), then:

$$T_{snso}(i) = T_{frz} \quad \text{(Eq. 3.6.489)}$$

If there is no snow ice (i.e., melt out entirely, $W_{ice,snso,new}(i) = 0$), then move the energy residual to the next layer below:

$$T_{snso}(i) = T_{frz} \quad \text{(Eq. 3.6.490)}$$

$$H_{M,phase}(i+1) = H_{M,phase}(i+1) + H_{M,phase,res} \quad \text{(Eq. 3.6.491)}$$

$$\Delta W_{phase}(i+1) = \frac{H_{M,phase}(i+1) \times \Delta t}{C_{LH,fus}} \quad \text{(Eq. 3.6.492)}$$

(5) The accumulated latent heat energy used for phase change ($E_{phase,snso}$, [W/m^2]) is:

$$E_{phase,snso} = E_{phase,snso} + C_{LH,fus} \times \frac{W_{ice,snso,old}(i) - W_{icesnso,new}(i)}{\Delta t} \quad \text{(Eq. 3.6.493)}$$

The accumulated snow melting rate ($Q_{melt,sno}$, [mm/s]) is ($i \leq 0$):

$$Q_{melt,sno} = Q_{melt,sno} + \max(0, \frac{W_{ice,snso,old}(i) - W_{ice,snso,new}(i)}{\Delta t}) \quad \text{(Eq. 3.6.494)}$$

6. Update snow and soil ice and water content:

For snow ice ($W_{ice,snow}$, [mm]) and liquid water content ($W_{liq,snow}$, [mm]):

$$W_{liq,snow}(i) = W_{liq,snso}(i) \qquad i \leq 0 \quad \text{(Eq. 3.6.495)}$$

$$W_{ice,snow}(i) = W_{ice,snso}(i) \qquad i \leq 0 \quad \text{(Eq. 3.6.496)}$$

For soil liquid water ($W_{liq,soil}$, ([m^3/m^3]) and total moisture content (θ_{soil} , [m^3/m^3]):

$$W_{liq,soil}(i) = \frac{W_{liq,snso}(i)}{1000 \times D_{zsnso}(i)} \qquad i > 0 \quad \text{(Eq. 3.6.497)}$$

$$\theta_{soil}(i) = \frac{W_{ice,snso}(i) + W_{liq,snso}(i)}{1000 \times D_{zsnso}(i)} \qquad i > 0 \quad \text{(Eq. 3.6.498)}$$

3.6.20 Heat Flux from Sprinkler Irrigation Evaporation

Description:

This part is to update surface sensible heat flux by removing the heat flux lost through sprinkler irrigation evaporation (only done when irrigation is activated over crop lands).

Relevant code modules:

Module: EnergyMainMod.F90

Subroutines: EnergyMain

Physics:

The total surface sensible heat flux ($H_{S,sfc}$, [W/m^2]) computed through the energy processes above will be subtracted by the latent heat loss through sprinkler irrigation evaporation ($H_{L,irr,sprinkler}$, [W/m^2]):

$$H_{S,sfc} = H_{S,sfc} - H_{L,irr,sprinkler} \quad \text{(Eq. 3.6.499)}$$

3.7 Hydrology

Description:

The Noah-MP main water (hydrology) module calculate all the hydrological processes, including canopy hydrology, snowpack hydrology, irrigation, soil infiltration, surface and subsurface runoff, and groundwater. The land water processes do not use the "tile" approach (i.e., vegetated and bared subgrid pixel) as the energy component. The grid-mean energy fluxes computed from the energy component are used to drive the grid water processes in Noah-MP.

Relevant code modules:

Module: WaterMainMod.F90

Subroutines: WaterMain

Physics:

3.7.1 Canopy Hydrology

Description:

This module is to compute canopy hydrology processes for intercepted rain and snow water, including canopy liquid water evaporation and dew, and canopy ice water sublimation and frost.

Relevant code modules:

Module: CanopyHydrologyMod.F90

Subroutines: CanopyHydrology

Physics:

1. Maximum canopy liquid and ice water (same as done in CanopyWaterInterceptMod.F90**):**

(1) Maximum liquid water held by canopy ($W_{liq,can,max}$, [mm]):

$$W_{liq,can,max} = W_{liq,can,holdcap} \times (E_{LAI} + E_{SAI}) \quad \text{(Eq. 3.7.1)}$$

where $W_{liq,can,holdcap}$ [mm] is the canopy holding capacity for liquid water as a Noah-MP table parameter depending on vegetation type, E_{LAI} is the effective leaf area index after burying by snow, and E_{SAI} is the effective stem area index after burying by snow. Both E_{LAI} and E_{SAI} are computed in the PhenologyMainMod.F90.

(2) Maximum ice/snow held by canopy ($W_{ice,can,max}$, [mm]):

$$W_{ice,can,max} = 6.6 \times (0.27 + \frac{46.0}{\rho_{snowfall}}) \times (E_{LAI} + E_{SAI}) \quad \text{(Eq. 3.7.2)}$$

where $\rho_{snowfall}$ [kg/m^3] is the bulk snowfall density (computed in AtmosForcingMod.F90).

The $W_{liq,can,max}$ and $W_{ice,can,max}$ are used in computing the wet fraction of canopy.

2. Canopy evaporation/sublimation, transpiration, and dew/frost:

If the canopy temperature is above freezing point ($T_v > T_{frz}$; i.e., FlagFrozenCanopy=.false.), then liquid water processes is activated:

$$q_{can,transp} = \max(0, \frac{H_{L,transp}}{C_{LH,evap}}) \quad \text{(Eq. 3.7.3)}$$

$$q_{can,evap} = \max(0, \frac{H_{L,can,evap}}{C_{LH,evap}}) \quad \text{(Eq. 3.7.4)}$$

$$q_{can,dew} = abs[\min\left(0, \frac{H_{L,can,evap}}{C_{LH,evap}}\right)] \quad \text{(Eq. 3.7.5)}$$

$$q_{can,subl} = 0.0 \quad \text{(Eq. 3.7.6)}$$

$$q_{can,frost} = 0.0 \quad \text{(Eq. 3.7.7)}$$

where $q_{can,transp}$, $q_{can,evap}$, $q_{can,dew}$ $q_{can,subl}$, and $q_{can,frost}$ [mm/s] are the canopy transpiration, evaporation, dew, sublimation, and frost water fluxes. $H_{L,transp}$ and $H_{L,can,evap}$ [W/m^2] are the latent heat fluxes (positive for evaporation and negative for dew; computed in EnergyMainMod.F90) at the canopy surface. $C_{LH,evap}$ [J/kg] is the specific latent heat of evaporation.

If the canopy temperature is below freezing point ($T_v \leq T_{frz}$; i.e., FlagFrozenCanopy=.true.), then ice processes is activated:

$$q_{can,transp} = \max(0, \frac{H_{L,transp}}{C_{LH,subl}}) \quad \text{(Eq. 3.7.8)}$$

$$q_{can,evap} = 0.0 \quad \text{(Eq. 3.7.9)}$$

$$q_{can,dew} = 0.0 \quad \text{(Eq. 3.7.10)}$$

$$q_{can,subl} = \max(0, \frac{H_{L,can,evap}}{C_{LH,subl}}) \quad \text{(Eq. 3.7.11)}$$

$$q_{can,frost} = abs[\min\left(0, \frac{H_{L,can,evap}}{C_{LH,subl}}\right)] \quad \text{(Eq. 3.7.12)}$$

where $C_{LH,subl}$ [J/kg] is the specific latent heat of sublimation.

3. Canopy water balance

It is most convenient to allow dew or frost to bring canopy liquid water ($W_{liq,can}$, [mm]) or ice ($W_{ice,can}$, [mm]) above the maximum canopy liquid water or ice holding capacity (which is done in the current Noah-MP). Otherwise, the canopy drip (computed in CanopyWaterInterceptMod.F90) needs to be re-adjusted.

$$q_{can,evap} = \min(q_{can,evap}, \frac{W_{liq,can}}{\Delta t}) \quad \text{(Eq. 3.7.13)}$$

$$q_{can,subl} = \min(q_{can,subl}, \frac{W_{ice,can}}{\Delta t}) \quad \text{(Eq. 3.7.14)}$$

The canopy liquid water and ice content are updated as follows:

$$W_{liq,can} = \max(0.0, (q_{can,dew} - q_{can,evap}) \times \Delta t) \quad \text{(Eq. 3.7.15)}$$

$$W_{ice,can} = \max(0.0, (q_{can,frost} - q_{can,subl}) \times \Delta t) \quad \text{(Eq. 3.7.16)}$$

If $W_{liq,can}$ or $W_{ice,can}$ [mm] is less than 10^{-6} [mm], then they are reset to 0.

Note that the canopy intercepted liquid water and ice amount have been added to canopy liquid water and ice content in CanopyWaterInterceptMod.F90.

4. Canopy wetted fraction (same as done in CanopyWaterInterceptMod.F90**):**

The canopy wet fraction (F_{wet}) is re-computed/updated here based on the updated canopy ice or liquid water content:

$$F_{wet} = \begin{cases} \frac{\max(0.0,\ W_{ice,can})}{\max(W_{ice,can,max},\ 10^{-6})}, & W_{ice,can} > 0.0 \\ \frac{\max(0.0,\ W_{liq,can})}{\max(W_{liq,can,max},\ 10^{-6})}, & W_{ice,can} \leq 0.0 \end{cases} \quad \text{(Eq. 3.7.17)}$$

$$F_{wet} = (\min(1.0, F_{wet}))^{0.667} \quad \text{(Eq. 3.7.18)}$$

5. Canopy phase change

The total canopy water ($W_{can,tot}$, [mm]):

$$W_{can,tot} = W_{liq,can} + W_{ice,can} \quad \text{(Eq. 3.7.19)}$$

(1) Canopy ice melting: when canopy ice exists ($W_{ice,can} > 10^{-6}$ [mm]) and canopy temperature is above freezing point ($T_v > T_{frz}$):

$$q_{can,melt} = \min\left(\frac{W_{ice,can}}{\Delta t},\ (T_v - T_{frz}) \times \frac{C_{ice} \times W_{ice,can}}{\rho_{ice} \times C_{LH,fus} \times \Delta t}\right) \quad \text{(Eq. 3.7.20)}$$

$$W_{ice,can} = \max(0.0,\ W_{ice,can} - q_{can,melt} \times \Delta t) \quad \text{(Eq. 3.7.21)}$$

$$W_{liq,can} = \max(0.0,\ W_{can,tot} - W_{ice,can}) \quad \text{(Eq. 3.7.22)}$$

$$T_v = F_{wet} \times T_{frz} + (1 - F_{wet}) \times T_v \quad \text{(Eq. 3.7.23)}$$

where $q_{can,melt}$ [mm/s] is the canopy ice melting rate, C_{ice} [J/m^3/K] is the specific heat capacity of ice, ρ_{ice} [kg/m^3] is the density of ice, and $C_{LH,fus}$ [J/kg] is the specific latent heat of fusion.

(2) Canopy liquid water freezing: when canopy liquid water exists ($W_{liq,can} > 10^{-6}$ [mm])) and canopy temperature is below freezing point ($T_v < T_{frz}$):

$$q_{can,frz} = \min\left(\frac{W_{liq,can}}{\Delta t},\ (T_{frz} - T_v) \times \frac{C_{wat} \times W_{liq,can}}{\rho_{wat} \times C_{LH,fus} \times \Delta t}\right) \quad \text{(Eq. 3.7.24)}$$

$$W_{liq,can} = \max(0.0,\ W_{liq,can} - q_{can,frz} \times \Delta t) \quad \text{(Eq. 3.7.25)}$$

$$W_{ice,can} = \max(0.0,\ W_{can,tot} - W_{liq,can}) \quad \text{(Eq. 3.7.26)}$$

$$T_v = F_{wet} \times T_{frz} + (1 - F_{wet}) \times T_v \quad \text{(Eq. 3.7.27)}$$

where $q_{can,frz}$ [mm/s] is the canopy liquid water freezing rate, C_{wat} [J/m^3/K] is the specific heat capacity of liquid water, and ρ_{wat} [kg/m^3] is the density of liquid water.

6. Update total canopy water and net evaporation:

The total canopy water ($W_{can,tot}$, [mm]):

$$W_{can,tot} = W_{liq,can} + W_{ice,can} \quad \text{(Eq. 3.7.28)}$$

The net canopy evaporation ($q_{can,evap,net}$, [mm/s]):

$$q_{can,evap,net} = q_{can,evap} + q_{can,subl} - q_{can,dew} - q_{can,frost} \quad \text{(Eq. 3.7.29)}$$

3.7.2 Ground (snowpack/soil) Surface Water Fluxes

Description:

This part is to compute the ground (soil/snow) surface sublimation, frost, evaporation, and dew water fluxes, which will be used in the subsequent snowpack and soil hydrological processes.

Relevant code modules:

Module: WaterMainMod.F90

Subroutines: WaterMain

Physics:

The ground (soil/snow) surface evaporation/sublimation and dew/frost rate are computed using the energy flux from the energy component:

$$Q_{vap} = \max(0.0, \frac{H_{L,grd}}{C_{LH,vap}}) \quad \text{(Eq. 3.7.30)}$$

$$Q_{cond} = abs\left[\min\left(0.0, \frac{H_{L,grd}}{C_{LH,vap}}\right)\right] \quad \text{(Eq. 3.7.31)}$$

$$Q_{net,grd} = Q_{vap} - Q_{cond} \quad \text{(Eq. 3.7.32)}$$

where Q_{vap} , Q_{cond} , and $Q_{net,grd}$ [mm/s] are the vaporization (evaporation/sublimation), condensation (dew/frost), and net water fluxes at the ground (soil/snow) surface. $H_{L,grd}$ [W/m^2] is the ground surface latent heat fluxes (positive for vaporization and negative for condensation;

computed in EnergyMainMod.F90) and $C_{LH,vap}$ [J/kg] is the specific latent heat of vaporization (evaporation/sublimation) at the ground surface (computed in PsychrometricVariableMod.F90). The snow and soil surface are differentiated based on SWE (W_{snow}, [mm]):

$$Q_{sno,subl} = \begin{cases} \min\left(Q_{vap}, \frac{W_{snow}}{\Delta t}\right), & W_{snow} > 0 \\ 0.0, & W_{snow} \le 0 \end{cases} \quad \text{(Eq. 3.7.33)}$$

$$Q_{soil,evap} = Q_{vap} - Q_{sno,subl} \quad \text{(Eq. 3.7.34)}$$

where $Q_{sno,subl}$ [mm/s] is the snowpack surface sublimation rate, and $Q_{soil,evap}$ [mm/s] is the soil surface evaporation rate.

$$Q_{sno,frost} = \begin{cases} Q_{cond}, & W_{snow} > 0 \\ 0.0, & W_{snow} \le 0 \end{cases} \quad \text{(Eq. 3.7.35)}$$

$$Q_{soil,dew} = Q_{cond} - Q_{sno,frost} \quad \text{(Eq. 3.7.36)}$$

where $Q_{sno,frost}$ [mm/s] is the snowpack surface frost rate, and $Q_{soil,dew}$ [mm/s] is the soil surface dew rate.

3.7.3 Snow Hydrology

Description:

This multi-layer snow module is to compute the evolution of snowpack (up to 3 layers) properties, including snow ice and liquid water content, snow thickness, and water flux out of snowpack bottom. The snow hydrological processes include snowfall after canopy interception, snowpack compaction, snow layer combination, snow layer division, and snow water state updates. These processes are driven by the energy fluxes and snow temperature changes computed in EnergyMainMod.F90. The change in snowpack SWE is balanced by the input snowfall and frost and the output snowmelt and sublimation. Only snowpack with explicit snow layers (snow depth >= 2.5cm) will go through snow layer compaction, combination, and division processes.

Relevant code modules:

Module: SnowWaterMainMod.F90

Subroutine: SnowWaterMain

Physics:

Overall snowpack water balance:

$$\frac{\Delta W_{snow}}{\Delta t} = Q_{snow} + Q_{sno,frost} - Q_{sno,melt} - Q_{sno,subl} \quad \text{(Eq. 3.7.37)}$$

where Q_{snow} [mm/s] is the snowfall rate below canopy on the snowpack surface, and $Q_{sno,melt}$ [mm/s] is the total melting snow water driven by snowpack surface energy balance:

$$Q_{sno,melt} = \frac{1}{C_{LH,fus}}(S_{W,grd,net} + L_{W,grd,net} + H_{pr,grd} - H_{L,grd} - H_{S,grd} - G_{sno2soil})$$

(Eq. 3.7.38)

where $C_{LH,fus}$ [J/kg] is the specific latent heat of fusion. $S_{W,grd,net}$, $L_{W,grd,net}$, $H_{pr,grd}$, $H_{L,grd}$, and $H_{S,grd}$ [W/m^2] are the net shortwave radiation flux, net longwave radiation flux, precipitation advected heat flux, latent heat flux, and sensible heat flux at the snowpack surface. $G_{sno2soil}$ [W/m^2] is the heat flux advected from snowpack to soil at the snow-soil interface. Note that current Noah-MP solves the snow and soil layer temperature together using a tri-diagonal matrix method, and hence $G_{sno2soil}$ is not explicitly represented in the model code. Current Noah-MP neglects heat transfer by movement of meltwater within the snowpack.

The Noah-MP code convention for snow layer index is as follows: the snow layer index for vector/matrix dimension starts from -N_{snow}+1 to 0, where N_{snow} is the maximum number of snow layers (currently is 3 in Noah-MP). So the snow vector quantities have the column dimension of -2:0 with -2 for top layer index and 0 for bottom layer index. The purpose of this negative indexing of snow vector column dimension is to have a continuous index for combined snow and soil layers (i.e., soil layer index starts from 1 to 4 for 4-layer soil). There is another snow layer indicator ($I_{n,sno}$) in Noah-MP to identify the actual number of snow layers:

$$I_{n,sno} = \begin{cases} -3, & 3\ layer\ snow \\ -2, & 2\ layer\ snow \\ -1, & 1\ layer\ snow \\ 0, & 0\ layer\ snow \end{cases} \quad \text{(Eq. 3.7.39)}$$

Thus, for loops over multiple snow layers, the index for vector column dimension starts at $I_{n,sno} + 1$ (top snow layer) and ends at 0 (bottom snow layer).

Note that for shallow snowpack without explicit snow layers (i.e., snow depth < 2.5 cm; $I_{n,sno} = 0$), the prognostic snow state variables are snow water equivalent (W_{snow}, [mm]) and snow depth (h_{snow}, [m]). For snowpack with explicit snow layers (i.e., snow depth >= 2.5 cm;

$I_{n,sno} < 0$), the prognostic snow state variables are snow layer liquid water and ice content ($W_{ice,sno}(i)$ and $W_{liq,sno}(i)$, i =0~2, is the snow layer index) and snow layer thickness ($D_{zsnso}(i)$, i =0~2, is the snow layer index), while W_{snow} and h_{snow} are diagnostic variables derived from $W_{ice,sno}$ and $W_{liq,sno}$ and D_{zsnso}.

3.7.3.1 Snowfall Below Canopy

Description:

This module is to update snow water equivalent (W_{snow}, [mm]) and snow depth (h_{snow}, [m]) due to snowfall below canopy computed from CanopyWaterInterceptMod.F90.

Relevant code modules:

Module: SnowfallBelowCanopyMod.F90

Subroutines: SnowfallAfterCanopyIntercept

Physics:

1. For shallow (snow depth < 2.5cm) snow without explicit snow layers (i.e., $I_{n,sno} = 0$),

$$h_{snow} = h_{snow} + \Delta h_{snow} \times \Delta t \quad \text{(Eq. 3.7.40)}$$

$$W_{snow} = W_{snow} + Q_{snow} \times \Delta t \quad \text{(Eq. 3.7.41)}$$

where h_{snow} [m] is the snow depth, and W_{snow} [mm] is the snow water equivalent. Q_{snow} [mm/s] is the snowfall rate below canopy on the ground, and Δh_{snow} [mm/s] is the snow depth increasing rate due to snowfall. Q_{snow} and Δh_{snow} are computed in CanopyWaterInterceptMod.F90.

If after snowfall, snow depth is larger than 2.5 cm, then creating a new layer:

$$I_{n,sno} = -1 \quad \text{(Eq. 3.7.42)}$$

$$D_{zsnso}(0) = h_{snow} \quad \text{(Eq. 3.7.43)}$$

$$h_{snow} = 0.0 \quad \text{(Eq. 3.7.44)}$$

$$T_{snso}(0) = \min(273.16, T_{sfc}) \quad \text{(Eq. 3.7.45)}$$

$$W_{ice,sno}(0) = W_{snow} \quad \text{(Eq. 3.7.46)}$$

$$W_{liq,sno}(0) = 0.0 \quad \text{(Eq. 3.7.47)}$$

Here h_{snow} is reset to 0 because h_{snow} becomes a diagnostic state variable for snowpack with explicit snow layers and will be updated based on D_{zsnso} later. Note that all new snowfall is added to the top snow as snow ice content.

2. For snowpack with explicit snow layers ($I_{n,sno} < 0$, i.e., snow depth >= 2.5cm):

$$D_{zsnso}\left(I_{n,sno}+1\right) = D_{zsnso}\left(I_{n,sno}+1\right) + \Delta h_{snow} \times \Delta t \qquad \text{(Eq. 3.7.48)}$$

$$W_{ice,sno}\left(I_{n,sno}+1\right) = W_{ice,sno}\left(I_{n,sno}+1\right) + Q_{snow} \times \Delta t \qquad \text{(Eq. 3.7.49)}$$

Note that $I_{n,sno}+1$ is the snow layer index for top snow and all new snowfall is added to the top snow as snow ice content.

3.7.3.2 Snowpack Compaction

Description:

This module is to update snow layer thickness via compaction due to destructive metamorphism, overburden, & melt. This module is only active for snowpack with explicit snow layers (snow depth >= 2.5 cm; $I_{n,sno} < 0$). The parameterizations are based on Anderson (1976).

Relevant code modules:

Module: SnowpackCompactionMod.F90

Subroutine: SnowpackCompaction

Physics:

The snow compaction process is only activated for snow layer ice content > 0.1 mm and void fraction ($V_{oid,sno}$) > 0.001 for each snow layer (i):

$$V_{oid,sno}(i) = 1.0 - \frac{\left(\frac{W_{ice,sno}(i)}{\rho_{ice}} + \frac{W_{liqd,sno}(i)}{\rho_{wat}}\right)}{D_{zsnso}(i)} \qquad \text{(Eq. 3.7.50)}$$

where ρ_{wat} and ρ_{ice} [kg/m^3] are the density of liquid water and ice, respectively.

1. Snow compaction due to destructive metamorphism/aging:

The snow layer compaction rate ($D_{DZ,dm}$, [1/s]) due to destructive metamorphism ("dm"):

$$D_{DZ,dm} = -C_{dm,1} \times e^{[-C_{dm,2} \times \max(0, T_{frz} - T_{snso}(i))]} \quad \text{(Eq. 3.7.51)}$$

where i (=-2~0) is the snow layer index, $C_{dm,1}$ [1/s] and $C_{dm,2}$ [1/K] are the compaction parameters due to destructive metamorphism, which are set in the NoahmpTable.TBL.

An upper limit ($D_{M,max}$, [kg/m^3]) of snow density for destructive metamorphism compaction is also set in NoahmpTable.TBL. If the partial snow ice density is larger than this upper limit, the compaction rate is updated as follows:

If $\frac{W_{ice,sno}(i)}{D_{zsnso}(i)} > D_{M,max}$, then

$$D_{DZ,dm} = D_{DZ,dm} \times e^{[-0.046 \times (\frac{W_{ice,sno}(i)}{D_{zsnso}(i)} - D_{M,max})]} \quad \text{(Eq. 3.7.52)}$$

The liquid water effect is also accounted for:

If $W_{liq,sno}(i) > 0.01 \times D_{zsnso}(i)$, then

$$D_{DZ,dm} = D_{DZ,dm} \times C_{dm,3} \quad \text{(Eq. 3.7.53)}$$

$C_{dm,3}$ is a compaction parameter due to destructive metamorphism set in the NoahmpTable.TBL.

2. Snow compaction due to overburden:

The snow layer compaction rate ($D_{DZ,bd}$, [1/s]) due to overburden ("bd"):

$$D_{DZ,bd} = -(W_{abv} + 0.5 \times W_{sno}(i)) \times \frac{e^{[-0.08 \times \max(0, T_{frz} - T_{snso}(i)) - C_{bd} \times \frac{W_{ice,sno}(i)}{D_{zsnso}(i)}]}}{\eta_0} \quad \text{(Eq. 3.7.54)}$$

$$W_{sno}(i) = W_{ice,sno}(i) + W_{liq,sno}(i) \quad \text{(Eq. 3.7.55)}$$

$$W_{abv} = \sum_{k=top}^{i} W_{sno}(k) \quad \text{(Eq. 3.7.56)}$$

where W_{abv} [mm] is the total snow burden for snow layers above the current layer, and $W_{sno}(i)$ [mm] is the snow burden of the current layer. C_{bd} [m^3/kg] is a compaction parameter due to overburden set in the NoahmpTable.TBL. η_0 [kg s/m^2] is the snow viscosity coefficient at the freezing point set in the NoahmpTable.TBL optimized by He et al. (2021).

3. Snow compaction due to melt:

If snowpack is undergoing melting, the snow layer compaction rate ($D_{DZ,melt}$, [1/s]) due to melt:

$$D_{DZ,melt} = -\frac{\max(0, \frac{(F_{ice,old}(i) - F_{ice}(i))}{\max(10^{-6}, F_{ice,old}(i))})}{\Delta t} \quad \text{(Eq. 3.7.57)}$$

$$F_{ice}(i) = \frac{W_{ice,sno}(i)}{W_{ice,sno}(i) + W_{liq,sno}(i)} \quad \text{(Eq. 3.7.58)}$$

Where $F_{ice,old}$ and F_{ice} are the snowpack ice fraction for each snow layer from last and current timestep.

4. Change of snow layer thickness due to compaction:

The total fractional change ($\Delta F_{DZ,sno}$; [1/s]) of snow layer thickness

$$\Delta F_{DZ,sno} = \max(-0.5, (D_{DZ,dm} + D_{DZ,bd} + D_{DZ,melt}) \times \Delta t) \qquad \text{(Eq. 3.7.59)}$$

Update snow layer thickness:

$$D_{zsnso}(i) = D_{zsnso}(i) \times (1 + \Delta F_{DZ,sno}) \qquad \text{(Eq. 3.7.60)}$$

$$D_{zsnso}(i) = \max(D_{zsnso}(i), \frac{W_{ice,sno}(i)}{\rho_{ice}} + \frac{W_{liq,sno}(i)}{\rho_{wat}}) \qquad \text{(Eq. 3.7.61)}$$

where *i* (=-2~0) is the snow layer index.

Further limit snow layer thickness to allow a reasonable snowpack density (50~500 kg/m^3):

$$D_{zsnso}(i) = \min(\max\left(D_{zsnso}(i), \frac{W_{ice,sno}(i)+W_{liq,sno}(i)}{500}\right), \frac{W_{ice,sno}(i)+W_{liq,sno}(i)}{50}) \qquad \text{(Eq. 3.7.62)}$$

3.7.3.3 Snow Layer Combination

Description:

This module is to compute snowpack layer combination processes to update snow ice, snow liquid water, snow thickness, and snow temperature. Currently, the minimum thicknesses for each snow layer from the top to the bottom layers are 2.5 cm, 2.5 cm, and 10 cm, respectively. The layer combination will be based on this minimum layer thickness threshold. This module is only active for snowpack with explicit snow layers ($Ind_{sno} < 0$).

Relevant code modules:

Module: SnowLayerCombineMod.F90

Subroutine: SnowLayerCombine

Module: SnowLayerWaterComboMod.F90

Subroutine: SnowLayerWaterCombo

Physics:

1. Combine snow layers with very low snow ice content ($W_{ice,sno}$ < 0.1 mm):

(1) if not the bottom snow layer, the current snow layer (indexed as *i*) is combined with the layer below (indexed as *i*+1):

$$W_{ice,sno}(i+1) = W_{ice,sno}(i+1) + W_{ice,sno}(i) \quad \text{(Eq. 3.7.63)}$$

$$W_{liq,sno}(i+1) = W_{liq,sno}(i+1) + W_{liq,sno}(i) \quad \text{(Eq. 3.7.64)}$$

$$D_{zsnso}(i+1) = D_{zsnso}(i+1) + D_{zsnso}(i) \quad \text{(Eq. 3.7.65)}$$

(2) if the bottom snow layer (*i* = 0),

For 3-layer or 2-layer snowpack ($I_{n,sno} < -1$), the bottom snow layer (indexed as *i*=0) is combined with the layer above (indexed as i-1):

$$W_{ice,sno}(-1) = W_{ice,sno}(-1) + W_{ice,sno}(0) \quad \text{(Eq. 3.7.66)}$$

$$W_{liq,sno}(-1) = W_{liq,sno}(-1) + W_{liq,sno}(0) \quad \text{(Eq. 3.7.67)}$$

$$D_{zsnso}(-1) = D_{zsnso}(-1) + D_{zsnso}(0) \quad \text{(Eq. 3.7.68)}$$

For 1-layer snowpack ($I_{n,sno} = -1$), because of the very low snow ice content, the 1-layer snowpack gets set to 0-layer snow (i.e., no explicit snow layer).

If $W_{ice,sno}(0)$ is larger than 0 (snow survives earlier sublimation), snow liquid water is assumed to be ponded on soil surface due to reset to no explicit snow layer:

$$W_{ponding,1} = W_{liq,sno}(0) \quad \text{(Eq. 3.7.69)}$$

$$W_{snow} = W_{ice,sno}(0) \quad \text{(Eq. 3.7.70)}$$

$$h_{snow} = D_{zsnso}(0) \quad \text{(Eq. 3.7.71)}$$

If snow ice content is smaller than 0 (snow ice over-sublimated earlier), all snow liquid water and ice are assumed to be ponded on soil surface:

$$W_{ponding,1} = W_{liq,sno}(0) + W_{ice,sno}(0) \quad \text{(Eq. 3.7.72)}$$

$$W_{snow} = 0 \quad \text{(Eq. 3.7.73)}$$

$$h_{snow} = 0 \quad \text{(Eq. 3.7.74)}$$

If $W_{ponding,1}$ < 0 (all snow is over-sublimated), the remaining over-sublimated water will be taken out of top soil layer ice:

$$W_{ice,soil}(1) = W_{ice,soil}(1) + \frac{W_{ponding,1}}{D_{zsnso}(1) \times 1000} \quad \text{(Eq. 3.7.75)}$$

$$W_{ponding,1} = 0 \quad \text{(Eq. 3.7.76)}$$

Then, for 1-layer snowpack with very low snow ice (no matter $W_{ice,sno}(0)$ > 0 or not), the following bottom snow state variables are reset to 0 due to the reset to no explicit snow layer:

$$W_{ice,sno}(0) = 0 \quad \text{(Eq. 3.7.77)}$$

$$W_{liq,sno}(0) = 0 \quad \text{(Eq. 3.7.78)}$$

$$D_{zsnso}(0) = 0 \quad \text{(Eq. 3.7.79)}$$

(3) For 3-layer and 2-layer snowpack, all snow quantities for non-bottom layers are shifted down by one layer due to the layer combination above:

$$W_{ice,sno}(i) = W_{ice,sno}(i-1) \quad \text{(Eq. 3.7.80)}$$

$$W_{liq,sno}(i) = W_{liq,sno}(i-1) \quad \text{(Eq. 3.7.81)}$$

$$D_{zsnso}(i) = D_{zsnso}(i-1) \quad \text{(Eq. 3.7.82)}$$

$$T_{snso}(i) = T_{snso}(i-1) \quad \text{(Eq. 3.7.83)}$$

where the current snow layer is indexed as i and the layer above is indexed as i-1.

The snow layer number indicator ($I_{n,sno}$) is also updated whenever two snow layers are combined:

$$I_{n,sno} = I_{n,sno} + 1 \quad \text{(Eq. 3.7.84)}$$

2. Conserving water for over-sublimated snow:

Because the over-sublimated snow water is taken out of the soil ice above, when soil ice becomes lower than zero ($W_{ice,soil}(1) < 0$; soil ice is also not sufficient for sublimation), then the remaining water is further taken out of top layer soil liquid water:

$$W_{liq,soil}(1) = W_{liq,soil}(1) + W_{ice,soil}(1) \quad \text{(Eq. 3.7.85)}$$

$$W_{ice,soil}(1) = 0 \quad \text{(Eq. 3.7.86)}$$

3. Update snow quantities after combination of layers with very low snow ice above:

If no explicit snow layer exists ($I_{n,sno}$=0) after combination above, then no further processes will be done at this point.

If there is still explicit snow layer(s) ($I_{n,sno}$<0), then

$$W_{snow} = \sum_{i=I_{n,sno}+1}^{0} (W_{ice,sno}(i) + W_{liq,sno}(i)) \quad \text{(Eq. 3.7.87)}$$

$$h_{snow} = \sum_{i=I_{n,sno}+1}^{0} D_{zsnso}(i) \quad \text{(Eq. 3.7.88)}$$

If h_{snow} < 2.5 cm (minimum thickness for top snow layer) and $I_{n,sno}$<0, then the snowpack is reset to no explicit layer and all snow liquid water is assumed to be ponded on soil surface:

$$W_{snow} = \sum_{i=I_{n,sno}+1}^{0} W_{ice,sno}(i) \quad \text{(Eq. 3.7.89)}$$

$$W_{ponding,2} = \sum_{i=I_{n,sno}+1}^{0} W_{liq,sno}(i) \quad \text{(Eq. 3.7.90)}$$

$$I_{n,sno} = 0 \quad \text{(Eq. 3.7.91)}$$

4. Combine layers for multi-layer snowpack based on minimum layer thickness:

As mentioned above, the minimum snow layer thicknesses ($D_{z,sno,min}$) from the top to the bottom layers are 2.5 cm, 2.5 cm, and 10 cm, respectively. For multi-layer snowpack, if a certain snow layer thickness is less than this minimum threshold, it will be combined with the layer below, except for the bottom layer which will be combined with the layer above. For middle snow layers, if the combined thickness of the current middle layer and the layer above is smaller than the combined thickness of the current middle layer and the layer below, then the middle layer will be combined with the layer above.

The specific combination process for the thickness, snow ice, snow water, and snow temperature of two neighboring layers is done in SnowLayerWaterComboMod.F90. To combine layer thickness, snow ice, and snow water, it is simply the sum of those in the two neighboring layers. To combine snow temperature, Noah-MP conserves the enthalpy of the snow ice and water in the two neighboring layers before and after combination. Thus, the combined snow temperature is an average weighted by the enthalpy of each layer.

After layer combination, all snow quantities for non-bottom layers are shifted down by one layer:

$$W_{ice,sno}(i) = W_{ice,sno}(i-1) \quad \text{(Eq. 3.7.92)}$$

$$W_{liq,sno}(i) = W_{liq,sno}(i-1) \quad \text{(Eq. 3.7.93)}$$

$$D_{zsnso}(i) = D_{zsnso}(i-1) \quad \text{(Eq. 3.7.94)}$$

$$T_{snso}(i) = T_{snso}(i-1) \quad \text{(Eq. 3.7.95)}$$

where the current snow layer is indexed as i and the layer above is indexed as i-1.

The snow layer number indicator ($I_{n,sno}$) is further updated whenever two snow layers are combined:

$$I_{n,sno} = I_{n,sno} + 1 \quad \text{(Eq. 3.7.96)}$$

If after combination, $I_{n,sno}$ >= -1 (i.e., no more than one-layer snow), then no further combination process will be done.

3.7.3.4 Snow Layer Division

Description:

This module is to compute snowpack layer division process to update snow ice, snow liquid water, snow thickness, and snow temperature. Currently, the maximum thickness for each snow layer from the top to the bottom layers are 5 cm, 20 cm, and no limit, respectively. The layer division will be based on this maximum layer thickness threshold. This module is only active for snowpack with explicit snow layers ($I_{n,sno} < 0$).

Relevant code modules:

Module: SnowLayerDivideMod.F90

Subroutine: SnowLayerDivide

Module: SnowLayerWaterComboMod.F90

Subroutine: SnowLayerWaterCombo

Physics:

1. For 1-layer snowpack ($I_{n,sno} = -1$):

If the snow layer thickness is larger than 5 cm, then the layer is divided equally into two identical layers (i.e., snowpack becomes 2 layers):

$$I_{n,sno} = -2 \quad \text{(Eq. 3.7.97)}$$

$$D_{zsnso}(0) = \frac{D_{zsnso}(0)}{2} \quad \text{(Eq. 3.7.98)}$$

$$W_{ice,sno}(0) = \frac{W_{ice,sno}(0)}{2} \quad \text{(Eq. 3.7.99)}$$

$$W_{liq,sno}(0) = \frac{W_{liq,sno}(0)}{2} \quad \text{(Eq. 3.7.100)}$$

$$D_{zsnso}(-1) = D_{zsnso}(0) \quad \text{(Eq. 3.7.101)}$$

$$W_{ice,sno}(-1) = W_{ice,sno}(0) \quad \text{(Eq. 3.7.102)}$$

$$W_{liq,sno}(-1) = W_{liq,sno}(0) \quad \text{(Eq. 3.7.103)}$$

$$T_{snso}(-1) = T_{snso}(0) \quad \text{(Eq. 3.7.104)}$$

After this division, snowpack becomes 2 layers and hence will continue to go through the 2-layer snowpack division process (if meet certain threshold) as described in the following subsection.

2. For 2-layer snowpack ($I_{n,sno} = -2$):

If the top layer thickness is larger than 5 cm, then the top layer is divided in a way that the top layer is 5 cm and the remaining is merged with the layer below:

$$D_{z,sno,remain} = D_{zsnso}(-1) - 0.05 \quad \text{(Eq. 3.7.105)}$$

$$W_{ice,sno,remain} = W_{ice,sno}(-1) \times \frac{D_{z,sno,remain}}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.106)}$$

$$W_{liq,sno,remain} = W_{liq,sno}(-1) \times \frac{D_{z,sno,remain}}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.107)}$$

$$T_{snso,remain} = T_{snso}(-1) \quad \text{(Eq. 3.7.108)}$$

$$D_{zsnso}(-1) = 0.05 \quad \text{(Eq. 3.7.109)}$$

$$W_{ice,sno}(-1) = W_{ice,sno}(-1) \times \frac{0.05}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.110)}$$

$$W_{liq,sno}(-1) = W_{liq,sno}(-1) \times \frac{0.05}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.111)}$$

Then, the remaining part (i.e., $D_{z,sno,remain}$, $W_{ice,sno,remain}$, $W_{liq,sno,remain}$, $T_{snso,remain}$) after top layer division will be further combined with the layer below (i.e., bottom layer for 2-layer snowpack) based on the combination process in SnowLayerWaterComboMod.F90. If after this combination, the bottom layer thickness is larger than 20 cm, then the bottom layer is further divided equally into two layers (i.e., snowpack becomes 3 layers) and the original top layer index is shifted up one layer (i.e., -1 index becomes -2 due to adding a third layer at the bottom):

$$D_{zsnso}(-2) = D_{zsnso}(-1) \quad \text{(Eq. 3.7.112)}$$

$$W_{ice,sno}(-2) = W_{ice,sno}(-1) \quad \text{(Eq. 3.7.113)}$$

$$W_{liq,sno}(-2) = W_{liq,sno}(-1) \quad \text{(Eq. 3.7.114)}$$

$$T_{snso}(-2) = T_{snso}(-1) \quad \text{(Eq. 3.7.115)}$$

The division of the original bottom layer:

$$I_{n,sno} = -3 \quad \text{(Eq. 3.7.116)}$$

$$\frac{dT}{dz}_{sno} = \frac{T_{snso}(-1) - T_{snso}(0)}{\frac{1}{2}(D_{zsnso}(-1) + D_{zsnso}(0))} \quad \text{(Eq. 3.7.117)}$$

$$D_{zsnso}(0) = \frac{D_{zsnso}(0)}{2} \quad \text{(Eq. 3.7.118)}$$

$$W_{ice,sno}(0) = \frac{W_{ice,sno}(0)}{2} \quad \text{(Eq. 3.7.119)}$$

$$W_{liq,sno}(0) = \frac{W_{liq,sno}(0)}{2} \quad \text{(Eq. 3.7.120)}$$

$$D_{zsnso}(-1) = D_{zsnso}(0) \quad \text{(Eq. 3.7.121)}$$

$$W_{ice,sno}(-1) = W_{ice,sno}(0) \quad \text{(Eq. 3.7.122)}$$

$$W_{liq,sno}(-1) = W_{liq,sno}(0) \quad \text{(Eq. 3.7.123)}$$

The snow temperature is divided to keep the same temperature layer gradient ($\frac{dT}{dz}_{sno}$):

$$T_{snso}(0) = T_{snso}(-1) - \frac{D_{zsnso}(-1)}{2} \times \frac{dT}{dz}_{sno} \quad \text{(Eq. 3.7.124)}$$

If $T_{snso}(0) \geq T_{frz}$, then

$$T_{snso}(0) = T_{snso}(-1) \quad \text{(Eq. 3.7.125)}$$

If $T_{snso}(0) < T_{frz}$, then

$$T_{snso}(-1) = T_{snso}(-1) + \frac{D_{zsnso}(-1)}{2} \times \frac{dT}{dz}_{sno} \quad \text{(Eq. 3.7.126)}$$

After this division, snowpack becomes 3 layers and hence will continue to go through the 3-layer snowpack division process (if certain thresholds are met) as described in the following subsection.

3. For 3-layer snowpack ($I_{n,sno} = -3$):

If the top layer thickness is larger than 5 cm, then the top layer will be divided in the same way as described in the 2-layer snowpack division process above.

If the middle snow layer thickness is larger than 20 cm, then the middle layer will be divided in a way that the middle layer is 20 cm and the remaining is merged with the layer below (i.e., bottom layer):

$$D_{z,sno,remain} = D_{zsnso}(-1) - 0.2 \quad \text{(Eq. 3.7.127)}$$

$$W_{ice,sno,remain} = W_{ice,sno}(-1) \times \frac{D_{z,sno,remain}}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.128)}$$

$$W_{liq,sno,remain} = W_{liq,sno}(-1) \times \frac{D_{z,sno,remain}}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.129)}$$

$$T_{snso,remain} = T_{snso}(-1) \quad \text{(Eq. 3.7.130)}$$

$$D_{zsnso}(-1) = 0.2 \quad \text{(Eq. 3.7.131)}$$

$$W_{ice,sno}(-1) = W_{ice,sno}(-1) \times \frac{0.2}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.132)}$$

$$W_{liq,sno}(-1) = W_{liq,sno}(-1) \times \frac{0.2}{D_{zsnso}(-1)} \quad \text{(Eq. 3.7.133)}$$

Then, the remaining part (i.e., $D_{z,sno,remain}$, $W_{ice,sno,remain}$, $W_{liq,sno,remain}$, $T_{snso,remain}$) after the middle layer division will be further combined with the layer below (i.e., the bottom layer) based on the combination process in SnowLayerWaterComboMod.F90.

Note that there is no maximum thickness threshold for the bottom snow layer in Noah-MP.

3.7.3.5 Snowpack Hydrological Process

Description:

This module is to update snow ice and liquid water content and snow height based on snowpack water balance (sublimation, frost, melting) and to compute water fluxes within and out of snowpack.

Relevant code modules:

Module: SnowpackHydrologyMod.F90

Subroutine: SnowpackHydrology

Module: SnowLayerCombineMod.F90

Subroutine: SnowLayerCombine

Physics:

1. If no snow exists anymore (W_{snow} = 0) after snow layer combination process (e.g., due to earlier over-sublimation):

The snow surface frost flux ($Q_{sno,frost}$, [mm/s]) and sublimation flux ($Q_{sno,subl}$, [mm/s]) computed from the subsection "Ground surface water flux" above are then applied to soil ice:

$$W_{ice,soil}(1) = W_{ice,soil}(1) + \frac{(Q_{sno,frost} - Q_{sno,subl}) \times \Delta t}{D_{zsnso}(1) \times 1000} \quad \text{(Eq. 3.7.134)}$$

If $W_{ice,soil}(1) < 0$ (soil ice is also not sufficient for sublimation), then the remaining water is taken out of top layer soil liquid water to conserve water:

$$W_{liq,soil}(1) = W_{liq,soil}(1) + W_{ice,soil}(1) \quad \text{(Eq. 3.7.135)}$$

$$W_{ice,soil}(1) = 0 \quad \text{(Eq. 3.7.136)}$$

2. For shallow snow without an explicit snow layer ($I_{n,sno} = 0$ and $W_{snow} > 0$):

$$W_{snow,new} = W_{snow,old} + (Q_{sno,frost} - Q_{sno,subl}) \times \Delta t \quad \text{(Eq. 3.7.137)}$$

$$h_{snow} = \max(0.0, h_{snow} \times \frac{W_{snow,new}}{W_{snow,old}}) \quad \text{(Eq. 3.7.138)}$$

The snow depth is further constrained to a reasonable snowpack density (50~500 kg/m^3):

$$h_{snow} = \min(\max\left(h_{snow}, \frac{W_{snow,new}}{500}\right), \frac{W_{snow,new}}{50}) \quad \text{(Eq. 3.7.139)}$$

The snow surface sublimation may be larger than existing snow mass (i.e., $W_{snow,new} < 0$). In this case, to conserve water, excessive sublimation is used to reduce soil water. Smaller time steps would tend to avoid this problem.

$$W_{ice,soil}(1) = W_{ice,soil}(1) + \frac{W_{snow,new}}{D_{zsnso}(1)\times 1000} \qquad \text{(Eq. 3.7.140)}$$

$$W_{snow,new} = 0 \qquad \text{(Eq. 3.7.141)}$$

$$h_{snow} = 0 \qquad \text{(Eq. 3.7.142)}$$

If $W_{ice,soil}(1) < 0$ (soil ice is also not sufficient for sublimation), then the remaining water is taken out of top layer soil liquid water to conserve water:

$$W_{liq,soil}(1) = W_{liqd,soil}(1) + W_{ice,soil}(1) \qquad \text{(Eq. 3.7.143)}$$

$$W_{ice,soil}(1) = 0 \qquad \text{(Eq. 3.7.144)}$$

3. For deep snow with explicit snow layers ($I_{n,sno} < 0$):

(1) The surface frost and sublimation are applied to the top snow layer:

$$W_{ice,sno}(I_{n,sno}+1) = W_{ice,sno}(I_{n,sno}+1) + (Q_{sno,frost} - Q_{sno,subl}) \times \Delta t \qquad \text{(Eq. 3.7.145)}$$

If $W_{ice,sno}\left(I_{n,sno}+1\right) < 10^{-6}$ (i.e., top snow layer becomes very shallow or even over-sublimated), then the snow layer combination process (SnowLayerCombineMod.F90) is called again to merge the top snow layer with the layer below.

Then, if $I_{n,sno} < 0$, the rain on snowpack is added to top layer snow liquid water content:

$$W_{liq,sno}\left(I_{n,sno}+1\right) = \max\left(0,\ W_{liq,sno}\left(I_{n,sno}+1\right) + Q_{rain} \times \Delta t\right) \qquad \text{(Eq. 3.7.146)}$$

where Q_{rain} [mm/s] is the below-canopy rain rate computed in CanopyWaterInterceptMod.F90.

(2) Compute snow porosity (used in interlayer meltwater movement process):

$$V_{ice}(i) = \min\left(1.0, \frac{W_{ice,sno}(i)}{D_{zsnso}(i)\times\rho_{ice}}\right) \qquad \text{(Eq. 3.7.147)}$$

$$P_{eff,sno}(i) = 1 - V_{ice}(i) \qquad \text{(Eq. 3.7.148)}$$

where *i* is the snow layer index, V_{ice} is the ice partial volume, and $P_{eff,sno}$ is the snow effective porosity.

(3) Compute interlayer snow water movement:

The liquid water in each snow layer is computed based on the input water (Q_{in}, [mm]) and snowpack liquid water holding capacity ($W_{liq,cap,sno}$, a parameter in NoahmpTable.TBL):

$$W_{liq,sno}(i) = W_{liq,sno}(i) + Q_{in}(i) \qquad \text{(Eq. 3.7.149)}$$

$$V_{liq}(i) = \frac{W_{liq,sno}(i)}{D_{zsnso}(i) \times \rho_{wat}} \quad \text{(Eq. 3.7.150)}$$

$$Q_{out}(i) = \max(0.0, (V_{liq}(i) - W_{liq,cap,sno} \times P_{eff,sno}(i)) \times D_{zsnso}(i)) \quad \text{(Eq. 3.7.151)}$$

where *i* is the snow layer index, V_{liq} is the liquid water partial volume, and Q_{out} [mm] is the water flux out of the snow layer.

For bottom snow layer (*i*=0):

$$Q_{out}(0) = \left[\max\left(\left(V_{liq}(0) - P_{eff,sno}(0)\right) \times D_{zsnso}(0), \quad F_{ret,sno} \times \Delta t \times Q_{out}(0)\right)\right] \times \rho_{wat}$$

(Eq. 3.7.152)

where $F_{ret,sno}$ [1/s] is the snowpack water release timescale factor set in NoahmpTable.TBL.

$$W_{lid,sno}(i) = W_{liq,sno}(i) - Q_{out}(i) \quad \text{(Eq. 3.7.153)}$$

For snow liquid water fraction ($\frac{W_{liq,sno}(i)}{LW_{liq,sno}(i)+W_{ice,sno}(i)}$) exceeding the maximum allowed snowpack liquid mass fraction ($f_{liq,sno,max}$, parameter in NoahmpTable.TBL), the water excess will be added to Q_{out}:

$$Q_{out}(i) = Q_{out}(i) + (W_{liq,sno}(i) - \frac{f_{liq,sno,max}}{(1-f_{liq,sno,max})} \times W_{ice,sno}(i)) \quad \text{(Eq. 3.7.154)}$$

$$W_{liq,sno}(i) = \frac{f_{liq,sno,max}}{(1-f_{liq,sno,max})} \times W_{ice,sno}(i) \quad \text{(Eq. 3.7.155)}$$

$$Q_{in}(i+1) = Q_{out}(i) \quad \text{(Eq. 3.7.156)}$$

This total water flux (Q_{snobot}, [mm/s]) out of snowpack bottom is:

$$Q_{snobot} = \frac{Q_{out}(0)}{\Delta t} \quad \text{(Eq. 3.7.157)}$$

which includes both snowmelt and rain through snowpack.

(4) Constrain snow layer thickness with snow density:

$$D_{zsnso}(i) = \max(D_{zsnso}(i), \frac{W_{ice,sno}(i)}{\rho_{ice}} + \frac{W_{liq,sno}(i)}{\rho_{wat}}) \quad \text{(Eq. 3.7.158)}$$

3.7.3.6 Update Snowpack Diagnostic Properties

Description:

This part is to update some key snowpack quantities after all the snowpack processes, including snow thickness and SWE.

Relevant code modules:

Module: SnowWaterMainMod.F90

Subroutine: SnowWaterMain

Physics:

1. For non-melt-out snow region ("glacier" region):

Some regions have non-seasonal snow (i.e., snow never melts out entirely in summer), thus for those "glacier" regions, Noah-MP sets a maximum SWE (W_{snow}, currently 5000 mm) and adds any excess water to glacier water flow ($Q_{sno,gla}$, [mm/s]). So if W_{snow} > 5000 mm, then:

$$\rho_{sno,bulk} = \frac{W_{ice,sno}(0)}{D_{zsnso}(0)} \quad \text{(Eq. 3.7.159)}$$

$$Q_{sno,gla} = W_{snow} - 5000 \quad \text{(Eq. 3.7.160)}$$

$$W_{ice,sno}(0) = W_{ice,sno}(0) - Q_{sno,gla} \quad \text{(Eq. 3.7.161)}$$

$$D_{zsnso}(0) = D_{zsnso}(0) - \frac{Q_{sno,gla}}{\rho_{sno,bulk}} \quad \text{(Eq. 3.7.162)}$$

$$Q_{sno,gla} = \frac{Q_{sno,gla}}{\Delta t} \quad \text{(Eq. 3.7.163)}$$

where $\rho_{sno,bulk}$ [kg/m^3] is the bulk snow density for the bottom snow layer. The $Q_{sno,gla}$ will be further added to the subsurface runoff for these regions later.

2. Update SWE and snow depth:

The total W_{snow} [mm] is a sum of $W_{ice,sno}$ and $W_{liq,sno}$ [mm] across all snow layers for snowpack with explicit layers ($I_{n,sno} < 0$).

The $Z_{snso}(i)$ [m] (depth of snow and soil layer bottom) is also updated based on $D_{zsnso}(i)$:

$$Z_{snso}(i) = Z_{snso}(i-1) + D_{zsnso}(i) \quad \text{(Eq. 3.7.164)}$$

The snow depth for snow with explicit layers ($ind_{sno} < 0$) is updated based on $DZ_{snso}(i)$:

$$h_{sno} = \sum_{i=I_{n,sno}+1}^{0} D_{zsnso}(i) \quad \text{(Eq. 3.7.165)}$$

3.7.4 Frozen Soil Treatment and Total Water Flux into Soil

Description:

This part is to adjust soil ice and liquid water content for frozen soil and compute the total water influx to soil surface after snowpack processes above.

Relevant code modules:

Module: WaterMainMod.F90

Subroutines： WaterMain

Physics:

1. Frozen soil treatment:

If $T_g \leq T_{frz}$ (i.e., frozen ground), then update soil surface ice content based on sublimation and frost rate:

$$W_{ice,soil}(1) = W_{ice,soil}(1) + \frac{(Q_{soil,dew} - Q_{soil,evap}) \times \Delta t}{D_{zsnso}(1) \times 1000} \quad \text{(Eq. 3.7.166)}$$

$$Q_{soil,dew} = 0 \quad \text{(Eq. 3.7.167)}$$

$$Q_{soil,evap} = 0 \quad \text{(Eq. 3.7.168)}$$

where $Q_{soil,dew}$ and $Q_{soil,evap}$ [mm/s] are soil surface dew and evaporation rates computed in Section 3.4.2. Due to frozen soil, evaporation is in the form of sublimation instead, and dew is in the form of frost.

If soil ice is over-sublimated ($W_{ice,soil}(1) < 0$), then the over-sublimated water is taken out of soil surface liquid water:

$$W_{liq,soil}(1) = W_{liq,soil}(1) + W_{ice,soil}(1) \quad \text{(Eq. 3.7.169)}$$

$$W_{ice,soil}(1) = 0 \quad \text{(Eq. 3.7.170)}$$

2. Compute total water flux into soil column from top:

The total water influx ($Q_{in,soil}$, [m/s]) into soil column from top includes snowmelt, surface ponding, rain through snowpack, and surface dew:

$$Q_{in,soil} = \frac{W_{ponding} + W_{ponding,1} + W_{ponding,2}}{\Delta t} \times 0.001 \quad \text{(Eq. 3.7.171)}$$

where $W_{ponding}$, $W_{ponding,1}$, and $W_{ponding,2}$ [mm] are surface ponding computed from different snow water processes above.

For non-snow case or snow without explicit layers ($I_{n,sno} = 0$):

$$Q_{in,soil} = Q_{in,soil} + (Q_{snobot} + Q_{soil,dew} + Q_{rain}) \times 0.001 \qquad \text{(Eq. 3.7.172)}$$

where Q_{snobot} [mm/s] is the water outflow from snowpack bottom, and Q_{rain} [mm/s] is the rain rate below canopy.

For snowpack with explicit layers ($I_{n,sno} < 0$):

$$Q_{in,soil} = Q_{in,soil} + (Q_{snobot} + Q_{soil,dew}) \times 0.001 \qquad \text{(Eq. 3.7.173)}$$

Note that Q_{rain} is not included for snowpack with explicit layers here, because it is already included in Q_{snobot} when computing snow hydrology process (SnowpackHydrologyMod.F90).

The soil surface evaporation water will be accounted for later when solving soil water process (SoilWaterMainMod.F90), but it is converted from mm/s to m/s here since the water flux input for soil water processes is m/s:

$$Q_{soil,evap} = Q_{soil,evap} \times 0.001 \qquad \text{(Eq. 3.7.174)}$$

If running Noah-MP with WRF-Hydro, then the surface water head ($H_{wat,sfc}$, [mm]) from the WRF-Hydro model is also accounted for:

$$Q_{in,soil} = Q_{in,soil} + \frac{H_{wat,sfc}}{\Delta t} \times 0.001 \qquad \text{(Eq. 3.7.175)}$$

If flood irrigation and/or micro-irrigation is activated, then the $Q_{in,soil}$ and/or surface soil water content ($W_{liq,soil}(1)$) will be further updated to include irrigation water influx. Please see below for irrigation processes.

3.7.5 Soil Water Loss due to Plant Transpiration

Description:

This part is to compute the soil water loss rate due to plant transpiration computed from the canopy hydrology process.

Relevant code modules:

Module: WaterMainMod.F90

Subroutines： WaterMain

Physics:

The soil water loss ($Q_{soil,transp}$, [m/s]) due to plant transpiration is computed as:

$$Q_{soil,transp}(i) = Q_{can,transp} \times F_{soil,transp}(i) \times 0.001 \quad \text{(Eq. 3.7.176)}$$

where i is the soil layer index from 1 (top) to the maximum root layer, and $Q_{can,transp}$ [mm/s] is the total plant transpiration computed in CanopyHydrologyMod.F90. $F_{soil,transp}$ (=0~1) is the soil transpiration factor computed in SoilWaterTranspirationMod.F90.

3.7.6 Dynamic Irrigation Schemes

Description:

The new dynamic irrigation scheme is developed to represent the main irrigation methods, and irrigation water management in different countries, including the United States (Valayamkunnath, 2019; Valayamkunnath et al., 2020; Valayamkunnath et al., 2021). The four main irrigation questions addressed with our new dynamic irrigation scheme are:

1) Where to irrigate?

2) When to irrigate?

3) What amount of water to irrigate? and

4) How to irrigate?

The dynamic irrigation scheme uses a 250-m resolution Moderate Resolution Imaging Spectroradiometer (MODIS) Irrigated Agriculture Dataset for the United States (MIrAD-US) (Pervez and Brown, 2010) to prescribe the irrigated croplands within the Noah-MP dynamic irrigation scheme. The MIrAD-US is available for 2002, 2007, 2012, and 2017 at 250-m and 1-km spatial resolutions. The dynamic irrigation scheme uses 30-arc seconds global irrigated area data from Meier et al. (2018) for the global Noah-MP experiments. This data covers the period from 1999 to 2012. The Noah-MP dynamic irrigation scheme will irrigate all the irrigated croplands if the irrigated area fraction in a model grid is greater than or equal to user-defined threshold.

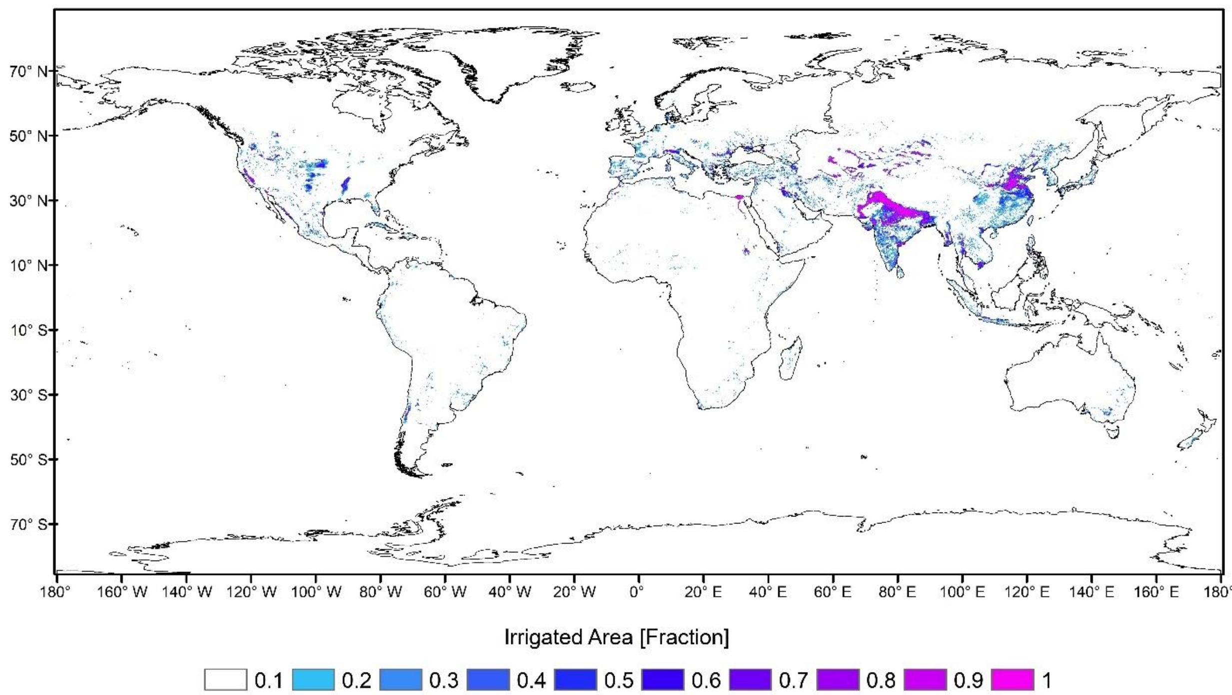


Figure 12. Global irrigated croplands. Both MIrAD-US and Meier et al. (2018) are re-gridded onto a 0.0833-degree resolution grid for a better data visualization.

Relevant code modules:

Module: IrrigationPrepareMod.F90

Subroutines: IrrigationPrepare

Module: IrrigationTriggerMod.F90

Subroutines: IrrigationTrigger

Submodule: IrrigationSprinklerMod.F90

Subroutines: IrrigationSprinkler

Submodule: IrrigationFloodMod.F90

Subroutines: IrrigationFlood

Submodule: IrrigationMicroMod.F90

Subroutines: IrrigationMicro

Submodule: IrrigationInfilPhilipMod.F90

Subroutines: IrrigationInfilPhilip

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptIrrigation options for irrigation	0*	No irrigation
	1	Irrigation ON
	2	irrigation trigger based on crop season Planting and harvesting dates
	3	irrigation trigger based on LeafAreaIndex threshold
OptIrrigationMethod options for irrigation method, only works when OptIrrigation > 0	0*	method based on geo_em fractions
	1	sprinkler method
	2	micro/drip irrigation
	3	surface flooding

Physics:

In the dynamic irrigation scheme, the irrigation water is estimated based on the soil moisture conditions. The scheme will trigger when the rootzone soil moisture falls below some threshold soil moisture level. Once the irrigation is triggered, it estimates the amount of water to be irrigated to bring back the soil moisture to field capacity. Irrigation water estimation by the dynamic irrigation scheme is summarized as follows.

$$I_W = \begin{cases} f_{irr} \times f_{veg} \times (\theta_{limit} - \theta_{available}); & if \ \left(\frac{\theta_{avail}}{\theta_{limit}}\right) \leq M_{AD} \\ 0 \quad ; & otherwise \end{cases} \quad \text{(Eq. 3.7.177)}$$

$$\theta_{limit} = \sum_{i=1}^{N_{Root}} \left[\theta_{soil,ref} - \theta_{soil,wp}\right] \times \Delta Z \quad \text{(Eq. 3.7.178)}$$

$$\theta_{avail} = \sum_{i=1}^{N_{Root}} \left[\theta_{soil} - \theta_{soil,wp}\right] \times \Delta Z \quad \text{(Eq. 3.7.179)}$$

where I_W [mm] is the total irrigation water, f_{irr} and f_{veg} are the irrigated area fraction and vegetation fraction in a Noah-MP grid, respectively, θ_{avail}, $\theta_{soil,wp}$, $\theta_{soil,ref}$ [m^3/m^3] are the volumetric soil moisture content at porosity, wilting point, and field capacity, respectively, for each soil layer, ΔZ [m] is the thickness of each soil layer, N_{Root} is the lowest soil layer with crop root, and M_{AD} is the management allowed deficit set in NoahmpTable.TBL

Once the total irrigation water is estimated, the dynamic irrigation scheme supplies the water using three different irrigation methods: sprinkler, drip, and flood irrigation. The irrigation scheme uses these methods fractions to estimate the irrigation water applied by each method:

$$I_{W,sprinkler} = f_{sprinkler} \times I_W \quad \text{(Eq. 3.7.180)}$$

$$I_{W,micro} = f_{micro} \times I_W \quad \text{(Eq. 3.7.181)}$$

$$I_{W,flood} = f_{flood} \times I_W \times \left(\frac{1}{1-\varepsilon}\right) \quad \text{(Eq. 3.7.182)}$$

where $I_{W,sprinkler}$ [mm] is sprinkler irrigation water, $I_{W,micro}$ [mm] is micro irrigation water, and $I_{W,flood}$ [mm] is flood irrigation water. To account for the runoff loss of flood irrigation water, we incorporated a water loss term, ε. After the irrigation water is estimated for individual methods, the dynamic irrigation scheme will apply water on the top of the first soil layer based on user-specified application rates for flooding and micro irrigations. The sprinkler irrigation water is treated as "precipitation" process. However, we constraint the application rates based on the Natural Resources Conservation Service (NRCS) - United States Department of Agriculture (USDA) irrigation design criteria (Mockus, 1964). The NRCS irrigation scheduling criteria used in the dynamic irrigation scheme are provided below.

$$Q_{sprinkler} = \begin{cases} R_{sprinkler}; & if\ \ R_{sprinkler} \leq I_f \\ I_f \quad ; & if\ I_f \leq R_{sprinkler} \end{cases} \qquad \text{(Eq. 3.7.183)}$$

$$Q_{micro} = \begin{cases} R_{micro}; & if\ \ R_{micro} \leq 0.5 \times I_f \\ 0.5 \times I_f \quad ; & if\ 0.5 \times I_f \leq R_{micro} \end{cases} \qquad \text{(Eq. 3.7.184)}$$

$$Q_{flood} = I_f \times \alpha \qquad \text{(Eq. 3.7.185)}$$

where Q [mm/s] stands for irrigation application rate estimated by different irrigation schemes, I_f [mm/s] is the infiltration rate estimated using Philip infiltration equation (Valiantzas, 2010), R [mm/s] represents the irrigation application rate specified by the user, and α is the flood irrigation application rate factor (specified in NoahmpTable.TBL). The α is greater than or equal to 1.0.

Additionally, in the dynamic irrigation scheme, we calculate the evaporative loss from sprinkler droplets ($E_{sprinkler}$, [mm/s]) using the Bavi et al. (2009) formulation:

$$E_{sprinkler} = \left(0.04375 \times e^{0.106 \times u} \times (e_s - e_a)^{-0.092} \times {T_{sfc}}^{-0.102}\right) \times Q_{sprinkler} \qquad \text{(Eq. 3.7.186)}$$

where u [m/s] is the wind speed, e_s [Pa] is the saturation vapor pressure, e_a [Pa] is the surface vapor pressure, and T_{sfc} [K] is the surface temperature.

3.7.7 Lake Surface Treatment

Description:

This part is to treat water and runoff fluxes for lake pixels.

Relevant code modules:

Module: WaterMainMod.F90

Subroutines： WaterMain

Physics:

For lake land type pixel, the surface runoff ($R_{sfc,lake}$, [mm/s]) is computed based on a maximum lake water storage ($W_{lake,max}$, [mm]) as set in NoahmpTable.TBL (currently 5000 mm):

$$R_{sfc,lake} = \begin{cases} 0.0, & W_{lake} < W_{lake,max} \\ Q_{in,soil} \times 1000, & W_{lake} \geq W_{lake,max} \end{cases} \quad \text{(Eq. 3.7.187)}$$

$$W_{lake} = W_{lake} + \left(Q_{in,soil} - Q_{soil,evap}\right) \times 1000 \times \Delta t - R_{sfc,lake} \times \Delta t \quad \text{(Eq. 3.7.188)}$$

where W_{lake} [mm] is the lake water storage, $Q_{in,soil}$ [m/s] is the total water influx to the soil/lake surface, and $Q_{soil,evap}$ [m/s] is the soil/lake surface evaporation rate.

3.7.8 Soil Hydrology

Description:

This module is to update soil moisture and compute all soil water processes, including surface runoff, infiltration, soil water diffusion, subsurface runoff, tile drainage, and groundwater. Note that Noah-MP allows the use of a different (longer) soil process timestep than the main model timestep. By default, the soil timestep is the same as the main model timestep.

Relevant code modules:

Module: SoilWaterMainMod.F90

Subroutine： SoilWaterMain

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptRunoffSurface options for surface runoff	1	TOPMODEL with groundwater
	2	TOPMODEL with an equilibrium water table
	3*	Schaake's surface and subsurface runoff (free drainage)
	4	BATS surface and subsurface runoff (free drainage)
	5	Miguez-Macho&Fan groundwater scheme
	6	Variable Infiltration Capacity (VIC) surface runoff scheme
	7	Xinanjiang Infiltration and surface runoff scheme

	8	Dynamic VIC surface runoff scheme
OptRunoffSubsurface options for subsurface runoff currently tested & recommended the same option# as surface runoff	1	TOPMODEL with groundwater
	2	TOPMODEL with an equilibrium water table
	3*	Schaake's surface and subsurface runoff (free drainage)
	4	BATS surface and subsurface runoff (free drainage)
	5	Miguez-Macho&Fan groundwater subsurface runoff
	6	Variable Infiltration Capacity (VIC) subsurface runoff
	7	XinAnJiang Infiltration and subsurface runoff
	8	Dynamic VIC subsurface runoff
OptTileDrainage options for tile drainage currently only tested & calibrated to work with surface runoff opt=3	0*	No tile drainage
	1	simple scheme
	2	Hooghoudt's scheme

Physics:

Soil water saturation excess and frozen soil treatment:

For each soil layer, the effective porosity ($\theta_{eff,pore}$, [m^3/m^3]) is:

$$\theta_{eff,pore} = \max(10^{-4},\ \theta_{soil,max} - W_{ice,soil}) \quad \text{(Eq. 3.7.189)}$$

where $\theta_{soil,max}$ [m^3/m^3] is the maximum (saturated) soil moisture set in the NoahmpTable.TBL and $W_{ice,soil}$ [m^3/m^3] is the soil ice content.

The accumulation of the soil water saturation excess ($\theta_{soil,excess}$, [m]) due to too large water out of snowpack bottom is:

$$\theta_{soil,excess} = \sum_{i=1}^{N_{soil}} (\max(0, W_{liq,soil}(i) - \theta_{eff,pore}(i)) \times D_{zsnso}(i)) \quad \text{(Eq. 3.7.190)}$$

where $W_{liq,soil}$ [m^3/m^3] is the soil liquid water content, N_{soil} is the number of soil layers, and D_{zsnso} [m] is the soil layer thickness.

The soil liquid water content for each soil layer is:

$$W_{liq,soil} = \min(\theta_{eff,pore},\ W_{liq,soil}) \quad \text{(Eq. 3.7.191)}$$

Then, the impermeable fraction ($f_{imp,soil}$) due to frozen soil for each soil layer is:

$$f_{imp,soil} = \frac{\max(0.0,\ e^{\left(-A\times\left(1-f_{ice,soil}\right)\right)} - e^{(-A)})}{(1.0 - e^{(-A)})} \quad \text{(Eq. 3.7.192)}$$

$$f_{ice,soil} = \min(1.0, \frac{W_{ice,soil}}{\theta_{soil,max}}) \quad \text{(Eq. 3.7.193)}$$

where A (= 4.0) is a hard-coded soil impermeability parameter, and $f_{ice,soil}$ is the soil ice fraction.

The maximum soil ice content ($W_{ice,soil,max}$, [m^3/m^3]), maximum impermeable fraction ($f_{imp,soil,max}$), and minimum soil liquid water content ($W_{liq,soil,min}$, [m^3/m^3]) are computed iteratively looping over all soil layers as:

$$W_{ice,soil,max} = W_{ice,soil}, \ if \ W_{ice,soil} > W_{ice,soil,max} \quad \text{(Eq. 3.7.194)}$$

$$W_{liq,soil,min} = W_{liq,soil}, \ if \ W_{liq,soil} < W_{liq,soil,min} \quad \text{(Eq. 3.7.195)}$$

$$f_{imp,soil,max} = f_{imp,soil}, \ if \ f_{imp,soil} > f_{imp,soil,max} \quad \text{(Eq. 3.7.196)}$$

If over urban pixels, then the first soil layer impermeable fraction is set to:

$$f_{imp,soil}(1) = 0.95 \quad \text{(Eq. 3.7.197)}$$

All the above quantities will be used in the following surface and subsurface runoff calculations.

3.7.8.1 Surface Runoff and Infiltration

Description:

This part is to compute surface runoff and surface infiltration rates based on different physics options.

Relevant code modules:

Module: RunoffSurfaceTopModelGrdMod.F90

Subroutine: RunoffSurfaceTopModelGrd

Module: RunoffSurfaceTopModelEquiMod.F90

Subroutine: RunoffSurfaceTopModelEqui

Module: RunoffSurfaceFreeDrainMod.F90

Subroutine: RunoffSurfaceFreeDrain

Module: RunoffSurfaceBatsMod.F90

Subroutine: RunoffSurfaceBATS

Module: RunoffSurfaceTopModelMmfMod.F90

Subroutine: RunoffSurfaceTopModelMMF

Module: RunoffSurfaceVicMod.F90

Subroutine: RunoffSurfaceVIC

Module: RunoffSurfaceXinAnJiangMod.F90

Subroutine: RunoffSurfaceXinAnJiang

Module: RunoffSurfaceDynamicVicMod.F90

Subroutine: RunoffSurfaceDynamicVic

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptRunoffSurface options for surface runoff	1	TOPMODEL with groundwater
	2	TOPMODEL with an equilibrium water table
	3*	Schaake's surface and subsurface runoff (free drainage)
	4	BATS surface and subsurface runoff (free drainage)
	5	Miguez-Macho&Fan groundwater scheme
	6	Variable Infiltration Capacity (VIC) surface runoff scheme
	7	XinAnJiang Infiltration and surface runoff scheme
	8	Dynamic VIC surface runoff scheme

Physics:

1. OptRunoffSurface = 1 (RunoffSurfaceTopModelGrdMod.F90**)**:

The TOPMODEL with groundwater scheme (Niu et al., 2007) is used to compute to compute surface runoff and surface infiltration rates. The surface runoff (R_{sfc}, [m/s]) and infiltration rate ($Q_{infil,sfc}$, [m/s]) are computed when the total water input flux ($Q_{in,soil}$, [m/s]) into the top soil surface is greater than zero as follows:

$$R_{sfc} = Q_{in,soil} \times [(1 - f_{imp,soil}(1)) \times f_{sat,soil} + f_{imp,soil}(1)] \quad \text{(Eq. 3.7.198)}$$

$$Q_{infil,sfc} = Q_{in,soil} - R_{sfc} \quad \text{(Eq. 3.7.199)}$$

where $Q_{in,soil}$ is computed in the above processes (rain, snowpack, and irrigation), $f_{imp,soil}(i)$ is the i^{th} soil layer impermeable fraction, and $f_{sat,soil}$ is the soil saturated fraction of the area calculated as:

$$f_{sat,soil} = f_{sat,soil,max} \times e^{-0.5 \times F_{rf,decay} \times d_{wt}} \quad \text{(Eq. 3.7.200)}$$

$$F_{rf,decay} = \frac{B_{exp}(1)}{3} \quad \text{(Eq. 3.7.201)}$$

where $F_{rf,decay}$ [1/m] is the runoff decay factor, B_{exp} is the soil B parameter (set in NoahmpTable.TBL), d_{wt} [m] is the water table depth (positive), and $f_{sat,soil,max}$ is the maximum saturated fraction of soil surface (=0.38 set in NoahmpTable.TBL). Particularly, the d_{wt} for OptRunoffSurface = 1 is computed using the Niu et al. (2007) groundwater scheme (GroundWaterTopModelMod.F90). Please see the subsurface runoff part below for details.

2. OptRunoffSurface = 2 (RunoffSurfaceTopModelEquiMod.F90**)**:

The TOPMODEL with equilibrium water table scheme (Niu et al., 2005) is used to compute to compute surface runoff and surface infiltration rates, which is similar to OptRunoffSurface = 1 due to the use of TOPMODEL surface runoff formulation. The surface runoff (R_{sfc}, [m/s]) and infiltration rate ($Q_{infil,sfc}$, [m/s]) are computed when the total water input flux ($Q_{in,soil}$, [m/s]) into the top soil surface is greater than zero as follows:

$$R_{sfc} = Q_{in,soil} \times [(1 - f_{imp,soil}(1)) \times f_{sat,soil} + f_{imp,soil}(1)] \quad \text{(Eq. 3.7.202)}$$

$$Q_{infil,sfc} = Q_{in,soil} - R_{sfc} \quad \text{(Eq. 3.7.203)}$$

where $Q_{in,soil}$ is computed in the above processes (rain, snowpack, and irrigation), $f_{imp,soil}(i)$ is the i^{th} soil layer impermeable fraction, and $f_{sat,soil}$ is the soil saturated fraction of the area calculated as:

$$f_{sat,soil} = f_{sat,soil,max} \times e^{-0.5 \times F_{rf,decay} \times d_{wt}} \quad \text{(Eq. 3.7.204)}$$

where $F_{rf,decay}$ [1/m] is the runoff decay factor (=2.0 for OptRunoffSurface = 2), d_{wt} [m] is the water table depth (positive), and $f_{sat,soil,max}$ is the maximum saturated fraction of soil surface (=0.38 set in NoahmpTable.TBL). Particularly, the d_{wt} for OptRunoffSurface = 2 is computed using equilibrium water table scheme (WaterTableEquilibriumMod.F90). Please see the subsurface runoff part below for details.

3. OptRunoffSurface = 3 (RunoffSurfaceFreeDrainMod.F90**)**:

This is the Schaake's free drainage scheme used to compute surface runoff and surface infiltration rates.

Relevant code modules:

Module: SoilHydraulicPropertyMod.F90

Subroutine: SoilDiffusivityConductivityOpt2

If the water input on soil surface is greater than zero ($Q_{in,soil} > 0$), the infiltration calculation is activated. First, the soil model timestep (Δt_{soil}, [s]) is converted to the ratio of a day (Δt_{frac}):

$$\Delta t_{frac} = \frac{\Delta t_{soil}}{86400} \quad \text{(Eq. 3.7.205)}$$

The difference ($\theta_{soil,m-wp}$, [m^3/m^3]) between saturated soil moisture ($\theta_{soil,max}$, [m^3/m^3]) and permanent wilting point soil moisture ($\theta_{soil,wilt}$, [m^3/m^3]) is:

$$\theta_{soil,m-wp} = \theta_{soil,max} - \theta_{soil,wilt} \quad \text{(Eq. 3.7.206)}$$

where $\theta_{soil,max}$ and $\theta_{soil,wilt}$ are parameters set in NoahmpTable.TBL.

For the first (top) soil layer,

$$W_{ice,soil,d} = -Z_{soil}(1) \times W_{ice,soil}(1) \quad \text{(Eq. 3.7.207)}$$

$$W_{soil,d,max}(1) = (-Z_{soil}(1) \times \theta_{soil,m-wp}) \times \left[1 - \frac{(W_{liq,soil}(1) + W_{ice,soil}(1) - \theta_{soil,wilt}(1))}{\theta_{soil,m-wp}}\right]$$

(Eq. 3.7.208)

where $W_{ice,soil,d}$ [m] is the accumulated soil ice content in the unit of depth, $W_{soil,d,max}$ [m] is the maximum holdable soil water content in the unit of depth, $W_{ice,soil}$ [m^3/m^3] and $W_{liq,soil}$ [m^3/m^3] are the soil ice and liquid water content, respectively.

For the 2nd through last (bottom) soil layers:

$$W_{ice,soil,d} = W_{ice,soil,d} + \sum_{i=2}^{N_{soil}}[(Z_{soil}(i-1) - Z_{soil}(i)) \times W_{ice,soil}(i)] \quad \text{(Eq. 3.7.209)}$$

$$W_{soil,d,max}(i) = \left[\left(Z_{soil}(i-1) - Z_{soil}(i)\right) \times \theta_{soil,m-wp}\right] \times \left[1 - \frac{\left(W_{liq,soil}(i) + W_{liq,soil}(i) - \theta_{soil,wilt}(i)\right)}{\theta_{soil,m-wp}}\right]$$

(Eq. 3.7.210)

$$W_{soil,max,tot} = \sum_{i=1}^{N_{soil}} W_{soil,d,max}(i) \quad \text{(Eq. 3.7.211)}$$

where Z_{soil} [m] is the depth (negative) of each soil layer bottom.

Then, the maximum soil infiltration rate ($Q_{infil,max}$, [m/s]) is:

$$Q_{infil,max} = \frac{P_x \times \left[\frac{W_{soil,max,tot} \times (1 - e^{-K_{dt} \times \Delta t_{frac}})}{P_x + W_{soil,max,tot} \times (1 - e^{-K_{dt} \times \Delta t_{frac}})}\right]}{\Delta t} \quad \text{(Eq. 3.7.212)}$$

$$P_x = \max(0, Q_{in,soil} \times \Delta t) \quad \text{(Eq. 3.7.213)}$$

where K_{dt} is the coefficient for computing maximum soil infiltration rate and P_x [m] is the total soil surface input water.

The impermeable fraction ($f_{imp,soil}$) due to frozen soil is expressed as:

$$f_{imp,soil} = \begin{cases} 1.0, & if\ W_{ice,soil,d} \le 0.01 \\ 1 - T \times e^{\left[\frac{-C_{frz} \times F_{frz}}{W_{ice,soil,d}}\right]}, & if\ W_{ice,soil,d} > 0.01 \end{cases} \quad \text{(Eq. 3.7.214)}$$

$$F_{frz} = F_{frz,g} \times \frac{\theta_{soil,max}}{\theta_{soil,ref}} \times \frac{0.412}{0.468} \quad \text{(Eq. 3.7.215)}$$

$$T = 1 + \sum_{j=1}^{C_{frz}-1} \left[\frac{\left(\frac{C_{frz} \times F_{frz}}{W_{ice,soil,d}}\right)^{(C_{frz}-j)}}{1 \times \prod_{k=j+1}^{C_{frz}-1} k} \right] \quad \text{(Eq. 3.7.216)}$$

where C_{frz} is the frozen soil factor (=3, hard-coded currently), $F_{frz,g}$ is the ground frozen factor (set in NoahmpTable.TBL), and $\theta_{soil,ref}$ [m^3/m^3] is the soil moisture field capacity (set in NoahmpTable.TBL).

The maximum soil infiltration rate is further corrected for frozen soil as follows:

$$Q_{infil,max} = Q_{infil,max} \times f_{imp,soil} \quad \text{(Eq. 3.7.217)}$$

$$Q_{infil,max} = \max(Q_{infil,max}, K_{soil,cond}) \quad \text{(Eq. 3.7.218)}$$

$$Q_{infil,max} = \min(Q_{infil,max}, P_x/\Delta t) \quad \text{(Eq. 3.7.219)}$$

where $K_{soil,cond}$ [m/s] is the soil hydraulic conductivity computed in SoilDiffusivityConductivityOpt2 subroutine in SoilHydraulicPropertyMod.F90 module (see subsection "Soil hydraulic properties" for details).

Finally, the surface runoff (R_{sfc}, [m/s]) and surface infiltration rate ($Q_{infil,sfc}$, [m/s]) are:

$$R_{sfc} = \max(0, Q_{in,soil} - Q_{infil,max}) \quad \text{(Eq. 3.7.220)}$$

$$Q_{infil,sfc} = Q_{in,soil} - R_{sfc} \quad \text{(Eq. 3.7.221)}$$

4. OptRunoffSurface = 4 (RunoffSurfaceBatsMod.F90**)**:

This is the BATS free drainage scheme used to compute surface runoff and surface infiltration rates. Most formulations in this scheme are similar to those in OptRunoffSurface=1, except a different parameterization for the soil saturated fraction of the area ($f_{sat,soil}$).

If the water input on the soil surface is greater than zero ($Q_{in,soil} > 0$), the surface runoff (R_{sfc}, [m/s]) and infiltration rate ($Q_{infil,sfc}$, [m/s]) are computed as follows:

$$R_{sfc} = Q_{in,soil} \times [(1 - f_{imp,soil}(1)) \times f_{sat,soil} + f_{imp,soil}(1)] \quad \text{(Eq. 3.7.222)}$$

$$Q_{infil,sfc} = Q_{in,soil} - R_{sfc} \quad \text{(Eq. 3.7.223)}$$

$$f_{sat,soil} = \left[\max(0.01, \frac{\sum_{i=1}^{N_{soil}} (\frac{\theta_{soil}(i)}{\theta_{soil,max}(i)} \times D_{zsnso}(i))}{\sum_{i=1}^{N_{soil}} D_{zsnso}(i)}) \right]^4 \quad \text{(Eq. 3.7.224)}$$

Note that if the total soil column depth is greater than 2 m, this scheme only computes the $f_{sat,soil}$ for the top 2-m soil layers.

5. OptRunoffSurface = 5 (RunoffSurfaceTopModelMmfMod.F90**)**:

The TOPMODEL with MMF groundwater scheme (Fan et al., 2007; Miguez-Macho et al. 2007) is used to compute to compute surface runoff and surface infiltration rates, with similar formulations to OptRunoffSurface = 1 due to the use of the TOPMODEL surface runoff formulation. The surface runoff (R_{sfc}, [m/s]) and infiltration rate ($Q_{infil,sfc}$, [m/s]) are computed when the total water input flux ($Q_{in,soil}$, [m/s]) into the top soil surface is greater than zero as follows:

$$R_{sfc} = Q_{in,soil} \times [(1 - f_{imp,soil}(1)) \times f_{sat,soil} + f_{imp,soil}(1)] \qquad \text{(Eq. 3.7.225)}$$

$$Q_{infil,sfc} = Q_{in,soil} - R_{sfc} \qquad \text{(Eq. 3.7.226)}$$

where $Q_{in,soil}$ is computed in the above processes (rain, snowpack, and irrigation), $f_{imp,soil}(i)$ is the i^{th} soil layer impermeable fraction, and $f_{sat,soil}$ is the soil saturated fraction of the area calculated as:

$$f_{sat,soil} = f_{sat,soil,max} \times e^{-0.5 \times F_{rf,decay} \times \max(0,(-2-d_{wt}))} \qquad \text{(Eq. 3.7.227)}$$

where $F_{rf,decay}$ [1/m] is the runoff decay factor (=6.0 for OptRunoffSurface = 5), d_{wt} [m] is the water table depth (negative), and $f_{sat,soil,max}$ is the maximum saturated fraction of soil surface (=0.38 set in NoahmpTable.TBL). Particularly, the d_{wt} for OptRunoffSurface = 5 is computed using the MMF groundwater scheme (ShallowWaterTableMmfMod.F90 and GroundWaterMmfMod.F90). Please see the subsurface runoff part and groundwater part below for details.

6. OptRunoffSurface = 6 (RunoffSurfaceVicMod.F90**)**:

This is the Variable Infiltration Capacity (VIC) Model surface runoff scheme (Wood et al., 1992, JGR). The implementation into Noah-MP is done by McDaniel et al. (2020). This part computes saturated area, surface infiltration, and surface runoff based on the VIC model runoff scheme (Wood et al., 1992; Liang et al., 1994). This scheme calculates saturation excess surface runoff.

Fraction of saturated soil in a grid (A) is calculated as

$$A = 1 - \left(1 - \left(\frac{W}{W_{max}}\right)^b\right) \qquad \text{(Eq. 3.7.228)}$$

The maximum infiltration capacity (i_{max}) within the gridcell is calculated as:

$$i_{max} = (1 + b) \times W_{max} \qquad \text{(Eq. 3.7.229)}$$

Infiltration capacity i over a gridcell is given by

$$i = i_{max} \times (1 - (1 - A)^{1/b}) \quad \text{(Eq. 3.7.230)}$$

If the effective rainfall on the land surface (P_e) is greater than zero, then the surface runoff (R_s) estimated by VIC runoff scheme is given as below:

If $i + P_e \geq i_{max}$,

$$R_s = P_e - W_{max} + W \quad \text{(Eq. 3.7.231)}$$

else if $i + P_e \leq i_{max}$,

$$R_s = P_e - W_{max} + W + W_{max} \times \left[1 - \frac{i + P_e}{i_{max}}\right]^{(1+b)} \quad \text{(Eq. 3.7.232)}$$

else if $i_{max} = 0$ (impervious):

$$R_s = P_e \quad \text{(Eq. 3.7.233)}$$

where b is a curve shape parameter, W [mm] is the current tension water storage in surface soil layers (layers 1 and 2), and W_{max} [mm] is the maximum tension water storage in surface soil layers (layers 1 and 2).

7. OptRunoffSurface = 7 (RunoffSurfaceXinAnJiangMod.F90**)**:

This is the XinAnJiang infiltration and surface runoff scheme (Jayawardena and Zhou, 2000). The implementation into Noah-MP is done by McDaniel et al. (2020). This uses a variable contributing area to estimate surface runoff (saturation excess runoff). The version of Xinanjiang implemented here uses a double parabolic curve to simulate tension water capacities within the gridcell (Jayawardena and Zhou, 2000).

Fraction of precipitation (P_i, [mm]) that does not fall on impervious area A_{im} is calculated as,

$$P_i = (1 - A_{im}) \times P \quad \text{(Eq. 3.7.234)}$$

where P [mm] is the total precipitation on the grid.

Surface runoff generated from impervious area (R_{im}, [mm]) is given by

$$R_{im} = P \times A_{im} \quad \text{(Eq. 3.7.235)}$$

Surface runoff (R_s, [mm]) from permeable soil is estimated as

$$R_s = R \times \left(1 - \left(1 - \frac{S}{S_{max}}\right)^{E_x}\right) \quad \text{(Eq. 3.7.236)}$$

$$R = \begin{cases} P_i \times \left[(0.5 - a)^{1-b} \times \left(\frac{W}{W_{max}}\right)^b\right] & , \quad if \frac{W}{W_{max}} \leq 0.5 - a \\ P_i \times \left[1 - (0.5 + a)^{1-b} \times \left(1 - \frac{W}{W_{max}}\right)^b\right], & Otherwise \end{cases} \quad \text{(Eq. 3.7.237)}$$

where W [mm] is the current tension water storage in surface soil layers (layers 1 and 2). R [mm] uses a double parabolic curve to determine the fraction of gridcell area that is at full tension storage and thus can contribute to runoff generation. This curve relies on shape parameters a and b, and maximum tension water storage W_{max} [mm] in surface soil layers (layers 1 and 2). S [mm] is the current storage of free water in surface soil layers (layers 1 and 2), refilled by runoff R from filled tension water areas, and drained by surface runoff R_s [mm] (see Knoben et al, 2019). Surface runoff generated depends on a parabolic equation to simulate variable contributing areas of the catchment, requiring maximum free water storage S_{max} (mm) and shape parameter E_x.
Total surface runoff (R_T, [mm]) is calculated as

$$R_T = R_s + R_{im} \qquad \text{(Eq. 3.7.238)}$$

8. OptRunoffSurface = 8 (RunoffSurfaceDynamicVicMod.F90**):**

This is the Dynamic VIC surface runoff scheme (Liang and Xie, 2001). The implementation into Noah-MP is done by McDaniel et al. (2020). This scheme computes saturated area, surface infiltration, and surface runoff based on the VIC model runoff scheme (Liang and Xie, 2003). This version of the VIC scheme dynamically calculates surface runoff based on saturation excess for the saturated area fraction (A_s) and infiltration excess for the (1-A_s) fraction.

Relevant code modules:

Module: SoilWaterInfilPhilipMod.F90
Subroutine: SoilWaterInfilPhilip
Module: SoilWaterInfilGreenAmptMod.F90
Subroutine: SoilWaterInfilGreenAmpt
Module: SoilWaterInfilSmithParlangeMod.F90
Subroutine: SoilWaterInfilSmithParlange
Module: RunoffSurfaceExcessDynamicVicMod.F90
Subroutine: RunoffSatExcessDynamicVic, RunoffInfilExcessDynamicVic

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptDynVicInfiltration	1*	Philip infiltration scheme

options for infiltration in dynamic VIC runoff scheme (i.e., only work with OptRunoffSurface = 8)	2	Green-Ampt infiltration scheme
	3	Smith-Parlange infiltration scheme

The saturation excess runoff (R_{se}) in the surface runoff parameterization is computed over area A_s. The magnitude of R_{se} can be expressed as a function of vertical depth y, which is the difference between precipitation and infiltration excess runoff over a time step Δt for a model grid.

$$R_{se}(y) = \begin{cases} y - \frac{i_m}{b+1} \times \left[\left(1 - \frac{i_0}{i_m}\right)^{b+1} - \left(1 - \frac{i_0+y}{i_m}\right)^{b+1} \right], & 0 \le y < i_m - i_0 \\ R_{se}(y)\,|_{y=i_m-i_0} + y - (i_m - i_0) & , i_m - i_0 < y \le P \end{cases} \quad \text{(Eq. 3.7.239)}$$

where i_m and i_0 are the maximum point soil moisture capacity and the point soil moisture capacity corresponding to the initial soil moisture, respectively. b is the soil moisture capacity shape parameter, and P is the amount of precipitation over a time step Δt.

For the (1-A_s) fraction, the magnitude of infiltration excess runoff R_{ie} is determined as,

$$R_{ie}(y) = \begin{cases} P - R_{se}(y) - f_{mm} \times \Delta t \times \left[1 - \left(1 - \frac{P - R_{se}(y)}{f_m \times \Delta t}\right)^{b+1}\right], & \frac{P - R_{se}(y)}{f_m \times \Delta t} \le 1 \\ P - R_{se}(y) - f_{mm} \times \Delta t & , \frac{P - R_{se}(y)}{f_m \times \Delta t} \ge 1 \end{cases} \quad \text{(Eq. 3.7.240)}$$

where f_{mm} is the average potential infiltration rate over the area of (1-A_s), which can be expressed as,

$$f_{mm} = \frac{f_m}{1+b} \quad \text{(Eq. 3.7.241)}$$

where f_m is the maximum potential infiltration rate, which is a function of soil moisture at each time step and, thus, should vary with time.

With the Dynamic VIC scheme, Noah-MP uses three infiltration measurement methods. Users can select one method using namelist option 'OptDynVicInfiltration'.

Case **OptDynVicInfiltration =1** (Philip's Method): If Philip infiltration equation is used, the spatially averaged point infiltration function is calculated as

$$f_{mm}\,(t) = \frac{S_p}{2} \times t^{-0.5} + K_p \quad \text{(Eq. 3.7.242)}$$

where S_p and K_p are the two parameters and can be estimated based on initial soil moisture and soil properties (Bras, 1990).

Case **OptDynVicInfiltration =2** (Green-Ampt Method):

$$f_{mm}\ (t) = K_{sat} + \left[\left(\frac{G \times \Delta\theta \times \Delta Z}{f_c}\right) \times (K_{sat} - K_\theta)\right] \qquad \text{(Eq. 3.7.243)}$$

where K_{sat} is the saturated hydraulic conductivity, f_c is the cumulative infiltration, K_θ is the K_{sat} at soil moisture content θ. G is a parameter set in the NoahmpTable.TBL.

Case **OptDynVicInfiltration =3** (Smith-Parlange Method):

$$f_{mm}\ (t) = K_{sat} + \frac{\gamma \times (K_{sat} - K_\theta)}{e^{\left(\frac{\gamma \times 10^{-5}}{G \times \Delta\theta \times \Delta Z}\right)} - 1} \qquad \text{(Eq. 3.7.244)}$$

where γ is equal to 0.82. In the above equations, infiltration rate is estimated based on the first soil layer of Noah-MP.

3.7.8.2 Soil Hydraulic Properties

Description:

This part is to compute soil hydraulic conductivity and water diffusivity based on two schemes with different treatments for frozen soil permeability.

Relevant code modules:

Module: SoilHydraulicPropertyMod.F90

Subroutine: SoilDiffusivityConductivityOpt1, SoilDiffusivityConductivityOpt2

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSoilPermeabilityFrozen	1*	Linear effects, more permeable (Niu and Yang, 2006, JHM)
options for frozen soil permeability	2	Nonlinear effects, less permeable (old)

Physics:

1. OptSoilPermeabilityFrozen = 1 (Linear effects, more permeable):

This scheme (SoilDiffusivityConductivityOpt1 subroutine) is based on Niu and Yang (2006, JHM).

The soil water diffusivity (λ_{soil}, [m^2/s]) is expressed as:

$$\lambda_{soil} = (1 - f_{imp,soil}) \times \lambda_{sat} \times \left[\max(0.01, \frac{\theta_{soil}}{\theta_{soil,max}})\right]^{B_{exp}+2} \quad \text{(Eq. 3.7.245)}$$

where λ_{sat} [m^2/s] is the saturated soil water diffusivity (set in NoahmpTable.TBL), $f_{imp,soil}$ is the soil layer impermeable fraction, B_{exp} is the soil exponential coefficient depending on soil texture (set in NoahmpTable.TBL), θ_{soil} [m^3/m^3] and $\theta_{soil,max}$ [m^3/m^3] are the actual and saturated soil moisture, respectively.

The soil hydraulic conductivity ($K_{hyd,soil}$, [m/s]) is expressed as:

$$K_{hyd,soil} = (1 - f_{imp,soil}) \times K_{hyd,sat} \times \left[\max(0.01, \frac{\theta_{soil}}{\theta_{soil,max}})\right]^{2\times B_{exp}+3} \quad \text{(Eq. 3.7.246)}$$

where $K_{hyd,sat}$ [m/s] is the saturated soil hydraulic conductivity (set in NoahmpTable.TBL),

2. OptSoilPermeabilityFrozen = 2 (Nonlinear effects, less permeable):

This scheme is computed in SoilDiffusivityConductivityOpt2 subroutine.

The soil water diffusivity (λ_{soil}, [m^2/s]) is expressed as:

$$\lambda_{soil} = \lambda_{sat} \times \left[\max(0.01, \frac{\theta_{soil}}{\theta_{soil,max}})\right]^{B_{exp}+2} \quad \text{(Eq. 3.7.247)}$$

If soil ice content is greater than zero ($Ice_{soil} > 0$), the soil water diffusivity

$$\lambda_{soil} = \left[\frac{1}{1 + \left(500 \times W_{ice,soil}\right)^3}\right] \times \left[\lambda_{sat} \times \left[\max\left(0.01, \frac{\theta_{soil}}{\theta_{soil,max}}\right)\right]^{B_{exp}+2}\right] + \left[1 - \frac{1}{1 + \left(500 \times W_{ice,soil}\right)^3}\right] \times \left[\lambda_{sat} \times \left[\min\left(\frac{0.05}{\theta_{soil,max}}, \max\left(0.01, \frac{\theta_{soil}}{\theta_{soil,max}}\right)\right)\right]^{B_{exp}+2}\right]$$

(Eq. 3.7.248)

where λ_{sat} [m^2/s] is the saturated soil water diffusivity (set in NoahmpTable.TBL), $f_{imp,soil}$ is the soil layer impermeable fraction, B_{exp} is the soil exponential coefficient depending on soil texture (set in NoahmpTable.TBL), θ_{soil} [m^3/m^3] and $\theta_{soil,max}$ [m^3/m^3] are the actual and saturated soil moisture, respectively.

The soil hydraulic conductivity ($K_{hyd,soil}$, [m/s]) is expressed as:

$$K_{hyd,soil} = K_{hyd,sat} \times \left[\max(0.01, \frac{\theta_{soil}}{\theta_{soil,max}})\right]^{2\times B_{exp}+3} \quad \text{(Eq. 3.7.249)}$$

where $K_{hyd,sat}$ [m/s] is the saturated soil hydraulic conductivity (set in NoahmpTable.TBL),

3.7.8.3 Soil Water Diffusion and Drainage

Description:

This part is to iteratively solve soil water diffusion and drainage processes.

Relevant code modules:

Module: SoilWaterDiffusionRichardsMod.F90

Subroutine: SoilWaterDiffusionRichards

Module: SoilMoistureSolverMod.F90

Subroutine: SoilMoistureSolver

Physics:

1. Noah-MP determines iteration times (N_{iter}) to solve soil water diffusion and moisture at a finer timestep:

$$\Delta t_{fine} = \frac{\Delta t_{soil}}{N_{iter}} \quad \text{(Eq. 3.7.250)}$$

The initial number of iterations (N_{iter}) is set to 3 (hard-coded currently). If the total surface infiltration water amount ($Q_{infil,sfc} \times \Delta t_{soil}$) is greater than the maximum holdable soil water for the first (top) soil layer ($\theta_{soil,max}(1) \times D_{zsnso}(1)$), then the N_{iter} is doubled to achieve a higher accuracy for the soil water solver.

2. Solve soil water diffusion at a finer timestep (Δt_{fine}):

Within each iteration, the following calculations are done to solve and update soil moisture and soil bottom drainage flux:

(1) If the water input on soil surface is greater than zero ($Q_{in,soil} > 0$) and OptRunoffSurface = 3, 6, 7, or 8, then Noah-MP re-calculates the corresponding surface runoff (R_{sfc}) and infiltration ($Q_{infil,sfc}$) at a finer timestep for these runoff options in every iteration and then computes an average surface runoff and infiltration rate for the N_{iter} iterations. The other surface runoff option

schemes (OptRunoffSurface = 1, 2, 4, and 5) do not need to be re-calculated because the infiltration and surface runoff formulation in those schemes are not dependent on timestep.

(2) Calculate the time tendency term of the Richards soil water diffusion equation in SoilWaterDiffusionRichardsMod.F90:

First, the soil hydraulic conductivity ($K_{hyd,soil}$, [m/s]) and diffusivity (λ_{soil}, [m^2/s]) are computed based on OptSoilPermeabilityFrozen (see "Soil Hydraulic Properties" above for details), which are then used in the following soil water diffusion equation calculations. If the MMF groundwater scheme is used (OptRunoffSubsurface=5), a new variable (θ_{ToWT}, [m^3/m^3]) for the soil moisture between soil bottom and the water table is also calculated as follows:

$$\theta_{ToWT} = \begin{cases} \theta_{ToWT,in} & \text{OptSoilPermeabilityFrozen} = 1 \\ \theta_{ToWT,in} \times \frac{W_{liq,soil}(N_{soil})}{\theta_{soil}(N_{soil})} & \text{OptSoilPermeabilityFrozen} = 2 \end{cases} \quad \text{(Eq. 3.7.251)}$$

where $\theta_{ToWT,in}$ [m^3/m^3] is the input soil moisture between soil bottom and the water table computed from last Noah-MP model timestep.

Then, the excess water flux in the diffusion equations ($Q_{soil,diff}$, [m/s]) is computed as:

$$Q_{soil,diff}(i) = \begin{cases} \lambda_{soil}(i) \times G_\theta(i) + K_{hyd,soil}(i) - Q_{infil,sfc} + Q_{soil,transp}(i) + Q_{soil,evap} & i = 1 \\ \lambda_{soil}(i) \times G_\theta(i) + K_{hyd,soil}(i) - \lambda_{soil}(i-1) \times G_\theta(i-1) - K_{hyd,soil}(i-1) + Q_{soil,transp}(i) & 1 < i < N_{soil} \\ -\lambda_{soil}(i-1) \times G_\theta(i-1) - K_{hyd,soil}(i-1) + Q_{soil,transp}(i) + Q_{soil,drain} & i = N_{soil} \end{cases}$$

(Eq. 3.7.252)

Where λ_{soil} [m^2/s] is the soil water diffusivity, $K_{hyd,soil}$ [m/s] is the soil hydraulic conductivity, $Q_{infil,sfc}$ [m/s] is the soil surface infiltration rate, $Q_{soil,transp}$ [m/s] is the soil water loss due plant transpiration, $Q_{soil,evap}$ [m/s] is the soil surface evaporation rate, $Q_{soil,drain}$ [m/s] is the soil bottom free drainage rate, and G_θ [1/m] is the soil moisture gradient between two neighboring layers and defined as:

$$G_\theta(i) = \begin{cases} 2 \times \frac{\theta_s(i) - \theta_s(i+1)}{-Z_{soil}(i+1)} & i = 1 \\ 2 \times \frac{\theta_s(i) - \theta_s(i+1)}{Z_{soil}(i-1) - Z_{soil}(i+1)} & 1 < i < N_{soil} \end{cases} \quad \text{(Eq. 3.7.253)}$$

$$\theta_s(i) = \begin{cases} \theta_{soil}(i) & \text{OptSoilPermeabilityFrozen} = 1 \\ W_{liq,soil}(i) & \text{OptSoilPermeabilityFrozen} = 2 \end{cases} \quad \text{(Eq. 3.7.254)}$$

where Z_{soil} [m] is the depth (negative) of each soil layer bottom and θ_{soil} [m^3/m^3] is the total soil moisture for each layer. $W_{liq,soil}$ [m^3/m^3] is the soil liquid water content for each layer.

Based on the subsurface runoff scheme (OptRunoffSubsurface), $Q_{soil,drain}$ is calculated as:

$$Q_{soil,drain} = \begin{cases} 0 & \text{OptRunoffSubsurface} = 1, 2 \\ S_{drain} \times K_{hyd,soil}(N_{soil}) & \text{OptRunoffSubsurface} = 3, 6, 7, 8 \\ \left(1 - f_{imp,soil,max}\right) \times K_{hyd,soil}(N_{soil}) & \text{OptRunoffSubsurface} = 4 \\ \lambda_{soil}(N_{soil}) \times G_{\theta}(N_{soil}) + K_{hyd,soil} & \text{OptRunoffSubsurface} = 5. \end{cases}$$

(Eq. 3.7.255)

where $f_{imp,soil,max}$ is the maximum soil impermeability fraction throughout the soil column, S_{drain} is the soil drainage slope index (0~1, set in NoahmpTable.TBL) and $G_{\theta}(N_{soil})$ for OptRunoffSubsurface = 5 is computed as:

$$G_{\theta}(N_{soil}) = 2 \times \frac{\theta_s(N_{soil}) - \theta_{s,bot}}{2 \times (Z_{soil}(N_{soil}-1) - Z_{soil}(N_{soil}))}$$ (Eq. 3.7.256)

where $\theta_{s,bot}$ [m^3/m^3] is the soil bottom moisture.

For shallow water table depth ($d_{wt} \geq 2 \times Z_{soil}(N_{soil}) - Z_{soil}(N_{soil} - 1)$; note that d_{wt} is negative for OptRunoffSubsurface = 5):

$$\theta_{s,bot} = \theta_{ToWT}$$ (Eq. 3.7.257)

For deep water table depth ($d_{wt} < 2 \times Z_{soil}(N_{soil}) - Z_{soil}(N_{soil} - 1)$), linear interpolation from the water table depth to the middle of auxiliary layer below the soil bottom:

$$\theta_{s,bot} = \theta_s(N_{soil}) - (\theta_s(N_{soil}) - \theta_{ToWT}) \times \frac{2 \times (Z_{soil}(N_{soil} - 1) - Z_{soil}(N_{soil}))}{(Z_{soil}(N_{soil} - 1) - Z_{soil}(N_{soil})) + Z_{soil}(N_{soil}) - d_{wt}}$$

(Eq. 3.7.258)

Specifically, the d_{wt} for OptRunoffSurface = 5 is computed using the MMF groundwater scheme (ShallowWaterTableMmfMod.F90 and GroundWaterMmfMod.F90). Please see the subsurface runoff part and groundwater part below for details.

(3) Compute (prepare) the matrix coefficients for the tri-diagonal matrix of the implicit time scheme in SoilWaterDiffusionRichardsMod.F90:

The left-hand side of the matrix coefficients for the tri-diagonal matrix are classified to the three groups (A_I, B_I, C_I; [1/s]):

For the first soil layer (i = 1):

$$D_{DZ}(i) = \frac{2}{-Z_{soil}(i+1)} \quad \text{(Eq. 3.7.259)}$$

$$\begin{cases} A_I(i) = 0.0 \\ B_I(i) = \lambda_{soil}(i) \times \frac{D_{DZ}(i)}{-Z_{soil}(i)} \\ C_I(i) = -B_I(i) \end{cases} \quad \text{(Eq. 3.7.260)}$$

For the middle soil layers ($1 < i < N_{soil}$):

$$D_{DZ}(i) = \frac{2}{Z_{soil}(i-1) - Z_{soil}(i+1)} \quad \text{(Eq. 3.7.261)}$$

$$\begin{cases} A_I(i) = -\lambda_{soil}(i-1) \times \frac{D_{DZ}(i-1)}{Z_{soil}(i-1) - Z_{soil}(i)} \\ B_I(i) = -(A_I(i) + C_I(i)) \\ C_I(i) = -\lambda_{soil}(i) \times \frac{D_{DZ}(i)}{Z_{soil}(i-1) - Z_{soil}(i)} \end{cases} \quad \text{(Eq. 3.7.262)}$$

For the bottom soil layer ($i = N_{soil}$):

$$\begin{cases} A_I(i) = -\lambda_{soil}(i-1) \times \frac{D_{DZ}(i-1)}{Z_{soil}(i-1) - Z_{soil}(i)} \\ B_I(i) = -(A_I(i) + C_I(i)) \\ C_I(i) = 0.0 \end{cases} \quad \text{(Eq. 3.7.263)}$$

For all soil layers, the right-hand side (R_{HS}, [1/s]) of the matrix coefficients for the tri-diagonal matrix is calculated as:

$$R_{HS}(i) = \begin{cases} \frac{Q_{soil,diff}(i)}{Z_{soil}(i)} & i = 1 \\ \frac{Q_{soil,diff}(i)}{Z_{soil}(i) - Z_{soil}(i-1)} & i \leq N_{soil} \end{cases} \quad \text{(Eq. 3.7.264)}$$

The A_I, B_I, C_I, and R_{HS} [1/s] are used in the following soil moisture tri-diagonal matrix solver in SoilMoistureSolverMod.F90.

(4) Solve the tri-diagonal matrix and update soil moisture in SoilMoistureSolverMod.F90 shown below.

Relevant code modules:

Module: MatrixSolverTriDiagonalMod.F90

Subroutine: MatrixSolverTriDiagonal

First, update the matrix coefficients to solve the saturation excess water for each soil layer:

$$\begin{cases} R_{HS}(i) = R_{HS}(i) \times \Delta t_{fine} \\ A_I(i) = A_I(i) \times \Delta t_{fine} \\ B_I(i) = 1 + B_I(i) \times \Delta t_{fine} \\ C_I(i) = C_I(i) \times \Delta t_{fine} \end{cases} \quad \text{(Eq. 3.7.265)}$$

Then, these matrix coefficients are input to the tri-diagonal matrix solver in MatrixSolverTriDiagonalMod.F90 (see below for details), which updates $C_I(i)$.

The soil liquid water content ($W_{liq,soil}$, [m^3/m^3]) for each soil layer is then updated based on $C_I(i)$:

$$W_{liq,soil}(i) = W_{liq,soil}(i) + C_I(i) \quad \text{(Eq. 3.7.266)}$$

If OptRunoffSubsurface = 5 (using MMF groundwater scheme), there is also soil moisture in the auxiliary layer between the soil bottom and water table (θ_{ToWT}), which needs to be updated:

For deep water table depth ($d_{wt} < Z_{soil}(N_{soil}) - D_{zsnso}(N_{soil})$, note that d_{wt} is negative for OptRunoffSubsurface = 5), $Q_{soil,drain}$ [m/s] is accumulated to a deep groundwater recharge term ($W_{rech,deep}$, [m]) as follows in order to update water table and soil moisture later in GroundWaterMmfMod.F90:

$$W_{rech,deep} = W_{rech,deep} + Q_{soil,drain} \times \Delta t_{fine} \quad \text{(Eq. 3.7.267)}$$

For shallow water table depth ($d_{wt} \geq Z_{soil}(N_{soil}) - D_{zsnso}(N_{soil})$), $Q_{soil,drain}$ [m/s] is accumulated to the soil moisture in the auxiliary layer (θ_{ToWT}):

$$\theta_{ToWT} = \theta_{ToWT} + \frac{Q_{soil,drain} \times \Delta t_{fine}}{D_{zsnso}(N_{soil})} \quad \text{(Eq. 3.7.268)}$$

Then the excess water above saturation in the auxiliary layer (W_{excess}, [m]) is

$$W_{excess} = \max(0, \theta_{ToWT} - \theta_{soil,max}(N_{soil})) \times D_{zsnso}(N_{soil}) \quad \text{(Eq. 3.7.269)}$$

The water deficiency in the auxiliary layer (W_{defi}, [m]) is

$$W_{defi} = \max(0, 10^{-4} - \theta_{ToWT}) \times D_{zsnso}(N_{soil}) \quad \text{(Eq. 3.7.270)}$$

The soil moisture in the auxiliary layer (θ_{ToWT}) is further adjusted as:

$$\theta_{ToWT} = \max(10^{-4}, \min\left(\theta_{ToWT}, \theta_{soil,max}(N_{soil})\right)) \quad \text{(Eq. 3.7.271)}$$

The excess water from the auxiliary layer (W_{excess}) is then added to the soil water content in the bottom layer:

$$W_{liq,soil}(N_{soil}) = W_{liq,soil}(N_{soil}) + \frac{W_{excess}}{D_{zsnso}(N_{soil})} \quad \text{(Eq. 3.7.272)}$$

Then the drainage flux at the soil column bottom is reduced accordingly:

$$Q_{soil,drain} = Q_{soil,drain} - \frac{W_{excess}}{\Delta t_{fine}} \quad \text{(Eq. 3.7.273)}$$

If there is water deficiency, the deep groundwater recharge term ($W_{rech,deep}$, [m]) is also reduced:

$$W_{rech,deep} = W_{rech,deep} - W_{defi} \quad \text{(Eq. 3.7.274)}$$

For all the Noah-MP soil layers (for all runoff schemes), excess water above saturation (W_{excess}, [m]) in each layer is moved to an unsaturated layer like in a bucket (first from the bottom layer to the top layer and then from the top layer to the bottom layer) as follows:

For the soil layer i from N_{soil} to 2:

$$P_{soil,eff} = \max(10^{-4}, \theta_{soil,max}(i) - W_{ice,soil}(i)) \quad \text{(Eq. 3.7.275)}$$

$$W_{excess} = \max(0, W_{liq,soil}(i) - P_{soil,eff}) \times D_{zsnso}(i) \quad \text{(Eq. 3.7.276)}$$

$$W_{liq,soil}(i) = \min(W_{liq,soil}(i), P_{soil,eff}) \quad \text{(Eq. 3.7.277)}$$

$$W_{liq,soil}(i-1) = W_{liq,soil}(i-1) + \frac{W_{excess}}{D_{zsnso}(i-1)} \quad \text{(Eq. 3.7.278)}$$

where $P_{soil,eff}$ [m^3/m^3] is the effective soil porosity.

For the first soil layer $i = 1$:

$$P_{soil,eff} = \max(10^{-4}, \theta_{soil,max}(1) - W_{ice,soil}(1)) \quad \text{(Eq. 3.7.279)}$$

$$W_{excess} = \max(0, W_{liq,soil}(1) - P_{soil,eff}) \times D_{zsnso}(1) \quad \text{(Eq. 3.7.280)}$$

$$W_{liq,soil}(1) = \min(W_{liq,soil}(1), P_{soil,eff}) \quad \text{(Eq. 3.7.281)}$$

If the excessive water is still greater than zero ($W_{excess} > 0$), then further move the excessive water from the top to the bottom soil layer, similar to the process above:

$$W_{liq,soil}(2) = W_{liq,soil}(2) + \frac{W_{excess}}{D_{zsnso}(2)} \quad \text{(Eq. 3.7.282)}$$

For the soil layer i from 2 to $N_{soil} - 1$:

$$P_{soil,eff} = \max(10^{-4}, \theta_{soil,max}(i) - W_{ice,soil}(i)) \quad \text{(Eq. 3.7.283)}$$

$$W_{excess} = \max(0, W_{liq,soil}(i) - P_{soil,eff}) \times D_{zsnso}(i) \quad \text{(Eq. 3.7.284)}$$

$$W_{liq,soil}(i) = \min(W_{liq,soil}(i), P_{soil,eff}) \quad \text{(Eq. 3.7.285)}$$

$$W_{liq,soil}(i+1) = W_{liq,soil}(i+1) + \frac{W_{excess}}{D_{zsnso}(i+1)} \quad \text{(Eq. 3.7.286)}$$

For the bottom soil layer $i = N_{soil}$:

$$P_{soil,eff} = \max(10^{-4}, \theta_{soil,max}(N_{soil}) - W_{ice,soil}(N_{soil})) \quad \text{(Eq. 3.7.287)}$$

$$W_{excess} = \max(0, W_{liq,soil}(N_{soil}) - P_{soil,eff}) \times D_{zsnso}(N_{soil}) \quad \text{(Eq. 3.7.288)}$$

$$W_{liq,soil}(N_{soil}) = \min(W_{liq,soil}(N_{soil}), P_{soil,eff}) \qquad \text{(Eq. 3.7.289)}$$

The remaining excess water above saturation (W_{excess}) will be accumulated to surface runoff as shown above.

Finally, the total soil moisture is updated:

$$\theta_{soil} = W_{liq,soil} + W_{ice,soil} \qquad \text{(Eq. 3.7.290)}$$

3. Update surface runoff and bottom drainage:

After the soil water solver computation (see above) at a finer timestep through N_{iter} iteration, the accumulated soil water saturation excess is:

$$W_{excess,tot} = \sum_{t=1}^{N_{iter}} W_{excess}(t) \qquad \text{(Eq. 3.7.291)}$$

the total surface runoff (R_{sfc}, [mm/s]) with unit conversion from m/s to mm/s is:

$$R_{sfc,tot} = \frac{\sum_{t=1}^{N_{iter}} R_{sfc}(t) \times 1000}{N_{iter}} + \frac{W_{excess,tot} \times 1000}{\Delta t_{soil}} \qquad \text{(Eq. 3.7.292)}$$

The total soil bottom drainage ($Q_{soil,drain,tot}$, [mm/s]) with unit conversion from m/s to mm/s is:

$$Q_{soil,drain,tot} = \frac{\sum_{t=1}^{N_{iter}} Q_{soil,drain}(t) \times 1000}{N_{iter}} \qquad \text{(Eq. 3.7.293)}$$

4. Tri-diagonal matrix solver in MatrixSolverTriDiagonalMod.F90:

```
INVERT (SOLVE) THE TRI-DIAGONAL MATRIX PROBLEM SHOWN BELOW:
###                                            ### ###   ###    ###   ###
#B(1), C(1),  0  ,  0  ,  0  ,   . . . ,    0   # #       #    #       #
#A(2), B(2), C(2),  0  ,  0  ,   . . . ,    0   # #       #    #       #
# 0  , A(3), B(3), C(3),  0  ,   . . . ,    0   # #       #    # D(3) #
# 0  ,  0  , A(4), B(4), C(4),   . . . ,    0   # # P(4) #    # D(4) #
# 0  ,  0  ,  0  , A(5), B(5),   . . . ,    0   # # P(5) #    # D(5) #
# .                                         .   # #  .   # = #   .   #
# .                                         .   # #  .   #   #   .   #
# .                                         .   # #  .   #   #   .   #
# 0  , . . . , 0 , A(M-2), B(M-2), C(M-2),  0   # #P(M-2)#   #D(M-2)#
# 0  , . . . , 0 ,   0   , A(M-1), B(M-1), C(M-1)# #P(M-1)#   #D(M-1)#
# 0  , . . . , 0 ,   0   ,   0   ,  A(M) ,  B(M) # # P(M) #   # D(M) #
###                                            ### ###   ###    ###   ###
```

The matrix coefficients (A, B, C, D) are inputs (A_i, B_i, C_i, R_{HS}) coming from the calculations above in SoilWaterDiffusionRichardsMod.F90 and SoilMoistureSolverMod.F90. To solve (invert) the tri-diagonal matrix:

First, initialize equation coefficient:

$$C(N_{soil}) = 0.0 \quad \text{(Eq. 3.7.294)}$$

$$P(N_{top}) = \frac{-C(N_{top})}{B(N_{top})} \quad \text{(Eq. 3.7.295)}$$

where N_{top} (= 1) is the first soil layer.

Then, solve the equation coefficients for the first soil layer:

$$\Delta(N_{top}) = \frac{D(N_{top})}{B(N_{top})} \quad \text{(Eq. 3.7.296)}$$

Solve the equation coefficients for soil layers *i* from 2 through N_{soil}:

$$P(i) = -C(i) \times \left[\frac{1}{B(i)+A(i)\times P(i-1)}\right] \quad \text{(Eq. 3.7.297)}$$

$$\Delta(i) = [D(i) - A(i) \times \Delta(i-1)] \times \left[\frac{1}{B(i)+A(i)\times P(i-1)}\right] \quad \text{(Eq. 3.7.298)}$$

Set *P* to Δ for the lowest soil layer:

$$P(N_{soil}) = \Delta(N_{soil}) \quad \text{(Eq. 3.7.299)}$$

Adjust *P* for soil layers *i* from 2 through N_{soil}:

$$ii = N_{soil} - i + (N_{top} - 1) + 1 \quad \text{(Eq. 3.7.300)}$$

$$P(ii) = P(ii) \times P(ii+1) + \Delta(N_{soil}) \quad \text{(Eq. 3.7.301)}$$

Here *P* is the output variable that is further used to update soil liquid water content as shown above.

3.7.8.4 Tile Drainage

Description:

This part is to compute the tile drainage process and update soil moisture and tile drainage discharge. The purpose of the tile drainage scheme is to estimate tile drainage discharge from the 2-m soil column. Currently, there are two tile drainage schemes in Noah-MP. The simple tile drainage scheme estimates the amount of water drained based on field capacity of Noah-MP soil layers whereas the Hooghoudt's tile drainage computes steady-state flow into the tile by applying Dupuit-Forchheimer assumptions for horizontal flow in an unconfined aquifer and Darcy's equation (Valayamkunnath et al., 2022).

Relevant code modules:

Module: TileDrainageSimpleMod.F90

Subroutines: TileDrainageSimple

Module: TileDrainageHooghoudtMod.F90

Subroutines: TileDrainageHooghoudt

Submodule: TileDrainageEquiDepthMod.F90

Subroutines: TileDrainageEquiDepth

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptTileDrainage	0*	No tile drainage
	1	on (simple scheme)
options for tile drainage currently only tested & calibrated to work with runoff option=3	2	on (Hooghoudt's scheme)

Physics:

1. Simple Tile Drainage Scheme (TileDrainageSimpleMod.F90)

The simple model estimates tile drainage as the excess water above field capacity of the soil and removes it from the user specified layer/layers as follows:

Effective field capacity in the i^{th} layer, $\theta_{FC,eff}(i)$ (volumetric fraction) is calculated as

$$\theta_{FC,eff}(\mathrm{i}) = \theta_{FC}(\mathrm{i}) - \theta_{ice}(\mathrm{i}) \quad \text{(Eq. 3.7.302)}$$

Saturation water, $Q_{sat}(\mathrm{i})$ [mm], is calculated as

$$Q_{sat}(\mathrm{i}) = \theta_{eff}(\mathrm{i}) - \theta_{FC,eff}(\mathrm{i}) \times F_{TD} \times \Delta Z \quad \text{(Eq. 3.7.303)}$$

Tile drained water [mm] is calculated as

$$Q_{TD} = \begin{cases} \sum_{i=1}^{N_{soil}} Q_{sat}(\mathrm{i}) & 0 < Q_{sat}(\mathrm{i}) \le D_c \\ 0 & Q_{sat}(\mathrm{i}) \le 0 \end{cases} \quad \text{(Eq. 3.7.304)}$$

where θ_{FC} is the field capacity (volumetric), θ_{ice} is the frozen soil moisture fraction (volumetric), F_{TD} is the multiplier factor for effective field capacity (volumetric), ΔZ [mm] is the thickness of the soil, and D_c is the tile drainage coefficient (provided in NoahmpTable.TBL). When the simple tile drainage scheme is used, a user can specify the layer to drain using "DrainSoilLayerInd" parameter provided in the NoahmpTable.TBL.

Physical Processes	Options	Details

DrainSoilLayerInd option for layer to drain	**0	from one specified layer using ‘TileDrainTubeDepth’
	1	from layers 1 & 2
	2	from layer layers 1, 2, and 3
	3	from layer 2 and 3
	4	from layer layers 3, 4
	5	from all the four layers

2. Hooghoudt’s Tile Drainage Scheme (TileDrainageHooghoudtMod.F90):

Hooghoudt's model computes steady-state flow into the tile by applying Dupuit-Forchheimer assumptions for horizontal flow in an unconfined aquifer and Darcy's equation (Valayamkunnath et al., 2022). The Hooghoudt's tile-drainage model is computationally simple, and therefore is commonly used to compute the tile drainage runoff in hydrology models. The Hoogoudt’s tile drainage formulation implemented into the Noah-MP/WRF-Hydro is as follows:

(1) Water table depth from land surface (Z_{WT}, [m]): Water table depth estimated by Noah-MP is employed.

(2) Effective lateral hydraulic conductivity of the soil column (K_{lat}, [m/s]) is computed as follows:

Thickness of saturated zone (T_s):

$$T_s(i) = \begin{cases} 0.0 & Z_{WT} > Z_{soil}(i) \\ Z_{soil}(i) - Z_{WT} & Z_{WT} < Z_{soil}(i) \end{cases} \quad \text{(Eq. 3.7.305)}$$

$$T_s = \sum_{i=1}^{N_{soil}} T_s(i) \quad \text{(Eq. 3.7.306)}$$

where $Z_{soil}(i)$ [m] is the depth to i^{th} soil layer bottom from land surface.

Total lateral flow through the saturated thickness per unit width of the soil column (q_x):

$$q_x = \sum_{i=1}^{N_{soil}} K_{lat}(i) \times T_s(i) \quad \text{(Eq. 3.7.307)}$$

$$K_{lat} = \frac{q_x}{T_s} \quad \text{(Eq. 3.7.308)}$$

(3) Equivalent depth calculation:

The tile drainage equivalent depth is calculated as

$$d_e = \frac{1}{8} \times \frac{\pi \times L}{\ln\left(\frac{L}{\pi \times r}\right) + F(x)} \quad \text{(Eq. 3.7.309)}$$

$$x = \frac{2 \times \pi \times D}{L} \quad \text{(Eq. 3.7.310)}$$

$$F(x) = \begin{cases} \frac{\pi^2}{4 \times x} + \ln\left(\frac{x}{2 \times \pi}\right) & x \le 0.5 \\ \sum_{n=1}^{\infty} \frac{4 \times e^{-2 \times n \times x}}{n \times (1 - e^{-2 \times n \times x})} & x > 0.5 \quad (n = 1,3,5,\ldots) \end{cases} \quad \text{(Eq. 3.7.311)}$$

where r [m] is the tile radius,

(4) Tile drainage estimation (q, [m/s]) using Hoogoudt's equation:

$$q = \begin{cases} 0 & Z_{WT} > Tile\ depth \\ \frac{8 \times K_{lat} \times D \times h + 4 \times K_{lat} \times h^2}{L^2} & Z_{WT} < Tile\ depth \end{cases} \quad \text{(Eq. 3.7.312)}$$

Tile flow per model time step:

$$Q = q \times \Delta \text{t} \quad \text{(Eq. 3.7.313)}$$

where Δt [s] is the model time step.

(5) Amount of water above field capacity (Q_{sat}):

Q_{sat} in each soil layer:

$$Q_{sat}(i) = \begin{cases} \theta_{soil}(i) - \theta_{FC}(i) & \theta_{soil}(i) > \theta_{FC}(i) \\ 0.0 & \theta_{soil}(i) \leq \theta_{FC}(i) \end{cases} \quad \text{(Eq. 3.7.314)}$$

where θ_{soil} [m^3/m^3] and θ_{FC} [m^3/m^3] are the soil moisture and the volumetric field capacity of the soil, respectively.

Q_{sat} in all the soil layers:

$$Q_{sat} = \sum_{i=1}^{N_{soil}} Q_{sat}(i) \times \Delta Z_{soil}(\text{i}) \quad \text{(Eq. 3.7.315)}$$

Limit tile flow Q with Q_{sat}: if ($Q > Q_{sat}$), then

$$Q = Q_{sat} \quad \text{(Eq. 3.7.316)}$$

$$Q_{TD} = Q \quad \text{(Eq. 3.7.317)}$$

where Q_{TD} [mm] is the tile drainage flow per model timestep.

(6) Update WRF-Hydro/NWM soil moisture after tile drainage:

Loop over each soil layer from $i = 1\ to\ i = N_{soil}$,

$$Q_{RM}(i) = Q_{sat}(i) - Q \quad \text{(Eq. 3.7.318)}$$

If ($Q_{RM} > 0.0$):

$$\theta_{soil}(i) = \theta_{FC}(i) + \frac{Q_{RM}(i)}{\Delta Z_{soil}(\text{i})} \quad \text{(Eq. 3.7.319)}$$

else:

$$\theta_{soil}(i) = \theta_{FC}(i) \quad \text{(Eq. 3.7.320)}$$

$$Q = Q - Q_{sat}(i) \quad \text{(Eq. 3.7.321)}$$

where Q_{RM} is the temporary variable in the soil moisture update loop, ΔZ_{soil} [mm] is the soil layer thickness, and i is the soil layer index from 1 to N_{soil}.

3.7.8.5 Subsurface Runoff

Description:

This part is to compute subsurface runoff fluxes based on different physics options. Currently, Noah-MP is only tested for selecting the same surface and subsurface runoff options.

Relevant code modules:

Module: RunoffSubSurfaceEquiWaterTableMod.F90

Subroutine: RunoffSubSurfaceEquiWaterTable

Module: RunoffSubSurfaceGroundWaterMod.F90

Subroutine: RunoffSubSurfaceGroundWater

Module: RunoffSubSurfaceDrainageMod.F90

Subroutine: RunoffSubSurfaceDrainage

Module: RunoffSubSurfaceShallowMmfMod.F90

Subroutine: RunoffSubSurfaceShallowWaterMMF

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptRunoffSubsurface Option for subsurface runoff currently tested & recommended the same option# as surface runoff	1	TOPMODEL with groundwater
	2	TOPMODEL with an equilibrium water table
	3*	Drainage-dependent subsurface runoff
	4	Drainage-dependent subsurface runoff
	5	Miguez-Macho&Fan groundwater subsurface runoff
	6	Drainage-dependent subsurface runoff
	7	Drainage-dependent subsurface runoff
	8	Drainage-dependent subsurface runoff

Physics:

1. OptRunoffSubsurface = 2 (RunoffSubSurfaceEquiWaterTableMod.F90**):**

The TOPMODEL with an equilibrium water table scheme (Niu et al., 2005) is used. This scheme is to calculate subsurface runoff using equilibrium water table depth.

Relevant code modules:

Module: WaterTableEquilibriumMod.F90

Subroutine: WaterTableEquilibrium

(1) The equilibrium water table depth is computed in WaterTableEquilibriumMod.F90 as follows: The initial value of the water table is assumed to be at a depth (d_{wt}, [m], positive) of 3 times the default Noah-MP soil column depth ($|Z_{soil}(N_{soil})|$, [m]):

$$d_{wt} = -3 \times Z_{soil}(N_{soil}) - 0.001 \quad \text{(Eq. 3.7.322)}$$

where $Z_{soil}(i)$ [m] is the depth (negative) of i^{th} soil layer bottom.

The original Noah-MP soil columns with N_{soil} layers are then extended to a hypothetical finer, deeper soil column with N_{fine} (=100 hard-coded currently) layers with a depth of $|3 \times Z_{soil}(N_{soil})|$:

$$D_{Z,fine} = \frac{-3 \times Z_{soil}(N_{soil})}{N_{fine}} \quad \text{(Eq. 3.7.323)}$$

$$Z_{fine}(i) = i \times D_{Z,fine} \quad \text{(Eq. 3.7.324)}$$

where $D_{Z,fine}$ [m] is the finer soil layer thickness (set to equal for each layer), $Z_{fine}(i)$ [m] is the depth (positive) of i^{th} fine soil layer bottom.

The total water deficit ($D_{\theta,org}$, [m]) for the original (coarse) soil column with a depth of $|Z_{soil}(N_{soil})|$ and N_{soil} layers is:

$$D_{\theta,org} = \sum_{i=1}^{N_{soil}} ((\theta_{soil,max} - W_{liq,soil}(i)) \times D_{zsnso}(i)) \quad \text{(Eq. 3.7.325)}$$

The total water deficit ($D_{\theta,org}$, [m]) accumulated until the L^{th} layer for the hypothetical finer, deeper soil column with N_{fine} layers with a depth of $|3 \times Z_{soil}(N_{soil})|$ is:

$$D_{\theta,fine}(L) = \sum_{i=1}^{L} \left\{ \theta_{soil,max} \times \left[1 - \left(1 + \frac{(d_{wt} - Z_{fine}(i))}{\varphi_{sat}} \right)^{-\frac{1}{B_{exp}}} \right] \times D_{Z,fine} \right\} \quad \text{(Eq. 3.7.326)}$$

where $\theta_{soil,max}$ [m^3/m^3] is the maximum (saturated) soil moisture, φ_{sat} [m] is the saturated soil matrix potential, and B_{exp} is the soil B parameter. Each of these three parameters is set in the NoahmpTable.TBL.

If $|D_{\theta,fine}(L) - D_{\theta,org}| \leq 0.01$, then

$$d_{wt} = Z_{fine}(L) \quad \text{(Eq. 3.7.327)}$$

(2) Compute subsurface runoff (R_{sub}, [mm/s]) based on the equilibrium water table depth (d_{wt}):

$$R_{sub} = (1 - f_{imp,soil,max}) \times C_{baseflow} \times e^{-I_{topo}} \times e^{(-F_{rf,decay} \times d_{wt})} \quad \text{(Eq. 3.7.328)}$$

where $f_{imp,soil,max}$ is the maximum impermeable fraction due to frozen soil, $C_{baseflow}$ [mm/s] is the base flow coefficient (=4.0 for OptRunoffSubsurface = 2), $F_{rf,decay}$ [1/m] is the runoff decay factor (=2.0 for OptRunoffSubsurface = 2), and I_{topo} is the gridcell mean topographic index (=10.5 set in the NoahmpTable.TBL).

(3) After the soil moisture solver iteration described in Sect 3.4.9.3, removal of soil water due to groundwater flow is needed for OptRunoffSubsurface = 2 as follows:

At each soil layer, the amount of soil water removed (W_{rm}, [mm]) is:

$$W_{rm}(i) = R_{sub} \times \Delta t_{soil} \times \frac{K_{hyd,soil}(\mathrm{i}) \times D_{zsnso}(i)}{\sum_{j=1}^{N_{soil}} (K_{hyd,soil}(\mathrm{j}) \times D_{zsnso}(j))} \quad \text{(Eq. 3.7.329)}$$

where $K_{hyd,soil}$ [m/s] is the soil hydraulic conductivity.

Then the soil liquid water content ($W_{liq,soil}$, [m^3/m^3]) for each layer is updated as:

$$W_{liq,soil}(i) = W_{liq,soil}(i) - \frac{W_{rm}(i)}{D_{zsnso}(i) \times 1000} \quad \text{(Eq. 3.7.330)}$$

2. Treatment for too low soil water for OptRunoffSubsurface ≠ 1:

Noah-MP limits the soil liquid water above a minimum threshold ($W_{soil,min}$ = 0.01 mm, currently hard-coded). If the soil water is too low, then water from a lower layer will be brought to increase the water to the minimum threshold.

First, the current soil water amount ($W_{soil,liq}$, [mm]) is calculated as:

$$W_{soil,liq}(i) = W_{liq,soil}(i) \times D_{zsnso}(i) \times 1000 \quad \text{(Eq. 3.7.331)}$$

Then, for soil layers from 1 (top) to $N_{soil} - 1$:

If $W_{soil,liq}(i) < 0.0$, then the water is brought to this layer from the lower layer to reach $W_{soil,min}$:

$$W_{soil,liq}(i) = W_{soil,liq}(i) + (W_{soil,min} - W_{soil,liq}(i)) \quad \text{(Eq. 3.7.332)}$$

$$W_{soil,liq}(i+1) = W_{soil,liq}(i+1) - (W_{soil,min} - W_{soil,liq}(i)) \quad \text{(Eq. 3.7.333)}$$

For the last soil layer ($i = N_{soil}$):

If $W_{soil,liq}(N_{soil}) < W_{soil,min}$, then the water is brought to this layer from the subsurface runoff to reach $W_{soil,min}$:

$$W_{soil,liq}(N_{soil}) = W_{soil,liq}(N_{soil}) + (W_{soil,min} - W_{soil,liq}(N_{soil})) \quad \text{(Eq. 3.7.334)}$$

$$R_{sub} = R_{sub} - \frac{W_{soil,min} - W_{soil,liq}(N_{soil})}{\Delta t_{soil}} \quad \text{(Eq. 3.7.335)}$$

For OptRunoffSubsurface = 5, the water is taken from the deep recharge ($W_{rech,deep}$, [m]) instead:

$$W_{rech,deep} = W_{rech,deep} - (W_{soil,min} - W_{soil,liq}(N_{soil})) \times 0.001 \quad \text{(Eq. 3.7.336)}$$

Note that R_{sub} has no physical meaning and should not be used for OptRunoffSubsurface = 5. The subsurface runoff uses another variable "QRF" in GroundWaterMmfMod.F90 for OptRunoffSubsurface = 5.

Finally, the soil volume liquid water content ($W_{liq,soil}(i)$, [m^3/m^3]) is updated:

$$W_{liq,soil}(i) = \frac{W_{soil,liq}(i)}{D_{zsnso}(i) \times 1000} \quad \text{(Eq. 3.7.337)}$$

3. OptRunoffSubsurface = 3, 4, 6, 7, 8 (RunoffSubSurfaceDrainageMod.F90**):**

For OptRunoffSubsurface = 3, 4, 6, 7, 8, the simple drainage-dependent subsurface runoff scheme is used:

$$R_{sub} = R_{sub} + Q_{soil,drain} \quad \text{(Eq. 3.7.338)}$$

where $Q_{soil,drain}$ [mm/s] is the soil water drainage computed in earlier sections.

4. OptRunoffSubsurface = 1 (RunoffSubSurfaceGroundWaterMod**):**

The TOPMODEL with groundwater scheme (Niu et al., 2007) is used in RunoffSubSurfaceGroundWaterMod. This scheme is to calculate subsurface runoff using groundwater discharge.

Relevant code modules:

Module: GroundWaterTopModelMod.F90

Subroutine: GroundWaterTopModel

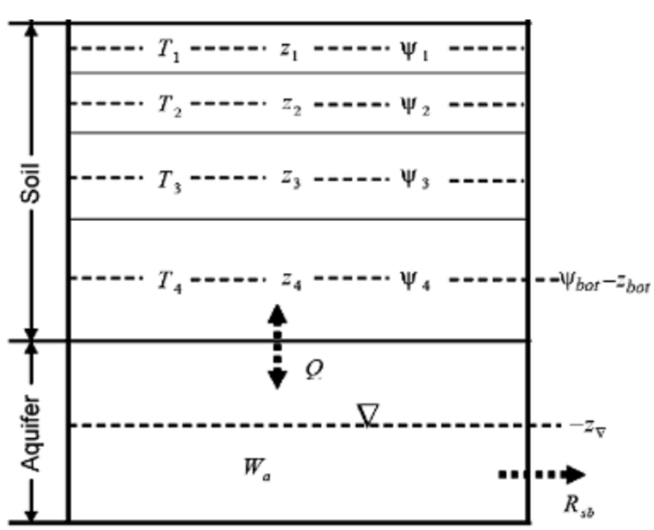


Figure 13. A demonstration of four soil layers and an aquifer layer for the OptRunoffSubsurface = 1 (TOPMODEL with groundwater) in Noah-MP. Adapted from Niu et al. (2007).

As shown in the figure above, an unconfined aquifer is defined as the part below the soil column in Noah-MP. The dynamics of water storage in the aquifer ($W_{aquifer}$, [mm]) can be expressed as:

$$\frac{\Delta W_{aquifer}}{\Delta t_{soil}} = Q_{rech} - Q_{dis} \qquad \text{(Eq. 3.7.339)}$$

where Q_{rech} [mm/s] and Q_{dis} [mm/s] are the recharge to and discharge from the aquifer, respectively, which are computed below.

(1) Prepare for calculations of recharge and discharge fluxes:

Calculate soil layer thickness (dz_{soil}, [mm]):

$$dz_{soil}(i) = \begin{cases} -Z_{soil}(1) \times 1000 & i = 1 \\ \left(Z_{soil}(i-1) - Z_{soil}(i)\right) \times 1000 & i = 2 \sim N_{soil} \end{cases} \qquad \text{(Eq. 3.7.340)}$$

Calculate the node (middle point) depth (Z_{node}, [m], positive) of each soil layer:

$$Z_{node}(i) = \begin{cases} \frac{-Z_{soil}(1)}{2} & i = 1 \\ -Z_{soil}(i-1) + \frac{Z_{soil}(i-1) - Z_{soil}(i)}{2} & i = 2 \sim N_{soil} \end{cases} \qquad \text{(Eq. 3.7.341)}$$

Calculate soil water quantities:

$$\theta_{soil} = W_{liq,soil} + W_{ice,soil} \qquad \text{(Eq. 3.7.342)}$$

$$W_{soil,liq} = W_{liq,soil} \times dz_{soil} \qquad \text{(Eq. 3.7.343)}$$

$$P_{soil,eff} = \max(0.01, \theta_{soil,max} - W_{ice,soil}) \qquad \text{(Eq. 3.7.344)}$$

where θ_{soil} [m^3/m^3] is the total soil moisture, $W_{liq,soil}$ [m^3/m^3] and $W_{ice,soil}$ [m^3/m^3] are the soil liquid water and ice content, respectively, $W_{soil,liq}$ [mm] is the soil liquid water amount, $P_{soil,eff}$ [m^3/m^3] is the effective soil porosity, and $\theta_{soil,max}$ [m^3/m^3] is the maximum (saturated) soil moisture (set in NoahmpTable.TBL).

(2) Compute the layer index of first unsaturated layer (the layer above water table):

Looping over soil layer from 2 to N_{soil}, if the water table depth (d_{wt}, [m], positive) is lower than or equal to the depth of a certain soil layer, then the layer index of first unsaturated layer (I_{unsat}) right above water table is assigned:

$$I_{unsat} = iz \qquad if\ d_{wt} \le -Z_{soil}(iz) \qquad \text{(Eq. 3.7.345)}$$

(3) Compute groundwater discharge (Q_{dis}, [mm/s]) and subsurface runoff (R_{sub}, [mm/s]):

$$Q_{dis} = (1 - f_{imp,soil,max}) \times C_{baseflow} \times e^{-I_{topo}} \times e^{[-F_{rf,decay} \times d_{wt}]} \quad \text{(Eq. 3.7.346)}$$

$$F_{rf,decay} = \frac{B_{exp}}{3} \quad \text{(Eq. 3.7.347)}$$

$$C_{baseflow} = K_{hyd,soil}(I_{unsat}) \times 1000 \times e^3 \quad \text{(Eq. 3.7.348)}$$

$$R_{sub} = Q_{dis} \quad \text{(Eq. 3.7.349)}$$

where $f_{imp,soil,max}$ is the maximum impermeable fraction due to frozen soil, $C_{baseflow}$ [mm/s] is the base flow coefficient, $K_{hyd,soil}$ [mm/s] is the soil water conductivity, $F_{rf,decay}$ [1/m] is the runoff decay factor, B_{exp} is the soil exponential *B* parameter (set in the NoahmpTable.TBL), I_{topo} is the global gridcell mean topographic index (=10.5 set in the NoahmpTable.TBL).

(4) Compute soil matric potential at the layer above the water table (first unsaturated layer):

$$f_{satdeg,soil}(I_{unsat}) = \max(0.01, \min(1.0, \frac{\theta_{soil}(I_{unsat})}{\theta_{soil,max}(I_{unsat})})) \quad \text{(Eq. 3.7.350)}$$

$$\psi_{soil}(I_{unsat}) = -\psi_{soil,sat}(I_{unsat}) \times 1000 \times [f_{satdeg,soil}(I_{unsat})]^{-B_{exp}} \quad \text{(Eq. 3.7.351)}$$

$$\psi_{soil}(I_{unsat}) = \max(-120000, C_{micropore} \times \psi_{soil}(I_{unsat})) \quad \text{(Eq. 3.7.352)}$$

where $f_{satdeg,soil}$ is the soil saturation degree for a certain soil layer, ψ_{soil} [mm] is the soil matric potential, $\psi_{soil,sat}$ [m] is the matric potential of saturated soil (set in NoahmpTable.TBL), B_{exp} is the soil exponential *B* parameter (set in NoahmpTable.TBL), $C_{micropore}$ is the soil micropore content, and I_{unsat} is the layer index of first unsaturated layer right above water table.

(5) Compute recharge (Q_{rech}, [mm/s]) to groundwater based on Darcy's Law:

$$Q_{rech} = -K_{hyd,aquifer}(I_{unsat}) \times \frac{-d_{wt} \times 10^3 - (\psi_{soil}(I_{unsat}) - Z_{node}(I_{unsat}) \times 10^3)}{(d_{wt} - Z_{node}(I_{unsat})) \times 10^3} \quad \text{(Eq. 3.7.353)}$$

$$K_{hyd,aquifer}(I_{unsat}) = \frac{2 \times [K_{hyd,soil}(I_{unsat}) \times K_{hyd,soil,sat}(I_{unsat})]}{K_{hyd,soil}(I_{unsat}) + K_{hyd,soil,sat}(I_{unsat})} \quad \text{(Eq. 3.7.354)}$$

$$Q_{rech} = \max(\frac{-10}{\Delta t_{soil}}, \min\left(\frac{10}{\Delta t_{soil}}, Q_{rech}\right)) \quad \text{(Eq. 3.7.355)}$$

where $K_{hyd,aquifer}$ [mm/s] is the aquifer hydraulic conductivity, $K_{hyd,soil}$ [mm/s] is the soil water conductivity, $K_{hyd,soil,sat}$ [mm/s] is the saturated soil water conductivity (set in

NoahmpTable.TBL), d_{wt} [m] is the water table depth, and Z_{node} [m] is the node (middle point) depth of each soil layer.

(6) Compute water storage in the aquifer and saturated soil:

The total water storage in the aquifer + saturated soil ($W_{aqu+soil}$, [mm]) is first updated as:

$$W_{aqu+soil} = W_{aqu+soil} + (Q_{rech} - Q_{dis}) \times \Delta t_{soil} \quad \text{(Eq. 3.7.356)}$$

Then both the total water storage and the water storage in the aquifer ($W_{aquifer}$, [mm]) are updated based on the location of water table depth as follows:

If $I_{unsat} = N_{soil}$ (i.e., water table is below the bottom of entire soil column and none of the soil layers is saturated), then

$$W_{aquifer} = W_{aquifer} + (Q_{rech} - Q_{dis}) \times \Delta t_{soil} \quad \text{(Eq. 3.7.357)}$$

$$W_{aqu+soil} = W_{aquifer} \quad \text{(Eq. 3.7.358)}$$

$$d_{wt} = (-Z_{soil}(N_{soil}) + 25) - \frac{W_{aquifer}}{Y_{gw} \times 10^3} \quad \text{(Eq. 3.7.359)}$$

where Y_{gw} is the specific groundwater yield (= 0.2 set in NoahmpTable.TBL). Then the soil water amount ($W_{soil,liq}$, [mm]) in the last soil layer is updated as:

$$W_{soil,liq}(N_{soil}) = W_{soil,liq}(N_{soil}) - Q_{rech} \times \Delta t_{soil} \quad \text{(Eq. 3.7.360)}$$

$$W_{soil,liq}(N_{soil}) = W_{soil,liq}(N_{soil}) + \max(0, W_{aquifer} - 5000) \quad \text{(Eq. 3.7.361)}$$

$$W_{aquifer} = \min(W_{aquifer}, 5000) \quad \text{(Eq. 3.7.362)}$$

Here, a maximum aquifer water amount of 5000 mm is set (Niu et al., 2007).

If $I_{unsat} = N_{soil} - 1$ (i.e., water table is within the bottom/last soil layer), then

$$d_{wt} = -Z_{soil}(N_{soil}) - \frac{W_{aqu+soil} - Y_{gw} \times 10^3 \times 25}{P_{soil,eff}(N_{soil}) \times 10^3} \quad \text{(Eq. 3.7.363)}$$

where $P_{soil,eff}$ [m^3/m^3] is the effective soil porosity.

If $I_{unsat} < N_{soil} - 1$ (i.e., water table is above the bottom/last soil layer), then

$$d_{wt} = -Z_{soil}(I_{unsat} + 1) - \frac{W_{aqu+soil} - Y_{gw} \times 10^3 \times 25 - \sum_{i=I_{unsat}+2}^{N_{soil}} [P_{soil,eff}(i) \times dz_{soil}(i)]}{P_{soil,eff}(I_{unsat}+1) \times 10^3} \quad \text{(Eq. 3.7.364)}$$

Then, for $I_{unsat} < N_{soil}$ (i.e., water table is above the bottom soil layer), the soil water amount ($W_{soil,liq}$, [mm]) is further updated by removing subsurface runoff:

$$W_{soil,liq}(i) = W_{soil,liq}(i) - \frac{Q_{dis} \times \Delta t_{soil} \times K_{hyd,aquifer}(i) \times dz_{soil}(i)}{\sum_{i=1}^{N_{soil}} [K_{hyd,aquifer}(i) \times dz_{soil}(i)]} \quad \text{(Eq. 3.7.365)}$$

$$d_{wt} = \max(1.5, d_{wt}) \quad \text{(Eq. 3.7.366)}$$

(7) Treatment for too-low soil water:

Noah-MP limits the soil liquid water to above a minimum threshold ($W_{soil,min}$ = 0.01 mm, currently hard-coded). If the soil water is too low, then water from a lower layer will be brought to increase the water to the minimum threshold.

Then, for soil layers from 1 (top) to $N_{soil} - 1$:

If $W_{soil,liq}(i) < 0.0$, then the water is brought to this layer from the lower layer to reach $W_{soil,min}$:

$$W_{soil,liq}(i) = W_{soil,liq}(i) + (W_{soil,min} - W_{soil,liq}(i)) \quad \text{(Eq. 3.7.367)}$$

$$W_{soil,liq}(i+1) = W_{soil,liq}(i+1) - (W_{soil,min} - W_{soil,liq}(i)) \quad \text{(Eq. 3.7.368)}$$

For the last soil layer ($i = N_{soil}$):

If $W_{soil,liq}(N_{soil}) < W_{soil,min}$, then the water is brought to this layer from the aquifer water storage to reach $W_{soil,min}$:

$$W_{soil,liq}(N_{soil}) = W_{soil,liq}(N_{soil}) + (W_{soil,min} - W_{soil,liq}(N_{soil})) \quad \text{(Eq. 3.7.369)}$$

$$W_{aquifer} = W_{aquifer} - (W_{soil,min} - W_{soil,liq}(N_{soil})) \quad \text{(Eq. 3.7.370)}$$

$$W_{aqu+soil} = W_{aqu+soil} - (W_{soil,min} - W_{soil,liq}(N_{soil})) \quad \text{(Eq. 3.7.371)}$$

(8) Update the soil volume liquid water content ($W_{liq,soil}(i)$, [m^3/m^3]):

$$W_{liq,soil}(i) = \frac{W_{soil,liq}(i)}{dz_{soil}(i)} \quad \text{(Eq. 3.7.372)}$$

5. OptRunoffSubsurface = 5 (RunoffSubSurfaceShallowWaterMmfMod.F90**):**

This module is to diagnose water table depth and computes recharge when the water table is shallow (within the Noah-MP soil column layers) based on the MMF groundwater scheme (Fan et al., 2007; Miguez-Macho et al. 2007). The computations for deep water table and lateral groundwater flow are done outside the Noah-MP column model in GroundWaterMmfMod.F90.

Relevant code modules:

Module: ShallowWaterTableMmfMod.F90

Subroutine: ShallowWaterTableMMF

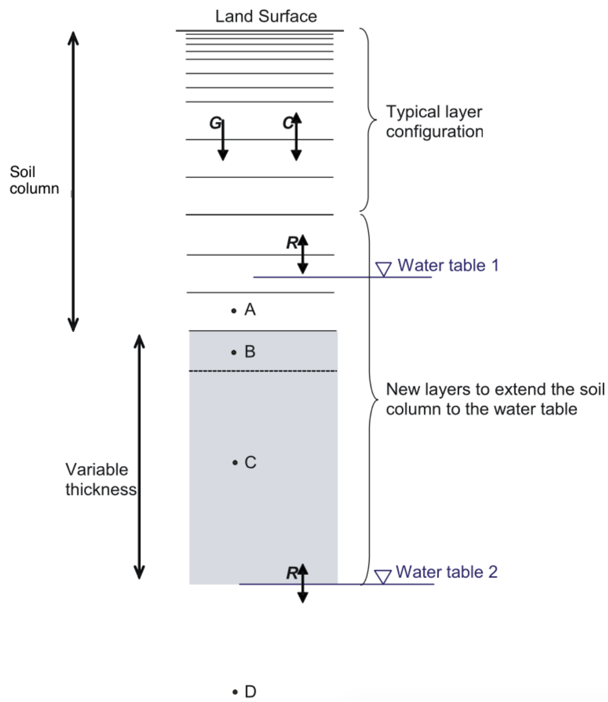


Figure 14. Adapted from Miguez-Macho et al. (2007). Soil water fluxes in LSM soil column (*G*, gravitational drain; *C*, capillary flux; *R*, water table recharge) and the extended soil column to the water table.

General description:

When the initial water table is shallow (within the Noah-MP soil column layers), as shown by "Water Table 1" in the above figure, within this soil layer, water content in the upper unsaturated portion is obtained by assuming that flux between this layer and the layer below is zero (no vertical flow between two saturated layers). This water content is used to compute the layer mean water content, which is needed to compute the water table recharge (leading to water table rise or drop). The recharge will fill or drain the pore space between saturation and the current water content in the unsaturated part of the layer.

When the initial water table is relatively deep (below the Noah-MP soil column bottom), as shown by "Water Table 2" in the above figure, a bottom layer is added that extends the soil bottom to the water table (shaded area in the above figure). The thickness of this layer (centered at point C) is spatiotemporally variable. Since this extended layer can be much thicker than the layer above (causing numerical issue for finite difference schemes), an auxiliary layer is added which contains point B (defined by a dashed line in the above figure). This auxiliary layer has equal thickness to the layer above which contains point A. The water content of point B is obtained by linear interpolation between A and C. Given the water content at A and B, the flux between the two can be calculated. In the same manner, an auxiliary layer is added below the water table, containing point D and having equal thickness to the layer containing C. The gradient between C and D determines the recharge flux between the two.

<u>Specific code processes:</u>

(1) Find the layer where the water table is currently located:

Looping over soil layer from N_{soil} (bottom) to 1 (top), if the water table depth (d_{wt}, [m], negative) meets that $d_{wt} + 10^{-6} < Z_{soil}(iz)$, then $I_{unsat} = iz$, where I_{unsat} is the the layer index of first unsaturated layer right above the water table depth.

Then the layer index for water table (I_{wtd}) is:

$$I_{wtd} = I_{unsat} + 1 \quad \text{(Eq. 3.7.373)}$$

(2) If the current water table is within the model soil layers (i.e., $I_{wtd} \le N_{soil}$):

$$d_{wt,old} = d_{wt} \quad \text{(Eq. 3.7.374)}$$

(2a) For $\theta_{soil}(I_{wtd}) > \theta_{soil,eq}(I_{wtd})$, where $\theta_{soil,eq}$ [m^3/m^3] is the equilibrium soil moisture computed in the MMF groundwater initialization process (drivers/NoahmpGroundwaterInitMod.F90), then:

If $\theta_{soil}(I_{wtd}) = \theta_{soil,max}(I_{wtd})$, the water table has moved to the layer above:

$$d_{wt} = Z_{soil}(I_{unsat}) \quad \text{(Eq. 3.7.375)}$$

$$W_{rech,sha} = -\left(d_{wt,old} - d_{wt}\right) \times [\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})] \quad \text{(Eq. 3.7.376)}$$

$$I_{unsat} = I_{unsat} - 1 \quad \text{(Eq. 3.7.377)}$$

$$I_{wtd} = I_{wtd} - 1 \quad \text{(Eq. 3.7.378)}$$

where $\theta_{soil,max}$ [m^3/m^3] is the maximum (saturated) soil moisture (set in NoahmpTable.TBL), and $W_{rech,sha}$ [m] is the shallow groundwater recharge water amount.

Then if $I_{wtd} \ge 1$ and $\theta_{soil}(I_{wtd}) > \theta_{soil,eq}(I_{wtd})$, i.e.,

$$d_{wt,old} = d_{wt} \quad \text{(Eq. 3.7.379)}$$

$$d_{wt} = \frac{\theta_{soil}(I_{wtd}) \times D_{zsnso}(I_{wtd}) - \theta_{soil,eq}(I_{wtd}) \times Z_{soil}(I_{unsat}) + \theta_{soil,max}(I_{wtd}) \times Z_{soil}(I_{wtd})}{\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})}$$

(Eq. 3.7.380)

$$d_{wt} = \min(Z_{soil}(I_{unsat}), d_{wt}) \quad \text{(Eq. 3.7.381)}$$

$$W_{rech,sha} = W_{rech,sha} - (d_{wt,old} - d_{wt}) \times [\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})] \quad \text{(Eq. 3.7.382)}$$

Note that $Z_{soil}(0) = 0$ in this case.

If $\theta_{soil}(I_{wtd}) < \theta_{soil,max}(I_{wtd})$, the water table stays at the current soil layer:

$$d_{wt} = \frac{\theta_{soil}(I_{wtd}) \times D_{zsnso}(I_{wtd}) - \theta_{soil,eq}(I_{wtd}) \times Z_{soil}(I_{unsat}) + \theta_{soil,max}(I_{wtd}) \times Z_{soil}(I_{wtd})}{\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})}$$

(Eq. 3.7.383)

$$d_{wt} = \min(Z_{soil}(I_{unsat}), d_{wt}) \quad \text{(Eq. 3.7.384)}$$

$$W_{rech,sha} = -(d_{wt,old} - d_{wt}) \times [\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})] \quad \text{(Eq. 3.7.385)}$$

(2b) For $\theta_{soil}(I_{wtd}) \le \theta_{soil,eq}(I_{wtd})$, the water table has moved to the layer below:

$$d_{wt} = Z_{soil}(I_{wtd}) \quad \text{(Eq. 3.7.386)}$$

$$W_{rech,sha} = -(d_{wt,old} - d_{wt}) \times [\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})] \quad \text{(Eq. 3.7.387)}$$

$$I_{unsat} = I_{unsat} + 1 \quad \text{(Eq. 3.7.388)}$$

$$I_{wtd} = I_{wtd} + 1 \quad \text{(Eq. 3.7.389)}$$

If the updated water table is still within the model soil layers (i.e., $I_{wtd} \le N_{soil}$):

$$d_{wt,old} = d_{wt} \quad \text{(Eq. 3.7.390)}$$

For $\theta_{soil}(I_{wtd}) > \theta_{soil,eq}(I_{wtd})$,

$$d_{wt} = \frac{\theta_{soil}(I_{wtd}) \times D_{zsnso}(I_{wtd}) - \theta_{soil,eq}(I_{wtd}) \times Z_{soil}(I_{unsat}) + \theta_{soil,max}(I_{wtd}) \times Z_{soil}(I_{wtd})}{\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})}$$

(Eq. 3.7.391)

$$d_{wt} = \min(Z_{soil}(I_{unsat}), d_{wt}) \quad \text{(Eq. 3.7.392)}$$

For $\theta_{soil}(I_{wtd}) \le \theta_{soil,eq}(I_{wtd})$,

$$d_{wt} = Z_{soil}(I_{wtd}) \quad \text{(Eq. 3.7.393)}$$

Then for both cases,

$$W_{rech,sha} = W_{rech,sha} - (d_{wt,old} - d_{wt}) \times [\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})] \quad \text{(Eq. 3.7.394)}$$

If the updated water table is below the model soil layers (i.e., $I_{wtd} > N_{soil}$):

$$d_{wt,old} = d_{wt} \quad \text{(Eq. 3.7.395)}$$

The equilibrium soil moisture in a hypothetical auxiliary layer below soil bottom is computed as:

$$\theta_{soil,eq,deep} = \theta_{soil,max}(N_{soil}) \times \left[\frac{-\psi_{soil,sat}(N_{soil})}{-\psi_{soil,sat}(N_{soil}) - D_{zsnso}(N_{soil})}\right]^{\frac{1}{B_{exp}(N_{soil})}} \quad \text{(Eq. 3.7.396)}$$

where $\psi_{soil,sat}$ [m] is the saturated soil matric potential (set in NoahmpTable.TBL) and B_{exp} is the soil B exponential parameter (set in NoahmpTable.TBL).

Then, water table depth is updated:

$$d_{wt} = \frac{\theta_{ToWT} \times D_{zsnso}(N_{soil}) - \theta_{soil,eq,deep} \times Z_{soil}(N_{soil}) + \theta_{soil,max}(N_{soil}) \times (Z_{soil}(N_{soil}) - D_{zsnso}(N_{soil}))}{\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}}$$

(Eq. 3.7.397)

$$d_{wt} = \min(Z_{soil}(N_{soil}), d_{wt}) \quad \text{(Eq. 3.7.398)}$$

where θ_{ToWT} [m^3/m^3] is the soil moisture between soil column bottom and the water table computed in the soil water diffusion part above. Then the recharge is updated:

$$W_{rech,sha} = W_{rech,sha} - (d_{wt,old} - d_{wt}) \times [\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}] \quad \text{(Eq. 3.7.399)}$$

(3) If the current water table is already below the model soil layers, but not by too much, in a hypothetical auxiliary layer below the soil bottom (i.e., $I_{wtd} > N_{soil}$ but $d_{wt} \geq Z_{soil}(N_{soil}) - D_{zsnso}(N_{soil})$):

$$d_{wt,old} = d_{wt} \quad \text{(Eq. 3.7.400)}$$

$$\theta_{soil,eq,deep} = \theta_{soil,max}(N_{soil}) \times \left[\frac{-\psi_{soil,sat}(N_{soil})}{-\psi_{soil,sat}(N_{soil}) - D_{zsnso}(N_{soil})}\right]^{\frac{1}{B_{exp}(N_{soil})}} \quad \text{(Eq. 3.7.401)}$$

(3a) If $\theta_{ToWT} > \theta_{soil,eq,deep}$, i.e., water table is still in this hypothetical auxiliary layer below the soil bottom, then

$$d_{wt} = \frac{\theta_{ToWT} \times D_{zsnso}(N_{soil}) - \theta_{soil,eq,deep} \times Z_{soil}(N_{soil}) + \theta_{soil,max}(N_{soil}) \times (Z_{soil}(N_{soil}) - D_{zsnso}(N_{soil}))}{\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}}$$

(Eq. 3.7.402)

$$d_{wt} = \min(Z_{soil}(N_{soil}), d_{wt}) \quad \text{(Eq. 3.7.403)}$$

$$W_{rech,sha} = -(d_{wt,old} - d_{wt}) \times [\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}] \quad \text{(Eq. 3.7.404)}$$

(3b) If $\theta_{ToWT} \leq \theta_{soil,eq,deep}$, i.e., the water table is at the lower border of this hypothetical auxiliary layer, then

$$W_{rech,sha} = -(d_{wt,old} - (Z_{soil}(N_{soil}) - D_{zsnso}(N_{soil}))) \times [\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}]$$

(Eq. 3.7.405)

$$d_{wt,old} = Z_{soil}(N_{soil}) - D_{zsnso}(N_{soil}) \quad \text{(Eq. 3.7.406)}$$

$$d_{wt} = d_{wt,old} - \frac{(\theta_{soil,eq,deep} - \theta_{ToWT}) \times D_{zsnso}(N_{soil})}{\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}} \quad \text{(Eq. 3.7.407)}$$

$$W_{rech,sha} = W_{rech,sha} - \frac{(\theta_{soil,eq,deep} - \theta_{ToWT}) \times D_{zsnso}(N_{soil})}{\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}} \times [\theta_{soil,max}(N_{soil}) - \theta_{soil,eq,deep}]$$

(Eq. 3.7.408)

$$\theta_{ToWT} = \theta_{soil,eq,deep} \quad \text{(Eq. 3.7.409)}$$

(4) Further adjust θ_{ToWT}:

$$\theta_{ToWT} = \begin{cases} \theta_{soil,max}(I_{unsat}) & if \;\; 0 < I_{unsat} < N_{soil} \\ \theta_{soil,max}(1) & if \;\; I_{unsat} \leq 0 \end{cases} \quad \text{(Eq. 3.7.410)}$$

(5) Update key water state and subsurface runoff:

$$W_{liq,soil} = \theta_{soil} - W_{ice,soil} \quad \text{(Eq. 3.7.411)}$$

$$R_{sub} = R_{sub} + Q_{soil,drain} \quad \text{(Eq. 3.7.412)}$$

$$W_{aquifer} = 0 \quad \text{(Eq. 3.7.413)}$$

where $Q_{soil,drain}$ [mm/s] is the soil water drainage computed in earlier sections. Note that R_{sub} has no physical meaning and should not be used for OptRunoffSubsurface = 5. The subsurface runoff for OptRunoffSubsurface = 5 uses another variable "QRF" in GroundWaterMmfMod.F90. The total water balance analysis for Noah-MP soil column plus MMF groundwater column will be presented below in the "MMF groundwater with lateral flow" section.

6. Update subsurface runoff for all options with glacier excess water flow:

$$R_{sub} = R_{sub} + Q_{sno,gla} \quad \text{(Eq. 3.7.414)}$$

where $Q_{sno,gla}$ [mm/s] is the excess glacier water flow computed from snowpack hydrology parts.

3.7.9 MMF Groundwater with Lateral Flow

Description:

This module is to compute lateral groundwater flow and the flux between groundwater and rivers as well as to update soil moisture and water table due to those fluxes based on the Miguez-Macho and Fan (MMF) groundwater scheme (Fan et al., 2007; Miguez-Macho et al. 2007). The implementation of the MMF groundwater scheme into Noah-MP is done by Barlage et al. (2015). This model is a 2-D (lateral flow only) and steady-state (equilibrium) groundwater module to estimate equilibrium water table. Note that this scheme is only activated when OptRunoffSubsurface = 5 and is outside the Noah-MP column model source code due to the treatment of lateral flow. This module is typically called in the parent/host model driver level instead of Noah-MP column model driver level. Thus, only brief descriptions of this MMF groundwater scheme are presented in this technical note. Interested readers are referred to those two original references for details. The refactor of the main MMF module (GroundWaterMmfMod.F90) is not conducted since it is not part of the core Noah-MP column model code, which will be completed in the future.

Relevant code modules:

Module: GroundWaterMmfMod.F90

Subroutine: WTABLE_mmf_noahmp, LateralFlow, UpdateWTD

Physics:

The general workflow for this model is to (1) compute groundwater lateral flow, (2) compute discharge flux (subsurface runoff) from groundwater to rivers in the same grid cell, (3) compute groundwater recharge for the deep water table (note that the shallow water table situation is handled within Noah-MP column model in ShallowWaterTableMmfMod.F90), (4) compute total water flux to or from groundwater in the grid cell, and (5) update water table depth and soil moisture.

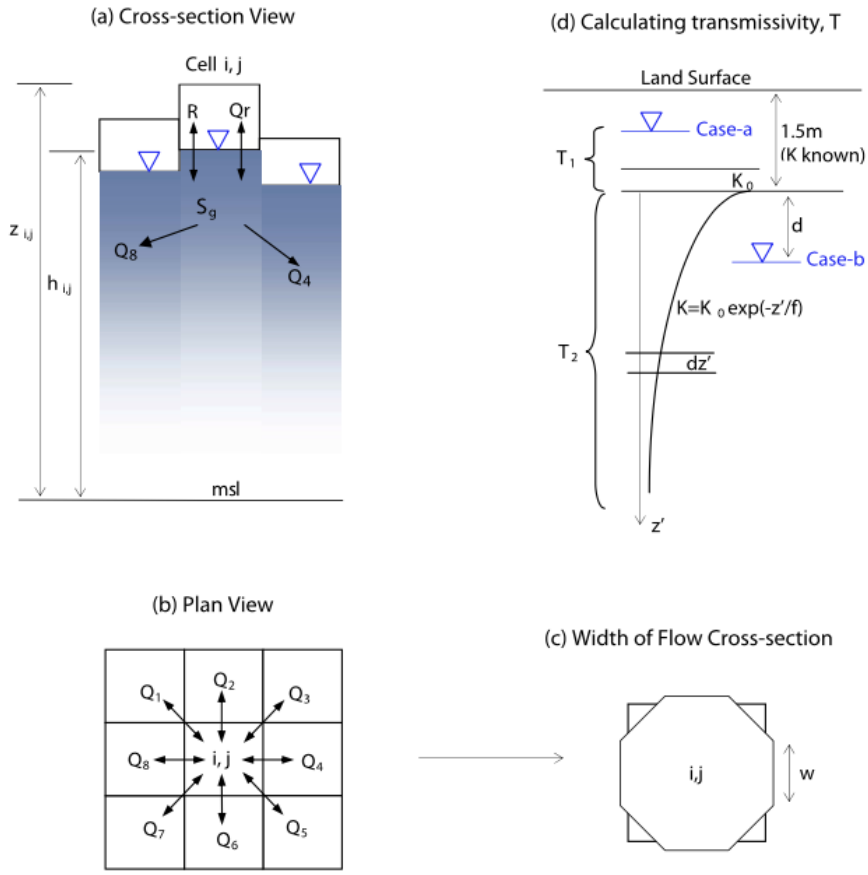


Figure 15. Adapted from Fan et al. (2007). (a) Groundwater store, S_g, and associated fluxes, cross-section view; (b) plan view of lateral groundwater flow to neighboring cells, Q_n (n = 1. . .8); (c) replacing the square grid cells with octagons to calculate the width (w) of flow cross section between two cells; and (d) calculating flow transmissivity (T) assuming exponential decay in hydraulic conductivity (K).

1. Groundwater mass balance:

As shown in the above figure, for a specific model grid box of size $\Delta x \times \Delta y$ [m^2], the groundwater mass balance is expressed as:

$$\frac{\Delta S_g}{\Delta t_{gw}} = \Delta x \Delta y R + \sum_{n=1}^{8} Q_n - Q_r \qquad \text{(Eq. 3.7.415)}$$

where S_g [m^3] is the groundwater storage, Δt_{gw} [s] is the groundwater timestep, R [m/s] is the net recharge flux between unsaturated soil and the groundwater, Q_n [m^3/s] is lateral flow to/from the neighboring grids, and Q_r [m^3/s] is the groundwater-river exchange.

For equilibrium groundwater state,

$$\frac{\Delta S_g}{\Delta t_{gw}} = 0 \quad \text{(Eq. 3.7.416)}$$

then

$$\Delta x \Delta y R = -\sum_{n=1}^{8} Q_n + Q_r \quad \text{(Eq. 3.7.417)}$$

Thus, the groundwater recharge balances the lateral flow to surrounding grids and discharge to the rivers within the grid.

2. Lateral flow (LateralFlow subroutine**):**

The lateral flow (Q_n, [m^3/s]) is computed using Darcy's Low as follows:

$$Q_n = w \times T \times \frac{h_n - h}{l} \quad \text{(Eq. 3.7.418)}$$

where Q_n is positive for flow entering the cell, w [m] is the width of flow cross section, T [m^2/s] is the flow transmissivity, h_n [m] is the water table head in the n^{th} neighbor, h [m] is the water table head in the center cell, and l is the distance between neighboring grids ($l = \Delta x$ for x-/y-direction, and $l = \sqrt{2}\Delta x$ for diagonal direction). As shown in the above figure, currently it assumes equal chance of flow in all 8 directions by replacing the square cell ($\Delta x = \Delta y$) with octagons of same surface area, and hence it gives:

$$w = \Delta x \sqrt{\frac{1}{2}\tan(\frac{\pi}{8})} \quad \text{(Eq. 3.7.419)}$$

To compute flow transmissivity, there are two cases as shown in the above figure panel (d).

(1) When the water table is above 1.5 m depth, the transmissivity is:

$$T = T_1 + T_2 \quad \text{(Eq. 3.7.420)}$$

$$T_1 = \sum K_{hyd}(i) \times \Delta z(i) = K_{hyd,0} \times (d_{wt} - (-1.5)) \quad \text{(Eq. 3.7.421)}$$

$$T_2 = \int_0^\infty K_{hyd} dz' = \int_0^\infty K_{hyd,0} \times e^{-\frac{z'}{f}} dz' = K_{hyd,0} \times f \quad \text{(Eq. 3.7.422)}$$

where d_{wt} [m] is the water table depth (negative). K_{hyd} [m/s] is the soil hydraulic conductivity, which is assumed to be the same ($K_{hyd,0}$) for layers between the water table and the 1.5 m depth, and decay exponentially with depth below 1.5 m depth as follows:

$$K_{hyd} = K_{hyd,0} \times e^{-\frac{z'}{f}} \quad \text{(Eq. 3.7.423)}$$

$K_{hyd,0}$ [m/s] is the hydraulic conductivity in the bottom soil layer of the 1.5 m column, z' is the depth below 1.5 m, and f is the e-folding length. Thus,

$$T = K_{hyd,0} \times (d_{wt} + 1.5 + f) \quad \text{(Eq. 3.7.424)}$$

(2) When the water table is below the 1.5 m depth by a distance of d [m], the transmissivity is:

$$T_2 = \int_d^\infty K_{hyd} dz' = \int_d^\infty K_{hyd,0} \times e^{-\frac{z'}{f}} dz' = K_{hyd,0} \times f \times e^{-\frac{z-h-1.5}{f}} \quad \text{(Eq. 3.7.425)}$$

where z [m] is the topographic elevation, $d = -1.5 - d_{wt} = z - h - 1.5$ and $h = d_{wt} + z$. Note that the water table head (h) and depth (d_{wt}) are all negative values.

To ensure the flow from grid A to B is the same as from B to A under the same hydraulic potential, the final transmissivity for both grids is the average of the transmissivity values for these two grids. The hydraulic conductivity is computed based on the saturated values:

$$K_{hyd,0} = K_{hyd,sat} \times f_{k,cor} \quad \text{(Eq. 3.7.426)}$$

where $K_{hyd,sat}$ [m/s] is the saturated soil hydraulic conductivity (set in NoahmpTable.TBL) and $f_{k,cor}$ is a correction/scaling factor (hard-coded currently). The e-folding length (f) is read from the input static dataset.

3. River conductance and groundwater discharge (WTABLE_mmf_noahmp subroutine**):**

The groundwater-river exchange flux depends on three factors: the elevation difference between the water table and the river stage, the hydraulic connection between the two reservoirs (river bed thickness and permeability), and the contact area (length times width of the river segment, neglecting depth).

According to Darcy's Law, the groundwater flux to river (Q_r, [m^3/s]) is expressed as:

$$Q_r = (d_{wt} - z_r) \times \frac{K_{hyd,rb}}{d_{z,rb}} \times (w_r \times L_r) \quad for\ (d_{wt} - z_r) > 0 \quad \text{(Eq. 3.7.427)}$$

where d_{wt} [m] is the water table depth, z_r [m] is the river bed depth from surface instead of sea level (note that the raw riverbed data read from input static data is the depth from sea level instead of surface and an adjustment of topography is conducted within the code), $K_{hyd,rb}$ [m/s] is the river bed hydraulic conductivity, $d_{z,rb}$ [m] is the thickness of river bed sediments, w_r [m] is the river width, and L_r [m] is the total river length. Further defining a parameter called river hydraulic conductance or river conductance (R_C, [m^2/s]) as follows:

$$R_C = \frac{K_{hyd,rb}}{d_{z,rb}} \times (w_r \times L_r) \quad \text{(Eq. 3.7.428)}$$

$$Q_r = (d_{wt} - z_r) \times R_C \quad \text{(Eq. 3.7.429)}$$

Note that because R_C is difficult to obtain from observations depending on river geometry and bed sediments, a parameterization of R_C is used as described below.

For river flux going to groundwater ($(d_{wt} - z_r) \leq 0$), the flux is expressed as:

$$Q_r = K_{hyd,rb} \times (w_r \times L_r) \quad for\ (d_{wt} - z_r) < 0 \qquad \text{(Eq. 3.7.430)}$$

This means that leakage from river to groundwater is at a constant rate. Note that the MMF scheme in Noah-MP currently does not allow water flux to go from river to groundwater, which needs more work in the future.

The river conductance (R_C) is parameterized as a product of an equilibrium part and a dynamic part. The equilibrium river conductance ($R_{C,eq}$, [m^2/s]) is linked to the equilibrium water table and represents the equilibrium capacity of river in meeting the land drainage demand. At equilibrium, the $R_{C,eq}$ is expressed as:

$$R_{C,eq} = \frac{\Delta x \Delta y R + \sum_{n=1}^{8} Q_n}{d_{wt,eq} - z_r} \qquad \text{(Eq. 3.7.431)}$$

The equilibrium water table ($d_{wt,eq}$) is derived by solving the following equation iteratively, updating water table depth and soil moisture:

$$\Delta x \Delta y R = -\sum_{n=1}^{8} Q_n \qquad \text{(Eq. 3.7.432)}$$

This is because if model grid cells are fine enough so that a cell is either a hillslope or a river cell, but not both, then Q_r may be dropped for a hillslope cell.

Then, define a dynamic river conductance factor ($F_{RC,dyn}$) as an exponential function of the deviation of water table from equilibrium as:

$$F_{RC,dyn} = e^{a \times (d_{wt} - d_{wt,eq})} \qquad \text{(Eq. 3.7.433)}$$

where the parameter a (=1.0, currently hard-coded) determines how fast R_C responds to deviations from equilibrium, which is function of local terrain.

Thus, the total river conductance (R_C) is

$$R_C = \begin{cases} R_{C,eq} \times F_{RC,dyn} & if\ d_{wt} > z_r\ and\ d_{wt,eq} > z_r \\ R_{C,eq} & otherwise \end{cases} \qquad \text{(Eq. 3.7.434)}$$

4. Recharge for deep water table (WTABLE_mmf_noahmp subroutine**):**

Note that the situation for shallow water table is handled within the Noah-MP column model in ShallowWaterTableMmfMod.F90 shown above. This module here is to handle deep water table ($d_{wt} < Z_{soil}(N_{soil}) - D_{Z,soil}(N_{soil})$) recharge.

The soil matric potential (ψ_{soil}, [m]) is computed as

$$\psi_{soil} = -\psi_{soil,sat} \times \left[\frac{\theta_{ToWT}}{\theta_{soil,max}}\right]^{-B_{exp}} \quad \text{(Eq. 3.7.435)}$$

where $\psi_{soil,sat}$ [m] is the saturated soil matric potential (set in NoahmpTable.TBL), θ_{ToWT} [m^3/m^3] is the soil moisture between soil column bottom and the water table, $\theta_{soil,max}$ [m^3/m^3] is the saturated soil moisture (set in NoahmpTable.TBL), and B_{exp} is the soil B exponential parameter (set in NoahmpTable.TBL).

The hydraulic conductivity for deep soil ($K_{hyd,deep}$) is computed as

$$K_{hyd,deep} = K_{hyd,sat} \times \left[\frac{\theta_{ToWT} + \theta_{soil,max}}{2 \times \theta_{soil,max}}\right]^{2 \times B_{exp} + 3} \quad \text{(Eq. 3.7.436)}$$

where $K_{hyd,sat}$ [m/s] is the saturated soil hydraulic conductivity (set in NoahmpTable.TBL).

Then, the amount of water across the water table interface ($W_{deep,wt}$, [m]) is computed as

$$W_{deep,wt} = -K_{hyd,deep} \times \Delta t_{gw} \times \left[\frac{-\psi_{soil,sat} - \psi_{soil}}{Z_{soil}(N_{soil}) - d_{wt}} - 1\right] \quad \text{(Eq. 3.7.437)}$$

Then, θ_{ToWT} is further updated based on the water flux across water table:

$$\theta_{ToWT} = \theta_{ToWT} + \frac{W_{rech,deep} - W_{deep,wt}}{Z_{soil}(N_{soil}) - d_{wt}} \quad \text{(Eq. 3.7.438)}$$

where $W_{rech,deep}$ [m] is the temporary recharge for deep water table computed from Noah-MP column soil processes.

The water excess (W_{excess}, [m]) and deficit ($W_{deficit}$, [m]) are then computed as:

$$W_{excess} = \max(0, \theta_{ToWT} - \theta_{soil,max}) \times (Z_{soil}(N_{soil}) - d_{wt}) \quad \text{(Eq. 3.7.439)}$$

$$W_{deficit} = \max(0, 10^{-4} - \theta_{ToWT}) \times (Z_{soil}(N_{soil}) - d_{wt}) \quad \text{(Eq. 3.7.440)}$$

The θ_{ToWT} and $W_{deep,wt}$ are further adjusted as

$$\theta_{ToWT} = \max(10^{-4}, \min(\theta_{ToWT}, \theta_{soil,max})) \quad \text{(Eq. 3.7.441)}$$

$$W_{deep,wt} = W_{deep,wt} + W_{excess} - W_{deficit} \quad \text{(Eq. 3.7.442)}$$

Then, the recharge for deep water table is updated as:

$$W_{rech,deep} = W_{deep,wt} \quad \text{(Eq. 3.7.443)}$$

5. Total flux from/to groundwater (WTABLE_mmf_noahmp subroutine**):**

Based on the groundwater mass balance equation above, the total net water amount from/to groundwater ($W_{gw,net}$, [m]) is computed as:

$$W_{gw,net} = \left(\sum_{n=1}^{8} Q_n - Q_r\right) \times \frac{\Delta t_{gw}}{\Delta x \times \Delta y} + W_{rech,deep} \quad \text{(Eq. 3.7.444)}$$

6. Update water table and soil moisture (UpdateWTD subroutine**):**

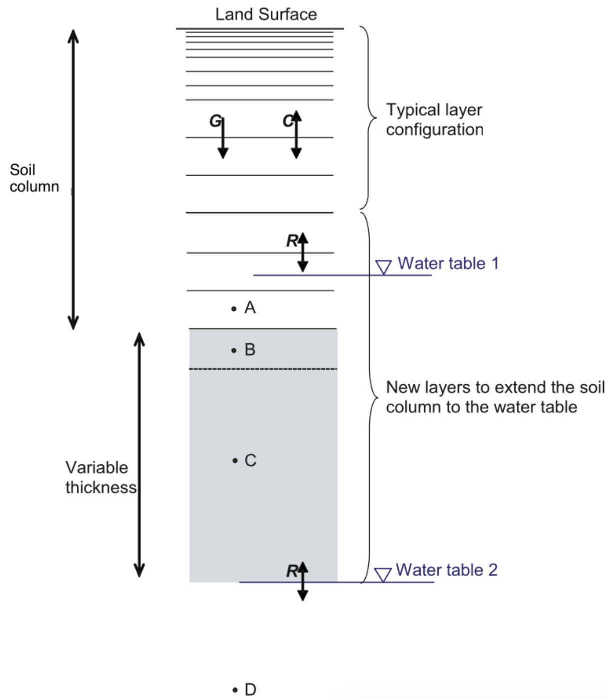


Figure 16. Adapted from Miguez-Macho et al. (2007). Soil water fluxes in LSM soil column (*G*, gravitational drain; *C*, capillary flux; *R*, water table recharge) and the extended soil column to the water table.

When the initial water table is shallow (within the Noah-MP soil column layers), please see the descriptions in the "RunoffSubSurfaceShallowWaterMmfMod.F90" section.

When the initial water table is deep (below the Noah-MP soil column bottom), as shown by "Water Table 2" in the above figure, a bottom layer is added that extends the soil bottom to the water table (shaded area in the above figure). The thickness of this layer (centered at point C) is spatiotemporally variable. Since this extended layer can be much thicker than the layer above (causing numerical issue for finite difference schemes), an auxiliary layer is added which contains point B (defined by a dashed line in the above figure). This auxiliary layer has equal thickness as the layer above which contains point A. The water content of point B is obtained by linear interpolation between A and C. Given the water content at A and B, the flux between the two can be calculated. In the same manner, an auxiliary layer is added below the water table, containing point D and with equal thickness as the layer containing C. The gradient between C and D determines the recharge flux between the two.

(6a) When water table goes up ($W_{gw,net} > 0$):

(6a-1) If the water table (d_{wt}, [m], negative) is already in the Noah-MP soil column layers ($d_{wt} \geq Z_{soil}(N_{soil})$), then first identify the layer index where the water table is located:

Looping over each soil layer from N_{soil}-1 to 1, if $d_{wt} < Z_{soil}(iz)$, then $I_{unsat} = iz$, where I_{unsat} is the the layer index of first unsaturated layer right above the water table depth. Then the layer index for water table (I_{wtd}) is:

$$I_{wtd} = I_{unsat} + 1 \qquad \text{(Eq. 3.7.445)}$$

Then the maximum amount of water that fills the water table soil layer is:

$$W_{max,wtd} = d_{Z,soil}(I_{wtd}) \times (\theta_{soil,max} - \theta_{soil}(I_{wtd})) \qquad \text{(Eq. 3.7.446)}$$

If $W_{gw,net} \leq W_{max,wtd}$ (not enough water to saturate the layer), then

$$\theta_{soil}(I_{wtd}) = \theta_{soil}(I_{wtd}) + \frac{W_{gw,net}}{d_{Z,soil}(I_{wtd})} \qquad \text{(Eq. 3.7.447)}$$

$$\theta_{soil}(I_{wtd}) = \min(\theta_{soil}(I_{wtd}), \theta_{soil,max}) \qquad \text{(Eq. 3.7.448)}$$

$$W_{gw,net} = 0 \qquad \text{(Eq. 3.7.449)}$$

If $\theta_{soil}(I_{wtd}) > \theta_{soil,eq}(I_{wtd})$, where $\theta_{soil,eq}$ [m^3/m^3] is the equilibrium soil moisture computed in the MMF groundwater initialization process (drivers/NoahmpGroundwaterInitMod.F90), then the water table is updated as

$$d_{wt} = \frac{\theta_{soil}(I_{wtd}) \times d_{Z,soil}(I_{wtd}) - \theta_{soil,eq}(I_{wtd}) \times Z_{soil}(I_{unsat}) + \theta_{soil,max} \times Z_{soil}(I_{wtd})}{\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})}$$

(Eq. 3.7.450)

$$d_{wt} = \min(Z_{soil}(I_{unsat}), d_{wt}) \qquad \text{(Eq. 3.7.451)}$$

If $W_{gw,net} > W_{max,wtd}$ (enough water to saturate the layer), then

$$\theta_{soil}(I_{wtd}) = \theta_{soil,max} \qquad \text{(Eq. 3.7.452)}$$

$$W_{gw,net} = W_{gw,net} - W_{max,wtd} \qquad \text{(Eq. 3.7.453)}$$

Then, move the water table up one layer at a time by looping over soil layers from I_{unsat} to the top layer via iterating the above (6a-1) processes of updating soil moisture and water table in each soil layer loop until $W_{gw,net}$ become zero.

(6a-2) If the water table (d_{wt}, [m], negative) is already below the Noah-MP soil column layers but not by too much in a hypothetical auxiliary layer below the soil bottom (i.e., $Z_{soil}(N_{soil}) > d_{wt} \geq Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil})$), then the equilibrium soil moisture for deep water table ($\theta_{soil,eq,deep}$) is computed as:

$$\theta_{soil,eq,deep} = \theta_{soil,max} \times \left[\frac{-\psi_{soil,sat}}{-\psi_{soil,sat} - d_{Z,soil}(N_{soil})}\right]^{\frac{1}{B_{exp}}} \quad \text{(Eq. 3.7.454)}$$

$$\theta_{soil,eq,deep} = \max(\theta_{soil,eq,deep}, 10^{-4}) \quad \text{(Eq. 3.7.455)}$$

Then the maximum amount of water that fills the water table soil layer is:

$$W_{max,wtd} = d_{Z,soil}(N_{soil}) \times (\theta_{soil,max} - \theta_{ToWT}) \quad \text{(Eq. 3.7.456)}$$

Similarly to (6a-1), the soil moisture in this auxiliary layer (θ_{ToWT}) is updated as follows.

If $W_{gw,net} \leq W_{max,wtd}$ (not water enough to saturate the layer), then

$$\theta_{ToWT} = \theta_{ToWT} + \frac{W_{gw,net}}{d_{Z,soil}(N_{soil})} \quad \text{(Eq. 3.7.457)}$$

$$\theta_{ToWT} = \min(\theta_{ToWT}, \theta_{soil,max}) \quad \text{(Eq. 3.7.458)}$$

$$W_{gw,net} = 0 \quad \text{(Eq. 3.7.459)}$$

If $\theta_{ToWT} > \theta_{soil,eq,deep}$, then the water table is updated as

$$d_{wt} = \frac{\theta_{ToWT} \times d_{Z,soil}(N_{soil}) - \theta_{soil,eq,deep} \times Z_{soil}(N_{soil}) + \theta_{soil,max} \times (Z_{soil}(N_{doil}) - d_{Z,soil}(N_{soil}))}{\theta_{soil,max} - \theta_{soil,eq,deep}}$$

(Eq. 3.7.460)

$$d_{wt} = \min(Z_{soil}(N_{soil}), d_{wt}) \quad \text{(Eq. 3.7.461)}$$

If $W_{gw,net} > W_{max,wtd}$ (enough water to saturate the layer), then

$$\theta_{ToWT} = \theta_{soil,max} \quad \text{(Eq. 3.7.462)}$$

$$W_{gw,net} = W_{gw,net} - W_{max,wtd} \quad \text{(Eq. 3.7.463)}$$

Then, move the water table up one layer at a time by looping over soil layers from N_{soil} to the top layer via iterating the above (6a-1) processes of updating soil moisture and water table in each soil layer loop until $W_{gw,net}$ become zero.

(6a-3) If the water table (d_{wt}, [m], negative) is already very deep (i.e., $d_{wt} < Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil})$), then the maximum amount of water that fills the water table auxiliary soil layer is:

$$W_{max,wtd} = (\theta_{soil,max} - \theta_{ToWT}) \times \left(Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil}) - d_{wt}\right) \quad \text{(Eq. 3.7.464)}$$

If $W_{gw,net} \leq W_{max,wtd}$ (not enough water to saturate the layer), then the water table depth is updated as:

$$d_{wt} = d_{wt} + \frac{W_{gw,net}}{\theta_{soil,max} - \theta_{ToWT}} \quad \text{(Eq. 3.7.465)}$$

$$W_{gw,net} = 0 \quad \text{(Eq. 3.7.466)}$$

If $W_{gw,net} > W_{max,wtd}$ (enough water to saturate the layer), then move water table up by one layer:

$$W_{gw,net} = W_{gw,net} - W_{max,wtd} \quad \text{(Eq. 3.7.467)}$$

$$d_{wt} = Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil}) \quad \text{(Eq. 3.7.468)}$$

$$W_{max,wtd} = d_{Z,soil}(N_{soil}) \times (\theta_{soil,max} - \theta_{ToWT}) \quad \text{(Eq. 3.7.469)}$$

Then, similarly to (6a-2), check the updated $W_{max,wtd}$ and $W_{gw,net}$ and repeat the above (6a-2) processes.

For all the three situations (6a-1, 6a-2, and 6a-3) above, if $W_{gw,net}$ is still larger than 0 after looping over all the Noah-MP soil layers from the bottom to the top layer, then the remaining $W_{gw,net}$ is assigned to the surface water springing (Q_{spring}, [m]):

$$Q_{spring} = W_{gw,net} \quad \text{(Eq. 3.7.470)}$$

(6b) When water table goes down ($W_{gw,net} < 0$):

(6b-1) If the water table (d_{wt}, [m], negative) is already in the Noah-MP soil column layers ($d_{wt} \geq Z_{soil}(N_{soil})$), then first identify the layer index where the water table locates:

Looping over each soil layer from N_{soil}-1 to 1, if $d_{wt} < Z_{soil}(iz)$, then $I_{unsat} = iz$, where I_{unsat} is the the layer index of the first unsaturated layer right above the water table depth. Then the layer index for water table (I_{wtd}) is:

$$I_{wtd} = I_{unsat} + 1 \quad \text{(Eq. 3.7.471)}$$

Then the maximum amount of water that can be yielded by the current water table soil layer is:

$$W_{max,yield} = d_{Z,soil}(I_{wtd}) \times (\theta_{soil}(I_{wtd}) - \max(\theta_{soil,eq}(I_{wtd}), W_{ice,soil}(I_{wtd}))) \quad \text{(Eq. 3.7.472)}$$

If $-W_{gw,net} \leq W_{max,yield}$ (not enough water loss to dry out the layer), then

$$\theta_{soil}(I_{wtd}) = \theta_{soil}(I_{wtd}) + \frac{W_{gw,net}}{d_{Z,soil}(I_{wtd})} \quad \text{(Eq. 3.7.473)}$$

$$W_{gw,net} = 0 \quad \text{(Eq. 3.7.474)}$$

If $\theta_{soil}(I_{wtd}) > \theta_{soil,eq}(I_{wtd})$, then the water table is updated as

$$d_{wt} = \frac{\theta_{soil}(I_{wtd}) \times d_{Z,soil}(I_{wtd}) - \theta_{soil,eq}(I_{wtd}) \times Z_{soil}(I_{unsat}) + \theta_{soil,max} \times Z_{soil}(I_{wtd})}{\theta_{soil,max}(I_{wtd}) - \theta_{soil,eq}(I_{wtd})}$$

(Eq. 3.7.475)

If $\theta_{soil}(I_{wtd}) \leq \theta_{soil,eq}(I_{wtd})$, then the water table is updated as (move one layer down)

$$d_{wt} = Z_{soil}(I_{wtd}) \quad \text{(Eq. 3.7.476)}$$

$$I_{unsat} = I_{unsat} + 1 \quad \text{(Eq. 3.7.477)}$$

If $-W_{gw,net} > W_{max,yield}$ (enough water loss to dry out the layer), then move water table down by one layer:

$$d_{wt} = Z_{soil}(I_{wtd}) \quad \text{(Eq. 3.7.478)}$$

$$I_{unsat} = I_{unsat} + 1 \quad \text{(Eq. 3.7.479)}$$

If $W_{max,yield} \geq 0$, then

$$\theta_{soil}(I_{wtd}) = \theta_{soil}(I_{wtd}) + \frac{W_{max,yield}}{d_{Z,soil}(I_{wtd})} \quad \text{(Eq. 3.7.480)}$$

$$W_{gw,net} = W_{gw,net} + W_{max,yield} \quad \text{(Eq. 3.7.481)}$$

Then, repeat the above (6b-1) processes by looping over soil layer from the initial I_{wtd} to N_{soil} until $W_{gw,net}$ become zero.

If after looping to N_{soil} (i.e., $I_{unsat} = N_{soil}$) and $W_{gw,net}$ is still lower than 0, then further move the water table down to the auxiliary layer as follows:

First, compute the deep soil equilibrium moisture:

$$\theta_{soil,eq,deep} = \theta_{soil,max} \times \left[\frac{-\psi_{soil,sat}}{-\psi_{soil,sat} - d_{Z,soil}(N_{soil})}\right]^{\frac{1}{B_{exp}}} \quad \text{(Eq. 3.7.482)}$$

$$\theta_{soil,eq,deep} = \max(\theta_{soil,eq,deep}, 10^{-4}) \quad \text{(Eq. 3.7.483)}$$

Then the maximum amount of water that can be yielded by the current water table soil layer is:

$$W_{max,yield} = dz_{soil}(N_{soil}) \times (\theta_{ToWT} - \theta_{soil,eq,deep})$$

If $-W_{gw,net} \leq W_{max,yield}$ (not enough water loss to dry out the layer), then

$$\theta_{ToWT} = \theta_{ToWT} + \frac{W_{gw,net}}{d_{Z,soil}(N_{soil})} \quad \text{(Eq. 3.7.484)}$$

$$d_{wt} = \frac{\theta_{ToWT} \times d_{Z,soil}(N_{soil}) - \theta_{soil,eq,deep} \times Z_{soil}(N_{soil}) + \theta_{soil,max} \times (Z_{soil}(N_{doil}) - d_{Z,soil}(N_{soil}))}{\theta_{soil,max} - \theta_{soil,eq,deep}}$$

(Eq. 3.7.485)

$$d_{wt} = \min(Z_{soil}(N_{doil}) - d_{Z,soil}(N_{soil}), d_{wt}) \quad \text{(Eq. 3.7.486)}$$

If $-W_{gw,net} > W_{max,yield}$ (enough water loss to dry out the layer), then move water table down by one layer:

$$d_{wt} = Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil}) \quad \text{(Eq. 3.7.487)}$$

$$\theta_{ToWT} = \theta_{ToWT} + \frac{W_{gw,net}}{d_{Z,soil}(N_{soil})} \quad \text{(Eq. 3.7.488)}$$

$$d_{wt} = d_{wt} - \frac{(\theta_{soil,eq,deep} - \theta_{ToWT}) \times d_{Z,soil}(N_{soil})}{\theta_{soil,max} - \theta_{soil,eq,deep}} \quad \text{(Eq. 3.7.489)}$$

$$\theta_{ToWT} = \theta_{soil,eq,deep} \quad \text{(Eq. 3.7.490)}$$

(6b-2) If the water table (d_{wt}, [m], negative) is already below the Noah-MP soil column layers but not by too much in a hypothetical auxiliary layer below the soil bottom (i.e., $Z_{soil}(N_{soil}) > d_{wt} \geq Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil})$), then the equilibrium soil moisture for deep water table ($\theta_{soil,eq,deep}$) is computed similarly to the above:

$$\theta_{soil,eq,deep} = \theta_{soil,max} \times \left[\frac{-\psi_{soil,sat}}{-\psi_{soil,sat} - d_{Z,soil}(N_{soil})}\right]^{\frac{1}{B_{exp}}} \quad \text{(Eq. 3.7.491)}$$

Then the maximum amount of water that can be yielded by the current water table soil layer is:

$$W_{max,yield} = d_{Z,soil}(N_{soil}) \times (\theta_{ToWT} - \theta_{soil,eq,deep}) \quad \text{(Eq. 3.7.492)}$$

Similarly to (6b-1), the soil moisture in this auxiliary layer (θ_{ToWT}) is updated as follows:

If $-W_{gw,net} \leq W_{max,yield}$ (not enough water loss to dry out the layer), then

$$\theta_{ToWT} = \theta_{ToWT} + \frac{W_{gw,net}}{d_{Z,soil}(N_{soil})} \quad \text{(Eq. 3.7.493)}$$

$$d_{wt} = \frac{\theta_{ToWT} \times d_{Z,soil}(N_{soil}) - \theta_{soil,eq,deep} \times Z_{soil}(N_{soil}) + \theta_{soil,max} \times (Z_{soil}(N_{doil}) - d_{Z,soil}(N_{soil}))}{\theta_{soil,max} - \theta_{soil,eq,deep}}$$

(Eq. 3.7.494)

$$d_{wt} = \min(Z_{soil}(N_{doil}) - d_{Z,soil}(N_{soil}), d_{wt}) \quad \text{(Eq. 3.7.495)}$$

If $-W_{gw,net} > W_{max,yield}$ (enough water loss to dry out the layer), then move water table down by one layer:

$$d_{wt} = Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil}) \quad \text{(Eq. 3.7.496)}$$

$$\theta_{ToWT} = \theta_{ToWT} + \frac{W_{gw,net}}{d_{Z,soil}(N_{soil})} \quad \text{(Eq. 3.7.497)}$$

$$d_{wt} = d_{wt} - \frac{(\theta_{soil,eq,deep} - \theta_{ToWT}) \times d_{Z,soil}(N_{soil})}{\theta_{soil,max} - \theta_{soil,eq,deep}} \quad \text{(Eq. 3.7.498)}$$

$$\theta_{ToWT} = \theta_{soil,eq,deep} \quad \text{(Eq. 3.7.499)}$$

(6b-3) If the water table (d_{wt}, [m], negative) is already very deep (i.e., $d_{wt} < Z_{soil}(N_{soil}) - d_{Z,soil}(N_{soil})$), then the very deep soil equilibrium moisture ($\theta_{soil,eq,deepbot}$) is computed as

$$\theta_{soil,eq,deepbot} = \theta_{soil,max} \times \left[\frac{-\psi_{soil,sat}}{-\psi_{soil,sat} - (Z_{soil}(N_{soil}) - d_{wt})}\right]^{\frac{1}{B_{exp}}} \quad \text{(Eq. 3.7.500)}$$

$$\theta_{soil,eq,deepbot} = \max(\theta_{soil,eq,deepbot}, 10^{-4}) \quad \text{(Eq. 3.7.501)}$$

$$d_{wt,old} = d_{wt} \quad \text{(Eq. 3.7.502)}$$

$$d_{wt} = d_{wt,old} + \frac{W_{gw,net}}{\theta_{soil,max} - \theta_{soil,eq,deepbot}} \quad \text{(Eq. 3.7.503)}$$

$$\theta_{ToWT} = \frac{\theta_{ToWT} \times (Z_{soil}(N_{soil}) - d_{wt,old}) + \theta_{soil,eq,deepbot} \times (d_{wt,old} - d_{wt})}{Z_{soil}(N_{soil}) - d_{wt}} \quad \text{(Eq. 3.7.504)}$$

For all the three situations (6b-1, 6b-2, and 6b-3) above, the surface water springing (Q_{spring}, [m]) is set to 0:

$$Q_{spring} = 0 \quad \text{(Eq. 3.7.505)}$$

7. Total model column water balance (Noah-MP soil column+MMF groundwater column): The MMF groundwater scheme adds an auxiliary layer and an unconfined aquifer below the original 2-m (currently as default) soil layers in Noah-MP and is called in a different timestep in the host model driver module (e.g., HRLDAS and WRF driver). These modifications are beyond the Noah-MP column model water balance check subroutine calculation, hence, complicating the water balance analysis. The column soil structure can now be divided into two parts, the 2-m soil layers as in default Noah-MP and the below 2-m part containing the auxiliary layer and the saturated aquifer.

(1) For the default 2-m soil, the water balance remains the same as in the Noah-MP column model: The fluxes to the top storage include precipitation (P), evapotranspiration (Q_{ET}), surface runoff (R_{sfc}) and subsurface runoff (R_{sub}) as the causes of storage change (ΔS_{2m}):

$$\Delta S_{2m} = P - Q_{ET} - R_{sfc} - R_{sub} \quad \text{(Eq. 3.7.506)}$$

A residual term can be defined for the top 2-m layers:

$$res_{2m} = P - Q_{ET} - R_{sfc} - R_{sub} - \Delta(\theta_{soil} \times 2 + W_{liq,can} + W_{liq,can} + W_{snow}) \quad \text{(Eq. 3.7.507)}$$

This water balance is checked by every soil timestep in Noah-MP column model.

(2) For the water balance below 2-m soils, R_{sub} leaving 2-m soil column becomes the water fluxes across the 2-m soil bottom boundary and the input (recharge) to the auxiliary layer below the soil bottom (positive for downwards and negative for upwards). Other fluxes include lateral fluxes (Q_{lat}; positive for convergence, negative for divergence), discharge to river (Q_r; positive for discharge from aquifer to river network), and surface water springing (Q_{spring}):

$$\Delta S_{gw} = R_{sub} + Q_{lat} - Q_r - Q_{spring} \quad \text{(Eq. 3.7.508)}$$

The storage for below 2-m soil contains two parts, the water in the auxiliary layer and water in aquifer.

The depth of this auxiliary layer is variable with a minimum depth of 1-m if water table is above 3-m, and stretching to the water table (-2-d_{wt}) if the water table is below 3-m. The depth of this auxiliary layer (d_{aux}, [m]) can be represented as:

$$d_{aux} = \begin{cases} 1 & (d_{wt} > -3) \\ -2 - d_{wt} & (d_{wt} < -3) \end{cases} \quad \text{(Eq. 3.7.509)}$$

The figure below shows examples for deep groundwater and shallow groundwater scenarios.

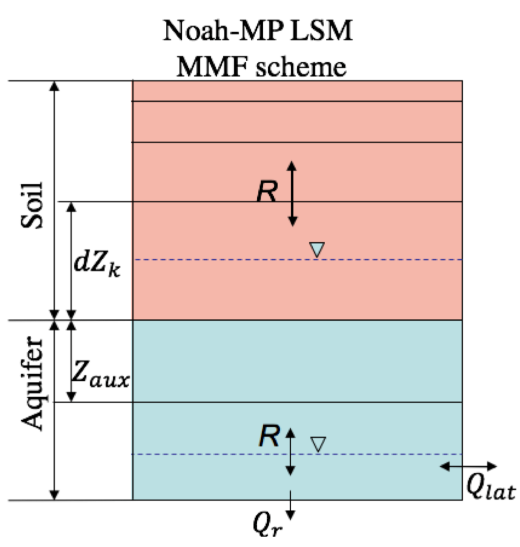


Figure 17. The structure diagram for the Noah-MP with MMF groundwater scheme. The four red layers indicate the 2-m resolved soil layers in Noah-MP. The two blue layers indicate the auxiliary

layer below the 2-m boundary and from 3-m to riverbed. Two water tables are shown with two blue dashed line, indicating two scenarios, deep and shallow groundwater (Zhang et al., 2020).

Scenario 1 – Deep water table (d_{wt} < -3):

For the deep groundwater scenario (d_{wt} <-3), the below 2-m storage contains the stretching auxiliary layer and the water in aquifer. The storage in the auxiliary layer is:

$$S_{aux} = \theta_{ToWT} \times d_{aux}, \quad d_{aux} = -2 - d_{wt} \qquad \text{(Eq. 3.7.510)}$$

The storage in the unconfined aquifer is the water in saturated soil from water table depth to its riverbed (d_{rb}, [m]):

$$S_{aquifer} = (d_{wt} - d_{rb}) \times \theta_{soil,max} \qquad \text{(Eq. 3.7.511)}$$

Riverbed is given by a spatially varied map in the input static file. The soil is completely saturated, so uses $\theta_{soil,max}$. Combining these two storage terms becomes the storage for below 2-m:

$$S_{gw} = \theta_{ToWT} \times d_{aux} + (d_{wt} - d_{rb}) \times \theta_{soil,max} \qquad \text{(Eq. 3.7.512)}$$

$$\Delta S_{gw} = R_{sub} + Q_{lat} - Q_r - Q_{spring} = \Delta(\theta_{ToWT} \times (-2 - d_{wt})) + \Delta(d_{wt} - d_{rb}) \times \theta_{soil,max} \qquad \text{(Eq. 3.7.513)}$$

And the residual term for the below 2-m Noah-MP soil column storage is:

$$res_{gw} = R_{sub} + Q_{lat} - Q_r - Q_{spring} - \Delta(\theta_{ToWT} \times (-2 - d_{wt}) + (d_{wt} - d_{rb}) \times \theta_{soil,max}) \qquad \text{(Eq. 3.7.514)}$$

Scenario 2 – shallow water table (d_{wt} > -3)

If the water table is above 3-m, in this case the auxiliary layer depth is 1-m and below the auxiliary layer is completely saturated. For this case, the auxiliary layer storage is:

$$S_{aux} = \theta_{ToWT} \times 1 \qquad \text{(Eq. 3.7.515)}$$

The water from below the auxiliary layer and riverbed is completely saturated:

$$S_{aquifer} = (-3 - d_{rb}) \times \theta_{soil,max} \qquad \text{(Eq. 3.7.516)}$$

Although the water table has already moved up above 3-m, the term $S_{aquifer}$ is still used here to indicate the water between auxiliary layer and riverbed. To calculate the total aquifer water, the depth between d_{wt} and -3 m needs to be considered, which is also completely saturated.

The residual term for the below 2-m soil column storage is:

$$res_{gw} = R_{sub} + Q_{lat} - Q_r - Q_{spring} - \Delta(\theta_{ToWT} \times 1 + (-3 - d_{rb}) \times \theta_{soil,max}) \qquad \text{(Eq. 3.7.517)}$$

Please also note that there is an internal flux, W_{rech}, crossing the water table depth between the unsaturated layer and saturated layer. The location of this W_{rech} flux also depends on water table depth. If the water table is below 3-m (d_{wt} <-3), the W_{rech} flux (i.e., $W_{rech,deep}$) is between the auxiliary layer and the unconfined aquifer. If the water table depth is above 3-m, the W_{rech} flux (i.e., $W_{rech,sha}$) is between the layer containing water table and the above unsaturated layer. This internal flux W_{rech} crossing water table depth is omitted when considering the water balance for the whole column soil.

Finally, the combined residual term for both above 2-m soil and below 2-m storage is:

$$res_{total} = \left(P - Q_{ET} - R_{sfc} + Q_{lat} - Q_r - Q_{spring}\right) - \Delta(\theta_{soil} \times 2 + \theta_{ToWT} \times d_{aux} + (d_{wt} - d_{rb}) \times \theta_{soil,max} + W_{liq,can} + W_{ice,can} + W_{snow}) \qquad \text{(Eq. 3.7.518)}$$

3.8 Biochemical Processes

Description:

There are currently two parts for the Noah-MP biochemical processes, one for dynamic vegetation treatment of generic natural vegetation land type and the other for explicit crop modeling for several specific crop types. Both are only for carbon-related processes, so treatments of other biochemical processes (e.g., nitrogen and phosphorus cycles) are needed in the future. The nitrogen dynamics has been developed in non-community Noah-MP versions managed by individual research groups (e.g., Cai et al., 2016), and will be integrated into the community Noah-MP version in the future.

Relevant code modules:

Module: BiochemNatureVegMainMod.F90

Subroutines: BiochemNatureVegMain

Module: BiochemCropMainMod.F90

Subroutines: BiochemCropMain

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptDynamicVeg Dynamic Vegetation	1	Off (use table LAI; use FVEG=SHDFAC from input)
	2	On (together with OPT_CRS=1)
	3	Off (use table LAI; calculate FVEG)
	4*	Off (use table LAI; use maximum FVEG)
	5	On (use maximum FVEG)
	6	On (use FVEG=SHDFAC from input)
	7	Off (use input LAI; use FVEG=SHDFAC from input)
	8	Off (use input LAI; calculate FVEG)
	9	Off (use input LAI; use maximum FVEG)
OptCropModel Explicit crop modeling	0*	No explicit crop model
	1	Liu et al. (2016) crop scheme

3.8.1 Carbon Process for Generic Vegetation

Description:

This is the dynamic vegetation carbon process module (BiochemNatureVegMainMod.F90) to handle generic vegetation types (including generic crop land types identified by MODIS/USGS land type data) as opposed to the explicit crop scheme in Sect 4.2. Currently, it only includes carbon processes based on Dickinson et al. (1998) and Niu et al. (2004). This dynamic vegetation carbon scheme is only activated for OptDynamicVeg = 2,5,6. The flag to activate this dynamic vegetation model is handled in PhenologyMainMod.F90.

Relevant code modules:

Module: BiochemNatureVegMainMod.F90

Subroutines: BiochemNatureVegMain

Module: CarbonFluxNatureVegMod.F90

Subroutines: CarbonFluxNatureVeg

Physics:

1. Prepare for carbon processes:

The soil water stress (W_{stress}; 1.0 for wilting) used for leaf respiration and death calculation is computed as:

$$W_{stress} = 1 - \sum_{i=1}^{N_{root}} F_{soil,transp}(i) \qquad \text{(Eq. 3.8.1)}$$

where N_{root} is the number of soil layers for root zone (set in NoahmpTable.TBL), and $F_{soil,transp}$ (= 0~1) is the soil transpiration factor computed in SoilWaterTranspirationMod.F90.

The total water amount in the root zone ($W_{rootzone}$, [m^3/m^3]) used for soil respiration calculation is computed as:

$$W_{rootzone} = \sum_{i=1}^{N_{root}} \left(\frac{\theta_{soil}(i)}{\theta_{soil,max}(i)} \times \frac{D_{zsnso}(i)}{-Z_{soil}(N_{root})} \right) \qquad \text{(Eq. 3.8.2)}$$

where θ_{soil} [m^3/m^3] is the soil moisture, $\theta_{soil,max}$ [m^3/m^3] is the saturated soil moisture (set in NoahmpTable.TBL), D_{zsnso} [m] is the thickness of snow/soil layer, and Z_{soil} [m] is the depth of each soil layer bottom from the surface (negative).

2. Carbon processes (CarbonFluxNatureVegMod.F90**):**

(1) Maintenance respiration rate for leaf, stem, root, and wood:

The maintenance respiration rate depends on several key factors, including the respiration reduction factor, the temperature factor, the foliage nitrogen factor, and the water stress factor.

The respiration reduction factor ($R_{S,f,red}$) is computed based on growing season:

$$R_{S,f,red} = \begin{cases} 0.5 & non-growing\ season \\ 1 & growing\ season \end{cases} \qquad \text{(Eq. 3.8.3)}$$

where the growing season indicator is determined in PhenologyMainMod.F90.

The foliage nitrogen factor ($F_{nf,rs}$) is

$$F_{nf,rs} = \min(1.0, \frac{C_{fn}}{\max(10^{-6},\ C_{fn,max})}) \qquad \text{(Eq. 3.8.4)}$$

where C_{fn} [%] is the foliage nitrogen concentration (=1 currently hard-coded in Noah-MP) and $C_{fn,max}$ is the maximum foliage nitrogen concentration when $C_{fn} = 1\%$ (set in NoahmpTable.TBL).

The temperature factor ($T_{f,rs}$) is

$$T_{f,rs} = \left(R_{S,q10}\right)^{\frac{T_v - 298.16}{10}} \qquad \text{(Eq. 3.8.5)}$$

where $R_{S,q10}$ is the change in maintenance respiration for every 10 degC temperature change and T_v [K] is the vegetation temperature.

Thus, the leaf respiration rate ($R_{S,leaf,mol}$, [umol CO_2/m^2/s]) is computed as:

$$R_{S,leaf,mol} = R_{S,leaf,maint,25C} \times T_{f,rs} \times F_{nf,rs} \times LAI \times R_{S,f,red} \times (1 - W_{stress}) \qquad \text{(Eq. 3.8.6)}$$

where LAI [m^2/m^2] is the leaf area index, W_{stress} is the water stress factor computed above, and $R_{S,leaf,maint,25C}$ [umol CO_2/m^2/s] is the leaf maintenance respiration rate at 25 degC (set in NoahmpTable.TBL).

The leaf maintenance respiration rate ($R_{S,leaf,maint}$, [gC/m^2/s]) is computed as:

$$R_{S,leaf,maint} = \min(\frac{M_{leaf} - M_{leaf,min}}{\Delta t}, R_{S,leaf,mol} \times 12 \times 10^{-6}) \qquad \text{(Eq. 3.8.7)}$$

where M_{leaf} [gC/m^2] is the leaf mass and $M_{leaf,min}$ [gC/m^2] is the minimum allowed leaf mass. The "12×10^{-6}" is a unit conversion factor from umol CO_2 to gC.

The leaf mass is initialized as:

$$M_{leaf} = \max(LAI, 0.05) \times \frac{1000}{\max(1, M_{SLA})} \qquad \text{(Eq. 3.8.8)}$$

where M_{SLA} [kg/m^2] is the single-sided leaf area per mass (set in NoahmpTable.TBL).

The minimum leaf mass is computed as:

$$M_{leaf,min} = LAI_{min} \times \frac{1000}{M_{SLA}} \qquad \text{(Eq. 3.8.9)}$$

where LAI_{min} is the minimum LAI (=0.05, currently set in NoahmpTable.TBL).

The root maintenance respiration rate ($R_{S,root,maint}$, [gC/m^2/s]) is computed as:

$$R_{S,root,maint} = R_{S,root,maint,25C} \times (M_{root} \times 10^{-3}) \times T_{f,rs} \times R_{S,f,red} \times 12 \times 10^{-6} \qquad \text{(Eq. 3.8.10)}$$

where $R_{S,root,maint,25C}$ [umol CO_2/kgC/s] is the root maintenance respiration rate at 25 degC (set in NoahmpTable.TBL) and M_{root} [gC/m^2] is the root mass initialized to 500.0 gC/m^2.

The stem maintenance respiration rate ($R_{S,stem,maint}$, [gC/m^2/s]) is computed as:

$$R_{S,stem,maint} = R_{S,stem,maint,25C} \times ((M_{stem} - M_{stem,min}) \times 10^{-3}) \times T_{f,rs} \times R_{S,f,red} \times 12 \times 10^{-6}$$

(Eq. 3.8.11)

where $R_{S,stem,maint,25C}$ [umol CO_2/kgC/s] is the stem maintenance respiration rate at 25 degC (set in NoahmpTable.TBL).

M_{stem} [gC/m^2] is the stem mass initialized as:

$$M_{stem} = \max(0.1 \times LAI, 0.05) \times \frac{1000}{3.0} \qquad \text{(Eq. 3.8.12)}$$

$M_{stem,min}$ [gC/m^2] is the minimum stem mass computed as:

$$M_{stem,min} = SAI_{min} \times \frac{1000}{3.0} \qquad \text{(Eq. 3.8.13)}$$

where SAI_{min} is the minimum SAI (=0.05, currently set in NoahmpTable.TBL).

The wood maintenance respiration rate ($R_{S,wood,maint}$, [gC/m^2/s]) is computed as:

$$R_{S,wood,maint} = C_{wood,RS} \times e^{0.08\times(T_v-298.16)} \times M_{wood} \times I_{wood} \quad \text{(Eq. 3.8.14)}$$

where $C_{wood,RS}$ [1/s] is the wood respiration coefficient, T_v [K] is the vegetation temperature, I_{wood} [=1 or 0] is the wood pool index depending woody or not (set in NoahmpTable.TBL), and M_{woo} [gC/m^2] is the wood mass (including woody roots) initialized to 500.0 gC/m^2.

(2) Carbon assimilation rate:

The total carbon assimilation rate ($C_{assim,tot}$, [gC/m^2/s]) is computed as:

$$C_{assim,tot} = P_{SN,tot} \times 12 \times 10^{-6} \quad \text{(Eq. 3.8.15)}$$

where $P_{SN,tot}$ [umol CO_2/m^2/s] is the total photosynthesis rate computed in the energy component (EnergyMainMod.F90).

(3) Carbon allocation to leaf, stem, root, and wood:

The fraction of carbon flux goes into leaf ($f_{c,leaf}$) is computed as:

$$f_{c,leaf} = e^{\left[0.01\times LAI\times(1-e^{0.75\times LAI})\right]} \quad \text{(Eq. 3.8.16)}$$

except when the vegetation type is evergreen broadleaf forest, then the “0.75” coefficient should be replaced by “0.5”.

The fraction of carbon flux goes into stem ($f_{c,stem}$) is computed as:

$$f_{c,stem} = \frac{LAI}{10} \times f_{c,leaf} \quad \text{(Eq. 3.8.17)}$$

Then the $f_{c,leaf}$ is further updated by removing $f_{c,stem}$ as follows:

$$f_{c,leaf} = f_{c,leaf} - f_{c,stem} \quad \text{(Eq. 3.8.18)}$$

The fraction of carbon flux goes into wood ($f_{c,wood}$) is computed as:

$$f_{c,wood} = (1 - f_{c,leaf} - f_{c,stem}) \times f_{rac,wood} \quad \text{(Eq. 3.8.19)}$$

$$f_{rac,wood} = \begin{cases} \left[1 - \dfrac{e^{(-F_{wood,alloc}\times R_{w2r}\times \frac{M_{root}}{M_{wood}})}}{F_{wood,alloc}}\right] \times I_{wood} & M_{wood} > 10^{-6} \\ I_{wood} & M_{wood} \le 10^{-6} \end{cases} \quad \text{(Eq. 3.8.20)}$$

where $f_{rac,wood}$ is the wood carbon fraction in wood and root pools, $F_{wood,alloc}$ is the wood carbon allocation factor (set in NoahmpTable.TBL), and R_{w2r} is the wood to root ratio (set in NoahmpTable.TBL).

The fraction of carbon flux goes into root ($f_{c,root}$) is computed as:

$$f_{c,root} = (1 - f_{c,leaf} - f_{c,stem}) \times (1 - f_{rac,wood}) \quad \text{(Eq. 3.8.21)}$$

(4) Turnover rates for leaf, stem, root, and wood:

The turnover rates ($R_{turnover}$, [gC/m^2/s]) for different components are computed as:

$$R_{turnover,leaf} = C_{turnover,leaf} \times 5 \times 10^{-7} \times M_{leaf} \quad \text{(Eq. 3.8.22)}$$

$$R_{turnover,stem} = C_{turnover,leaf} \times 5 \times 10^{-7} \times M_{stem} \quad \text{(Eq. 3.8.23)}$$

$$R_{turnover,root} = C_{turnover,root} \times M_{root} \quad \text{(Eq. 3.8.24)}$$

$$R_{turnover,wood} = 9.5 \times 10^{-10} \times M_{wood} \quad \text{(Eq. 3.8.25)}$$

where $C_{turnover,leaf}$ [1/s] and $C_{turnover,root}$ [1/s] are leaf and root turnover rate coefficient, respectively (set in NoahmpTable.TBL).

(5) Seasonal death rate for leaf and stem:

The seasonal leaf and stem death rates are based on temperature and water stress. Water stress (W_{stress}, computed above) is set to 1 at permanent wilting point.

The death coefficients due to temperature and water stress are computed as:

$$f_{temp,death} = e^{[-0.3 \times \max(0, T_v - T_{d,leaf})]} \times \frac{M_{leaf}}{120} \quad \text{(Eq. 3.8.26)}$$

$$f_{water,death} = e^{[(W_{stress} - 1) \times C_{d,water}]} \quad \text{(Eq. 3.8.27)}$$

where $f_{temp,death}$ and $f_{water,death}$ are the death coefficients due to temperature and water stress, respectively. $T_{d,leaf}$ [K] is the characteristic temperature for leaf freezing (set in NoahmpTable.TBL), and $C_{d,water}$ is the water stress coefficient (set in NoahmpTable.TBL).

Then, the seasonal death rates for leaf ($D_{y,leaf}$, [gC/m^2/s]) and stem ($D_{y,stem}$, [gC/m^2/s]) are:

$$D_{y,leaf} = M_{leaf} \times 10^{-6} \times (f_{temp,death} \times C_{d,t,leaf} + f_{water,death} \times C_{d,w,leaf}) \quad \text{(Eq. 3.8.28)}$$

$$D_{y,stem} = M_{stem} \times 10^{-6} \times (f_{temp,death} \times C_{d,t,leaf} + f_{water,death} \times C_{d,w,leaf}) \quad \text{(Eq. 3.8.29)}$$

where $C_{d,t,leaf}$ [1/s] and $C_{d,w,leaf}$ [1/s] are the temperature and water coefficients for leaf death, respectively (set in NoahmpTable.TBL).

(6) Growth respiration for leaf, stem, root, and wood:

The growth respiration rates ($R_{S,grow}$, [gC/m^2/s]) for leaf, stem, root, and wood are computed as:

$$R_{S,leaf,grow} = \max(0, F_{grow} \times (f_{c,leaf} \times C_{assim,tot} - R_{S,leaf,maint}) \qquad \text{(Eq. 3.8.30)}$$

$$R_{S,stem,grow} = \max(0, F_{grow} \times (f_{c,stem} \times C_{assim,tot} - R_{S,stem,maint}) \qquad \text{(Eq. 3.8.31)}$$

$$R_{S,root,grow} = \max(0, F_{grow} \times (f_{c,root} \times C_{assim,tot} - R_{S,root,maint}) \qquad \text{(Eq. 3.8.32)}$$

$$R_{S,wood,grow} = \max(0, F_{grow} \times (f_{c,wood} \times C_{assim,tot} - R_{S,wood,maint}) \qquad \text{(Eq. 3.8.33)}$$

where $C_{assim,tot}$ [gC/m^2/s] is the total carbon assimilation rate computed above and F_{grow} is the growth respiration fraction (set in NoahmpTable.TBL).

(7) Adjustment for low temperature limit for photosynthesis:

The added net primary productivity ($N_{PP,add}$, [gC/m^2/s]) to leaf and stem after respiration is:

$$N_{PP,add,leaf} = \max(0, f_{c,leaf} \times C_{assim,tot} - R_{S,leaf,grow} - R_{S,leaf,maint}) \qquad \text{(Eq. 3.8.34)}$$

$$N_{PP,add,stem} = \max(0, f_{c,stem} \times C_{assim,tot} - R_{S,stem,grow} - R_{S,stem,maint}) \qquad \text{(Eq. 3.8.35)}$$

When the vegetation temperature (T_v, [K]) is below the minimum temperature for photosynthesis ($T_{min,photosyn}$, [K]), then

$$N_{PP,add,leaf} = 0 \quad if\ T_v < T_{min,photosyn} \qquad \text{(Eq. 3.8.36)}$$

$$N_{PP,add,stem} = 0 \quad if\ T_v < T_{min,photosyn} \qquad \text{(Eq. 3.8.37)}$$

(8) Limit death rate for leaf and stem to avoid reducing mass below the minimum threshold:

The seasonal death rates of leaf and stem are further updated to ensure leaf and stem mass are not below their minimum values ($M_{leaf,min}$ and $M_{stem,min}$ computed above) with the mass conservation:

$$\Delta M_{leaf,max} = \frac{M_{leaf} - M_{leaf,min}}{\Delta t} \qquad \text{(Eq. 3.8.38)}$$

$$\Delta M_{stem,max} = \frac{M_{stem} - M_{stem,min}}{\Delta t} \qquad \text{(Eq. 3.8.39)}$$

$$D_{y,leaf} = \min(D_{y,leaf}, \Delta M_{leaf,max} + N_{PP,add,leaf} - R_{turnover,leaf}) \qquad \text{(Eq. 3.8.40)}$$

$$D_{y,stem} = \min(D_{y,stem}, \Delta M_{stem,max} + N_{PP,add,stem} - R_{turnover,stem}) \qquad \text{(Eq. 3.8.41)}$$

(9) Compute net primary productivities for leaf, stem, root, and wood:

The total net primary productivity (N_{PP}, [gC/m^2/s]) is computed as:

$$N_{PP,leaf} = \max(N_{PP,add,leaf}, -\Delta M_{leaf,max}) \qquad \text{(Eq. 3.8.42)}$$

$$N_{PP,stem} = \max(N_{PP,add,stem}, -\Delta M_{stem,max}) \qquad \text{(Eq. 3.8.43)}$$

$$N_{PP,root} = f_{c,root} \times C_{assim,tot} - R_{S,root,grow} - R_{S,root,maint} \quad \text{(Eq. 3.8.44)}$$

$$N_{PP,wood} = f_{c,wood} \times C_{assim,tot} - R_{S,wood,grow} - R_{S,wood,maint} \quad \text{(Eq. 3.8.45)}$$

(10) Update mass for leaf, stem, root, and wood:

Based on the above mass change, the masses for leaf, stem, root, and wood are updated as:

$$M_{leaf} = M_{leaf} + (N_{PP,leaf} - R_{turnover,leaf} - D_{y,leaf}) \times \Delta t \quad \text{(Eq. 3.8.46)}$$

$$M_{stem} = M_{stem} + (N_{PP,stem} - R_{turnover,stem} - D_{y,stem}) \times \Delta t \quad \text{(Eq. 3.8.47)}$$

$$M_{root} = M_{root} + (N_{PP,root} - R_{turnover,root}) \times \Delta t \quad \text{(Eq. 3.8.48)}$$

$$M_{wood} = \left[M_{wood} + (N_{PP,wood} - R_{turnover,wood}) \times \Delta t\right] \times I_{wood} \quad \text{(Eq. 3.8.49)}$$

If $M_{root} < 0$, then re-adjust root mass to zero as:

$$R_{turnover,root} = N_{PP,root} \quad \text{(Eq. 3.8.50)}$$

$$M_{root} = 0 \quad \text{(Eq. 3.8.51)}$$

(11) Compute soil carbon budget:

The short-lived soil carbon pool (typically in shallow soil layers, $P_{c,fast}$, [gC/m^2]) is:

$$P_{c,fast} = P_{c,fast} + (R_{turnover,leaf} + R_{turnover,stem} + R_{turnover,root} + R_{turnover,wood} + D_{y,leaf} + D_{y,stem}) \times \Delta t \quad \text{(Eq. 3.8.52)}$$

The soil temperature factor for microbial respiration ($F_{resp,micro,temp}$) is:

$$F_{resp,micro,temp} = 2^{\left[\frac{T_{soil}(1) - 283.16}{10}\right]} \quad \text{(Eq. 3.8.53)}$$

where $T_{soil}(1)$ is the soil temperature for the first layer.

The soil water factor for microbial respiration ($F_{resp,micro,water}$) is:

$$F_{resp,micro,water} = \frac{0.23 \times W_{rootzone}}{(0.2 + W_{rootzone}) \times (0.23 + W_{rootzone})} \quad \text{(Eq. 3.8.54)}$$

where $W_{rootzone}$ [m^3/m^3] is the root-zone soil water computed above.

Thus, the soil microbial respiration rate ($R_{S,soil,micro}$, [gC/m^2/s]) is computed as:

$$R_{S,soil,micro} = F_{resp,micro,temp} \times F_{resp,micro,water} \times 12 \times 10^{-6} \times \max(0, P_{c,fast} \times 10^{-3}) \times f_{resp,micro} \quad \text{(Eq. 3.8.55)}$$

where $f_{resp,micro}$ [umol/kgC/s] is the microbial respiration coefficient (set in NoahmpTable.TBL).

The rate for short-lived carbon decay to long-lived/stable carbon ($R_{c,decay}$, [gC/m^2/s]) is:

$$R_{c,decay} = 0.1 \times R_{S,soil,micro} \quad \text{(Eq. 3.8.56)}$$

Then, both short-lived and long-lived (stable, typically in deep soil) carbon storage are updated:

$$P_{c,fast} = P_{c,fast} - (R_{S,soil,micro} + R_{c,decay}) \times \Delta t \qquad \text{(Eq. 3.8.57)}$$

$$P_{c,stable} = P_{c,stable} + R_{c,decay} \times \Delta t \qquad \text{(Eq. 3.8.58)}$$

The net carbon flux from land to the atmosphere ($F_{c,land2atmos}$, [gC/m^2/s]) is:

$$F_{c,land2atmos} = -C_{assim,tot} + R_{S,leaf,grow} + R_{S,leaf,maint} + R_{S,stem,grow} + R_{S,stem,maint} + R_{S,root,grow} + R_{S,root,maint} + R_{S,wood,grow} + R_{S,wood,maint} + 0.9 \times R_{S,soil,micro} \qquad \text{(Eq. 3.8.59)}$$

(12) Compute diagnostic output variables and update LAI and SAI:

The total gross primary productivity (G_{PP}, [gC/m^2/s]) is:

$$G_{PP} = C_{assim,tot} \qquad \text{(Eq. 3.8.60)}$$

The total net primary productivity ($N_{PP,tot}$, [gC/m^2/s]) is:

$$N_{PP,tot} = N_{PP,leaf} + N_{PP,stem} + N_{PP,root} + N_{PP,wood} \qquad \text{(Eq. 3.8.61)}$$

The total plant respiration (net ecosystem respiration) rate ($R_{S,plant,tot}$, [gC/m^2/s]) is:

$$R_{S,plant,tot} = R_{S,leaf,grow} + R_{S,leaf,maint} + R_{S,stem,grow} + R_{S,stem,maint} + R_{S,root,grow} + R_{S,root,maint} + R_{S,wood,grow} + R_{S,wood,maint} \qquad \text{(Eq. 3.8.62)}$$

The total soil (organic) respiration rate ($R_{S,soil,tot}$, [gC/m^2/s]) is:

$$R_{S,soil,tot} = 0.9 \times R_{S,soil,micro} \qquad \text{(Eq. 3.8.63)}$$

The net ecosystem exchange (N_{EE}, [gCO_2/m^2/s]) is:

$$N_{EE} = (R_{S,plant,tot} + R_{S,soil,tot} - G_{PP}) \times \frac{44}{12} \qquad \text{(Eq. 3.8.64)}$$

The total soil carbon storage ($P_{c,soil}$, [gC/m^2]) is:

$$P_{c,soil} = P_{c,fast} + P_{c,stable} \qquad \text{(Eq. 3.8.65)}$$

The total living (biomass) carbon storage ($P_{c,live}$, [gC/m^2]) is:

$$P_{c,live} = M_{leaf} + M_{stem} + M_{root} + M_{wood} \qquad \text{(Eq. 3.8.66)}$$

Finally, the LAI and SAI are further updated:

$$LAI = \max(LAI_{min}, M_{leaf} \times \frac{M_{SLA}}{1000}) \qquad \text{(Eq. 3.8.67)}$$

$$SAI = \max(SAI_{min}, M_{stem} \times \frac{3}{1000}) \qquad \text{(Eq. 3.8.68)}$$

3.8.2 Carbon Process for Explicit Crop Modeling

Description:

This is the dynamic crop carbon process module (BiochemCropMainMod.F90) to handle explicit crop types. Currently, it only includes carbon processes based on Liu et al. (2016) and Zhang et al. (2020). This dynamic crop carbon model is only activated for OptCropModel = 1 and CropType >0 (explicit crop type instead of generic crop from MODIS/USGS). The flag to activate this crop model is handled in PhenologyMainMod.F90. The initial model development and specific details on model parameters have been well documented in Liu et al. (2016), in which two major crops in the U.S., corn and soybean, were incorporated from two sites in Bondville, IL, and Mead, NE, respectively. Further, Zhang et al. (2020) attempted to transfer the site-scale modeling study to large regional simulation in the U.S. cornbelt and lower Mississippi river valley. The modeling capability for spring wheat is currently under development. Note that although there are also sorghum, rice, and winter wheat crop types in the NoahmpTable.TBL, they are not used currently because parameters are incorrect. There is recent study developing new spring wheat crop dynamics (Zhang et al., 2023), which will be implemented in the community Noah-MP version in the future.

Relevant code modules:

Module: BiochemCropMainMod.F90

Subroutines: BiochemCropMain

Module: CarbonFluxCropMod.F90

Subroutines: CarbonFluxCrop

Module: CropGrowDegreeDayMod.F90

Subroutines: CropGrowDegreeDay

Module: CropPhotosynthesisMod.F90

Subroutines CropPhotosynthesis

Physics:

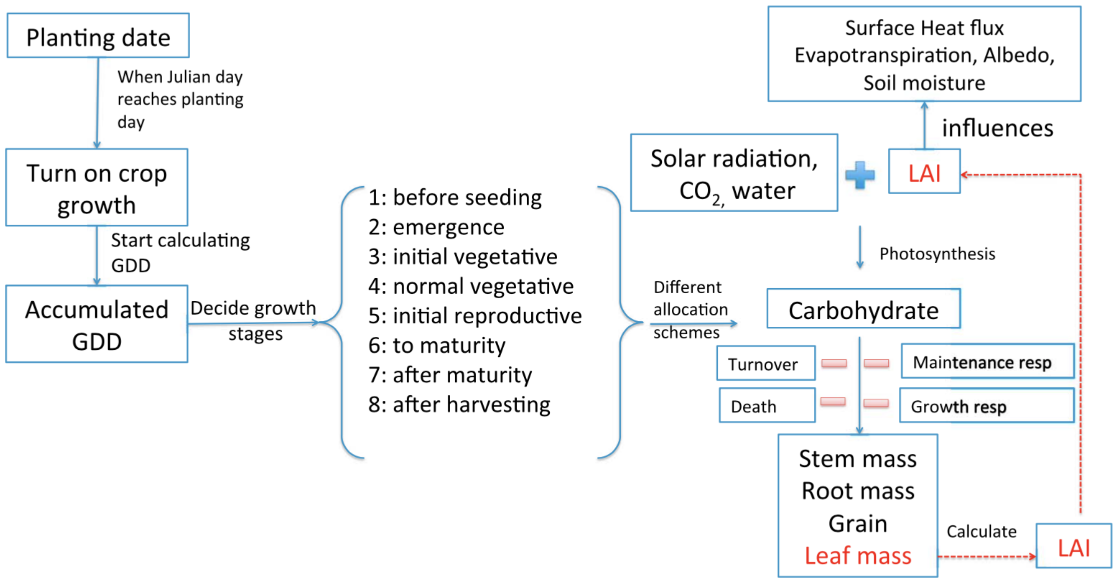


Figure 18. Flowchart of the Liu et al. (2016) crop model (adapted from Liu et al. (2016)).

1. Prepare for carbon processes:

Similar to the generic dynamic vegetation model shown above in Sect. 4.1, the soil water stress (W_{stress}; 1.0 for wilting) used for leaf respiration and death calculation is computed as:

$$W_{stress} = 1 - \sum_{i=1}^{N_{root}} F_{soil,transp}(i) \qquad \text{(Eq. 3.8.69)}$$

where N_{root} is the number of soil layers for root zone (set in NoahmpTable.TBL), and $F_{soil,transp}$ (= 0~1) is the soil transpiration factor computed in SoilWaterTranspirationMod.F90.

The total water amount in the root zone ($W_{rootzone}$, [m^3/m^3]) used for soil respiration calculation is computed as:

$$W_{rootzone} = \sum_{i=1}^{N_{root}} \left(\frac{\theta_{soil}(i)}{\theta_{soil,max}(i)} \times \frac{D_{zsnso}(i)}{-Z_{soil}(N_{root})} \right) \qquad \text{(Eq. 3.8.70)}$$

where θ_{soil} [m^3/m^3] is the soil moisture, $\theta_{soil,max}$ [m^3/m^3] is the saturated soil moisture (set in NoahmpTable.TBL), D_{zsnso} [m] is the thickness of snow/soil layer, and Z_{soil} [m] is the depth of each soil layer bottom from the surface (negative).

2. Crop photosynthesis (CropPhotosynthesisMod.F90**):**

Note that currently this module is not used by Noah-MP, whereas the crop photosynthesis rate ($P_{SN,tot}$ [umol CO_2/m^2/s]) comes from the computation in the energy component (EnergyMainMod.F90) for the consideration of model consistency in photosynthesis and energy balance calculations. Such treatment is the same as the dynamic vegetation scheme for generic vegetation in the above section.

However, for completeness, here the crop photosynthesis scheme (an empirical parameterization) is also briefly described below, though not used currently.

(1) Compute the maximum CO_2 assimilation rate ($A_{CO2,max}$, [g CO_2/m^2/s]):

$$A_{CO2,max} = \begin{cases} 10^{-10} & if\ T_{air,2m} < T_{assim,min} \\ \frac{T_{air,2m} - T_{assim,min}}{T_{assim,max} - T_{assim,min}} \times A_{CO2,ref,max} & if\ T_{assim,min} \le T_{air,2m} < T_{assim,max} \\ A_{CO2,ref,max} & if\ T_{assim,max} \le T_{air,2m} < T_{max-assim,max} \\ A_{CO2,ref,max} - 0.2 \times (T_{air,2m} - T_{max-assim,max}) & if\ T_{air,2m} \ge T_{max-assim,max} \end{cases}$$

(Eq. 3.8.71)

$$A_{CO2,max} = \max(A_{CO2,max}, 0.01) \quad \text{(Eq. 3.8.72)}$$

where $T_{air,2m}$ [degC] is the 2-m air temperature, $T_{assim,min}$ [degC] is the minimum temperature for CO_2 assimilation, $T_{assim,max}$ [degC] is the maximum temperature until which CO_2 assimilation linearly increases, $T_{max-assim,max}$ [degC] is the maximum temperature until which CO_2 assimilation remains at the maximum reference CO_2 assimilation rate ($A_{CO2,ref,max}$, [gCO_2/m^2/s]). All these parameters except $T_{air,2m}$ are set in NoahmpTable.TBL.

(2) Compute light extinction coefficients:

$$L_1 = 0.1127 \times LAI \quad \text{(Eq. 3.8.73)}$$

$$L_2 = 0.5 \times LAI \quad \text{(Eq. 3.8.74)}$$

$$L_3 = 0.8873 \times LAI \quad \text{(Eq. 3.8.75)}$$

where LAI is the leaf area index, which is set to 0.05 if smaller than 0.05.

The light extinction parameters are computed as:

$$I_1 = \max(\ 10^{-10}, C_{ext} \times R_{photo} \times e^{-C_{ext} \times L_1}) \quad \text{(Eq. 3.8.76)}$$

$$I_2 = \max(\ 10^{-10}, C_{ext} \times R_{photo} \times e^{-C_{ext} \times L_2}) \quad \text{(Eq. 3.8.77)}$$

$$I_3 = \max(\ 10^{-10}, C_{ext} \times R_{photo} \times e^{-C_{ext} \times L_3}) \quad \text{(Eq. 3.8.78)}$$

$$R_{photo} = f_{photo} \times S_{down} \times 0.0036 \quad \text{(Eq. 3.8.79)}$$

where R_{photo} [MJ/m^2/h] is the photosynthetically active radiation, C_{ext} is the light extinction coefficient (set in NoahmpTable.TBL), f_{photo} is the fraction of incoming solar radiation to photosynthetically active radiation (set in NoahmpTable.TBL), and S_{down} [W/m^2] is the downward shortwave radiation (from forcing).

(3) Compute actual photosynthesis coefficients:

$$A_1 = A_{CO2,max} \times (1 - e^{-\frac{L_{u,eff} \times I_1}{A_{CO2,max}}}) \quad \text{(Eq. 3.8.80)}$$

$$A_2 = A_{CO2,max} \times (1 - e^{-\frac{L_{u,eff} \times I_2}{A_{CO2,max}}}) \times 1.6 \quad \text{(Eq. 3.8.81)}$$

$$A_3 = A_{CO2,max} \times (1 - e^{-\frac{L_{u,eff} \times I_3}{A_{CO2,max}}}) \quad \text{(Eq. 3.8.82)}$$

where $L_{u,eff}$ is the light use efficiency (set in NoahmpTable.TBL).

(4) Compute photosynthesis rate (i.e., CO_2 assimilation rate, [g CO_2/m^2/h]):

$$A_{CO2,tot} = \frac{A_1 + A_2 + A_3}{3.6} \times LAI \quad \text{(Eq. 3.8.83)}$$

$$LAI = \begin{cases} 0.05 & if\ LAI \le 0.05 \\ LAI & if\ 0.05 < LAI \le 4 \\ 4 & if\ LAI > 4. \end{cases} \quad \text{(Eq. 3.8.84)}$$

$$A_{CO2,tot} = A_{CO2,tot} \times F_{assim,red} \quad \text{(Eq. 3.8.85)}$$

where $F_{assim,red}$ is the CO_2 assimilation reduction factor (0-1) (caused by e.g., pest and weeds; set in NoahmpTable.TBL).

The final crop photosynthesis rate ($P_{SN,crop,tot}$ [umol CO_2/m^2/s]) is:

$$P_{SN,crop,tot} = A_{CO2,tot} \times \frac{10^6}{44 \times 3600} \quad \text{(Eq. 3.8.86)}$$

3. Crop growing stages (CropCrowDegreeDayMod.F90**):**

This part is to determine the crop growing stage based on day of year and temperature.

(1) Compute planting and harvest indices:

The planting index (I_{plant}) is initialized to 1 (turned on) and harvest index ($I_{harvest}$) is initialized to 1 (turned off). If the current model day of the year is earlier than the planting date (set in NoahmpTable.TBL), then the planting is turned off ($I_{plant} = 0$). If the current model day of the year is later than the harvest date (set in NoahmpTable.TBL), then the harvest is turned on ($I_{harvest} = 0$).

(2) Compute growing degree days based on temperature:

$$T_{diff} = \begin{cases} 0 & if\ T_{air,2m} < T_{gdd,base} \\ T_{air,2m} - T_{gdd,base} & if\ T_{gdd,base} \le T_{air,2m} < T_{gdd,max} \\ T_{gdd,max} - T_{gdd,base} & if\ LT_{air,2m} \ge T_{gdd,max} \end{cases} \quad \text{(Eq. 3.8.87)}$$

where T_{diff} [degC] is the temperature difference, $T_{air,2m}$ [degC] is the 2-m air temperature, $T_{gdd,base}$ [degC] is the base temperature for growing degree day accumulation, and $T_{gdd,max}$ [degC] is the maximum temperature for growing degree day accumulation.

Then the growing degree day (G_{DD}) and is computed as:

$$G_{DD} = (G_{DD} + T_{diff} \times \frac{\Delta t}{86400}) \times I_{plant} \times I_{harvest} \quad \text{(Eq. 3.8.88)}$$

The G_{DD} count ($G_{DD,cnt}$) is computed as:

$$G_{DD,cnt} = G_{DD} \quad \text{(Eq. 3.8.89)}$$

(3) Compute crop growing stage ($G_{S,crop}$) based on G_{DD}:

The growing stage is first initialized to the "before planting" stage ($G_{S,crop} = 1$).

$$G_{S,crop} = \begin{cases} 2 & if\ G_{DD,cnt} > 0 \\ 3 & if\ G_{DD,cnt} \geq G_{DD,emerg} \\ 4 & if\ G_{DD,cnt} \geq G_{DD,init,veg} \\ 5 & if\ G_{DD,cnt} \geq G_{DD,post,veg} \\ 6 & if\ G_{DD,cnt} \geq G_{DD,reprod} \\ 7 & if\ G_{DD,cnt} \geq G_{DD,mature} \\ 8 & if\ D_{OY} \geq D_{harvest} \\ 1 & if\ D_{OY} < D_{plant} \end{cases} \quad \text{(Eq. 3.8.90)}$$

where $G_{DD,emerg}$ is the G_{DD} from seeding to emergence stages, $G_{DD,init,veg}$ is the G_{DD} from seeding to initial vegetative stages, $G_{DD,post,veg}$ is the G_{DD} from seeding to post-vegetative stages, $G_{DD,reprod}$ is the G_{DD} from seeding to initial reproductive stages, $G_{DD,mature}$ is the G_{DD} from seeding to physical maturity. $D_{harvest}$ and D_{plant} are the dates for harvest and planting. All these parameters are set in NoahmpTable.TBL. D_{OY} is the current model day of the year.

4. Crop carbon processes (CarbonFluxCropMod.F90**):**

This part is to compute all carbon processes for crop, which is similar to the treatment of the generic vegetation above with additional treatment for crop grains.

(1) Carbon and carbohydrate assimilation rate:

The total carbon ($C_{assim,tot}$, [gC/m^2/s]) and carbohydrate ($C_{H2O,assim,tot}$, [gCH$_2$O/m^2/s]) assimilation rates are computed as:

$$C_{assim,tot} = P_{SN,tot} \times 12 \times 10^{-6} \quad \text{(Eq. 3.8.91)}$$

$$C_{H2O,assim,tot} = P_{SN,tot} \times 30 \times 10^{-6} \quad \text{(Eq. 3.8.92)}$$

where $P_{SN,tot}$ [umol CO_2/m^2/s] is the total photosynthesis rate computed in the energy component (EnergyMainMod.F90).

(2) Maintenance respiration rate for leaf, stem, root, and grain:

The maintenance respiration rate depends on several key factors, including the temperature factor, the foliage nitrogen factor, and the water stress factor.

The foliage nitrogen factor ($F_{nf,rs}$) is

$$F_{nf,rs} = \min(1.0, \frac{C_{fn}}{\max(10^{-6}, C_{fn,max})}) \quad \text{(Eq. 3.8.93)}$$

where C_{fn} [%] is the foliage nitrogen concentration (=1 currently hard-coded in Noah-MP) and $C_{fn,max}$ is the maximum foliage nitrogen concentration when $C_{fn} = 1\%$ (set in NoahmpTable.TBL).

The temperature factor ($T_{f,rs}$) is

$$T_{f,rs} = \left(R_{S,q10}\right)^{\frac{T_v - 298.16}{10}} \quad \text{(Eq. 3.8.94)}$$

where $R_{S,q10}$ is the change in maintenance respiration for every 10 degC temperature change and T_v [K] is the vegetation temperature.

Thus, the leaf respiration rate ($R_{S,leaf,mol}$, [umol CO_2/m^2/s]) is computed as:

$$R_{S,leaf,mol} = R_{S,leaf,maint,25C} \times T_{f,rs} \times F_{nf,rs} \times LAI \times (1 - W_{stress}) \quad \text{(Eq. 3.8.95)}$$

where *LAI* [m^2/m^2] is the leaf area index, W_{stress} is the water stress factor computed above, and $R_{S,leaf,maint,25C}$ [umol CO_2/m^2/s] is the leaf maintenance respiration rate at 25 degC (set in NoahmpTable.TBL).

The leaf maintenance respiration rate ($R_{S,leaf,maint}$, [gCH_2O/m^2/s]) is computed as:

$$R_{S,leaf,maint} = \min(\frac{M_{leaf} - M_{leaf,min}}{\Delta t}, R_{S,leaf,mol} \times 30 \times 10^{-6}) \quad \text{(Eq. 3.8.96)}$$

where M_{leaf} [gCH_2O/m^2] is the leaf mass and $M_{leaf,min}$ [gCH_2O/m^2] is the minimum allowed leaf mass. The "30×10^{-6}" is a unit conversion factor from umol CO_2 to gCH_2O.

The leaf mass is initialized as:

$$M_{leaf} = \max(LAI, 0.05) \times \frac{1000}{M_{SLA}} \quad \text{(Eq. 3.8.97)}$$

where M_{SLA} [kg/m^2] is the single-sided leaf area per mass (=15 for corn, 30 for soybean, and 35 for other crop types).

The minimum leaf mass is computed as:

$$M_{leaf,min} = LAI_{min} \times \frac{1000}{35} \quad \text{(Eq. 3.8.98)}$$

where LAI_{min} is the minimum LAI (=0.05, currently set in NoahmpTable.TBL).

The root maintenance respiration rate ($R_{S,root,maint}$, [gCH$_2$O/m^2/s]) is computed as:

$$R_{S,root,maint} = R_{S,root,maint,25C} \times (M_{root} \times 10^{-3}) \times T_{f,rs} \times 30 \times 10^{-6} \quad \text{(Eq. 3.8.99)}$$

where $R_{S,root,maint,25C}$ [umol CO$_2$/kgC/s] is the root maintenance respiration rate at 25 degC (set in NoahmpTable.TBL) and M_{root} [gCH$_2$O/m^2] is the root mass initialized to 500.0 gCH$_2$O/m^2.

The stem maintenance respiration rate ($R_{S,stem,maint}$, [gCH$_2$O/m^2/s]) is computed as:

$$R_{S,stem,maint} = R_{S,stem,maint,25C} \times (M_{stem} \times 10^{-3}) \times T_{f,rs} \times 30 \times 10^{-6} \quad \text{(Eq. 3.8.100)}$$

where $R_{S,stem,maint,25C}$ [umol CO$_2$/kgC/s] is the stem maintenance respiration rate at 25 degC (set in NoahmpTable.TBL).

M_{stem} [gCH$_2$O/m^2] is the stem mass initialized as:

$$M_{stem} = \max(0.1 \times LAI, 0.05) \times \frac{1000}{3} \quad \text{(Eq. 3.8.101)}$$

$M_{stem,min}$ [gCH$_2$O/m^2] is the minimum stem mass computed as:

$$M_{stem,min} = SAI_{min} \times \frac{1000}{3} \quad \text{(Eq. 3.8.102)}$$

where SAI_{min} is the minimum SAI (=0.05, currently set in NoahmpTable.TBL).

The grain maintenance respiration rate ($R_{S,grain,maint}$, [gCH$_2$O/m^2/s]) is computed as:

$$R_{S,grain,maint} = R_{S,grain,maint,25C} \times (M_{grain} \times 10^{-3}) \times T_{f,rs} \times 30 \times 10^{-6} \quad \text{(Eq. 3.8.103)}$$

where $R_{S,grain,maint,25C}$ [umol CO$_2$/kgC/s] is the stem maintenance respiration rate at 25 degC (set in NoahmpTable.TBL), and M_{grain} [gCH$_2$O/m^2] is the grain mass.

(3) Growth respiration for leaf, stem, root, and grain:

The growth respiration rates ($R_{S,grow}$, [gCH$_2$O/m^2/s]) for leaf, stem, root, and grain are computed as:

$$R_{S,leaf,grow} = \max(0, F_{grow} \times (f_{c,leaf}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,leaf,maint}$$

(Eq. 3.8.104)

$$R_{S,stem,grow} = \max(0, F_{grow} \times (f_{c,stem}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,stem,maint})$$

(Eq. 3.8.105)

$$R_{S,root,grow} = \max(0, F_{grow} \times (f_{c,root}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,root,maint})$$

(Eq. 3.8.106)

$$R_{S,grain,grow} = \max(0, F_{grow} \times (f_{c,grain}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,grain,maint})$$

(Eq. 3.8.107)

where $C_{H2O,assim,tot}$ [gCH$_2$O/m^2/s] is the total carbohydrate assimilation rate computed above and F_{grow} is the growth respiration fraction (set in NoahmpTable.TBL). $f_{c,leaf}$, $f_{c,stem}$, $f_{c,root}$, and $f_{c,grain}$ are the fraction of carbohydrate flux goes into crop leaf, stem, root, and grain, respectively (set in NoahmpTable.TBL), as a function of crop growing stage ($G_{S,crop}$) computed above in CropCrowDegreeDayMod.F90.

(4) Turnover rates for leaf, stem, and root:

The turnover rates ($R_{turnover}$, [gCH$_2$O/m^2/s]) for different components are computed as:

$$R_{turnover,leaf} = C_{turnover,leaf}(G_{S,crop}) \times 10^{-6} \times M_{leaf} \quad \text{(Eq. 3.8.108)}$$

$$R_{turnover,stem} = C_{turnover,leaf}(G_{S,crop}) \times 10^{-6} \times M_{stem} \quad \text{(Eq. 3.8.109)}$$

$$R_{turnover,root} = C_{turnover,root}(G_{S,crop}) \times 10^{-6} \times M_{root} \quad \text{(Eq. 3.8.110)}$$

where $C_{turnover,leaf}$, $C_{turnover,stem}$, and $C_{turnover,root}$ [1/s] are the turnover rate coefficients for leaf, stem, and root, respectively (set in NoahmpTable.TBL), as a function of crop growing stage ($G_{S,crop}$).

(5) Seasonal death rate for crop leaf:

The seasonal leaf death rate is based on temperature and water stress. Water stress (W_{stress}, computed above) is set to 1 at permanent wilting point.

The death coefficients due to temperature and water stress are computed as:

$$f_{temp,death} = e^{[-0.3 \times \max(0, T_v - T_{d,leaf})]} \times \frac{M_{leaf}}{120} \quad \text{(Eq. 3.8.111)}$$

$$f_{water,death} = e^{[(W_{stress} - 1) \times C_{d,water}]} \quad \text{(Eq. 3.8.112)}$$

where $f_{temp,death}$ and $f_{water,death}$ are the death coefficients due to temperature and water stress, respectively. $T_{d,leaf}$ [K] is the characteristic temperature for leaf freezing (set in NoahmpTable.TBL), and $C_{d,water}$ is the water stress coefficient (set in NoahmpTable.TBL).

Then, the seasonal death rate for leaf ($D_{y,leaf}$, [gCH$_2$O/m^2/s]) is:

$$D_{y,leaf} = M_{leaf} \times 10^{-6} \times (f_{temp,death} \times C_{d,t,leaf}(G_{S,crop}) + f_{water,death} \times C_{d,w,leaf}(G_{S,crop})) \quad \text{(Eq. 3.8.113)}$$

where $C_{d,t,leaf}$ [1/s] and $C_{d,w,leaf}$ [1/s] are the temperature and water coefficients for leaf death, respectively (set in NoahmpTable.TBL), as a function of crop growing stage ($G_{S,crop}$).

(6) Allocation of carbohydrate flux to leaf and stem at each growing stage:

The added net primary productivity ($N_{PP,add}$, [gCH$_2$O/m^2/s]) to leaf and stem after respiration is:

$$N_{PP,add,leaf} = f_{c,leaf}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,leaf,grow} - R_{S,leaf,maint} \quad \text{(Eq. 3.8.114)}$$

$$N_{PP,add,stem} = f_{c,stem}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,stem,grow} - R_{S,stem,maint} \quad \text{(Eq. 3.8.115)}$$

(7) Limit leaf death rate and leaf and stem turnover rates to avoid reducing mass below the minimum threshold:

The seasonal death rate and turnover rates are further updated to ensure leaf and stem mass are not below its minimum value ($M_{leaf,min}$ and $M_{stem,min}$ computed above) with the mass conservation:

$$\Delta M_{leaf,max} = \frac{M_{leaf} - M_{leaf,min}}{\Delta t} \quad \text{(Eq. 3.8.116)}$$

$$\Delta M_{stem,max} = \frac{M_{stem} - M_{stem,min}}{\Delta t} \quad \text{(Eq. 3.8.117)}$$

$$R_{turnover,leaf} = \min(R_{turnover,leaf}, \Delta M_{leaf,max} + N_{PP,add,leaf} \quad \text{(Eq. 3.8.118)}$$

$$R_{turnover,stem} = \min(R_{turnover,stem}, \Delta M_{stem,max} + N_{PP,add,stem} \quad \text{(Eq. 3.8.119)}$$

$$D_{y,leaf} = \min(D_{y,leaf}, \Delta M_{leaf,max} + N_{PP,add,leaf} - R_{turnover,leaf}) \quad \text{(Eq. 3.8.120)}$$

(8) Compute net primary productivities for leaf, stem, root, and grain:

The total net primary productivity (N_{PP}, [gCH$_2$O/m^2/s]) is computed as:

$$N_{PP,leaf} = N_{PP,add,leaf} \quad \text{(Eq. 3.8.121)}$$

$$N_{PP,stem} = N_{PP,add,stem} \quad \text{(Eq. 3.8.122)}$$

$$N_{PP,root} = f_{c,root}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,root,grow} - R_{S,root,maint} \quad \text{(Eq. 3.8.123)}$$

$$N_{PP,grain} = f_{c,grain}(G_{S,crop}) \times C_{H2O,assim,tot} - R_{S,grain,grow} - R_{S,grain,maint} \quad \text{(Eq. 3.8.124)}$$

(9) Update mass for leaf, stem, root, and grain:

Based on the above mass change, the masses ([gCH_2O/m^2]) for leaf, stem, root, and grain are updated as:

$$M_{leaf} = M_{leaf} + (N_{PP,leaf} - R_{turnover,leaf} - D_{y,leaf}) \times \Delta t \quad \text{(Eq. 3.8.125)}$$

$$M_{stem} = M_{stem} + (N_{PP,stem} - R_{turnover,stem}) \times \Delta t \quad \text{(Eq. 3.8.126)}$$

$$M_{root} = M_{root} + (N_{PP,root} - R_{turnover,root}) \times \Delta t \quad \text{(Eq. 3.8.127)}$$

$$M_{grain} = M_{grain} + N_{PP,grain} \times \Delta t \quad \text{(Eq. 3.8.128)}$$

Then, compute the carbohydrate translocation ($C_{x2grain}$, [gCH_2O/m^2]) from leaf, steam and root to grain:

$$C_{leaf2grain} = M_{leaf} \times f_{leaf2grain}(G_{S,crop}) \times \frac{\Delta t}{3600} \quad \text{(Eq. 3.8.129)}$$

$$C_{stem2grain} = M_{stem} \times f_{stem2grain}(G_{S,crop}) \times \frac{\Delta t}{3600} \quad \text{(Eq. 3.8.130)}$$

$$C_{root2grain} = M_{root} \times f_{root2grain}(G_{S,crop}) \times \frac{\Delta t}{3600} \quad \text{(Eq. 3.8.131)}$$

where $f_{leaf2grain}$ is the fraction of carbohydrate translocation from leaf, stem, or root to grain (set in NoahmpTable.TBL) as a function of crop growing stage ($G_{S,crop}$).

Then, the masses ([gCH_2O/m^2]) for leaf, stem, root, and grain are further updated:

$$M_{leaf} = M_{leaf} - C_{leaf2grain} \quad \text{(Eq. 3.8.132)}$$

$$M_{stem} = M_{stem} - C_{stem2grain} \quad \text{(Eq. 3.8.133)}$$

$$M_{root} = M_{root} - C_{root2grain} \quad \text{(Eq. 3.8.134)}$$

$$M_{grain} = M_{grain} + C_{leaf2grain} + C_{stem2grain} + C_{root2grain} \quad \text{(Eq. 3.8.135)}$$

If $M_{root} < 0$, then

$$R_{turnover,root} = N_{PP,root} \quad \text{(Eq. 3.8.136)}$$

$$M_{root} = 0 \quad \text{(Eq. 3.8.137)}$$

If $M_{grain} < 0$, then

$$M_{grain} = 0 \quad \text{(Eq. 3.8.138)}$$

(10) Compute soil carbon budget:

The short-lived soil carbon pool (typically in shallow soil layers, $P_{c,fast}$, [gC/m^2]) is:

$$P_{c,fast} = P_{c,fast} + (R_{turnover,leaf} + R_{turnover,stem} + R_{turnover,root} + D_{y,leaf}) \times \Delta t \times \frac{12}{30}$$

(Eq. 3.8.139)

The soil temperature factor for microbial respiration ($F_{resp,micro,temp}$) is:

$$F_{resp,micro,temp} = 2^{\left[\frac{T_{soil}(1)-283.16}{10}\right]} \quad \text{(Eq. 3.8.140)}$$

where $T_{soil}(1)$ is the soil temperature for the first layer.

The soil water factor for microbial respiration ($F_{resp,micro,water}$) is:

$$F_{resp,micro,water} = \frac{0.23 \times W_{rootzone}}{(0.2+W_{rootzone}) \times (0.23+W_{rootzone})} \quad \text{(Eq. 3.8.141)}$$

where $W_{rootzone}$ [m^3/m^3] is the root-zone soil water computed above.

Thus, the soil microbial respiration rate ($R_{S,soil,micro}$, [$gCH_2O/m^2/s$]) is computed as:

$$R_{S,soil,micro} = F_{resp,micro,temp} \times F_{resp,micro,water} \times 30 \times 10^{-6} \times \max(0, P_{c,fast} \times 10^{-3}) \times f_{resp,micro} \quad \text{(Eq. 3.8.142)}$$

where $f_{resp,micro}$ [umol/kgC/s] is the microbial respiration coefficient (set in NoahmpTable.TBL).

The rate for short-lived carbon decay to long-lived/stable carbon ($R_{c,decay}$, [$gCH_2O/m^2/s$]) is:

$$R_{c,decay} = 0.1 \times R_{S,soil,micro} \quad \text{(Eq. 3.8.143)}$$

Then, both short-lived and long-lived (stable, typically in deep soil) carbon storage ([gC/m^2]) are updated:

$$P_{c,fast} = P_{c,fast} - (R_{S,soil,micro} + R_{c,decay}) \times \Delta t \times \frac{12}{30} \quad \text{(Eq. 3.8.144)}$$

$$P_{c,stable} = P_{c,stable} + R_{c,decay} \times \Delta t \times \frac{12}{30} \quad \text{(Eq. 3.8.145)}$$

The net carbon flux from land to the atmosphere ($F_{c,land2atmos}$, [$gC/m^2/s$]) is:

$$F_{c,land2atmos} = -C_{assim,tot} + (R_{S,leaf,grow} + R_{S,leaf,maint} + R_{S,stem,grow} + R_{S,stem,maint} + R_{S,root,grow} + R_{S,root,maint} + R_{S,grain,grow} + R_{S,grain,maint} + 0.9 \times R_{S,soil,micro}) \times \frac{12}{30} \quad \text{(Eq. 3.8.146)}$$

(11) Compute diagnostic output variables and update LAI and SAI:

The total gross primary productivity (G_{PP}, [$gC/m^2/s$]) is:

$$G_{PP} = C_{H2O,assim,tot} \times \frac{12}{30} \quad \text{(Eq. 3.8.147)}$$

The total net primary productivity ($N_{PP,tot}$, [$gC/m^2/s$]) is:

$$N_{PP,tot} = (N_{PP,leaf} + N_{PP,stem} + N_{PP,root} + N_{PP,grain}) \times \frac{12}{30} \quad \text{(Eq. 3.8.148)}$$

The total plant respiration (net ecosystem respiration) rate ($R_{S,plant,tot}$, [$gC/m^2/s$]) is:

$$R_{S,plant,tot} = (R_{S,leaf,grow} + R_{S,leaf,maint} + R_{S,stem,grow} + R_{S,stem,maint} + R_{S,root,grow} + R_{S,root,maint} + R_{S,grain,grow} + R_{S,grain,maint}) \times \frac{12}{30} \quad \text{(Eq. 3.8.149)}$$

The total soil (organic) respiration rate ($R_{S,soil,tot}$, [gC/m^2/s]) is:

$$R_{S,soil,tot} = 0.9 \times R_{S,soil,micro} \times \frac{12}{30} \quad \text{(Eq. 3.8.150)}$$

The net ecosystem exchange (N_{EE}, [gCO_2/m^2/s]) is:

$$N_{EE} = (R_{S,plant,tot} + R_{S,soil,tot} - G_{PP}) \times \frac{44}{12} \quad \text{(Eq. 3.8.151)}$$

The total soil carbon storage ($P_{c,soil}$, [gC/m^2]) is:

$$P_{c,soil} = P_{c,fast} + P_{c,stable} \quad \text{(Eq. 3.8.152)}$$

The total living (biomass) carbon storage ($P_{c,live}$, [gC/m^2]) is:

$$P_{c,live} = (M_{leaf} + M_{stem} + M_{root} + M_{grain}) \times \frac{12}{30} \quad \text{(Eq. 3.8.153)}$$

The LAI and SAI are further updated:

$$LAI = \max(LAI_{min}, M_{leaf} \times C_{bio2lai}) \quad \text{(Eq. 3.8.154)}$$

$$SAI = \max(SAI_{min}, M_{stem} \times \frac{3}{1000}) \quad \text{(Eq. 3.8.155)}$$

Further updates for post-harvest stage (if $G_{S,crop} = 8$):

$$LAI = 0.05 \quad \text{(Eq. 3.8.156)}$$

$$SAI = 0.05 \quad \text{(Eq. 3.8.157)}$$

$$M_{leaf} = M_{leaf,min} \quad \text{(Eq. 3.8.158)}$$

$$M_{stem} = M_{stem,min} \quad \text{(Eq. 3.8.159)}$$

$$M_{root} = 0 \quad \text{(Eq. 3.8.160)}$$

$$M_{grain} = 0 \quad \text{(Eq. 3.8.161)}$$

5. Key crop parameter descriptions (for crop parameters in NoahmpTable.TBL**):**

The table below presents an example for corn and soybean parameters listed in NoahmpTable.TBL. The stomata parameters are shaded in green background and dynamic vegetation/crop parameters are shaded in orange background. The parameters that have been modified are highlighted with stars and their values are in bold text. Their references are provided after the table below.

	Subroutine/Parameter	MP default	Liu et al. (2016)		Zhe Zhang	
1	STOMATA	generic crop	corn	soybean	corn	soybean
2	C3PSN (C3=1, C4=0)	1	0	1	0	1
3	KC25	30.0	30.0	30.0	30.0	30.0
4	AKC	2.1	2.1	2.1	2.1	2.1
5	KO25	3.E4	3.E4	3.E4	3.E4	3.E4
6	AKO	1.2	1.2	1.2	1.2	1.2
7	AVCMX	2.4	2.4	2.4	2.4	2.4
8	VCMX25*	80.0	80.0	80.0	**60.0**	80.0
9	BP*	2.E3	2.E3	2.E3	**4.E4**	**1.E4**
10	MP*	9.	9.	9.	**4.**	9.
11	FOLNMX	1.5	1.5	1.5	1.5	1.5
12	QE25*	0.06	0.06	0.06	**0.05**	0.06
13	KP*	4000.*Vcmx	4000.*Vcmx	-	**20000.*Vcmx**	-
14	CARBON/ CARBON_CROP*	generic crop	corn	soybean	corn	soybean
15	C3C4 (C3=1, C4=2)	-	2	1	2	1
16	Q10MR	2.0	2.0	2.0	2.0	2.0
17	FOLN_MX	1.5	1.5	1.5	1.5	1.5
18	Q10MR	2.0	2.0	2.0	2.0	2.0
19	LFMR25(RMF25*)	1.0	1.0	1.0	**0.8**	1.0
20	STMR25(RMS25)	0.1	0.05	0.05	0.05	0.05
21	RTMR25(RMR25)	0.0	0.05	0.05	0.05	0.05
22	GRAINMR25	-	0.0	0.0	0.0	0.0
23	GDD parameters	-	Same as Liu	Same as Liu	Same as Liu	Same as Liu
24	DILE_FC(DILEFC)	0.50	Same …	Same …	Same …	Same …
25	DILF_FW(DILEFW)	0.20	Same …	Same …	Same …	Same …
26	LF_OVRC(LTOVRC)	1.6	Same …	Same …	Same …	Same …
27	ST_OVRC	-	Same …	Same …	Same …	Same …
28	RT_OVRC	-	Same …	Same …	Same …	Same …
29	LFPT	-	Same …	Same …	Same …	Same …
30	STPT	-	Same …	Same …	Same …	Same …
31	RTPT	-	Same …	Same …	Same …	Same …
32	GRAINPT	-	Same …	Same …	Same …	Same …
33	STCT*	-	0.00005	0.00005	**0.**	**0.**
34	RTCT*	-	0.0005	0.0005	**0.**	**0.**

Reference for the parameter table above:

VCMX25*: 60 for corn is adjustable according to Bonan et al. (2011);

BP*: 4.0E4 for corn and 1.0E4 for soybean, are from Sellers et al. (1996);

MP*: 4 for corn, is from Collatz et al. (1992) and Sellers et al. (1996);

KP*: 20000 $\times$ Vcmx for corn, is from CLM4.5 tech note;

LFRM25*: 0.8 for corn is from Collatz et al. (1992);

From table rows #23-32: These parameters have different names in BiochemNatureVegMainMod and BiochemCropMainMod modules. GDD and biomass partition parameters are the same as those in Liu et al. (2016);

STCT and RTCT* are two parameters for translocation of stem and root to grain, which are used in BiochemCropMainMod code but not documented in Liu et al. (2016). They are adjustable.

3.9 Energy and Water Balance Check

Description:

This part is to check the surface energy and water balance for each model timestep. The water balance is for 2-m Noah-MP soil column without the MMF groundwater terms if activated.

Relevant code modules:

Module: BalanceErrorCheckMod.F90

Subroutines: BalanceWaterCheck, BalanceEnergyCheck

Physics:

1. Water balance check (BalanceWaterCheck subroutine):

First, if irrigation is activated for cropland pixels (FlagCropland = true and $f_{irri} \geq f_{irri,thr}$, where f_{irri} is the irrigation fraction for each grid and $f_{irri,thr}$ is the minimum threshold for grid irrigation fraction to activate irrigation), then aggregating total irrigation water to the precipitation (as total input water) for subsequent water balance check:

$$P_{tot} = P_{tot} + (W_{irri,sprinkler} + W_{irri,micro} + W_{irri,flood}) \times \frac{1000}{\Delta t} \quad \text{(Eq. 3.9.1)}$$

where P_{tot} [mm/s] is the total precipitation rate. $W_{irri,sprinkler}$, $W_{irri,micro}$, and $W_{irri,flood}$ [m/timestep] are the water amounts for sprinkler, micro, and flood irrigation per timestep (Δt), respectively.

The water balance is only checked over land points:

The water storage after all Noah-MP processes (W_{end}, [mm]) are computed as:

$$W_{end} = W_{ice,can} + W_{liq,can} + W_{snow} + W_{aquifer} + W_{soil} \quad \text{(Eq. 3.9.2)}$$

$$W_{soil} = \sum_{i=1}^{N_{soil}} [\theta_{soil}(i) \times D_{zsnso}(i) \times 1000] \quad \text{(Eq. 3.9.3)}$$

where $W_{ice,can}$ [mm] is the canopy ice, $W_{liq,can}$ [mm] is the canopy liquid water, W_{snow} [mm] is the snow water equivalent, $W_{aquifer}$ [mm] is the aquifer water storage, W_{soil} [mm] is the soil column water storage, θ_{soil} [m^3/m^3] is the soil moisture for each soil layer, and D_{zsnso} [m] is the thickness for each snow and soil layer.

Thus, the water balance error/residual (W_{error}, [mm]) is computed as:

$$W_{error} = W_{end} - W_{beg} - (P_{tot} - Q_{evap,can} - Q_{transp} - Q_{evap,soil} - R_{sfc} - R_{sub} - Q_{tiledrain}) \times \Delta t \quad \text{(Eq. 3.9.4)}$$

where $Q_{evap,can}$ [mm/s] is the canopy evaporation rate, Q_{transp} [mm/s] is the plant transpiration rate, $Q_{evap,soil}$ [mm/s] is the net soil evaporation rate, R_{sfc} [mm/s] is the surface runoff, R_{sub} [mm/s] is the subsurface runoff, and $Q_{tiledrain}$ [mm/s] is the tile drainage rate. W_{beg} [mm] is the initial water storage before any Noah-MP processes computed at the beginning of the model code (BalanceWaterInit subroutine) as described above (also repeat here):

$$W_{beg} = W_{ice,can} + W_{liq,can} + W_{snow} + W_{aquifer} + W_{soil} \quad \text{(Eq. 3.9.5)}$$

If the water balance error/residual (W_{error}) is greater than 0.1 mm for a model timestep, then Noah-MP will stop and report error information.

2. Energy balance check (BalanceEnergyCheck subroutine):

(1) Check surface shortwave radiation balance error/residual ($E_{error,sw}$, [W/m^2]):

$$E_{error,sw} = S_{W,down} - (S_{W,sfc,abs} + S_{W,sfc,refl}) \quad \text{(Eq. 3.9.6)}$$

where $S_{W,down}$ [W/m^2] is the total downward shortwave radiation (from input forcing), $S_{W,sfc,abs}$ [W/m^2] and $S_{W,sfc,refl}$ [W/m^2] are the total surface absorbed and reflected shortwave radiation, respectively.

If this energy error ($E_{error,sw}$) is greater than 0.01 W/m^2, then Noah-MP will stop and report error information.

(2) Check total surface energy balance error/residual ($E_{error,tot}$, [W/m^2]):

$$E_{error,tot} = S_{W,veg,abs} + S_{W,grd,abs} + H_{pr,tot} - (L_{W,sfc,net} + H_{S,sfc} + H_{L,can} + H_{L,grd} + H_{L,transp} + H_{L,irri,evap} + G_{grd}) \quad \text{(Eq. 3.9.7)}$$

where $S_{W,veg,abs}$ [W/m^2] and $S_{W,grd,abs}$ [W/m^2] are the absorbed shortwave radiation by vegetation and ground (soil/snow), respectively. $H_{pr,tot}$ [W/m^2] is the total heat flux advected from precipitation to land surface. $L_{W,sfc,net}$ [W/m^2] is the total net surface longwave radiation. $H_{S,sfc}$ [W/m^2] is the total surface sensible heat flux. G_{grd} [W/m^2] is the ground heat flux into soil/snow layers. $H_{L,can}$, $H_{L,grd}$, $H_{L,transp}$, and $H_{L,irri,evap}$ [W/m^2] are the latent heat flux caused by canopy evaporation, ground evaporation, plant transpiration, and irrigation water evaporation, respectively.

If this energy error ($E_{error,tot}$) is greater than 0.01 W/m^2, then Noah-MP will stop and report error information.

4. Noah-MP Glacier

Description:

The purpose of the Noah-MP glacier module is to compute the glacier surface water and energy balance and fluxes. Currently, the Noah-MP glacier process in NoahmpMainGlacierMod.F90 is treated in a very similar way to bare grounds in the main Noah-MP non-glacier processes described in the previous sections, such that all the vegetation-related processes have been removed and a special treatment of glacier ice phase change has been added. As a result, the glacier ice layer is defined the same as the soil layer in non-glacier processes. The glacier processes include processing atmospheric forcing, precipitation heat advection, surface energy balance calculation, surface water balance calculation, and finally energy and water balance checks. However, more systematic treatments for glacier ice processes (e.g., Crocus glacier model) are needed in the future.

Relevant code modules:

Module: NoahmpMainGlacierMod.F90

Subroutines: NoahmpMainGlacier

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptGlacierTreatment	1*	include phase change of glacier ice
options for glacier treatment	2	Glacier ice treatment more like original Noah

Physics:

4.1 Atmospheric Forcing Processing for Glacier

Description:

This module is exactly the same as the one used in the main Noah-MP non-glacier column model (Section 3.1), which is to process the input atmospheric forcing variables (surface temperature, precipitation, surface pressure, wind, specific humidity, downward shortwave and longwave radiation) for use in Noah-MP glacier physics.

Relevant code modules:

Module: AtmosForcingMod.F90

Subroutines: ProcessAtmosForcing

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptRainSnowPartition	1*	Jordan (1991) scheme
	2	BATS: when TemperatureAirRefHeight < freezing point+2.2
options for partitioning precipitation into rainfall & snowfall	3	TemperatureAirRefHeight < freezing point
	4	Use WRF microphysics output
	5	Use wetbulb temperature (Wang et al., 2019)

Physics:

Please see Section 3.1 (“Atmospheric forcing processing”) for details.

4.2 Initialization of Key Glacier State Variables

Description:

These two Noah-MP initialization modules are to initialize key state variables, including glacier and snow layer thickness and initial glacier surface water storage. Note that because the glacier ice layers mimic the soil layers of non-glacier grids, most treatments of the glacier ice layer properties (e.g., number of layers, layer thickness and depth, etc.) will be the same as those of the soil layers of non-glacier grids.

Relevant code modules:

Module: GeneralInitGlacierMod.F90

Subroutines: GeneralInitGlacier

Module: BalanceErrorCheckGlacierMod.F90

Subroutines: BalanceWaterInitGlacier

Physics:

1. Initialize snow and glacier ice (equivalent to soil layer in the current Noah-MP glacier setting) i^{th} layer thickness (D_{zsnso}, [m]):

$$D_{zsnso}(i) = \begin{cases} -Z_{snso}(i)\,, & i = -N_{sno} + 1 \\ Z_{snso}(i-1) - Z_{snso}(i)\,, & i = -N_{sno} + 2, \dots, N_{ice} \end{cases} \quad \text{(Eq. 4.2.1)}$$

where $Z_{snso}(i)$ [m] is the depth (negative) of snow or glacier ice i^{th} layer bottom from snow surface, N_{sno} is the actual number of snow layers, and N_{ice} is the number of glacier ice layers. Note that N_{ice} is the same as N_{soil} in Noah-MP currently.

2. Initialize water storage before any Noah-MP processes (W_{beg}, [mm]):

$$W_{beg} = W_{snow} \quad \text{(Eq. 4.2.2)}$$

where W_{snow} [mm] is the snow water equivalent (SWE). Note that for the current Noah-MP glacier treatment, the glacier ice is not accounted for in glacier surface water balance check and hence not included in the initial water storage term.

4.3 Precipitation Heat Advection for Glacier

Description:

This module is to estimate heat flux advected from precipitation (snowfall and rainfall) to glacier ground.

Relevant code modules:

Module: PrecipitationHeatAdvectGlacierMod.F90

Subroutines: PrecipitationHeatAdvectGlacier

Physics:

1. Heat advection from rainfall:

$$H_{pr,air2grd} = P_{rain} \times \frac{C_{wat}}{1000} \times (T_{sfc} - T_{grd}) \quad \text{(Eq.4.3.1)}$$

where $H_{pr,air2grd}$ [W/m^2] is the heat advection from air to glacier ground. P_{rain} [mm/s] is the total liquid precipitation from forcing processed in Sect. 6.1. C_{wat} [J/m^3/K] is the volumetric heat capacity of water (defined in ConstantDefineMod.F90). T_{sfc} [K] and T_{grd} [K] are the temperature for surface air (assumed to equal to precipitation temperature) and ground.

2. Heat advection from snowfall:

$$H_{pr,air2grd} = H_{pr,air2grd} + P_{snow} \times \frac{C_{ice}}{1000} \times (T_{sfc} - T_{grd}) \quad \text{(Eq.4.3.2)}$$

where C_{ice} [J/m^3/K] is the volumetric heat capacity of ice (defined in ConstantDefineMod.F90). P_{snow} [mm/s] is the total liquid precipitation from forcing processed in Sect. 6.1.

3. Net heat advection:

The heat advection from rainfall and snowfall are merged together first and then:

$$H_{pr,baregrd} = H_{pr,air2grd} \quad \text{(Eq.4.3.3)}$$

where $H_{pr,baregrd}$ [W/m^2] is the net precipitation heat advected flux for bare ground. Currently, the net precipitation advection flux (which is typically very small) are limited to -20~20 W/m^2 for the model stability consideration.

4.4 Energy for Glacier Ground

Description:

The Noah-MP glacier energy module is very similar to the energy process for bare grounds in the main Noah-MP non-glacier grids, so that all the vegetation-related processes have been removed (canopy radiative transfer and longwave emission, stomatal resistance and transpiration) and a different treatment of glacier ice phase change is added.

Relevant code modules:

Module: EnergyMainGlacierMod.F90

Subroutines: EnergyMainGlacier

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSnowSoilTempTime options for snow/soil temperature time scheme (only layer 1)	1*	semi-implicit; flux top boundary condition
	2	full implicit (original Noah); temperature top boundary condition
	3	same as 1, but snow cover for skin temperature calculation (generally improves snow)

Physics:

4.4.1 Glacier Snow Cover Fraction

Description:

This module is to compute glacier ground snow cover fraction, which is currently set to 1.0 whenever these is snow on the glacier ice (SWE>0), and is treated differently from the snow cover parameterization (Niu and Yang, 2007) used in the non-glacier grids. Some improvement and unification of the glacier snow cover treatment needs to be done in the future.

Relevant code modules:

Module: SnowCoverGlacierMod.F90

Subroutines: SnowCoverGlacier

Physics:

When there is snow on the glacier ground ($W_{snow} > 0$), then 100% snow cover is assumed:

$$f_{snow} = 1.0 \qquad if \quad W_{snow} > 0 \qquad \text{(Eq. 4.4.1)}$$

where f_{snow} is the snow cover fraction.

4.4.2 Glacier Ground Roughness Property

Description:

This module is to compute the glacier ground roughness length, displacement height, and surface reference height.

Relevant code modules:

Module: GroundRoughnessPropertyGlacierMod.F90

Subroutines: GroundRoughnessPropertyGlacier

Physics:

The glacier ground roughness length for momentum ($z_{0m,grd}$, [m]) is given as

$$z_{0m,grd} = z_{0,snow} \quad \text{(Eq. 4.4.2)}$$

where $z_{0,snow}$ [m] is the snow surface roughness length (set in NoahmpTable.TBL).

The surface roughness length ($z_{0m,sfc}$, [m]) is then given by

$$z_{0m,sfc} = z_{0m,grd} \quad \text{(Eq. 4.4.3)}$$

The zero-plane displacement height for ground ($d_{0,grd}$, [m]) and surface ($d_{0,sfc}$, [m]) is calculated as

$$d_{0,grd} = h_{snow} \quad \text{(Eq. 4.4.4)}$$

$$d_{0,sfc} = d_{0,grd} \quad \text{(Eq. 4.4.5)}$$

where h_{snow} [m] is the snow height.

The surface reference height above ground (z_{ref}, [m]) is calculated as

$$z_{ref} = d_{0,sfc} + z'_a \quad \text{(Eq. 4.4.6)}$$

where z'_a [m] is the reference height above surface zero plane received from atmospheric models or set in the host model namelist.

4.4.3 Glacier Ground Thermal Property

Description:

This module is to estimate thermal conductivity and heat capacity of snow and glacier ice.

Relevant code modules:

Module: GroundThermalPropertyGlacierMod.F90

Subroutines: GroundThermalPropertyGlacier

Submodule: SnowThermalPropertyMod.F90

Subroutines: SnowThermalProperty

Submodule: GlaicerIceThermalPropertyMod.F90

Subroutines: GlacierIceThermalProperty

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSnowThermConduct	1*	Stieglitz(yen,1965) scheme
	2	Anderson, 1976 scheme
options for snow thermal conductivity	3	constant
	4	Verseghy (1991) scheme
	5	Douvill(Yen, 1981) scheme

Physics:

1. Snow thermal properties (SnowThermalPropertyMod.F90**):**

The snow thermal property calculation is the same as the one used for the main Noah-MP non-glacier grids in SnowThermalPropertyMod.F90. Please see the description of the subroutine "SnowThermalProperty" above in the main Noah-MP Energy section.

2. Glacier ice thermal properties (GlaicerIceThermalPropertyMod.F90**):**

Thermal properties of glacier ice are estimated based on the old Noah empirical parameterizations. The heat capacity of glacier ice ($C_{p,gla}$, [J/m^3/K]) is calculated as:

$$C_{p,gla}(i) = 10^6 \times (0.8194 + 0.1309 \times Z_{mid}(i)) \qquad \text{(Eq. 4.4.7)}$$

where Z_{mid} [m] is the depth of the middle point of the i^{th} glacier ice layer.

The thermal conductivity of glacier ice (k_{gla}, [W/m/K]) is calculated as:

$$k_{gla}(i) = 0.32333 + (0.10073 \times Z_{mid}(i)) \qquad \text{(Eq. 4.4.8)}$$

3. Glacier ice and snow interface:

A temporary phase change factor used for melting/freezing of snow and glacier ice is computed as:

$$f_{phase}(i) = \frac{\Delta t}{C_{p,gla/sno}(i) \times D_{zsnso}(i)} \quad \text{(Eq. 4.4.9)}$$

where i is the snow and glacier ice layer index, $C_{p,gla/sno}$ [J/m^3/K] is the heat capacity of snow or glacier ice computed above, and D_{zsnso} [m] is the thickness for each snow and glacier ice layer.

The thermal conductivity of snow and glacier ice interface ($k_{sno/gla}(i = 1)$) is estimated as:

When there is explicit snow layer on the glacier ice ($I_{n,sno} < 0$):

$$k_{sno/gla}(i = 1) = \frac{k_{gla}(1) \times D_{zsnso}(1) + k_{snow}(0) \times D_{zsnso}(0)}{D_{zsnso}(1) + D_{zsnso}(0)} \quad \text{(Eq. 4.4.10)}$$

When there is no explicit snow layer on the glacier ice ($I_{n,sno} = 0$):

$$k_{sno/gla}(i = 1) = \frac{k_{gla}(1) \times D_{zsnso}(1) + 0.35 \times h_{snow}}{D_{zsnso}(1) + h_{snow}} \quad \text{(Eq. 4.4.11)}$$

where $k_{snow}(0)$ [W/m/K] is the thermal conductivity for the bottom snow layer, $k_{gla}(1)$ [W/m/K] is the thermal conductivity for the top glacier ice layer, h_{snow} [m] is the snow height, and $I_{n,sno}$ is the snow layer index (see the snow section for details above).

4.4.4 Glacier Surface Albedo

Description:

This module is to estimate glacier surface albedo, which is similar to the main non-glacier Noah-MP process except that the soil albedo treatment is replaced by land ice albedo and there is no canopy two stream radiative transfer.

Relevant code modules:

Module: SurfaceAlbedoGlacierMod.F90

Subroutines: SurfaceAlbedoGlacier

Submodule: SnowAgingBatsMod.F90

Subroutines: SnowAgingBats

Submodule: SnowAlbedoBatsMod.F90

Subroutines: SnowAlbedoBats

Submodule: SnowAlbedoClassMod.F90

Subroutines: SnowAlbedoClass

Submodule: GroundAlbedoGlacierMod.F90

Subroutines: GroundAlbedoGlacier

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptSnowAlbedo options for ground snow surface albedo	1*	BATS snow albedo scheme
	2	CLASS snow albedo scheme

Physics:

When there is no sunlight (i.e., solar zenith angle is 90deg or larger), all the snow and ground albedo quantities are set to zero. The albedo calculations are only activated when there is sunlight.

1. Snow albedo and aging (SnowAgingBatsMod.F90, SnowAlbedoBatsMod.F90, and SnowAlbedoClassMod.F90)**:**

The snow albedo (α_{snow}) calculation will be the same as that used for the main Noah-MP non-glacier grids in SnowAgingBatsMod.F90, SnowAlbedoBatsMod.F90, and SnowAlbedoClassMod.F90. Please see the earlier descriptions in this document above for details.

2. Glacier ground albedo (GroundAlbedoGlacierMod.F90):

The glacier ground albedo ($\alpha_{gla,grd}$) is a weighted average of land ice albedo and snow albedo based on snow cover fraction (f_{snow}):

$$\alpha_{gla,grd} = \alpha_{ice} \times (1 - f_{snow}) + \alpha_{snow} \times f_{snow} \quad \text{(Eq. 4.4.12)}$$

where α_{ice} is the land ice albedo specified in NoahmpTable.TBL. Noah-MP accounts for two solar bands (visible and NIR) and two light conditions (diffuse and direct radiation) for snow, ground, and surface albedo calculations.

3. Glacier surface albedo (SurfaceAlbedoGlacierMod.F90):

The glacier surface albedo ($\alpha_{gla,sfc}$) is the same as ground albedo:

$$\alpha_{gla,sfc} = \alpha_{gla,grd} \quad \text{(Eq. 4.4.13)}$$

4.4.5 Glacier Surface Shortwave Radiation

Description:

This module is to estimate glacier surface radiation, which is similar to the main non-glacier Noah-MP process except that no canopy-related radiative fluxes are accounted for.

Relevant code modules:

Module: SurfaceRadiationGlacierMod.F90

Subroutines: SurfaceRadiationGlacier

Physics:

The total shortwave radiation absorbed by glacier ground ($S_{W,grd,abs}$, [W/m^2]) and surface ($S_{W,sfc,abs}$, [W/m^2]) are computed as:

$$S_{W,grd,abs} = \sum_{j=dif,dir} \sum_{i=vis,nir} \left[S_{W,down,j}(i) \times \left(1 - \alpha_{gla,grd,j}(i)\right)\right] \qquad \text{(Eq. 4.4.14)}$$

$$S_{W,sfc,abs} = S_{W,grd,abs} \qquad \text{(Eq. 4.4.15)}$$

The total shortwave radiation reflected by glacier ground ($S_{W,grd,refl}$, [W/m^2]) and surface ($S_{W,sfc,refl}$, [W/m^2]) are computed as:

$$S_{W,grd,refl} = \sum_{j=dif,dir} \sum_{i=vis,nir} \left[S_{W,down,j}(i) \times \alpha_{gla,grd,j}(i)\right] \qquad \text{(Eq. 4.4.16)}$$

$$S_{W,sfc,refl} = S_{W,grd,refl} \qquad \text{(Eq. 4.4.17)}$$

4.4.6 Glacier Surface Longwave Emissivity

Description:

This module is to estimate glacier surface longwave radiation emissivity, which is similar to the main non-glacier Noah-MP process except that no canopy emissivity is included.

Relevant code modules:

Module: SurfaceEmissivityGlacierMod.F90

Subroutines: SurfaceEmissivityGlacier

Physics:

The effective glacier ground (ε_{grd}) and surface emissivity (ε_{sfc}) are computed as a weighted average of snow emissivity (ε_{snow}) and land ice emissivity (ε_{ice}) based on snow cover fraction (f_{snow}):

$$\varepsilon_{grd} = \varepsilon_{ice} \times (1 - f_{snow}) + \varepsilon_{snow} \times f_{snow} \qquad \text{(Eq. 4.4.18)}$$

$$\varepsilon_{sfc} = \varepsilon_{grd} \qquad \text{(Eq. 4.4.19)}$$

where ε_{snow} and ε_{ice} are parameters set in the NoahmpTable.TBL.

4.4.7 Glacier Surface Resistance for Ground Evaporation

Description:

This module is to estimate glacier surface resistance to ground evaporation and sublimation, which represents the resistance imposed during the molecular diffusion by the surface (as opposed to aerodynamic resistance).

Relevant code modules:

Module: ResistanceGroundEvaporationGlacierMod.F90

Subroutines: ResistanceGroundEvaporationGlacier

Physics:

The glacier ground resistance to evaporation/sublimation ($R_{grd,evap}$) is set to 1.0.

The relative humidity in surface glacier/snow air space ($RH_{grd,gla}$) is set to 1.0.

4.4.8 Glacier Ground Psychrometric Variables

Description:

This module is to estimate psychrometric variables for glacier ground.

Relevant code modules:

Module: PsychrometricVariableGlacierMod.F90

Subroutines: PsychrometricVariableGlacier

Physics:

The latent heat of vaporization for the glacier ground ($C_{LH,grd,vap}$, [J/kg]) is:

$$C_{LH,grd,vap} = C_{LH,subl} \quad \text{(Eq. 4.4.20)}$$

where $C_{LH,subl}$ [J/kg] is the specific latent heat of sublimation (defined in ConstantDefineMod.F90).

The psychrometric constant for the glacier ground (γ_{grd}, [Pa/K]) is:

$$\gamma_{grd} = \frac{C_{p,air} \times P_{sfc}}{0.622 \times C_{LH,grd,vap}} \quad \text{(Eq. 4.4.21)}$$

where $C_{p,air}$ [J/kg/K] is the air heat capacity (defined in ConstantDefineMod.F90) and P_{sfc} [Pa] is the surface air pressure.

4.4.9 Glacier Ground Energy Flux and Temperature

Description:

This module is to compute glacier surface energy fluxes and balance as well as update ground temperature. The calculation is very similar to the bare ground energy computation for non-glacier Noah-MP grids (SurfaceEnergyFluxBareGroundMod.F90).

Relevant code modules:

Module: SurfaceEnergyFluxGlacierMod.F90

Subroutines: SurfaceEnergyFluxGlacier

Module: VaporPressureSaturationMod.F90

Subroutines: VaporPressureSaturation

Module: ResistanceBareGroundMostMod.F90

Subroutines: ResistanceBareGroundMOST

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptGlacierTreatment options for glacier treatment	1*	include phase change of glacier ice
	2	Glacier ice treatment more like original Noah

Physics:

The majority of the calculations are exactly the same as those of bare ground energy flux and balance calculations in SurfaceEnergyFluxBareGroundMod.F90 (see above for details). The following are several differences that are specific to glacier ground treatments.

(1) Only one surface resistance scheme, Monin-Obukhov Similarity Theory (MOST), is used for glacier, as opposed to the two options (MOST and Chen et al. (1997) schemes) for the non-glacier grids.

(2) If snow depth is zero (i.e., no snowpack on glacier ice) and OptGlacierTreatment = 2 (i.e., original Noah type of glacier ice treatment), then the ground latent heat coefficient is set to 0, which means Noah-MP does not allow any sublimation of glacier ice in OptGlacierTreatment = 2.

4.4.10 Grid Mean Glacier Energy Quantities

Description:

This part is to compute the grid-level mean energy fluxes and temperatures for glacier surfaces, which are treated the same as the bare ground in the non-glacier Noah-MP processes.

Relevant code modules:

Module: EnergyMainGlacierMod.F90

Subroutines: EnergyMainGlacier

Physics:

The grid-level mean energy quantities are assigned values of those for glacier bare ground, in the same way as the bare ground treatment for non-glacier processes in EnergyMainMod.F90.

4.4.11 Glacier Surface Longwave Emission and Radiative Temperature

Description:

This part is to compute the glacier surface total emitted longwave radiation and radiative/skin temperature based on the glacier surface emissivity computed in Sect. 6.4.6.

Relevant code modules:

Module: EnergyMainGlacierMod.F90

Subroutines: EnergyMainGlacier

Physics:

The glacier surface emitted longwave radiation and radiative/skin temperature are computed in the same way as the non-glacier Noah-MP grid calculations (see EnergyMainMod.F90 for details), except that the glacier surface emissivity is computed without canopy effects in Sect. 6.4.6.

4.4.12 Glacier Snow and Ice Temperature Computation

Description:

This part is to compute the snow (if it exists) and glacier ice temperatures. Note that snow temperatures during melting seasons may exceed melting point but later in GlacierPhaseChangeMod.F90, the snow temperatures are reset to melting point for melting snow. The calculations of glacier surface snow and ice temperatures are treated in the same way as those of non-glacier grid calculations (SoilSnowTemperatureMainMod.F90).

Relevant code modules:

Module: GlacierTemperatureMainMod.F90

Subroutines: GlacierTemperatureMain

Module: GlacierTemperatureSolverMod.F90

Subroutines: GlacierTemperatureSolver

Module: GlacierThermalDiffusionMod.F90

Subroutines: GlacierThermalDiffusion

Physics:

First, the solar penetration through snow and glacier ice are ignored, which needs more work.

1. Glacier thermal diffusion calculation (GlacierThermalDiffusionMod.F90):

This module is to calculate the right-hand side of the time tendency term of the glacier ice and snow thermal diffusion equation. Currently, snow and glacier ice layers are coupled in solving the equations. This module also computes/prepares the matrix coefficients for the tri-diagonal matrix of the implicit time scheme.

This GlacierThermalDiffusionMod module is the same as the SoilSnowThermalDiffusionMod module (see above for details), except soil layers are replaced with glacier ice layers and using the glacier ice thermal properties (computed in GroundThermalPropertyGlacierMod.F90 above) to replace soil thermal properties.

2. Glacier temperature solver (GlacierTemperatureSolverMod.F90):

This module is to compute glacier snow and ice layer temperatures using a tri-diagonal matrix solution dependent on the output from GlacierThermalDiffusionMod.F90 above. This GlacierTemperatureSolverMod module is the same as the SoilSnowTemperatureSolverMod module (see above for details), except treating soil layers as glacier ice layers.

4.4.13 Glacier Surface Temperature Adjustments for Snow Condition

Description:

This part is to adjust surface temperature based on snow conditions for OptSnowSoilTempTime = 2. This is treated the same as for non-glacier grid calculations except that the canopy temperature adjustment is removed.

Relevant code modules:

Module: EnergyMainGlacierMod.F90

Subroutines: EnergyMainGlacier

Physics:

For OptSnowSoilTempTime = 2, if there is snowpack exist and the ground temperature is above freezing point, then both the glacier ground and surface temperatures are reset to the freezing point.

4.4.14 Glacier Phase Change

Description:

This part is to compute the phase change (melting/freezing) of snow and glacier ice. The snowpack phase change is treated the same way as that of non-glacier grids (SoilSnowWaterPhaseChangeMod.F90), whereas some specific treatments for glacier ice phase changes have been added.

Relevant code modules:

Module: GlacierPhaseChangeMod.F90

Subroutines: GlacierPhaseChange

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptGlacierTreatment options for glacier treatment	1*	include phase change of glacier ice
	2	Glacier ice treatment more like original Noah

Physics:

1. Snowpack phase change above glacier ice:

The snowpack phase change is treated first in the same way as that of non-glacier grids in SoilSnowWaterPhaseChangeMod.F90.

For the melting/freezing of snow without an explicit layer, if OptGlacierTreatment = 2, then the snow melting/freezing treatment is the same as that of non-glacier grids in SoilSnowWaterPhaseChangeMod.F90. If OptGlacierTreatment = 1, the snow melting/freezing treatment is still the same but in the place of glacier ice treatment because very shallow snowpack melting/freezing may affect the top glacier ice layer in contact with this shallow snowpack for OptGlacierTreatment = 1.

2. Glacier ice phase change:

For OptGlacierTreatment = 2, there is no glacier ice phase change treatment.

For OptGlacierTreatment = 1, the glacier ice phase change is treated as follows:

(1) The liquid water ($W_{gla,liq}$, [mm]) and ice ($W_{gla,ice}$, [mm]) mass for each glacier ice layer are computed before melting/freezing:

$$W_{gla,liq}(i) = W_{liq,gla}(i) \times D_{zsnso}(i) \times 1000 \qquad \text{(Eq. 4.4.22)}$$

$$W_{gla,ice}(i) = (1 - W_{liq,gla}(i)) \times D_{zsnso}(i) \times 1000 \qquad \text{(Eq. 4.4.23)}$$

where i starts from 1 to N_{ice} (i.e., N_{soil}; note that currently the Noah-MP glacier model simply replaces soil layers with glacier ice layers), and D_{zsnso} is the thickness for each snow and glacier ice layer.

(2) Determine melting or freezing states (similar to snowpack treatment):

When $W_{gla,ice} > 0$ and $T_{gla} \geq T_{frz}$, the glacier ice melts ($I_{melt} = 1$).

When $W_{gla,liq} > 0$ and $T_{gla} < T_{frz}$, the glacier liquid water freezes ($I_{melt} = 2$).

where I_{melt} is the snow melt indicator.

(3) Compute energy residual (surplus or loss; $E_{gla,res}$, [W/m^2]) for melting or freezing (similar to snowpack treatment):

$$E_{gla,res} = \frac{T_{gla} - T_{frz}}{f_{phase}} \qquad \text{(Eq. 4.4.24)}$$

where f_{phase} is the phase change factor for glacier computed in the above module "GroundThermalPropertyGlacierMod.F90".

If $I_{melt} = 1$ and $E_{gla,res} < 0$, then both quantities are reset to 0.

If $I_{melt} = 2$ and $E_{gla,res} > 0$, then both quantities are reset to 0.

The potential water mass change (ΔW_{gla}, [mm]) is:

$$\Delta W_{gla} = \frac{E_{gla,res} \times \Delta t}{C_{LH,fus}} \qquad \text{(Eq. 4.4.25)}$$

where $C_{LH,fus}$ [J/kg] is the specific latent heat of fusion of water (defined in ConstantDefineMod.F90).

(4) Treatment of snow without an explicit layer: see part 1 above.

(5) Compute the rate of melting/freezing for glacier ice layers:

This part is the same as the treatment for snowpack by looping from the top glacier ice layer to the bottom ice layer to compute the melted/frozen water mass, energy residuals, and layer temperatures based on the melting/freezing state and ice/liquid water mass for each glacier ice layer.

(6) Adjust glacier ice melting/freezing state and temperature for additional energy residuals:

First, if there are still two or more glacier ice layers with temperatures both above and below the freezing point (i.e., temperature imbalance among different glacier ice layers), then further loop from the top to bottom glacier ice layers to reduce temperature for the layer ("Layer Warm") with temperature above freezing point and move excess heat (E_{res}) to the layer ("Layer Cold") with temperature below freezing point. If the "Layer Cold" is cold enough to absorb all the E_{res}, then the "Layer Cold" temperature is increased according to the E_{res} and the E_{res} is reset to 0. If the "Layer Cold" is not cold enough to absorb all the E_{res}, then the "Layer Cold" temperature is set to freezing point, and the E_{res} is further updated by accounting for the "Layer Cold" temperature increase. The "Layer Warm" temperature is updated based on the updated E_{res}.

Then, similar to above, the cold layer temperature is increased to remove the temperature imbalance among different glacier ice layers.

Finally, after the above excess heat redistribution among different glacier ice layers, if there is still a layer with temperature above freezing point and another layer with ice mass larger than zero, then the excess heat that layer has above freezing point will be redistributed to another layer and contribute to melting the ice. Both layer temperatures will be updated accordingly based on whether the ice melting is enough to absorb all the excess heat. This treatment is similar to the case where there is still a layer with temperature below freezing point and another layer with liquid water mass larger than zero. In this case, the excess cold from this layer is redistributed to freeze the liquid water in another layer.

3. Update snow and glacier ice content:

Based on the melting and freezing treatments in the above two parts, the snowpack and glacier ice and liquid water content are updated. Particularly, for OptGlacierTreatment = 2, the total glacier moisture is always set to 1.0 and glacier liquid water content is always set to 0, which assumes the glacier ice is frozen forever and there is no glacier ice phase change. For OptGlacierTreatment = 1, the total glacier moisture is always set to 1.0 but it accounts for the glacier liquid water within ice layers as described in part 2 above (i.e., allow glacier ice to melt).

4.5 Hydrology for Glacier

Description:

The Noah-MP glacier water (hydrology) module calculates snowpack hydrology, glacier ice water, and surface and subsurface runoff. The treatments are very similar to the non-glacier Noah-MP hydrological processes show above (WaterMainMod.F90).

Relevant code modules:

Module: WaterMainGlacierMod.F90

Subroutines: WaterMainGlacier

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptGlacierTreatment options for glacier treatment	1*	include phase change of glacier ice
	2	Glacier ice treatment more like original Noah

Physics:

4.5.1 Glacier (snow/ice) Surface Water Fluxes

Description:

This part is to compute the ground (snow/ice) surface sublimation, frost, evaporation, and dew water fluxes, which will be used in the subsequent snowpack and glacier ice hydrological processes.

Relevant code modules:

Module: WaterMainGlacierMod.F90

Subroutines: WaterMainGlacier

Physics:

This part is the same as the treatment for the non-glacier model grids (see WaterMainMod.F90). The ground surface evaporation/sublimation and dew/frost rates are computed using the energy flux from the glacier energy component:

$$Q_{vap} = \max(0.0, \frac{H_{L,grd}}{C_{LH,vap}}) \qquad \text{(Eq. 4.5.1)}$$

$$Q_{cond} = abs[\min\left(0.0, \frac{H_{L,grd}}{C_{LH,vap}}\right)] \qquad \text{(Eq. 4.5.2)}$$

$$Q_{net,grd} = Q_{vap} - Q_{cond} \qquad \text{(Eq. 4.5.3)}$$

where Q_{vap}, Q_{cond}, and $Q_{net,grd}$ [mm/s] are the vaporization (evaporation/sublimation), condensation (dew/frost), and net water fluxes at the ground surface. $H_{L,grd}$ [W/m^2] is the glacier surface latent heat fluxes (positive for vaporization and negative for condensation; computed in EnergyMainGlacierMod.F90) and $C_{LH,vap}$ [J/kg] is the specific latent heat of vaporization (evaporation/sublimation) at the glacier surface (computed in PsychrometricVariableGlacierMod.F90).

The total sublimation and frost rates for snow and surface ice:

$$Q_{sno/ice,subl} = Q_{vap} \qquad \text{(Eq. 4.5.4)}$$

where $Q_{sno/ice,subl}$ [mm/s] is the snow and ice surface sublimation rate.

$$Q_{sno/ice,frost} = Q_{cond} \qquad \text{(Eq. 4.5.5)}$$

where $Q_{sno/ice,frost}$ [mm/s] is the snow and ice surface frost rate.

The snow depth increase rate (Δh_{snow}, [m/s]) due to snowfall is computed as:

$$\Delta h_{snow} = \frac{P_{snow}}{\rho_{snow}} \qquad \text{(Eq. 4.5.6)}$$

where P_{snow} [mm/s] is the total snowfall rate on the ground (from AtmosForcingMod.F90) and ρ_{snow} [kg/m^3] is the bulk snowfall density (from AtmosForcingMod.F90).

4.5.2 Snow Hydrology for Glacier

Description:

This multi-layer snowpack module is the same as that for non-glacier grid treatments (Please see Section 3.7.3 (SnowWaterMainMod.F90) for details). Specifically, this part is to compute the evolution of snowpack (up to 3 layers) properties, including snow ice and liquid water content, snow thickness, and water flux out of the snowpack bottom. The snow hydrological processes include snowpack compaction, snow layer combination, snow layer division, and snow water state updates. These processes are driven by the energy fluxes and snow temperature changes computed in EnergyMainGlacierMod.F90. The change in snowpack SWE is balanced by the input snowfall and frost and output snowmelt and sublimation. Only snowpack with explicit snow layers (snow depth >= 2.5 cm) will go through snow layer compaction, combination, and division processes.

Relevant code modules:

Module: SnowWaterMainGlacierMod.F90

Subroutine: SnowWaterMainGlacier

Physics:

Overall snowpack water balance (same as non-glacier grid treatment**):**

$$\frac{\Delta W_{snow}}{\Delta t} = Q_{snow} + Q_{sno,frost} - Q_{sno,melt} - Q_{sno,subl} \qquad \text{(Eq. 4.5.7)}$$

where Q_{snow} [mm/s] is the snowfall rate on the snowpack surface, and $Q_{sno,melt}$ [mm/s] is the total melting snow water driven by snowpack surface energy balance:

$$Q_{sno,melt} = \frac{1}{C_{LH,fus}}(S_{W,grd,net} + L_{W,grd,net} + H_{pr,grd} - H_{L,grd} - H_{L,grd} - G_{sno2gla})$$

where $C_{LH,fus}$ [J/kg] is the specific latent heat of fusion. $S_{W,grd,net}$, $L_{W,grd,net}$, $H_{pr,grd}$, $H_{L,grd}$, and $H_{S,grd}$ [W/m^2] are the net shortwave radiation flux, net longwave radiation flux, precipitation advected heat flux, latent heat flux, and sensible heat flux at the snowpack surface. $G_{sno2gla}$ [W/m^2] is the heat flux advected from snowpack to glacier ice at the snow-ice interface. Note that $G_{sno2gla}$ is only implicitly expressed in Noah-MP due to the use of a coupled snow and glacier ice layer temperature solver based on a tri-diagonal matrix solution. Currently, Noah-MP neglects heat transfer by movement of meltwater within the snowpack.

The Noah-MP code convention for snow layer index is as follows: the snow layer index for vector/matrix dimension starts from -N_{snow}+1 to 0, where N_{snow} is the maximum number of snow layers (currently is 3 in Noah-MP). The snow vector quantities have the column dimension of -2:0 with -2 for top layer index and 0 for bottom layer index. The purpose of this negative indexing of snow vector column dimension is to have a continuous index for combined snow and glacier ice layers (i.e., glacier ice layer index starts from 1 to 4 for 4-layer ice, same as soil treatment for non-glacier grids). There is another snow layer indicator ($I_{n,sno}$) in Noah-MP to identify the actual number of snow layers:

$$I_{n,sno} = \begin{cases} -3, & 3\ layer\ snow \\ -2, & 2\ layer\ snow \\ -1, & 1\ layer\ snow \\ 0, & 0\ layer\ snow \end{cases} \qquad \text{(Eq. 4.5.8)}$$

Thus, for loops over multiple snow layers, the index for vector column dimension starts at $I_{n,sno} + 1$ (top snow layer) and ends at 0 (bottom snow layer).

Note that for shallow snowpack without explicit snow layers (i.e., snow depth < 2.5cm; $I_{n,sno} = 0$), the prognostic snow state variables are W_{snow} and snow depth (h_{snow}). For snowpack with explicit snow layers (i.e., snow depth >= 2.5cm; $I_{n,sno} < 0$), the prognostic snow state variables are snow layer liquid water and ice content ($W_{ice,sno}(i)$ and $W_{liq,sno}(i)$, i =0~2, is the snow layer index) and snow layer thickness ($D_{zsnso}(i)$, i =0~2, is the snow layer index), while W_{snow} and h_{snow} are diagnostic variables derived from $W_{ice,sno}$ and $W_{liq,sno}$ and D_{zsnso}.

4.5.2.1 Snowfall over Glacier

Description:

This module is to update snow water equivalent (SWE) and snow depth due to snowfall on the glacier ground. The treatment is the same as that for non-glacier snowpack (SnowfallBelowCanopyMod.F90).

Relevant code modules:

Module: SnowfallGlacierMod.F90

Subroutines: SnowfallGlacier

Physics:

The treatment is the same as that for non-glacier snowpack (SnowfallBelowCanopyMod.F90).

4.5.2.2 Snowpack Compaction

Description:

This module is to update snow layer thickness via compaction due to destructive metamorphism, overburden, & melt. This module is only active for snowpack with explicit snow layers (snow depth >= 2.5 cm; $I_{n,sno} < 0$). The treatment is the same as that for non-glacier snowpack (SnowpackCompactionMod.F90).

Relevant code modules:

Module: SnowpackCompactionMod.F90

Subroutine: SnowpackCompaction

Physics:

The treatment is the same as that for non-glacier snowpack (SnowpackCompactionMod.F90).

4.5.2.3 Snow Layer Combination

Description:

This module is to compute snowpack layer combination processes to update snow ice, snow liquid water, snow thickness, and snow temperature. Currently, the minimum thicknesses for each snow layer from the top to the bottom layers are 2.5 cm, 2.5 cm, and 10 cm, respectively. The layer combination will be based on this minimum layer thickness threshold. This module is only active for snowpack with explicit snow layers ($I_{n,sno} < 0$). The treatment is the same as that for non-glacier snowpack (SnowLayerCombineMod.F90).

Relevant code modules:

Module: SnowLayerCombineMod.F90

Subroutine: SnowLayerCombine

Module: SnowLayerWaterComboMod.F90

Subroutine: SnowLayerWaterCombo

Physics:

The treatment is the same as that for non-glacier snowpack (SnowLayerCombineMod.F90).

4.5.2.4 Snow Layer Division

Description:

This module is to compute snowpack layer division processes to update snow ice, snow liquid water, snow thickness, and snow temperature. Currently, the maximum thicknesses for each snow layer from the top to the bottom layers are 5 cm, 20 cm, and no limit, respectively. The layer division will be based on this maximum layer thickness threshold. This module is only active for snowpack with explicit snow layers ($I_{n,sno} < 0$). The treatment is the same as that for non-glacier snowpack (SnowLayerDivideMod.F90).

Relevant code modules:

Module: SnowLayerDivideMod.F90

Subroutine: SnowLayerDivide

Module: SnowLayerWaterComboMod.F90

Subroutine: SnowLayerWaterCombo

Physics:

The treatment is the same as that for non-glacier snowpack (SnowLayerDivideMod.F90).

4.5.2.5 Snowpack Hydrological Process

Description:

This module is to update snow ice and liquid water content and snow height based on snowpack water balance (sublimation, frost, melting) and to compute water fluxes within and out of snowpack. The treatment is similar to that for non-glacier snowpack (SnowpackHydrologyMod.F90), with some slightly different treatments based on glacier ice treatment options (OptGlacierTreatment).

Relevant code modules:

Module: SnowpackHydrologyGlacierMod.F90

Subroutine： SnowpackHydrologyGlacier

Module: SnowLayerCombineMod.F90

Subroutine： SnowLayerCombine

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptGlacierTreatment options for glacier treatment	1*	include phase change of glacier ice
	2	Glacier ice treatment more like original Noah

Physics:

1. If no snow exists anymore (W_{snow}=0) after snow layer combination process (e.g., due to earlier over-sublimation):

(1) For OptGlacierTreatment = 1 (allow glacier ice to melt and sublimate):

The snow surface frost flux ($Q_{sno,frost}$) and sublimation flux ($Q_{sno,subl}$) computed above are then applied to glacier ice:

$$W_{ice,gla}(1) = W_{ice,gla}(1) + \frac{(Q_{sno,frost} - Q_{sno,subl}) \times \Delta t}{D_{zsnso}(1) \times 1000} \quad \text{(Eq. 4.5.9)}$$

(2) For OptGlacierTreatment = 2 (does not allow glacier ice to melt and sublimate):

The latent heat flux for snow surface frost flux ($Q_{sno,frost}$) and sublimation flux ($Q_{sno,subl}$) are assigned to surface sensible heat flux and the frost and sublimation fluxes are set to 0:

$$H_{S,gla} = H_{S,gla} - (Q_{sno,frost} - Q_{sno,subl}) \times C_{LH,subl} \quad \text{(Eq. 4.5.10)}$$

$$Q_{sno,frost} = 0 \quad \text{(Eq. 4.5.11)}$$

$$Q_{sno,subl} = 0 \quad \text{(Eq. 4.5.12)}$$

where $C_{LH,subl}$ [J/kg] is the specific latent heat for ice sublimation.

2. For shallow snow without an explicit snow layer ($I_{n,sno} = 0$ and W_{snow}>0):

For OptGlacierTreatment = 1 (allow glacier ice to melt and sublimate), the treatment is the same as for the non-glacier snowpack (SnowpackHydrologyMod.F90):

$$W_{snow,new} = W_{snow,old} + (Q_{sno,frost} - Q_{sno,subl}) \times \Delta t \quad \text{(Eq. 4.5.13)}$$

$$h_{snow} = \max(0.0, h_{snow} \times \frac{W_{snow,new}}{W_{snow,old}}) \quad \text{(Eq. 4.5.14)}$$

The snow depth is further constrained to a reasonable snowpack density (50~500 kg/m^3):

$$h_{snow} = \min(\max\left(h_{snow}, \frac{W_{snow,new}}{500}\right), \frac{W_{snow,new}}{50}) \quad \text{(Eq. 4.5.15)}$$

For OptGlacierTreatment = 2 (does not allow glacier ice to melt and sublimate):

The latent heat flux for $Q_{sno,frost}$ and $Q_{sno,subl}$ are assigned to surface sensible heat flux and the frost and sublimation fluxes are set to 0:

$$H_{S,gla} = H_{S,gla} - (Q_{sno,frost} - Q_{sno,subl}) \times C_{LH,subl} \quad \text{(Eq. 4.5.16)}$$

$$Q_{sno,frost} = 0 \quad \text{(Eq. 4.5.17)}$$

$$Q_{sno,subl} = 0 \quad \text{(Eq. 4.5.18)}$$

The snow surface sublimation may be larger than existing snow mass (i.e., $W_{snow,new} < 0$). In this case, to conserve water, excessive sublimation is used to reduce glacier ice. Smaller time steps would tend to avoid this problem.

$$W_{ice,gla}(1) = W_{ice,gla}(1) + \frac{W_{snow,new}}{D_{zsnso}(1) \times 1000} \quad \text{(Eq. 4.5.19)}$$

$$W_{snow,new} = 0 \quad \text{(Eq. 4.5.20)}$$

$$h_{snow} = 0 \quad \text{(Eq. 4.5.21)}$$

If $W_{ice,gla}(1) < 0$ (glacier ice is also not sufficient for sublimation), then the remaining water is taken out of top layer glacier liquid water to conserve water:

$$W_{liq,gla}(1) = W_{liq,gla}(1) + W_{ice,gla}(1) \quad \text{(Eq. 4.5.22)}$$

$$W_{ice,gla}(1) = 0 \quad \text{(Eq. 4.5.23)}$$

The above part may need more work since glacier ice change is only allowed for OptGlacierTreatment = 1, but now it is applied to both options.

3. For deep snow with explicit snow layers ($I_{n,sno} < 0$):

This part is the same as the non-glacier grid snowpack treatment (SnowpackHydrologyMod.F90), except that in the calculation of interlayer snow water flux, there is no treatment for snow water retention in the bottom snow layer and maximum allowed snowpack liquid mass fraction. This is because the snow water retention in the bottom snow layer was only added to better simulate streamflow over non-glacier snow basins.

4.5.2.6 Update Snowpack Diagnostic Properties

Description:

This part is to update some key snowpack quantities after all the snowpack processes, including snow thickness and SWE. The treatment is the same as that for non-glacier snowpack (SnowWaterMainMod.F90).

Relevant code modules:

Module: SnowWaterMainGlacierMod.F90

Subroutine： SnowWaterMainGlacier

Physics:

The treatment is the same as that for non-glacier snowpack (SnowWaterMainMod.F90).

4.5.3 Glacier Surface Runoff

Description:

This part is to compute glacier surface runoff based on the water flux balance computed from previous glacier snowpack processes.

Relevant code modules:

Module: WaterMainGlacierMod.F90

Subroutines： WaterMainGlacier

Physics:

First, the ponding water from different snowpack processes above are added to the glacier surface runoff ($R_{sfc,gla}$, [mm/s]):

$$R_{sfc,gla} = \frac{W_{Ponding} + W_{Ponding,1} + W_{Ponding,2}}{\Delta t} \quad \text{(Eq. 4.5.24)}$$

where $W_{Ponding}$, $W_{Ponding,1}$, and $W_{Ponding,2}$ [mm] are surface ponding from different snow water processes above (i.e., ponding water from thin snow melting, thin snow layer combination, and transition from multilayer snow to thin snow, where thin snow means snow without explicit layers).

For non-snowpack case or snowpack without explicit layers ($I_{n,sno} = 0$):

$$R_{sfc,gla} = R_{sfc,gla} + Q_{snobot} + P_{rain} \quad \text{(Eq. 4.5.25)}$$

where Q_{snobot} [mm/s] is the water outflow from snowpack bottom, and P_{rain} [mm/s] is the rain rate on the glacier surface.

For snowpack with explicit layers ($I_{n,sno} < 0$):

$$R_{sfc,gla} = R_{sfc,gla} + Q_{snobot} \quad \text{(Eq. 4.5.26)}$$

Note that P_{rain} is not included for snowpack with explicit layers here, because it is already included in Q_{snobot} when computing snow hydrology process (SnowpackHydrologyMod.F90).

4.5.4 Glacier Subsurface Runoff

Description:

This part is to compute glacier subsurface runoff.

Relevant code modules:

Module: WaterMainGlacierMod.F90

Subroutines： WaterMainGlacier

Relevant Noah-MP namelist options:

Noah-MP Physics	Option	Notes (* indicates the default option)
OptGlacierTreatment	1*	include phase change of glacier ice
options for glacier treatment	2	Glacier ice treatment more like original Noah

Physics:

Overall, the subsurface runoff here is used as a water balancer for the excess glacier water flow (over the maximum snow height) and the sublimation loss.

(1) For OptGlacierTreatment = 1 (allow glacier ice to melt and sublimate):

First, compute the water ($W_{gla,loss}$, [mm/s]) that is required from ice layers below to replace glacier loss due to sublimation:

$$W_{gla,loss} = \sum_{i=1}^{N_{ice}} \left[W_{ice,gla}(i) + W_{liq,gla}(i) - W_{ice,gla,old}(i) - W_{liq,gla,old}(i) \right] \times D_{zsnso}(i) \quad \text{(Eq. 4.5.27)}$$

$$W_{gla,loss} = \frac{W_{gla,loss} \times 1000}{\Delta t} \quad \text{(Eq. 4.5.28)}$$

The glacier ice content is updated as:

$$W_{ice,gla} = \min(1.0, W_{ice,gla,old}) \quad \text{(Eq. 4.5.29)}$$

$$W_{liq,gla} = 1.0 - W_{ice,gla} \quad \text{(Eq. 4.5.30)}$$

Thus, the subsurface runoff is computed as:

$$R_{sub,gla} = Q_{sno,gla} + W_{gla,loss} \quad \text{(Eq. 4.5.31)}$$

where $Q_{sno,gla}$ [mm/s] is the excess glacier water flow computed from snowpack hydrology parts.

(2) For OptGlacierTreatment = 2 (does not allow glacier ice to melt and sublimate):

The glacier ice content is set to always 1.0:

$$W_{ice,gla} = 1.0 \quad \text{(Eq. 4.5.32)}$$

$$W_{liq,gla} = 1.0 - W_{ice,gla} = 0 \quad \text{(Eq. 4.5.33)}$$

Thus, the subsurface runoff is the glacier excess flow:

$$R_{sub,gla} = Q_{sno,gla} \quad \text{(Eq. 4.5.34)}$$

where $Q_{sno,gla}$ [mm/s] is the excess glacier water flow computed from snowpack hydrology parts.

4.6 Glacier Energy and Water Balance Check

Description:

This part is to check the glacier surface energy and water balance for each model timestep. Note that the glacier ice layer is not included in the water and energy balance check.

Relevant code modules:

Module: BalanceErrorCheckGlacierMod.F90

Subroutines: BalanceWaterCheckGlacier, BalanceEnergyCheckGlacier

Physics:

1. Water balance check (BalanceWaterCheckGlacier subroutine):

The water storage after glacier processes (W_{end}, [mm]) is computed as:

$$W_{end} = W_{snow} \quad \text{(Eq. 4.6.1)}$$

Thus, the water balance error/residual (W_{error}, [mm]) is computed as:

$$W_{error} = W_{end} - W_{beg} - (P_{tot} - Q_{evap,gla} - R_{sfc,gla} - R_{sub,gla}) \times \Delta t \quad \text{(Eq. 4.6.2)}$$

Where P_{tot} [mm/s] is the total precipitation, $Q_{evap,gla}$ [mm/s] is the net glacier sublimation/frost rate, $R_{sfc,gla}$ [mm/s] is the surface runoff, and $R_{sub,gla}$ [mm/s] is the subsurface runoff. W_{beg} [mm] is the initial water storage before any glacier processes computed at the beginning of the model code (BalanceWaterInitGlacier subroutine).

If the water balance error/residual (W_{error}) is greater than 0.1 mm for a model timestep, then Noah-MP will stop and report error information.

2. Energy balance check (BalanceEnergyCheckGlacier subroutine):

(1) Check surface shortwave radiation balance error/residual ($E_{error,sw}$, [W/m^2]):

$$E_{error,sw} = S_{W,down} - (S_{W,sfc,abs} + S_{W,sfc,refl}) \quad \text{(Eq. 4.6.3)}$$

where $S_{W,down}$ [W/m^2] is the total downward shortwave radiation (from input forcing), $S_{W,sfc,abs}$ [W/m^2] and $S_{W,sfc,refl}$ [W/m^2] are the total surface absorbed and reflected shortwave radiation, respectively.

If this energy error ($E_{error,sw}$) is greater than 0.01 W/m^2, then Noah-MP will stop and report error information.

(2) Check total surface energy balance error/residual ($E_{error,tot}$, [W/m^2]):

$$E_{error,tot} = S_{W,grd,abs} + H_{pr,tot} - (L_{W,sfc,net} + H_{S,sfc} + H_{L,grd} + G_{gla}) \qquad \text{(Eq. 4.6.4)}$$

where $S_{W,grd,abs}$ [W/m^2] is the absorbed shortwave radiation by glacier ground. $H_{pr,tot}$ [W/m^2] is the total heat flux advected from precipitation to glacier surface. $L_{W,sfc,net}$ [W/m^2] is the total net surface longwave radiation. $H_{S,sfc}$ [W/m^2] is the total surface sensible heat flux. G_{gla} [W/m^2] is the ground heat flux into snow/ice layers. $H_{L,grd}$ [W/m^2] is the latent heat flux caused by ground sublimation/frost.

If this energy error ($E_{error,tot}$) is greater than 0.01 W/m^2, then Noah-MP will stop and report error information.

5. Noah-MP Code Archiving and Contribution

The Noah-MP model code is regularly released to the public at: https://github.com/NCAR/noahmp. This is the official Noah-MP unified GitHub repository for code downloading and contribution. The GitHub above archives the core Noah-MP source code, Noah-MP driver, and parameter tables in a unified Noah-MP GitHub repository.

Note that Noah-MP is a community model, developed through contributions from the entire Noah-MP scientific community. For questions related to code contribution, maintenance, or release of the NCAR GitHub repository above, please contact: Cenlin He (cenlinhe@ucar.edu) and/or Fei Chen (feichen@ucar.edu).

The host/parent models that use Noah-MP as their land component are archived in their respective host model GitHub repositories. At NCAR, we also maintain and release the driver and interface of the coupling between Noah-MP and host models. At NCAR, we also maintain and release the HRLDAS host model (https://github.com/NCAR/hrldas). The HRLDAS and WRF host model GitHub repositories are linked to the unified community Noah-MP GitHub repository (https://github.com/NCAR/noahmp). We recommend host models (e.g., UFS, NWM, LIS, etc.)

establish similar links to the unified community Noah-MP GitHub repository to allow seamless updating and synthesizing of Noah-MP code.

Model developers can make code developments based on the "develop" branch and create a pull request to the "develop" branch. The pull request will be reviewed by the Noah-MP review committee and release team and, if everything looks good, the new code development in pull request will be merged to the "develop" branch. Eventually, new code in the "develop" branch will be merged to the "master" branch during regular updates to the official Noah-MP model release.

Some suggestions for model developers to contribute to the community Noah-MP code through the GitHub repository (typical procedures):

Step (1): Create a fork of the official Noah-MP repository to your own GitHub account;

Step (2): Make code updates based on the "develop" branch of the forked repository under your own account;

Step (3): Finalize and test the code updates you make;

Step (4): Submit a pull request for your model updates from your own forked GitHub repository to the develop branch of the official community Noah-MP GitHub repository;

Step (5): The Noah-MP review committee and release team will review and test the model updates in the submitted pull request and discusses with the developer if there is any problem;

Step (6): The Noah-MP review committee and release team confirms the pull request and merges the updated code to the develop branch in the official Noah-MP GitHub repository;

Step (7): The Noah-MP release team will merge the updated develop branch to the master branch and version-release branch during the regular annual model release.

6. Noah-MP Input, Preprocessing, and Setup

To run Noah-MP offline as a data-driven land surface model, users need to provide several input datasets to drive the model, including the geographical static data (e.g., land type, domain/grid information, topography, vegetation and soil static properties) and atmospheric forcing data (e.g., downward shortwave and longwave radiation, wind speed, specific humidity, surface pressure, surface temperature, and precipitation). The above core Noah-MP model code needs to be embedded in a host/parent model framework to run, e.g., HRLDAS, WRF. NOAA/UFS, NASA/LIS, NWM.

Here, the HRLDAS/Noah-MP offline simulation set up is briefly described.

(1) Create your geo_em file (including geographical static input for your study domain) through WRF/WPS: https://www2.mmm.ucar.edu/wrf/OnLineTutorial/Basics/geogrid.php

(2) Download HLRDAS/Noah-MP code:

git clone --recurse-submodules https://github.com/NCAR/hrldas

(3) Compile HRLDAS/Noah-MP code:

Go to /root_path/hrldas/, type: ./configure (which generates user_build_options file);

Modify the user_build_options file to be consistent with the compiler and related software package version and path in your machine;

type: make all (which does the compilation);

If successfully, there should be two executables created: "HRLDAS_forcing/create_forcing.exe" and "run/hrldas.exe". If you want to re-compile, type "make clean", then "make all".

(4) Prepare atmospheric forcing and initial condition data:

Get the raw forcing data from an external website (e.g., NLDAS, GLDAS, ERA5, etc.);

Extract the needed initial conditions and forcing variables;

Convert the initial conditions and forcing variables to the netcdf format;

Re-grid the initial conditions and forcing data to the study domain and grids;

Several instructions for commonly used forcing data have been provided at https://github.com/NCAR/hrldas, under the /root_path/hrldas/docs/ directory. Particularly, the HRLDAS_forcing directory provides the necessary scripts to pre-process these typically used forcing data.

If successful, this step should generate the initial file: HRLDAS_setup_*.nc file and the forcing files: LDASIN_*datetime*, where *datetime* would be yyyymmddhh.

(5) Run HRLDAS/Noah-MP:

Go to the /root_path/hrldas/run/ directory

Copy and edit the namelist:

cp examples/*/namelist.hrldas.*   namelist.hrldas

edit the namelist for the simulation time period and physics options

then type: ./hrldas.exe

Particularly, for activated irrigation, tile drainage, or crop modeling, users also need to provide specific static input data (e.g., the maps of irrigation fraction and type, crop type, tile drainage fraction). These can be added through the WRF/WPS package. Also, for explicit crop modeling, users need to edit DEFAULT_CROP option in the NoahmpTable.TBL to choose a specific crop type. Users can also modify the parameter values listed in the NoahmpTable.TBL for specific studies.

7. Noah-MP Outputs File Description

The HRLDAS/Noah-MP output file will by default be named as: *datetime*.LDASOUT_DOMAIN1, where *datetime* would be yyyymmddhh. Users can also change the output frequency in the namelist.hrldas. There is also a restart file output: RESTART. *datetime* _DOMAIN1 if users activate the restart file output in the namelist.hrldas.

The specific output variables can be modified in /root_path/hrldas/IO_code/module_NoahMP_hrldas_driver.F. The output variable names and filenames can be different for different host models coupled with Noah-MP.

8. Noah-MP Variable Glossary

Noah-MP model constants

Variable physical meaning/definition	New name	Original name	Variable Type	Unit
gravity acceleration	ConstGravityAcc	GRAV	Real	m/s^2
Stefan-Boltzmann constant	ConstStefanBoltzmann	SB	Real	$W/m^2/K^4$
von Karman constant	ConstVonKarman	VKC	Real	-
freezing/melting temperature point	ConstFreezePoint	TFRZ	Real	K
latent heat of sublimation	ConstLatHeatSublim	HSUB	Real	J/kg
latent heat of evaporization	ConstLatHeatEvap	HVAP	Real	J/kg

latent heat of fusion	ConstLatHeatFusion	HFUS	Real	J/kg
specific heat capacity of water	ConstHeatCapacWater	CWAT	Real	$J/m^3/K$
specific heat capacity of ice	ConstHeatCapacIce	CICE	Real	$J/m^3/K$
specific heat capacity of dry air	ConstHeatCapacAir	CPAIR	Real	$J/m^3/K$
thermal conductivity of water	ConstThermConductWater	TKWAT	Real	W/m/K
thermal conductivity of ice	ConstThermConductIce	TKICE	Real	W/m/K
thermal conductivity of air	ConstThermConductAir	TKAIR	Real	W/m/K
gas constant for dry air	ConstGasDryAir	RAIR	Real	J/kg/K
gas constant for water vapor	ConstGasWaterVapor	RW	Real	J/kg/K
density of water	ConstDensityWater	DENH2O	Real	kg/m^3
density of ice	ConstDensityIce	DENICE	Real	kg/m^3
pi value	ConstPI	PAI	Real	-
graupel bulk density	ConstDensityGraupel	RHO_GRPL	Real	kg/m^3
hail bulk density	ConstDensityHail	RHO_HAIL	Real	kg/m^3

Noah-MP configuration type

Variable physical meaning/definition	New name	Original name	Variable Type	Unit
Namelist				
options for dynamic vegetation	OptDynamicVeg	OPT_DVEG	Integer	-
options for canopy stomatal resistance	OptStomataResistance	OPT_CRS	Integer	-
options for soil moisture factor for stomatal resistance	OptSoilWaterTranspiration	OPT_BTR	Integer	-
options for surface runoff	OptRunoffSurface	OPT_RUNSRF	Integer	-
options for subsurface runoff	OptRunoffSubsurface	OPT_RUNSUB	Integer	-
options for surface layer drag coeff	OptSurfaceDrag	OPT_SFC	Integer	-
options for supercooled liquid water (or ice fraction)	OptSoilSupercoolWater	OPT_FRZ	Integer	-
options for frozen soil permeability	OptSoilPermeabilityFrozen	OPT_INF	Integer	-
options for canopy radiation transfer	OptCanopyRadiationTransfer	OPT_RAD	Integer	-
options for ground snow surface albedo	OptSnowAlbedo	OPT_ALB	Integer	-
options for partitioning precipitation into rainfall & snowfall	OptRainSnowPartition	OPT_SNF	Integer	-
options for lower boundary condition of soil temperature	OptSoilTemperatureBottom	OPT_TBOT	Integer	-
options for snow/soil temperature time scheme (only layer 1)	OptSnowSoilTempTime	OPT_STC	Integer	-
options for surface resistent to evaporation/sublimation	OptGroundResistanceEvap	OPT_RSF	Integer	-

options for defining soil properties	OptSoilProperty	OPT_SOIL	Integer	-
options for pedotransfer functions (used when OPT_SOIL = 3)	OptPedotransfer	OPT_PEDO	Integer	-
options for crop model	OptCropModel	OPT_CROP	Integer	-
options for irrigation	OptIrrigation	OPT_IRR	Integer	-
options for irrigation method	OptIrrigationMethod	OPT_IRRM	Integer	-
options for snow thermal conductivity schemes	OptSnowThermConduct	OPT_TKSNO	Integer	-
options for infiltration in dynamic VIC runoff scheme	OptDynVicInfiltration	OPT_INFDV	Integer	-
options for tile drainage (currently only tested & calibrated to work with opt_run=3)	OptTileDrainage	OPT_TDRN	Integer	-
options for glacier treatment	OptGlacierTreatment	OPT_GLA	Integer	-
Domain				
landuse data name (USGS or MODIS_IGBP)	LandUseDataName	LLANDUSE	String	-
flag to identify croplands	FlagCropland	CROPLU	Logical	-
flag to identify urban points	FlagUrban	URBAN_FLAG	Logical	-
flag to activate dynamic vegetation model	FlagDynamicVeg	DVEG_ACTIVE	Logical	-
flag to activate crop model	FlagDynamicCrop	CROP_ACTIVE	Logical	-
grid index I	GridIndexI	ILOC	Integer	-
grid index J	GridIndexJ	JLOC	Integer	-
vegetation type	VegType	VEGTYP	Integer	-
crop type	CropType	CROPTYP	Integer	-
maximum number of snow layers	NumSnowLayerMax	NSNOW	Integer	-
actual number of snow layers (negative)	NumSnowLayerNeg	ISNOW	Integer	-
number of soil layers	NumSoilLayer	NSOIL	Integer	-
number of shortwave radiation bands	NumSWRadBand	NBAND	Integer	-
Number of days in the particular year	NumDayInYear	YEARLEN	Integer	-
soil type for each soil layer	SoilType	SOILTYP	Integer	-
surface type 1-soil; 2-lake	SurfaceType	IST	Integer	-
soil texture type	SoilColor	SOILCOLOR	Integer	-
indicator for sea-ice and land ice surface/point (1=sea ice; -1=land ice; 0=non-ice)	IndicatorIceSfc	ICE	Integer	-
land type index for water point	IndexWaterPoint	ISWATER	Integer	-
land type index for barren land point	IndexBarrenPoint	ISBARREN	Integer	-
land type index for ice point	IndexIcePoint	ISICE	Integer	-
land type index for cropland point	IndexCropPoint	ISCROP	Integer	-
land type index for evergreen broadleaf (EBL) Forest	IndexEBLForest	EBLFOREST	Integer	-

number of crop growth stages	NumCropGrowStage	NSTAGE	Integer	-
underground runoff slope term	RunoffSlopeType	SLOPETYP	Integer	-
Julian day of year (floating point)	DayJulianInYear	JULIAN	Real	-
latitude	Latitude	LAT	Real	deg
main Noah-MP model time step	MainTimeStep	DT	Real	sec
model grid spacing size	GridSize	DX	Real	m
thickness of lowest atmosphere layer	ThicknessAtmosBotLayer	DZ8W	Real	m
cosine solar zenith angle (0-1)	CosSolarZenithAngle	COSZ	Real	-
layer-bottom depth from soil surface	DepthSoilLayer	ZSOIL	Real	m
Reference Height above surface zero plane (including vegetation)	RefHeightAboveSfc	ZREF	Real	m
depth from soil surface for lower boundary soil temperature forcing	DepthSoilTempBottom	ZBOT	Real	m
soil layer thickness	ThicknessSoilLayer	ZLAYER	Real	m
snow/soil layer thickness	ThicknessSnowSoilLayer	DZSNSO	Real	m
layer-bottom depth from snow surface (positive)	DepthSnowSoilLayer	ZSNSO	Real	m
logical flag to determine if calculating soil processes	FlagSoilProcess	calculate_soil	Logical	-
number of time step for calculating soil processes	NumSoilTimeStep	soil_update_steps	Integer	-
time step for calculating soil processes	SoilTimeStep	dt_soil	Real	sec

Noah-MP forcing type

Variable physical meaning/definition	New name	Original name	Variable Type	Unit
specific humidity (water vapor / moist air) at reference height	SpecHumidityRefHeight	Q2	Real	kg/kg
air temperature at reference height	TemperatureAirRefHeight	SFCTMP	Real	K
wind speed in eastward direction at reference height	WindEastwardRefHeight	UU	Real	m/s
wind speed in northward direction at reference height	WindNorthwardRefHeight	VV	Real	m/s
bottom boundary condition for soil temperature	TemperatureSoilBottom	TBOT	Real	K
downward shortwave radiation at reference height	RadSwDownRefHeight	SOLDN	Real	W/m^2
downward longwave radiation at reference height	RadLwDownRefHeight	LWDN	Real	W/m^2
convective precipitation rate at reference height	PrecipConvRefHeight	PRCPCONV	Real	mm/s

non-convective precipitation rate at reference height	PrecipNonConvRefHeight	PRCPNONC	Real	mm/s
shallow convective precipitation rate at reference height	PrecipShConvRefHeight	PRCPSHCV	Real	mm/s
snow rate at reference height	PrecipSnowRefHeight	PRCPSNOW	Real	mm/s
graupel rate at reference height	PrecipGraupelRefHeight	PRCPGRPL	Real	mm/s
hail rate at reference height	PrecipHailRefHeight	PRCPHAIL	Real	mm/s
air pressure at reference height	PressureAirRefHeight	SFCPRS	Real	Pa
air pressure at the surface-atmosphere interface level	PressureAirSurface	PSFC	Real	Pa

Noah-MP Biochemistry type

Variable physical meaning/definition	New name	Original name	Variable Type	Unit
State				
plant growing stage	PlantGrowStage	PGS	Integer	-
Planting index (0=off, 1=on)	IndexPlanting	IPA	Integer	-
Harvest index (0=on,1=off)	IndexHarvest	IHA	Integer	-
growing season index (0=off, 1=on)	IndexGrowSeason	IGS	Real	-
total soil carbon	CarbonMassSoilTot	TOTSC	Real	gC/m^2
total living carbon	CarbonMassLiveTot	TOTLB	Real	gC/m^2
foliage nitrogen concentration	NitrogenConcFoliage	FOLN	Real	%
leaf mass	LeafMass	LFMASS	Real	$g(CH_2O$ or $C)/m^2$
mass of fine roots	RootMass	RTMASS	Real	$g(CH_2O$ or $C)/m^2$
stem mass	StemMass	STMASS	Real	$g(CH_2O$ or $C)/m^2$
mass of wood (incl. woody roots)	WoodMass	WOOD	Real	$g(CH_2O$ or $C)/m^2$
stable carbon in deep soil	CarbonMassDeepSoil	STBLCP	Real	gC/m^2
short-lived carbon in shallow soil	CarbonMassShallowSoil	FASTCP	Real	gC/m^2
leaf area per unit mass	LeafAreaPerMass	LAPM	Real	m^2/g
stem area per unit mass	StemAreaPerMass	SAPM	Real	m^2/g
minimum leaf mass	LeafMassMin	LFMSMN	Real	$g(CH_2O$ or $C)/m^2$
minimum stem mass	StemMassMin	STMSMN	Real	$g(CH_2O$ or $C)/m^2$
fraction of carbon flux allocated to leaves	CarbonFracToLeaf	LEAFPT	Real	-
fraction of carbon flux allocated to roots	CarbonFracToRoot	ROOTPT	Real	-

fraction of carbon flux allocated to stem	CarbonFracToStem	STEMPT	Real	-
fraction of carbon flux allocated to wood	CarbonFracToWood	WOODPT	Real	-
wood carbon fraction in (root+wood) carbon	WoodCarbonFrac	WOODF	Real	-
fraction of carbon to root and wood	CarbonFracToWoodRoot	NONLEF	Real	-
growing degree days	GrowDegreeDay	GDD	Real	-
mass of GRAIN	GrainMass	GRAIN	Real	gCH_2O/m^2
soil water factor for microbial respiration	MicroRespFactorSoilWater	FSW	Real	-
soil temperature factor for microbial respiration	MicroRespFactorSoilTemp	FST	Real	-
foliage nitrogen adjustemt factor to respiration (<= 1)	RespFacNitrogenFoliage	FNF	Real	-
temperature factor for respiration	RespFacTemperature	TF	Real	-
respiration reduction factor (<= 1)	RespReductionFac	RF	Real	-
Flux				
total photosynthesis [+]	PhotosynTotal	PSN	Real	umol $CO_2/m^2/s$
sunlit leaf photosynthesis	PhotosynLeafSunlit	PSNSUN	Real	umol $CO_2/m^2/s$
shaded leaf photosynthesis	PhotosynLeafShade	PSNSHA	Real	umol $CO_2/m^2/s$
crop photosynthesis rate	PhotosynCrop	PSNCROP	Real	umol $CO_2/m^2/s$
net instantaneous assimilation	GrossPriProduction	GPP	Real	$gC/m^2/s$
net ecosystem exchange	NetEcoExchange	NEE	Real	$gCO_2/m^2/s$
net primary productivity	NetPriProductionTot	NPP	Real	$gC/m^2/s$
leaf net primary productivity	NetPriProductionLeaf	NPPL	Real	$g(CH_2O$ or $C)/m^2/s$
root net primary productivity	NetPriProductionRoot	NPPR	Real	$g(CH_2O$ or $C)/m^2/s$
wood net primary productivity	NetPriProductionWood	NPPW	Real	$g(CH_2O$ or $C)/m^2/s$
stem net primary productivity	NetPriProductionStem	NPPS	Real	$g(CH_2O$ or $C)/m^2/s$
grain net primary productivity	NetPriProductionGrain	NPPG	Real	$gCH_2O/m^2/s$
carbon flux to atmosphere	CarbonToAtmos	CFLUX	Real	$gC/m^2/s$
growth respiration rate for leaf	GrowthRespLeaf	GRLEAF	Real	$g(CH_2O$ or $C)/m^2/s$
growth respiration rate for root	GrowthRespRoot	GRROOT	Real	$g(CH_2O$ or $C)/m^2/s$
growth respiration rate for wood	GrowthRespWood	GRWOOD	Real	$g(CH_2O$ or $C)/m^2/s$
growth respiration rate for stem	GrowthRespStem	GRSTEM	Real	$g(CH_2O$ or $C)/m^2/s$

growth respiration rate for grain	GrowthRespGrain	GRGRAIN	Real	$gCH_2O/m^2/s$
maximum leaf mass available to change	LeafMassMaxChg	LFDEL	Real	$g(CH_2O$ or $C)/m^2/s$
maximum stem mass available to change	StemMassMaxChg	STDEL	Real	$g(CH_2O$ or $C)/m^2/s$
decay rate of fast carbon to slow carbon	CarbonDecayToStable	STABLC	Real	$g(CH_2O$ or $C)/m^2/s$
leaf respiration	RespirationLeaf	RESP	Real	umol $CO_2/m^2/s$
stem respiration	RespirationStem	RSSTEM	Real	$g(CH_2O$ or $C)/m^2/s$
soil respiration per time step	RespirationSoil	RSSOIL	Real	$g(CH_2O$ or $C)/m^2/s$
total plant respiration (leaf, root, stem, wood, grain)	RespirationPlantTot	AUTORS	Real	$gC/m^2/s$
soil heterotrophic (organic) respiration	RespirationSoilOrg	HETERS	Real	$gC/m^2/s$
wood respiration rate	RespirationWood	RSWOOD	Real	$g(CH_2O$ or $C)/m^2/s$
fine root respiration rate	RespirationRoot	RSROOT	Real	$g(CH_2O$ or $C)/m^2/s$
grain respiration rate	RespirationGrain	RSGRAIN	Real	$gCH_2O/m^2/s$
leaf maintenance respiration rate	RespirationLeafMaint	RSLEAF	Real	$g(CH_2O$ or $C)/m^2/s$
root to grain conversion	ConvRootToGrain	RTCONVERT	Real	$gCH_2O/m^2/s$
stem to grain conversion	ConvStemToGrain	STCONVERT	Real	$gCH_2O/m^2/s$
leaf to grain conversion	ConvLeafToGrain	LFCONVERT	Real	$gCH_2O/m^2/s$
leaf turnover rate	TurnoverLeaf	LFTOVR	Real	$g(CH_2O$ or $C)/m^2/s$
stem turnover rate	TurnoverStem	STTOVR	Real	$g(CH_2O$ or $C)/m^2/s$
wood turnover rate	TurnoverWood	WDTOVR	Real	$g(CH_2O$ or $C)/m^2/s$
root turnover rate	TurnoverRoot	RTTOVR	Real	$g(CH_2O$ or $C)/m^2/s$
grain turnover rate	TurnoverGrain	GRTOVR	Real	$gCH_2O/m^2/s$
death rate of leaf mass	DeathLeaf	DIELF	Real	$g(CH_2O$ or $C)/m^2/s$
death rate of stem mass	DeathStem	DIEST	Real	$g(CH_2O$ or $C)/m^2/s$
carbon assimilated rate	CarbonAssim	CARBFX	Real	$gC/m^2/s$
carbohydrate assimilated rate	CarbohydrAssim	CBHYDRAFX	Real	$gCH_2O/m^2/s$
Parameters				
Planting date	DatePlanting	PLTDAY	Integer	-
Harvest date	DateHarvest	HSDAY	Integer	-
Base temperature for GDD accumulation	TempBaseGrowDegDay	GDDTBASE	Real	degC

Maximum temperature for GDD accumulation	TempMaxGrowDegDay	GDDTCUT	Real	degC
growing degree day from seeding to emergence	GrowDegDayEmerg	GDDS1	Real	-
growing degree day from seeding to initial vegetative	GrowDegDayInitVeg	GDDS2	Real	-
growing degree day from seeding to post vegetative	GrowDegDayPostVeg	GDDS3	Real	-
growing degree day from seeding to initial reproductive	GrowDegDayInitReprod	GDDS4	Real	-
growing degree day from seeding to physical maturity	GrowDegDayMature	GDDS5	Real	-
reference maximum CO2 assimilation rate	CarbonAssimRefMax	AREF	Real	$gCO_2/m^2/s$
CO2 assimilation reduction factor(0-1) (caused by non-modeling part,e.g.pest,weeds)	CarbonAssimReducFac	PSNRF	Real	-
Fraction of incoming solar radiation to photosynthetically	PhotosynRadFrac	I2PAR	Real	-
Minimum temperature for CO2 assimilation	TempMinCarbonAssim	TASSIM0	Real	degC
Maximum temperature to which CO2 assimilation linearly increasing	TempMaxCarbonAssim	TASSIM1	Real	degC
CO2 assmilation rate remain at CarbonAssimRefMax until reaching this temperature	TempMaxCarbonAssimMax	TASSIM2	Real	degC
light extinction coefficient	LightExtCoeff	K	Real	-
initial light use efficiency	LighUseEfficiency	EPSI	Real	-
change in maintenance respiration for every 10-deg C temperature change	RespMaintQ10	Q10MR, ARM	Real	-
foliage nitrogen concentration when f(n)=1	NitrogenConcFoliageMax	FOLN_MX, FOLNMX	Real	%
characteristic T for leaf freezing	TemperaureLeafFreeze	LEFREEZ, TDLEF	Real	K
coefficient for temperature leaf stress death for crop	LeafDeathTempCoeffCrop	DILE_FC	Real	1/s
coefficient for water leaf stress death for crop	LeafDeathWaterCoeffCrop	DILE_FW	Real	1/s
coefficient for leaf temperature stress death for generic veg	LeafDeathTempCoeffVeg	DILEFC	Real	1/s
coefficient for leaf water stress death for generic veg	LeafDeathWaterCoeffVeg	DILEFW	Real	1/s
fraction of growth respiration	GrowthRespFrac	FRA_GR, FRAGR	Real	-
leaf turnover coefficient for crop	TurnoverCoeffLeafCrop	LF_OVRC	Real	1/s
leaf turnover coefficient for generic vegetation	TurnoverCoeffLeafVeg	LTOVRC	Real	1/s

stem turnover coefficient for crop	TurnoverCoeffStemCrop	ST_OVRC	Real	1/s
root turnover coefficient for crop	TurnoverCoeffRootCrop	RT_OVRC	Real	1/s
root turnover coefficient for generic vegetation	TurnoverCoeffRootVeg	RTOVRC	Real	1/s
grain maintenance respiration at 25C	RespMaintGrain25C	GRAINMR25	Real	umol CO_2/kg bio/s
fraction of carbohydrate flux to leaf for crop	CarbohydrFracToLeaf	LFPT	Real	-
fraction of carbohydrate flux to stem for crop	CarbohydrFracToStem	STPT	Real	-
fraction of carbohydrate flux to root for crop	CarbohydrFracToRoot	RTPT	Real	-
fraction of carbohydrate flux to grain for crop	CarbohydrFracToGrain	GRAINPT	Real	-
leaf area per living leaf mass	LeafAreaPerBiomass	BIO2LAI	Real	m^2/g
C3 photosynthetic pathway indicator: 0.0 = c4, 1.0 = c3	PhotosynPathC3	C3PSN	Real	-
quantum efficiency at 25c	QuantumEfficiency25C	QE25	Real	umol CO_2 / umol photon
maximum rate of carboxylation at 25c	CarboxylRateMax25C	VCMX25	Real	umol CO_2/m^2/s
change in maximum rate of carboxylation for every 10-deg C temperature change	CarboxylRateMaxQ10	AVCMX	Real	-
slope of conductance-to-photosynthesis relationship	SlopeConductToPhotosyn	MP	Real	-
leaf maintenance respiration at 25C	RespMaintLeaf25C	LFMR25, RMF25	Real	umol CO_2/m^2/s
stem maintenance respiration at 25C	RespMaintStem25C	STMR25, RMS25	Real	umol CO_2 / kgCH_2O/s
root maintenance respiration at 25C	RespMaintRoot25C	RTMR25, RMR25	Real	umol CO_2 /kg bio/s
minimum temperature for photosynthesis	TemperatureMinPhotosyn	TMIN	Real	K
single-side leaf area per mass	LeafAreaPerMass1side	SLA	Real	m^2/kg
wood to non-wood ratio	WoodToRootRatio	WRRAT	Real	-
wood pool index (0~1) depending on woody or not	WoodPoolIndex	WDPOOL	Real	-
microbial respiration coefficient	MicroRespCoeff	MRP	Real	umol CO_2 /kgC/ s
minimum stem area index	StemAreaIndexMin	XSAMIN	Real	m^2/m^2
minimum leaf area index	LeafAreaIndexMin	LAIMIN	Real	m^2/m^2
present wood allocation factor	WoodAllocFac	BF	Real	-
water stress coeficient	WaterStressCoeff	WSTRC	Real	-
wood respiration coeficient	WoodRespCoeff	RSWOODC	Real	1/s
fraction of carbohydrate flux transallocate from leaf to grain	CarbohydrLeafToGrain	LFCT	Real	-

fraction of carbohydrate flux transallocate from stem to grain	CarbohydrStemToGrain	STCT	Real	-
fraction of carbohydrate flux transallocate from root to grain	CarbohydrRootToGrain	RTCT	Real	-

Noah-MP water type

Variable physical meaning/definition	New name	Original name	Variable Type	Unit
State				
wetted or snowed fraction of canopy	CanopyWetFrac	FWET	Real	-
canopy intercepted liquid water	CanopyLiqWater	CANLIQ	Real	mm
canopy intercepted ice	CanopyIce	CANICE	Real	mm
canopy intercepted total water (CANICE+CANLIQ)	CanopyTotalWater	CMC	Real	mm
canopy capacity for snow interception	CanopyIceMax	MAXSNO	Real	mm
canopy capacity for liquid water interception	CanopyLiqWaterMax	MAXLIQ	Real	mm
ice fraction at previous timestep	SnowIceFracPrev	FICEOLD_SNOW	Real	-
ice fraction in snow layers	SnowIceFrac	FICE_SNOW	Real	-
bulk density of snowfall	SnowfallDensity	BDFALL	Real	kg/m^3
snow cover fraction	SnowCoverFrac	FSNO	Real	-
partial volume ice of snow	SnowIceVol	SNICEV	Real	m^3/m^3
partial volume liq of snow	SnowLiqWaterVol	SNLIQV	Real	m^3/m^3
snow effective porosity	SnowEffPorosity	EPORE_SNOW	Real	m^3/m^3
snow layer ice	SnowIce	SNICE	Real	mm
snow layer liquid water	SnowLiqWater	SNLIQ	Real	mm
snow mass at previous time step	SnowWaterEquivPrev	SNEQVO	Real	mm
snow water eqivalent	SnowWaterEquiv	SNEQV	Real	mm
snow depth	SnowDepth	SNOWH	Real	mm
ice fraction in soil layers	SoilIceFrac	FICE_SOIL	Real	-
equilibrium soil water content	SoilMoistureEqui	SMCEQ	Real	m^3/m^3
soil water content between bottom of the soil and water table	SoilMoistureToWT	SMCWTD	Real	m^3/m^3
soil moisture (ice + liq.)	SoilMoisture	SMC	Real	m^3/m^3
maximum soil ice content	SoilIceMax	SICEMAX	Real	m^3/m^3
soil ice content	SoilIce	SICE	Real	m^3/m^3
minimum soil liquid water content	SoilLiqWaterMin	SH2OMIN	Real	m^3/m^3
liquid soil moisture	SoilLiqWater	SH2O	Real	m^3/m^3
soil effective porosity	SoilEffPorosity	EPORE_SOIL	Real	m^3/m^3
supercooled water in soil	SoilSupercoolWater	SUPERCOOL	Real	kg/m^2
saturation excess of the total soil	SoilSaturationExcess	WPLUS	Real	m
soil matric potential	SoilMatPotential	PSI	Real	m

Soil water transpiration factor (0 - 1)	SoilTranspFac	BTRANI	Real	-
accumulated soil water transpiration factor (0 - 1)	SoilTranspFacAcc	BTRAN	Real	-
soil hydraulic/water conductivity	SoilWatConductivity	WCND	Real	m/s
soil water diffusivity	SoilWatDiffusivity	WDF	Real	m^2/s
maximum soil imperviousness fraction	SoilImpervFracMax	FCRMAX	Real	-
imperviousness fraction due to frozen soil	SoilImpervFrac	FCR	Real	-
Fractional Saturated area from soil moisture	SoilSaturateFrac	FSAT	Real	-
root zone soil water	SoilWaterRootZone	WROOT	Real	m^3/m^3
soil water stress	SoilWaterStress	WSTRES	Real	-
water table from WRF-hydro model	WaterTableHydro	WATBLED	Real	m
depth to water table	WaterTableDepth	ZWT	Real	m
lake water storage (can be neg.)	WaterStorageLake	WSLAKE	Real	mm
water storage in aquifer and saturated soil	WaterStorageSoilAqf	WT	Real	mm
water storage in aquifer	WaterStorageAquifer	WA	Real	mm
groundwater recharge to or from the water table when deep	RechargeGwDeepWT	DEEPRECH	Real	m
groundwater recharge to or from the water table when shallow	RechargeGwShallowWT	RECH	Real	m
water balance error	WaterBalanceError	ERRWAT	Real	mm
total water storage at the begining before Noah-MP process	WaterStorageTotBeg	BEG_WB	Real	mm
total water storage at the end of Noah-MP process	WaterStorageTotEnd	END_WB	Real	mm
phase change index [0-none;1-melt;2-refreeze]	IndexPhaseChange	IMELT	Real	-
surface ponding from thin snow liquid during layer combination	PondSfcThinSnwComb	PONDING1	Real	mm
surface ponding from thin snow liquid during transition from multilayer to no layer	PondSfcThinSnwTrans	PONDING2	Real	mm
surface ponding from snowmelt when thin snow has no layer	PondSfcThinSnwMelt	PONDING	Real	mm
fraction of irrigated portion of grid under flood irrigation (0 to 1)	IrrigationFracFlood	FIFAC	Real	-
fraction of irrigated portion of grid under micro irrigation (0 to 1)	IrrigationFracMicro	MIFAC	Real	-
fraction of irrigated portion of grid under sprinkler irrigation (0 to 1)	IrrigationFracSprinkler	SIFAC	Real	-
total input irrigation fraction of a grid	IrrigationFracGrid	IRRFRA	Real	-
flood irrigation water amount	IrrigationAmtFlood	IRAMTFI	Real	m

micro irrigation water amount	IrrigationAmtMicro	IRAMTMI	Real	m
Sprinkler irrigation water amount	IrrigationAmtSprinkler	IRAMTSI	Real	m
irrigation event number, Flood	IrrigationCntFlood	IRCNTFI	Integer	-
irrigation event number, Micro	IrrigationCntMicro	IRCNTMI	Integer	-
irrigation event number, Sprinkler	IrrigationCntSprinkler	IRCNTSI	Integer	-
tile drainage fraction map	TileDrainFrac	TDFRACMP	Real	-
surface water head	WaterHeadSfc	sfcheadrt	Real	mm
fraction of the gridcell that receives precipitation	PrecipAreaFrac	FP	Real	-
fraction of snowfall in total precipitation	FrozenPrecipFrac	FPICE	Real	-
Flux				
snowfall on the ground	SnowfallGround	QSNOW	Real	mm/s
rain at ground surface	RainfallGround	QRAIN	Real	mm/s
net evaporation of canopy intercepted total (liq+ice) water	EvapCanopyNet	ECAN	Real	mm/s
transpiration rate	Transpiration	ETRAN	Real	mm/s
ground (soil/snow) evaporation rate	EvapGroundNet	EDIR	Real	mm/s
surface runoff	RunoffSurface	RUNSRF	Real	mm / dt_soil
baseflow (saturation excess)	RunoffSubsurface	RUNSUB	Real	mm / dt_soil
total water (snowmelt + rain through pack) out of snowpack bottom	SnowBotOutflow	QSNBOT	Real	mm/s
total convective precipitation at reference height	PrecipConvTotRefHeight	QPRECC	Real	mm/s
large-scale precipitation	PrecipLargeSclRefHeight	QPRECL	Real	mm/s
canopy liquid water evaporation rate	EvapCanopyLiq	QEVAC	Real	mm/s
canopy water dew rate	DewCanopyLiq	QDEWC	Real	mm/s
canopy ice frost rate	FrostCanopyIce	QFROC	Real	mm/s
canopy ice sublimation rate	SublimCanopyIce	QSUBC	Real	mm/s
canopy ice melting rate	MeltCanopyIce	QMELTC	Real	mm/s
canopy water refreezing rate	FreezeCanopyLiq	QFRZC	Real	mm/s
snow depth increasing rate due to snowfall	SnowDepthIncr	SNOWHIN	Real	m/s
water input on soil surface	SoilSfcInflow	QINSUR	Real	mm/s
groundwater recharge rate	RechargeGw	QIN	Real	mm/s
groundwater discharge rate	DischargeGw	QDIS	Real	mm/s
rate of compaction of snowpack due to destructive metamorphism/aging	CompactionSnowAging	DDZ1	Real	1/s
rate of compaction of snowpack due to overburden	CompactionSnowBurden	DDZ2	Real	1/s
rate of compaction of snowpack due to melt	CompactionSnowMelt	DDZ3	Real	1/s
snow surface ice frost rate	FrostSnowSfcIce	QSNFRO	Real	mm/s

snow surface ice sublimation rate	SublimSnowSfcIce	QSNSUB	Real	mm/s
soil surface water dew rate	DewSoilSfcLiq	QSDEW	Real	mm/s
soil surface water evaporation	EvapSoilSfcLiq	QSEVA	Real	mm/s
ground vapor condense rate total (dew+frost)	CondenseVapGrd	QDEW	Real	mm/s
ground vaporize rate total (evap+sublim)	VaporizeGrd	QVAP	Real	mm/s
glacier snow excess flow	GlacierExcessFlow	SNOFLOW	Real	mm/s
flood irrigation water rate	IrrigationRateFlood	IRFIRATE	Real	m/time step
micro irrigation water rate	IrrigationRateMicro	IRMIRATE	Real	m/time step
sprinkler irrigation water rate	IrrigationRateSprinkler	IRSIRATE	Real	m/time step
loss rate of irrigation water to evaporation,sprinkler	IrriEvapLossSprinkler	IREVPLOS	Real	m/time step
infiltration rate at surface	SoilInfil	PDDUM	Real	mm/s
soil bottom drainage	DrainSoilBot	QDRAIN	Real	m/s
tile drainage	TileDrain	QTLDRN	Real	mm / dt_soil
transpiration water loss from soil layers	TranspWatLossSoil	ETRANI	Real	mm/s
evaporation of sprinkler irrigation water	EvapIrriSprinkler	EIRR	Real	mm/s
liquid rainfall rate at reference height	RainfallRefHeight	RAIN	Real	mm/s
snowfall rate at reference height	SnowfallRefHeight	SNOW	Real	mm/s
total precipitation at reference height	PrecipTotRefHeight	PRCP	Real	mm/s
canopy interception rate for rain	InterceptCanopyRain	QINTR	Real	mm/s
interception (loading) rate for snowfall	InterceptCanopySnow	QINTS	Real	mm/s
drip rate for rain	DripCanopyRain	QDRIPR	Real	mm/s
drip (unloading) rate for intercepted snow	DripCanopySnow	QDRIPS	Real	mm/s
throughfall for rain	ThroughfallRain	QTHROR	Real	mm/s
throughfall of snowfall	ThroughfallSnow	QTHROS	Real	mm/s
snowmelt rate	MeltGroundSnow	QMELT	Real	mm/s
total surface water vapor flux to atmosphere	WaterToAtmosTotal	QFX	Real	mm/s
rate of total snow compaction	CompactionSnowTot	PDZDTC	Real	fraction/ dt_main
accumulated water flux into soil during soil timestep	SoilSfcInflowAcc	ACC_QINSUR	Real	m/s * dt_soil/ dt_main
accumulated soil surface evaporation during soil timestep	EvapSoilSfcLiqAcc	ACC_QSEVA	Real	m/s * dt_soil/ dt_main

accumualted transpiration water loss from soil layers during soil timestep	TranspWatLossSoilAcc	ACC_ETRANI	Real	m/s * dt_soil/ dt_main
accumulated snow,soil,canopy water change per soil timestep	SfcWaterTotChgAcc	ACC_DWATER	Real	mm / dt_soil
accumulated precipitation per soil timestep	PrecipTotAcc	ACC_PRCP	Real	mm / dt_soil
accumulated net canopy evaporation per soil timestep	EvapCanopyNetAcc	ACC_ECAN	Real	mm / dt_soil
accumulated transpiration per soil timestep	TranspirationAcc	ACC_ETRAN	Real	mm / dt_soil
accumulated net ground (soil/snow) evaporation per soil timestep	EvapGroundNetAcc	ACC_EDIR	Real	mm / dt_soil
mean soil surface evaporation during each soil timestep	EvapSoilSfcLiqMean	QSEVA_avg	Real	m/s
mean water flux into soil during each soil timestep	SoilSfcInflowMean	QINSUR_avg	Real	m/s
mean transpiration water loss from soil layers during soil timestep	TranspWatLossSoilMean	ETRANI_avg	Real	m/s
Parameters				
Reference Soil Conductivity parameter (used in runoff formulation)	SoilConductivityRef	REFDK	Real	-
Reference Soil Infiltration Parameter (used in runoff formulation)	SoilInfilFacRef	REFKDT	Real	-
Parameter in infiltration parameterization	SoilInfilMaxCoeff	KDT	Real	-
parameter to calculate frozen soil impervious fraction	SoilImpervFracCoeff	FRZX	Real	-
Frozen ground parameter to frozen soil impervious fraction	GroundFrzCoeff	FRZK	Real	-
slope index for soil drainage	SoilDrainSlope	SLOPE	Real	-
soil exponent B parameter	SoilExpCoeffB	BEXP	Real	-
dry soil moisture threshold (not used)	SoilMoistureDry	SMCDRY	Real	m^3/m^3
wilting point soil moisture (volumetric)	SoilMoistureWilt	SMCWLT	Real	m^3/m^3
reference soil moisture (field capacity) (volumetric)	SoilMoistureFieldCap	SMCREF	Real	m^3/m^3
porosity, saturated value of soil moisture (volumetric)	SoilMoistureSat	SMCMAX	Real	m^3/m^3
saturated soil matric potential	SoilMatPotentialSat	PSISAT	Real	-
saturated soil hydraulic conductivity	SoilWatConductivitySat	DKSAT	Real	m/s
gridcell mean topgraphic index (global mean)	GridTopoIndex	TIMEAN	Real	-
maximum surface soil saturated fraction (global mean)	SoilSfcSatFracMax	FSATMX	Real	-

liquid water holding capacity for snowpack	SnowLiqHoldCap	SSI	Real	m^3/m^3
new snow mass to fully cover old snow	SnowMassFullCoverOld	SWEMX	Real	mm
maximum intercepted liquid water per unit lai+sai	CanopyLiqHoldCap	CH2OP	Real	mm
snowmelt m parameter	SnowMeltFac	MFSNO	Real	-
overburden snow compaction parameter	SnowCompactBurdenFac	C2_SnowCompact	Real	m^3/kg
snow desctructive metamorphism compaction parameter1	SnowCompactAgingFac1	C3_SnowCompact	Real	1/s
snow desctructive metamorphism compaction parameter2	SnowCompactAgingFac2	C4_SnowCompact	Real	1/K
snow desctructive metamorphism compaction parameter3	SnowCompactAgingFac3	C5_SnowCompact	Real	-
upper Limit on destructive metamorphism compaction	SnowCompactAgingMax	DM_SnowCompact	Real	kg/m^3
snow viscosity coefficient	SnowViscosityCoeff	ETA0_SnowCompact	Real	kg s / m^2
maximum liquid water fraction in snow	SnowLiqFracMax	SNLIQMAXFRAC	Real	-
snowpack water release timescale factor	SnowLiqReleaseFac	SNOW_RET_FAC	Real	1/s
flood irrigation application rate factor	IrriFloodRateFac	FIRTFAC	Real	-
saturated soil hydraulic diffusivity	SoilWatDiffusivitySat	DWSAT	Real	m^2/s
micro irrigation rate	IrriMicroRate	MICIR_RATE	Real	mm/hr
DVIC heterogeniety parameter for infiltration	InfilHeteroDynVic	BBVIC	Real	-
DVIC Mean Capillary Drive for infiltration models	InfilCapillaryDynVic	GDVIC	Real	m
DVIC model infiltration parameter	InfilFacDynVic	BDVIC	Real	-
VIC model infiltration parameter	InfilFacVic	BVIC	Real	-
Tension water distribution inflection parameter	TensionWatDistrInfl	AXAJ	Real	-
Tension water distribution shape parameter	TensionWatDistrShp	BXAJ	Real	-
Free water distribution shape parameter	FreeWatDistrShp	XXAJ	Real	-
soil layer index for tile drainage	DrainSoilLayerInd	DRAIN_LAYER_OPT	Real	-
depth of drain tube from the soil surface for simple scheme	TileDrainTubeDepth	TD_DEPTH	Real	m
Depth of drain for Hooghoudt scheme	TileDrainDepth	TD_DDRAIN	Real	m
drainage coefficent for Hooghoudt scheme	TileDrainCoeff	TD_DCOEF	Real	m/day

drainage coefficient for simple scheme	TileDrainCoeffSp	TD_DC	Real	mm/day
drainage factor for soil moisture	DrainFacSoilWat	TDSMC_FAC	Real	-
Actual depth of tile drainage to impermeable layer form surface	DrainDepthToImperv	TD_ADEPTH	Real	m
distance between two drain tubes or tiles	DrainTubeDist	TD_SPAC	Real	m
effective radius of drain tubes	DrainTubeRadius	TD_RADI	Real	m
depth to impervious layer from drain water level	DrainWatDepToImperv	TD_D	Real	m
multiplication factor to determine lateral hydraulic conductivity	LateralWatCondFac	KLAT_FAC	Real	-
runoff decay factor	RunoffDecayFac	FFF	Real	1/m
baseflow coefficient	BaseflowCoeff	RSBMX	Real	mm/s
specific yield for groundwater scheme option=1	SpecYieldGw	ROUS	Real	-
microprore content (0.0-1.0) for groundwater scheme option=1	MicroPoreContent	CMIC	Real	-
maximum lake water storage	WaterStorageLakeMax	WSLMAX	Real	mm
maximum SWE at glaciers	SnoWatEqvMaxGlacier	SWEMAXGLA	Real	mm
number of soil layers with root present	NumSoilLayerRoot	NROOT	Real	-
snow cover factor	SnowCoverFac	SCFFAC	Real	m
maximum fresh snowfall density	SnowfallDensityMax	SNOWDEN_MAX	Real	kg/m^3
mm/h, sprinkler irrigation rate	IrriSprinklerRate	SPRIR_RATE	Real	mm/hr
irrigation Fraction threshold in a grid	IrriFracThreshold	IRR_FRAC	Real	-
precipitation threshold to stop irrigation trigger	IrriStopPrecipThr	IR_RAIN	Real	mm/hr
number of days before harvest date to stop irrigation	IrriStopDayBfHarvest	IRR_HAR	Real	-
minimum lai to trigger irrigation	IrriTriggerLaiMin	IRR_LAI	Real	-
management allowable deficit (0-1)	SoilWatDeficitAllow	IRR_MAD	Real	-
fraction of flood irrigation loss (0-1)	IrriFloodLossFrac	FILOSS	Real	-
soil matric potential for wilting point	SoilMatPotentialWilt	PSIWLT	Real	m

Noah-MP energy type

Variable physical meaning/definition	New name	Original name	Variable Type	Unit
State				
canopy air temperature	TemperatureCanopyAir	TAH	Real	K
surface radiative temperature	TemperatureRadSfc	TRAD	Real	K

grid mean surface temperature	TemperatureSfc	TS	Real	K
2-m air temperature over vegetated part	TemperatureAir2mVeg	T2MV	Real	K
2-m air temperature over bare ground part	TemperatureAir2mBare	T2MB	Real	K
density of air at reference height	DensityAirRefHeight	RHOAIR	Real	kg/m^3
potential temperature at reference height	TemperaturePotRefHeight	THAIR	Real	K
2-meter air temperature grid mean	TemperatureAir2m	T2M	Real	K
vapor pressure air at reference height	PressureVaporRefHeight	EAIR	Real	Pa
canopy air vapor pressure	PressureVaporCanAir	EAH	Real	Pa
vegetation/canopy temperature	TemperatureCanopy	TV	Real	K
ground temperature vegetated	TemperatureGrdVeg	TGV	Real	K
ground temperature bare ground	TemperatureGrdBare	TGB	Real	K
snow/soil temperature	TemperatureSoilSnow	STC	Real	K
green vegetation fraction [0.0-1.0]	VegFrac	FVEG	Real	-
surface albedo	AlbedoSfc	ALBEDO	Real	-
snow albedo (direct)	AlbedoSnowDir	ALBSND	Real	-
snow albedo (diffuse)	AlbedoSnowDif	ALBSNI	Real	-
sunlit leaf area index	LeafAreaIndSunlit	LAISUN	Real	m^2/m^2
shaded leaf area index	LeafAreaIndShade	LAISHA	Real	m^2/m^2
between canopy gap fraction for beam	GapBtwCanopy	BGAP	Real	-
within canopy gap fraction for beam	GapInCanopy	WGAP	Real	-
wind stress: east-west bare ground	WindStressEwBare	TAUXB	Real	N/m^2
wind stress: north-south bare ground	WindStressNsBare	TAUYB	Real	N/m^2
wind stress: e-w above canopy	WindStressEwVeg	TAUXV	Real	N/m^2
wind stress: n-s above canopy	WindStressNsVeg	TAUYV	Real	N/m^2
wind stress: e-w grid mean	WindStressEwSfc	TAUX	Real	N/m^2
wind stress: n-s grid mean	WindStressNsSfc	TAUY	Real	N/m^2
effective leaf area index, after burying by snow	LeafAreaIndEff	ELAI	Real	m^2/m^2
effective stem area index, after burying by snow	StemAreaIndEff	ESAI	Real	m^2/m^2
leaf area index	LeafAreaIndex	LAI	Real	m^2/m^2
stem area index	StemAreaIndex	SAI	Real	m^2/m^2
frozen canopy identifier (logical) used to define latent heat pathway	FlagFrozenCanopy	FROZEN_CANOPY	Real	-
frozen ground identifier (logical) used to define latent heat pathway	FlagFrozenGround	FROZEN_GROUND	Real	-
Surface emissivity	EmissivitySfc	EMISSI	Real	-
snow layer volumetric specific heat capacity	HeatCapacVolSnow	CVSNO	Real	$J/m^3/K$

soil layer volumetric specific heat capacity	HeatCapacVolSoil	CVSOIL	Real	$J/m^3/K$
snow layer thermal conductivity	ThermConductSnow	TKSNO	Real	W/m/K
soil layer thermal conductivity	ThermConductSoil	TKSOIL	Real	W/m/K
heat capacity for snow and soil layers	HeatCapacSoilSnow	HCPCT	Real	$J/m^3/K$
thermal conductivity for snow and soil layers	ThermConductSoilSnow	DF	Real	W/m/K
phase change energy factor for snow and soil	PhaseChgFacSoilSnow	FACT	Real	-
snow age factor	SnowAgeFac	FAGE	Real	-
non-dimensional snow age	SnowAgeNondim	TAUSS	Real	-
ground albedo (direct beam: vis, nir)	AlbedoGrdDir	ALBGRD	Real	-
ground albedo (diffuse: vis, nir)	AlbedoGrdDif	ALBGRI	Real	-
soil albedo (direct)	AlbedoSoilDir	ALBSOD	Real	-
soil albedo (diffuse)	AlbedoSoilDif	ALBSOI	Real	-
effective one-sided leaf+stem area index = ELAI+ESAI	VegAreaIndEff	VAI	Real	m^2/m^2
leaf/stem reflectance weighted by fraction LAI and SAI	ReflectanceVeg	RHO	Real	-
leaf/stem transmittance weighted by fraction LAI and SAI	TransmittanceVeg	TAU	Real	-
surface albedo (direct)	AlbedoSfcDir	ALBD	Real	-
surface albedo (diffuse)	AlbedoSfcDif	ALBI	Real	-
projected leaf+stem area in solar direction	VegAreaProjDir	GDIR	Real	m^2/m^2
gap fraction for diffue light	GapCanopyDif	KOPEN	Real	-
total gap fraction for direct beam (<=1-shafac)	GapCanopyDir	GAP	Real	-
vegetation emissivity	EmissivityVeg	EMV	Real	-
ground emissivity	EmissivityGrd	EMG	Real	-
snow albedo at last time step (CLASS type)	AlbedoSnowPrev	ALBOLD	Real	-
sunlit fraction of canopy	CanopySunlitFrac	FSUN	Real	-
shaded fraction of canopy	CanopyShadeFrac	FSHA	Real	-
canopy saturation vapor pressure derivative with temperature d(es)/dt at TV	VapPresSatCanTempD	DESTV	Real	Pa/K
below-canopy saturation vapor pressure derivative with temperature d(es)/dt at TG	VapPresSatGrdVegTempD	DESTG	Real	Pa/K
bare ground saturation vapor pressure derivative with temperature d(es)/dt at TG	VapPresSatGrdBareTempD	DESTB	Real	Pa/K
canopy saturation vapor pressure at TV	VapPresSatCanopy	ESTV	Real	Pa

below-canopy saturation vapor pressure at TG	VapPresSatGrdVeg	ESTG	Real	Pa
bare ground saturation vapor pressure at TG	VapPresSatGrdBare	ESTB	Real	Pa
atmospheric co2 partial pressure	PressureAtmosCO2	CO2AIR	Real	Pa
atmospheric o2 partial pressure	PressureAtmosO2	O2AIR	Real	Pa
bulk leaf boundary layer resistance	ResistanceLeafBoundary	RB	Real	s/m
reference height above ground	RefHeightAboveGrd	ZLVL	Real	m^2/m^2
ground temperature	TemperatureGrd	TG	Real	K
sunlit leaf stomatal resistance	ResistanceStomataSunlit	RSSUN	Real	s/m
shaded leaf stomatal resistance	ResistanceStomataShade	RSSHA	Real	s/m
fraction of canopy buried by snow	CanopyFracSnowBury	FB_snow	Real	-
root-zone averaged temperature	TemperatureRootZone	TROOT	Real	K
zero plane displacement bare ground	ZeroPlaneDispGrd	ZPDG	Real	m
zero plane displacement surface	ZeroPlaneDispSfc	ZPD	Real	m
aerodynamic resistance for momentum, between zlvl and zpd+z0, above canopy	ResistanceMomAbvCan	RAMC	Real	s/m
aerodynamic resistance for water vapor, between zlvl and zpd+z0, above canopy	ResistanceLhAbvCan	RAWC	Real	s/m
aerodynamic resistance for sensible heat, between zlvl and zpd+z0, above canopy	ResistanceShAbvCan	RAHC	Real	s/m
aerodynamic resistance for momentum, between heights zpd+z0h and z0hg, below canopy	ResistanceMomUndCan	RAMG	Real	s/m
aerodynamic resistance for sensible heat, between heights zpd+z0h and z0hg, below canopy	ResistanceShUndCan	RAHG	Real	s/m
aerodynamic resistance for water vapor, between heights zpd+z0h and z0hg, below canopy	ResistanceLhUndCan	RAWG	Real	s/m
aerodynamic resistance for momentum, bare ground	ResistanceMomBareGrd	RAMB	Real	s/m
aerodynamic resistance for sensible heat, bare ground	ResistanceShBareGrd	RAHB	Real	s/m
aerodynamic resistance for water vapor, bare ground	ResistanceLhBareGrd	RAWB	Real	s/m
roughness length, momentum, ground	RoughLenMomGrd	Z0MG	Real	m
roughness length, sensible heat, ground, vegetated	RoughLenShVegGrd	Z0HG	Real	m
roughness length, sensible heat, canopy	RoughLenShCanopy	Z0HV	Real	m

roughness length, sensible heat, ground	RoughLenShBareGrd	Z0HB	Real	m
roughness length, momentum, surface	RoughLenMomSfc	Z0M	Real	m
canopy height [note: CanopyHeight >= RoughLenMomGrd]	CanopyHeight	HCAN	Real	m
wind speed at top of canopy	WindSpdCanopyTop	UC	Real	m/s
canopy wind extinction coefficient	WindExtCoeffCanopy	CWPC	Real	-
wind speed at reference height	WindSpdRefHeight	UR	Real	m/s
specific humidity at 2m bare ground	SpecHumidity2mBare	Q2B	Real	kg/kg
specific humidity at 2m vegetated	SpecHumidity2mVeg	Q2V	Real	kg/kg
specific humidity at 2m grid mean	SpecHumidity2m	Q2E	Real	kg/kg
specific humidity at surface	SpecHumiditySfc	QSFC	Real	kg/kg
specific humidity at the surface grid mean	SpecHumiditySfcMean	Q1	Real	kg/kg
roughness length, momentum, surface, sent to coupled atmospheric model	RoughLenMomSfcToAtm	Z0WRF	Real	m
raltive humidity in ground soil/snow air space	RelHumidityGrd	RHSUR	Real	-
ground resistance to soil/snow evaporation/sublimation	ResistanceGrdEvap	RSURF	Real	s/m
depth from snow surface for lower boundary soil temperature forcing	DepthSoilTempBotToSno	ZBOTSNO	Real	m
psychrometric constant, canopy	PsychConstCanopy	GAMMAV	Real	Pa/K
psychrometric constant, ground	PsychConstGrd	GAMMAG	Real	Pa/K
latent heat of vaporization/subli, canopy	LatHeatVapCanopy	LATHEAV	Real	J/kg
latent heat of vaporization/subli, ground	LatHeatVapGrd	LATHEAG	Real	J/kg
friction velocity in vertical direction, bare ground	FrictionVelVertBare	WSTARB	Real	m/s
friction velocity in vertical direction, vegetated above canopy	FrictionVelVertVeg	WSTARV	Real	m/s
friction velocity, bare ground	FrictionVelBare	FVB	Real	m/s
friction velocity, vegetated	FrictionVelVeg	FVV	Real	m/s
exchange coefficient for momentum, above ZPD, vegetated	ExchCoeffMomAbvCan	CMV	Real	m/s
exchange coefficient for sensible heat, 2-m, vegetated (from diagnostic)	ExchCoeffSh2mVeg	CHV2, CAH2	Real	m/s
exchange coefficient for sensible heat, 2-m, vegetated (from MOST scheme)	ExchCoeffSh2mVegMo	CH2V, CH2	Real	m/s

exchange coefficient for sensible heat, above ZPD, vegetated	ExchCoeffShAbvCan	CHV	Real	m/s
exchange coefficient for sensible heat, leaf level	ExchCoeffShLeaf	CHLEAF	Real	m/s
exchange coefficient for sensible heat, under canopy	ExchCoeffShUndCan	CHUC	Real	m/s
exchange coefficient for sensible heat, 2m, bare ground (from MOST scheme)	ExchCoeffSh2mBareMo	CH2B	Real	m/s
exchange coefficient for sensible heat, above ZPD, bare ground	ExchCoeffShBare	CHB	Real	m/s
exchange coefficient for sensible heat, 2m, bare ground (from diagnostic)	ExchCoeffSh2mBare	EHB2	Real	m/s
exchange coefficient for momentum, above ZPD, bare ground	ExchCoeffMomBare	CMB	Real	m/s
exchange coefficient for momentum, surface grid mean	ExchCoeffMomSfc	CM	Real	m/s
exchange coefficient for sensible heat, surface grid mean	ExchCoeffShSfc	CH	Real	m/s
exchange coefficient for latent heat,, canopy air to ZLVL air	ExchCoeffLhAbvCan	CAW	Real	m/s
exchange coefficient for latent heat, ground to canopy air	ExchCoeffLhUndCan	CGW	Real	m/s
exchange coefficient for transpiration latent heat, leaf to canopy air	ExchCoeffLhTransp	CTW	Real	m/s
exchange coefficient for leaf evaporation latent heat, leaf to canopy air	ExchCoeffLhEvap	CEW	Real	m/s
M-O momentum stability correction, above ZPDG bare ground	MoStabCorrMomBare	FMB	Real	-
M-O sensible heat stability correction, above ZPDG bare ground	MoStabCorrShBare	FHB	Real	-
M-O momentum stability correction, 2m, bare ground	MoStabCorrMomBare2m	FM2B	Real	-
M-O sen heat stability correction, 2m, bare ground	MoStabCorrShBare2m	FH2B	Real	-
Monin-Obukhov stability parameter (z/L), above ZPD, bare ground	MoStabParaBare	MOZB	Real	-
Monin-Obukhov stability parameter (2/L), 2m, bare ground	MoStabParaBare2m	MOZ2B	Real	-
Monin-Obukhov length, above ZPD, bare ground	MoLengthBare	MOLB	Real	m

M-O momentum stability correction, above ZPD, vegetated	MoStabCorrMomAbvCan	FMV	Real	-
M-O momentum stability correction, 2m, vegetated	MoStabCorrMomVeg2m	FM2V	Real	-
M-O sen heat stability correction, above ZPD, vegetated	MoStabCorrShAbvCan	FHV	Real	-
M-O sensible heat stability correction, 2m, vegetated	MoStabCorrShVeg2m	FH2V	Real	-
Monin-Obukhov stability parameter (2/L), 2m, vegetated	MoStabParaVeg2m	MOZ2V	Real	-
Monin-Obukhov stability parameter (z/L), surface, vegetated	MoStabParaAbvCan	MOZV	Real	-
Monin-Obukhov length, vegetated	MoLengthAbvCan	MOLV	Real	m
M-O sensible heat stability correction, below canopy ground	MoStabCorrShUndCan	FHG	Real	-
Monin-Obukhov stability parameter, below canopy ground	MoStabParaUndCan	MOZG	Real	-
Monin-Obukhov length, below canopy ground	MoLengthUndCan	MOLG	Real	m
error in surface energy balance	EnergyBalanceError	ERRENG	Real	W/m^2
error in shortwave radiation balance	RadSwBalanceError	ERRSW	Real	W/m^2
glacier ice layer volumetric specific heat	HeatCapacGlaIce	CVGLAICE	Real	$J/m^3/K$
glacier ice thermal conductivity	ThermConductGlaIce	TKGLAICE	Real	W/m/K
Flux				
incoming downward direct solar radiation	RadSwDownDir	SOLAD	Real	W/m^2
incoming diffuse solar radiation	RadSwDownDif	SOLAI	Real	W/m^2
solar radiation absorbed by vegetation	RadSwAbsVeg	SAV	Real	W/m^2
solar radiation absorbed by ground	RadSwAbsGrd	SAG	Real	W/m^2
total absorbed solar radiation	RadSwAbsSfc	FSA	Real	W/m^2
canopy net longwave radiation [+ to atm]	RadLwNetCanopy	IRC	Real	W/m^2
vegetated ground net longwave radiation [+ to atm]	RadLwNetVegGrd	IRG	Real	W/m^2
bare ground net longwave radiation [+ to atm]	RadLwNetBareGrd	IRB	Real	W/m^2
total net LW radiation [+ to atm]	RadLwNetSfc	FIRA	Real	W/m^2
reflected solar radiation by vegetation	RadSwReflVeg	FSRV	Real	W/m^2
reflected solar radiation by ground	RadSwReflGrd	FSRG	Real	W/m^2
total reflected solar radiation	RadSwReflSfc	FSR	Real	W/m^2
emitted outgoing IR	RadLwEmitSfc	FIRE	Real	W/m^2
total sensible heat flux [+ to atm]	HeatSensibleSfc	FSH	Real	W/m^2
canopy latent heat flux [+ to atm]	HeatLatentCanopy	FCEV	Real	W/m^2

total ground latent heat [+ to atm]	HeatLatentGrd	FGEV	Real	W/m^2
latent heat flux from transpiration [+ to atm]	HeatLatentTransp	FCTR	Real	W/m^2
canopy sensible heat flux [+ to atm]	HeatSensibleCanopy	SHC	Real	W/m^2
vegetated ground sensible heat flux [+ to atm]	HeatSensibleVegGrd	SHG	Real	W/m^2
bare ground sensible heat flux [+ to atm]	HeatSensibleBareGrd	SHB	Real	W/m^2
latent heat flux from bareground [+ to atm]	HeatLatentBareGrd	EVB	Real	W/m^2
canopy evaporation heat flux [+ to atm]	HeatLatentCanEvap	EVC	Real	W/m^2
canopy transpiration heat [+ to atm]	HeatLatentCanTransp	TR	Real	W/m^2
bare ground heat flux [+ to soil/snow]	HeatGroundBareGrd	GHB	Real	W/m^2
vegetated ground heat flux [+ = to soil/snow]	HeatGroundVegGrd	GHV	Real	W/m^2
total ground heat flux [+ = to soil/snow]	HeatGroundTot	SSOIL	Real	W/m^2
light penetrating through snow and soil water	RadSwPenetrateGrd	PHI	Real	W/m^2
ground evaporation heat flux [+= to atm]	HeatLatentVegGrd	EVG	Real	W/m^2
total photosynthesis active energy absorbed by canopy	RadPhotoActAbsCan	APAR	Real	W/m^2
average absorbed par for sunlit leaves	RadPhotoActAbsSunlit	PARSUN	Real	W/m^2
average absorbed par for shaded leaves	RadPhotoActAbsShade	PARSHA	Real	W/m^2
irrigation latent heating due to sprinkler evaporation	HeatLatentIrriEvap	FIRR	Real	W/m^2
precipitation advected heat - canopy net	HeatPrecipAdvCanopy	PAHV	Real	W/m^2
precipitation advected heat - vegetated ground net	HeatPrecipAdvVegGrd	PAHG	Real	W/m^2
precipitation advected heat - bare ground net	HeatPrecipAdvBareGrd	PAHB	Real	W/m^2
precipitation advected heat - total	HeatPrecipAdvSfc	PAH	Real	W/m^2
solar radiation flux abs by vegetation per unit direct flux	RadSwAbsVegDir	FABD	Real	-
solar radiation flux abs by vegetation per unit diffuse flux	RadSwAbsVegDif	FABI	Real	-
solar radiation flux reflected by veg layer (per unit direct flux)	RadSwReflVegDir	FREVD	Real	-
solar radiation flux reflected by veg layer (per unit diffuse flux)	RadSwReflVegDif	FREVI	Real	-

solar radiation flux reflected by ground (per unit direct flux)	RadSwReflGrdDir	FREGD	Real	-
solar radiation flux reflected by ground (per unit diffuse flux)	RadSwReflGrdDif	FREGI	Real	-
downward solar direct flux below veg per unit direct flux	RadSwDirTranGrdDir	FTDD	Real	-
downward solar direct flux below veg per unit diffue flux (= 0)	RadSwDirTranGrdDif	FTDI	Real	-
downward solar diffuse flux below veg (per unit dir flux)	RadSwDifTranGrdDir	FTID	Real	-
downward solar diffuse flux below veg (per unit dif flux)	RadSwDifTranGrdDif	FTII	Real	-
energy influx from soil bottom	HeatFromSoilBot	EFLXB	Real	J/m^2
accumulated ground heat flux during soil timestep	HeatGroundTotAcc	ACC_SSOIL	Real	W/m^2 * dt_soil / dt_main
mean ground heat flux during soil timestep	HeatGroundTotMean	SSOIL_avg	Real	W/m^2
canopy heat storage change	HeatCanStorageChg	CANHS	Real	W/m^2
Parameters				
soil quartz content	SoilQuartzFrac	QUARTZ	Real	-
volumetric soil heat capacity	SoilHeatCapacity	CSOIL	Real	$J/m^3/K$
Zilitinkevich Coefficient for exchange coefficient calculation	ZilitinkevichCoeff	CZIL	Real	-
emissivity 1-soil; 2-lake	EmissivitySoilLake	EG	Real	-
saturated soil albedos: 1=vis, 2=nir	AlbedoSoilSat	ALBSAT	Real	-
dry soil albedos: 1=vis, 2=nir	AlbedoSoilDry	ALBDRY	Real	-
albedo land ice: 1=vis, 2=nir	AlbedoLandIce	ALBICE	Real	-
albedo frozen lakes: 1=vis, 2=nir	AlbedoLakeFrz	ALBLAK	Real	-
green vegetation fraction [0.0-1.0]	VegFracGreen	SHDFAC	Real	-
annual max vegetation fraction	VegFracAnnMax	SHDMAX	Real	-
characteristic leaf dimension	LeafDimLength	DLEAF	Real	m
height of canopy top	HeightCanopyTop	HVT	Real	m
Height of canopy bottom	HeightCanopyBot	HVB	Real	m
tree density	TreeDensity	DEN	Real	no. of trunks per m^2
tree crown radius	TreeCrownRadius	RC	Real	m
monthly stem area index, one-sided	StemAreaIndexMon	SAIM	Real	m^2/m^2
monthly leaf area index, one-sided	LeafAreaIndexMon	LAIM	Real	m^2/m^2
Scattering coefficient for snow	ScatterCoeffSnow	OMEGAS	Real	-
Upscattering parameters for snow for direct radiation	UpscatterCoeffSnowDir	BETADS	Real	-
Upscattering parameters for snow for diffuse radiation	UpscatterCoeffSnowDif	BETAIS	Real	-

snow aging parameter for BATS snow albedo	SnowAgeFacBats	TAU0	Real	-
vapor diffusion snow growth factor for BATS snow albedo	SnowGrowVapFacBats	GRAIN_GROWTH	Real	-
extra snow growth factor near freezing BATS snow albedo	SnowGrowFrzFacBats	EXTRA_GROWTH	Real	-
dirt and soot effect factor BATS snow albedo	SnowSootFacBats	DIRT_SOOT	Real	-
zenith angle snow albedo adjustment for BATS	SolarZenithAdjBats	BATS_COSZ	Real	-
new snow visible albedo for BATS	FreshSnoAlbVisBats	BATS_VIS_NEW	Real	-
new snow NIR albedo for BATS	FreshSnoAlbNirBats	BATS_NIR_NEW	Real	-
age factor for diffuse visible snow albedo for BATS	SnoAgeFacDifVisBats	BATS_VIS_AGE	Real	-
age factor for diffuse NIR snow albedo for BATS	SnoAgeFacDifNirBats	BATS_NIR_AGE	Real	-
cosz factor for direct visible snow albedo for BATS	SzaFacDirVisBats	BATS_VIS_DIR	Real	-
cosz factor for direct NIR snow albedo for BATS	SzaFacDirNirBats	BATS_NIR_DIR	Real	-
reference snow albedo in CLASS scheme	SnowAlbRefClass	CLASS_ALB_REF	Real	-
snow aging e-folding time in CLASS scheme	SnowAgeFacClass	CLASS_SNO_AGE	Real	s
fresh snow albedo in CLASS scheme	SnowAlbFreshClass	CLASS_ALB_NEW	Real	-
minimum stomatal resistance in Jarvis scheme	ResistanceStomataMin	RSMIN	Real	s/m
maximal stomatal resistance in Jarvis scheme	ResistanceStomataMax	RSMAX	Real	s/m
surface resistance for snow	ResistanceSnowSfc	RSURF_SNOW	Real	s/m
exponent in the shape parameter for soil resistance option 1	ResistanceSoilExp	RSURF_EXP	Real	-
ice surface emissivity	EmissivityIceSfc	EICE	Real	-
snow emissivity	EmissivitySnow	SNOW_EMIS	Real	-
empirical canopy wind parameter	CanopyWindExtFac	CWPVT	Real	-
snow surface roughness length (0.002)	RoughLenMomSnow	Z0SNO	Real	m
Bare-soil roughness length (under the canopy)	RoughLenMomSoil	Z0SOIL	Real	m
momentum roughness length vegetated surface	RoughLenMomVeg	Z0MVT	Real	m
lake surface roughness length	RoughLenMomLake	Z0LAKE	Real	m
co2 Michaelis-Menten constant at 25c	Co2MmConst25C	KC25	Real	Pa

change in co2 Michaelis-Menten constant for every 10-deg C temperature change	Co2MmConstQ10	AKC	Real	-
o2 michaelis-menten constant at 25c	O2MmConst25C	KO25	Real	Pa
change in o2 michaelis-menten constant for every 10-deg C temperature change	O2MmConstQ10	AKO	Real	-
minimum leaf conductance	ConductanceLeafMin	BP	Real	umol / m^2/s
Parameter used in radiation stress function in Jarvis scheme	RadiationStressFac	RGL	Real	-
Optimum transpiration air temperature in Jarvis scheme	AirTempOptimTransp	TOPT	Real	K
Parameter used in vapor pressure deficit function in Jarvis scheme	VaporPresDeficitFac	HS	Real	-
leaf/stem orientation index	CanopyOrientIndex	XL	Real	-
leaf reflectance: 1=vis, 2=nir	ReflectanceLeaf	RHOL	Real	-
stem reflectance: 1=vis, 2=nir	ReflectanceStem	RHOS	Real	-
leaf transmittance: 1=vis, 2=nir	TransmittanceLeaf	TAUL	Real	-
stem transmittance: 1=vis, 2=nir	TransmittanceStem	TAUS	Real	-
canopy biomass heat capacity parameter	HeatCapacCanFac	CBIOM	Real	m

9. References

Abolafia-Rosenzweig, R., C. He, S. M. Skiles, F. Chen, and D. Gochis (2022): Evaluation and optimization of snow albedo scheme in Noah-MP land surface model using in-situ spectral observations in the Colorado Rockies, Journal of Advances in Modeling Earth Systems, doi:10.1029/2022MS003141

Abolafia-Rosenzweig, R., C. He, S. Burns, and F. Chen (2021): Implementation and evaluation of a unified turbulence parameterization throughout the canopy and roughness sublayer in Noah-MP snow simulations, Journal of Advances in Modeling Earth Systems, 13(11), e2021MS002665, https://doi.org/10.1029/2021MS002665

Anderson, E. A. (1976), A point energy and mass balance model of a snow cover, NOAA Tech. Rep. NWS 19, 150 pp., Off. of Hydrol., Natl. Weather Serv., Silver Spring, Md.

Ball, J. T., I. E. Woodrow, and J. A. Berry (1987), A model predicting sto- matal conductance and its contribution to the control of photosynthesis under different environmental conditions, in Process in Photosynthesis Research, vol. 1, edited by J. Biggins, pp. 221–234, Martinus Nijhoff, Dordrecht, Netherlands.

Barlage, M., Tewari, M., Chen, F., Miguez-Macho, G., Yang, Z. L., & Niu, G. Y. (2015). The effect of groundwater interaction in North American regional climate simulations with WRF/Noah-MP. Climatic Change, 129, 485-498.

Bavi, A., Kashkuli, H. A., Boroomand, S., Naseri, A., & Albaji, M. (2009). Evaporation losses from sprinkler irrigation systems under various operating conditions. Journal of Applied Sciences, 9(3), 597-600.

Bonan, G. B. (1996), A land surface model (LSM version 1.0) for ecolog- ical, hydrological, and atmospheric studies: Technical description and user's guide, NCAR Tech. Note NCAR/TN-417+STR, 150 pp., Natl. Cent. for Atmos. Res., Boulder, Colo.

Bonan, G. B., Lawrence, P. J., Oleson, K. W., Levis, S., Jung, M., Reichstein, M., et al. (2011). Improving canopy processes in the Community Land Model version 4 (CLM4) using global flux fields empirically inferred from FLUXNET data. Journal of Geophysical Research, 116(G2), 1–22. https://doi.org/10.1029/2010jg001593

Bonan, G.B. (2008), Ecological Climatology: Concepts and Applications (2nd edition), Cambridge University Press, Cambridge, 550 pp.

Brutsaert, W. A. (1982), Evaporation Into the Atmosphere, 299 pp., D. Reidel, Dordrecht, Netherlands.

Cai, X., Yang, Z. L., Fisher, J. B., Zhang, X., Barlage, M., & Chen, F. (2016). Integration of nitrogen dynamics into the Noah-MP land surface model v1. 1 for climate and environmental predictions. Geoscientific Model Development, 9(1), 1-15.

Chen, F., K. Mitchell, J. Schaake, Y. Xue, H. Pan, V, Koren, Y. Duan, M. Ek, and A, Betts (1996), Modeling of land-surface evaporation by four schemes and comparison with FIFE observations, J. Geophys. Res.,101, 7251-7268.

Chen, F., Z. Janjic, and K. Mitchell (1997), Impact of atmospheric surface layer parameterization in the new land- surface scheme of the NCEP Mesoscale Eta numerical model, Bound.-Layer Meteor., 185, 391-421.

Collatz, G., Ribas-Carbo, M. and Berry, J.: Coupled Photosynthesis-Stomatal Conductance Model for Leaves of C4 Plants, Funct. Plant Biol., 19(5), 519, doi:10.1071/PP9920519, 1992.

Dickinson, R. E. (1983), Land surface processes and climate-surface albe- dos and energy balance, in Theory of Climate, Adv. Geophys., vol. 25, edited by B. Saltzman, pp. 305–353, Academic, San Diego, Calif.

Dickinson, R. E., A. Henderson-Sellers, and P. J. Kennedy (1993), Bio- sphere-Atmosphere Transfer Scheme (BATS) version 1e as coupled to the NCAR Community Climate Model, NCAR Tech. Note NCAR/TN- 387+STR, 80 pp., Natl. Cent. for Atmos. Res., Boulder, Colo.

Dickinson, R. E., M. Shaikh, R. Bryant, and L. Graumlich (1998), Interac- tive canopies for a climate model, J. Clim., 11, 2823–2836, doi:10.1175/ 1520-0442(1998)011<2823:ICFACM>2.0.CO;2.

Douville, H., J.-F. Royer, and J.-F. Mahfouf (1995) A new snow parameterization for the Météo-France climate model. Part I: validation in stand-alone experiments, Climate Dynamics, 12, 21-35.

Ek, M., and Mahrt, L. (1991). OSU 1-D PBL model user's guide. Version, 1(120), 97331-2209.

Ek, M. B., Mitchell, K. E., Lin, Y., Rogers, E., Grunmann, P., Koren, V., et al. (2003). Implementation of Noah land surface model advances in the National Centers for Environmental Prediction operational mesoscale Eta model. Journal of Geophysical Research: Atmospheres, 108, 2002JD003296. https://doi.org/10.1029/2002JD003296

Fan, Y., Miguez-Macho, G., Weaver, C. P., Walko, R., & Robock, A. (2007). Incorporating water table dynamics in climate modeling: 1. Water table observations and equilibrium water table simulations. Journal of Geophysical Research: Atmospheres, 112(D10).

Flerchinger, G.N. and K.E. Saxton (1989), Simultaneous heat and water model of a freezing snow-residue-soil system I. Theory and development, Trans. ASAE, 32(2), 565-571.

He, C., F. Chen, M. Barlage, C. Liu, A. Newman, W. Tang, K. Ikeda, and R. Rasmussen (2019): Can convection-permitting modeling provide decent precipitation for offline high-resolution snowpack simulations over mountains, J. Geophys. Res.-Atmos, 124, https://doi.org/10.1029/2019JD030823

He, C., F. Chen, R. Abolafia-Rosenzweig, K. Ikeda, C. Liu, and R. Rasmussen (2021): What causes the unobserved early-spring snowpack ablation in convection-permitting WRF modeling over Utah mountains?, J. Geophys. Res.-Atmos, 126(22), e2021JD035284, https://doi.org/10.1029/2021JD035284

Jarvis, P. G. (1976), The interpretation of the variations in leaf water poten- tial and stomatal conductance found in canopies in the field, Philos. Trans. R. Soc. B, 273, 593–610, doi:10.1098/rstb.1976.0035.

Jayawardena, A. W. and Zhou, M. C.: A modified spatial soil moisture storage capacity distribution curve for the Xinanjiang model, Journal of Hydrology, 227, 93–113, https://doi.org/10.1016/S0022-1694(99)00173-0, 2000.

Jiang, Y., Chen, F., Gao, Y., He, C., Barlage, M., & Huang, W. (2020). Assessment of uncertainty sources in snow cover simulation in the Tibetan Plateau. Journal of Geophysical Research: Atmospheres, 125(18), e2020JD032674.

Jordan, R. (1991), A one-dimensional temperature model for a snow cover, Spec. Rep. 91–16, Cold Reg. Res. and Eng. Lab., U.S. Army Corps of Eng., Hanover, N. H.

Knoben, W. J., Freer, J. E., Fowler, K. J., Peel, M. C., & Woods, R. A. (2019). Modular Assessment of Rainfall–Runoff Models Toolbox (MARRMoT) v1. 2: an open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations. Geoscientific Model Development, 12(6), 2463-2480.

Koren, V., J. Schaake, K. Mitchell, Q.-Y. Duan, F. Chen, and J. M. Baker (1999), A parameterization of snowpack and frozen ground intended for NCEP weather and climate models, J. Geophys. Res., 104(D16), 19569– 19586.

Li, L., Yang, Z. L., Matheny, A. M., Zheng, H., Swenson, S. C., Lawrence, D. M., ... & Leung, L. R. (2021). Representation of plant hydraulics in the Noah-MP land surface model: Model development and multiscale evaluation. Journal of Advances in Modeling Earth Systems, 13(4), e2020MS002214.

Liang, X., Lettenmaier, D. P., Wood, E. F., & Burges, S. J. (1994). A simple hydrologically based model of land surface water and energy fluxes for general circulation models. Journal of Geophysical Research: Atmospheres, 99(D7), 14415-14428.

Liang, X., & Xie, Z. (2003). Important factors in land–atmosphere interactions: surface runoff generations and interactions between surface and groundwater. Global and Planetary Change, 38(1-2), 101-114.

Liu, X., Chen, F., Barlage, M., Zhou, G., & Niyogi, D. (2016). Noah-MP-Crop: Introducing dynamic crop growth in the Noah-MP land surface model. Journal of Geophysical Research: Atmospheres, 121(23), 13,953-13,972. https://doi.org/10.1002/2016JD025597

Łobocki, L. (1993), A Procedure for the Derivation of Surface-Layer Bulk Relationships from Simplified Second Order Closure Models, J. Appl. Meteorol., 32, 126−138.

Lynch-Stieglitz, M. (1994), The development and validation of a simple snow model for the GISS GCM, J. Climate, 7, 1842-1855.

Mahrt, L., and Ek, M. (1984). The influence of atmospheric stability on potential evaporation. Journal of Applied Meteorology and Climatology, 23(2), 222-234.

Mahrt, L., and Pan, H. (1984). A two-layer model of soil hydrology. Boundary-Layer Meteorology, 29, 1-20.

McDaniel, R., Liu, Y., Valayamkunnath, P., Barlage, M., Gochis, D., Cosgrove, B. A., & Flowers, T. (2020, December). Moisture condition impact and seasonality of National Water Model performance under different runoff-infiltration partitioning schemes. In AGU Fall Meeting Abstracts (Vol. 2020, pp. H111-0028).

Mellor, G. L., and T. Yamada (1982), Development of a turbulence closure model for geophysical fluid problems, Rev. Geophys., 20(4), 851−875, doi:10.1029/RG020i004p00851.

Meier, J., Zabel, F., & Mauser, W. (2018). A global approach to estimate irrigated areas–a comparison between different data and statistics. Hydrology and Earth System Sciences, 22(2), 1119-1133.

Miguez-Macho, G., Fan, Y., Weaver, C. P., Walko, R., & Robock, A. (2007). Incorporating water table dynamics in climate modeling: 2. Formulation, validation, and soil moisture simulation. Journal of Geophysical Research: Atmospheres, 112(D13).

Mockus, V. (1964). National engineering handbook. US Soil Conservation Service: Washington, DC, USA, 4.

Niu, G.-Y. and Z.-L. Yang (2004), The effects of canopy processes on snow surface energy and mass balances, Journal of Geophysical Research, 109, D23111, doi:10.1029/2004JD004884.

Niu, G.-Y. and Z.-L. Yang, 2006: Effects of frozen soil on snowmelt runoff and soil water storage at a continental scale, Journal of Hydrometeorology, 7 (5), 937-952.

Niu, G.-Y., and Z.-L. Yang (2007), An observation-based formulation of snow cover fraction and its evaluation over large North American river basins, J. Geophys. Res., 112, D21101, doi:10.1029/2007JD008674.

Niu, G.-Y., Z.-L. Yang, R.E. Dickinson, and L.E. Gulden, 2005: A simple TOPMODEL-based runoff parameterization (SIMTOP) for use in global climate models, Journal of Geophysical Research, 110, D21106, doi:10.1029/2005JD006111.

Niu, G.-Y., Z.-L. Yang, R. E. Dickinson, L. E. Gulden, and H. Su, 2007: Development of a simple groundwater model for use in climate models and evaluation with Gravity Recovery and Climate Experiment data, J. Geophys. Res., 112, D07103, doi:10.1029/2006JD007522.

Niu, G.-Y., Z.-L. Yang, K. E. Mitchell, F. Chen, M. B. Ek, M. Barlage, A. Kumar, K. Manning, D. Niyogi, E. Rosero, M. Tewari, and Y.-L. Xia (2011), The community Noah land surface model with multiparameterization options (Noah-MP): 1. Model description and evaluation with local-scale measurements, J. Geophys. Res., 116, D12109, doi:10.1029/2010JD015139.

Oleson, K.W., Y. Dai, G. Bonan, M. Bosilovich, R. Dickinson, P. Dirmeyer, F. Hoffman, P. Houser, S. Levis, G.-Y. Niu, P. Thornton, M. Vertenstein, Z.-L. Yang, and X. Zeng (2004), Technical Description of the Community Land Model (CLM), NCAR/TN-461+STR, 174 pp. (Available at www.cgd.ucar.edu/tss/clm/distribution/clm3.0/index.html.)

Oleson, K. W., Lawrence. D. M., Bonan, G. B., B. Drewniak, M. Huang, C. D. Koven, S. Levis, F. Li, W. J. Riley, Z. M. Subin, S. C. Swenson, P. E. Thornton, A. Bozbiyik, R. Fisher, C. L. Heald, E. Kluzek, J.F. Lamarque, P. J. Lawrence, L. R. Leung, W. Lipscomb, S. Muzszala, D. M. Ricciuto, W. Sacks, Y. Sun, J. Tang, and Z.-L. Yang (2013), Technical Description of version 4.5 of the Community Land Model (CLM). NCAR Technical Note NCAR/TN-503+STR, 420 pp., doi:10.5065/D6RR1W7M.

Pan, H. L., and Mahrt, L. (1987). Interaction between soil hydrology and boundary-layer development. Boundary-Layer Meteorology, 38, 185-202.

Paulson, C. A. (1970), The Mathematical Representation of Wind Speed and Temperature Profiles in the Unstable Atmospheric Surface Layer, J. Appl. Meteorol., 9, 857–861.

Pervez, M. S., & Brown, J. F. (2010). Mapping irrigated lands at 250-m scale by merging MODIS data and national agricultural statistics. Remote Sensing, 2(10), 2388-2412.

Sakaguchi, K., & Zeng, X. (2009). Effects of soil wetness, plant litter, and under-canopy atmospheric stability on ground evaporation in the Community Land Model (CLM3. 5). Journal of Geophysical Research: Atmospheres, 114(D1).

Saxton, K. E., & Rawls, W. J. (2006). Soil water characteristic estimates by texture and organic matter for hydrologic solutions. Soil science society of America Journal, 70(5), 1569-1578.

Sellers, P. J. (1985), Canopy reflectance, photosynthesis and transpiration, Int. J. Remote Sens., 6, 1335–1372, doi:10.1080/01431168508948283.

Sellers, P. J., D. A. Randall, G. J. Collatz, J. A. Berry, C. B. Field, D. A. Dazlich, C. Zhang, G. D. Collelo, and L. Bounoua (1996a), A revised land surface parameterization (SiB2) for atmospheric GCMs. Part I: Model formulation, J. Clim., 9, 676–705, doi:10.1175/1520-0442 (1996)009<0676:ARLSPF>2.0.CO;2.

Sellers, P. J., M. D. Heiser, and F. G. Hall (1992), Relations between surface conductance and spectral vegetation indices at intermediate (100 m2 to 15 km2) length scales, J. Geophys. Res., 97, 19,033–19,059, doi:10.1029/92JD01096.

Schaake, J. C., V. I. Koren, Q.-Y. Duan, K. E. Mitchell, and F. Chen (1996), Simple water balance model for estimating runoff at different spatial and temporal scales, J. Geophys. Res., 101, 7461–7475, doi:10.1029/95JD02892.

Smith, R. E., Smettem, K. R., & Broadbridge, P. (2002). Infiltration theory for hydrologic applications. American Geophysical Union.

Valiantzas, J. D. (2010). New linearized two-parameter infiltration equation for direct determination of conductivity and sorptivity. Journal of Hydrology, 384(1-2), 1-13.

Valayamkunnath, P. (2019). Understanding the Role of Vegetation Dynamics and Anthropogenic induced Changes on the Terrestrial Water Cycle (Doctoral dissertation, Virginia Tech).

Valayamkunnath, P., Hession, W. C., & Chen, F. (2020, January). Understanding the Role of Vegetation Dynamics and Anthropogenic-Induced Changes on the Terrestrial Water Cycle. In 100th American Meteorological Society Annual Meeting. AMS.

Valayamkunnath, P., Chen, F., Barlage, M. J., Gochis, D. J., Franz, K. J., & Cosgrove, B. A. (2021, January). Impact of Agriculture Management Practices on the National Water Model Simulated Streamflow. In 101st American Meteorological Society Annual Meeting. AMS.

Valayamkunnath, P., Gochis, D. J., Chen, F., Barlage, M., & Franz, K. J. (2022). Modeling the hydrologic influence of subsurface tile drainage using the National Water Model. Water Resources Research, 58(4), e2021WR031242.

Verseghy, D. L. (1991), CLASS-A Canadian land surface scheme for GCMS: I. Soil model, Int. J. Climatol., 11, 111–133, doi:10.1002/joc.3370110202.

Waller, P., & Yitayew, M. (2015). Irrigation and drainage engineering. Springer.

Wang, Y. H., Broxton, P., Fang, Y., Behrangi, A., Barlage, M., Zeng, X., & Niu, G. Y. (2019). A wet-bulb temperature-based rain-snow partitioning scheme improves snowpack prediction over the drier western United States. Geophysical Research Letters, 46(23), 13825-13835.

Wang, W., Yang, K., Zhao, L., Zheng, Z., Lu, H., Mamtimin, A., ... & Moore, J. C. (2020). Characterizing surface albedo of shallow fresh snow and its importance for snow ablation on the interior of the Tibetan Plateau. Journal of Hydrometeorology, 21(4), 815-827.

Wang, W., He, C., Moore, J., Wang, G., & Niu, G. Y. (2022). Physics-Based Narrowband Optical Parameters for Snow Albedo Simulation in Climate Models. Journal of Advances in Modeling Earth Systems, 14(1), e2020MS002431.

Wood, E. F., Lettenmaier, D. P., & Zartarian, V. G. (1992). A land-surface hydrology parameterization with subgrid variability for general circulation models. Journal of Geophysical Research: Atmospheres, 97(D3), 2717-2728.

Xue, Y., P. J. Sellers, J. L. Kinter, and J. Shukla (1991), A simplified biosphere model for global climate studies, J. Clim., 4, 345 – 364, doi:10.1175/1520-0442(1991)004<0345:ASBMFG>2.0.CO;2.

Yang, Z.-L., and R. E. Dickinson (1996), Description of the Biosphere-Atmosphere Transfer Scheme (BATS) for the soil moisture workshop and evaluation of its performance, Global Planet. Change, 13, 117-134, doi:10.1016/0921-8181(95)00041-0.

Yang, R., and M.A. Friedl (2003), Modeling the effects of 3-D vegetation structure on surface radiation and energy balance in boreal forests, Journal of Geophysical Research, Atmospheres, 108 (D16), 8615, doi: 10.1029/2002JD003109.

Yang, Z.-L., G.-Y. Niu, K. E. Mitchell, F. Chen, M. B. Ek, M. Barlage, L. Longuevergne, K. Manning, D. Niyogi, M. Tewari, and Y.-L. Xia (2011). The community Noah land surface model with multiparameterization options (Noah-MP): 2. Evaluation over global river basins, J. Geophys. Res., 116, D12110, doi:10.1029/2010JD015140.

Yen, Y. C. (1965), Heat transfer Characteristics of Naturally Compacted Snow. U.S. Army Cold Regions Research and Engineering Laboratory. Research Report 166, Hanover, NH, 9 pp.

Yen, Y.-C., (1981), Review of thermal properties of snow, ice and sea ice, Cold Regions Research and Engineering Laboratory (CRREL) Report 81-10, U.S. Army Cold Reg. Res. and Eng. Lab., Hanover, NH.

Zhang, Z., Barlage, M., Chen, F., Li, Y., Helgason, W., Xu, X., et al. (2020). Joint modeling of crop and irrigation in the central United States using the Noah-MP land surface model. Journal of Advances in Modeling Earth Systems, 12(7), e2020MS002159.

Zhang, Z., Li, Y., Chen, F., Harder, P., Helgason, W., Famiglietti, J., Valayamkunnath, P., He, C., and Li, Z.: Developing Spring Wheat in the Noah-MP LSM (v4.4) for Growing Season Dynamics and Responses to Temperature Stress, Geosci. Model Dev. Discuss. [preprint], https://doi.org/10.5194/gmd-2022-311, in review, 2023.