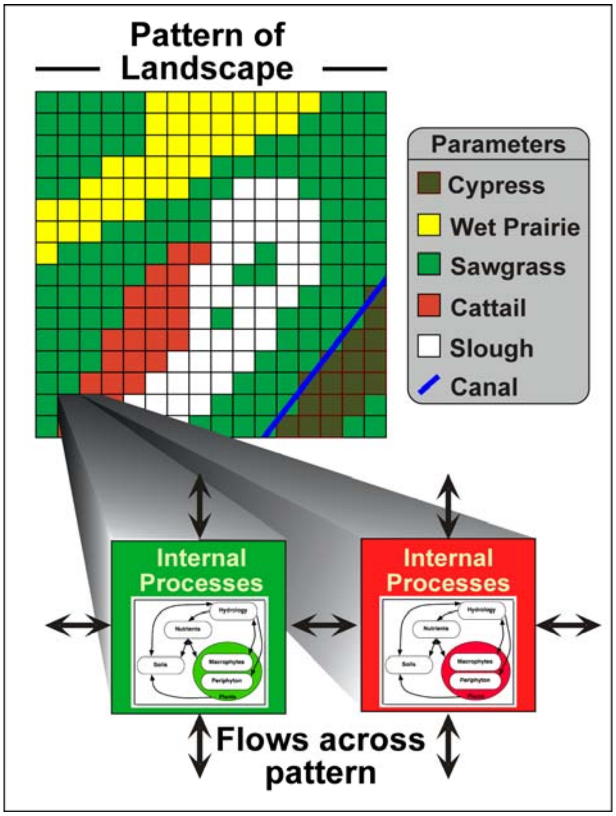
Documentation of the Everglades Landscape Model: ELM v2.5

Chapter 5: Model Structure



http://my.sfwmd.gov/elm

July 10, 2006

Chapter 5: Model Structure

Chapter 5: Model Structure 5-1
5.1 Overview 5-2
5.1.1 ELM conceptual model 5-3
5.1.2 State variables 5-5
5.1.3 Format of algorithm descriptions 5-6
5.2 Source code 5-7
5.3 Main controller 5-9
5.4 Data-input modules 5-10
5.5 Dynamic solutions: sequencing 5-12
5.6 Vertical solutions 5-14
5.6.1 Globals module 5-15
5.6.2 Hydrology module 5-19
5.6.3 Phosphorus, salt/tracer modules 5-29
5.6.4 Periphyton module 5-39
5.6.5 Macrophyte module 5-48
5.6.6 Floc module 5-58
5.6.7 Soils module 5-64
5.7 Horizontal solutions 5-72
5.7.1 Water management: Structure flows module 5-73
5.7.2 Water management: Canal-marsh flux module 5-80
5.7.3 Overland flow module 5-89
5.7.4 Groundwater flow module 5-95
5.8 Habitat succession module 5-103
5.9 Literature cited 5-105

5.1 Overview

The Everglades Landscape Model (ELM) is a spatially distributed simulation using integrated hydro-ecological process modules. With a structured programming approach, the hydrologic, biogeochemical, and biological processes (such as evapotranspiration, soil oxidation, and plant growth) are contained in code modules that are activated by the user at runtime. Being "data-driven", the model relies on databases to modify scenarios of water management, while computer source code remains constant.

This Chapter on Model Structure is organized in a hierarchical fashion that parallels the model structure itself, starting with an overview of the modeling framework. The bulk of the Chapter is then devoted to parsing the simple conceptual model into a higher level of detail for each dynamic module. For each hydro-ecological module, a conceptual model diagram shows the internal interactions and their linkages with other modules. A module Overview provides a text summary of the module's purpose, followed by a verbal and mathematical description of the assumptions and all of the associated equations, variables, and parameters. To most readily understand the important interactions of the dynamic hydro-ecological modules, we recommend that the reader uses the hyper-linked version of this Chapter found on the ELM web site.

A separate User's Guide Chapter includes information on the required computing environment[1] and the basic steps needed to install and use an ELM project.

Using an Open Source[2] philosophy, we hope to encourage collaboration in the modeling community. Towards that end, all source code (and data) necessary for an ELM project is available for download on the ELM web site, and all code in the ELM project is documented in detail using the automated "Doxygen" documentation system. This online, source-code level documentation extends beyond the scientific algorithms described in this Chapter, including details of all of the functions that are compiled in the (ANSI C) code project.

We recommend viewing the hyper-linked version of the algorithm interactions and equations on the ELM web site (Development tab at http://my.sfwmd.gov/elm).

[1] Unix operating system (Linux, Darwin, or Solaris) using Open Source software.
[2] http://www.opensource.org/

5.1.1 ELM conceptual model

The General Ecosystem Conceptual Model presented in an earlier Chapter (Conceptual Model Chapter) forms the basis for the quantitative formulation of the ELM. For this version of ELM, we explicitly integrate fully dynamic flux equations of hydrology, nutrients, plants, and soils within a hydro-ecological "unit" model (Figure 5.1). We hypothesize that these capture the fundamental characteristics of habitats within the Everglades landscape: the dynamic ecological interactions among hydrology, biogeochemistry, and plant biology are critical to understanding and predicting changes within this ever-changing wetland system.

Within this framework of the "unit" model, we sought to quantify the simplest set of ecosystem processes that are fundamental to changes in habitats, or assemblages of vegetation types. Note that, compared to the General Ecosystem Conceptual Model presented earlier, the ELM is simpler in that the effects of fire and consumer interactions are assumed to be inherent in hydrologic disturbances and the long-term dynamic storages and fluxes of the plants. In some respects the modeled interactions are quite simplistic. Importantly, however, we made considerable effort to optimize the balance between realism, which tends to increase model complexity, and (the relative paucity of) supporting data/knowledge, which tends to "scale-back" and simplify a model implementation.

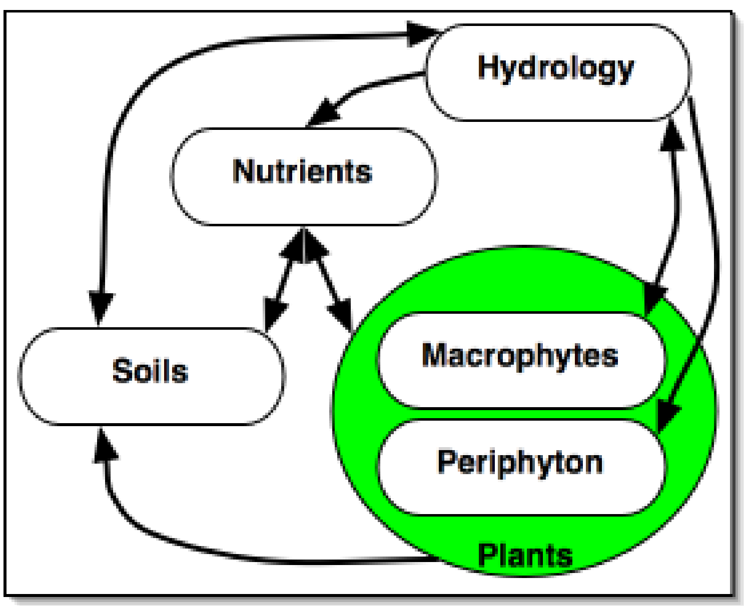


Figure 5.1. The conceptual "unit" model of general ecosystem dynamics incorporated into the ELM.

Within the "unit" model, we assumed that the dynamics occur within a homogenous spatial unit. Significant insights into ecosystem processes may be achieved by focusing on a particular site or homogenous area. However, imperative to understanding landscapes such as the Everglades is the acknowledgement of spatial heterogeneity. In the ELM, ecosystem dynamics are made spatially-explicit by considering the flows and interactions across habitat types that are heterogeneously distributed across a regular model grid (Figure 5.2). The processes internal to grid cells can vary according to habitat type, each of which may have different hydro-ecological parameter sets. Flows of water and nutrients among grid cells are thus affected by changes within cells of the habitat mosaic, and this pattern can change over time as cumulative conditions in grid cells become more favorable for one habitat vs. another.

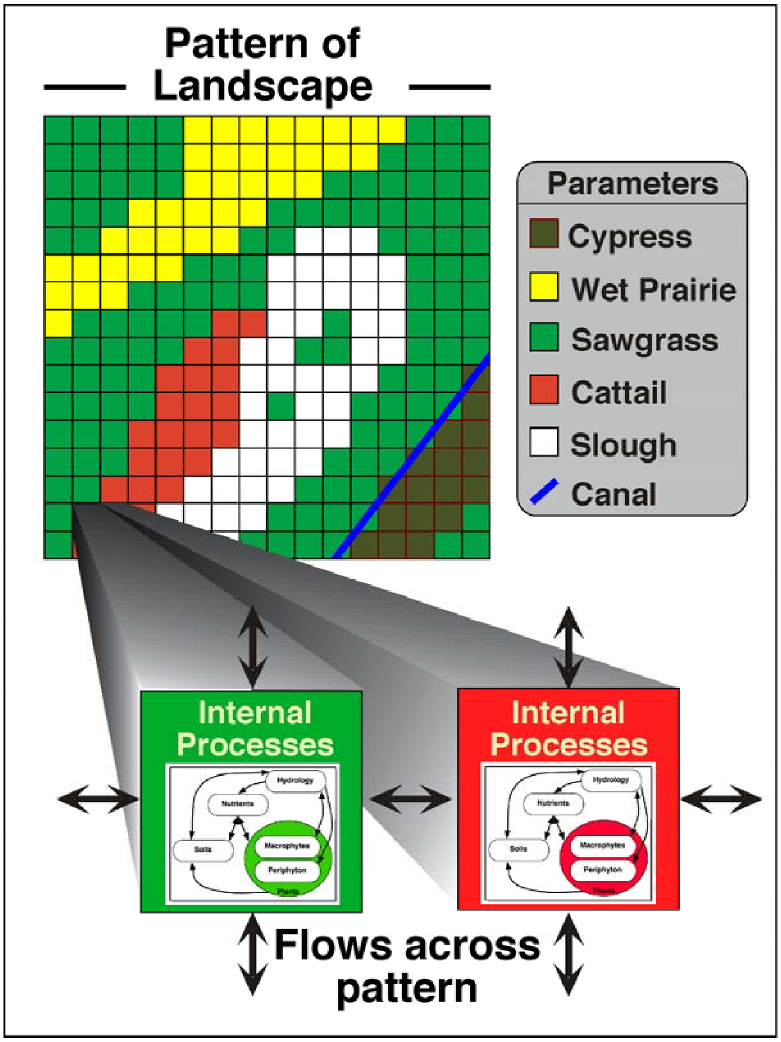


Figure 5.2. The conceptualization of how the “unit” model of general ecosystem dynamics is applied across the heterogeneous spatial grid of different habitat types. Each habitat type within the patterned landscape can be parameterized differently, affecting the internal process dynamics within different grid cells. In turn, the results of the internal processing can affect the direction and magnitude of the flows of water and nutrients across the landscape pattern. Succession, or switching, of habitat types can occur as cumulative conditions warrant.

While the “unit” model dynamics are relatively simple approximations of ecosystems, model complexity arises in its application as a distributed hydro-ecological simulation. The ELM hydrologic processes are relatively simple in their details, with the model simulating the primary hydrologic “drivers” of the Everglades wetlands. The ELM incorporates both overland and subsurface groundwater flows, coupling the surface and ground water exchanges at each time step. Vital to surface (and subsurface) hydrology in the Everglades are the managed flows through water control structures, which are directed into canal vector networks and/or into marsh grid cells of the model. These managed flows transport nutrients through the system, and have major impacts on the spatial pattern of nutrient loads and distribution – and thus the ecology of the landscape.

5.1.2 State variables

The ELM conceptual model presented above shows the fundamental interactions that are captured in the simulation. Further details of how this is implemented may be seen in the diagram of the within-cell interactions among the major state variables[3] (Figure 5.3). These dynamic interactions shown in Figure 5.3 can be split into those occurring above-ground and below-ground, with the same code (but different parameter sets) used in all habitat types distributed through the landscape, from sloughs to forested uplands. Spatial flows that affect these variables are summarized in the later Chapter sections that describe each of the "Horizontal solutions".

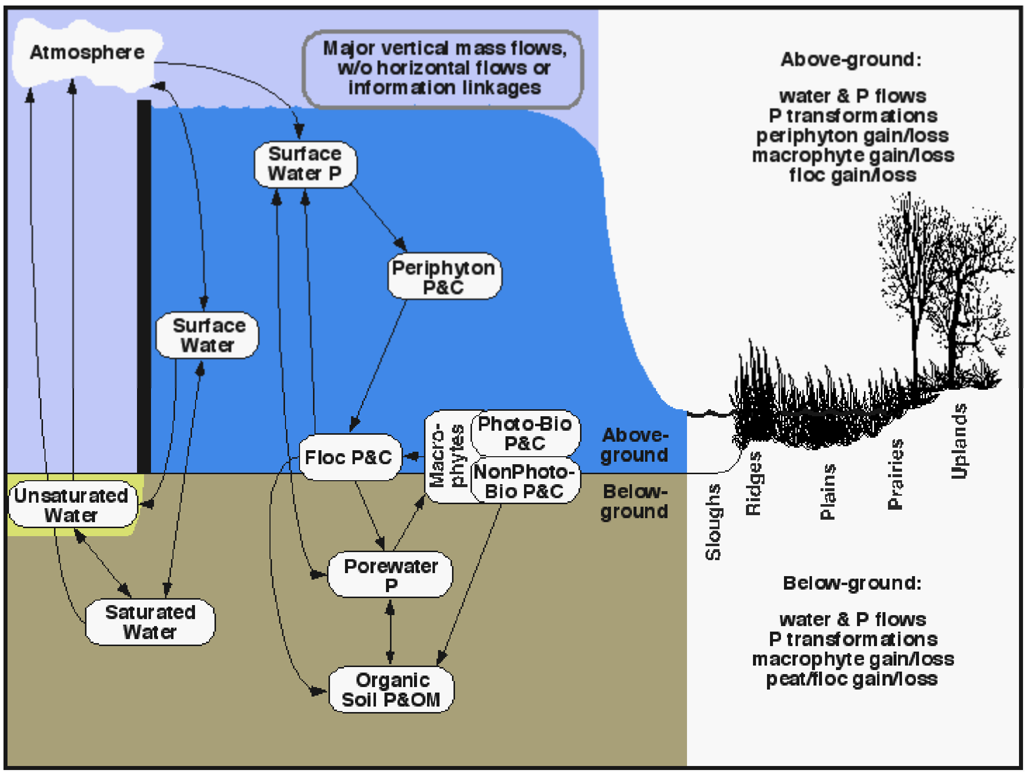


Figure 5.3. The details of the conceptual model of the ELM. State variables are in oval boxes, linked by the major flow pathways among those variables. Abbreviations: P = Phosphorus; C = Carbon; OM = Organic Matter; Photo-Bio = Photosynthetic Biomass of macrophytes; NonPhoto-Bio = NonPhotosynthetic Biomass of macrophytes; Floc = Flocculent layer on/above soil.

For hydrologic dynamics, the surface, unsaturated and saturated storage state variables are measured in terms of the height of water volumes within a grid cell (or canal). Phosphorus in the surface water and porewater storages are known as masses within the cell or canal. Carbon mass is the common unit of flux among the biotic storages of

[3] Because the salt/tracer constituent does not currently affect model dynamics, the two state variables associated with this module are not shown.

periphyton and macrophytes, along with the storage in abiotic flocculent organic material (floc). Carbon is converted to mass of organic material when considering storage in the consolidated soil beneath the floc layer. Mass of phosphorus is maintained via parallel state variables associated with these carbon and organic matter fluxes. Mass balance is strictly maintained (and verified) in the model.

5.1.2.1 Solution methods

To update the state variables, the method of solving the model's finite difference equations is the simple Euler method of integration, without complexities such as forward looking methods. Daily time steps are used in all of the "unit" model vertical solutions, whereas the horizontal solutions are generally dependent on grid cell resolution for the appropriate time step, as described later in the relevant modules' sections. (The regional $1km^2$ ELM application uses a 2-hour time step for most horizontal solutions). The User's Guide Chapter discusses topics such as selection of time steps and the associated run times[4] of the model at different scales. We note here, however, that the horizontal solutions that are primarily hydrologic in origin comprise ~75% of the total model runtime. The following is a breakdown of relative CPU time[5] for generalized classes of modules in the regional implementation:

- 51% total CPU time on water management fluxes
- 26% total CPU time on surface/ground water raster fluxes (incl. vertical integration)
- 19% total CPU time on unit model "vertical" fluxes
- 4% total CPU time on other tasks (budgets, input/output, etc)

5.1.3 Format of algorithm descriptions

We separate the descriptions of the algorithms into those primarily involving solutions of vertical flows/processes, and those involving horizontal flows. The vertical solutions are primarily those involving the "unit" model, while the horizontal solutions involve spatial flows of water and constituents among raster grid cells and/or canal vectors. Prior to the sections that describe each module of vertical and horizontal solutions, we present the main program's sequence of principal function calls. The nature of the input data functions is then briefly presented.

In the descriptions of the algorithms in each module, a common format is used. Text descriptions of the basic assumptions are followed by "pseudo-code" of all of the equations used in algorithm calculations within the module, organized as follows:

- *State variables:* The difference equation(s) that is solved to update the state variable, such as surface water height or carbon biomass of periphyton. These equations are shown first in the presentations of each module, but they are actually dependent on the below intermediate calculations.

---

[4] On a 2.66 GHz laptop, it takes somewhat more than one hour to run a 20-year, regional application of ELM.

[5] Expressed in percent of total CPU seconds for each aggregation of tasks; profiling was done on the ELM v2.3 code in a 19-year simulation, using the Analyzer in Sun Forte Developer 6.

- *Attributes:* These may include calculations of intermediate variables such as the depth of the unsaturated zone, or the current concentration of phosphorus in the water column.
- *Control functions:* These may include the relationship between root depth & the current water levels relative to transpiration demand, or the degree of nutrient limitation on periphyton growth.
- *Fluxes:* The potential and actual fluxes, constrained by the attributes and control functions previously described; these may include actual evapotranspiration losses, or gross primary production gains by periphyton.

Following the equations are tables containing the units and definitions of all state variables, intermediate variables, and parameters used in that function. A listing and location reference is given for all dependent variables whose values are calculated in another module. At the end of each module description is a glossary of any intrinsic functions (e.g., Abs(x) = Absolute value of x) that are used in the pseudo-code.

5.1.3.1 Navigational tool

Most of the remainder of this Chapter is used to describe the algorithms in each module, including the interaction among modules. The Model Structure section of the ELM web site contains this same text and figures, but provides hyper-links among the conceptual diagrams of each module. This method of perusing the ELM algorithms is highly recommended in order to more readily understand the important linkages among modules.

5.2 Source code

The ANSI C language source code of the entire ELM project is fully documented using the automated documentation tool Doxygen[6]. All ELM source code (and requisite data) is available for download from the ELM web site[7], and the Doxygen-generated documentation is available in that same location of the web site (not in this document). This web-based source code documentation is primarily targeted to an audience of programmers, but its easy navigation can be useful to clarify a user's understanding of details of dependencies, methods, etc.

Figure 5.4 below shows a simple example of Doxygen-generated documentation of the "f_Manning" function (also described in a later Chapter section on Water Management: Canal-Marsh Flux Module). This function contains the Manning's equation for surface water exchange between a cell and canal. The Figure shows a call graph that indicates "f_Manning" is called by the parent function of "FluxChannel" (that iterates the water and nutrient fluxes between a canal vector and it's adjoining grid cells). Briefly defined are the parameters that are passed into the function, along with the value that is returned by the function. The definitions of functions/macros (Abs, sgn) and a parameter (GP_mannDepthPow) that it references are available via hyperlinks. The actual C code (with hyperlinked functions and parameter) is listed at the end of the example.

---

[6] The Open Source Doxygen application is available at http://www.stack.nl/~dimitri/doxygen/

[7] Source code link in the Development tab at http://my.sfwmd.gov/elm

The remainder of this Chapter specifically avoids the syntax and complexities of source code and Doxygen-generated web pages, and instead focuses on the scientific understanding of the model algorithms.

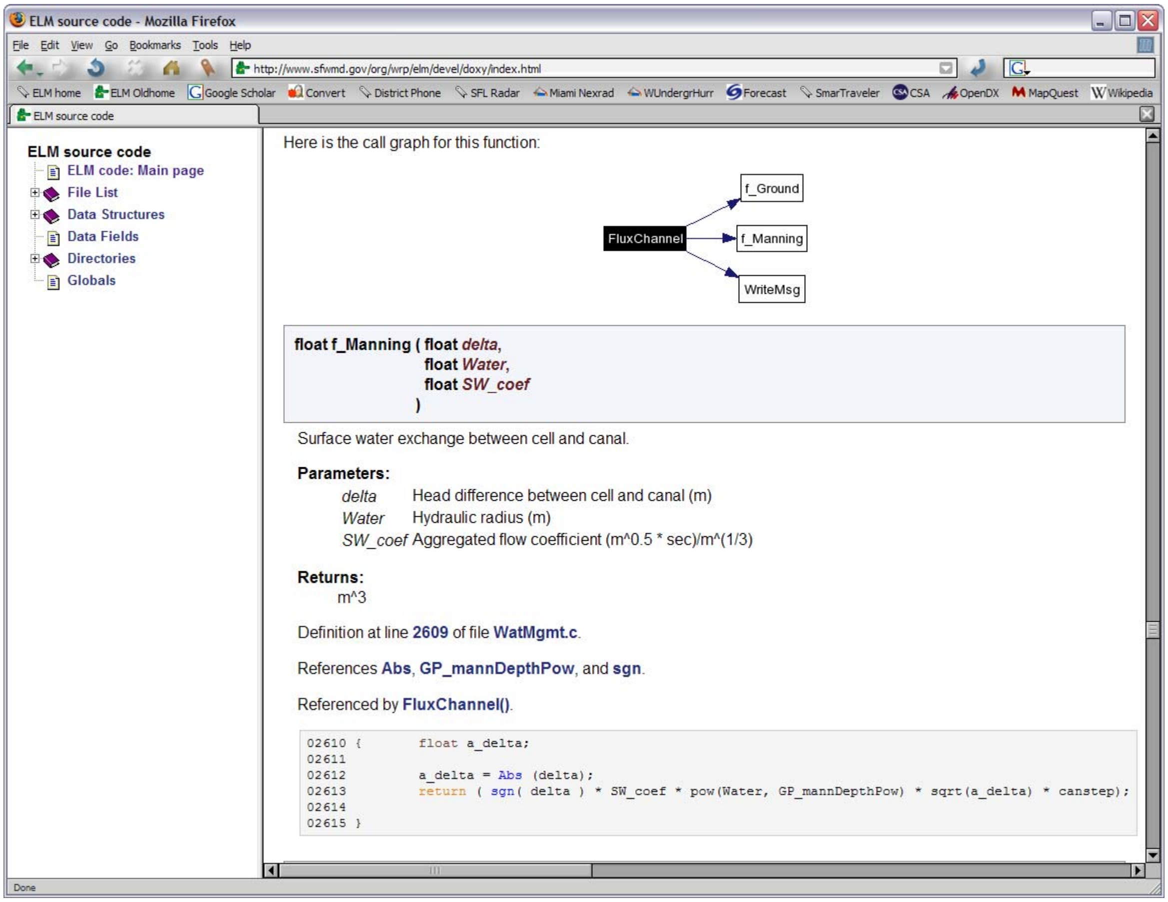
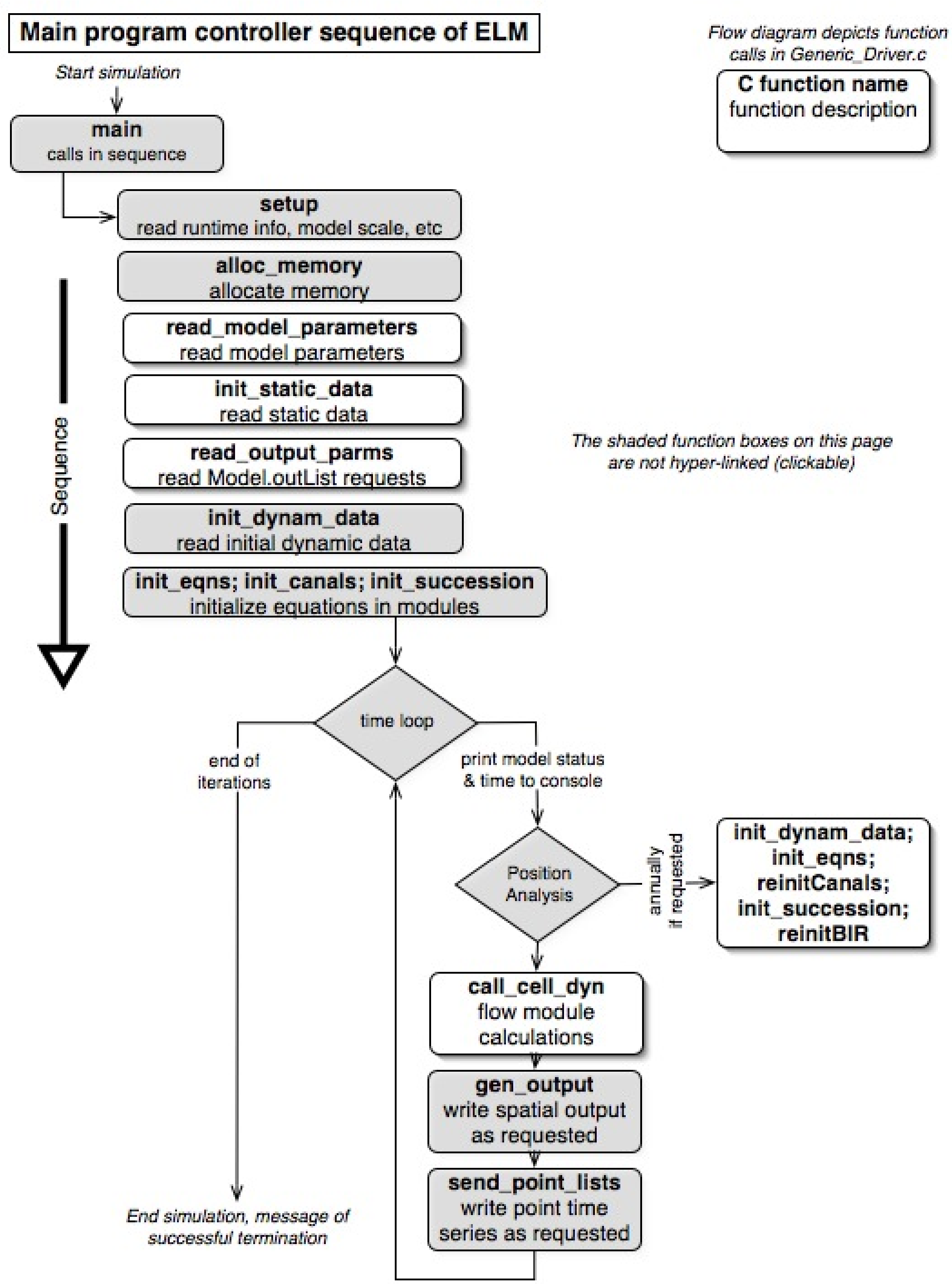


Figure 5.4. Source code documentation example. Primarily intended for an audience of programmers , this is an example of the web-based documentation of a function in the C source code of ELM. After the ELM developers populated the source code with specific “tags”, the Open Source program Doxygen automatically generated well-structured web pages that describe all functions compiled in the ELM project, showing call graphs, descriptions of the purpose of each function, hyperlinked dependencies, definitions of data structures, variables, and many other aspects of the source code, The call graph shown was actually generated for the preceding function (that calls “f_Manning”). The remainder of this Chapter does not use the detailed Doxygen-based information. For the Doxygen-generated documentation, see the Development tab, Hyper-linked source code documentation link at http://www.my.sfwmd.gov/elm .

5.3 Main controller

The Figure below summarizes all of the primary function calls during an execution of the ELM. The "call_cell_dyn" and the data input functions are expanded upon in the next sections of this Chapter.



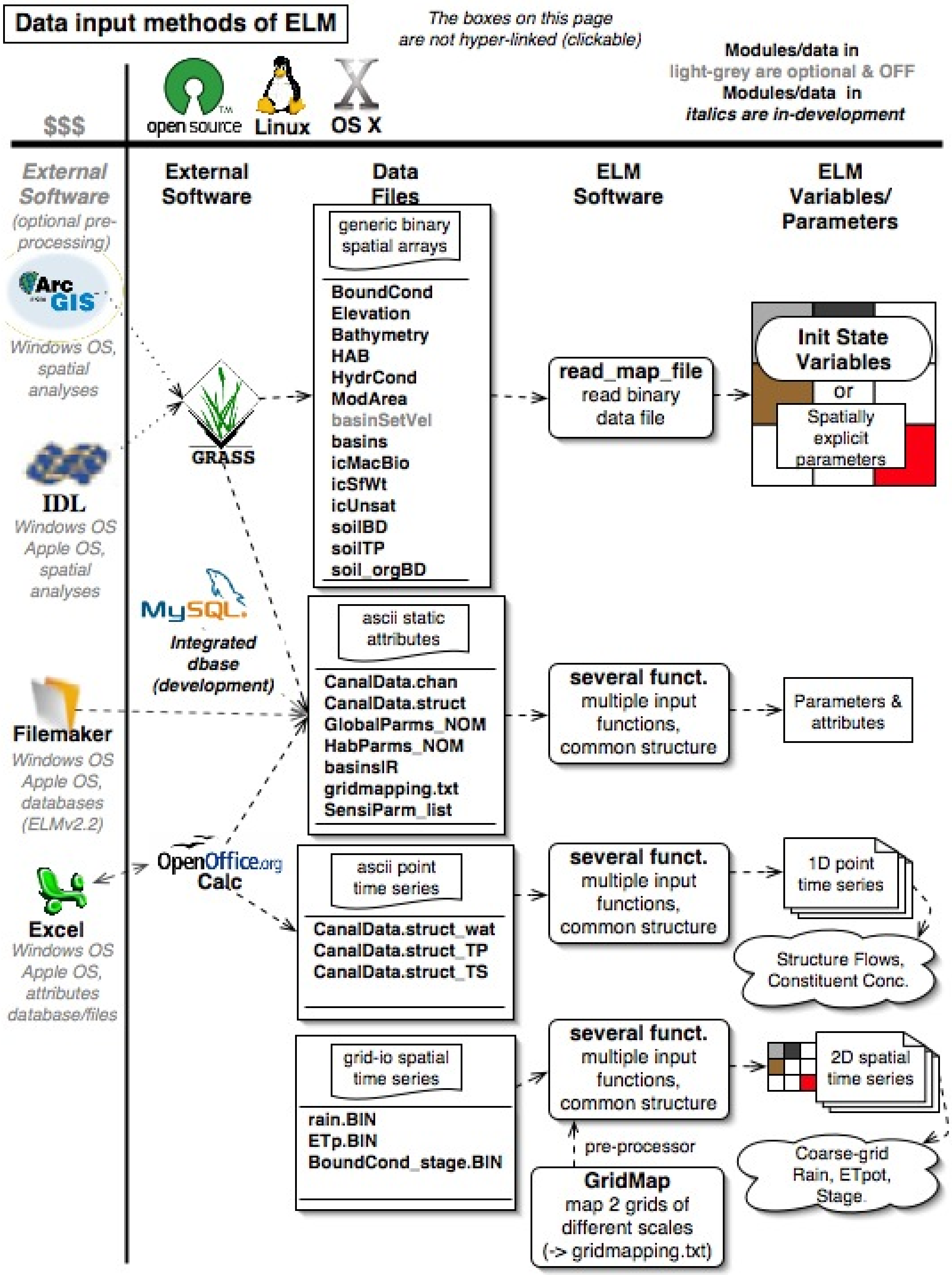
5.4 Data-input modules

Open Source software is all that is necessary to make full use of the ELM project (see User's Guide Chapter). All model input files are either ASCII text (i.e., exported from Open Office spreadsheet databases), generic binary map data (created/read in GRASS or any other spatial tool), or "grid_io" (spatial time series format used in SFWMM input/output, with editing tools freely available). The MySQL relational databases, that will replace Open Office spreadsheet databases[8], have not been completed for the current ELM version. GRASS is the primary GIS tool used for ELM, and is recommended due to its advanced raster GIS capabilities, and the availability of ELM scripts for visualizing input and output data in raster, vector, and point formats.

The Figure on the following page provides an overview of the pre-processing tools and the input methods within the ELM code. The Doxygen-generated source code documentation can be consulted (on ELM web site) for further information on source code input/output methods.

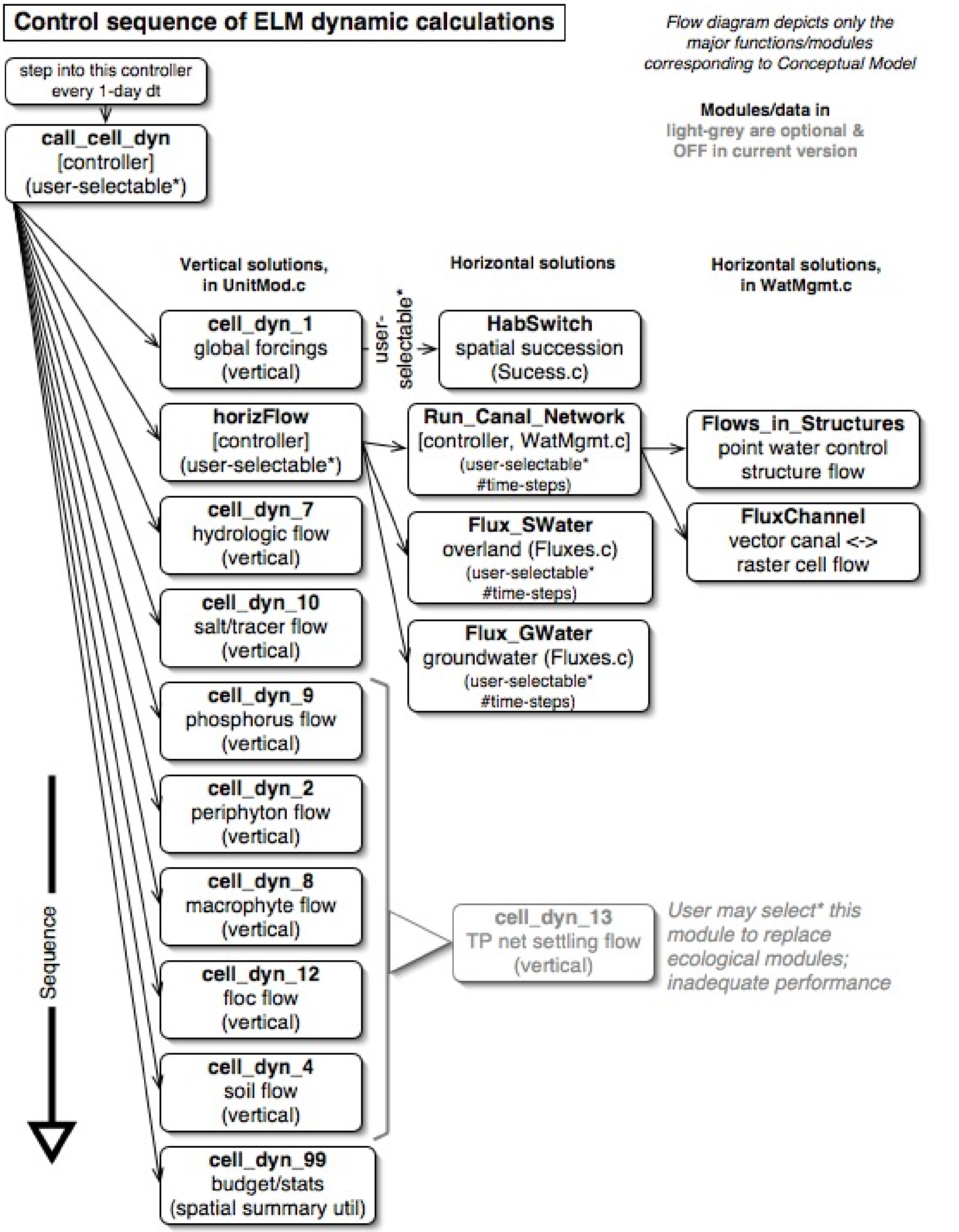
---

8 FileMaker Pro databases were used in prior versions of ELM. The relational database of water control structure attributes remains in FileMaker Pro, but its functionality is not required to use ELM.


Data input methods of ELM
The boxes on this page are not hyper-linked (clickable)
Modules/data in light-grey are optional & OFF
Modules/data in italics are in-development
$$$
open source
Linux
OS X
External Software (optional pre-processing)
External Software
Data Files
ELM Software
ELM Variables/ Parameters
ArcGIS
Windows OS, spatial analyses
IDL
Windows OS Apple OS, spatial analyses
Filemaker
Windows OS Apple OS, databases (ELMv2.2)
Excel
Windows OS Apple OS, attributes database/files
GRASS
MySQL
Integrated dbase (development)
OpenOffice.org Calc
generic binary spatial arrays
BoundCond
Elevation
Bathymetry
HAB
HydrCond
ModArea
basinSetVel
basins
icMacBio
icSfWt
icUnsat
soilBD
soilTP
soil_orgBD
read_map_file
read binary data file
Init State Variables
or
Spatially explicit parameters
ascii static attributes
CanalData.chan
CanalData.struct
GlobalParms_NOM
HabParms_NOM
basinsIR
gridmapping.txt
SensiParm_list
several funct. multiple input functions, common structure
Parameters & attributes
ascii point time series
CanalData.struct_wat
CanalData.struct_TP
CanalData.struct_TS
several funct. multiple input functions, common structure
1D point time series
Structure Flows, Constituent Conc.
grid-io spatial time series
rain.BIN
ETp.BIN
BoundCond_stage.BIN
several funct. multiple input functions, common structure
2D spatial time series
Coarse-grid Rain, ETpot, Stage.
pre-processor
GridMap
map 2 grids of different scales (-> gridmapping.txt)

5.5 Dynamic solutions: sequencing

The “call_cell_dyn” controller function calls dynamic modules in the order (changeable by the user) shown in the diagram below. Each of the dynamic modules is described in a separate section of this Chapter.

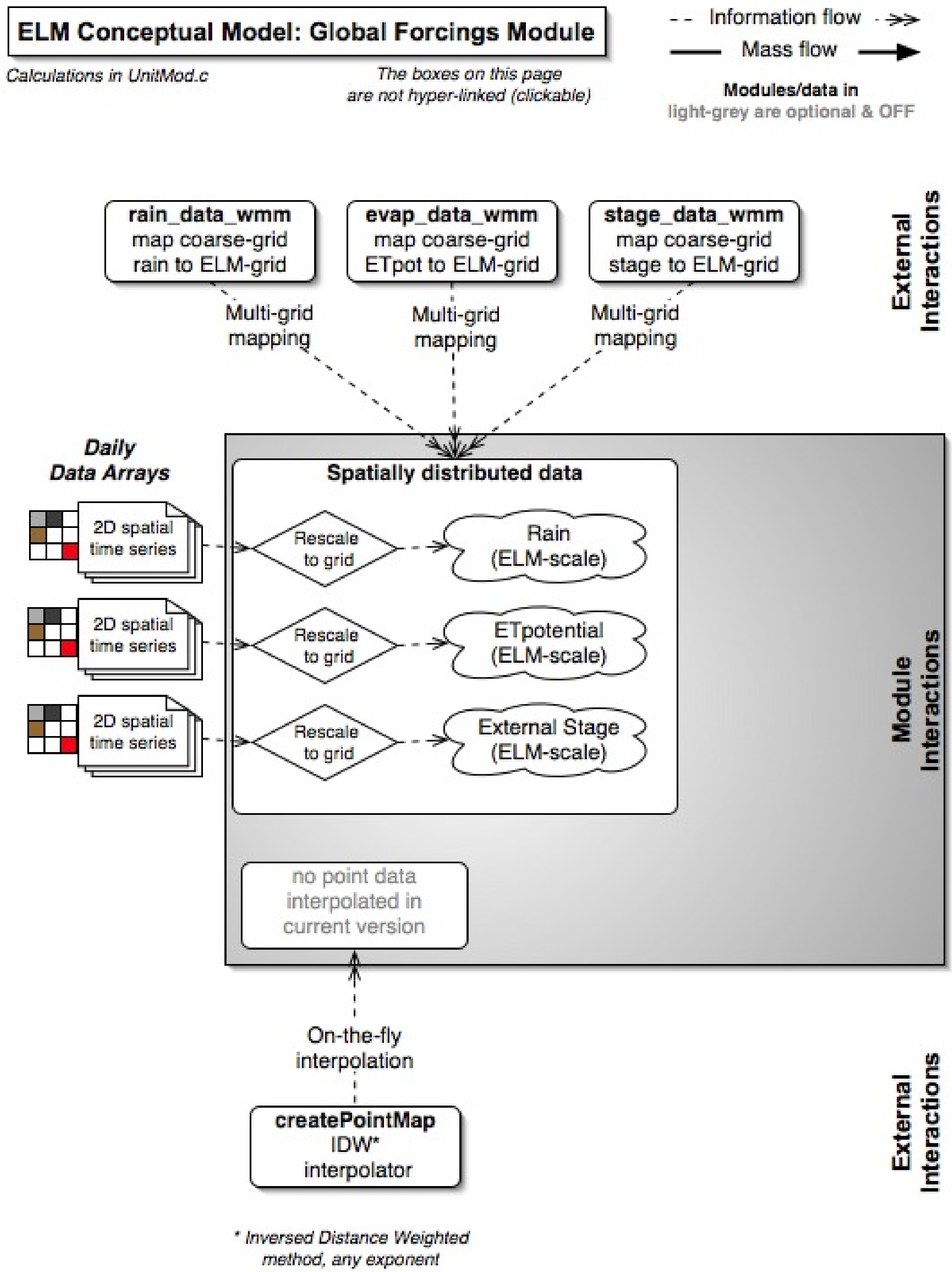


* At run-time, user selects modules to be executed, along with other model attributes

5.6 Vertical solutions

These modules calculate the vertical solutions for all of the physical, chemical, and biological dynamics of the ecological "unit" model (Fitz et al. 1996). That manuscript can be consulted for further background on these active modules, along with other modules that are not used in the current ELM application. See the above/previous diagram on sequencing of these modules.

5.6.1 Globals module



Overview: Globals Module

The Globals Module serves primarily as an data-processing function for meteorological data that are either heterogeneously or homogenously distributed across ELM grid cells, depending on the data type. The call to the habitat succession module is made in this Globals module.

Globals Module Description

Because potential evapotranspiration (ET) is input data instead of being calculated from individual meteorological variables (as done in ELMv2.1), this module serves basically two active functions in the current version. A series of pre-calibrated equations (Nikolov and Zeller 1992) calculate the daily solar radiation incoming to the upper atmosphere, while data-distribution functions provide a daily time series of potential ET and rainfall at the ELM grid scale. The former (radiation) is globally distributed (homogenous) across all grid cells in the model domain. This solar radiation algorithm calculates daily solar radiation at the top of the atmosphere based on julian date, latitude, solar declination, and other factors. The input data of 1) potential ET, 2) rainfall, and 3) stage are input to the ELM at the coarse grid cells of the data source (SFWMM v5.4), and mapped in this module to the grid resolution of the ELM. The call to the habitat-switching function is made in this module.

Globals Module Equations

State Variable update calculations

calculated within spatial loop across model grid rows, columns

function call to habitat switching module
HAB = HabSwitch (ix, iy, SURFACE_WAT, TPtoSOIL, FIREdummy, HAB)

Dependent upon:
1) attribute calculations

none

2) control function calculations

none

3) flux calculations

none

4) attribute calculations, only used in other modules

##Nikolov and Zeller(1992) generic algorithm to calculate SOLRADATMOS (single spatial value that is uniform across model domain, intermediate calculations shown)
DAYJUL = (Mod(TIME,365.0) >0.0) ? (Mod(TIME,365.0)) : (365.0)

DAYLENGTH = AMPL*Sin((DAYJUL-79.0)*0.01721)+12.0

SOLDEC1 = 0.39785*Sin(4.868961+0.017203*DAYJUL
+0.033446*Sin(6.224111+0.017202*DAYJUL))

SOLCOSDEC = sqrt(1.0-SOLDEC1*SOLDEC1)

SOLELEV_SINE = Sin(***GP_LATRAD***)*SOLDEC1+Cos(***GP_LATRAD***)*SOLCOSDEC

SOLALTCORR = (1.0-Exp(-0.014*(***GP_ALTIT***-274.0)/(SOLELEV_SINE*274.0)))

SOLDEC = Arctan(SOLDEC1/sqrt(1.0-SOLDEC1*SOLDEC1))

SOLRISSET_HA1 = -Tan(***GP_LATRAD***)*Tan(SOLDEC)

SOLRISSET_HA = ((SOLRISSET_HA1==0.0)) ? (PI*0.5) : (((SOLRISSET_HA1<0.0)) ? (PI+Arctan(sqrt(1.0-SOLRISSET_HA1*SOLRISSET_HA1)/SOLRISSET_HA1)) : (Arctan(sqrt(1.0-SOLRISSET_HA1*SOLRISSET_HA1)/SOLRISSET_HA1)))

SOLRADATMOS = 458.37*2.0*(1.0+0.033*Cos(360.0/365.0*PI/180.0*DAYJUL)) * (Cos(***GP_LATRAD***)*Cos(SOLDEC)*Sin(SOLRISSET_HA) + SOLRISSET_HA*180.0/(57.296*PI)*Sin(***GP_LATRAD***)*Sin(SOLDEC))

External variables used

total julian day count, GenericDriver.c
TIME

SURFACE_WAT (see Hydrology module)

TPtoSOIL (see Soils module)

FIRE_DIRECT (Fire module not used, fire data not needed)

calculated once during initialization
AMPL = Exp(7.42+0.045****LATRAD****180.0/PI)/3600.0

Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
SOLRADATMOS	attribute	cal/cm^2/d	solar radiation received at the top of the atmosphere
AIR_TEMP	attribute	deg C	Air temperature, daily average at ground level
HAB	state	dimless	Habitat, or vegetation community type (integer attribute, defining database parameter lookups)

Time series forcing data

function call to map rainfall data (tenths of mm/d) to model grid cells
stat=rain_data_wmm(wmm_rain)

function call to map potential ET data (tenths of mm/d) to model grid cells
stat=evap_data_wmm(wmm_evap)

air temperature is constant data in v2.2 only
AIR_TEMP = 25.0

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
GP_ALTIT	global	m	regional altitude of land surface
GP_LATDEG	global	deg.min	regional latitude (degrees.minutes, don't convert min to decimal deg)
GP_LATRAD	global	radians	regional latitude, calculated conversion to radians during

			initialization

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description

none

Intrinsic C or ELM functions

exp(x) = Exp(x) => e raised to the x^{th} power

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

Mod(x,y) = modulus (remainder) of x divided by y

Sin(x) => sine of (x in radians)

Cos(x) => cosine of (x in radians)

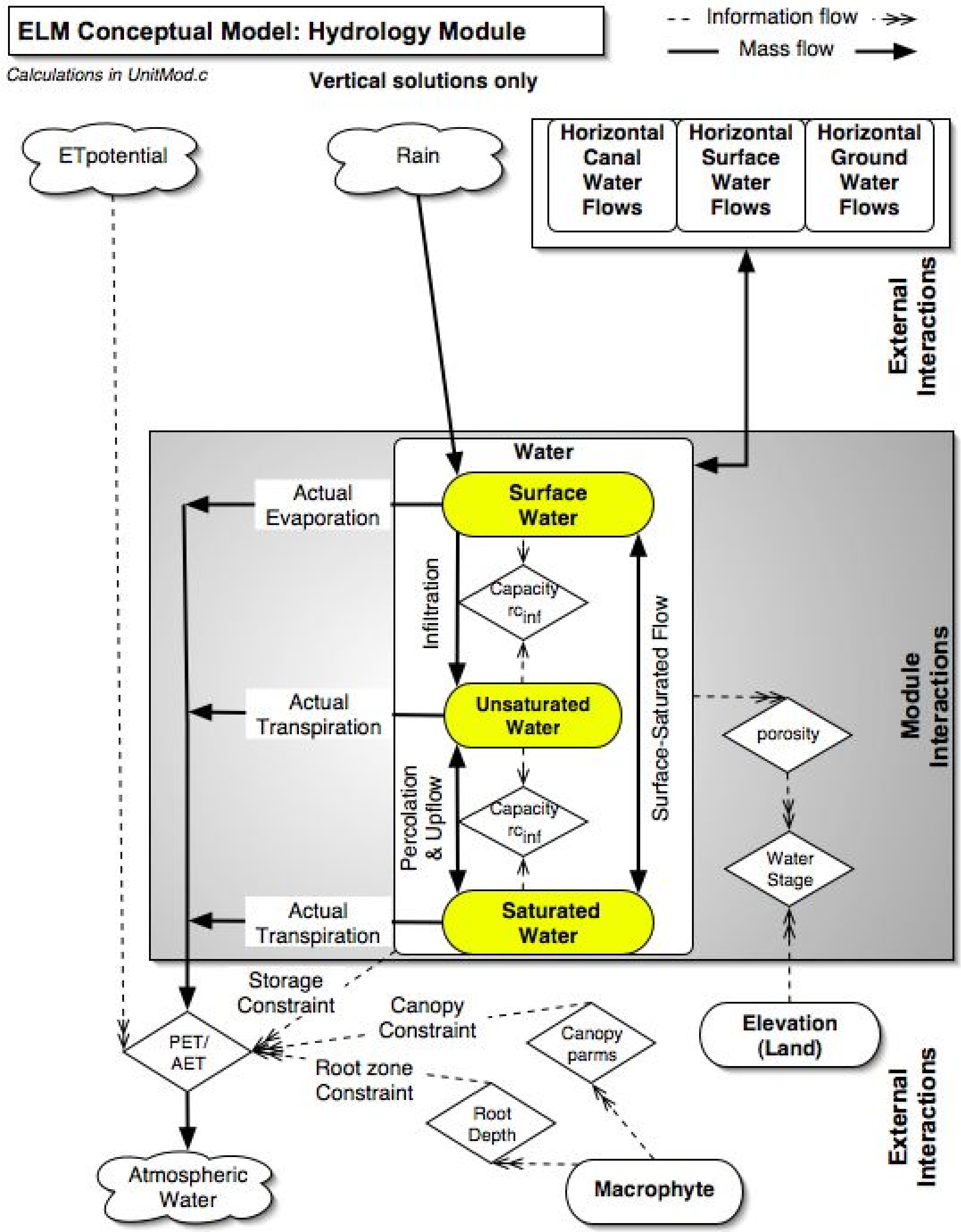
Arctan(x) => arc tangent of (x in radians)

Tan(x) => tangent of (x in radians)

PI => the constant pi

sqrt(x) => square root of (x)

5.6.2 Hydrology module



Overview: Hydrology Module

This Hydrology Module serves primarily to update the grid-cell water storages due to vertical fluxes among surface, unsaturated, and saturated storage state variables. Hydrology is a critical "driver" of the landscape, in that it is necessary to understand and get the water "right" in order to sustain a healthy Everglades. Vertical flows among those storages involve rainfall, evaporation, infiltration, percolation, and transpiration. Hydrology is one of the "fast" processes that can change significantly on time scales on the order of hours, but climate change can produce decadal shifts in dynamics of the regional hydrologic cycle. While rainfall in south Florida is seasonal, it is variable both within seasons and among years. Intense rainfall events are often heterogeneously distributed at local scales; tropical disturbances can deluge the entire region. The pattern of water distribution (hydropattern) across the landscape is driven not only by rainfall inputs and (atmospheric- and macrophyte- mediated) evapotranspiration losses, but is intensively managed via the operations of the water management infrastructure (canals, levees, water control structures, see Water Management Modules). Changes to water depths and flows can alter the habitat because different macrophyte species and algal/periphyton assemblages have distinct hydrologic adaptations. Likewise, changing water depths can alter the soils through increased accretion rates when wet for prolonged periods (i.e., long hydroperiods). On the other hand, soil losses increase with the oxidation occurring under short hydroperiods. This increased soil oxidation increases the nutrient availability surface/soil waters. Soil nutrient chemistry is also affected by water exchanges between surface and soil/sediment water storages, a vertical advective process driven by groundwater losses due to plant transpiration and/or horizontal groundwater flows (Raster Flux Modules).

Hydrology Module Description

Water is held in three state variables: 1) SURFACE_WAT is water that is stored above the sediment/soil surface; 2) UNSAT_WAT is stored in the pore spaces of the sediment/soil complex, but not saturating that zone; and 3) SAT_WAT is water saturating the pore spaces of the sediment/soil complex. Simulating the fluxes among these variables allows the depiction of wet, moist and dry environments. Flux among the variables depends on a variety of processes. Horizontal flow of surface and saturated ground water is simulated in other code modules. We ignore details of processes that occur on a time scale faster than the daily time step, such as vertical movement of a saturated wetting front in infiltration events. The longer-term results of storage in a small landscape can be effectively captured within the day-to-weekly time scale.

Surface water loss to storage in the sediment/soil can occur via two pathways: 1) infiltration from the surface water to an unsaturated soil water zone, based on measured infiltration rates for different soil types, and 2) surface water flow to the saturated water storage at a rate that depends on the rate of water loss in saturated storage. Any remaining surface water is available for evaporation. Surface water evaporation is simulated separately from water loss due to transpiration by plants. Total potential evapotranspiration is input as pre-processed data provided by the SFWMM developers. Loss of water by plant transpiration occurs either from the unsaturated or saturated water storages depending on the presence/absence of roots within the zone.

Vertical fluxes of water occur among all three of the water storage compartments. If surface water is present, and there is available volume in the unsaturated storage of the sediment, then water infiltrates into the unsaturated zone at a rate determined by the infiltration rate for the habitat type. The available capacity of the unsaturated zone is calculated from the porosity and current volume of water in unsaturated storage, which also determines the moisture proportion in unsaturated storage. We assume that the water in unsaturated storage is distributed homogeneously within that zone, ignoring the presence of any wetted front and the heterogeneities associated with processes occurring on faster time scales.

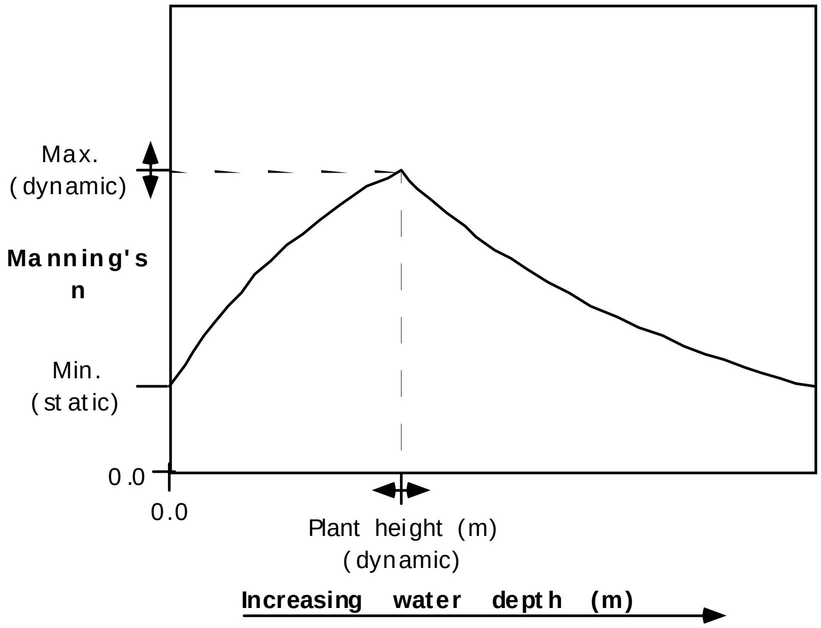
When the sediment is fully saturated, surface water may flow into the saturated layer to replace outflow from the saturated storage at the rate determined by the loss of saturated water. We assume that the rate of vertical movement of water from the surface to the saturated zone is at least as fast as that of losses from saturated storage via horizontal flows and transpiration. Because the unsaturated zone varies in depth, the model determines the relative degree to which surface water flows towards the unsaturated and saturated storage zones in the transition from significant depths of ponded surface water to little surface water and increasing depths of unsaturated storage. This allows for the presence of a vanishingly small unsaturated depth in the presence of small depth of overlying surface water.

Any moisture in excess of field capacity may percolate from the unsaturated storage to saturated storage, determined by the hydraulic conductivity of the sediment for unsaturated conditions. The unsaturated hydraulic conductivity for each habitat (sediment) type is decreased from the saturated hydraulic conductivity as a function of decreasing sediment moisture.

We developed an algorithm that incorporates the effects of dynamic vegetation height and biomass on hydrologic flows (Fitz et al. 1996, Fitz and Sklar 1999):

$$n = n_{\max} - \left| (n_{\max} - n_{\min}) \left(2^{\left(1 - h/mac\right)} - 1 \right) \right|$$

where n is the dynamic Manning's roughness coefficient, n_{min} and n_{max} are the respective minimum and maximum roughness parameters associated with a cell's macrophyte/soil characteristics, h is water depth (m), and mac is the macrophyte height. As shown in the below Figure, this function returns a positive roughness coefficient whose value ranges from a vegetation-free minimum to a maximum at the point of full plant immersion (Petryk et al. 1975). As water depth increases over that of the macrophyte height, the roughness decreases to an asymptote at the baseline sediment roughness (Nalluri and Judy 1989). The roughness coefficient is calculated in this module, for application to spatial fluxes in horizontal solution modules.



The positive relationship of Manning's n with increased depth has been demonstrated by USGS (Everglades-specific) flume and Everglades field studies (Jenter and Schaffranek 1996, Carter et al. 1999, Lee and Carter 1999, 2002). As pointed out by Jenter and Schaffranek (1996), "...for a uniform stand of sawgrass with no litter layer, the value of n increases with flow depth.". We use this relationship in the ELM Manning's n calculation, and it is used by the USGS SICS[9] model. As water depth further increases[10], the ELM algorithm decreases Manning's n as the plants bend and are overtopped by water in a strata with no vegetation resistance.

Hydrology Module Equations

State Variable update calculations

calculated within spatial loop across model grid rows, columns

SURFACE_WAT = SURFACE_WAT + (SF_WT_FROM_RAIN - SF_WT_EVAP - SF_WT_INFILTRATION - SF_WT_TO_SAT_DOWNFLOW) * ***DT***

UNSAT_WATER = UNSAT_WATER + (SF_WT_INFILTRATION - UNSAT_TO_SAT_FL - UNSAT_TRANSP) * ***DT***

SAT_WATER = SAT_WATER + (UNSAT_TO_SAT_FL + SF_WT_TO_SAT_DOWNFLOW - SAT_WT_TRANSP) * ***DT***

Dependent upon:

1) attribute calculations

calculated within spatial loop across model grid rows, columns

SAT_WT_HEAD = SAT_WATER/***HP_HYD_POROSITY***;

UNSAT_DEPTH = SED_ELEV-SAT_WT_HEAD;

---

[9] Southern Inland and Coastal Systems numerical model for the SE region of ENP

[10] To a habitat-specific threshold depth

UNSAT_CAP = UNSAT_DEPTH****HP_HYD_POROSITY***

UNSAT_MOIST_PRP = (UNSAT_CAP>0.0) ? (Min(UNSAT_WATER/UNSAT_CAP,1.0)) : (1.0)

UNSAT_WT_POT = Max(UNSAT_CAP-UNSAT_WATER,0.0)

UNSAT_AVAIL = Max(UNSAT_MOIST_PRP-***field_cap***/***HP_HYD_POROSITY***,0.0)

LAI_eff = (MAC_HEIGHT>0.0) ? (Max(1.0 - SURFACE_WAT/MAC_HEIGHT, 0.0)*MAC_LAI) : (0.0)

f_LAI_eff = exp(-LAI_eff)

2) control function calculations

calculated within spatial loop across model grid rows, columns

SatWat_Root_CF = Exp(-10.0* Max(UNSAT_DEPTH- ***HP_NPHBIO_ROOTDEPTH***,0.0));

HYD_WATER_AVAIL = (UNSAT_DEPTH > ***HP_NPHBIO_ROOTDEPTH***) ? (Max(UNSAT_MOIST_PRP, SatWat_Root_CF)) : (1.0)

MAC_WATER_AVAIL_CF = graph8(0x0,HYD_WATER_AVAIL)

SAT_VS_UNSAT = 1/Exp(100.0*Max((SURFACE_WAT-UNSAT_DEPTH),0.0))

UNSAT_HYD_COND_CF = graph7(0x0,UNSAT_MOIST_PRP)

3) flux calculations

calculated within spatial loop across model grid rows, columns

HYD_EVAP_CALC = wmm_evap * 0.0001* ***GP_calibET***

HYD_TOT_POT_TRANSP = HYD_EVAP_CALC *(1.0-f_LAI_eff);

HYD_SAT_POT_TRANS = HYD_TOT_POT_TRANSP*SatWat_Root_CF;

HYD_UNSAT_POT_TRANS = (UNSAT_DEPTH > ***HP_NPHBIO_ROOTDEPTH***) ? (HYD_TOT_POT_TRANSP*MAC_WATER_AVAIL_CF) : (0.0)

SF_WT_FROM_RAIN = wmm_rain*0.0001

SF_WT_TO_SAT_DOWNFLOW = ((1.0-SAT_VS_UNSAT) *UNSAT_WT_POT****DT***>SURFACE_WAT) ? (SURFACE_WAT/***DT***) : ((1.0-SAT_VS_UNSAT)*UNSAT_WT_POT)

SF_WT_POT_INF = ((SAT_VS_UNSAT* ***HP_HYD_RCINFILT***+ SF_WT_TO_SAT_DOWNFLOW) ****DT***>SURFACE_WAT) ? ((SURFACE_WAT-SF_WT_TO_SAT_DOWNFLOW****DT***)/***DT***) : (SAT_VS_UNSAT****HYD_RCINFILT***)

SF_WT_INFILTRATION = (SF_WT_POT_INF****DT*** > (UNSAT_WT_POT-SF_WT_TO_SAT_DOWNFLOW****DT***)) ? ((UNSAT_WT_POT-SF_WT_TO_SAT_DOWNFLOW****DT***)/***DT***) : (SF_WT_POT_INF)

SFWAT_PR1 = SF_WT_INFILTRATION+SF_WT_TO_SAT_DOWNFLOW

SF_WT_EVAP = ((f_LAI_eff*HYD_EVAP_CALC+SFWAT_PR1)****DT***>SURFACE_WAT) ? ((SURFACE_WAT-SFWAT_PR1****DT***)/***DT***) : (f_LAI_eff*HYD_EVAP_CALC)

UNSAT_PERC = Min(***HP_HYD_RCINFILT****UNSAT_HYD_COND_CF,UNSAT_AVAIL*UNSAT_WATER)

UNSAT_TO_SAT_FL = (UNSAT_PERC****DT*** > UNSAT_WATER) ? (UNSAT_WATER/***DT***) : (UNSAT_PERC)

UNSAT_TRANSP = ((HYD_UNSAT_POT_TRANS+UNSAT_TO_SAT_FL)****DT***>UNSAT_WATER) ? ((UNSAT_WATER-UNSAT_TO_SAT_FL****DT***)/***DT***) : (HYD_UNSAT_POT_TRANS)

SAT_WT_TRANSP = ((HYD_SAT_POT_TRANS)****DT*** > SAT_WATER) ? ((SAT_WATER)/***DT***) : (HYD_SAT_POT_TRANS);

4) attribute calculations, only used in other modules

calculated within spatial loop across model grid rows, columns

mann_height = Max((***GP_mann_height_coef****MAC_HEIGHT)*(***GP_mann_height_coef****MAC_HEIGHT), 0.01)

N_density = Max(***HP_MAC_MAXROUGH*** * MAC_REL_BIOM, ***HP_MAC_MINROUGH***)

HYD_MANNINGS_N = Max(-Abs((N_density- ***HP_MAC_MINROUGH***) *(pow(2.0,(1.0-SURFACE_WAT/mann_height))-1.0)) + N_density, ***HP_MAC_MINROUGH***);

HYD_DOM_ACTWAT_VOL = (Min(***HP_DOM_MAXDEPTH***[HAB],UNSAT_DEPTH)*UNSAT_MOIST_PRP + Max(***HP_DOM_MAXDEPTH***[HAB]-UNSAT_DEPTH, 0.0)* ***HP_HYD_POROSITY***) * ***CELL_SIZE***

HYD_DOM_ACTWAT_PRES = (HYD_DOM_ACTWAT_VOL > ***CELL_SIZE****0.01) ? (1.0) : (0.0)

HYD_SED_WAT_VOL = (SAT_WATER+UNSAT_WATER)****CELL_SIZE***

SFWT_VOL = SURFACE_WAT****CELL_SIZE***

HydTotHd = SAT_WT_HEAD+SURFACE_WAT

H2O_TEMP= AIR_TEMP

External variables used

MAC_HEIGHT (see Macrophyte module)

MAC_LAI (see Macrophyte module)

MAC_REL_BIOM (see Macrophyte module)

AIR_TEMP (see Globals module)

Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
HYD_DOM_ACTWAT_PRES	attribute	dimless	Logical flag (true or false) denoting PRESence of WATer in the DOM_ACTive zone depth (DOM_MAXDEPTH)
HYD_DOM_ACTWAT_VOL	stateConvert	m^3	HYDrologic, water VOLume storage in the DOM_ACTive zone depth (DOM_MAXDEPTH)
HYD_EVAP_CALC	ratePotential	m/d	HYDrologic, total potential EVAPotranspiration (was calculated variable in v2.1, now data input)
HYD_MANNINGS_N	attribute	d/(m^(1/3))	HYDrologic, calculated MANNING'S N surface roughness, (based on empirically-derived surface roughness

			coeficient)
HYD_SAT_POT_TRANS	ratePotential	m/d	HYDrologic, POTential TRANSpiration loss from SATurated water storage
HYD_SED_WAT_VOL	stateConvert	m^3	HYDrologic, WATer VOLume stored in soil/SEDiment storage
HYD_TOT_POT_TRANSP	ratePotential	m/d	HYDrologic, total POTential TRANSpiration loss (from saturated and unsaturated water storages)
HYD_TRANSP	rateActual	m/d	HYDrologic, sum of actual TRANSPiration loss from saturated and unsaturated water storages (reporting purposes only)
HYD_UNSAT_POT_TRANS	ratePotential	m/d	HYDrologic, POTential TRANSpiration loss from UNSATurated water storage
HYD_WATER_AVAIL	controlFunction	dimless	HYDrologic, control function (0-1) of proportion of WATer in upper soil profile that is AVAILable for plant uptake, including unsaturated storage withdrawal, and small capillary withdrawal from saturated storage, depending on relative depths
HydTotHd	stateConvert	m	Hydrologic, Total hydraulic Head (or stage), not used in calculations, only for reporting purposes
MAC_WATER_AVAIL_CF	controlFunction	dimless	empirical data as a (0-1) control function, the proportion (Y) of water available to plants as a function of proportion (0-1) of water available in upper soil profile (X, HYD_WATER_AVAIL (generally, simply 1:1 relationship)
SAT_VS_UNSAT	controlFunction	dimless	control function (0-1), determining relative magnitude of potential surface- to SATurated VS UNSATurated storage flow, having effects under conditions of extremely shallow ponded depths (ca. a couple cm or less)
SAT_WATER	state	m	height of the SATurated WATER storage volume (excluding soil/sediment volume)
SAT_WT_HEAD	stateConvert	m	SATurated WaTer hydraulic HEAD (does not include any overlying surface water)
SAT_WT_TRANSP	rateActual	m/d	actual TRANSPiration loss from SATurated WaTer storage
SatWat_Root_CF	controlFunction	dimless	control function (0-1) that is intermediate calculation used in HYD_WATER_AVAIL
SF_WT_EVAP	rateActual	m/d	actual EVAPoration loss from SurFace WaTer storage
SF_WT_FROM_RAIN	rateA	m/d	RAINfall gain to the SurFace WaTer

	ctual		storage
SF_WT_INFILTRATION	rateActual	m/d	SurFace WaTer loss due to INFILTRATION into the unsaturated storage zone
SF_WT_POT_INF	ratePotential	m/d	SurFace WaTer POTential loss due to INFiltration into the unsaturated storage zone
SF_WT_TO_SAT_DOWNFLOW	rateActual	m/d	SurFace WaTer DOWNFLOW TO SATurated storage
SFWT_VOL	stateConvert	m^3	SurFace WaTer storage VOLume
SURFACE_WAT	state	m	height of the SurFace WaTer storage VOLume
UNSAT_AVAIL	attribute	dimless	proportion (0-1) of UNSATurated water storage in pore space that is AVAILable for gravitational flow (above field capacity)
UNSAT_CAP	attribute	m	potential total storage CAPacity (pore space) in the height of the current UNSATurated zone
UNSAT_DEPTH	stateConvert	m	DEPTH (height) of the UNSATurated zone (including pore space)
UNSAT_HYD_COND_CF	controlFunction	dimless	empirical data as a control function (0-1), the proportion (Y) of maximum vertical water infiltration rate through soil as a function of soil moisture proportion (0-1) (X, UNSAT_MOIST_PRP)
UNSAT_MOIST_PRP	attribute	dimless	MOISTure PRoPortion (0-1) in UNSATurated storage
UNSAT_PERC	ratePotential	m/d	potential PERColation loss from UNSATurated storage to saturated storage
UNSAT_TO_SAT_FL	rateActual	m/d	PERColation loss from UNSATurated storage to saturated storage
UNSAT_TRANSP	rateActual	m/d	actual TRANSPiration loss from UNSATurated water storage
UNSAT_WATER	state	m	height of the UNSATurated WATER storage volume (excluding soil/sediment volume)
UNSAT_WT_POT	attribute	m	UNSATurated WaTer storage POTential storage that is not filled (<= UNSAT_CAP)
H2O_TEMP	attribute	deg C	Temperature of ponded surface water, daily average (=AIR_TEMP in v2.1)

Time series forcing data

wmm_evap (see Globals module, units= tenths of mm/d)

wmm_rain (seeGlobals module, units= tenths of mm/d)

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
GP_mann_height_coef	global	dimless	proportion of height at which macrophyte starts to bend over in flowing systems
GP_calibET	global	dimless	calibration parameter, multiply potential ET input data

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description
HP_HYD_RCINFILT	hab-spec	m/d	Rate of infiltration into the unsaturated water storage zone.
HP_HYD_POROSITY	hab-spec	dimless	Porosity of the aquifer, average from the sediment to base datum. Field capacity = porosity - specific yield; ensure that alterations to porosity and specific yield are consistent in your parameterization. Must be non-zero.
HP_HYD_SPEC_YIELD	hab-spec	dimless	Proportion of total sediment/soil volume, for a given soil type, that represents water able to be drained by gravity. Field capacity = porosity - specific yield; ensure that alterations to porosity and specific yield are consistent in your parameterization.
field_cap = ***HP_HYD_POROSITY*** - ***HP_HYD_SPEC_YIELD***	hab-spec	dimless	Proportion of total sediment/soil volume, for a given soil type, that represents water able to be drained by gravity.
HP_NPHBIO_ROOTDEPTH	hab-spec	m	Depth of roots below the sediment/soil zone (positive value) for the community.
HP_MAC_MAXROUGH	hab-spec	d/(m^(1/3))	The maximum Manning's n roughness associated with present vegetation when fully inundated by water. The relation of the total manning's n to water depth ranges along the continuum from the roughness due to sediment only and roughness imparted by inundation of plants by water depth. Be sure this max value > the minimum roughness coeff.
HP_MAC_MINROUGH	hab-spec	d/(m^(1/3))	The minimum Manning's roughness coefficient for minimal/no vegetation. Be sure this value is less than the roughness coeff for the vegetation.

Intrinsic C or ELM functions

exp(x) = Exp(x) => e raised to the x^{th} power

Max(x,y) => maximum of variable x or y

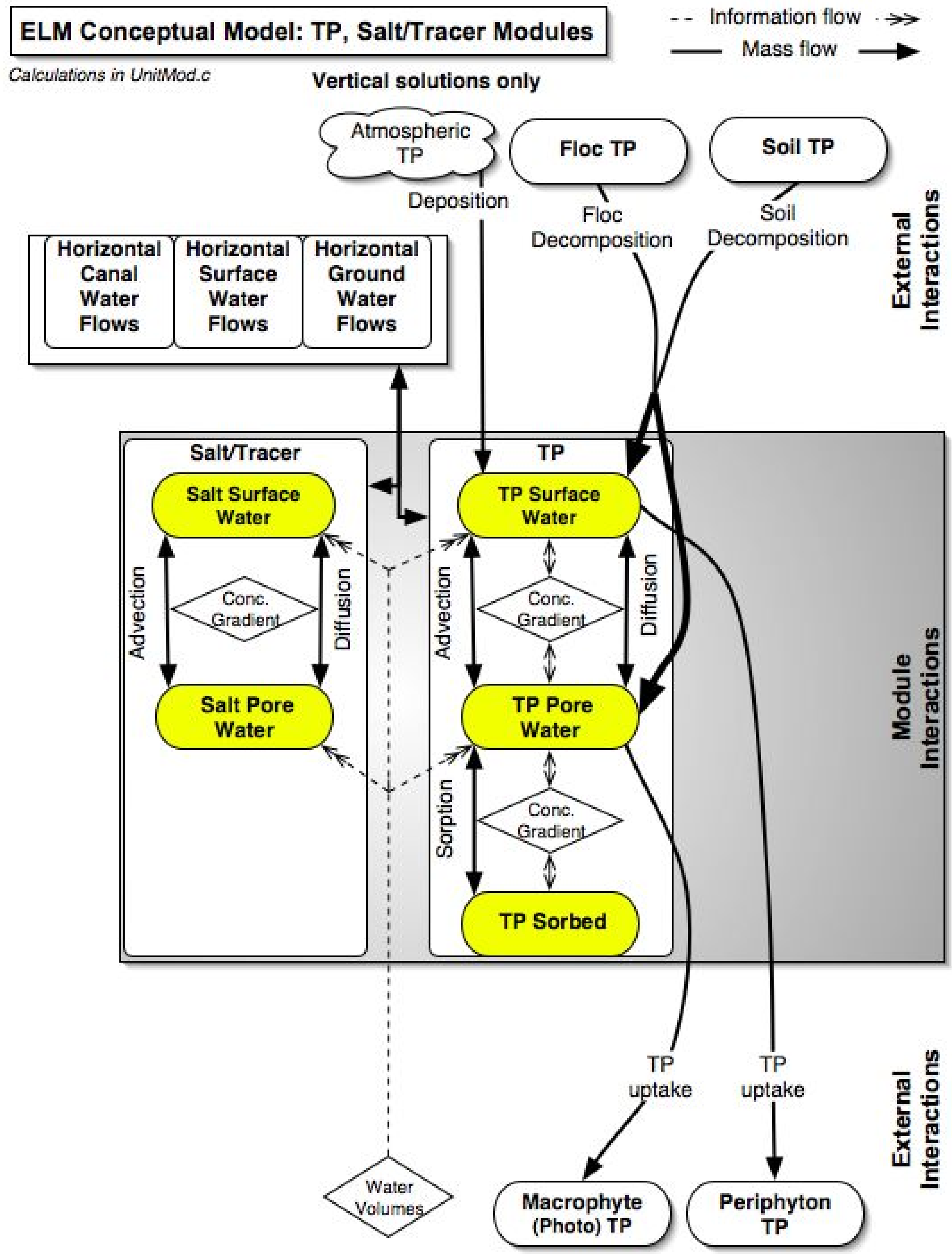
Min(x,y) => minimum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

yVar = graph_(0x0, xVar) => empirical data graph, returning value of yVar as function of current xVar value

pow(x,y) => x raised to the y^{th} power (generally avoided if possible due to execution time of C library)

5.6.3 Phosphorus, salt/tracer modules



Overview: Phosphorus and Salt/Tracer Modules

These Modules serve primarily to update the constituent state variables of phosphorus and salt/tracer, in response to the vertical fluxes among the surface water and sediment/soil storages. Water quality has been responsible for shifts in primary productivity and species composition of macrophyte and periphyton communities, and is another primary "driver" of the landscape at fast (weekly to annual) time scales. Because the predominant "native" Everglades macrophyte and periphyton communities have adapted to oligotrophic (low nutrient) waters, increases in nutrients (i.e., eutrophication) can be detrimental to the structure and the function of those communities. Phosphorus is generally the more limiting nutrient in the freshwater Everglades, while nitrogen (currently inoperative in ELM) tends to govern plant productivity rates in the southern Everglades/Florida bay where estuarine gradients occur. Typically, anthropogenic (manmade) loading of otherwise-limiting nutrients causes ecological imbalance, shifting the structure and function of the ecosystem. Management of flows through water control structures and canals (Water Management Modules) has significantly modified the distribution of these nutrient loads and concentrations across the landscape. Different macrophyte and periphyton communities can uptake nutrients at varying rates (see respective plant Modules), changing the ambient water quality (and changing the plant tissues and growth). As water exchanges among surface and soil/sediment porewaters, the associated nutrient fluxes can alter the microbially mediated rates of soil/sediment decomposition (Soil and Floc Modules), releasing nutrients in inorganic forms that are more available for biotic uptake. Along with nutrient availability, salinity gradients in the southern Everglades/Florida Bay have the potential to modify communities that have adapted to particular environmental conditions.

Phosphorus Module Description

The principal objective of the current Phosphorus module is to simulate vertical atmospheric deposition and the vertical diffusive and advective phosphorus fluxes, as a part of the broader objective of capturing inter-annual and seasonal trends in the regional gradients of water column phosphorus. In the Phosphorus Module, total atmospheric deposition of phosphorus is considered by applying a constant concentration to rainfall to achieve a long term, region-wide annual deposition rate (approximately 27 mg P/m^2/yr in the current model version).

The processes of soil sorption-desorption are calculated using a modified Freunlich equation (Richardson and Vaithiyanathan 1995):

$$P\_sorb(t) = P\_sorb(t-1) + \left(k_{sb} P_{pwat}^{\ 0.8} - P\_sorb(t-1)\right) dt$$

where P_sorb(time) is sorbed phosphorus at time t or time t-1, k_{sb} is the adsorption coefficient (L kg^{-1}), P_{pwat} is the P concentration in the soil pore water (mg L^{-1}), and dt is the time increment.

Uptake by live plants and implicit microbial soil communities are considered in those respective modules. Common to both the Phosphorus and Salt/Tracer Modules are the downward advection of constituents from surface water storage, and the two-way diffusive flux across the soil/sediment and surface water storages. Upflow due to horizontal subsurface flows are accommodated in the integration of surface water and groundwater in the Groundwater Flux Module.

Phosphorus Module Equations

all calculated within spatial loop across model grid rows, columns

State Variable update calculations

TP_SF_WT = TP_SF_WT + (TP_UPFLOW + TP_FR_RAIN - TP_DNFLOW) * ***DT***

TP_SED_WT = TP_SED_WT + (TP_DNFLOW - TP_UPFLOW - TP_SORBTION) * ***DT***

TP_SED_WT_AZ = TP_SED_WT_AZ + (TP_DNFLOW - TP_UPFLOW - TP_SORBTION) * ***DT***

TP_SORB = TP_SORB + (TP_SORBTION) * ***DT***

##TP_SF_WT calculated second time, after first difference equation update of TP_SF_WT
TP_SF_WT = TP_SF_WT - TP_settl * ***DT***

Dependent upon:
1) attribute calculations

TP_SFWT_CONC = (SFWT_VOL > 0.0) ? (TP_SF_WT/SFWT_VOL) : (0.0)

PO4Pconc = Max(TP_SFWT_CONC****GP_PO4toTP*** + 0.001* ***GP_PO4toTP***int,0.0)

TP_SED_CONC = (HYD_SED_WAT_VOL>0.0) ? (TP_SED_WT / HYD_SED_WAT_VOL) : (0.0)

TP_SED_WT_AZ = TP_SED_CONC * TP_Act_to_Tot * HYD_DOM_ACTWAT_VOL

TP_SEDWT_CONCACT =(HYD_DOM_ACTWAT_PRES > 0.0) ? (TP_SED_WT_AZ/HYD_DOM_ACTWAT_VOL) : (TP_SED_CONC)

TP_K = Max(***GP_TP_K_SLOPE****TP_SORBCONC+ ***GP_TP_K_INTER***,0.0)

2) control function calculations

none

3) flux calculations

TP_FR_RAIN = SF_WT_FROM_RAIN****CELL_SIZE**** ***GP_TP_IN_RAIN****0.001

*## 8.64 = sec/day * 1e-4 m^2/cm^2*
TP_UPFLOW_POT = Max((TP_SEDWT_CONCACT-PO4Pconc) * ***GP_TP_DIFFCOEF****8.64/ ***GP_TP_DIFFDEPTH****CELL_SIZE***,0.0)

TP_UPFLOW = ((TP_UPFLOW_POT)****DT***>TP_SED_WT_AZ) ? ((TP_SED_WT_AZ)/***DT***) : (TP_UPFLOW_POT)

TP_SORB_POT = (HYD_DOM_ACTWAT_PRES>0.0) ? (0.001 *(TP_K*(pow(Max(TP_SEDWT_CONCACT,0.0),0.8)) *0.001*(DEPOS_ORG_MAT****CELL_SIZE***+DIM)-TP_SORB)) : (0.0)

if (TP_SORB_POT>0.0) then TP_SORBTION = ((TP_SORB_POT+TP_UPFLOW)****DT***>TP_SED_WT_AZ) ? ((TP_SED_WT_AZ-TP_UPFLOW****DT***)/***DT***) : (TP_SORB_POT)

if (TP_SORB_POT<=0.0) then TP_SORBTION = ((-TP_SORB_POT)****DT***>TP_SORB) ? ((-TP_SORB)/***DT***) : (TP_SORB_POT)

TP_DNFLOW_POT = (SF_WT_INFILTRATION+SF_WT_TO_SAT_DOWNFLOW)****CELL_SIZE****TP_SFWT_CONC + Max((PO4Pconc-TP_SEDWT_CONCACT) * ***GP_TP_DIFFCOEF****8.64/ ***GP_TP_DIFFDEPTH****** CELL_SIZE***,0.0)

TP_DNFLOW = ((TP_DNFLOW_POT)****DT*** > TP_SF_WT) ? ((TP_SF_WT)/***DT***) : (TP_DNFLOW_POT)

4) attribute calculations, only used in other modules

TP_SED_CONC = (HYD_SED_WAT_VOL>0.0) ? (TP_SED_WT / HYD_SED_WAT_VOL) : (0.0)

TP_SEDWT_CONCACT = (HYD_DOM_ACTWAT_PRES > 0.0) ? (TP_SED_WT_AZ/HYD_DOM_ACTWAT_VOL) : (TP_SED_CONC)

TP_SEDWT_CONCACTMG = TP_SEDWT_CONCACT* ***conv_kgTOg***

TP_SORBCONC = ((DEPOS_ORG_MAT****CELL_SIZE*** + DIM)>0.0) ? (TP_SORB* ***conv_kgTOg*** / (DEPOS_ORG_MAT****CELL_SIZE*** + DIM)) : (0.0)

TP_SFWT_CONC = (SFWT_VOL > 0.0) ? (TP_SF_WT/SFWT_VOL) : (0.0)

TP_SFWT_CONC_MG = (SURFACE_WAT > ***GP_DetentZ***) ? (TP_SFWT_CONC* ***conv_kgTOg***) : (0.0)

##Below are calculated after first difference equation update of TP_SF_WT (in later version, may be incorporated into cell_dyn13 instead of this module)

PO4Pconc = Max(TP_SFWT_CONC_MG* ***GP_PO4toTP*** + ***GP_PO4toTPint***,0.0)

nonPO4Pconc = Max(TP_SFWT_CONC_MG-PO4Pconc,0.0)

TPpartic = nonPO4Pconc * (1.0-exp(-nonPO4Pconc/ ***GP_TPpart_thresh***)) *0.001 * SFWT_VOL

TPsettlRat = (SURFACE_WAT > ***GP_DetentZ***) ? (***GP_settlVel***/SURFACE_WAT) : 0.0

TP_settl_pot = TPsettlRat * TPpartic

TP_settl = ((TP_settl_pot)****DT*** > TPpartic) ? ((TPpartic)/***DT***) : (TP_settl_pot)

TP_SFWT_CONC = (SFWT_VOL > 0.0) ? (TP_SF_WT/SFWT_VOL) : (0.0)

TP_SFWT_CONC_MG = (SURFACE_WAT > ***GP_DetentZ***) ? (TP_SFWT_CONC* ***conv_kgTOg***) : (0.0)

External variables used

SFWT_VOL (see Hydrology Module)

HYD_SED_WAT_VOL (see Hydrology Module)

HYD_DOM_ACTWAT_VOL (see Hydrology Module)

HYD_DOM_ACTWAT_PRES (see Hydrology Module)

SF_WT_FROM_RAIN (see Hydrology Module)

SF_WT_INFILTRATION (see Hydrology Module)

SF_WT_TO_SAT_DOWNFLOW (see Hydrology Module)

TP_Act_to_Tot (see Soils Module)

DEPOS_ORG_MAT (see Soils Module)

DIM (see Soils Module)

Phosphorus Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
nonPO4Pconc	attribute	mgP/L	concentration of ~bio-unavailable form of total phosphorus (loosely stated, "non-PO4") storage in water column (note units of mgP/L)
PO4Pconc	attribute	mgP/L	concentration of inorganic PO4 (~bio-available) form of total phosphorus storage in water column (note units of mgP/L)
TP_DNFLOW	rateActual	kgP/d	Total Phosphorus DowNFLOW loss from surface water TP storage to saturated water TP storage via advection and diffusion
TP_DNFLOW_POT	ratePotential	kgP/d	Total Phosphorus DowNFLOW POTential loss from surface water TP storage to saturated water TP storage via advection and diffusion
TP_FR_RAIN	rateActual	kgP/d	Total Phosphorus DowNFLOW gained from atmospheric deposition (via a rainfall TP concentration)
TP_K	attribute	mgP/L	Total Phosphorus K value calculated for Freundlich sorption eqn
TP_SED_CONC	attribute	kgP/m^3	Total Phosphorus CONCentration in entire SEDiment/soil water volume
TP_SED_WT	state	kgP	Total Phosphorus stored in entire SEDiment/soil WaTer volume
TP_SED_WT_AZ	state	kgP	Total Phosphorus stored in Active Zone of SEDiment/soil WaTer volume
TP_SEDWT_CONCACT	attribute	kgP/m^3	Total Phosphorus CONCentration in ACTive SEDiment/soil WaTer volume
TP_SEDWT_CONCACTMG	attribute	mgP/L	Total Phosphorus CONCentration in ACTive SEDiment/soil WaTer volume
TP_settl	rateActual	kgP/d	Total Phosphorus settled (deposited) out of storage in surface water (Everglades Water Quality Model module calc'd differently from ELM phosphorus module)
TP_settl_pot	ratePotential	kgP/d	Total Phosphorus that may potentially be settled (deposited) out of storage in surface water (Everglades Water Quality Model module calc'd differently from ELM phosphorus module)
TP_SF_WT	state	kgP	Total Phosphorus stored in SurFace WaTer volume
TP_SFWT_CONC	attribute	kgP/m^3	Total Phosphorus CONCentration in SurFace WaTer volume
TP_SFWT_CONC_MG	attribute	mgP/L	Total Phosphorus CONCentration in SurFace WaTer volume
TP_SORB	state	kgP	Total Phosphorus storage that is SORBed to sediment/soils

TP_SORB_POT	ratePotential	kgP/d	Total Phosphorus POTential flux of adSORBtion to (positive) or deSORBtion from (negative) sediment/soils (Note the negative values in this flux variable: neg values are not accomodated in default unsigned char map output)
TP_SORBCONC	attribute	gP/kg_soil	Total Phosphorus CONCentration SORBed to (organic and inorganic) soil mass (note units of gP/kg_soil)
TP_SORBTION	rateActual	kgP/d	Total Phosphorus flux of adSORBTION to (positive) or deSORBTION from (negative) sediment/soils
TP_UPFLOW	rateActual	kgP/d	Total Phosphorus UPFLOW gain to surface water TP storage from saturated water TP storage via diffusion (advection handled separately in surface-ground water integration module within fluxes.c source)
TP_UPFLOW_POT	ratePotential	kgP/d	Total Phosphorus UPFLOW POTential gain to surface water TP storage from saturated water TP storage via diffusion (advection handled separately in surface-ground water integration module within fluxes.c source)
TPpartic	attribute	kgP	mass of particulate form of total phosphorus storage in water column (<= mass of nonPO4Pconc)
TPsettlRat	rateActual	1/d	Total Phosphorus settling rate (Everglades Water Quality Model module calc'd differently from ELM phosphorus module)

Time series forcing data

none

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
conv_kgTOg	global	dimless	conversion, kg->g
GP_DetentZ	global	m	detention depth in a grid cell, below which surface flows do not occur
GP_PO4toTP	global	dimless	slope of empirical regression of predicting PO4 from TP from long-term historical data, northern Everglades locations
GP_PO4toTPint	global	mg/l	intercept of empirical regression of predicting PO4 from TP from long-term historical data, northern

			Everglades locations
GP_TP_K_SLOPE	global	dimless	slope for Freundlich soil sorption eqn
GP_TP_K_INTER	global	mg/L	intercept for Freundlich soil sorption eqn
GP_TP_DIFFCOEF	global	cm^2/sec	Phosphorus molecular (surface-soil water) diffusion coefficient.
GP_TP_DIFFDEPTH	global	m	depth of surface-soil water diffusion zone
GP_TP_IN_RAIN	global	mg/L	TP concentration in rainfall (will be switching to new data for versions > ELMv2.2)
GP_TPpart_thresh	global	mg/L	TP conc used for predicting particulate P for settling
GP_settlVel	global	m/d	ELM (NOT EWQM emulation) mean settling velocity of particulate phosphorus (NOT of Total Phosphorus)

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description

none

Intrinsic C or ELM functions

Max(x,y) => maximum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

pow(x,y) => x raised to the yth power (generally avoided if possible due to execution time of C library)

Salt/Tracer Module Description

The principal objective of the current Salt/Tracer module is to simulate the vertical diffusive and advective fluxes of conservative water column constituents, as a part of the broader objective of capturing inter-annual and seasonal trends in the regional gradients of this constituent. In a very simple implementation, this module only considers the downward advection of constituents from surface water storage, and the two-way diffusive flux across the soil/sediment and surface water storages. Upflow due to horizontal subsurface flows are accomodated in the integration of surface water and groundwater in the Groundwater Flux Module. Currently (ELM v2.2), the model considers a single conservative constituent, with the primary focus on the use of Chloride input data as a “conservative” tracer to aid in understanding relative rates of horizontal water flow (see Water Management and Raster Flux Modules) in different parts of the system.

Salt/Tracer Module Equations

all calculated within spatial loop across model grid rows, columns

State Variable update calculations

SALT_SED_WT = SALT_SED_WT + (SALT_SFWAT_DOWNFL - SALT_SED_TO_SF_FLOW) * ***DT***

SALT_SURF_WT = SALT_SURF_WT + (SALT_SED_TO_SF_FLOW - SALT_SFWAT_DOWNFL) * ***DT***

Dependent upon:

1) attribute calculations

SAL_SF_WT_mb = (SFWT_VOL > 0.0) ? (SALT_SURF_WT/SFWT_VOL) : (0.0)

SAL_SED_WT = (HYD_SED_WAT_VOL>0.0) ? (SALT_SED_WT/HYD_SED_WAT_VOL) : (0.0)

2) control function calculations

none

3) flux calculations

*## 8.64 = sec/day * 1e-4 m^2/cm^2*

SALT_SFWAT_DOWNFL_POT = (SF_WT_INFILTRATION + SF_WT_TO_SAT_DOWNFLOW)* ***CELL_SIZE****SAL_SF_WT_mb+ Max((SAL_SF_WT_mb-SAL_SED_WT) * ***GP_TP_DIFFCOEF****8.64/ ***GP_TP_DIFFDEPTH**** ***CELL_SIZE***,0.0)

SALT_SFWAT_DOWNFL = (SALT_SFWAT_DOWNFL_POT****DT***>SALT_SURF_WT) ? (SALT_SURF_WT/***DT***) : (SALT_SFWAT_DOWNFL_POT)

SALT_SED_TO_SF_FLOW_pot = Max((SAL_SED_WT-SAL_SF_WT_mb) * ***GP_TP_DIFFCOEF****8.64/ ***GP_TP_DIFFDEPTH**** ***CELL_SIZE***,0.0)

SALT_SED_TO_SF_FLOW = (SALT_SED_TO_SF_FLOW_pot****DT***>SALT_SED_WT) ? (SALT_SED_WT/***DT***) : (SALT_SED_TO_SF_FLOW_pot)

4) attribute calculations, only used in other modules

none

External variables used

SFWT_VOL (see Hydrology Module)

HYD_SED_WAT_VOL (see Hydrology Module)

SF_WT_INFILTRATION (see Hydrology Module)

SF_WT_TO_SAT_DOWNFLOW (see Hydrology Module)

Salt/Tracer Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
SAL_SED_WT	attribute	kgSalt/ m^3	SALinity in SEDiment/soil WaTer storage (can be any conservative solute w/ consistent units - salt/tracer does not affect any other calculation in v2.2)
SAL_SF_WT	attribute	kgSalt/ m^3	SALinity in SurFace WaTer storage (can be any conservative solute w/

			consistent units - salt/tracer does not affect any other calculation in v2.2)
SALT_SED_TO_SF_FLOW	rateActual	kgSalt/d	SALT FLOW from SEDiment/soil water storage TO SurFace water storage via diffusion (advection handled separately in surface-ground water integration module within fluxes.c source)
SALT_SED_TO_SF_FLOW_pot	ratePotential	kgSalt/d	SALT FLOW potential from SEDiment/soil water storage TO SurFace water storage via diffusion (advection handled separately in surface-ground water integration module within fluxes.c source)
SALT_SED_WT	state	kgSalt	SALT mass in SEDiment/soil WaTer storage (can be any conservative solute w/ consistent units - salt/tracer does not affect any other calculation in v2.2)
SALT_SFWAT_DOWNFL	rateActual	kgSalt/d	SALT DOWNFLow from SurFace WATer storage to sediment/soil water storage via diffusion and advection
SALT_SFWAT_DOWNFL_POT	ratePotential	kgSalt/d	SALT DOWNFLow POTential from SurFace WATer storage to sediment/soil water storage via diffusion and advection
SALT_SURF_WT	state	kgSalt	SALT mass in SURFace WaTer storage (can be any conservative solute w/ consistent units - salt/tracer does not affect any other calculation in v2.2)

Time series forcing data
none

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	m/d	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
GP_TP_DIFFCOEF	global	cm^2/sec	Phosphorus molecular (surface-soil water) diffusion coefficient.
GP_TP_DIFFDEPTH	global	m	depth of surface-soil water diffusion zone

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description

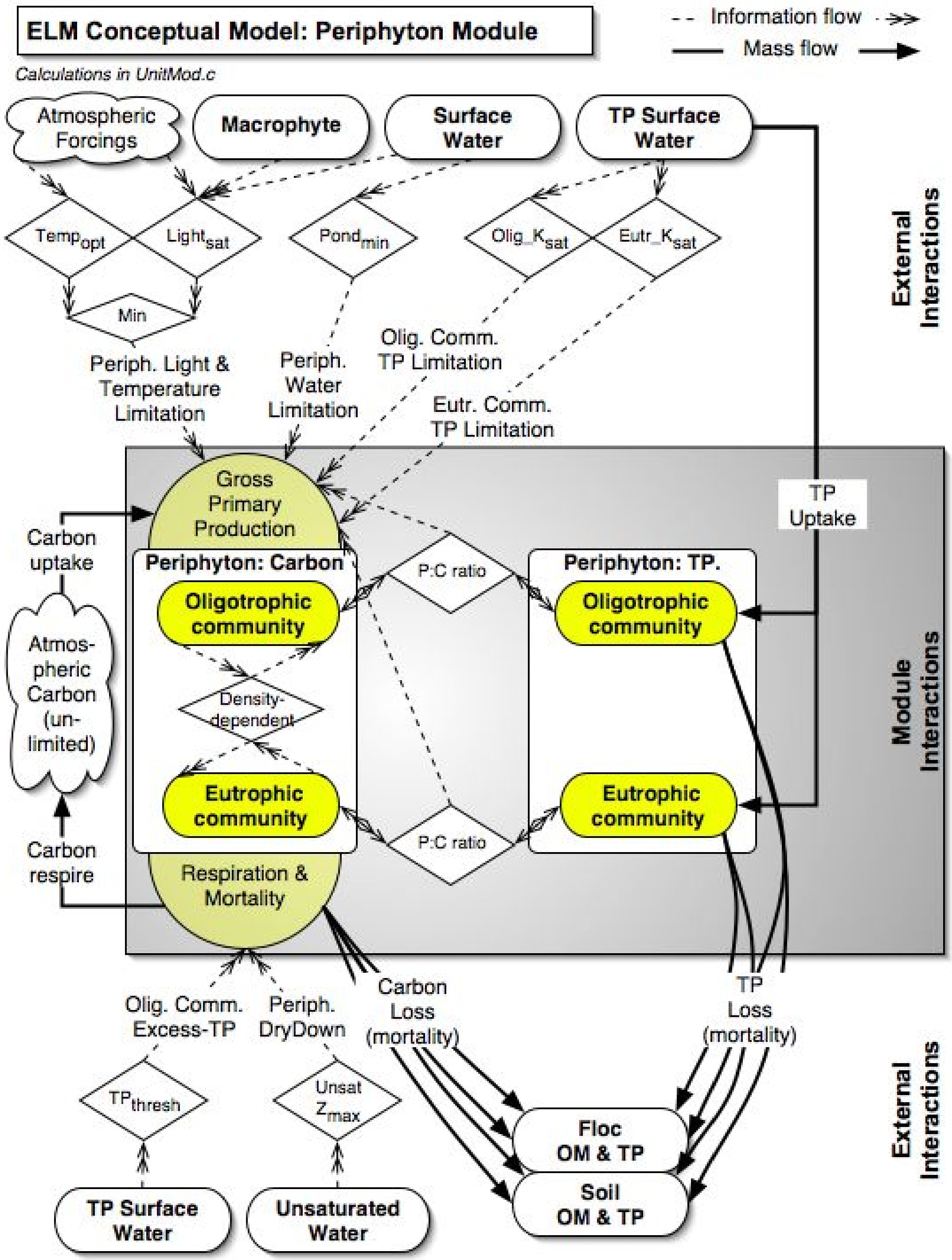
none

Intrinsic C or ELM functions

Max(x,y) => maximum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

5.6.4 Periphyton module



Overview: Periphyton Module

Periphyton are found attached to macrophyte stems, floating as mats in the water column, and as a benthic layer on top of the soil. Long considered an integral part of the animal food web, periphyton respond rapidly to changes in water quality and hydroperiod. Like macrophytes, "native" periphyton are adapted to oligotrophic (low nutrient) conditions, while a variety of other periphyton are common in eutrophic (high nutrient) waters. Another important control on periphyton and algae is light availability: at intermediate and high plant densities (such as in high nutrient areas), emergent marsh macrophytes shade periphyton, and (to some extent) prevent healthy communities from developing. Capable of senescing during dry periods and coming back to high growth levels upon rehydration, there are a variety of different types of periphyton species & communities, depending on the subregion of the Everglades and its local environmental conditions.

Periphyton Module Description

The general form of the equations that describe changes to a periphyton carbon stock is:

$$S(t) = S(t-1) + (P - R - M)dt,$$

where *S(time)* is the standing stock of periphyton (g C m^{-2}) at time *t* or *t-1*, *P* is the gross primary production gain (g C m^{-2} d^{-1}), *R* is the respiration loss (g C m^{-2} d^{-1}), *M* is the mortality loss (g C m^{-2} d^{-1}), and *dt* is the time interval (days). The actual rates are products of the periphyton stock and maximum specific rates that are constrained by control functions:

$$P = S(t-1) \cdot P_{max} CF_P$$

$$R = S(t-1) \cdot R_{max} CF_R$$

$$M = S(t-1) \cdot M_{max} CF_M$$

where P_{max}, R_{max}, and M_{max} are the maximum specific rates (d^{-1}) of, respectively, gross primary production, respiration, and mortality; the CF_P, CF_R, and CF_M are the (dimensionless, 0 – 1) control functions constraining gross production, respiration, and mortality, respectively.

The control function constraining gross primary production includes a density-dependent feedback and a control function involving several environmental parameters. This combined control function is a multiplicative expression of relative effects of light intensity (e.g., macrophyte shading), temperature (seasonality), and nutrient availability.

The dimensionless control function due to light intensity in the water column is based on Steele's (1965) photoinhibition formulation integrated over depth (Bowie et al. 1985). The temperature control function (Jorgensen 1976) describes the biological responses to temperature relative to a temperature optimum and a minimum. Whereas earlier ELM versions (Fitz et al. 1996, Fitz and Sklar 1999) quantified nutrient limitation using Monod half-saturation kinetics, this relationship appeared to behave inadequately in the oligotrophic conditions of much of the Everglades, apparently with excessive constraint on growth under those ambient conditions. There is evidence that phosphatase activity of the periphyton assemblage tends to increase under low nutrient conditions (Newman et al. 2003), thus potentially making phosphorus less limiting and deviating from Monod kinetics. Moreover, while some experimental data existed for half-saturation values of periphyton (Scinto and Reddy submitted) in laboratory settings, there was little information available on growth responses at low nutrient concentrations. Our alternative nutrient control function formulation uses an exponential function, and a relationship to the parameter whose definition remains related to saturation kinetic experiments.

The periphyton module considers two communities of periphyton[11]: those adapted to oligotrophic ("calcareous") and eutrophic ("non-calcareous") conditions such as those observed along Everglades nutrient gradients (McCormick et al. 1996). Both periphyton communities are simulated with the same form of dynamic equations, but have different nutrient limitation parameters, different mortality responses to elevated phosphorus concentrations, and have simple density-dependent inter-community competition.

Periphyton Module Equations

all calculated within spatial loop across model grid rows, columns

State Variable update calculations

NC_ALG = NC_ALG + (NC_ALG_GPP - NC_ALG_RESP - NC_ALG_MORT) * DT

C_ALG = C_ALG + (C_ALG_GPP - C_ALG_RESP - C_ALG_MORT) * DT

Dependent upon:

1) attribute calculations

ALG_REFUGE = ***HP_ALG_MAX**** ***GP_ALG_REF_MULT***

ALG_SAT = ***HP_ALG_MAX****0.9

NC_ALG_AVAIL_MORT = Max(NC_ALG-ALG_REFUGE,0)

C_ALG_AVAIL_MORT = Max(C_ALG-ALG_REFUGE,0)

bio-avail P (PO4) is calc'd from TP, using pre-processed regression for predicting PO4 from TP

assume that periphyton (microbial) alkaline phosphatase activity keeps PO4 at least 10% of TP conc

PO4Pconc = Max(TP_SFWT_CONC_MG* ***GP_PO4toTP*** + ***GP_PO4toTPint***, 0.10 * TP_SFWT_CONC_MG)

light, water, temperature controls apply to both calc and non-calc

ALG_LIGHT_EXTINCT = ***GP_alg_light_ext_coef***

algal self-shading implicit in density-dependent constraint function later

ALG_INCID_LIGHT = SOLRADGRD*Exp(-MAC_LAI* ***GP_ALG_SHADE_FACTOR***)

Z_extinct = SURFACE_WAT*ALG_LIGHT_EXTINCT

I_ISat = ALG_INCID_LIGHT/***GP_ALG_LIGHT_SAT***

2) control function calculations

averaged over whole water column (based on Steele 1965)

---

[11] The names of the periphyton state variables are rooted in the term "algae", originating from the generalized nature of the module that was developed for algal communities. While periphyton are actually assemblages of microbial and algal biota, the aggregate, net-carbon fixing behavior of this assemblage is explicitly considered in its parameterization. Similarly, the somewhat archaic identifiers of "calcareous" and "non-calcareous" are more properly described as oligotrophic and eutrophic communities, as the calcitic attributes of the periphyton are not considered in the model.

ALG_LIGHT_CF = (Z_extinct > 0.0) ? (2.718/Z_extinct * (Exp(-I_ISat * Exp(-Z_extinct)) - Exp(-I_ISat))) : (I_ISat*Exp(1.0-I_ISat))

low-water growth constraint ready for something better based on data
ALG_WAT_CF = (SURFACE_WAT>0.0) ? (1.0) : (0.0)

Jorgensen 1976; 5 deg C is minimum temperature parameter
ALG_TEMP_CF = Exp(-2.3 * ABS((H2O_TEMP- ***GP_ALG_TEMP_OPT***)/(***GP_ALG_TEMP_OPT***-5.0)))

min_litTemp = Min(ALG_LIGHT_CF,ALG_TEMP_CF)

the 2 communities have same form of growth response to avail phosphorus
NC_ALG_NUT_CF = Exp(-***GP_alg_uptake_coef*** * Max(***GP_NC_ALG_KS_P***-PO4Pconc, 0.0)/ ***GP_NC_ALG_KS_P***)

C_ALG_NUT_CF = Exp(-***GP_alg_uptake_coef*** * Max(***GP_C_ALG_KS_P***-PO4Pconc, 0.0)/ ***GP_C_ALG_KS_P***)

the form of the control function assumes that at very low P conc, the alkaline phosphatase activity of the microbial assemblage scavenges P, maintaining a minimum nutrient availability to community
NC_ALG_PROD_CF = Min(min_litTemp,ALG_WAT_CF)*Max(NC_ALG_NUT_CF, ***alg_alkP_min***)

C_ALG_PROD_CF = Min(min_litTemp,ALG_WAT_CF)*Max(C_ALG_NUT_CF, ***GP_alg_alkP_min***)

3) flux calculations

NC_ALG_RESP_POT = (UNSAT_DEPTH> ***GP_algMortDepth***) ? (0.0) : (***GP_ALG_RC_RESP****ALG_TEMP_CF*NC_ALG_AVAIL_MORT)

C_ALG_RESP_POT = (UNSAT_DEPTH> ***GP_algMortDepth***) ? (0.0) : (***GP_ALG_RC_RESP****ALG_TEMP_CF *C_ALG_AVAIL_MORT)

NC_ALG_RESP = (NC_ALG_RESP_POT*DT>NC_ALG_AVAIL_MORT) ? (NC_ALG_AVAIL_MORT/DT) : (NC_ALG_RESP_POT)

C_ALG_RESP = (C_ALG_RESP_POT*DT>C_ALG_AVAIL_MORT) ? (C_ALG_AVAIL_MORT/DT) : (C_ALG_RESP_POT)

this is the threshold control function that increases calcareous/native periph mortality (likely due to loss of calcareous sheath) as P conc. increases
C_ALG_thresh_CF = Min(exp(***GP_alg_R_accel****Max(TP_SFWT_CONC_MG- ***GP_C_ALG_threshTP***,0.0)/***GP_C_ALG_threshTP***), 100.0)

NC_ALG_MORT_POT = (UNSAT_DEPTH>***GP_algMortDepth***) ? (NC_ALG_AVAIL_MORT* ***GP_ALG_RC_MORT_DRY***) : (NC_ALG_AVAIL_MORT* ***GP_ALG_RC_MORT***)

C_ALG_MORT_POT = (UNSAT_DEPTH> ***GP_algMortDepth***) ? (C_ALG_AVAIL_MORT* ***GP_ALG_RC_MORT_DRY***) : (C_ALG_thresh_CF * C_ALG_AVAIL_MORT* ***GP_ALG_RC_MORT***)

NC_ALG_MORT = ((NC_ALG_MORT_POT+NC_ALG_RESP)*DT>NC_ALG_AVAIL_MORT) ? ((NC_ALG_AVAIL_MORT-NC_ALG_RESP*DT)/DT) : (NC_ALG_MORT_POT)

C_ALG_MORT = ((C_ALG_MORT_POT+C_ALG_RESP)*DT>C_ALG_AVAIL_MORT) ? ((C_ALG_AVAIL_MORT-C_ALG_RESP*DT)/DT) : (C_ALG_MORT_POT)

gross production of the 2 communities, with density constraint on both noncalc and calc, competition effect accentuated by calc algae
NC_ALG_GPP = NC_ALG_PROD_CF* ***GP_ALG_RC_PROD****NC_ALG * Max((1.0-(***GP_AlgComp****C_ALG+NC_ALG)/ ***HP_ALG_MAX***),0.0)

C_ALG_GPP = C_ALG_PROD_CF* ***GP_ALG_RC_PROD****C_ALG * Max((1.0-(C_ALG+NC_ALG)/ ***HP_ALG_MAX***),0.0)

P uptake is dependent on available P and is relative to a maximum P:C ratio for the tissue

NC_ALG_GPP_P = NC_ALG_GPP * ***GP_ALG_PC*** * NC_ALG_NUT_CF * Max(1.0-NC_ALG_PC/ ***GP_ALG_PC***, 0.0)

C_ALG_GPP_P = C_ALG_GPP * ***GP_ALG_PC*** * C_ALG_NUT_CF * Max(1.0-C_ALG_PC/ ***GP_ALG_PC***, 0.0)

check for available P mass (the nutCF does not) (unit conversion to g P)

PO4P = Min(PO4Pconc * SFWT_VOL, 1000.0*TP_SF_WT)

reduc = ((NC_ALG_GPP_P+C_ALG_GPP_P) > 0) ? (PO4P / ((NC_ALG_GPP_P+C_ALG_GPP_P)*CELL_SIZE*DT)) : (1.0)

can have high conc, but low mass of P avail, in presence of high peri biomass and high demand, reduce the production proportionally if excess demand is found

if (reduc < 1.0) NC_ALG_GPP = NC_ALG_GPP * reduc

if (reduc < 1.0) NC_ALG_GPP_P = NC_ALG_GPP_P * reduc

if (reduc < 1.0) C_ALG_GPP = C_ALG_GPP * reduc

if (reduc < 1.0) C_ALG_GPP_P = C_ALG_GPP_P * reduc

4) phosphorus associated with carbon stocks & flows

mortPot = NC_ALG_MORT * NC_ALG_PC

NC_ALG_MORT_P = (mortPot*DT>NC_ALG_P) ? (NC_ALG_P/DT) : (mortPot)

mortPot = C_ALG_MORT * C_ALG_PC

C_ALG_MORT_P = (mortPot*DT>C_ALG_P) ? (C_ALG_P/DT) : (mortPot)

NC_ALG_P = NC_ALG_P + (NC_ALG_GPP_P - NC_ALG_MORT_P) * DT

C_ALG_P = C_ALG_P + (C_ALG_GPP_P - C_ALG_MORT_P) * DT

default to 3% of max P:C

NC_ALG_PC = (NC_ALG>0.0) ? (NC_ALG_P/ NC_ALG) : (***GP_ALG_PC*** * 0.03)

C_ALG_PC = (C_ALG>0.0) ? (C_ALG_P/ C_ALG) : (***GP_ALG_PC*** * 0.03)

gP/m2 => kg P

TP_SFWT_UPTAK = (NC_ALG_GPP_P+C_ALG_GPP_P)*0.001*CELL_SIZE

TP_SF_WT = TP_SF_WT - TP_SFWT_UPTAK * DT

TP_SFWT_CONC = (SFWT_VOL > 0.0) ? (TP_SF_WT/SFWT_VOL) : (0.0)

used for reporting and other modules to evaluate P conc when water is present

TP_SFWT_CONC_MG = (SURFACE_WAT > DetentZ) ? (TP_SFWT_CONC*1000.0) : (0.0)

External variables used

TP_SF_WT (see TP/Salt module)

TP_SFWT_CONC_MG (see TP/Salt module)

SOLRADGRD (see Globals module)

MAC_LAI (see Macrophyte module)

SURFACE_WAT (see Hydrology module)

SFWT_VOL (see Hydrology module)

UNSAT_DEPTH (see Hydrology module)

H2O_TEMP (see Hydrology module)

Periphyton Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
C_ALG_AVAIL_MORT	attribute	gC/m^2	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) biomass AVAILable for MORTality losses
NC_ALG_AVAIL_MORT	attribute	gC/m^2	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) biomass AVAILable for MORTality losses
ALG_LIGHT_CF	controlFunction	dimless	total periphyton (generalized, ALGae) growth Control Function (0-1) of degree of LIGHT limitation
ALG_TEMP_CF	controlFunction	dimless	total periphyton (generalized, ALGae) growth Control Function (0-1) of degree of TEMPerature limitation
ALG_WAT_CF	controlFunction	dimless	total periphyton (generalized, ALGae) growth Control Function (0-1) of degree of WATer limitation
C_ALG_NUT_CF	controlFunction	dimless	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) growth Control Function (0-1) of degree of NUTrient limitation
C_ALG_PROD_CF	controlFunction	dimless	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) growth Control Function (0-1) of degree of combined limitations on gross carbon primary PRODuction
NC_ALG_NUT_CF	controlFunction	dimless	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) growth Control Function (0-1) of degree of NUTrient limitation
NC_ALG_PROD_CF	controlFunction	dimless	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) growth Control Function (0-1) of degree of combined limitations on PRODuction
ALG_INCID_LIGHT	forcing	cal/cm^2/d	for ALGal growth, INCIDint LIGHT intensity reaching the water surface through macrophyte canopy
TP_SFWT_UPTAK	rateActual	kgP/d	Total Phosphorus UPTAKe from SurFace WaTer due to periphyton primary production
C_ALG_GPP	rateActual	gC/m2/d	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) Gross Primary Production gains
C_ALG_MORT	rateActual	gC/m2/d	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) MORTality losses
C_ALG_NPP	rateActual	gC/m2/d	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) Net

			Primary Production gains
C_ALG_RESP	rateActual	gC/m2/d	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) RESPiration losses
NC_ALG_GPP	rateActual	gC/m2/d	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) Gross Primary Production gains
NC_ALG_MORT	rateActual	gC/m2/d	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) MORTality losses
NC_ALG_NPP	rateActual	gC/m2/d	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) Net Primary Production gains
NC_ALG_RESP	rateActual	gC/m2/d	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) RESPiration losses
C_ALG_MORT_POT	ratePotential	gC/m2/d	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) MORTality POTential losses
C_ALG_RESP_POT	ratePotential	gC/m2/d	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) RESPiration POTential losses
NC_ALG_MORT_POT	ratePotential	gC/m2/d	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) MORTality POTential losses
NC_ALG_RESP_POT	ratePotential	gC/m2/d	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) RESPiration POTential losses
C_ALG	state	gC/m^2	oligotrophic periphyton (archaic, Calcareous, generalized, ALGae) biomass
NC_ALG	state	gC/m^2	eutrophic periphyton (archaic, NonCalcareous, generalized, ALGae) biomass
ALG_REFUGE	static	gC/m^2	total periphyton (generalized, ALGae) biomass REFUGE, below which resp/mortality losses do not occur (static, set= ***ALG_REF_MULT*** * ***ALG_MAX***[habitat] parameters)
ALG_LIGHT_EXTINCT	static	1/m	for ALGal growth, LIGHT EXTINCTion through suspended particles etc in surface water column (STATIC, set= ***alg_light_ext_coef***)

Time series forcing data

none

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell

conv_kgTOg	global	dimless	conversion, kg to g
GP_alg_alkP_min	global	dimless	minimum possible constraint level (0-1) on phosphorus uptake and growth; value>0 indicative of non-zero nutrient limitation due to APActivity
GP_alg_light_ext_coef	global	dimless	light extinction parameter, currently used to fully define (statically) extinction
GP_ALG_LIGHT_SAT	global	cal/cm^2/d	Saturating light intensity for algal photosyn (langley/d = cal/cm^2 per day)
GP_ALG_PC	global	gP/gC	Initial phophorus:carbon ratio in all algae/periphyton
GP_alg_R_accel	global	dimless	acceleration of mortality (via assumed loss of calcareous sheath) of oligotrophic community under high phosphorus conditions
GP_ALG_RC_MORT	global	1/d	Baseline specific rate of algal (periphyton) mortality. Note that this is in the presence of water.
GP_ALG_RC_MORT_DRY	global	1/d	Specific mortality rate of benthic algae (periphyton) in "drydown" conditions.
GP_ALG_RC_PROD	global	1/d	Maximum specific rate observed/attainable of algal (periphyton) gross primary production.
GP_ALG_RC_RESP	global	1/d	Max specific rate of algal respiration.
GP_ALG_REF_MULT	global	dimless	proportion of max attainable periphyton biomass, defining a refuge density (from losses)
GP_ALG_SHADE_FACTOR	global	dimless	calibration parm to modify LAI in shading fcn
GP_ALG_TEMP_OPT	global	deg C	Optimal temperature for algal primary production (degrees C). Also used in respiration control.
GP_alg_uptake_coef	global	dimless	parameter for exp function describing uptake kinetics
GP_AlgComp	global	dimless	algal density-dep competition, with parameter >1.0 increasing competitive "ability" of oligotrophic periphyton
GP_algMortDepth	global	m	depth of the unsat zone below which accelerated "drydown" alg mort occurs
GP_C_ALG_KS_P	global	mg/L	half-saturation conc of avail phosphorus for uptake kinetics, oligotrophic (was calcareous) periph
GP_C_ALG_threshTP	global	mg/L	TP conc above which oligotrophic (was calcareous) periphyton have elevated mortality (via assumed loss of calcareous sheath)
GP_NC_ALG_KS_P	global	mg/L	half-saturation conc of avail phosphorus for uptake kinetics, eutrophic (was non-calcareous)
GP_PO4toTP	global	dimless	slope of empirical regression of predicting PO4 from TP from long-term historical data, northern

			Everglades locations
GP_PO4toTPint	global	mg/l	intercept of empirical regression of predicting PO4 from TP from long-term historical data, northern Everglades locations

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description
HP_ALG_MAX	habspec	gC/m^2	Maximum attainable (observed) algal biomass density.

Intrinsic C or ELM functions

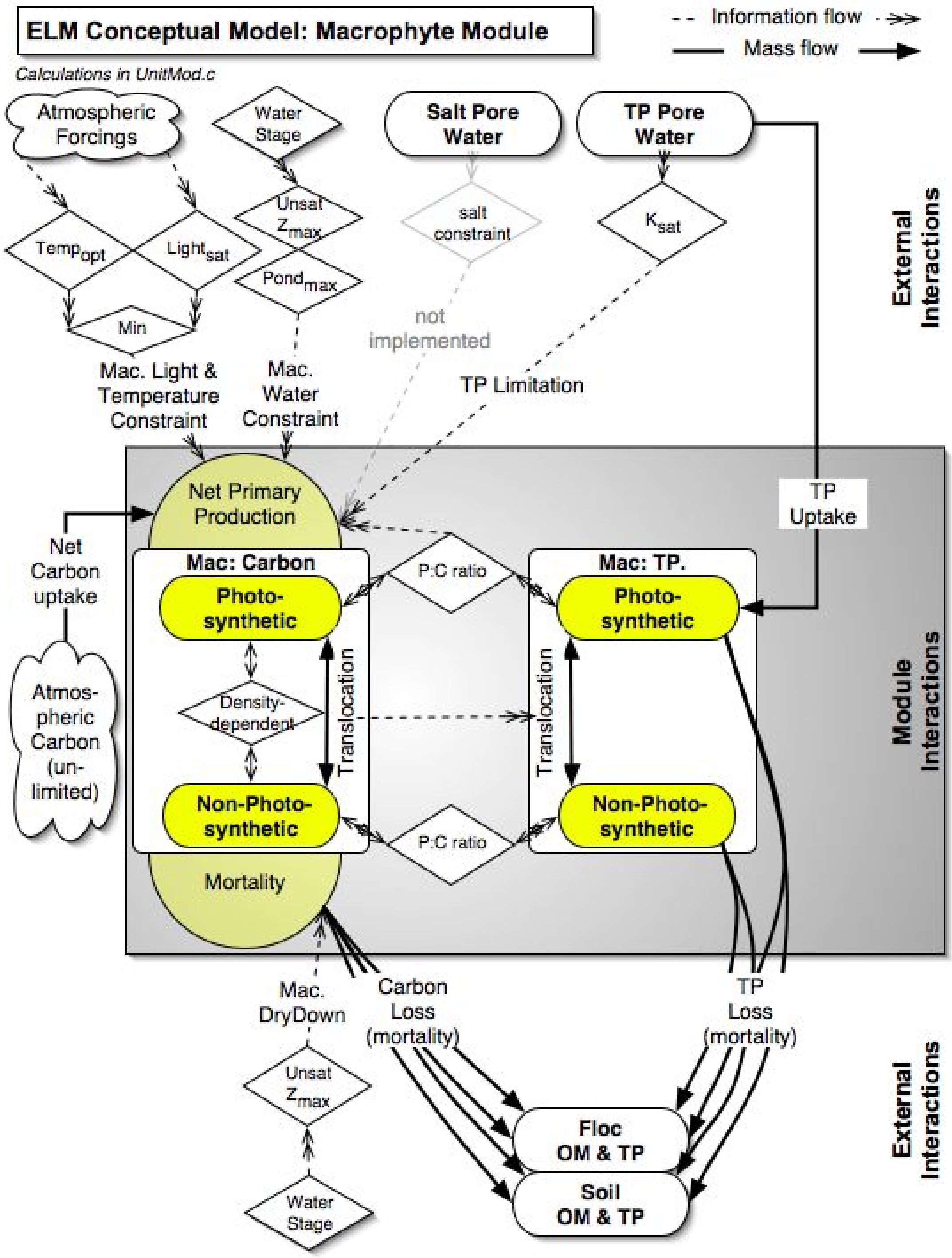
exp(x) = Exp(x) => e raised to the xth power

Max(x,y) => maximum of variable x or y

Min(x,y) => minimum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

5.6.5 Macrophyte module



Overview: Macrophyte Module

Macrophytes are a primary determinant of the habitat quality in the Everglades landscape, which is largely defined by its heterogeneous mosaic of macrophytic vegetation that is dynamic over both annual and decadal time scales. There is a high diversity of plants in this region, ranging from emergent marsh plants such as the ubiquitous sawgrass, to hardwood trees of tree islands and mangrove forests. These, and many other common species, form a wide variety of plant communities with very different nutrient requirements, distinct hydrologic needs, and dynamic effects on the hydrologic cycle itself. Different adaptations by these plants create the habitat mosaic in response to a changing environment. For example, cattail is a "nuisance" species that grows rapidly in response to elevated nutrient availability, has morphological characteristics that allow it to thrive in flooded conditions, and easily colonizes areas that have been disturbed. Sawgrass, on the other hand, is a very dominant species in much of the Everglades where there are oligotrophic (low nutrient) conditions and "natural" fluctuations of water levels and disturbances. With mortality or dieback of leaves and roots of these plants, comes the accumulation of organic matter in the form of peat soils. Where regions of the Everglades have undergone successional shifts in plant communities, animal communities (not considered in ELM) are invariably affected. The ELM assumes that the higher trophic levels respond to these changes in habitat, without the animal communities affecting the regional landscape over long time periods.

Macrophyte Module Description

Macrophytes are simulated using two state variables, photosynthetic and non-photosynthetic carbon biomass. This partition is used to represent variations in plant carbon storage and the concomitant carbon:nutrient ratios in subsequent detrital dynamics from the two stocks. As in the Periphyton Module, this module aggregates all macrophyte species into one stock using average parameter values. While all macrophytes communities (or habitat types) are simulated by one set of equations, their behavior varies according to set of parameters that are specific to each habitat type (see Data Chapter). The Succession Module (separate section in this Chapter) provides the mechanism for switching among habitat types as the cumulative environmental conditions warrant it.

The general form of the equations that describe changes to a macrophyte photosynthetic carbon stock is:

$$S(t) = S(t-1) + (P - TR - M)dt,$$

where *S(time)* is the standing stock of macrophytes (kg C m^{-2}) at time *t* or *t-1*, *P* is the net primary production gain (kg C m^{-2} d^{-1}), *TR* is the translocation loss/gain (kg C m^{-2} d^{-1}), *M* is the mortality loss (kg C m^{-2} d^{-1}), and *dt* is the time interval (days). The actual rates are products of the macrophyte stock and maximum specific rates that are constrained by control functions:

$$P = S(t-1) \cdot P_{max} CF_P$$

$$M = S(t-1) \cdot M_{max} CF_M$$

where P_{max} and M_{max} are the maximum specific rates (d^{-1}) of, respectively, net primary production and mortality; the CF_P and CF_M are the (dimensionless, 0 – 1) control functions constraining net production and mortality, respectively.

Biomass is added to macrophytes through the photosynthetic pathway that determines net production of photosynthetic biomass, with the maximum rate of net production limited by a production control function that considers the most limiting constraint due to either light, temperature, or water, multiplied by the nutrient constraint. Using a form similar to that for periphyton gross production, the rate is further constrained by maximum density considerations.

The nutrient control function is similar to that for periphyton and soil (i.e., implicit microbial) modules, but responds to phosphorus in the soil/sediment water instead of in the surface water. Whereas earlier ELM versions quantified nutrient limitation using Monod half-saturation kinetics (Fitz et al. 1996) (Fitz and Sklar 1999), this relationship appeared to behave inadequately in the oligotrophic conditions of much of the Everglades, with excessive constraint on growth under low (often ambient) conditions. The Monod form assumes enzyme kinetics, with a linear response below saturating nutrient concentrations. There is evidence that phosphatase activity tends to increase under low nutrient conditions (Newman et al. 2003), thus potentially making phosphorus less limiting in general, and deviating from Monod kinetics.

The light control function is based on a simple Steele (1965) formula representing the effects of light limitation and photoinhibition, without self-shading. The temperature control function (Jorgensen 1976) describes the biological responses to air temperature relative to a temperature optimum and a minimum, using the same form as that in the soil (i.e., implicit microbial) and periphyton modules. Water availability to plants is a function of the soil moisture, the depth of the unsaturated zone and the root depth. Water is not limiting at all if the roots reach the saturated zone. When the unsaturated water table is shallower than the root zone depth, the value returned is the moisture proportion in the unsaturated zone plus an exponentially decreasing amount from the saturated zone. Thus water may be available to the root system when the roots do not reach the saturated zone due to the capillary draw of water from a nearby saturated layer.

If carbon fixed by the photosynthetic pathway is in excess of that needed for net growth of shoot and leaf biomass, that carbon is translocated to the nonphotosynthetic stock, thus assuming a very simple homeostatic mechanism between roots and shoots.

Mortality within the photosynthetic stock is determined from current water stress. The maximum specific rate of mortality is limited by the water stress limitation. Mortality of the nonphotosynthetic module is assumed to occur at a constant rate. The effects of salinity and other factors simulated in the model could be incorporated into a control function depending on the model requirements.

Macrophytes have direct feedbacks on the physical hydrology that are important to overall model dynamics. The areal density of stems and trunks is calculated based on data for the plant type such as Steward and Ornes (1975) for a subtropical sedge. These data and the plant height are used in determining a Manning's roughness coefficient (see the Hydrology Module) for each community type.

Macrophyte Module Equations

all calculated within spatial loop across model grid rows, columns

State Variable update calculations (carbon only)

MAC_NOPH_BIOMAS = MAC_NOPH_BIOMAS + (NPHBIO_TRANSLOC - NPHBIO_MORT - PHBIO_TRANSLOC) * ***DT***

MAC_PH_BIOMAS = MAC_PH_BIOMAS + (PHBIO_TRANSLOC + PHBIO_NPP - PHBIO_MORT - NPHBIO_TRANSLOC) * ***DT***

Dependent upon:

1) attribute calculations

these thresholds need updating when a habitat type of a grid cell changes

MAC_MAX_BIO = ***HP_NPHBIO_MAX***+ ***HP_PHBIO_MAX***

NPHBIO_REFUGE = ***HP_NPHBIO_MAX**** ***GP_MAC_REFUG_MULT***

NPHBIO_SAT = ***HP_NPHBIO_MAX****0.9

PHBIO_REFUGE = ***HP_PHBIO_MAX**** ***GP_MAC_REFUG_MULT***

PHBIO_SAT = ***HP_PHBIO_MAX****0.9

MAC_PHtoNPH_Init = ***HP_PHBIO_MAX*** / ***HP_NPHBIO_MAX***

MAC_PHtoNPH = (MAC_NOPH_BIOMAS>0.0) ? (MAC_PH_BIOMAS / MAC_NOPH_BIOMAS) : (MAC_PHtoNPH_Init)

phbio_ddep = Max(1.0-Max((PHBIO_SAT-MAC_PH_BIOMAS) /(PHBIO_SAT-PHBIO_REFUGE),0.0),0.0)

PHBIO_AVAIL = MAC_PH_BIOMAS*phbio_ddep

nphbio_ddep = Max(1.0-Max((NPHBIO_SAT-MAC_NOPH_BIOMAS)/(NPHBIO_SAT-NPHBIO_REFUGE),0.0),0.0)

NPHBIO_AVAIL = MAC_NOPH_BIOMAS*nphbio_ddep

2) control function calculations

MAC_LIGHT_CF = SOLRADGRD/***MAC_LIGHTSAT****Exp(1.0-SOLRADGRD/ ***HP_MAC_LIGHTSAT***)

Jorgensen 1976; 5 deg C is minimum temperature parameter

MAC_TEMP_CF = Exp(-2.3 * ABS((AIR_TEMP- ***HP_MAC_TEMPOPT***)/(***HP_MAC_TEMPOPT-5.0***)))

MAC_WATER_CF = Min(MAC_WATER_AVAIL_CF, Max(1.0-Max((SURFACE_WAT-***HP_MAC_WAT_TOLER***)/ ***HP_MAC_WAT_TOLER***,0.0),0.0))

MAC_NUT_CF = Exp(-***GP_mac_uptake_coef*** * Max(***HP_MAC_KSP***-TP_SEDWT_CONCACTMG, 0.0)/ ***HP_MAC_KSP***)

min_litTemp = Min(MAC_LIGHT_CF, MAC_TEMP_CF)

MAC_PROD_CF = Min(min_litTemp,MAC_WATER_CF)*MAC_NUT_CF

3) flux calculations

PHBIO_NPP = ***HP_PHBIO_RCNPP****MAC_PROD_CF*MAC_PH_BIOMAS * (1.0-MAC_TOT_BIOM/MAC_MAX_BIO)

NPP_P = PHBIO_NPP * ***HP_PHBIO_PC*** * Max(MAC_NUT_CF*2.0,1.0) * Max(1.0-mac_ph_PC/ ***HP_PHBIO_PC***, 0.0)

check for available P mass that will be taken up from sed water in active zone (nutCF does not)

reduc = (NPP_P > 0.0) ? (TP_SED_WT_AZ / (NPP_P****CELL_SIZE*****DT***)) : (1.0)

if (reduc < 1.0) PHBIO_NPP = PHBIO_NPP * reduc

if (reduc < 1.0) NPP_P = NPP_P * reduc

NPHBIO_TRANSLOC_POT = (MAC_PHtoNPH>MAC_PHtoNPH_Init) ? (exp(***HP_MAC_TRANSLOC_RC*** *(MAC_PHtoNPH-MAC_PHtoNPH_Init)) - 1.0) : (0.0)

NPHBIO_TRANSLOC = (NPHBIO_TRANSLOC_POT****DT*** >PHBIO_AVAIL) ? (PHBIO_AVAIL/***DT***) : (NPHBIO_TRANSLOC_POT)

PHBIO_MORT_POT = ***HP_PHBIO_RCMORT*** * PHBIO_AVAIL * (1.0 + (1.0-MAC_WATER_AVAIL_CF))/2.0

PHBIO_MORT = ((PHBIO_MORT_POT+NPHBIO_TRANSLOC)****DT***>PHBIO_AVAIL) ? ((PHBIO_AVAIL-NPHBIO_TRANSLOC****DT***)/***DT***) : (PHBIO_MORT_POT)

PHBIO_TRANSLOC_POT = (MAC_PHtoNPH<MAC_PHtoNPH_Init) ? (exp(***HP_MAC_TRANSLOC_RC*** *(MAC_PHtoNPH_Init-MAC_PHtoNPH)) - 1.0) : (0.0)

PHBIO_TRANSLOC = (PHBIO_TRANSLOC_POT****DT*** >NPHBIO_AVAIL) ? (NPHBIO_AVAIL/***DT***) : (PHBIO_TRANSLOC_POT)

decreased non-photobiomass mortality w/ increasing photobiomass
NPHBIO_MORT_POT = NPHBIO_AVAIL* ***HP_PHBIO_RCMORT*** * (1.0 + Max(1.0-MAC_PH_BIOMAS/ ***HP_PHBIO_MAX***,0.0))/2.0

NPHBIO_MORT = ((PHBIO_TRANSLOC+NPHBIO_MORT_POT)****DT***>NPHBIO_AVAIL) ? ((NPHBIO_AVAIL-PHBIO_TRANSLOC****DT***)/***DT***) : (NPHBIO_MORT_POT)

4) attribute calculations, used in other modules

MAC_TOT_BIOM = MAC_PH_BIOMAS+MAC_NOPH_BIOMAS

MAC_REL_BIOM = (MAC_TOT_BIOM > 0.0) ? MAC_TOT_BIOM/MAC_MAX_BIO : ***0.0001***

MAC_HEIGHT = pow(MAC_REL_BIOM,***0.33***)* ***HP_MAC_MAXHT***

MAC_LAI = MAC_REL_BIOM* ***HP_MAC_MAXLAI***

5) phosphorus and organic matter associated with carbon stocks & flows

change of grid-cell habitat (including macrophyte) type necessitates dynamic accounting of all variables
P and OM fluxes
phbio_npp_P = NPP_P /* within-plant variable stoichiometry */

phbio_npp_OM = PHBIO_NPP / ***HP_PHBIO_CTOOM*** /* habitat-specfic stoichiometry */

phbio_mort_P = PHBIO_MORT * mac_ph_PC

phbio_mort_OM = PHBIO_MORT / mac_ph_CtoOM

phbio_transl_P = PHBIO_TRANSLOC * mac_nph_PC

phbio_transl_OM = PHBIO_TRANSLOC / mac_nph_CtoOM

nphbio_transl_P = NPHBIO_TRANSLOC * mac_ph_PC

nphbio_transl_OM = NPHBIO_TRANSLOC / mac_ph_CtoOM

nphbio_mort_P = NPHBIO_MORT * mac_nph_PC

nphbio_mort_OM = NPHBIO_MORT / mac_nph_CtoOM

mac_nph_P = mac_nph_P + (nphbio_transl_P - nphbio_mort_P - phbio_transl_P) * ***DT***

default to 0.3 of max for habitat

mac_nph_PC = (MAC_NOPH_BIOMAS > 0.0) ? (mac_nph_P / MAC_NOPH_BIOMAS) : 0.3 * ***HP_NPHBIO_PC***

mac_nph_OM = mac_nph_OM + (nphbio_transl_OM - nphbio_mort_OM - phbio_transl_OM) * ***DT***

mac_nph_CtoOM = (mac_nph_OM > 0.0) ? (MAC_NOPH_BIOMAS / mac_nph_OM) : ***HP_NPHBIO_CTOOM***

mac_ph_P = mac_ph_P + (phbio_transl_P + phbio_npp_P - phbio_mort_P - nphbio_transl_P) * ***DT***

default to 0.3 of max for habitat

mac_ph_PC = (MAC_PH_BIOMAS > 0.0) ? (mac_ph_P / MAC_PH_BIOMAS) : 0.3 * ***HP_PHBIO_PC***

mac_ph_OM = mac_ph_OM + (phbio_transl_OM + phbio_npp_OM - phbio_mort_OM - nphbio_transl_OM) * ***DT***

mac_ph_CtoOM = (mac_ph_OM > 0.0) ? (MAC_PH_BIOMAS / mac_ph_OM) : ***HP_PHBIO_CTOOM***

TP_SEDWT_UPTAKE = phbio_npp_P****CELL_SIZE***

TP_SED_WT = TP_SED_WT - (TP_SEDWT_UPTAKE) * ***DT***

TP_SED_CONC = (HYD_SED_WAT_VOL>0.0) ? (TP_SED_WT / HYD_SED_WAT_VOL) : (0.0)

*## this is the active zone, where uptake, sorption, and mineralization take place */*

TP_SED_WT_AZ = TP_SED_WT_AZ - (TP_SEDWT_UPTAKE) * ***DT***

TP_SEDWT_CONCACT = (HYD_DOM_ACTWAT_PRES > 0.0) ? (TP_SED_WT_AZ/HYD_DOM_ACTWAT_VOL) : (TP_SED_CONC)

TP_SEDWT_CONCACTMG = TP_SEDWT_CONCACT*conv_kgTOg /

External variables used

SOLRADGRD (see Globals module)

AIR_TEMP (see Globals module)

TP_SED_WT (see TP/Salt module)

TP_SED_WT_AZ (see TP/Salt module)

TP_SEDWT_CONCACTMG (see TP/Salt module)

SURFACE_WAT (see Hydrology module)

HYD_SED_WAT_VOL (see Hydrology module)

HYD_DOM_ACTWAT_PRES (see Hydrology module)

HYD_DOM_ACTWAT_VOL (see Hydrology module)

MAC_WATER_AVAIL_CF (see Hydrology module)

Macrophyte Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
mac_nph_PC_rep	attribute	mgP/kgC	macrophyte nonphotosynthetic tissues Phosphorus to Carbon concentration (units converted for reporting purposes)

mac_ph_PC_rep	attribute	mgP/kgC	macrophyte photosynthetic tissues Phosphorus to Carbon concentration (units converted for reporting purposes)
MAC_HEIGHT	attribute	m	HEIGHT of MACrophytes above ground surface
NPHBIO_AVAIL	attribute	kgC/m^2	NonPHototsynthetic macrophyte BIOmass AVAILable for losses via mortality and translocation
PHBIO_AVAIL	attribute	kgC/m^2	PHototsynthetic macrophyte BIOmass AVAILable for losses via mortality and translocation
MAC_LAI	attribute	dimless	MACrophyte Leaf Area Index of the proportion of leaf surface area to ground surface area
MAC_REL_BIOM	attribute	dimless	proportion of MACrophyte BIOMass RELative to its maximum attainable
MAC_LIGHT_CF	controlFunction	dimless	MACrophyte growth Control Function (0-1) of degree of LIGHT limitation
MAC_NUT_CF	controlFunction	dimless	MACrophyte growth Control Function (0-1) of degree of NUTrient limitation
MAC_PROD_CF	controlFunction	dimless	MACrophyte growth Control Function (0-1) of degree of combined limitations on net carbon primary PRODuction
MAC_TEMP_CF	controlFunction	dimless	MACrophyte growth Control Function (0-1) of degree of TEMPerature limitation
MAC_WATER_CF	controlFunction	dimless	MACrophytes growth Control Function (0-1) of degree of WATer limitation
MAC_SALT_CF	controlFunction	dimless	MACrophyte growth Control Function (0-1) of degree of SALT constraint; inoperative in v2.2, hardwired=1.0
TP_SEDWT_UPTAKE	rateActual	kgP/d	Total Phosphorus UPTAKE from SEDment/soil WaTer due to macrophyte net primary production
NPHBIO_MORT	rateActual	kgC/m^2/d	NonPHototsynthetic macrophyte BIOmass MORTality losses
NPHBIO_TRANSLOC	rateActual	kgC/m^2/d	NonPHotosynthetic macrophyte biomass TRANSLOCation gain from photosynthetic biomass
PHBIO_MORT	rateActual	kgC/m^2/d	PHototsynthetic macrophyte BIOmass MORTality losses
PHBIO_NPP	rateActual	kgC/m^2/d	PHototsynthetic macrophyte BIOmass Net Primary Production growth gain
PHBIO_TRANSLOC	rateActual	kgC/m^2/d	PHotosynthetic macrophyte biomass TRANSLOCation gain from non-photosynthetic biomass
NPHBIO_MORT_POT	ratePotential	kgC/m^2/d	NonPHototsynthetic macrophyte macrophyte BIOmass MORTality POTential losses
NPHBIO_TRANSLOC_POT	rateP	kgC/m^	NonPHotosynthetic macrophyte

	otential	2/d	biomass TRANSLOCation POTential gain from photosynthetic biomass
PHBIO_MORT_POT	ratePotential	kgC/m^2/d	PHototsynthetic macrophyte macrophyte BIOmass MORTality POTential losses
PHBIO_TRANSLOC_POT	ratePotential	kgC/m^2/d	PHotosynthetic macrophyte biomass TRANSLOCation POTential gain from non-photosynthetic biomass
mac_nph_P	state	kgP/m^2	macrophytes live non-photosynthetic tissue (Phosphorus) biomass
mac_ph_P	state	kgP/m^2	macrophytes live photosynthetic tissue (Phosphorus) biomass
mac_nph_OM	state	kgOM/m^2	macrophytes live non-photosynthetic tissue (Organic Matter) biomass (bookeeping, only used for mass balance when cell changes habitats)
mac_ph_OM	state	kgOM/m^2	macrophytes live photosynthetic tissue (Organic Matter) biomass (bookeeping, only used for mass balance when cell changes habitats)
MAC_NOPH_BIOMAS	state	kgC/m^2	MACrophytes live NOn-PHotosynthetic tissue (carbon) BIOMASs
MAC_PH_BIOMAS	state	kgC/m^2	MACrophytes live PHotosynthetic tissue (carbon) BIOMASs
MAC_TOT_BIOM	state Convert	kgC/m^2	MACrophytes live TOTal tissue BIOMASs
MAC_MAX_BIO	static	kgC/m^2	MACrophytes MAXimum attainable BIOmass (sum of two parameters)
NPHBIO_REFUGE	static	kgC/m^2	NonPHototsynthetic macrophyte BIOmass REFUGE density (from losses)
NPHBIO_SAT	static	kgC/m^2	NonPHotosynthetic macrophyte BIOmass SATuration density (90% of the maximum attainable)
PHBIO_REFUGE	static	kgC/m^2	PHototsynthetic macrophyte BIOmass REFUGE density (from losses)
PHBIO_SAT	static	kgC/m^2	PHotosynthetic macrophyte BIOmass SATuration density (90% of the maximum attainable)

Time series forcing data

none

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
GP_MAC_REFUG_MULT	global	dimless	proportion of max attainable macrophyte biomass, defining a refuge density (from losses)

GP_mac_uptake_coef	global	dimless	parameter for exp function describing nutrient uptake kinetics

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description
HP_NPHBIO_MAX	habspec	kgC/m^2	Maximum attainable (observed) biomass density of nonphotosynthetic tissue.
HP_PHBIO_MAX	habspec	kgC/m^2	Maximum attainable (observed) biomass density of photosynthetic tissue.
HP_MAC_KSP	habspec	mgP/L	Half saturation coeff of phosphorus for the nutrient uptake kinetics of macrophytes.
HP_MAC_MAXLAI	habspec	dimless	Maximum observed/attainable Leaf Area Index for a mature community (= area of leaves/area of ground).
HP_MAC_LIGHTSAT	habspec	cal/cm^2/d	Saturating light intensity (langleys/d = cal/cm^2 per day) for macrophyte growth kinetics.
HP_MAC_MAXHT	habspec	m	Maximum observed/attainable height of mature plant community (associated with a unit plant density at maturity).
HP_MAC_TEMPOPT	habspec	deg C	Optimal temperature for maximum primary production growth rate.
HP_NPHBIO_CTOOM	habspec	gC/gOM	Initial ratio of organic carbon to total organic material in NonPhotoBiomass (ash free dry weight).
HP_NPHBIO_PC	habspec	gP/gC	Initial phosphorus:carbon ratio in NonPhotoBiomass (ash free dry weight).
HP_PHBIO_CTOOM	habspec	gC/gOM	Initial ratio of organic carbon to total organic material in PhotoBiomass (ash free dry weight).
HP_PHBIO_PC	habspec	gP/gC	Initial phosphorus:carbon ratio in PhotoBiomass (ash free dry weight).
HP_PHBIO_RCNPP	habspec	1/d	Maximum observed/attainable specific rate of net primary production.
HP_PHBIO_RCMORT	habspec	1/d	Baseline specific rate of photobiomass mortality.
HP_MAC_WAT_TOLER	habspec	m	Depth of ponded surface water above which plant growth becomes restricted. Used in growth control function.
HP_MAC_TRANSLOC_RC	habspec	1/d	Simple, bi-directional baseline translocation rate between Non-photo and Photo biomass; a gradual equilibrium used, while evaluating a more process-based algorithm

Intrinsic C or ELM functions

exp(x) = Exp(x) => e raised to the xth power

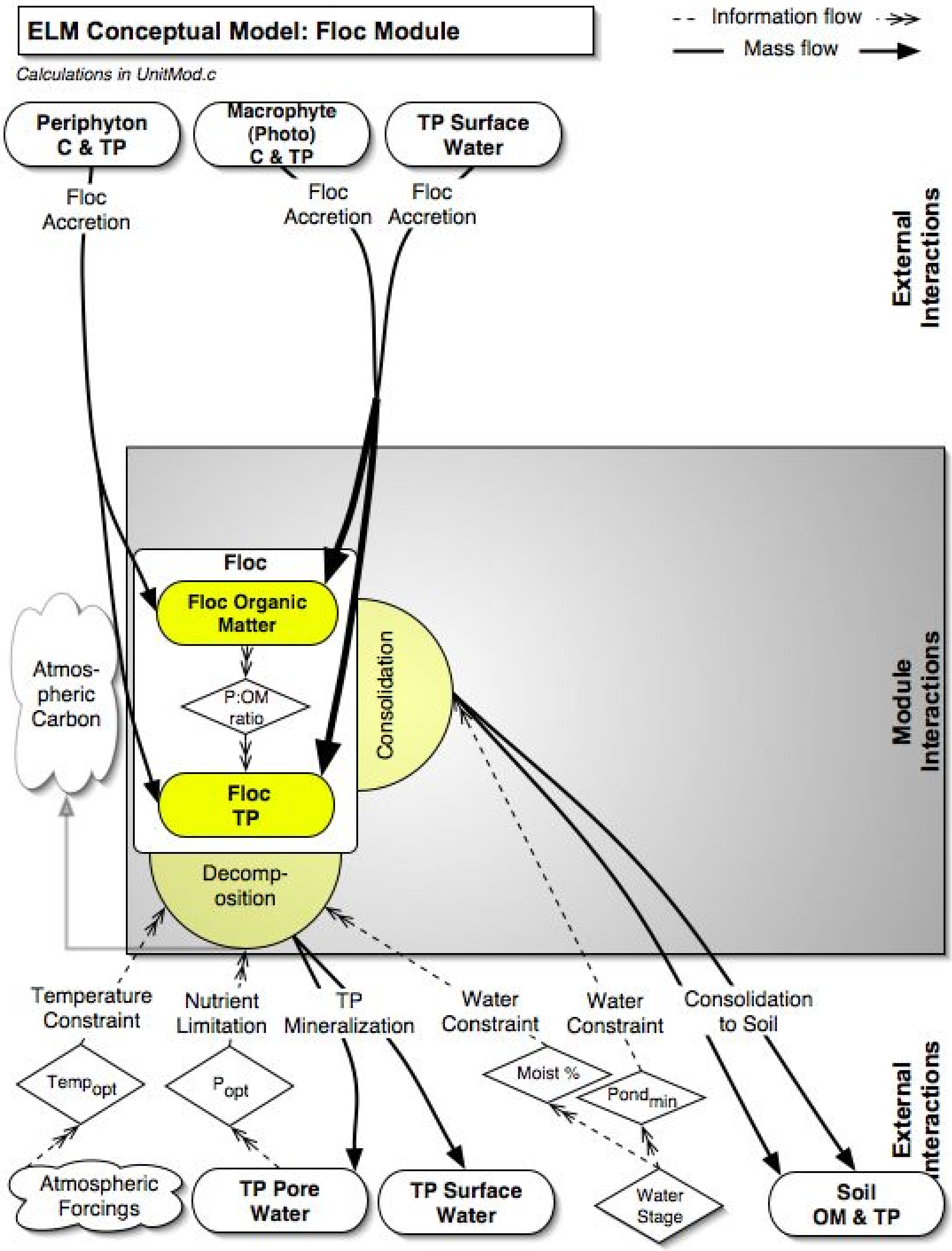
Max(x,y) => maximum of variable x or y

Min(x,y) => minimum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

pow(x,y) => x raised to the yth power (generally avoided if possible due to execution time of C library)

5.6.6 Floc module



Overview: Floc Module

This module updates the vertical dynamics of the flocculent organic material that is at the interface between the consolidated soil and the surface water column. Throughout much of the Everglades is an upper-soil layer of flocculent (fluffy) organic material that is partly live periphyton, but principally the organic material from dead periphyton and macrophytes. This "floc" appears to play a critical role in nutrient cycling and transport of organic material among habitats and, potentially forms part of a detrital food web for animals.

Floc Module Description

This “Floc” matter is very fine-grained organic detritus, and is assumed to be highly labile and relatively transient relative to the underlying soil matrix. Organic matter and phosphorus are added to the Floc state variable due to settling from water column and mortality of periphyton and macrophytes. Using the same form of equations in the Soil Module, floc is lost through aerobic decomposition that is constrained by temperature, nutrients, and moisture (in absence of surface water). Floc depositional losses to the underlying soil occur at a baseline rate, with more rapid consolidation into soil as the floc layer becomes deeper or when surface water is absent (with the highest rate potential). As a module that was added to ELM (v2.1) in order to better match fluxes and stocks of nutrients in the water column, soil and periphyton, the Floc appears to be (at least) an important biogeochemical driver of the nutrient status of the ecosystem. However, there are significant gaps in our understanding of Floc dynamics under the wide range of conditions in the Everglades, and thus the module is very basic compared to the complex dynamics that likely exist in the ecosystem(s).

Floc Module Equations

all calculated within spatial loop across model grid rows, columns

State Variable update calculations

FLOC = FLOC + (Floc_settl + Floc_fr_phBio + FLOC_FR_ALGAE - FLOC_DECOMP - FLOC_DEPO) * ***DT***

FlocP = FlocP + (FlocP_settl + FlocP_PhBio + FlocP_FR_ALGAE - FlocP_DECOMP - FlocP_DEPO) * ***DT***

Dependent upon:

1) attribute calculations

FLOC_FR_ALGAE = (C_ALG_MORT + NC_ALG_MORT) / ***GP_ALG_C_TO_OM****0.001

FlocP_FR_ALGAE = (NC_ALG_MORT_P + C_ALG_MORT_P) * 0.001

Floc_fr_phBio = phbio_mort_OM

FlocP_PhBio = phbio_mort_P

FlocP_settl = TP_settl / ***CELL_SIZE***

Floc_settl = FlocP_settl / ***GP_TP_P_OM***

FLOC_Z = FLOC / ***GP_Floc_BD***

FlocP_OM = (FLOC>0.0) ? (FlocP/FLOC) : (0.0)

2) control function calculations

FLOC_DECOMP_QUAL_CF = Exp(-***GP_DOM_decomp_coef*** * Max(***GP_DOM_DECOMP_POPT***- (TP_SFWT_CONC_MG+TP_SEDWT_CONCACTMG)/2.0, 0.0)/ ***GP_DOM_DECOMP_POPT***)

soil_MOIST_CF = (UNSAT_DEPTH> ***HP_DOM_AEROBTHIN***) ? (Max(UNSAT_MOIST_PRP,0.0)) : (1.0)

3) flux calculations

the Floc substrate quality is 10x greater than that of bulk soil

FLOC_DECOMP_POT = ***GP_calibDecomp*** * ***10.0****DOM_RCDECOMP*FLOC*DOM_TEMP_CF *FLOC_DECOMP_QUAL_CF * soil_MOIST_CF

FLOC_DECOMP = ((FLOC_DECOMP_POT)****DT***>FLOC) ? ((FLOC)/***DT***) : (FLOC_DECOMP_POT)

FlocP_DECOMP_pot = FLOC_DECOMP * FlocP_OM

FlocP_DECOMP = ((FlocP_DECOMP_pot)****DT***>FlocP) ? ((FlocP)/***DT***) : (FlocP_DECOMP_pot)

FLOC_DEPO_POT = (SURFACE_WAT > ***GP_DetentZ***) ? (FLOC_Z/ ***GP_FlocMax*** * FLOC* ***GP_Floc_rcSoil***) : (FLOC* ***GP_Floc_rcSoil***)

FLOC_DEPO = ((FLOC_DEPO_POT+FLOC_DECOMP)****DT***>FLOC) ? ((FLOC-FLOC_DECOMP****DT***)/***DT***) : (FLOC_DEPO_POT)

FlocP_DEPO_pot = FLOC_DEPO * FlocP_OM

FlocP_DEPO = ((FlocP_DEPO_pot+FlocP_DECOMP)****DT***>FlocP) ? ((FlocP-FlocP_DECOMP****DT***)/***DT***) : (FlocP_DEPO_pot)

4) attributes calculated after floc updates, used in other modules

90% of the decomp contributes to soil/sediment; 10% to surface water P

TP_SED_MINER = ***0.90*** * FlocP_DECOMP * ***CELL_SIZE***

TP_SFWT_MINER = ***0.10*** * FlocP_DECOMP * ***CELL_SIZE*** ;

state variable updates

TP_SED_WT = TP_SED_WT + (TP_SED_MINER) * ***DT***;

TP_SED_WT_AZ = TP_SED_WT_AZ + (TP_SED_MINER) * ***DT***;

TP_SF_WT = TP_SF_WT + (TP_SFWT_MINER) * ***DT***

TP_SED_CONC = (HYD_SED_WAT_VOL>0.0) ? (TP_SED_WT / HYD_SED_WAT_VOL) : (0.0)

TP_SEDWT_CONCACT = (HYD_DOM_ACTWAT_PRES > 0.0) ? (TP_SED_WT_AZ/HYD_DOM_ACTWAT_VOL) : (TP_SED_CONC)

TP_SFWT_CONC = (SFWT_VOL > 0.0) ? (TP_SF_WT/SFWT_VOL) : (0.0)

External variables used

DOM_TEMP_CF (see Soils module)

C_ALG_MORT (see Periphyton module)

C_ALG_MORT_P (see Periphyton module)

NC_ALG_MORT (see Periphyton module)

NC_ALG_MORT_P (see Periphyton module)

phbio_mort_OM (see Macrophyte module)

phbio_mort_P (see Macrophyte module)

TP_settl (see TP/Salt module)

TP_SFWT_CONC_MG (see TP/Salt module)

TP_SEDWT_CONCACTMG (see TP/Salt module)

UNSAT_DEPTH (see Hydrology module)

UNSAT_MOIST_PRP (see Hydrology module)

SURFACE_WAT (see Hydrology module)

Floc Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
FlocP_OMrep	attribute	mgP/kg OM	Phosphorus concentration of the Flocculent Organic Matter (units converted to this for reporting purposes)
FlocP_OM	attribute	kgP/kgOM	Phosphorus concentration in the Flocculent Organic Matter
FLOC_DECOMP_QUAL_CF	controlFunction	dimless	FLOCculent organic matter - DECOMPosition Control Function (0-1) of degree of nutrient QUALity limitation
soil_MOIST_CF	controlFunction	dimless	Deposited Organic Matter Control Function of degree of MOISTure limitation
FlocP_FR_ALGAE	rateActual	kgP/m^2/d	Phosphorus in the FLOCculent organic matter gained from FRom mortality of periphyton (generalized, ALGAE)
FlocP_PhBio	rateActual	kgP/m^2/d	Phosphorus in the FLOCculent organic matter gained from mortality of photosynthetic Biomass of macrophytes
FlocP_settl	rateActual	kgP/m^2/d	Phosphorus in the FLOCculent organic matter gained from (flocculation &) settling out of water column
FlocP_DECOMP	rateActual	kgP/m^2/d	Phosphorus in the FLOCculent organic matter - DECOMPosition losses
FlocP_DEPO	rateActual	kgP/m^2/d	Phosphorus in the FLOCculent organic matter - DEPosition losses
TP_SED_MINER	rateActual	kgP/d	Total Phosphorus gained in SEDiment/soil water due to floc MINERalization
TP_SFWT_MINER	rateActual	kgP/d	Total Phosphorus gained in SurFace WaTer due to floc MINERalization
Floc_fr_phBio	rateActual	kgOM/m^2/d	FLOCculent organic matter gained from mortality of photosynthetic Biomass of macrophytes
Floc_settl	rateA	kgOM/m	FLOCculent organic matter gained

	ctual	^2/d	from (flocculation &) settling out of water column
FLOC_DECOMP	rateActual	kgOM/m^2/d	FLOCculent organic matter - DECOMPosition losses
FLOC_DEPO	rateActual	kgOM/m^2/d	FLOCculent organic matter - DEPosition losses
FLOC_FR_ALGAE	rateActual	kgOM/m^2/d	FLOCculent organic matter gained FRom mortality of periphyton (generalized, ALGAE)
FlocP_DECOMP_pot	ratePotential	kgP/m^2/d	Phosphorus in the FLOCculent organic matter - DECOMPosition potential losses
FlocP_DEPO_pot	ratePotential	kgP/m^2/d	Phosphorus in the FLOCculent organic matter - DEPosition potential losses
FLOC_DECOMP_POT	ratePotential	kgOM/m^2/d	FLOCculent organic matter - DECOMPosition POTential losses
FLOC_DEPO_POT	ratePotential	kgOM/m^2/d	FLOCculent organic matter - DEPosition POTential losses
FlocP	state	kgP/m^2	Phosphorus in the FLOCculent organic matter at the interface between soil and surface water
FLOC	state	kgOM/m^2	FLOCculent organic matter at the interface between soil and surface water
FLOC_Z	stateConvert	m	FLOCculent organic matter depth at the interface between soil and surface water

Time series forcing data
none

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
GP_DetentZ	global	m	detention depth in a grid cell, below which surface flows do not occur
GP_calibDecomp	global	dimless	calibration parameter, multiply potential decomposition rate of organic matter
GP_ALG_C_TO_OM	global	gC/gOM	Mass ratio of organic carbon to total organic material in algae (ash free dry weight).
GP_DOM_decomp_coef	global	dimless	parameter for exp function describing decomposition kinetics
GP_DOM_DECOMP_POPT	global	mg/L	Optimal phosphorus concentration in water for maximal decomposition of organic matter
GP_TP_P_OM	global	gP/gOM	phosphorus to organic matter ratio of

			particulate phosphorus (ash-free masses)
GP_Floc_BD	global	mg/cm3	bulk density of floc layer (mg/cm3 == kg/m3)
GP_FlocMax	global	m	max floc depth observed/attainable
GP_Floc_rcSoil	global	1/d	baseline rate of floc layer consolidation into the soil matrix (under flooded conditions)

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description
HP_DOM_AEROBTHIN	hab-spec	m	The thin aerobic zone in a flooded wetland. Note that aerobic total depth is defined to include any zone of soil/sediment that is unsaturated or devoid of water.

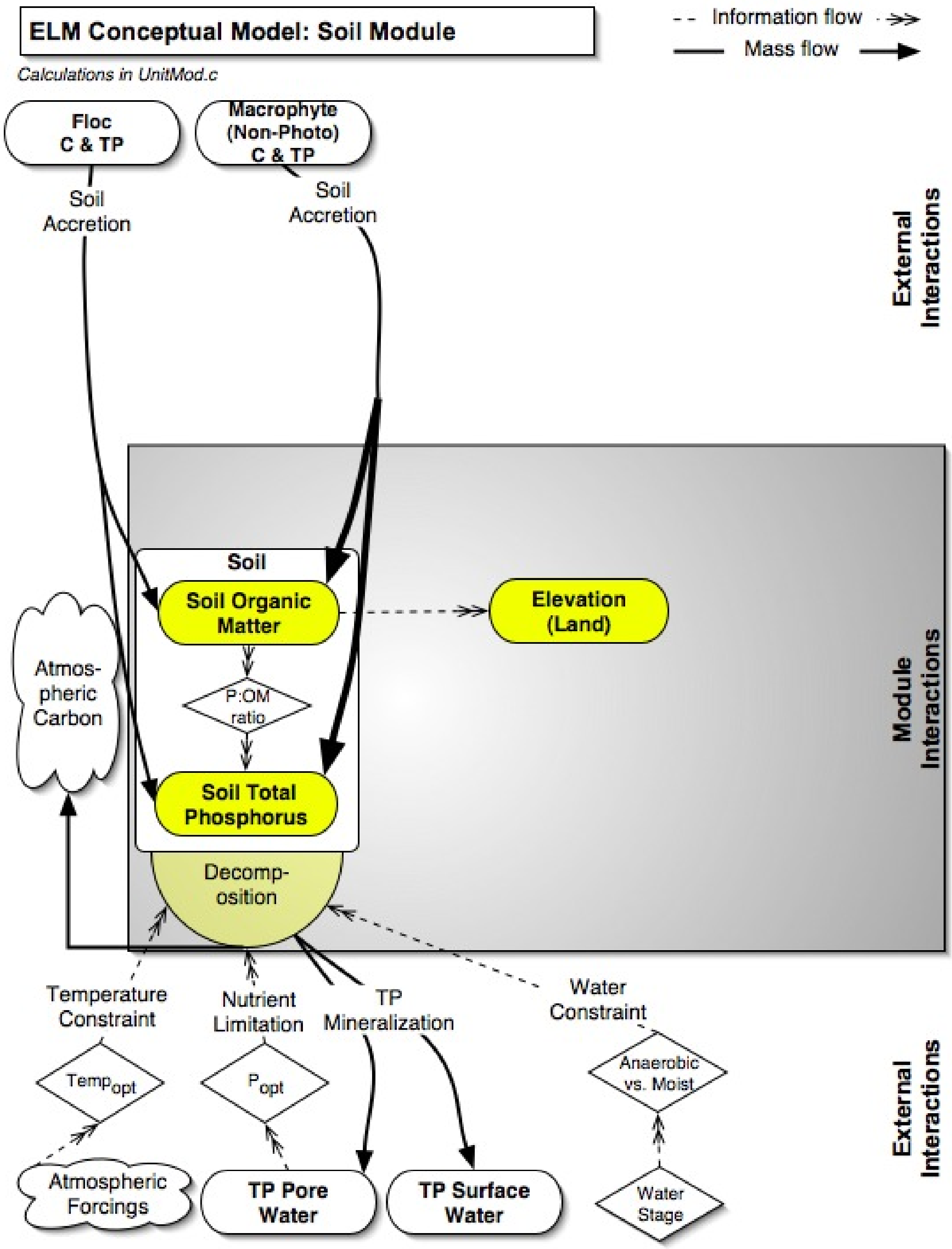
Intrinsic C or ELM functions

exp(x) = Exp(x) => e raised to the x^{th} power

Max(x,y) => maximum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

5.6.7 Soils module



Overview: Soils Module

This module updates the vertical dynamics of the soil, with dynamic changes in the Deposited Organic Matter and the associated Deposited Organic Phosphorus (excluding floc matter). Soils and sediments are in a long-term balance between processes of accumulation and oxidation, closely integrated with the development of different habitats. In regions of long hydroperiods, where water ponds for much of the year, peat soils tend to accrete organic material resulting from plant mortality and floc consolidation. Under shorter hydroperiods, when those soils are exposed more frequently to the air (and thus more aerobic conditions), oxidation of the organic matter reduces the depth of peat. This process is governed by microbial dynamics, and can be accelerated with higher nutrient availability. The oxidation (mineralization) of soil releases nutrients from tightly bound organic forms into inorganic chemical forms that are more readily available to plants and microbes. Disturbances such as severe droughts can have significant impacts on peat soils, oxidizing the organic carbon, but leaving behind much of the phosphorus to which the ecosystem may respond.

Soils Module Description

The principal objectives of the current soil module are to capture multi-decadal trends in the regional gradients in organic soil accretion/oxidation and phosphorus concentration of the upper soil matrix. The soil organic matter and phosphorus content variables are assumed homogenous in vertical profile, overlain by the separate Floc variable (that is calculated in a separate Module described in this Chapter). The general form of this critical soil dynamic is:

$$S(t) = S(t-1) + (A - D)dt,$$

where *S(time)* is the standing stock of organic matter (OM) of soil (kg OM m^{-2}) at time *t* or *t-1*, *A* is the accretion gain (kg OM m^{-2} d^{-1}), *D* is the decomposition loss (kg OM m^{-2} d^{-1}), and *dt* is the time interval (days). The actual rate of accretion is determined in the donor (macrophyte and floc) modules. The actual decomposition is the product of the soil organic matter stock and the maximum specific decomposition rate that is constrained by control functions: depending on water levels, soil is lost through aerobic and anaerobic decomposition that is constrained by temperature, nutrients, and moisture. The maximum depth of the active soil zone in which these dynamics occur is determined by a habitat-specific parameter (generally ca. 30 cm, similar to the macrophyte root zone depth).

The mass of Deposited Organic Matter and the mass of phosphorus associated with that stock are updated as separate variables, and thus the phosphorus ratio of the soil changes in response to the phosphorus concentrations of its input masses. The inorganic component of the soil remains constant at the mass that was initialized in the simulation. The relative magnitudes of organic matter accretion and decomposition determines the change in land surface elevation, assuming a fixed soil bulk density. These simplifying assumptions may be relaxed as increased information becomes available on soil processes such as decomposition rates under varying conditions, flocculation and compaction rates of different soils, and other principal dynamics.

Soils Module Equations

all calculated within spatial loop across model grid rows, columns

State Variable update calculations

DEPOS_ORG_MAT = DEPOS_ORG_MAT + (DOM_fr_nphBio + DOM_FR_FLOC - DOM_DECOMP) * ***DT***

DOP = DOP + (DOP_nphBio + DOP_FLOC - DOP_DECOMP) * ***DT***

Dependent upon:

1) attribute calculations

DOM_SED_AEROB_Z = Min(Max(UNSAT_DEPTH, ***HP_DOM_AEROBTHIN***), ***HP_DOM_MAXDEPTH***);

DOM_SED_ANAEROB_Z = ***HP_DOM_MAXDEPTH***-DOM_SED_AEROB_Z;

DOM_fr_nphBio = nphbio_mort_OM

DOM_FR_FLOC = FLOC_DEPO

DOP_nphBio = nphbio_mort_P

DOP_FLOC = FlocP_DEPO

2) control function calculations

DOM_QUALITY_CF = Min(Exp(-***GP_DOM_decomp_coef*** * Max(***GP_DOM_DECOMP_POPT***-TP_SEDWT_CONCACTMG, 0.0)/ ***GP_DOM_DECOMP_POPT***),1.0)

Jorgensen 1976 ; 5 deg C is minimum temperature parameter

DOM_TEMP_CF = Exp(-2.3 * ABS((H2O_TEMP - ***GP_DOM_DECOMP_TOPT***) / (***GP_DOM_DECOMP_TOPT*** - 5.0)))

3) flux calculations

DOM_DECOMP_POT = ***GP_calibDecomp*** * ***GP_DOM_RCDECOMP*** * DOM_QUALITY_CF * DOM_TEMP_CF * DEPOS_ORG_MAT * (Min(DOM_SED_AEROB_Z/ ***GP_DOM_MAXDEPTH***,1.0) * soil_MOIST_CF + ***GP_DOM_DECOMPRED*** * Min(DOM_SED_ANAEROB_Z/ ***HP_DOM_MAXDEPTH***,1.0))

DOM_DECOMP = (DOM_DECOMP_POT****DT*** > DEPOS_ORG_MAT) ? (DEPOS_ORG_MAT/***DT***) : (DOM_DECOMP_POT)

DOP_DECOMP = DOM_DECOMP * DOM_P_OM

4) attributes calculated after DOM/DOP updates, used in other modules

DOM_Z = DEPOS_ORG_MAT / DOM_BD

SED_ELEV = DOM_Z+Inorg_Z+SED_INACT_Z

DOM_P_OM = (DEPOS_ORG_MAT>0.0) ? (DOP / DEPOS_ORG_MAT) : (0.0)

TPsoil = DOP * ***CELL_SIZE*** + TP_SORB

TPtoSOIL = ((DEPOS_ORG_MAT * ***CELL_SIZE*** + DIM) > 0.0) ? (TPsoil / (DEPOS_ORG_MAT * ***CELL_SIZE*** + DIM)) : (0.0)

TPtoVOL = (***CELL_SIZE*** * DOM_Z>0.0) ? (TPsoil / (***CELL_SIZE*** * DOM_Z)) : (0.0)

TP_sedMin = (1.0 - ***HP_DOM_AEROBTHIN*** / ***HP_DOM_MAXDEPTH***) * DOP_DECOMP * ***CELL_SIZE***

TP_SED_WT = TP_SED_WT + TP_sedMin * ***DT***;

TP_SED_WT_AZ = TP_SED_WT_AZ + TP_sedMin * ***DT***;

TP_SED_CONC = (HYD_SED_WAT_VOL>0.0) ? (TP_SED_WT / HYD_SED_WAT_VOL) : (0.0)

TP_SEDWT_CONCACT = (HYD_DOM_ACTWAT_PRES > 0.0) ? (TP_SED_WT_AZ/HYD_DOM_ACTWAT_VOL) : (TP_SED_CONC)

TP_SEDWT_CONCACTMG = TP_SEDWT_CONCACT*1000.0

TP_Act_to_Tot = 1.0 / ***HP_TP_CONC_GRAD***

if there is no surface water present, assume that this relative contribution will be an additional sorbed component that is introduced to surface water column immediately upon hydration with surface water

TP_sfMin = ***HP_DOM_AEROBTHIN*** / ***HP_DOM_MAXDEPTH*** * DOP_DECOMP * ***CELL_SIZE***

TP_SF_WT = TP_SF_WT + TP_sfMin * ***DT***

TP_SFWT_CONC = (SFWT_VOL > 0.0) ? (TP_SF_WT/SFWT_VOL) : (0.0)

TP_SFWT_CONC_MG = (SURFACE_WAT > ***GP_DetentZ***) ? (TP_SFWT_CONC*1000.0) : (0.0)

used only for output as Performance Measure (with unit conversions)

P_SUM_CELL = ((C_ALG_P + NC_ALG_P) * 0.001 * ***CELL_SIZE*** + (mac_nph_P + mac_ph_P) * ***CELL_SIZE*** + TP_SORB + (FlocP + DOP) * ***CELL_SIZE*** + TP_SED_WT + TP_SF_WT) /CELL_SIZE * 1000.0

Constant attributes calculated only at model initialization (outside Module)

BulkD = input data

DOM_BD = input data

ELEVATION = input data

Bathymetry = input data

SED_INACT_Z = ELEVATION-Bathymetry+***DATUM_DISTANCE***- ***HP_DOM_MAXDEPTH***

Inorg_Z = (1.0 - (DOM_BD / BulkD)) * ***HP_DOM_MAXDEPTH***

DIM = (BulkD - DOM_BD) * ***HP_DOM_MAXDEPTH*** * ***CELL_SIZE***

External variables used

nphbio_mort_OM (see Macrophyte module)

nphbio_mort_P (see Macrophyte module)

FLOC_DEPO (see Floc module)

FlocP_DEPO (see Floc module)

soil_MOIST_CF (see Floc module)

TP_SEDWT_CONCACTMG (see TP/Salt module)

TP_SORB (see TP/Salt module)

UNSAT_DEPTH (see Hydrology module)

HYD_SED_WAT_VOL (see Hydrology module)

HYD_DOM_ACTWAT_VOL (see Hydrology module)

HYD_DOM_ACTWAT_PRES (see Hydrology module)

SFWT_VOL (see Hydrology module)

H2O_TEMP (see Hydrology module)

Soils Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
DOM_SED_AEROB_Z	attribute	m	Deposited Organic Matter SEDiment/soil AEROBic profile depth (Z) (incl. pore space)
DOM_SED_ANAEROB_Z	attribute	m	Deposited Organic Matter SEDiment/soil ANAEROBic profile depth (Z) (incl. pore space)
SED_ELEV	attribute	m	total land surface ELEVation of the entire SEDiment/soil complex, including model DATUM_DISTANCE depth below NGVD 1929)
TPtoVOL	attribute	kgP/m^3_soil	Total Phosphorus concentration in soil VOLume
DOM_P_OM	attribute	kgP/kgOM	Deposited Organic Matter Phoshorus concentration (relative to Organic Matter)
TPtoSOIL	attribute	kgP/kg_soil	Total Phosphorus concentration in SOIL mass
P_SUM_CELL	attribute	gP/m^2	SUM of all (biotic/abiotic) storages of Phosphorus (in CELLs) (for reporting only, thus units converted to gP/m^2)
DOM_QUALITY_CF	controlFunction	dimless	Deposited Organic Matter Control Function of degree of limitation by surrounding nutrient availability, i.e., QUALITY
DOM_TEMP_CF	controlFunction	dimless	Deposited Organic Matter Control Function of degree of TEMPerature limitation
DOP_DECOMP	rateActual	kgP/m^2/d	Deposited Organic Phosphorus DECOMPosition losses
TP_sedMin	rateActual	kgP/d	Total Phosphorus gained in sediment/soil water due to deposited organic matter (soil) Mineralization
TP_sfMin	rateActual	kgP/d	Total Phosphorus gained in surface water due to deposited organic matter (soil) Mineralization
DOM_fr_nphBio	rateActual	kgOM/m^2/d	Deposited Organic Matter gained from mortality of non-photosynthetic Biomass of macrophytes
DOM_DECOMP	rateActual	kgOM/m^2/d	Deposited Organic Matter DECOMPosition losses
DOM_FR_FLOC	rateActual	kgOM/m^2/d	Deposited Organic Matter gained FRom FLOCculent organic matter deposition
DOM_DECOMP_POT	rateP	kgOM/m	Deposited Organic Matter

	otential	^2/d	DECOMPosition POTential losses
DOP	state	kgP/m^2	Deposited Organic Phosphorus (better name is accreted organic phosphorus AOP) mass in upper soil zone (not including floc layer, sorbed P, nor water P storage)
DEPOS_ORG_MAT	state	kgOM/m^2	DEPOSited ORGanic MATter (better name is accreted organic matter, AOM) mass in upper soil zone (not including floc layer)
DOM_Z	state Convert	m	Deposited Organic Matter mass in upper soil zone converted to depth (Z) (organic component only, accounting for bulk density)
DIM	static	kg InorgM	Deposited Inorganic Matter mass in upper soil zone (inorganic component only)
Inorg_Z	static	m	deposited Inorganic matter in upper soil zone mass converted to depth (Z) (inorganic component only, accounting for bulk density)
ELEVATION	static	m	initial land surface ELEVATION of the entire sediment/soil complex (m NGVD 1929), not including the model DATUM_DISTANCE depth below NGVD 1929
Bathymetry	static	m	Bathymetry of estuarine areas, as depth of the sediment/soil surface below NGVD 1929, positive values not including the model DATUM_DISTANCE depth below NGVD 1929
SED_INACT_Z	static	m	SEDiment/soil INACTive Zone height (=distance below DOM_MAXDEPTH parameter)
BulkD	static	kgSoil/m^3soil	Bulk Density of soil
DOM_BD	static	kgOM/m^3soil	Bulk Density of (only) the Deposited Organic Matter component of the soil
TP_Act_to_Tot	static	dimless	Total Phosphorus concentration in the upper Active DOM zone relative to average concentration the Total soil/sediment zone down to base_datum; algorithm will change to a dynamic variable

Time series forcing data

none

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description

DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
GP_DATUM_DISTANCE	global	m	distance below NGVD'29 to base datum
GP_DetentZ	global	m	detention depth in a grid cell, below which surface flows do not occur
GP_calibDecomp	global	dimless	calibration parameter, multiply potential decomposition rate of organic matter
GP_DOM_RCDECOMP	global	1/d	Maximum observed/attainable specific rate of organic matter decomposition (w/o limitations)
GP_DOM_DECOMPRED	global	dimless	under anaerobic conditions, proportional reduction of the maximum rate of aerobic decomposition
GP_DOM_decomp_coef	global	dimless	parameter for exp function describing decomposition kinetics
GP_DOM_DECOMP_POPT	global	mg/L	Optimal phosphorus concentration in water for maximal decomposition of organic matter
GP_DOM_DECOMP_TOPT	global	deg C	Optimal temperature for maximal decomposition of organic matter
GP_sorbToTP	global	dimless	initial condition only, the ratio of sorbed phosphorus to total phosphorus in soil

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description
HP_DOM_MAXDEPTH	habsp ec	m	Maximum depth (positive, from sediment surface) of Deposited Organic Matter to consider in model. This determines the depth of the active DOM zone for all model dynamics via: 1) decomposition, 2) sorption/desorption of nutrients, and 3) nutrient uptake by macrophytes. This generally should be <= the max root depth parm (less than root depth in case of trees).
HP_DOM_AEROBTHIN	habsp ec	m	The thin aerobic zone in a flooded wetland. Note that aerobic total depth is defined to include any zone of soil/sediment that is unsaturated or devoid of water.
HP_TP_CONC_GRAD	habsp ec	dimless	For concentration gradient, provide the ratio of this nutrient in the inactive DOM zone to that in the active DOM zone. Used in partitioning the mass of sediment nutrients to different concentrations in the shallow active DOM zone and the deeper inactive zone.

Intrinsic C or ELM functions

exp(x) = Exp(x) => e raised to the x^{th} power

Max(x,y) => maximum of variable x or y

Min(x,y) => minimum of variable x or y

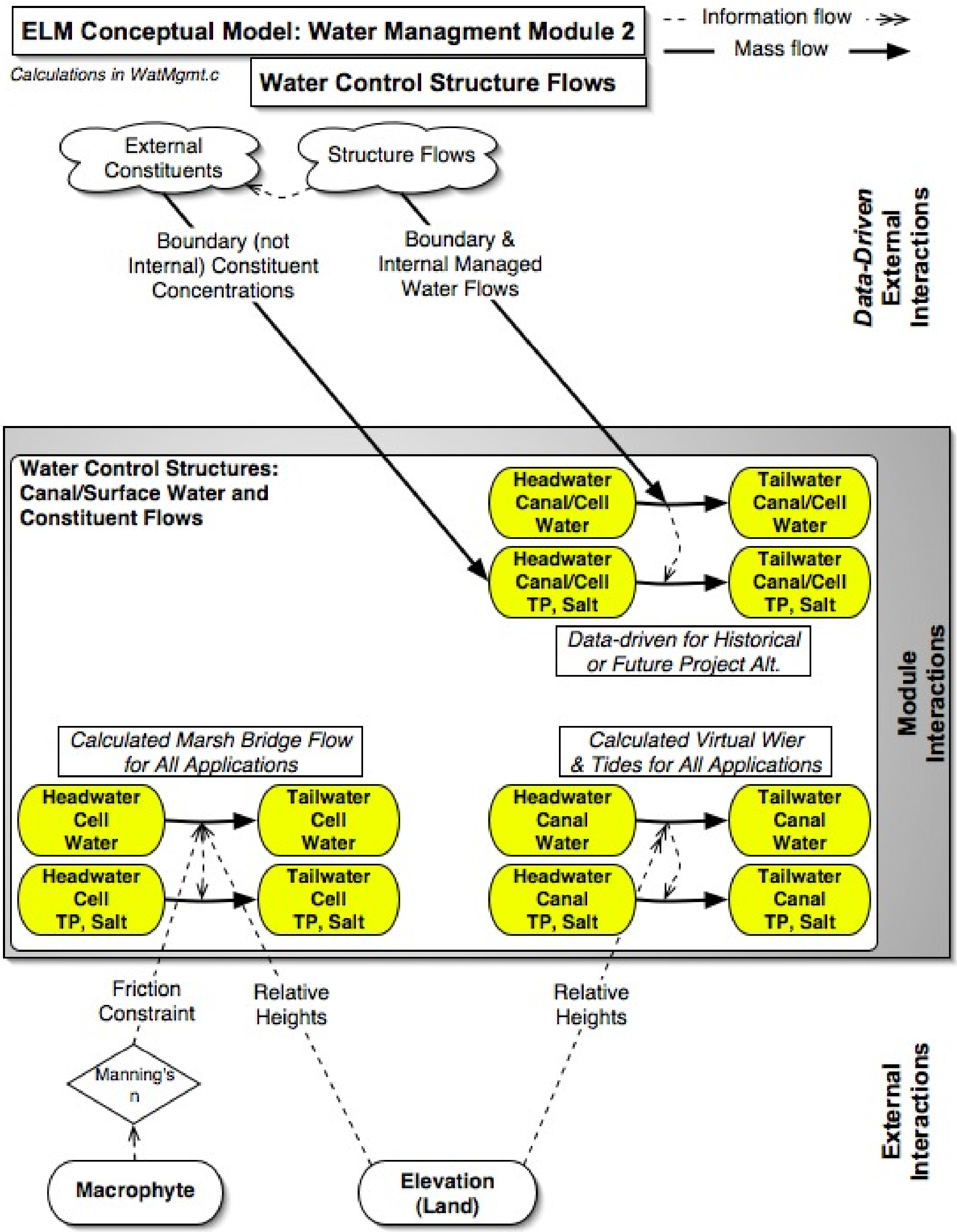
(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

ABS(x) => absolute value of (x)

5.7 Horizontal solutions

These modules calculate spatial flows of surface water, groundwater, and associated constituents (phosphorus and salt/tracer) in the (mostly) horizontal dimensions across raster grid cells and vector canals.

5.7.1 Water management: Structure flows module



Overview: Water Control Structure Flows Module

The Water Management Modules provide the mechanisms for distributing managed flows of water and constituents (phosphorus and salt/tracer) in a network of canals, levees, and water control structures. This Water Control Structure Flows Module describes the water and constituent flows into and out of canals and grid cells through point water control structures. All managed daily flows are derived from either historical observations or output from other models such as the South Florida Water Management Model (SFWMM), but un-managed flows are calculated internal to the model.

Water Control Structure Flows Module Description

The attributes of the water control structures are defined in a relational (FilemakerPro) database, and exported into an ASCII (text) input file for the model. Among the variety of attributes in this database are the definitions of the source (canal ID or cell ID[12]) and destination (canal ID or cell ID) water and constituent storages. The database also defines whether flows are to be driven by time-series input data or to be calculated in the model. As indicated in the Water Management Canal-Marsh Flux Module section, because some canals extend over large distances, the model segments a number of Everglades canal reaches into model canal reaches that are separated by “virtual” water control structures that equilibrate stages in two canals at every time step. This segmentation minimizes the potential grid-cell dispersion of constituents (nutrients and salt/tracer) from canals along very long canal reaches, as homogeneity of constituents is assumed along the length of the reach.

All managed water control structures (i.e., “real-world” structures) require daily time series data from historical observations or output from other models such as the SFWMM. Additionally, any water control structure that introduces water into the model domain must have some estimate of the associated constituents to flux with that “new” water. The constituent concentration may either be a fixed, long term mean value, or a daily time series of concentrations (derived from observations or from other models). Daily water and constituent flows are passed through a water control structure using one of four source-destination relationships: 1) flow from a canal to a canal, 2) flow from a cell to a cell, 3) flow from a canal to a cell, or 4) flow from a cell to a canal.

The data-driven flows are simple functions of the input data, with checks on any source-volume constraint. External boundary condition flows (into or out of the active domain of the model) are fluxes to or from a reserved cell (row 1, column 1) that is outside of the model domain.

In the case of “virtual” structures that equilibrate two canal reaches (that are portions of a longer, continuous “real-world” canal), a simple mass-balance equilibrium is sought between the two segments at each canal time step. The elevation drop along the length of the reach from the upstream to downstream end is known, and the land surface height at the midpoint each canal reach is used in estimating stage along both continuous reaches: stages based on those elevations are equilibrated at every time step (in the positive

[12] The cell ID is the row and column grid location, which is calculated in the database from the geographic coordinates of the structure, and is thus independent of the scale of the model application.

downstream direction only). In the case of an under-bridge “virtual” structure between wetland grid cells, the overland flow equation for grid cell fluxes is called to calculate the overland flow using an open-water Manning’s n coeficient (see Surface Water Raster Flux Module for equation description). In another use of virtual structures, tidal boundary conditions are imposed with a data-driven head/tail water target stage that is imposed on virtual structures associated with vectors of tidal rivers/creeks (aka “canals”) and cells external to the model domain. A long-term monthly mean tidal stage recurs annually through use of a input graph function, and the data are interpolated to daily head or the tail water target stages for the river vector. A high flow coefficient is imposed on the potential flux due to the head difference between target and the internal vector, exchanging water between the river vector and the target. A constant salinity selected by the user is imposed on each tidal flux. As with any “canal” vector, river vectors are segmented and joined by equilibrating virtual structures as described above.

Constraints for mass balance are imposed on the data-driven and the calculated water control structure flows during each time step, preventing head reversals or flows greater than the volume available in the donor grid cell or canal. Again, mass of constituents (nutrients, salt/tracer) is passed along in a mass-balance calculation based upon the water volume flux from the source storage.

Water Control Structure Flows Module Equations

Flux calculations

The below calculations are performed inside a ("while") loop through each individual water control structure.
While most flows are data-driven using either historical observations or output from other models (primarily the SFWMM), there are special cases of calculated flows (virtual structure flows between marsh cells (under-bridge) and canal-canal or canal->cell virtual structure flows).
Depending on the source and destination of a water control structure, there are four combinations of canal and grid-cell flows through the structures.

Canal-to-canal flow (always internal to model domain)
##
Calculate the data-driven flow demand through the current structure during this iteration
flow = arrayPump * ***canstep***

In a cycle across all structures, the current iteration flow is summed with any other (current iteration) data-driven flows from the current source-water reach
ChanHistOut = ChanHistOut + flow

If the sum of all data-driven outflows from the canal reach during this iteration exceeds the volume available, all flows are reduced by the necessary (equal) proportion for mass balance (and warnings are printed to the file "Driver1.out").
The mass of constituents are calculated for each flow in a mass balance transfer.
After completing this cycle through all outflows from a reach, (and reducing the flow volumes if necessary), the actual water volume and constituent mass flows are summed for use in the Water Management Canal-Marsh Flux Module. However, water volumes and constituent masses flowing into any grid-cell destinations update those cell storages at this point.
Once processed through such a cycle, a structure flow from the source canal reach is not processed again.
##
Calculate flow if current structure is a virtual structure.

Virtual structures are always processed AFTER all data-driven demands are met (due to omission from cycling through the structure-list during any volume-available checks, and due to their order in the water control structure list).
HeadH_drop = 0.5 * elev_drop_fr

HeadT_drop = 0.5 * elev_drop_to

In both head and tail, add net data-driven flows to determine hydraulic potential (grid cell elevation, SED_ELEV, is at water control structure)
HeadH = HeadH_drop + SED_ELEV - depth_fr + wat_depth_fr + (sumHistIn_fr - sumHistOut_fr)/area_fr

In tailwater only, check to see if other virtual struct has added water already (cumulative "sumRuleIn"), add to head
HeadT = -HeadT_drop + SED_ELEV - depth_to + wat_depth_to + (sumRuleIn + sumHistIn_to - sumHistOut_to)/area_to

Flow is only considered in the positive (head to tail water) direction
flow = area_fr * area_to / (area_fr + area_to) * (HeadH - HeadT)

The actual water volume and constituent mass flows are summed (including data-driven flows) for use in the Water Management Canal-Marsh Flux Module.

Cell-to-cell flow (can involve flows to/from cells external to model domain)
##
Calculate the data-driven flow demand through the current structure during this iteration
flow = arrayPump * ***canstep***

Unlike a canal reach, a single grid cell can be source-water for at most one water control structure - a check is made to ensure the flow is not greater than the currently available volume in the cell.
The water volume flow is used to update the volumes in the source and destination grid cells, along with sums of the constituent mass, for use in the Water Management Canal-Marsh Flux Module.
##
Calculate flow if current structure is a virtual structure.
The only case allowed for here is under-bridge flow (e.g., Alligator Alley bridges) parameterized with an model domain-wide array of Manning's n that is encoded as open-water, n=0.05.
Using water depths and elevations of the source and destinations cells, a call is made to the raster surface water flux functions (see Surface Water Raster Flux Module), updating water and constituents in the source and recipient cells.

Canal-to-cell flow (can involve flows to cells external to model domain)
##
Calculate the data-driven flow demand through the current structure during this iteration
flow = arrayPump * ***canstep***

In a cycle across all structures, the current iteration flow is summed with any other (current iteration) data-driven flows from the current source-water reach
ChanHistOut = ChanHistOut + flow

If the sum of all data-driven outflows from the canal reach during this iteration exceeds the volume available, all flows are reduced by the necessary (equal) proportion for mass balance (and warnings are printed to the file "Driver1.out").
The mass of constituents are calculated for each flow in a mass balance transfer.
After completing this cycle through all outflows from a reach, (and reducing the flow volumes if necessary), the actual water volume and constituent mass flows are summed for use in the

Water Management Canal-Marsh Flux Module. However, water volumes and constituent masses flowing into any grid-cell destinations update those cell storages at this point.
Once processed through such a cycle, a structure flow from the source canal reach is not processed again.

Cell-to-canal flow (can involve flows from cells external to model domain)
##
Calculate the data-driven flow demand through the current structure during this iteration
flow = arrayPump * ***canstep***

Unlike a canal reach, a single grid cell can be source-water for at most one water control structure - a check is made to ensure the flow is not greater than the currently available volume in the cell.
The water volume flow is used to update the volumes in the source and destination grid cells, along with sums of the constituent mass, for use in the Water Management Canal-Marsh Flux Module.

Process the next water control structure within the ("while") loop

External cell-based variables used

SED_ELEV (see Soils module)

SURFACE_WAT (see Hydrology module)

HYD_MANNINGS_N (see Hydrology module)

SALT_SURF_WT (see Salt/Tracer module)

TP_SF_WT (see Phosphorus module)

External canal-based variables used

none (in abbreviated equations)

Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
flow	RateActual	m^3/***canstep***	water flow volume through structure for an iteration
ChanHistOut	attribute	m^3	temporary variable, summing all data-driven flows during one iteration from a particular source canal or grid-cell
elev_drop_fr	attribute	m	land surface elevation difference from beginning to end of a source canal reach
elev_drop_to	attribute	m	land surface elevation difference from beginning to end of a destination canal reach
HeadH	attribute	m	Hydraulic Head in Headwater (source)
HeadT	attribute	m	Hydraulic Head in Tailwater (destination)

Time series forcing data

Variable Name	Type	Units	Description
arrayPump	RatePotenti	m^3/d	input data array of daily time-series water volume flows through managed

	al		structures
arrayP	attribute	kgP/m^3	input data array of daily time-series of Total Phosphorus concentration associated with structures that flow into model from external regions (variable not used in abbreviated equations)
arrayS	attribute	kgSalt/m^3	input data array of daily time-series of Salt/tracer concentration associated with structures that flow into model from external regions (variable not used in abbreviated equations)

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
hyd_iter	global	dimless	number of horizontal iterations per ***DT***
canstep= DT/hyd_iter	local	day	time step for horizontal canal solutions

Static canal-specific parameters

Parameter Name	Type	Units	Description
area_fr, area_to	attribute	m^2	area of entire canal reach, the source (fr), destination (to) reaches

Static structure-specific parameters

none of below parameters used in abbreviated equations

Parameter Name	Type	Units	Description
#flag	attribute	dimless	attribute indicating operational status of structure (<0 = off, 0 = calculated, >0 = data-driven)
#S_nam	attribute	dimless	structure name
#histTP	attribute	dimless or mgP/L	attribute indicating a single long-term mean TP concentration, or pointer to time series input data
#histTS	attribute	dimless or gSalt/L	attribute indicating a single long-term mean Salt/tracer concentration, or pointer to time series input data
#str_cell_i	attribute	dimless	row location of structure
#str_cell_j	attribute	dimless	column location of structure
#canal_fr	attribute	dimless	canal ID of structure water source
#canal_to	attribute	dimless	canal ID of structure water destination
#cell_i_fr	attribute	dimless	row location of structure water source
#cell_j_fr	attribu	dimless	column location of structure water

	te		source
#cell_i_to	attribu te	dimless	row location of structure water destination
#cell_j_to	attribu te	dimless	column location of structure water destination

Static habitat-specific parameters (linked to HAB value of grid-cell)

none

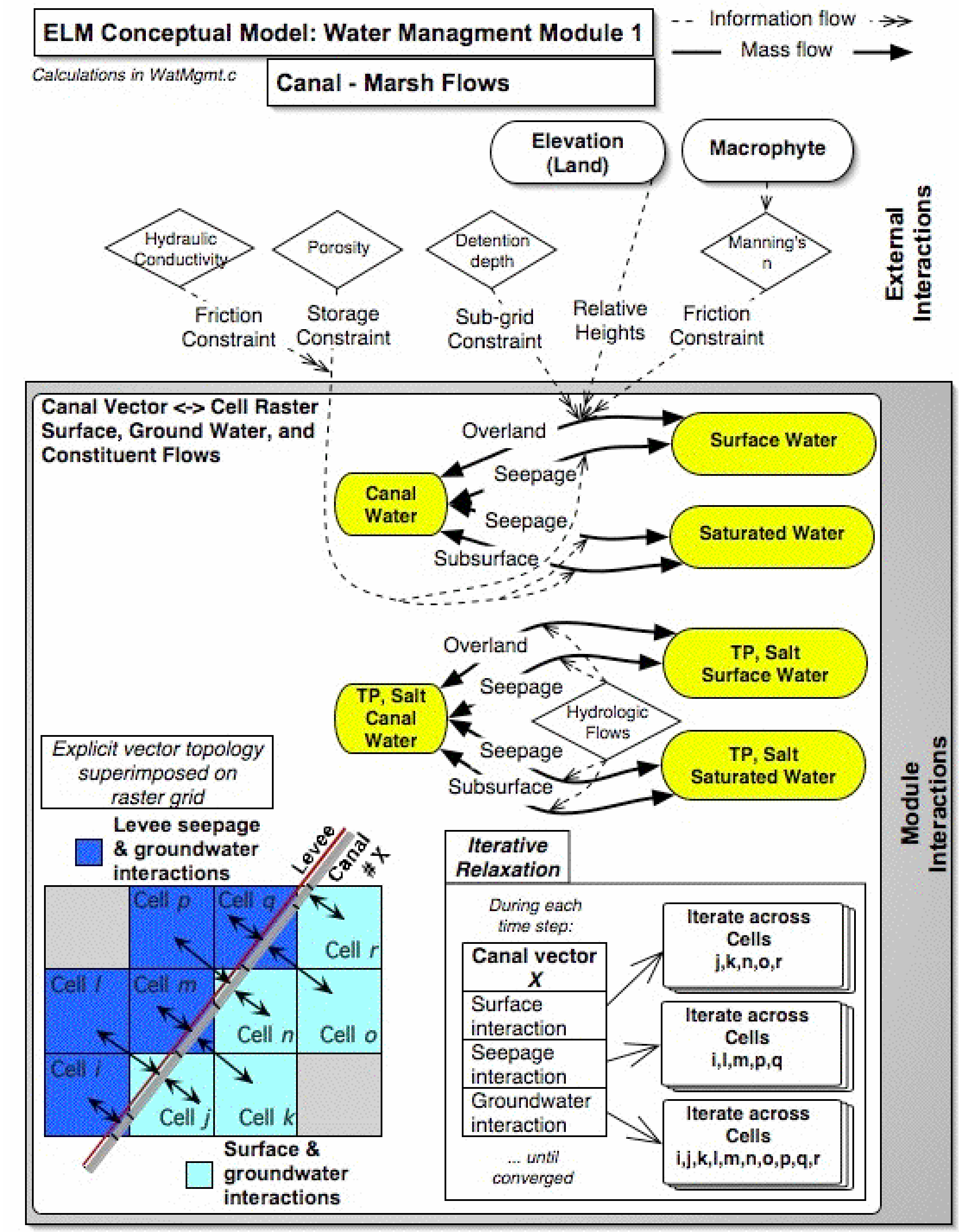
Static spatially-distributed parameters

none

Intrinsic C or ELM functions

none (in abbreviated equations)

5.7.2 Water management: Canal-marsh flux module



Overview: Water Management Canal-Marsh Flux Module

The Water Management Modules provide the mechanisms for distributing managed flows of water and constituents in a network of canals, levees, and water control structures. This Canal-Marsh Flux Module dynamically exchanges surface/ground- water and constituents among the canal vectors and the raster grid cells. The topology of the network is calculated such that the vectors overlie cells in their true geographic orientation and maintain the correct area of interaction among the raster and vector object types. Flux equations determine the flow of water and constituents along canals, with exchange of water and nutrients among grid cells and canal vectors via overland, seepage, or groundwater flow. The Water Management Water Control Structure Flows Module describes the flows into and out of canals and grid cells through point water control structures.

Water Management Canal-Marsh Flux Module Description

The attributes of the canal reaches in the network are defined in an ASCII (text) datafile that is input to the model. (A script is used to import the canal data into the GRASS GIS for visualization and editing of the canal network topology). Canal reaches are assumed to have homogenous width, depth, slope, levee (if present) hydraulic conductivity, and constituent concentration throughout the length of the canal reach. A levee is assumed to have negligible width. At initialization time of the model, the geometric relationships of the canal vectors and raster grid cells is calculated. Canal reaches are defined by vectors of any shape, beginning and ending with water control structure points. Because some canals extend over large distances, the model segments a number of Everglades canal reaches into model canal reaches that are separated by “virtual” water control structures (see Water Management Water Control Structure Flows Module). This segmentation minimizes the potential grid-cell dispersion of constituents (nutrients and salt/tracer) from canals along very long canal reaches, as homogeneity of constituents is assumed along the length of the reach.

The exact geographic coordinates of the multiple points forming a curved or straight canal reach (and the exact locations of grid cells) are used to determine the area of interactions among each segment (piece of a reach along a grid cell) of a canal reach vector with the adjacent grid cells. (Canal reaches intersect grid cells at any angle, and the area of interaction is known from the geometry). In this scheme, the mode of interaction of a grid cell with a canal (e.g., levee seepage vs. overland flow) is determined by the placement of the vector canal (and levee, if any) relative to the center of the grid cell. By comparing where a vector segment lies relative to the center of a cell, it is first determined whether a cell should be marked as being to the left or right of the vector (as shown in conceptual model diagram). For example, if more than half of the cell area lies to the right of the vector, then the cell is assigned as a right cell. Note (as shown in the figure) that it is not only the transected cells that can be marked as interacting with the canal vector. This set of interacting grid cells becomes associated as an object (in a C data structure) for a canal reach that defines its interacting cells. Based upon this determination of interacting cells, the presence or absence of a levee(s) associated with each canal reach (none, both sides, left side, right side) is used to calculate (and statically store) the allowable flow directions in the raster grid cells (modifying the “ON_MAP” array attributes for use in the Surface Water Raster Flux Module).

While we developed this unique raster-vector topology for cell-canal relationships, the ELM uses the fundamentals of the mass balance approach for canal-cell fluxes that was originally developed for the South Florida Water Management Model. This method is applied to the Water Management Canal-Marsh Module to calculate the exchange of water and constituents between a vector canal reach and the multiple grid cells that interact with that reach. Additions or subtractions to/from the canal reach from water control structure flows are known at the start of a canal time step (Water Management Water Control Structure Flows Module). In an iterative relaxation (not true equilibration) procedure during a single canal time step, a new canal depth is estimated and the canal-cell exchanges along the entire reach are calculated. Comparing the new estimated depth with the past depth adjusted for all flow exchanges, the error in the estimate is quickly decreased to a threshold value (10 microns in recent versions, including current) to converge on a solution. In calculating the exchange of canal surface waters with either surface water or subsurface groundwater in interacting cells, the model uses simple applications of the Manning's equation or Darcy's equation, respectively within an explicit, finite-difference framework (see Surface Water and Groundwater Flux Modules for equations and further background). Surface water exchange can occur between surface storages in the canals and in interacting cells. Levee-seepage exchange occurs between surface water in canals and surface or groundwater in interacting cells. Groundwater storage in interacting grid cells can exchange with surface water in canals. Constraints for stability and mass balance are imposed on the calculated flux during each time step, preventing head reversals or flows greater than the volume available in the donor grid cell or canal. Mass of constituents (nutrients, salt/tracer) is passed along in a mass-balance calculation based upon the water volume flux between cells and canal.

Water Management Canal-Marsh Flux Module Equations

Geometry calculations

At model initialization time (Canal_Network_Init function), the geometry of canal and grid cell attributes is used to determines which grid cells interact with canal vectors, and their mode of interaction.

A canal reach is defined by two (upstream & downstream) water control structures, with each reach having a unique numeric ID.

Canal vector geographic coordinates are defined in the input CanalData.chan text file (see DataRead Module).

The water control structures may be actual water management structures, or "virtual structures" used in partitioning long, continuous actual canals into multiple model reaches.

Canal reaches may be straight lines or curves, with the area of interaction of (grid-cell associated) segments of each with adjoining grid cells known from the geometry calculations during initialization.

Flux calculations

The below calculations are performed inside an iterative ("do-while") relaxation routine for EACH individual canal reach, exchanging water among the canal reach and adjoining cells, then estimating the new canal water depth.

*## After each iteration, the estimate of the new canal depth is compared to the old-depth-plus the (positive/negative) canal-cell and water control structure exchanges: when the error between those estimates becomes less than the chosen threshold **(F_ERROR**, in input file= CanalData.chan), we have the solution for the new canal depth.*

This "iterative relaxation" routine is the same concept that is documented for the South Florida Water Management Model.

This procedure is calculated only for grid cells that are inside the active model domain (where ON_MAP is true, >0).

Start the iterative relaxation routine
At the start of an iteration of the relaxation routine, make a new estimate of the water depth in the canal.
The first estimate is a very crude one, and the relaxation routine refines that quickly by modifying "factor" based upon the error in the last iteration. (In this, "factor" is increased/decreased or changed in sign, depending on the direction of the error).
CanWatDep = CanWatDep + factor

During one iteration, start of the loop across all grid cells belonging to a canal reach
cellLoc_i = address of grid cell at row x, column y
account for (non-zero) increased roughness associated with edge of canal

SW_coef = (HYD_MANNINGS_N[cellLoc_i] == 0.0) ? 0 : ***SW_flow_coef*** / (***edgeMann*** > 0 ? (HYD_MANNINGS_N[cellLoc_i] + ***edgeMann***)/2.0 : HYD_MANNINGS_N[cellLoc_i])

GW_head = SAT_WATER[cellLoc_i]/***HP_HYD_POROSITY***[cellLoc_i]

tot_head = SURFACE_WAT[cellLoc_i] + SED_ELEV[cellLoc_i]

CH_bottElev = SED_ELEV[cellLoc_i] - ***depth***

dh = (CH_bottElev + CanWatDep) - tot_head

H_rad_ch = (***seg_area*** * ramp(CanWatDep - ***depth***) + SURFACE_WAT[cellLoc_i] * (CELL_SIZE-seg_area)) / ***CELL_SIZE***

H_rad_cell = (***seg_area*** * ramp(CanWatDep - ***depth***) + SURFACE_WAT[cellLoc_i] * (CELL_SIZE- ***seg_area***)) / ***CELL_SIZE***

For positive flows from canal (dh > 0.0), two calculations for cross sectional heights:

h_GWflow = Min(***depth***, CanWatDep)

h_SPflow = Max(CH_bottElev + CanWatDep - SED_ELEV[cellLoc_i], 0.0);

For negative flows into canal (dh < 0.0), two calculations for cross sectional heights:

h_GWflow = Max(GW_head-CH_bottElev, 0.0);

h_SPflow = Max(tot_head-SED_ELEV[cellLoc_i], 0.0);

Depending on the location of levee(s), if any, a choice of canal-cell flux calculations is made:

Levee on both sides of canal reach
*## Levee seepage, fluxL, and Groundwater, fluxG, flows along both sides of reach */*

fluxL = (h_SPflow > 0.0) ? (dh * ***I_Length*** * ***SPG_coef*** / (0.5****celWid***) * h_SPflow * ***canstep***) : (0.0);

fluxG = (h_GWflow > 0.0) ? (dh * ***I_Length*** * ***GW_coef*** / (0.5****celWid***) * h_GWflow * ***canstep***) : (0.0);

Levee absent from both sides of canal reach
*## Overland Surface flow, fluxS, along both sides of reach */*
For positive slope, flux from canal (dh > 0):

fluxS = sgn(dh) * SW_coef * pow(H_rad_cell, ***GP_mannDepthPow***) * sqrt(Abs(dh)) * ***canstep***)

For negative slope, flux from cell into canal provided SURFACE_WAT[cellLoc_i] > DetentZ :

fluxS = sgn(dh) * SW_coef * pow(H_rad_cell, ***GP_mannDepthPow***) * sqrt(Abs(dh)) * ***canstep***)

Constrain flow from cell to volume available
if (-fluxS > (SURFACE_WAT[cellLoc_i]-***GP_DetentZ***) ****CELL_SIZE***) fluxS = -(SURFACE_WAT[cellLoc_i]-DetentZ)****CELL_SIZE***;

Subsurface Groundwater, fluxG, flow along both sides of reach
fluxG = (h_GWflow > 0.0) ? (dh * ***l_Length*** * ***GW_coef*** / (0.5****celWid***) * h_GWflow * ***canstep***) : (0.0);

Levee on left side of canal reach
*Overland flow, fluxS, along right side of reach */*
For positive slope, flux from canal (dh > 0):
fluxS = sgn(dh) * SW_coef * pow(H_rad_cell, ***GP_mannDepthPow***) * sqrt(Abs(dh)) * ***canstep***)

For negative slope, flux from cell into canal provided SURFACE_WAT[cellLoc_i] > **GP_*DetentZ*** *:*
fluxS = sgn(dh) * SW_coef * pow(H_rad_cell, ***GP_mannDepthPow***) * sqrt(Abs(dh)) * ***canstep***)

Constrain flow from cell to volume available
if (-fluxS > (SURFACE_WAT[cellLoc_i]- ***GP_DetentZ***) ****CELL_SIZE***) fluxS = -(SURFACE_WAT[cellLoc_i]- ***GP_DetentZ***)****CELL_SIZE***

Levee seepage flow, fluxL, along left side of reach
fluxL = (h_SPflow > 0.0) ? (dh * ***l_Length*** * ***SPG_coef*** / (0.5****celWid***) * h_SPflow * ***canstep***) : (0.0)

Subsurface Groundwater, fluxG, flow along both sides of reach
fluxG = (h_GWflow > 0.0) ? (dh * ***l_Length*** * ***GW_coef*** / (0.5****celWid***) * h_GWflow * ***canstep***) : (0.0);

Levee on right side of canal reach
*Overland flow, fluxS, along left side of reach */*
For positive slope, flux from canal (dh > 0):
fluxS = sgn(dh) * SW_coef * pow(H_rad_cell, ***GP_mannDepthPow***) * sqrt(Abs(dh)) * ***canstep***)

For negative slope, flux from cell into canal provided SURFACE_WAT[cellLoc_i] > ***DetentZ****:*
fluxS = sgn(dh) * SW_coef * pow(H_rad_cell, ***GP_mannDepthPow***) * sqrt(Abs(dh)) * ***canstep***)

Constrain flow from cell to volume available
if (-fluxS > (SURFACE_WAT[cellLoc_i]- ***GP_DetentZ***) ****CELL_SIZE***) fluxS = -(SURFACE_WAT[cellLoc_i]- ***GP_DetentZ***)****CELL_SIZE***

Levee seepage flow, fluxL, along right side of reach
fluxL = (h_SPflow > 0.0) ? (dh * ***l_Length*** * ***SPG_coef*** / (0.5****celWid***) * h_SPflow * ***canstep***) : (0.0)

Subsurface Groundwater, fluxG, flow along both sides of reach
fluxG = (h_GWflow > 0.0) ? (dh * ***l_Length*** * ***GW_coef*** / (0.5****celWid***) * h_GWflow * ***canstep***) : (0.0);

After fluxing water between a grid cell and canal reach, make three head and volume flow constraints:
The first constraint reduces the magnitude of the positive surface flux if the receiving cell would have a hydraulic head greater than the canal.

The second constaint reduces the magnitude of the negative surface flux if the receiving canal would have a hydraulic head greater than the cell.
The third constraint reduces the magnitude of the positive fluxes if the canal would be drained below its minimum depth.

Ending the loop across all grid cells belonging to a canal reach,
sum the total canal-cell fluxes along all grid cells of the canal reach during this iteration

T_flux_S = T_flux_S + fluxS

T_flux_G = T_flux_G + fluxG

T_flux_L = T_flux_L + fluxL

Now that all of the grid cell-canal fluxes have been estimated, determine the error between the newly estimated canal water depth and the previous canal water depth plus calculated flows.

error = (CanWatDep - wat_depth) - (Qin - Qout - T_flux_S - T_flux_G - T_flux_L)/ ***area***;

Still in the iterative relaxation routine, this error is used in start (top) of next iteration in the iterative relaxation routine above

At this point after solution convergence in the iterative relaxation routine, the canal reach water depth is updated with that from the converged solution.

wat_depth = CanWatDep

The water and constituent state variables in the canal reach and grid cells are updated in a set of mass balance calculations using the mass in the donor cell or canal storage variables and the water flux between those storages.

External cell-based variables used

SED_ELEV (see Soils module)

SURFACE_WAT (see Hydrology module)

SAT_WATER (see Hydrology module)

HYD_MANNINGS_N (see Hydrology module)

SALT_SURF_WT (see Salt/Tracer module)

TP_SF_WT (see Phosphorus module)

SALT_SED_WT (see Salt/Tracer module)

TP_SED_WT (see Phosphorus module)

External canal-based variables used

Qin (see Water Management Water Control Structure Flows module)

Qout (see Water Management Water Control Structure Flows module)

Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
SW_coef	attribute	m^0.5 sec/(d/(m^(1/3)))	Surface Water flow coefficient (includes dynamic Manning's n)
GW_head	attribute	m	groundwater head

tot_head	attribute	m	total hydraulic head
CH_bottElev	attribute	m	elev of bottom of canal at cell location
wat_depth	attribute	m	depth of water in canal from the previous canal time step
CanWatDep	attribute	m	estimated depth of water in canal during relaxation procedure
factor	attribute	dimless	the factor by which the CanWatDepth estimate is additively increased/decreased after an iteration of the relaxation routine
error	attribute	m	error between the newly estimated canal water depth and the previous canal water depth plus calculated flows
dh	attribute	m	difference in depths between canal reach and cell
H_rad_ch	attribute	m	hydraulic radius of canal reach for overland flow out of reach (canal and cell share same)
H_rad_cell	attribute	m	hydraulic radius of cell for overland flow into canal reach (canal and cell share same)
h_GWflow	attribute	m	height of the water cross section associated with the groundwater reach-cell flow
h_SPflow	attribute	m	height of the water cross section associated with the seepage reach-cell flow
fluxS	RateActual	m^3/d	Surface water flux between a segment of a canal reach and grid cell
fluxL	RateActual	m^3/d	Levee-seepage water flux between a segment of a canal reach and grid cell
fluxG	RateActual	m^3/d	Groundwater flux between a segment of a canal reach and grid cell
T_flux_S	RateActual	m^3/d	Total sum of Surface water fluxes between an entire canal reach and all grid cells associated with that reach
T_flux_L	RateActual	m^3/d	Total sum of Levee-seepage water fluxes between an entire canal reach and all grid cells associated with that reach
T_flux_G	RateActual	m^3/d	Total sum of Groundwater fluxes between an entire canal reach and all grid cells associated with that reach

Time series forcing data

none

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions

hyd_iter	global	dimless	number of horizontal iterations per ***DT***
canstep= DT/hyd_iter	local	day	time step for horizontal canal solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
celWid= CELL_SIZE^0.5	local	m	width of grid cell
sec_per_day = 86400	local	sec	number of seconds in a day
GP_DetentZ	global	m	detention depth in a grid cell, below which surface flows do not occur
GP_mannDepthPow	global	dimless	power used in manning's equation water depth
GP_calibGWat	global	dimless	calibration parameter, multiply groundwater cell-cell flow calculation

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description
HYD_POROSITY	hab-spec	dimless	Porosity of the aquifer, average from the sediment to base datum. Field capacity = porosity - specific yield; ensure that alterations to porosity and specific yield are consistent in your parameterization. Must be non-zero.

Static spatially-distributed parameters

Parameter Name	Type	Units	Description
HYD_RCCONDUCT	distributed	m/d	HYDraulic CONDUCTvity Rate Constant of surficial aquifer
GW_coef= HYD_RCCONDUCT * GP_calibGWat * HYD_POROSITY	distributed	m/d	aggregated GroundWater flow coeficient

Static canal-global parameters

Parameter Name	Type	Units	Description
F_ERROR	attribute	m	maximum allowable error in estimate of new water height in the canal-cell iterations
C_F	attribute	dimless	flow acceleration parameter, reserved for sensitivity experiments only (=1.0)

Static canal-specific parameters

Parameter Name	Type	Units	Description
depth	attribute	m	depth of canal, from bottom to rim of canal reach (not including levee)
width	attribute	m	width of canal reach (negative widths cause reach to be ignored)
cond	attribute	m/d	levee hydraulic conductivity, calibration parameter
length	attribute	m	length of entire canal reach

area	attribute	m^2	area of entire canal reach = length * width
edgeMann	attribute	d/(m^ (-1/3))	Manning's n associated w/ edge of canal, to accommodate topographic lip/berm and/or denser veg along canal length
I_Length	attribute	m	mean length of cells along reach (cell-associated) segments
seg_area= ***I_Length * width***	attribute	m^2	mean area of reach segments along each reach
SW_flow_coef= ***sqrt(I_Length) * sec_per_day * C_F***	attribute	m^0.5 sec	overland flow coefficient (C_F is multiplier only used for sensitivity)
SPG_coef= ***cond * GP_calibGWat***	attribute	m/d	aggregated seepage flow coefficient

Intrinsic C or ELM functions

sgn(x) => returns the sign (positive or negative, -1 or 1) of (x)

Min(x,y) => minimum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

ABS(x) = Abs(x) => absolute value of (x)

ramp(x) => negative (x) set =0, otherwise =(x) {precaution only for infinitesimally negative values - mass balance is evaluated (always output in budg_XYZ output files) at multiple spatial scales (several cell, whole-domain) and temporal scales, w/o losses: computational error in water storage height is on the order of +/- 10 microns accumulated over 20 years, maximum magnitude of (positive/negative) error is on the order of 1 micron accumulated over a 30-day period}

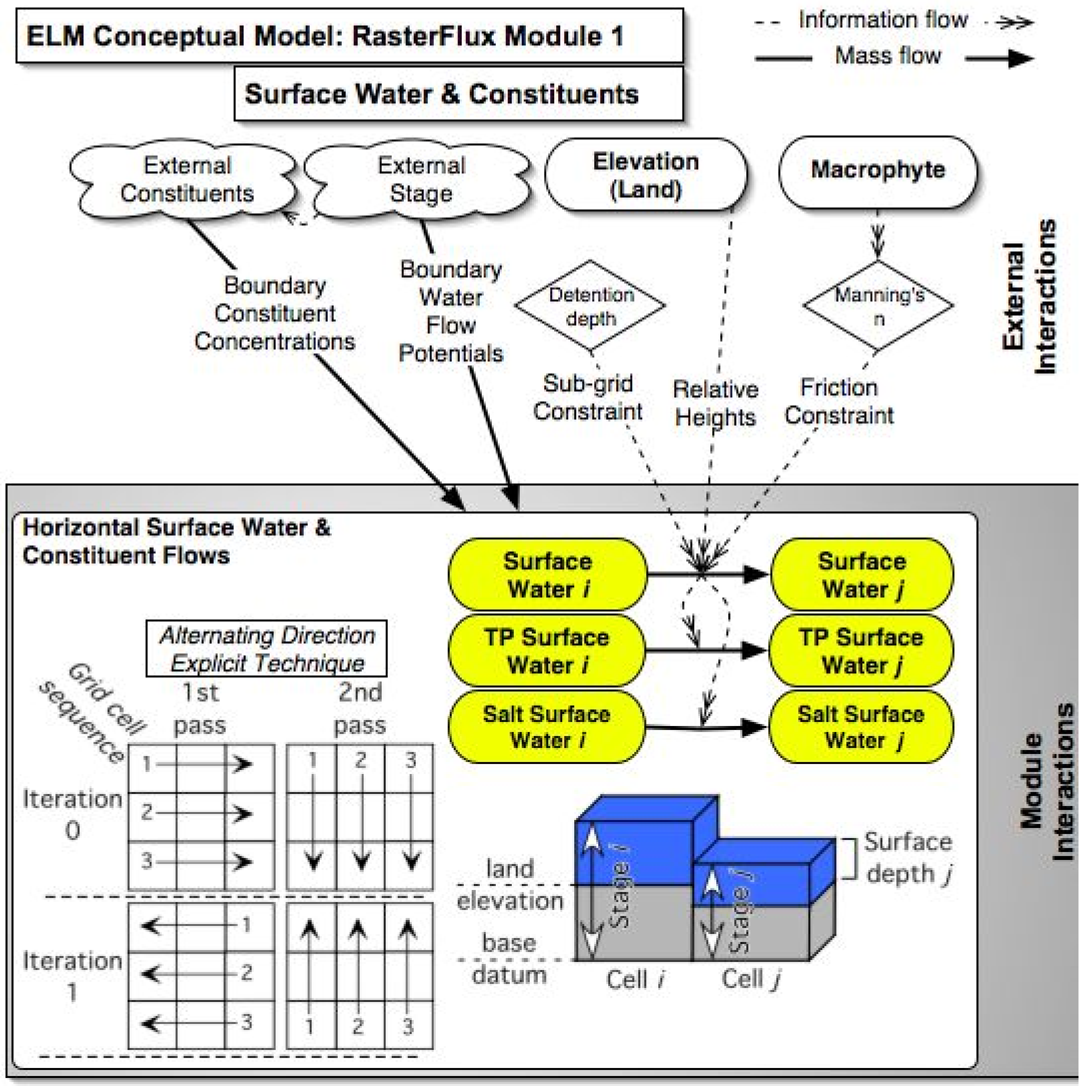
(x) != (y) => logical condition where (x) is not equal to (y)

if (x) equation => if (x) condition is true (==1), then execute "equation"

T(x,y) => single-integer array address of grid cell at location row x, column y (used in [cellLoc_])

sqrt(x) => square root of (x)

5.7.3 Overland flow module



External Interactions

Overview: Surface Water Raster Flux Module

This Surface Water Raster Flux Module serves to update the surface water storage state variable due to horizontal overland flow among (raster) grid cells. The (vertical)

Hydrology Module describes many of the dynamics associated with ELM hydrology, while this module description is specific to overland flow. These surface water flows are an important transport mechanism for constituents (phosphorus and salt/tracer) in the landscape, and canal fluxes can more rapidly transport water and constituents across long distances (see Water Management Modules). The overland surface flows are highly dependent upon the resistance to flow by macrophytes, while groundwater flows (Groundwater Raster Flux Module) and seepage through levees (Water Management Module) vary significantly across the region depending on aquifer (or levee) transmissivity.

Surface Water Raster Flux Module Description

Flow restrictions among grid cells are evaluated first. Based upon the geometry of levee vectors relative to square grid cells (calculated in the Water Management Module), grid cell flows may either not be allowed, allowed in the north-south direction, allowed in the east-west direction, or allowed in the direction of both axes. Flow restrictions between grid cells inside the model domain and grid cells outside the domain along the boundary are determined from a static input map layer: if overland surface flows are allowed, the stage and constituent concentration of an exterior boundary cell are determined. These stage data are daily values from another model such as the SFWMM.

The flow between two adjacent cells is determined from a simplification of the well-known open channel, diffusion flow model in an explicit, finite-difference framework. Omitting any inertial or acceleration terms, the continuity equation is simply a two-dimensional flux driven by differences in slope of the water surfaces. The flux between a pair of grid cells in the model domain’s array is described by the empirical Manning's equation for overland flow:

$$Q = \frac{D^{\frac{5}{3}} L^{\frac{1}{2}} \Delta h^{\frac{1}{2}}}{n}$$

where Q is the volumetric flow velocity ($m^3\ d^{-1}$), D is the water depth (= hydraulic radius, m) above ground elevation, L is the length of a grid cell (m), Δh is the difference (m) in water stage between the source and destination cells, and n is the empirically-derived Manning’s roughness coefficient. Using an explicit numerical method, the solution is iterated in both the row-wise and the column-wise directions during each time step, the direction alternates (east-west and west-east, north-south and south-north) after each time step. This Alternating Direction Explicit solution minimizes the directional bias that is associated with a uniform- direction solution. Constraints for stability and mass balance are imposed on the calculated flux during each time step, preventing head reversals or flows greater than the volume available in the donor grid cell. The mass of constituents (nutrients, salt/tracer) is passed along in a mass-balance calculation based upon the water volume flux between cells.

Numerical dispersion of constituents (due to grid scale and time step in the finite difference solution) is calculated, and numerical dispersive flux adjusted to equal that associated with a user-selected grid cell length using a simple Anti-Numerical Dispersion algorithm. This algorithm is extended to increase/decrease dispersion (via a dispersion number parameter) to approximate actual dispersive flux in the simulated system (Wool et al. in press).

Surface Water Raster Flux Module Equations

Flux calculations

All equations shown are calculated within an Alternating Direction (each iteration) spatial loop across model grid rows, columns
[cellLoci] defines model grid address of cell "i"
[cellLocj] defines model grid address of cell "j"
Flux is positive/negative, from cell "i" to cell "j"

Pairs of grid cells are checked for (static) flow attributes in the spatial loop.
For a cell at [cellLoc_], the possible flow attributes are:
ON_MAP[cellLoc_]=0 => External to the model active domain
ON_MAP[cellLoc_]=1 => Allow (internal) flow in no direction (due to calculated levee-interaction geometry)
ON_MAP[cellLoc_]=2 => Allow (internal) flow to east<->west (due to calculated levee-interaction geometry)
ON_MAP[cellLoc_]=3 => Allow (internal) flow to south<->north (due to calculated levee-interaction geometry)
ON_MAP[cellLoc_]=4 => Allow (internal) flow in all directions (due to calculated (no) levee-interaction geometry)

If a single cell in a pair is external to the model domain (example, ON_MAP[cellLoci]=0),
allowance of internal<->external flow depends on an attribute of the other cell (i.e., [cellLocj]):
BCondFlow[cellLocj]=1 => Allow no flows external to model domain
BCondFlow[cellLocj]=3 => Allow surface water flows to/from external boundary cell
BCondFlow[cellLocj]=4 => Allow groundwater flows to/from external boundary cell

The function "Flux_SWcells" calculates and returns a cell-to-cell Flux in height units (m)
The case is shown for when both cell i and j are internal to the model domain, with flow allowed between the cells.
When one of the cells is external to the domain, and the pair of cells has been defined as allowing surface water boundary flows, the stage of that external cell (cellLoci in this example) is estimated as: HEADi = SED_ELEV[cellLocj] + Max(SURFACE_WAT[cellLocj]-0.05,0.0)
Code exists, but is not executed in v2.2, to replace the estimated stage/head value with input data from another model (e.g., SFWMM).

MANNINGS_N = (***HYD_MANNINGS_N***[cellLoci] + ***HYD_MANNINGS_N***[cellLocj])/2.0

HEADi = SURFACE_WAT[cellLoci] + SED_ELEV[cellLoci]

HEADj = SURFACE_WAT[cellLocj] + SED_ELEV[cellLocj]

deltaHEAD = HEADi - HEADj

a_deltaHEAD = ABS (deltaHEAD)

For positive head differences (deltaHEAD > 0), execute these four equations:
if(SURFACE_WAT[cellLoci] < ***DetentZ***) ## do nothing (return a Flux value of 0.0)

Flux = (MANNINGS_N != 0) ? (pow(a_deltaHEAD, ***GP_mannHeadPow***) * ***sec_per_day*** / MANNINGS_N * pow(SURFACE_WAT[cellLoci], ***GP_mannDepthPow***)****step_Cell***) : (0.0)

Flux = (Flux > ramp(SURFACE_WAT[cellLoci] - ***GP_DetentZ***)) ? (ramp(SURFACE_WAT[cellLoci] - ***DetentZ***)) : (Flux)

if ((HEADi - Flux) < (HEADj + Flux))Flux = Min (deltaHEAD/2.0,
ramp(SURFACE_WAT[cellLoci] - ***GP_DetentZ***))

For negative head differences (deltaHEAD < 0), execute these four equations:
if (SURFACE_WAT[cellLocj] < ***GP_DetentZ***) ## do nothing (return a Flux value of 0.0)

Flux = (MANNINGS_N != 0) ? (- pow(a_deltaHEAD, ***GP_mannHeadPow***) * ***sec_per_day*** /
MANNINGS_N * pow(SURFACE_WAT[cellLocj], ***GP_mannDepthPow***)****step_Cell***) : (0.0)

Flux = (-Flux > ramp(SURFACE_WAT[cellLocj] - ***GP_DetentZ***)) ? (-
ramp(SURFACE_WAT[cellLocj] – ***GP_DetentZ***)) : (Flux)

if ((HEADi - Flux) > (HEADj + Flux)) Flux = - Min (a_deltaHEAD/2.0,
ramp(SURFACE_WAT[cellLocj] - ***GP_DetentZ***))

Result is the water flux between cells

The function "Flux_SWstuff" calculates the mass of constituents that move with the cell-to-cell Flux, updating the water and constituent state variables
Dispersion of constituents dependent on water velocity, calculated in "Disp_Calc" function
water velocity
veloc = Abs(Flux) * celWid/((Flux >0.0) ? (depth_i) : (depth_j)) / (sfstep)

numerical dispersion
disp_num = 0.5 * veloc * (celWid - veloc * sfstep)

velocity adjusted for numerical dispersion
veloc_adj = (veloc * celWid - disp_num)/celWid

Flux adjusted for numerical dispersion, and actual (parameter-based) dispersion
FluxAdj = dispParm_scaled * veloc_adj * sfstep * ((Flux >0.0) ? (depth_i) : (depth_j))/celWid

use adjusted Flux to determine the proportion of flow to use in constituent flux
fl_prop_i = (SURFACE_WAT[cellLoci]>0.0) ? (Max(Flux-FluxAdj,0.0) /
SURFACE_WAT[cellLoci]) : (0.0)

fl_prop_j = (SURFACE_WAT[cellLocj]>0.0) ? (Min(Flux+FluxAdj,0.0) /
SURFACE_WAT[cellLocj]) : (0.0)

fl_prop_i = Min(fl_prop_i, 1.0)

fl_prop_j = Min(fl_prop_j, 1.0)

For positive Flux values, execute these two equations to calculate mass of the constituent flux:
m1 = SALT_SURF_WT[cellLoci]*fl_prop_i

m3 = TP_SF_WT[cellLoci]*fl_prop_i

For negative Flux values, execute these two equations to calculate mass of the constituent flux:
m1 = SALT_SURF_WT[cellLocj]*fl_prop_j

m3 = TP_SF_WT[cellLocj]*fl_prop_j

update the constituent and water state variables
SALT_SURF_WT[cellLocj] += m1

TP_SF_WT[cellLocj] += m3

SALT_SURF_WT[cellLoci] -= m1

TP_SF_WT[cellLoci] -= m3

SURFACE_WAT[cellLocj] += Flux

SURFACE_WAT[cellLoci] -= Flux

External variables used

SED_ELEV (see Soils module)

HYD_MANNINGS_N (see Hydrology module)

SURFACE_WAT (see Hydrology module)

SALT_SURF_WT (see Salt/Tracer module)

TP_SF_WT (see Phosphorus module)

Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
HEADi, HEADj	attribute	m	hydraulic head in cell `i`, and in cell `j`
deltaHEAD	attribute	m	difference between hydraulic heads in cell `i`, and in cell `j`
a_deltaHEAD	attribute	m	absolute value of difference between hydraulic heads in cell `i`, and in cell `j`
Flux	attribute	m	water fluxed between cell `i`, and cell `j`
m1	attribute	kg	mass of constituent 1 fluxed from donor cell
m3	attribute	kg	mass of constituent 3 fluxed from donor cell

Time series forcing data

none (v2.3 and higher will have dynamic stage input data for grid cells along domain border)

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
GP_DetentZ	global	m	detention depth in a grid cell, below which surface flows do not occur
GP_mannDepthPow	global	dimless	power used in manning's equation water depth
GP_mannHeadPow	global	dimless	power used in manning's equation head difference
GP_dispParm	global	dimless	calibration parameter, can be ~representative of Dispersion Number estimates; a value of 0 removes any dispersion adjustments (leaving only the numerical dispersion of model scale)
GP_dispLenRef	global	m	reference length for which numerical dispersion (of finite difference sol'n) approximates actual turbulent diffusion, or dispersion

dispParm_scaled = (1.0 - GP_dispLenRef/celWid) * GP_dispParm	global	dimless	aggregated dispersion parameter
hyd_iter	global	dimless	number of horizontal iterations per ***DT***
sfstep = DT/hyd_iter	local	day	time step for horizontal surface water solutions
sq_celWid = CELL_SIZE^0.25	local	m^0.5	square root of cell width
celWid = CELL_SIZE^0.5	local	m	cell width
step_Cell = sq_celWid * sfstep/CELL_SIZE	local	m^(-1.5) * day	aggregation of static parameters (to reduce number of calculations per ***sfstep***)
sec_per_day = 86400	local	sec	number of seconds in a day

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description

none

Intrinsic C or ELM functions

Min(x,y) => minimum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

pow(x,y) => x raised to the yth power (generally avoided if possible due to execution time of C library)

ABS(x) => absolute value of (x)

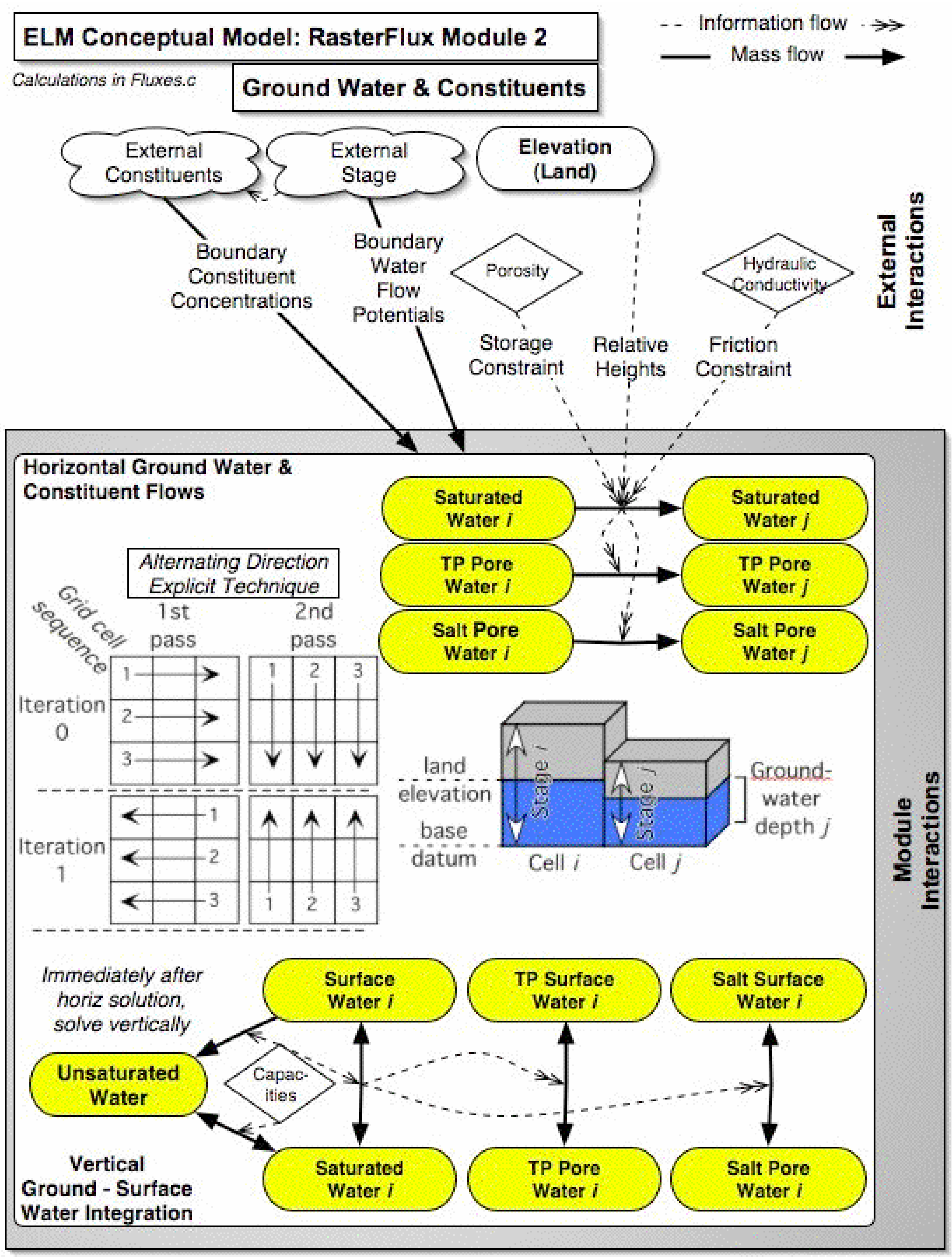
ramp(x) => negative (x) set =0, otherwise =(x) (precaution only for infinitesimally negative values - mass balance is evaluated (always output in budg_XYZ output files) at multiple spatial scales (several cell, whole-domain) and temporal scales, w/o losses: computational error in water storage height is on the order of +/- 10 microns accumulated over 20 years, maximum magnitude of (positive/negative) error is on the order of 1 micron accumulated over a 30-day period)

(x) != (y) => logical condition where (x) is not equal to (y)

if (x) equation => if (x) condition is true (==1), then execute `equation`

T(x,y) => single-integer array address of grid cell at location row x, column y (used in [cellLoc_])

5.7.4 Groundwater flow module



Overview: Groundwater Raster Flux Module

This Groundwater Raster Flux Module serves to update the ground water storage state variable due to horizontal subsurface flow among (raster) grid cells. The (vertical) Hydrology Module describes many of the dynamics associated with ELM hydrology, while this module description is specific to subsurface horizontal flow and its integration with surface water. These groundwater flows transport the constituents (phosphorus and salt/tracer) in addition to water in the landscape, and are highly dependent upon the aquifer transmissivity. Particularly in the central/southern extent of the eastern domain of the Everglades (esp. WCA-3B), the very high transmissivities make groundwater flows an important component of the overall hydrologic budget. Because the ELM domain encompasses only the "natural" wetlands of the greater Everglades, groundwater flows calculations use a very simple computational scheme, explicitly excluding highly transient aquifer dynamics associated with wellfields and related urban/agricultural features. While a number of vertical processes are solved in the (vertical) Hydrology Module, the explicit integration of surface water and groundwater (with associated constituents) is determined in this Groundwater Module immediately following the horizontal (surface and) groundwater flux calculations.

Groundwater Raster Flux Module Description

As with surface water flows, flow restrictions among grid cells are evaluated first. However, the only restriction for the groundwater system is that of the domain boundary. this determination of allowable flow between grid cells inside the model domain and grid cells outside the domain along the boundary are determined from a static input map layer: if subsurface groundwater flows are allowed, the stage and constituent concentration of an exterior boundary cell are determined. These stage data are daily values from another model such as the SFWMM.

The flow between two adjacent cells is determined from a simple application of the well-known Darcy's Law:

$$Q = K\frac{(h_1 - h_2)}{L} W \cdot D$$

where Q = flow (m^3 d^{-1} per m^2), K = hydraulic conductivity of aquifer (m d^{-1}), h_1 & h_2 are hydraulic heads measured along flow path (m), L = distance between heads (m), W = width of cross-sectional flow (m), and D = height of cross-sectional flow (m). Within an explicit, finite-difference framework, omitting any inertial or acceleration terms, the continuity equation is simply a two-dimensional flux driven by differences in slope of the hydraulic heads and the thickness of the saturated layer within an unconfined, vertically homogenous aquifer. Cell-cell head gradients are assumed to be small relative to the thickness of the aquifer down to the model base datum (which extends many meters below the land surface). The flux between a pair of grid cells in the rectangular array is described by the empirical Darcy's equation for saturated media, using an explicit numerical solution. The time step for horizontal groundwater flows is twice that of the horizontal surface water flows. Iterated in both the row-wise and the column-wise directions during each time step, the direction alternates (east-west and west-east, north-south and south-north) after each time step. This Alternating Direction Explicit solution minimizes the directional bias that is associated with a uni-directional solution. Constraints for stability and mass balance are imposed on the calculated flux during each time step, preventing head reversals or flows greater than the volume available in the donor grid cell. Mass of constituents (nutrients, salt/tracer) is passed along in a mass-balance calculation based upon the water volume flux between cells. Numerical dispersion due to the 1 km^2 grid scale and associated horizontal groundwater time step is assumed to approximate the (poorly known) actual physical dispersion associated with water flow velocities in this regional aquifer.

Groundwater Raster Flux Module Equations

Flux calculations

All equations shown are calculated within an Alternating Direction (each iteration) spatial loop across model grid rows, columns
[cellLoci] defines model grid address of cell "i"
[cellLocj] defines model grid address of cell "j"
Flux is positive/negative, from cell "i" to cell "j"

Pairs of grid cells are checked for (static) flow attributes in the spatial loop.
For a cell at [cellLoc_], the possible flow attributes are:
ON_MAP[cellLoc_]=0 => External to the model active domain
ON_MAP[cellLoc_]=1 => Allow (internal) flow in no direction (due to calculated levee-interaction geometry)
ON_MAP[cellLoc_]=2 => Allow (internal) flow to east<->west (due to calculated levee-interaction geometry)
ON_MAP[cellLoc_]=3 => Allow (internal) flow to south<->north (due to calculated levee-interaction geometry)
ON_MAP[cellLoc_]=4 => Allow (internal) flow in all directions (due to calculated (no) levee-interaction geometry)

If a single cell in a pair is external to the model domain (example, ON_MAP[cellLoci]=0),
allowance of internal<->external flow depends on an attribute of the other cell (i.e., [cellLocj]):
BCondFlow[cellLocj]=1 => Allow no flows external to model domain
BCondFlow[cellLocj]=3 => Allow surface water flows to/from external boundary cell

The function "Flux_GWcells" calculates and returns a cell-to-cell Flux in height units (m)
The case is shown for when both cell i and j are internal to the model domain, with flow allowed between the cells.
When one of the cells is external to the domain, and the pair of cells has been defined as allowing groundwater boundary flows, the stage of that external cell (cellLoci in this example) is estimated using:
*##   **HP_HYD_POROSITY**[cellLoci] = **HP_HYD_POROSITY**[cellLocj]*
and, when internal stage (tot_head_j) is greater than internal land surface elevation plus 20cm (SED_ELEV[cellLocj] + 0.20), estimates are:
SAT_WATER[cellLoci] = SAT_WATER[cellLocj]
SURFACE_WAT[cellLoci] = Max(SURFACE_WAT[cellLocj] - 0.3, 0.0)
or, when internal stage (tot_head_j) is less than/equal to internal land surface elevation plus 20cm (i.e., SED_ELEV[cellLocj] + 0.20), estimates are:
SAT_WATER[cellLoci] = SAT_WATER[cellLocj]-0.05
SURFACE_WAT[cellLoci] = 0.0
Code exists, but is not executed in v2.2, to replace the estimated values with input stage data from another model (e.g., SFWMM).

RCCONDUCT = (HYD_RCCONDUCT[cellLoci] + HYD_RCCONDUCT[cellLocj])/2.0

tot_head_i = SURFACE_WAT[cellLoci] + SAT_WATER[cellLoci] / ***HP_HYD_POROSITY***[cellLoci]

tot_head_j = SURFACE_WAT[cellLocj] + SAT_WATER[cellLocj] / ***HP_HYD_POROSITY*** [cellLocj]

deltaHEAD = tot_head_i - tot_head_j

*## For positive head differences (if the deltaHEAD > **GP_DetentZ**), assign the donor and recipient cell location attributes*
cell_don = cellLoci, cell_rec = cellLocj, sign = 1

*## For negative head differences (if the deltaHEAD < - **GP_DetentZ**), assign the donor and recipient cell location attributes*
cell_don=cellLocj, cell_rec=cellLoci, sign = -1

Potential cell-cell horizontal flux eqn (Darcy's eqn simplified to slope across square cells).
This is the maximum (height of) water vol to flux under fully saturated conditions.
Flux = Min(Abs(deltaHEAD) * ***GP_calibGWat*** * RCCONDUCT * SAT_WATER[cell_don] / ***CELL_SIZE*** * ***gwstep*** , SAT_WATER[cell_don]);

The below is an iterative ("do while") routine that (1) integrates the surface, saturated, and unsaturated water, and (2) checks to ensure the heads do not reverse in a time step due to large fluxes.
If heads do reverse, the total Flux is decremented in an iterative manner until there is no reversal

The total potential flux is apportioned to (1) the horizontal component that fluxes to an adjacent cell and (2) the vertical component that remains in the donor cell after the horizontal outflow from a donor cell.
Thus, an unsaturated zone is created, or increased in size, with loss of saturated water from the donor cell; this lateral gravitational flow leaves behind the field capacity moisture in an unsat zone. (If donor-cell surface water is present, it potentially will replace the unsaturated soil capacity within the same time step in this routine).

fluxTOunsat_don = Flux / ***HP_HYD_POROSITY*** [cell_don] * (***HP_HYD_POROSITY*** [cell_don] – ***HP_HYD_SPEC_YIELD***[cell_don])

fluxHoriz = Flux - fluxTOunsat_don

*## Donor cell, new **post-flux** capacities*
UnsatZ_don = SED_ELEV[cell_don] - (SAT_WATER[cell_don]- fluxHoriz) / ***HP_HYD_POROSITY*** [cell_don]

UnsatCap_don = UnsatZ_don * ***HP_HYD_POROSITY*** [cell_don]

UnsatPot_don = UnsatCap_don - (UNSAT_WATER[cell_don]+fluxTOunsat_don)

Donor cell, determining the pathway of flow (to sat vs. unsat) of surface water depending on depth of an unsat zone relative to the surface water. With a relatively deep unsat zone, this downflow tends to zero (infiltration occurs within the vertical hydrology module of UnitMod.c)
Sat_vs_unsat = 1/Exp(100.0*Max((SURFACE_WAT[cell_don]-UnsatZ_don),0.0))

Donor cell, sf-unsat-sat fluxes
Surface water downflow is assumed to be as fast as horizontal groundwater outflows.
In presence of surface water in the donor cell (only), the surface-to-saturated flow is determined.

sfTOsat_don = ((1.0-Sat_vs_unsat)*UnsatPot_don>SURFACE_WAT[cell_don]) ? (SURFACE_WAT[cell_don]) : ((1.0-Sat_vs_unsat)*UnsatPot_don)

With downflow of surface water into an unsat zone, the proportion of that height that is made into saturated storage is allocated to the sat storage variable
If surface volume downflow is larger than the unsaturated capacity, i.e., (sfTOsat_don >= UnsatPot_don)

sfTOsat_don = UnsatPot_don

unsatTOsat_don = UNSAT_WATER[cell_don]

Otherwise, allocate to saturated storage whatever proportion of unsat zone that is now saturated by sfwat downflow

unsatTOsat_don = (UnsatZ_don > 0.0) ? ((sfTOsat_don/ ***HP_HYD_POROSITY*** [cell_don]) / UnsatZ_don * UNSAT_WATER[cell_don]) : (0.0)

H_pot_don = (SAT_WATER[cell_don] - fluxTOunsat_don - fluxHoriz + sfTOsat_don + unsatTOsat_don) / ***HP_HYD_POROSITY*** [cell_don] +(SURFACE_WAT[cell_don] - sfTOsat_don)

*## Recipient cell, **pre-flux** capacities*

UnsatZ_rec = SED_ELEV[cell_rec] - SAT_WATER[cell_rec] / ***HP_HYD_POROSITY*** [cell_rec]

UnsatCap_rec = UnsatZ_rec * ***HP_HYD_POROSITY*** [cell_rec]

UnsatPot_rec = UnsatCap_rec - UNSAT_WATER[cell_rec]

Recipient cell, sf-unsat-sat fluxes

horizTOsat_rec = fluxHoriz

satTOsf_rec = Max(fluxHoriz - UnsatPot_rec, 0.0)

Recipient cell, incorporation of unsat moisture into sat storage with rising water table due to horiz inflow

unsatTOsat_rec = (UnsatZ_rec > 0.0) ? (((horizTOsat_rec-satTOsf_rec)/ ***HP_HYD_POROSITY*** [cell_rec]) / UnsatZ_rec * UNSAT_WATER[cell_rec]) : (0.0)

H_pot_rec = (SAT_WATER[cell_rec] + horizTOsat_rec + unsatTOsat_rec - satTOsf_rec) / ***HP_HYD_POROSITY*** [cell_rec] + (SURFACE_WAT[cell_rec] + satTOsf_rec) ;

Check for a head reversal - if a head reversal is > ***MinCheck****, reduce the potential Flux by 10%, and cycle through above donor-recipient calculations until an equilibrium is achieved*

Update the water state variables

SURFACE_WAT[cell_don] += (-sfTOsat_don);

UNSAT_WATER[cell_don] += (fluxTOunsat_don - unsatTOsat_don) ;

SAT_WATER[cell_don] += (sfTOsat_don + unsatTOsat_don - fluxTOunsat_don - fluxHoriz);

```
SURFACE_WAT[cell_rec]  += ( satTOsf_rec);

UNSAT_WATER[cell_rec]  += (-unsatTOsat_rec);

SAT_WATER[cell_rec] += (horizTOsat_rec + unsatTOsat_rec - satTOsf_rec); /*
    (horizTOsat_rec + satTOsf_rec) = fluxHoriz */
```

The constituent state variables are updated in a set of mass balance calculations using the mass in the donor cell storage variables and the water flux among the variables

External variables used

SED_ELEV (see Soils module)

DOM_MAXDEPTH (see Soils module)

SURFACE_WAT (see Hydrology module)

UNSAT_WATER (see Hydrology module)

SAT_WATER (see Hydrology module)

SALT_SURF_WT (see Salt/Tracer module)

TP_SF_WT (see Phosphorus module)

SALT_SED_WT (see Salt/Tracer module)

TP_SED_WT (see Phosphorus module)

Module Variable and Parameter Definitions

Module variables

Variable Name	Type	Units	Description
Flux	rateActual	m/d	potential/actual horizontal flux of groundwater between grid cells
fluxTOunsat_don	ratePotential	m/d	donor cell, field capacity volume (height) remaining in unsaturated zone associated with a horizontal flux
fluxHoriz	ratePotential	m/d	the actual water volume (height) that may flux horizontally (leaving field capacity in donor cell)
Sat_vs_unsat	controlFunction	dimless	same control function (0,1) used in Hydrologic Module to determine relative pathway of flow from surface storage (into saturated vs. unsaturated)
RCCONDUCT	attribute	m/d	mean hydraulic conductivity of the donor and recipient cells
UnsatZ_don	attribute	m	donor cell, new unsat zone depth after calculated groundwater flow
UnsatZ_rec	attribute	m	recipient cell, old unsat zone depth before calculated groundwater flow
UnsatCap_don	attribute	m	donor cell, maximum pore space capacity in the depth of new unsaturated zone
UnsatCap_rec	attribute	m	recipient cell, maximum pore space capacity in the depth of old unsaturated zone
UnsatPot_don	attribu	m	donor cell, (height of) the volume of

	te		pore space (soil "removed") that is unoccupied in the unsat zone
UnsatPot_rec	attribute	m	recipient cell, (height of) the volume of pore space (soil "removed") that is unoccupied in the unsat zone
sfTOsat_don	ratePotential	m/d	donor cell, surface to saturated flow
unsatTOsat_don	ratePotential	m/d	donor cell, unsaturated to saturated flow
unsatTOsat_rec	ratePotential	m/d	recipient cell, unsaturated to saturated flow
H_pot_don	attribute	m	donor cell, potential new head
H_pot_rec	attribute	m	recipient cell, potential new head
horizTOsat_rec	ratePotential	m/d	recipient cell, horizontal inflow to soil into saturated storage (== fluxHoriz)
satTOsf_rec	ratePotential	m/d	recipient cell, upflow to surface beyond soil capacity

Time series forcing data

none (v2.3 and higher will have dynamic stage input data for grid cells along domain border)

Static global parameters (all grid-cells)

Parameter Name	Type	Units	Description
DT	global	day	Time step for vertical solutions
CELL_SIZE	global	m^2	surface area of a model grid cell
DetentZ	global	m	detention depth in a grid cell, below which surface flows do not occur
MinCheck	global	dimless	small threshold number, for relative error-checking (not a multiplier etc)
GP_calibGWat	global	dimless	calibration parameter, multiply groundwater cell-cell flow calculation
hyd_iter	global	dimless	number of horizontal iterations per ***DT***
gwstep = DT/hyd_iter/2	local	day	time step for horizontal groundwater solutions

Static habitat-specific parameters (linked to HAB value of grid-cell)

Parameter Name	Type	Units	Description
HP_HYD_POROSITY	hab-spec	dimless	Porosity of the aquifer, average from the sediment to base datum. Field capacity = porosity - specific yield; ensure that alterations to porosity and specific yield are consistent in your parameterization. Must be non-

			zero.
HP_HYD_SPEC_YIELD	hab-spec	dimless	Proportion of total sediment/soil volume, for a given soil type, that represents water able to be drained by gravity. Field capacity = porosity - specific yield; ensure that alterations to porosity and specific yield are consistent in your parameterization.

Static spatially-distributed parameters

Parameter Name	Type	Units	Description
HYD_RCCONDUCT	distributed	m/d	HYDraulic CONDUCTvity Rate Constant of surficial aquifer

Intrinsic C or ELM functions

Min(x,y) => minimum of variable x or y

(x) ? (y) : (z) => if (x is true, or 1), then (return value y), else (return value z)

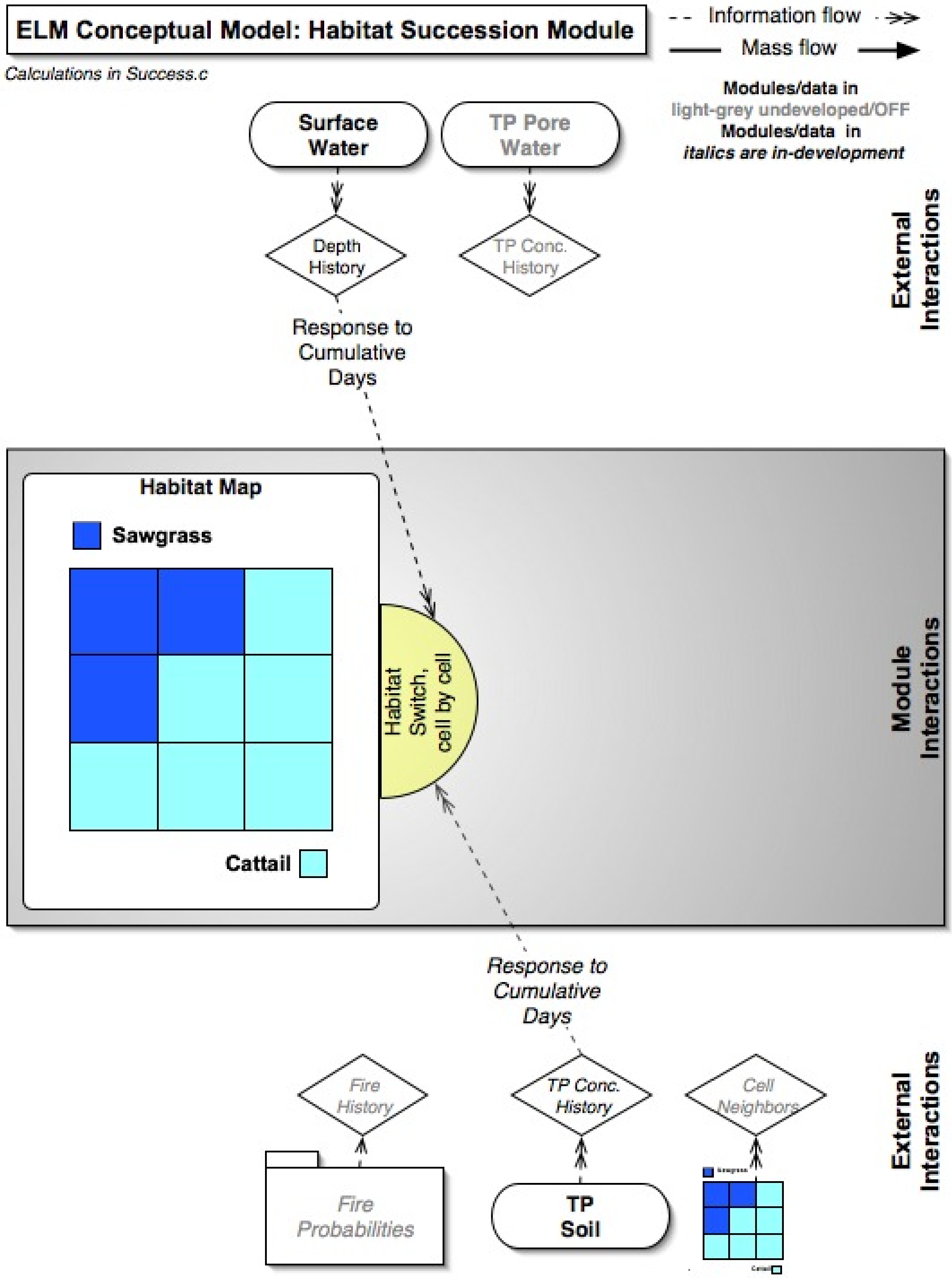
ABS(x) = Abs(x) => absolute value of (x)

(x) != (y) => logical condition where (x) is not equal to (y)

if (x) equation => if (x) condition is true (==1), then execute `equation`

T(x,y) => single-integer array address of grid cell at location row x, column y (used in [cellLoc_])

5.8 Habitat succession module



Overview: Succession Module

The habitat succession module in ELM v2.5 is a simple switching algorithm that responds to cumulative history of surface water and soil phosphorus. The design and performance was described in an earlier version (ELM v1.0) of a subregional application (Fitz and Sklar 1999).

Succession Module Description

Habitat succession was simulated by simple switching algorithm based on the cumulative impacts of both soil phosphorus and water depth. For each cell we evaluated the number of weeks that contained conditions favorable for each targeted habitat type, switching to the new habitat type when conditions merited. Each model cell was evaluated on a daily basis to determine if a) the soil phosphorus concentration was within the range defined by the habitat-specific parameters HP_PhosLo and HP_PhosHi and b) the ponded surface water depth was within the range defined by HP_SfDepthLo and HP_SfDepthHi. If a cell met either criteria for a targeted habitat, a counter was incremented for that habitat type, regardless of the cell's current habitat type designation. When counters for phosphorus and water depth conditions in a cell exceeded the criteria for the elapsed number of weeks defined by HP_PhosInt and HP_SfDepthInt, respectively, for a different habitat, the cell's habitat type classification switched to the new type and counters were set to 0. For this version, we considered the switching among three habitat types: sawgrass, cattail, and a mixture of sawgrass and cattail.

5.9 Literature cited

Bowie, G. L., W. B. Mills, D. B. Porcella, C. L. Campbell, J. R. Pagenkopf, G. L. Rupp, K. M. Johnson, P. W. H. Chan, S. A. Gherini, and C. E. Chamberlin. 1985. Rates, constants, and kinetics formulations in surface water quality modeling (2nd edition). EPA/600/3-85/040, U.S. Environmental Protection Agency, Office of Research and Development, Athens, GA, Athens, GA.

Carter, V., J. T. Reel, N. B. Rybicki, H. A. Ruhl, P. T. Gammon, and J. Lee. 1999. Vegetative Resistance to Flow in South Florida: Summary of Vegetation Sampling at Sites NESRS3 and P33, Shark River Slough, November, 1996. OFR-99-218, USGS.

Fitz, H. C., E. B. DeBellevue, R. Costanza, R. Boumans, T. Maxwell, L. Wainger, and F. H. Sklar. 1996. Development of a general ecosystem model for a range of scales and ecosystems. Ecological Modelling **88**:263-295.

Fitz, H. C., and F. H. Sklar. 1999. Ecosystem analysis of phosphorus impacts and altered hydrology in the Everglades: a landscape modeling approach. Pages 585-620 *in* K. R. Reddy, G. A. O'Connor, and C. L. Schelske, editors. Phosphorus Biogeochemistry in Subtropical Ecosystems. Lewis Publishers, Boca Raton, FL.

Jenter, H., and R. W. Schaffranek. 1996. Vegetation Affects Water Movement in the Florida Everglades. FS-147-96, USGS.

Jorgensen, S. E. 1976. A eutrophication model for a lake. Ecological Modelling **2**:147-165.

Lee, J., and V. Carter. 1999. Field Measurement of Flow Resistance in the Florida Everglades. USGS, http://sofia.usgs.gov/projects/vege_resist/vegeabsfrsf2.html.

Lee, J., and V. Carter. 2002. Vegetative resistance to flow in the Florida Everglades. USGS, http://sofia.usgs.gov/projects/vege_resist/vegeab2.html.

McCormick, P. V., P. S. Rawlick, K. Lurding, E. P. Smith, and F. H. Sklar. 1996. Periphyton-water quality relationships along a nutrient gradient in the Florida Everglades. Journal of the North American Benthological Society **15**:433-449.

Nalluri, C., and N. D. Judy. 1989. Factors affecting roughness in coefficient in vegetated channels. Pages 589 *in* B. C. Yen, editor. Channel flow and catchment runoff: centennial of Manning's formula and kuichling's rational formula, University of Virginia.

Newman, S., P. V. McCormick, and J. G. Backus. 2003. Phosphatase activity as an early warning indicator of wetland eutrophication: problems and prospects. Journal of Applied Phycology:45-59.

Nikolov, N. T., and K. F. Zeller. 1992. A solar radiation algorithm for ecosystem dynamics models. Ecological Modelling **61**:149-168.

Petryk, S., A. M. Asce, and B. III. 1975. Analysis of flow through vegetation. Journal of Hydraulics division **HY7**:871-883.

Richardson, C. J., and P. Vaithiyanathan. 1995. Phosphorus sorption characteristics of Everglades soils along a eutrophication gradient. Soil Science Society of America Journal **59**:1782-1788.

Scinto, L. J., and K. R. Reddy. submitted. Phosphorus uptake and partitoning by periphyton in a sub-tropical freshwater wetland. Limnology and Oceanography.

Steele, J. H. 1965. Notes on some theoretical problems in production ecology. Pages 393-398 *in* C. R. Goldman, editor. Primary production in aquatic environments. University of California Press, Berkeley, CA.

Steward, K. K., and W. H. Ornes. 1975. The autecology of sawgrass in the Florida everglades. Ecology **56**:162-171.

Wool, T. A., R. B. Ambrose, J. L. Martin, and E. A. Comer. in press. Water Quality Analysis Simulation Program (WASP) Version 6.0 Draft: User's Manual. US Environmental Protection Agency - Region 4, Atlanta, GA.