VELMA Version 2.0

User Manual and Technical Documentation

September 17, 2014

U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Western Ecology Division
Corvallis, Oregon 97333

Robert B. McKane (mckane.bob@epa.gov)[1]
Allen Brookes (brookes.allen@epa.gov)[1]
Kevin Djang (djang.kevin@epa.gov)[2]
Marc Stieglitz (marc.stieglitz@ce.gatech.edu)[3]
Alex G. Abdelnour (abdelnouralex@gmail.com)[3,5]
Feifei Pan (feifei.Pan@unt.edu)[3,4]
Jonathan J. Halama (halama.jonathan@epa.gov)[1]
Paul B. Pettus (pettus.paul@epa.gov)[1]
Donald L. Phillips (phillips.donald@epa.gov)[1]

[1]U.S. Environmental Protection Agency, Western Ecology Division, Corvallis, OR 97333

[2]CSC, Corvallis, OR 97333

[3]School of Civil & Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332

[4]Current address: Department of Geography, University of North Texas, Denton, TX 76203

[5]Current address: McKinsey & Company, 133 Peachtree Street, N.E., Atlanta, GA 30303

ACKNOWLEDGEMENTS

The information in this document has been funded in part by the U.S. Environmental Protection Agency. It has been subjected to the agency's peer and administrative review, and it has been approved for publication as an EPA document. This research was additionally supported in part by NSF grants 0439620, 0436118, and 0922100. We thank our many colleagues who have contributed data and advice during the course of developing VELMA and its applications: Sherri Johnson, Barbara Bond, Mark Harmon, Fred Swanson, Julia Jones, Suzanne Remillard, Theresa Valentine, and Don Henshaw with the H.J. Andrews Long Term Ecological Research (LTER) program; John Blair, John Briggs, and Loretta Johnson with the Konza Prairie LTER program; Don Weller and Tom Jordan, Smithsonian Environmental Research Center; Bill Peterjohn, West Virginia University; Larry Band, University of North Carolina; Irena Creed, University of Western Ontario; George Ice, Oregon State University; Ed Rastetter and Bonnie Kwiatkowski, Ecosystems Center, Marine Biological Laboratory; Brenda Groskinsky, Don Phillips, Jana Compton, Renee Brooks, and many others at the U.S. Environmental Protection Agency.

DISCLAIMER

Neither the United States Environmental Protection Agency (USEPA) nor the Georgia Institute of Technology (GT) makes any warranty or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, product, or process described herein. Reference to any special commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply endorsement, recommendation, or favoring by the USEPA or GT. The views and opinions of the authors do not necessarily reflect those of the USEPA or GIT.

Table of Contents

I. Introduction

VELMA – Visualizing Ecosystems for Land Management Assessments – is a spatially distributed, eco-hydrological model that links a land surface hydrology model with a terrestrial biogeochemistry model for simulating the integrated responses of vegetation, soil, and water resources to interacting stressors. For example, VELMA can simulate how changes in climate and land use interact to affect soil water storage, surface and subsurface runoff, vertical drainage, evapotranspiration, vegetation and soil carbon and nitrogen dynamics, and transport of nitrate, ammonium, and dissolved organic carbon and nitrogen to water bodies. VELMA differs from other existing eco-hydrology models in its simplicity, flexibility, and theoretical foundation. The model has a user-friendly Graphics User Interface (GUI) for easy input of model parameter values. In addition, advanced visualization of simulation results can enhance understanding of results and underlying concepts. VELMA's visualization and interactivity features are packaged in an open-source, open-platform programming environment (Java / Eclipse). The development team for VELMA version 2.0 includes Dr. Bob McKane and coworkers at the U.S. Environmental Protection Agency's Western Ecology Division, Dr. Marc Stieglitz and coworkers at the Georgia Institute of Technology, and Dr. Feifei Pan at the University of North Texas.

1.1. Application of VELMA to Various Ecosystems

VELMA's hydrological and biogeochemical submodels have been verified for simulating the effects of changes in climate and land use on streamflow, stream chemistry, and ecosystem carbon and nitrogen dynamics (Abdelnour et al. 2011, 2013; http://onlinelibrary.wiley.com/doi/10.1029/2010WR010165/full; http://onlinelibrary.wiley.com/doi/10.1029/2012WR012994/full). To date, we have calibrated VELMA for a wide range of major ecosystem types across North America, focusing primarily on data-rich sites in the National Science Foundation's Long Term Ecological Research (LTER) network (http://www.lternet.edu/lter-sites). These ecosystem types include temperate forest LTER sites in Oregon, New Hampshire, and North Carolina; the Konza Prairie LTER in Kansas; agricultural watersheds in the Chesapeake Bay and Willamette River Basin; and the arctic tundra LTER in Alaska.

1.2. Decision Support

We developed VELMA to help support two recent sustainability initiatives by the EPA Office of Research and Development: the Safe and Sustainable Waters Research Program (SSWR) and the Sustainable and Healthy Communities Research Program (SHC). Our goal is to provide comprehensive decision support tools that can help communities, tribes, land managers and policy makers address present needs without compromising the ability of society and the environment to meet the economic, social and environmental needs of future generations. **Key decision support goals are to (1) assess the effectiveness of natural and engineered green infrastructure for protecting water quality of streams and estuaries, and (2) quantify the ecosystem goods and services that ecosystems provide for humans.**

VELMA was recently redesigned (version 2.0, described herein) to better address both these goals. Green infrastructure (GI) involves the establishment of riparian buffers (streamside vegetation), cover crops, constructed wetlands, and other measures to intercept, store and transform nutrients, toxics and other contaminants that might otherwise reach surface and ground waters. GI enhancements have also strengthened model capabilities for quantifying how alternative land use and policy scenarios affect tradeoffs among important ecosystem services – that is, the capacity of an ecosystem to provide clean water, flood control, food and fiber, climate (greenhouse gas) regulation, fish and wildlife habitat, among others (Millennium Ecosystem Assessment 2005). Model development has been guided by the principle of parsimony to enable VELMA to efficiently address multiple spatial and temporal scales – plots to basins, days to centuries.

While VELMA has already proven useful for quantifying how such ecosystem services interact and respond in concert to environmental changes (http://eco.confex.com/eco/2012/webprogram/Paper37040.html; http://eco.confex.com/eco/2010/techprogram/P25975.HTM), it is important to also quantify the economic and social impacts associated with such changes. Therefore, we are collaborating with Oregon State University to link VELMA with ENVISION, a well-established decision support tool that integrates landscape GIS layers, ecological models, economic valuation models, and user-defined stressor scenarios (http://envision.bioe.orst.edu/). The recently completed Envision-VELMA linkage is described in our SHC 2.1.4.2 product for September 2014.

To date, we have prototyped ENVISION for the 30,000 km^2 Willamette River Basin in Oregon to examine how alternative scenarios (2010 – 2060) of land use and human population growth affect "bundled" ecosystem services. This work examines the capacity of the landscape to support projected increases in human populations under alternative growth plans (smart growth, unmanaged growth, and status quo) and consequent trade-offs in provisioning of agricultural and forest products, clean supplies of water, carbon sequestration, and habitat for wildlife populations (Bolte et al. in review; Bolte et al. 2011, http://www.thesolutionsjournal.com/node/1019).

1.3. Products and Impacts

The major product of this research will be a set of broadly applicable decision support tools that enable communities, client offices and other stakeholders to (1) assess the effectiveness of green infrastructure options for protecting water quality, (2) quantify tradeoffs among ecosystem goods and services associated with alternative land use decision scenarios, and (3) generate community sustainability indicators and their trajectories to help communities balance environmental, economic and social criteria over timescales relevant to immediate needs and long-term (decades to centuries) planning goals.

Using a participatory planning and outreach approach that integrates researchers, stakeholders and decision makers, we will address several questions pertinent to community sustainability:

- Can methodologies be developed to quantify and value ecosystem services, so that this natural capital can be better accounted for in decisions that affect the supply of the goods and services upon which human well-being depends?
- What are the uncertainties associated with various decision options? Is there an optimal “decision path” for restoring the natural capital needed to sustainably support communities dependent on resource-based economies, such as the agricultural, forest, and fishing industries?
- Can those factors that have the greatest potential to improve future trajectories of ecosystem services and human well-being be identified? For example, what green and grey infrastructure improvements, carbon and nitrogen management practices, and growth and development policies can most effectively be managed to attain a sustainable and desirable future?

The linkage of VELMA and ENVISION is intended to provide stakeholders with a user-friendly, visual interface for exploring the consequences of alternative climate and land use scenarios on ecosystem service tradeoffs. Outputs will be computer-generated visualizations of predicted changes in multiple ecosystem services, both in biophysical and economic terms. Our overall goal is to provide a framework for integrated assessments that identify policy and management strategies for entire ecosystems and the bundled services they provide, rather than piecemeal assessments of individual services.

1.4. References

Abdelnour, A., M. Stieglitz, F. Pan, and R. McKane (2011) Catchment hydrological responses to forest harvest amount and spatial pattern, *Water Resources Research*, 47, W09521, doi:10.1029/2010WR010165.

Abdelnour, A., R. McKane, M. Stieglitz, F. Pan, and Y. Cheng (2013) Effects of harvest on carbon and nitrogen dynamics in a Pacific Northwest forest catchment, *Water Resources Research*, 49, doi:10.1029/2012WR012994.

Abdelnour, A., R. McKane, M. Stieglitz and F. Pan. Catchment biogeochemical responses to forest harvest amount and spatial pattern. Submitted to *Water Resources Research.*

Bolte, J., R. McKane, D. Phillips, N. Schumaker, D. White, A. Brookes, C. Burdick and D. Olszyk (in review), An extensible decision support system for evaluating ecosystem services under alternative future scenarios – a Willamette River Basin case study. EPA publication number ORD-002136

Bolte, J., R. McKane, D. Phillips, N. Schumaker, D. White, A. Brookes, and D. Olszyk (2011) In Oregon, the EPA calculates nature’s worth now and in the future. *Solutions* 2(6): 35-41.

McKane, R., A. Abdelnour, A. Brookes, C. Burdick, K. Djang, T.E. Jordan, B. Kwiatkowski, F. Pan, W.T. Peterjohn, M. Stieglitz and D.E. Weller (2012) Identifying green infrastructure BMPs for reducing nitrogen export to a Chesapeake Bay agricultural stream: model synthesis and extension of experimental data. *The Ecological Society of America 97th Annual Meeting*, Portland, OR, August 2012.

McKane, R., M. Stieglitz, A. Abdelnour, F. Pan, B. Bond, S. Johnson (2010) An integrated eco-hydrologic modeling framework for assessing the effects of interacting stressors on multiple ecosystem services. *The Ecological Society of America 95th Annual Meeting*, Pittsburgh, PA, August 2010.

MEA, Millennium Ecosystem Assessment (2005) Ecosystems and human well-being: synthesis. Island, Washington, DC.

II. Changes in VELMA Version 2.0

The original version of VELMA (henceforth referred to as version 1.0 or v1.0) has been previously described (Abdelnour et al. 2011, 2013). VELMA version 2.0 (v2.0), described herein, includes a number of important changes that are designed to facilitate green infrastructure and ecosystem service applications of interest to communities, tribes, land managers and policy makers. In summary, this required:

- Enhancements to the hydrological and biogeochemical submodels to better simulate effects of green infrastructure management practices on the fate and transport of water, nutrients and toxics across multiple spatial and temporal scales – plots to basins, days to centuries (details below).
- A new graphical-user-interface (GUI) to assist novice and experienced model users in scenario development, model calibration, and visualization and interpretation of results.
- Rewriting the program code in Java / Eclipse to facilitate debugging and open source (community) model development. The version 1.0 Processing code is no longer supported.
- A new user manual (this document) that provides a step-by-step guide to setting up and applying VELMA v2.0.

Details follow on changes to the hydrological and biogeochemical submodels in VELMA v2.0:

1) Hydrological submodel:
    a. Evapotranspiration (ET) submodel revised to include
        i. effect of leaf biomass (surrogate for leaf area); replaces submodel for simulating effect of stand age on ET (Abdelnour et al. 2011)
        ii. effect of stand age on ET in forest systems
2) Biogeochemical submodel:
    a. Single land cover simulator revised to include multiple land cover types within watersheds
    b. Plant Soil Model (PSM) approach for simulating plant biomass in aggregate (Stieglitz et al. 2006; Abdelnour et al. 2013) replaced by a Leaf-Stem-Root (LSR) submodel that simulates carbon (C) and nitrogen (N) dynamics for leaves, aboveground stems belowground stems, and fine roots (<2 mm diameter).
    c. Decomposition submodel replaced by Potter et al. (1993) submodel and further adapted to work with LSR submodel.
    d. New plant biomass mortality submodel based on changes in NPP/mortality ratio during succession
    e. Plant nitrogen uptake submodel modified to include fine root biomass dynamics
    f. Nitrification and denitrification submodels revised based on unpublished corrections to Del Grosso et al. (2001)

g. New nitrogen fixation submodel

h. New atmospheric nitrogen deposition submodel that accounts for time lag between dry deposition on leaf surfaces and subsequent precipitation-driven transfer to plant-available soil nitrogen pools

i. Improved disturbance submodel and GUI for scheduling and simulating effects of fertilization, harvest, grazing and fire on ecohydrological processes

3) New spatial climate simulator that includes effects of elevation, slope and aspect on pixel-specific temperature and precipitation

4) Supporting submodels

a. DEM processor (PDEM) rewritten in Java with new GUI

b. New Python software (VELMA Raster Rally GIS toolbox) for preparing spatial data files (DEM, land cover, plant biomass, nutrient pools and soil type maps) for VELMA initialization

References

Abdelnour, A., Stieglitz, M., Pan, F., & McKane, R. 2011. Catchment hydrological responses to forest harvest amount and spatial pattern. *Water Resources Research* 47(9).

Abdelnour, A., B McKane, R., Stieglitz, M., Pan, F., & Cheng, Y. (2013). Effects of harvest on carbon and nitrogen dynamics in a Pacific Northwest forest catchment. Water Resources Research 49:1292-1313.

McKane, R.B., A. Abdelnour, A. F. Brookes, C. A. Burdick, K. Djang, T. Jordan, B. Kwiatkowski, F. Pan, W. Peterjohn, M. Stieglitz, and D. Weller. 2012. Identifying green infrastructure BMPs for reducing nitrogen export to a Chesapeake Bay agricultural stream: model synthesis and extension of experimental data. Presented at Ecological Society of America, Portland, OR, August 05 - 10, 2012.

McKane, R.B. 2014. Using VELMA to Quantify and Visualize the Effectiveness of Green Infrastructure Options for Protecting Water Quality. Seminar slide presentation for EPA-ORD Green Infrastructure Seminar Series, July 30, 2014.

III. How to Use This Manual

This user manual and its supporting documents are written and organized for several types of VELMA user groups:

Group 1: User describes questions and goals, VELMA team does the rest.
Example: EPA client offices (Regional offices, Office of Water, etc.) requiring information on potential effects of a change in policy on water quality and other environmental endpoints. VELMA team functions as user types 3 and 4, below, to address client questions and goals.

Group 2: User assembles GIS data, creates scenarios, runs simulations, analyzes data; VELMA team calibrates hydrology & biogeochemistry submodels and supplies initial input files (parameter values and, if necessary, initial GIS data and climate drivers).
Example: federal and state land management agencies, watershed councils and other community groups with sufficient expertise.

Group 3: User works independently to assemble model input files, calibrate parameters, and analyze model output.
Example: academics and other professionals with expertise in hydrology, biogeochemistry and GIS methods.

Group 4: User works independently, as for (3), and modifies the program code and ecohydrological equations. Suggested code modifications can be submitted to the VELMA development team, who will review and may incorporate proposed modifications into updated model versions.
Example: academics and other professionals with expertise in hydrology, biogeochemistry, GIS methods, and computer programming and software design.

This user manual is a reasonably complete "how to" for operating VELMA v2.0. However, rather than getting immediately bogged down in details, we recommend that all users begin with a quick example and condensed version of the user manual.

Use these steps:

1. Open the "VELMA v2.0" subfolder located inside the "VELMA Model" folder
2. Open the PDF "VELMA 2.0 Quick Example". This condensed version (12 pages) of the complete User Manual (176 pages) will guide you through a quick example of how to start VELMA and run it for a pre-configured set of input files and parameters located in the path\folder named "VELMA Model\VELMA v2.0\BlueRiver_Example". This application is for a headwater forest catchment at the HJ Andrews Experimental Forest in western Oregon (Abdelnour et al. 2011, 2013). The Quick Example PDF will walk you through the remaining steps, below.
3. Confirm that you have either a Java 7 JRE installed and accessible on your machine.
4. Start the VELMA GUI.
5. Load the example Simulation Configuration File.
6. Set the Location of the Simulation Configuration's Input Data Files.
7. Set the Location Where the VELMA Simulator Will Writer Results Files.
8. Save Your Parameter Changes.
9. Click the Start Button to Start a Simulation Run.
10. Look in the Specified initialOutputDataLocationRoot Results Directory for Results.

These are the most basic steps for using VELMA. All user groups can benefit from the details included in the complete User Manual (this document), although persons in each group likely will have a different approach to using the manual.

For example, if you are a Group 2 VELMA user interested in investigating watershed responses to climate change, you could develop and apply climate scenarios of your own design by following instructions in section IV in this manual. You could then apply your scenarios to an existing VELMA application – for example, the "BlueRiver_Example" located in the "VELMA v20" folder with this User Manual. Our hope is that some Group 2 users will make the jump to Group 3 as they gain experience with VELMA and modeling methods in general.

If you are a Group 3 or Group 4 VELMA user interested in developing new simulator configurations (VELMA applications), you may want to step through all or most of the remainder of this manual. There are a fair number of steps, but we've tried to organize these in a logical way. In particular, "Chapter VI – Steps for Configuring All Parameters TOC Sections 0.0 – 25.0" is our attempt to break up the parameterization process into manageable pieces. Each section and its subsections are focused on parameters associated with a particular ecosystem component or process.

If you are a Group 4 "super-user" interested in developing new algorithms or other features for VELMA, please contact the authors of this user manual. We will be happy to provide instructions for accessing the VELMA program code through an open source (Subversion / SVN) site where you can download the current release, and later upload your proposed enhancements. Our intent is to make VELMA an open-source community model that is freely accessible and will improve over time.

Before beginning, we recommend that all users become familiar with two "box and arrow" figures describing VELMA's hydrological and biogeochemical pools and fluxes. The first figure is located in Appendix 6.2 section A3 from Abdelnour et al. (2013). It describes the original (VELMA v1.0) model structure and nicely captures the interaction of hydrological and biogeochemical processes in VELMA. The second is Figure 1 on page 43 under "Important Information for Sections 5.0 – 25.0 of the All Parameters TOC". It describes VELMA v2.0's Plant Soil Model (PSM) submodel, for which a Leaf-Stem-Root (LSR) submodel has replaced the single (aggregate) plant biomass pool in VELMA v1.0. Together, these figures will give users a good sense of how the model works, and how each section of the User Manual relates to the others.

IV. Required Input Data Files for VELMA Simulations

The primary input file for a given simulation run is an XML file of initialization parameters known as the *simulator configuration file*[1] (often abbreviated as the "simconfig" or "the XML" file). In addition to the simulator configuration file, the following data files must be complete and available for the VELMA Simulator to run:

FILE	TYPE	CONTAINS
Flat-Processed DEM	Spatial / Grid ASCII (.asc)	Elevation in meters
Cover Species ID Map	Spatial / Grid ASCII (.asc)	Cover Species ID integers
Cover Species Age Map	Spatial / Grid ASCII (.asc)	Cover Species age in years
Soil Parameters ID Map	Spatial / Grid ASCII (.asc)	Soil Parameterization ID integers
Precipitation Driver Data	Temporal / (.csv or .txt)	Precipitation per in mm
Air Temperature Driver Data	Temporal / (.csv or .txt)	Air Temperature in degrees C

If your simulation is using the optional spatially-explicit weather model, the following files must also be complete and available:

FILE	TYPE	CONTAINS
Head Index Map	Spatial / ESRI Grid ASCII (.asc)	Heat index values
Precipitation Coefficients	Tabular (.csv)	Coefficient values
Air Temperature Coefficients	Tabular (.csv)	Coefficient values

Finally, the following files are not required for the VELMA Simulator to run, but providing them for a simulation run is recommended. Their availability allows the simulator to provide more run-time information and some additional results data. (E.g. The VELMA Simulator can only compute a Nash-Sutcliffe coefficient value for a simulation run's runoff data if the Observed Runoff data file was provided as part of the input data).

FILE	TYPE	CONTAINS
Observed Runoff	Temporal (.csv or .txt)	Observed runoff in mm
Observed Stream Chemistry	Temporal Tabular (.csv)	Observed chemistry values

[1] "Simulator Configuration" and "Simulation Configuration" are both used as the term for this primary file. The former emphasizes the fact that the file's contents configure the simulator, while the latter emphasizes the (equally true) fact that the file's contents form a particular configuration of a given simulation scenario.

The Simulator Configuration file specifies the Names of all other input data files.

When you run a VELMA Simulation, the simulator engine is given the simulator configuration file. The first thing the simulator engine does is initialize itself, using the key-value pairs in the file. Each of the required or optional files mentioned above is present as a key-value pair in the simulator configuration file.

FILE	**CONFIGURATION ID KEY**
Flat-Processed DEM	`/calibration/VelmaInputs.properties/input_dem`
Cover Species ID Map	`/calibration/VelmaInputs.properties/coverSpeciesIndexMapFileName`
Cover Species Age Map	`/calibration/VelmaInputs.properties/coverAgeMapFileName`
Soil Parameters ID Map	`/calibration/VelmaInputs.properties/soilParametersIndexMapFileName`
Precipitation Driver Data	`/weather/SpatialWeatherModel/rainDriverDataFileName`
Air Temperature Driver Data	`/weather/SpatialWeatherModel/airTemperatureDriverDataFileName`
Head Index Map	`/weather/SpatialWeatherModel/heatIndexMapFileName`
Precipitation Coefficients	`/weather/SpatialWeatherModel/rainCoefficientDataFileName`
Air Temperature Coefficients	`/weather/SpatialWeatherModel/airTemperatureCoefficientsDataFileName`
Observed Runoff	`/calibration/VelmaInputs.properties/input_runoff`
Observed Stream Chemistry	`/calibration/VelmaInputs.properties/input_stream_chem`

ID keys for required input data files *must* have the name of an existing, valid and readable file[2] specified, while optional input data files ID keys may be left blank.

[2] Sometimes just the name and extension of the file will do, sometimes the fully-specified path must be included as part of the name, and sometimes the path is optional. This unfortunately-confusing state of affairs exists due the history of the Simulator's development. For the input data files discussed here, supply only the name and extension as an ID Key's value.

The Simulator Configuration file specifies the Location of all other input data files.

The simulation configuration specifies the location of a directory (a.k.a. "folder") and the VELMA simulator engine looks in that directory for the input data files specified by the appropriate ID Keys (as listed in the previous table). The simulation configuration uses two, separate ID Keys to specify one fully-qualified path + directory name:

Location Path	`/startups/VelmaStartups.properties/inputDataLocationRootName`
Directory Name	`/startups/VelmaStartups.properties/inputDataLocationDirName`

The fully-qualified location of input data files is **Location Path** + "/" + **Directory Name**.
For example, the following ID Key values:

ID Key	Value
`/startups/VelmaStartups.properties/inputDataLocationRootName`	`C:/MyVELMA_InputData`
`/startups/VelmaStartups.properties/inputDataLocationDirName`	`LittleMtn_Watershed`

… indicate that the VELMA simulator should look for input data files in this directory:
`C:\MyVELMA_InputData\LittleMtn_Watershed`

IMPORTANT NOTE
The example above uses Microsoft Window file system "back slash" separators, however you should ALWAYS use "forward slash" (a.k.a. Unix-style) separators when specifying path name values for the simulator configuration's ID Keys.

Given the fully-qualified input data location in the example above, suppose the LittleMtn_Watershed's DEM file name (for ID Key `/calibration/VelmaInputs.properties/input_dem`) was specified as:
`LittleMtn_30m_DEM.asc`
The VELMA simulator would attempt to open and read elevation data from the file:
`C:\MyVELMA_InputData\LittleMtn_Watershed\LittleMtn_30m_DEM.asc`

Details of Specific File Formats

Broadly speaking, the VELMA simulator needs two types of input or driver data: Spatial and Temporal. A spatial data value is something associated with a particular cell location. A temporal data value is something associated with a particular time step of a simulation run. Spatial data is nearly always input from files in ESRI Grid ASCII format (.asc), while temporal data is input from files with either comma-separated value format (.csv), or simple text files (.txt)

ESRI Grid ASCII (.asc)

A raster GIS file format specified by ESRI. See this Wikipedia article for an overview:
"ESRI grid"
http://en.wikipedia.org/wiki/ESRI_grid

Comma-Separated Value (.csv)

An informal, de-facto standard format that represents tabular data in a file as a sequence of lines (rows) of comma-separated values (columns). There is no formal standard for CSV data, but good overviews are provided by this Wikipedia article:
“Comma-separated values”
http://en.wikipedia.org/wiki/Comma-separated_value
And this RFC document:
“RFC 4180 – Common Format and MIME Type for Comma-Separated Values (CSV) Files”
http://tools.ietf.org/html/rfc4180

Simple Text Files (.txt)

These are plain-text files of the sort created by Microsoft’s Text Editor, or by saving a Microsoft Word document as “Plain Text”.

Details of Specific Input Data Files

The Flat-Processed DEM File

A spatial data file (.asc) containing elevation values (in meters) for every cell in the simulation area. The term “Flat-Processed” indicates that the contents of the file have been pre-processed by the JPDEM flat-processing utility. The DEM file is the “master” film for a simulation run: all other spatially-explicit data is assumed to have the same row, column, cell size and x, y offset values as the DEM file.

See “Appendix 7: Creating Flat-Processed DEM Data For The VELMA Simulator and Determining Outlet and Watershed Delineation” for details on the use of the JPDEM.jar program to flat process a DEM file.

The Cover Species ID Map File

A spatial data file (.asc) containing cover species ID numbers for every cell in the simulation area. The ID numbers must be integers, and must correspond to one or more of the simulator configuration’s `/cover/…/uniqueId` ID Key values.

The Cover Species Age Map File

A spatial data file (.asc) containing ages (in years) for every cell in the simulation area. The age of a given cell in the file represents that cell’s cover species’ age in years at the simulation start year (specified in the simulator configuration by the `/calibration/VelmaInputs.properties/syear` ID Key’s value).

The Soil Parameters ID Map File

A spatial data file (.asc) containing Soil Parameterization ID numbers for every cell in the simulation area. The ID numbers must be integers, and must correspond to one or more of the simulator configuration’s `/soil/…/uniqueId` ID Key values .

The Precipitation Driver Data File

A temporal data file containing a rain value in millimeters for each day of the simulation.
The data is formatted as one value per line in the file, and the file should have as many lines as there are days between the simulator configuration’s specified `forcing_start` and `forcing_end` values.
For example, if the simulation configuration has the following values specified:

ID Key	Value
`/calibration/VelmaInputs.properties/forcing_start`	`2000`
`/calibration/VelmaInputs.properties/forcing_end`	`2001`
`/weather/SpatialWeatherModel/rainDriverDataFileName`	`MyRainData.csv`

… the VELMA simulator will expect the file MyRainData.csv to contain exactly 731 lines of data (366 days + 365 days for years 2000 + 2001) one value per line. The first five lines of the file might look like this:

```
        29.6500000
       67.1500000
       14.4000000
       28.3250000
       33.4000000
       [ … ]
```

Notice that although MyRainData.csv is a .csv (comma-separated values) file, it has no header row, and no commas (because there is only one “column” of data). Notice also that there is leading whitespace in front of the data values; leading whitespace is not required, and is ignored.

Because there is no difference between a single-column, no-header-row, comma-separated values (.csv) file, and a simple plain text (.txt), the VELMA simulator will accept either a .csv or .txt file extension for the precipitation driver file, although the .csv extension is preferred.

The Air Temperature Driver Data File

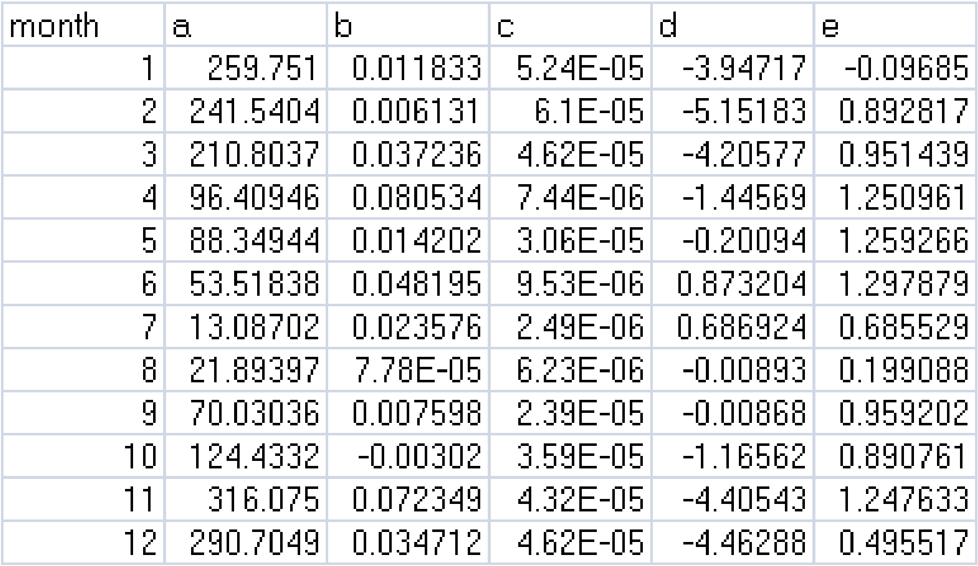
A temporal data file containing a value in centigrade for mean daily temperature (i.e., the average of daily Tmin and Tmax) for each day of the simulation. The data is formatted according to the same rules as the precipitation driver data file; one value per line, one line per possible simulation day.

The Heat Index Map File

A spatial data file (.asc) containing a heat index value for every cell in the simulation area. The heat index value is computed from elevation, slope and other factors.
The Heat Index Map File is only required when the spatially-explicit weather model is used. Otherwise, the file may be left unspecified in the simulator configuration.

The Precipitation Coefficients File

A comma-separated values (.csv) file containing a table of coefficients for the spatial weather model’s precipitation equation. The file must contain 13 rows of 6 comma-separated values (columns) each. The first row is a header row, and the remaining 12 rows provide values for the equation’s a, b, c, d and e coefficients for each month of the year.
Here is an example table of values:



month	a	b	c	d	e
1	259.751	0.011833	5.24E-05	-3.94717	-0.09685
2	241.5404	0.006131	6.1E-05	-5.15183	0.892817
3	210.8037	0.037236	4.62E-05	-4.20577	0.951439
4	96.40946	0.080534	7.44E-06	-1.44569	1.250961
5	88.34944	0.014202	3.06E-05	-0.20094	1.259266
6	53.51838	0.048195	9.53E-06	0.873204	1.297879
7	13.08702	0.023576	2.49E-06	0.686924	0.685529
8	21.89397	7.78E-05	6.23E-06	-0.00893	0.199088
9	70.03036	0.007598	2.39E-05	-0.00868	0.959202
10	124.4332	-0.00302	3.59E-05	-1.16562	0.890761
11	316.075	0.072349	4.32E-05	-4.40543	1.247633
12	290.7049	0.034712	4.62E-05	-4.46288	0.495517

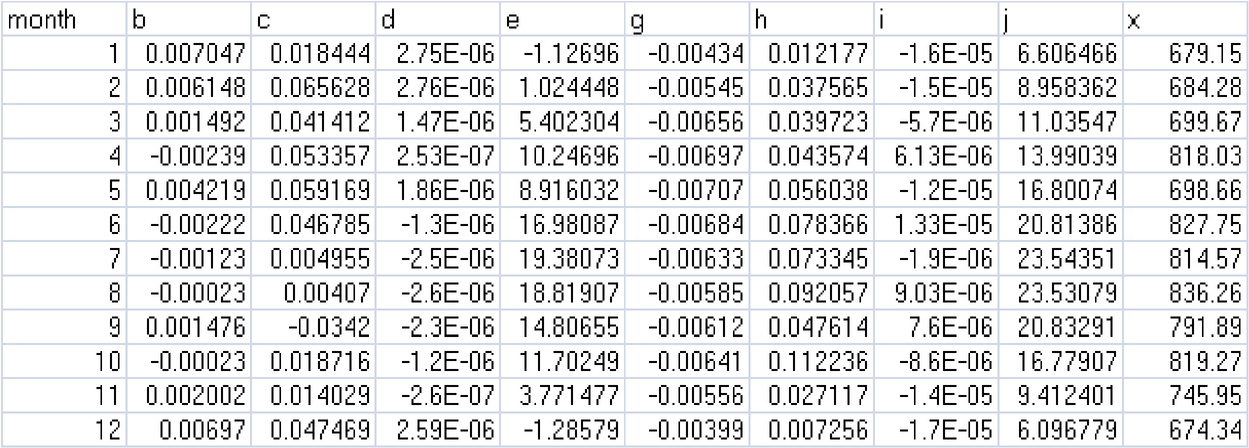
and here are the lines from the corresponding .csv data file:

```
month,a,b,c,d,e
1,259.751,0.011833,5.24E-05,-3.94717,-0.09685
2,241.5404,0.006131,6.10E-05,-5.15183,0.892817
3,210.8037,0.037236,4.62E-05,-4.20577,0.951439
4,96.40946,0.080534,7.44E-06,-1.44569,1.250961
5,88.34944,0.014202,3.06E-05,-0.20094,1.259266
6,53.51838,0.048195,9.53E-06,0.873204,1.297879
7,13.08702,0.023576,2.49E-06,0.686924,0.685529
8,21.89397,7.78E-05,6.23E-06,-0.00893,0.199088
9,70.03036,0.007598,2.39E-05,-0.00868,0.959202
10,124.4332,-0.00302,3.59E-05,-1.16562,0.890761
11,316.075,0.072349,4.32E-05,-4.40543,1.247633
12,290.7049,0.034712,4.62E-05,-4.46288,0.495517
```

The Precipitation Coefficients File is only required when the spatially-explicit weather model is used. Otherwise, the file may be left unspecified in the simulator configuration.

The Air Temperature Coefficients File

A comma-separated values (.csv) file containing a table of coefficients for the spatial weather model's air temperature equation. The file must contain 13 rows of 10 comma-separated values (columns) each. The first row is a header row, and the remaining 12 rows provide values for the equation's a,..,j and x coefficients for each month of the year.
Here is an example table of values:



month	b	c	d	e	g	h	i	j	x
1	0.007047	0.018444	2.75E-06	-1.12696	-0.00434	0.012177	-1.6E-05	6.606466	679.15
2	0.006148	0.065628	2.76E-06	1.024448	-0.00545	0.037565	-1.5E-05	8.958362	684.28
3	0.001492	0.041412	1.47E-06	5.402304	-0.00656	0.039723	-5.7E-06	11.03547	699.67
4	-0.00239	0.053357	2.53E-07	10.24696	-0.00697	0.043574	6.13E-06	13.99039	818.03
5	0.004219	0.059169	1.86E-06	8.916032	-0.00707	0.056038	-1.2E-05	16.80074	698.66
6	-0.00222	0.046785	-1.3E-06	16.98087	-0.00684	0.078366	1.33E-05	20.81386	827.75
7	-0.00123	0.004955	-2.5E-06	19.38073	-0.00633	0.073345	-1.9E-06	23.54351	814.57
8	-0.00023	0.00407	-2.6E-06	18.81907	-0.00585	0.092057	9.03E-06	23.53079	836.26
9	0.001476	-0.0342	-2.3E-06	14.80655	-0.00612	0.047614	7.6E-06	20.83291	791.89
10	-0.00023	0.018716	-1.2E-06	11.70249	-0.00641	0.112236	-8.6E-06	16.77907	819.27
11	0.002002	0.014029	-2.6E-07	3.771477	-0.00556	0.027117	-1.4E-05	9.412401	745.95
12	0.00697	0.047469	2.59E-06	-1.28579	-0.00399	0.007256	-1.7E-05	6.096779	674.34

and here are the lines from the corresponding .csv data file:

```
month,b,c,d,e,g,h,i,j,x
1,0.007047,0.018444,2.75E-06,-1.12696,-0.00434,0.012177,-1.60E-05,6.606466,679.15
2,0.006148,0.065628,2.75E-06,1.024448,-0.00545,0.037565,-1.50E-05,8.958362,684.28
3,0.001492,0.041412,1.47E-06,5.402304,-0.00656,0.039723,-5.70E-06,11.03547,699.67
4,-0.00239,0.053357,2.53E-07,10.24696,-0.00697,0.043574,6.13E-06,13.99039,818.03
5,0.004219,0.059169,1.86E-06,8.916032,-0.00707,0.056038,-1.20E-05,16.80074,698.66
6,-0.00222,0.046785,-1.30E-06,16.98087,-0.00684,0.078366,1.33E-05,20.81386,827.75
7,-0.00123,0.004955,-2.50E-06,19.38073,-0.00633,0.073345,-1.90E-06,23.54351,814.57
8,-0.00023,0.00407,-2.60E-06,18.81907,-0.00585,0.092057,9.03E-06,23.53079,836.26
9,0.001476,-0.0342,-2.30E-06,14.80655,-0.00612,0.047614,7.60E-06,20.83291,791.89
10,-0.00023,0.018716,-1.20E-06,11.70249,-0.00641,0.112236,-8.60E-06,16.77907,819.27
11,0.002002,0.014029,-2.60E-07,3.771477,-0.00556,0.027117,-1.40E-05,9.412401,745.95
12,0.00697,0.047469,2.59E-06,-1.28579,-0.00399,0.007256,-1.70E-05,6.096779,674.34
```

The Air Temperature Coefficients File is only required when the spatially-explicit weather model is used. Otherwise, the file may be left unspecified in the simulator configuration.

The Observed Runoff Data File

An (optional, but recommended) temporal data file containing a runoff value in millimeters for each day of the simulation. The data is formatted according to the same rules as the precipitation driver data file; one value per line, one line per possible simulation day. The VELMA simulator engine can run without an observed runoff data file, but the Nash-Sutcliffe coefficient value automatically computed as part of the simulation results will be invalid if the observed runoff data is unavailable.

The Observed Stream Chemistry Data File

An optional temporal data file containing observed DON, NH_4, DOC and NO_3 loss values for each day of the simulation. The data is formatted as four comma-separated values per line, one line per possible simulation day. The first five lines of an observed stream chemistry data file might look like this:

```
0.000409543,7.73582E-05,0.013059883,9.10096E-06
4.76427E-05,9.52854E-06,0.002572705,0
9.50966E-05,2.8529E-05,0.006276374,4.75483E-06
4.76506E-05,4.76506E-06,0.002001324,0
4.76506E-05,4.76506E-06,0.001620119,0
[ … ]
```

In this example snippet, the first day’s observed loss values would be:

```
DON = 0.000409543
```

```
NH4 = 7.73582E-05
DOC = 0.013059883
NO3 = 9.10096E-06
```

Note that, like the Observed Runoff data file, there is no header row (initial line of column header titles) in the observed stream chemistry file.

V. Creating a Simulator Configuration (New VELMA Application)

Required Input Files

The following simulation data input files must be available prior to creating a new VELMA application:

1. JPDEM flat-processed Grid ASCII (.asc) DEM map file. (see JPDEM user guide in Appendix 7)
2. Cover Species ID Grid ASCII map (.asc) file.
3. Cover Species Age Grid ASCII map (.asc) file.
4. Soil Parameters ID Grid ASCII map (.asc) file.
5. Daily Rainfall driver data file (.txt or .csv) file.
6. Daily Air Temperature data file (.txt or .csv) file.

Configuration Steps

With the above files available, perform the following steps:

1. Start the JVelma graphical user interface (GUI)
2. Create a New Simulator Configuration
3. Specify Startup Parameters
4. Specify Input Parameters
5. Specify Weather Parameters
6. Add and Specify Parameters for Each Cover Species
7. Add and Specify Parameters for Each Soil Parameterization

Additional details for these steps appear in the following pages.

Start the JVelma GUI

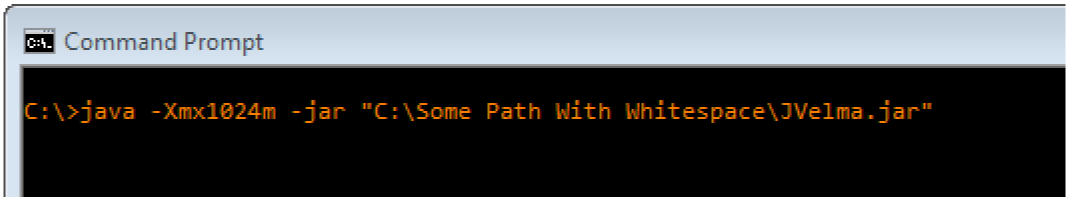
To start the JVelma GUI, open a Windows command prompt, and launch the GUI via Java and the JVelma.jar file.

Here is an example of the form the startup command takes:

```
C:\> java -Xmx1024m -jar C:\Some\full\Path\JVelma.jar
```

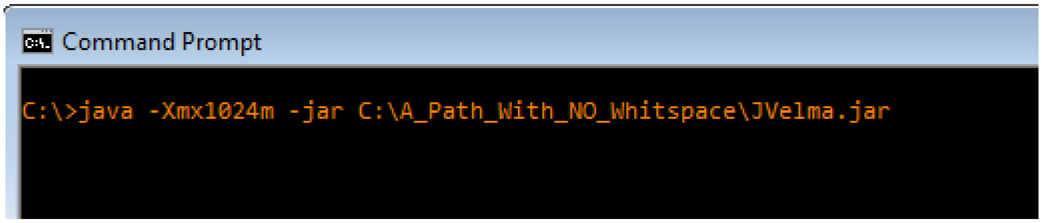
Of course, replace C:\Some\full\Path\ with the actual, fully-qualified path name of the location where your copy of the JVelma.jar file resides.

Here are a couple of screen captures showing actual command lines for starting JVelma:



```
C:\>java -Xmx1024m -jar "C:\Some Path With Whitespace\JVelma.jar"
```

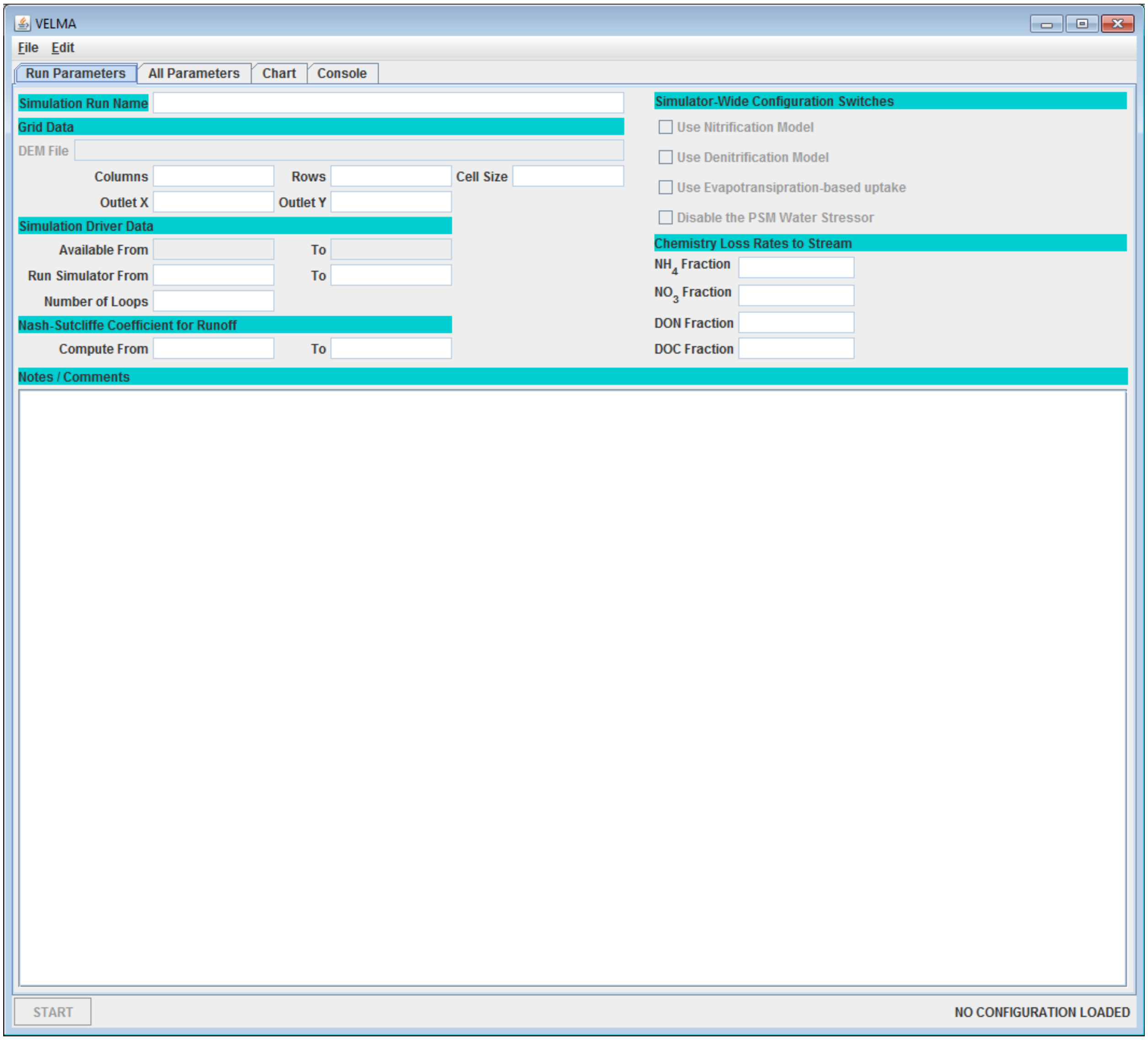
(In the above screen-capture, the double-quotes around the fully-qualified path+name of the JVelma.jar file are required because the text of the path contains whitespace.)



```
C:\>java -Xmx1024m -jar C:\A_Path_With_NO_Whitspace\JVelma.jar
```

(When there is no whitespace in the entire path the double-quotes are not required.)
The “-Xmx1024m” command line option specifies the amount of memory available to the JVelma GUI and simulator. The value 1024 is in Megabytes, so this command line allocates a 1 Gigabyte memory space for the JVelma GUI and simulator to run in. You can allocate more memory than 1 GB (e.g. the option “-Xmx4096m” would allocate 4 GB) but if you allocate more memory than your computer can make available, JVelma will fail to start properly.

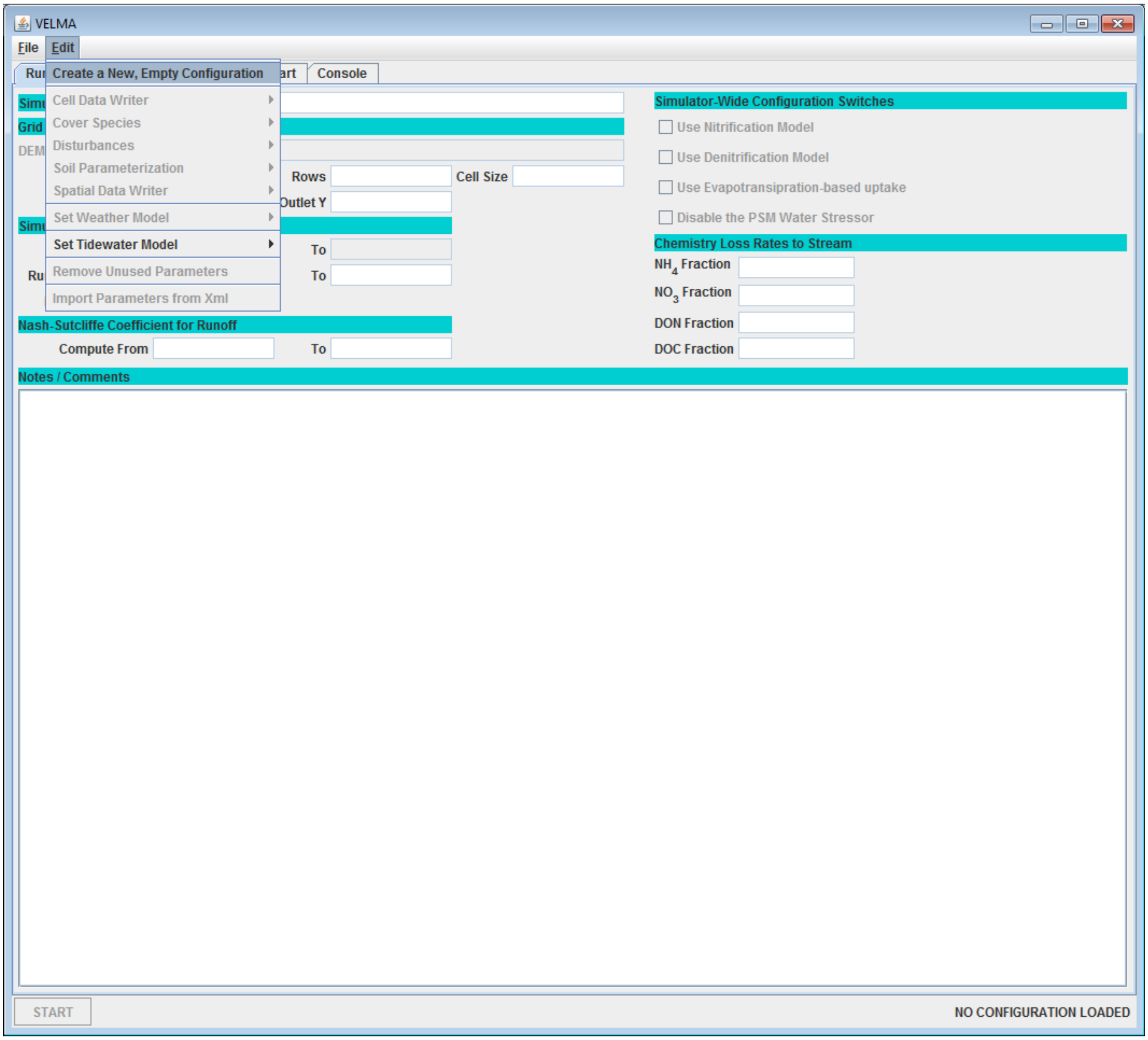
Once you type in the command line and press the enter key, the JVelma GUI should begin running. Initially, the GUI looks like this:



NO CONFIGURATION LOADED

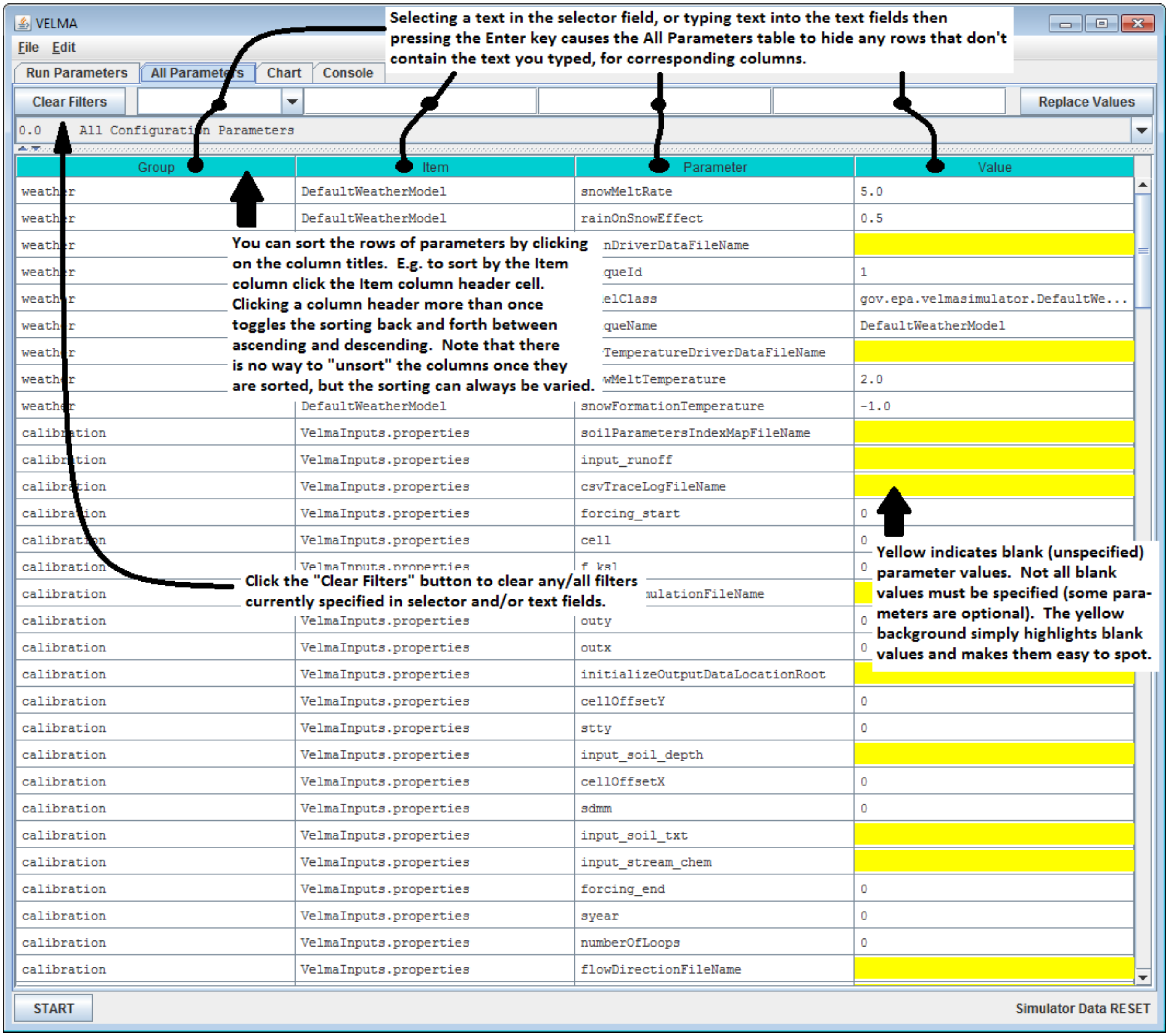
Using the GUI

Click the **Edit** menu’s **Create a New, Empty Configuration** item:



Clicking this item creates a new, empty simulation configuration in the JVelma GUI and opens the **All Parameters** table view. The All Parameters table lists the Group, Item and Parameter keys and the current value of all the parameters used by a single VELMA simulation run. When we speak of the “Simulation Configuration”, this list of parameters is what we’re speaking of.

For demonstration purposes, here is an annotated screen-captured snapshot of the All Parameters table for an existing simulator configuration (cells under the Value column will be blank for new simulator configurations):



Selecting a text in the selector field, or typing text into the text fields then pressing the Enter key causes the All Parameters table to hide any rows that don't contain the text you typed, for corresponding columns.

You can sort the rows of parameters by clicking on the column titles. E.g. to sort by the Item column click the Item column header cell. Clicking a column header more than once toggles the sorting back and forth between ascending and descending. Note that there is no way to "unsort" the columns once they are sorted, but the sorting can always be varied.

Click the "Clear Filters" button to clear any/all filters currently specified in selector and/or text fields.

Yellow indicates blank (unspecified) parameter values. Not all blank values must be specified (some parameters are optional). The yellow background simply highlights blank values and makes them easy to spot.

Group	Item	Parameter	Value
weather	DefaultWeatherModel	snowMeltRate	5.0
weather	DefaultWeatherModel	rainOnSnowEffect	0.5
weather		nDriverDataFileName	
weather		queId	1
weather		elClass	gov.epa.velmasimulator.DefaultWe...
weather		queName	DefaultWeatherModel
weather		TemperatureDriverDataFileName	
weather		wMeltTemperature	2.0
weather	DefaultWeatherModel	snowFormationTemperature	-1.0
calibration	VelmaInputs.properties	soilParametersIndexMapFileName	
calibration	VelmaInputs.properties	input_runoff	
calibration	VelmaInputs.properties	csvTraceLogFileName	
calibration	VelmaInputs.properties	forcing_start	0
calibration	VelmaInputs.properties	cell	0
calibration	VelmaInputs.properties	f_ksl	0
calibration		ulationFileName	
calibration	VelmaInputs.properties	outy	0
calibration	VelmaInputs.properties	outx	0
calibration	VelmaInputs.properties	initializeOutputDataLocationRoot	
calibration	VelmaInputs.properties	cellOffsetY	0
calibration	VelmaInputs.properties	stty	0
calibration	VelmaInputs.properties	input_soil_depth	
calibration	VelmaInputs.properties	cellOffsetX	0
calibration	VelmaInputs.properties	sdmm	0
calibration	VelmaInputs.properties	input_soil_txt	
calibration	VelmaInputs.properties	input_stream_chem	
calibration	VelmaInputs.properties	forcing_end	0
calibration	VelmaInputs.properties	syear	0
calibration	VelmaInputs.properties	numberOfLoops	0
calibration	VelmaInputs.properties	flowDirectionFileName	

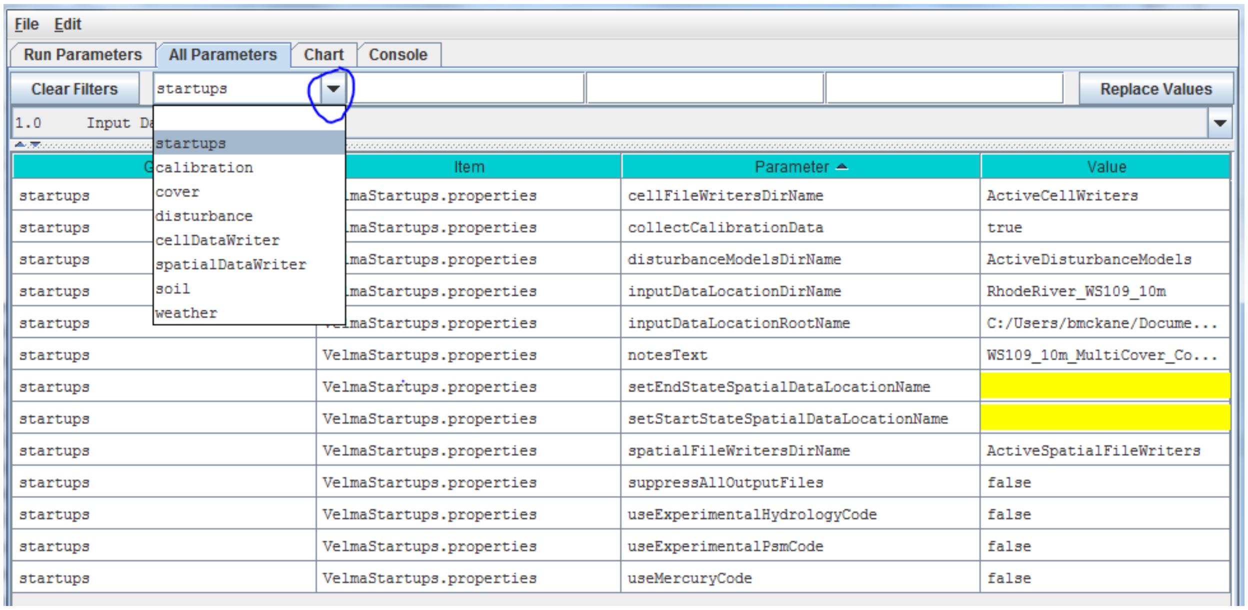
In addition to filtering by an arbitrary text string, the filter field at the top of the table can be set to any of several pre-composed filtering expressions.

For example, In the All Parameters table, click “Clear Filters” at the left of side of the filter fields, then click the drop-down button (▾) on the right side of the leftmost filter field to display a drop-down menu (see demonstration, below).

Using the GUI to Specify Startup and Input Parameters

Specify Startup Parameters

In the All Parameters table, click the first (leftmost) filter field's drop-down button and select "startups" from the drop-down list:



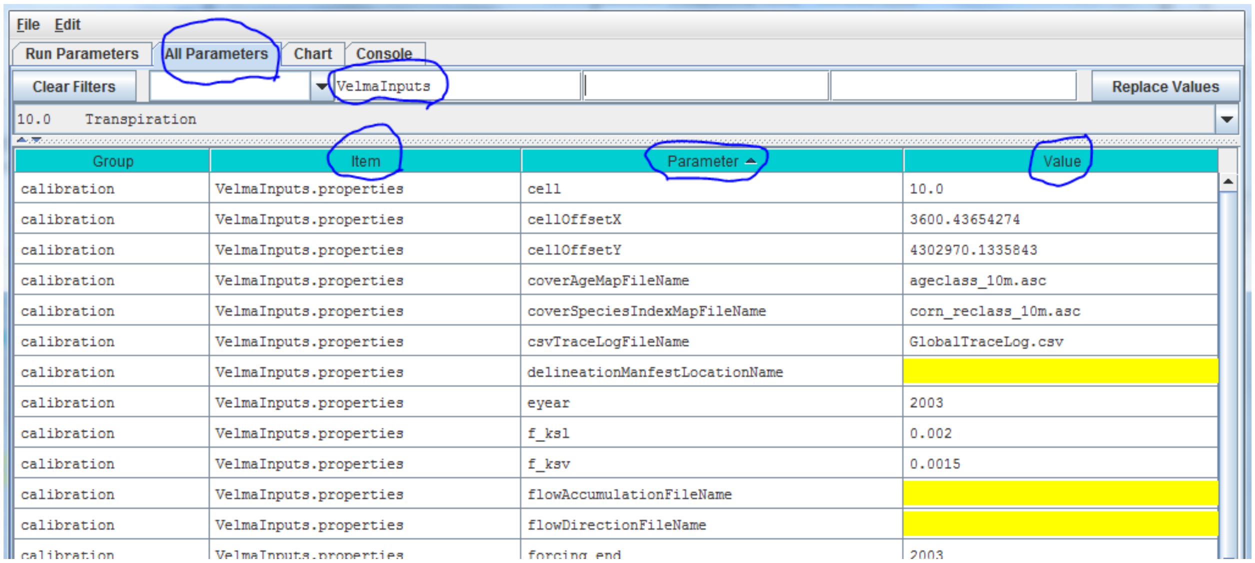
G	Item	Parameter ▲	Value
startups	maStartups.properties	cellFileWritersDirName	ActiveCellWriters
startups	maStartups.properties	collectCalibrationData	true
startups	maStartups.properties	disturbanceModelsDirName	ActiveDisturbanceModels
startups	maStartups.properties	inputDataLocationDirName	RhodeRiver_WS109_10m
startups	maStartups.properties	inputDataLocationRootName	C:/Users/bmckane/Docume...
startups	VelmaStartups.properties	notesText	WS109_10m_MultiCover_Co...
startups	VelmaStartups.properties	setEndStateSpatialDataLocationName	
startups	VelmaStartups.properties	setStartStateSpatialDataLocationName	
startups	VelmaStartups.properties	spatialFileWritersDirName	ActiveSpatialFileWriters
startups	VelmaStartups.properties	suppressAllOutputFiles	false
startups	VelmaStartups.properties	useExperimentalHydrologyCode	false
startups	VelmaStartups.properties	useExperimentalPsmCode	false
startups	VelmaStartups.properties	useMercuryCode	false

The resulting display shows 13 "startups" parameters under the 3rd column. For now, we will focus on 2 startup parameters that VELMA uses to locate simulation input files at the start of a simulation. You will need to specify a valid value in the 4th column for these:

`inputDataLocationDirName`	The name of the directory containing your simulation input data files
`inputDataLocationRootName`	The fully-qualified path above the directory name you specified for inputDataLocationDirName

Specify Input Parameters

Under the All Parameters menu tab, in the 2nd filter field to the left of the "Clear Filters" button, type "VelmaInputs" (case sensitive, without the double-quotes):



10.0 Transpiration

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	cell	10.0
calibration	VelmaInputs.properties	cellOffsetX	3600.43654274
calibration	VelmaInputs.properties	cellOffsetY	4302970.1335843
calibration	VelmaInputs.properties	coverAgeMapFileName	ageclass_10m.asc
calibration	VelmaInputs.properties	coverSpeciesIndexMapFileName	corn_reclass_10m.asc
calibration	VelmaInputs.properties	csvTraceLogFileName	GlobalTraceLog.csv
calibration	VelmaInputs.properties	delineationManfestLocationName	
calibration	VelmaInputs.properties	eyear	2003
calibration	VelmaInputs.properties	f_ksl	0.002
calibration	VelmaInputs.properties	f_ksv	0.0015
calibration	VelmaInputs.properties	flowAccumulationFileName	
calibration	VelmaInputs.properties	flowDirectionFileName	
calibration	VelmaInputs.properties	forcing_end	2003

In the displayed subset of configuration parameters, enter values for the following parameters:

`input_dem` The name (plus extension, “.asc”) of this simulation’s DEM spatial data file. The simulator will look for the file in the directory specified by the inputDataLocationDirName + “/” + inputDataLocationRootName that you specified as Startups parameters.

`cell` The width of a cell in the DEM grid (should match the “cellsize” header value in the DEM Grid ASCII file and will be a value in meters).

`ncol` The number of columns in the DEM grid (should match the DEM Grid ASCII file’s ncols header value).

`nrow` The number of rows in the DEM grid (should match the DEM Grid ASCII file’s nrows header value).

`cellOffsetX` Horizontal (column) offset of the DEM's zeroth pixel in meters.

`cellOffsetY` Vertical (row) offset of the DEM's zeroth pixel in meters.

`outx` The zero-based x-coordinate (column index) of the outlet cell for this simulation’s watershed.

`outy` The zero-based y-coordinate (row index) of the outlet cell for this simulation’s watershed.

`coverAgeMapFileName` The name (and “.asc” extension) of this simulation’s cover species age spatial data map.

`coverSpeciesIndexMapFileName` The name (and “.asc” extension) of this simulation’s cover species ID spatial data map.

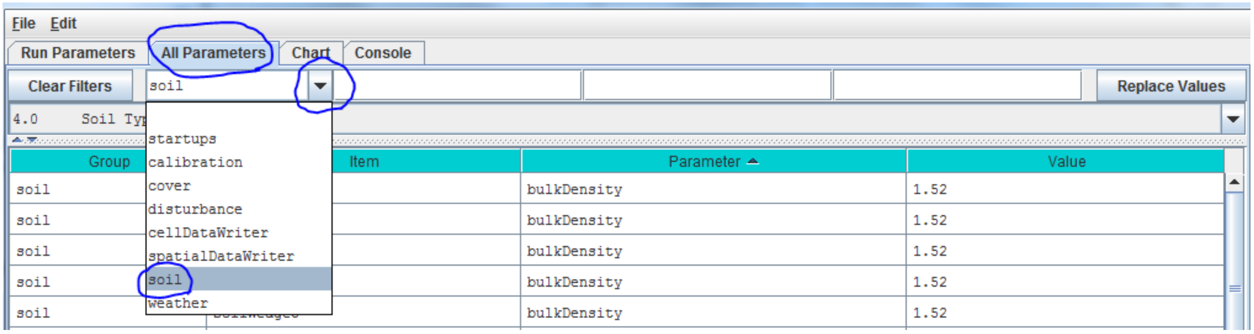
`soilParametersIndexMapFileName` The name (and “.asc” extension) of this simulation’s soil parameterizations ID spatial data map.

`forcing_start` The first year (inclusive) of temporal data available for simulation runs.

`forcing_end` The last year (inclusive) of temporal data available for simulation runs.

`syear`	The starting year of the simulation run.
`eyear`	The ending year of the simulation run.
`input_runoff`	The name of the observed runoff data for simulation runs. (Specifying this file is optional, but highly recommended.)
`input_stream_chem`	The name of the observed stream chemistry data for simulation runs. (Specifying this file is optional, but highly recommended.)
`run_index`	The name of the directory where the VELMA simulation will save simulation results files.
`initializeOutputDataLocationRoot`	The fully-qualified path above the directory name you specified for the `run_index`.

The drop-down filter field menu above the “Group” column can also be used to add and specify parameters for the other parameter categories listed: cover, disturbance, `cellDataWriter`, `spatialDataWriter`, soil, and weather:



soil

4.0 Soil Ty

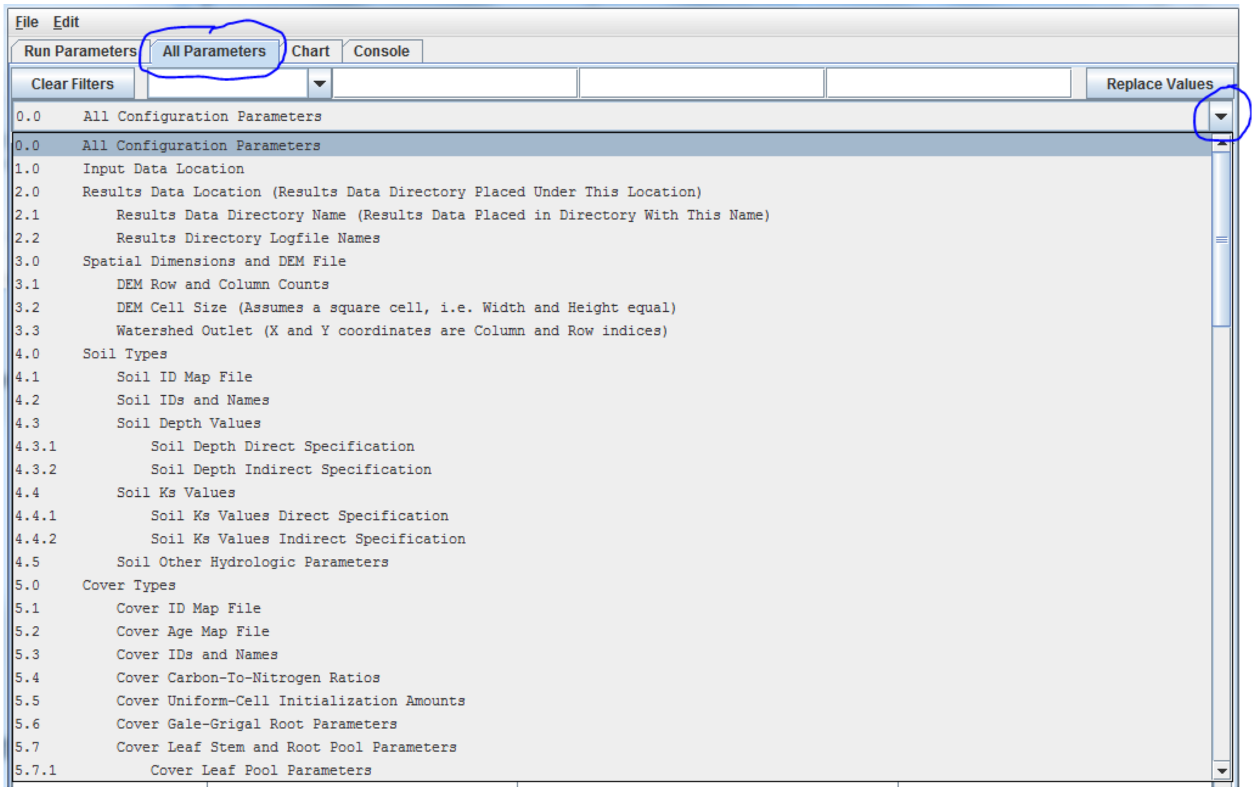
Group	Item	Parameter ▲	Value
soil		bulkDensity	1.52
soil		bulkDensity	1.52
soil		bulkDensity	1.52
soil		bulkDensity	1.52
soil		bulkDensity	1.52

However, we recommend that you use the “All Parameters Table of Contents” described in the next section to configure these and other subsets of parameters.

Overview of the All Parameters Table of Contents (TOC)

The “All Parameters Table of Contents” (TOC) provides users with a logical sequence of steps for creating a new simulator configuration, or for quickly locating and calibrating parameters for existing configurations.

To access the All Parameter TOC, click the GUI’s “All Parameters” tab, then click the drop-down button (▼) located on the right side of the 4-window filter field (don’t confuse this button with the one located next to the leftmost filter field):



```
0.0     All Configuration Parameters
0.0     All Configuration Parameters
1.0     Input Data Location
2.0     Results Data Location (Results Data Directory Placed Under This Location)
2.1         Results Data Directory Name (Results Data Placed in Directory With This Name)
2.2         Results Directory Logfile Names
3.0     Spatial Dimensions and DEM File
3.1         DEM Row and Column Counts
3.2         DEM Cell Size (Assumes a square cell, i.e. Width and Height equal)
3.3         Watershed Outlet (X and Y coordinates are Column and Row indices)
4.0     Soil Types
4.1         Soil ID Map File
4.2         Soil IDs and Names
4.3         Soil Depth Values
4.3.1           Soil Depth Direct Specification
4.3.2           Soil Depth Indirect Specification
4.4         Soil Ks Values
4.4.1           Soil Ks Values Direct Specification
4.4.2           Soil Ks Values Indirect Specification
4.5         Soil Other Hydrologic Parameters
5.0     Cover Types
5.1         Cover ID Map File
5.2         Cover Age Map File
5.3         Cover IDs and Names
5.4         Cover Carbon-To-Nitrogen Ratios
5.5         Cover Uniform-Cell Initialization Amounts
5.6         Cover Gale-Grigal Root Parameters
5.7         Cover Leaf Stem and Root Pool Parameters
5.7.1           Cover Leaf Pool Parameters
```

That drop-down menu that appears is designed to organize subsets of model parameters in a "Table of Contents" format. The screenshot above shows the first page of the All Parameters TOC. Additional pages can be viewed by holding your mouse cursor over the TOC and scrolling down.

In all, the TOC includes 26 main sections (0.0 – 25.0), each of which addresses a particular simulation configuration procedure.

The All Parameters TOC is designed to:

- Provide a structured step-by-step guide for building a new simulator configuration.
- Help users quickly make specific adjustments to an existing simulator configuration, for example, to change the directory path and/or folder name for input and output files, or to calibrate a particular subset of parameters for a hydrological or biogeochemical process.

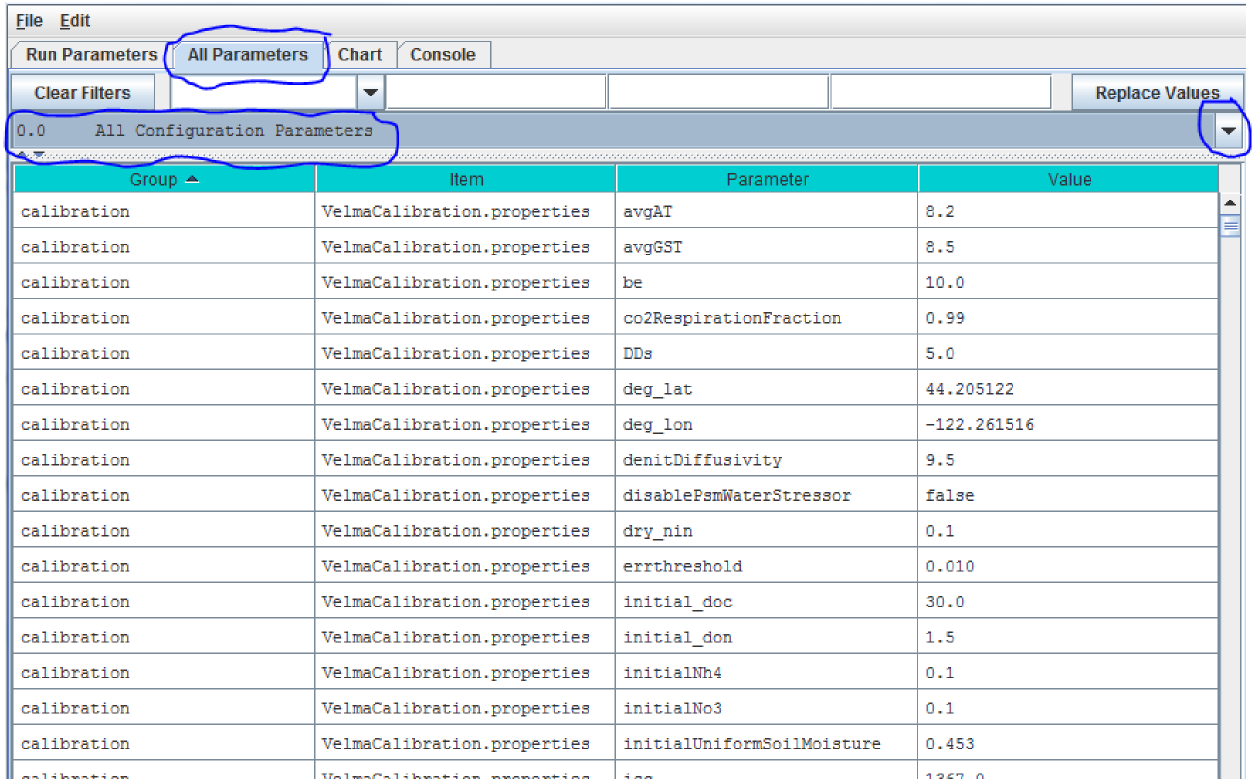
All Parameters Table of Contents

Click an underlined hyperlink to jump to configuration instructions

VI. Steps for Configuring All Parameters TOC Sections 0.0 – 25.0

0.0 – All Configuration Parameters (link to All Parameters TOC)

Using the drop-down button (▾) on the right side of the filter field, select *All Configuration Parameters* from the *All Parameters* menu.



Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	avgAT	8.2
calibration	VelmaCalibration.properties	avgGST	8.5
calibration	VelmaCalibration.properties	be	10.0
calibration	VelmaCalibration.properties	co2RespirationFraction	0.99
calibration	VelmaCalibration.properties	DDs	5.0
calibration	VelmaCalibration.properties	deg_lat	44.205122
calibration	VelmaCalibration.properties	deg_lon	-122.261516
calibration	VelmaCalibration.properties	denitDiffusivity	9.5
calibration	VelmaCalibration.properties	disablePsmWaterStressor	false
calibration	VelmaCalibration.properties	dry_nin	0.1
calibration	VelmaCalibration.properties	errthreshold	0.010
calibration	VelmaCalibration.properties	initial_doc	30.0
calibration	VelmaCalibration.properties	initial_don	1.5
calibration	VelmaCalibration.properties	initialNh4	0.1
calibration	VelmaCalibration.properties	initialNo3	0.1
calibration	VelmaCalibration.properties	initialUniformSoilMoisture	0.453
calibration	VelmaCalibration.properties	isc	1367.0

This view displays all *Groups*, *Items*, *Parameters* and *Values* for a simulator configuration in a single, scrollable view. To see a pop-up description of any *Group*, *Item* or *Parameter* in any cell, hover the mouse cursor over the cell.

Alternatively, these descriptions are summarized here in an Excel file located here:

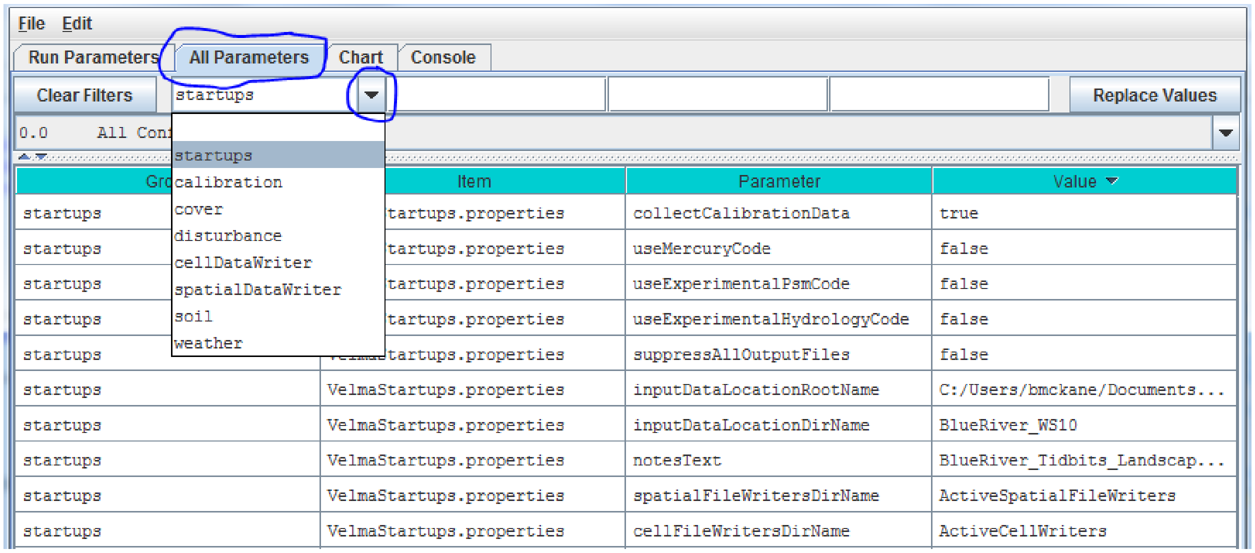
Filename: VELMA 2.0_Description of Calibration Parameters.xlsx

Folder location: VELMA Model\Supporting Documents\Excel Calibration Files

The *All Configuration Parameters* view contains an overwhelming amount of information, but the user has some search options for organizing and selecting things of interest. Experienced users may find that any of the first three search options, below, are useful for finding a specific parameter of interest in an existing simulator configuration.

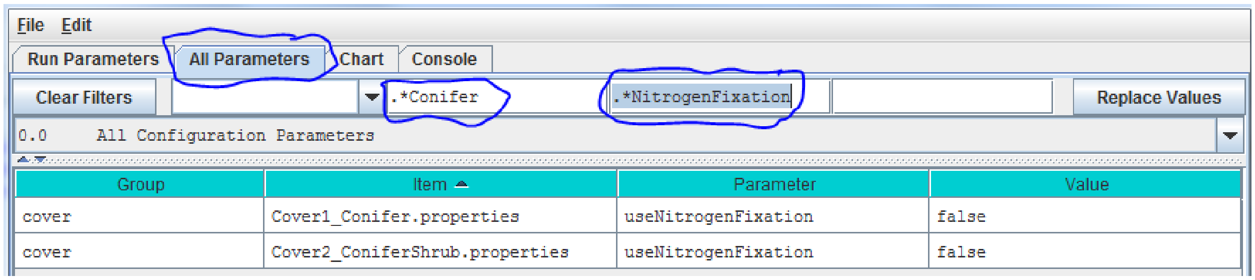
Option 1 – For some purposes it is useful to use the *All Configuration Parameters* view to locate, sort and select parameters of interest. For example, you can sort columns by double clicking the *Group*, *Item*, *Parameter* or *Value* column headers (turquois). You can also right click a particular cell and select "Set Column Filter to cellname", which will display values for the selected item across all cover types, soil types, disturbance type, etc. For example, this is useful if your watershed has more than one cover or soil type.

Option 2 – Use the drop-down button (▼) on the right side of the first filter field (above *Group*) to select from the list of model configuration categories: startups, calibration, cover, disturbance, `cellDataWriter`, `spatailDataWriter`, soil, and weather (the meaning and use of these categories will be explained below, under sections 1.0 to 25.0).



Gro	Item	Parameter	Value
startups	tartups.properties	collectCalibrationData	true
startups	tartups.properties	useMercuryCode	false
startups	tartups.properties	useExperimentalPsmCode	false
startups	tartups.properties	useExperimentalHydrologyCode	false
startups	tartups.properties	suppressAllOutputFiles	false
startups	VelmaStartups.properties	inputDataLocationRootName	C:/Users/bmckane/Documents...
startups	VelmaStartups.properties	inputDataLocationDirName	BlueRiver_WS10
startups	VelmaStartups.properties	notesText	BlueRiver_Tidbits_Landscap...
startups	VelmaStartups.properties	spatialFileWritersDirName	ActiveSpatialFileWriters
startups	VelmaStartups.properties	cellFileWritersDirName	ActiveCellWriters

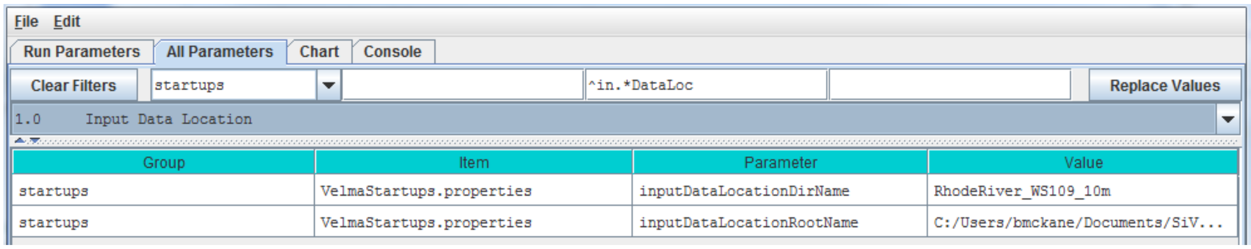
Option 3 – Type a search string in the filter fields located above any of the four turquois column headers. Suppose you want to make sure the nitrogen fixation subroutine is turned off for a coniferous cover type. As the screenshot below shows, by typing ".*Conifer in the "Item" filter field and ".*NitrogenFixation" in the "Parameter" filter field. This search verified that the "useNitrogenFixation" parameter is turned off (false).



Group	Item	Parameter	Value
cover	Cover1_Conifer.properties	useNitrogenFixation	false
cover	Cover2_ConiferShrub.properties	useNitrogenFixation	false

Option 4 (recommended) – Although experienced users may find Options 1 - 3 handy for quickly locating specific Items, Parameters or Values, we recommend that all users stick with using sections 1.0 – 25.0 of the All Parameters TOC when configuring a new VELMA application. These sections collectively contain the same information listed under "All Configuration Parameters", but in a sequence of steps organized to assist users in building new simulator configurations or editing existing ones. Details follow.

1.0 – Input Data Location (link to All Parameters TOC)



startups | ^in.*DataLoc

1.0 Input Data Location

Group	Item	Parameter	Value
startups	VelmaStartups.properties	inputDataLocationDirName	RhodeRiver_WS109_10m
startups	VelmaStartups.properties	inputDataLocationRootName	C:/Users/bmckane/Documents/SiV...

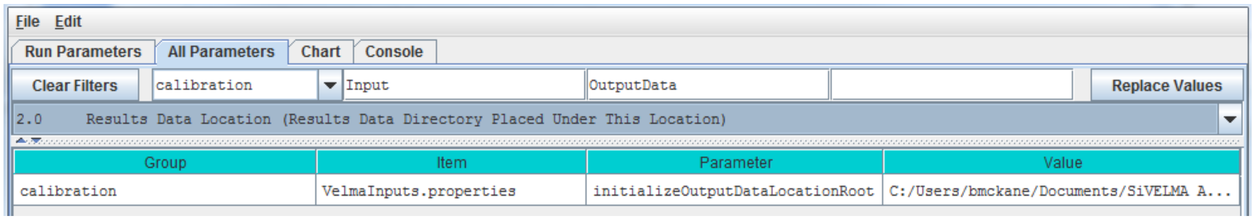
Parameter Definitions

`inputDataLocationDirName`	The name of a subdirectory of the inputDataLocationRootName directory that contains the set of input data for a specific simulation run. When left unspecified or commented-out defaults to "". (I.e. VELMA will look for input data in the inputDataLocationRootName directory itself)
`inputDataLocationRootName`	Fully-qualified path to the directory containing input data directories. When left unspecified or commented-out defaults to the "../data/" subdirectory of VELMA's installation directory.

For the `inputDataLocationRootName` parameter, specify a directory name for the folder that contains the set of input data for a given simulation configuration. The `inputDataLocationRootName` is the path to that directory.

2.0 – Results Data Location (link to All Parameters TOC)

For the *initializeOutputDataLocationRoot* parameter, specify the root directory under which VELMA will place simulation output files for a particular model run.



calibration | Input | OutputData

2.0 Results Data Location (Results Data Directory Placed Under This Location)

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	initializeOutputDataLocationRoot	C:/Users/bmckane/Documents/SiVELMA A...

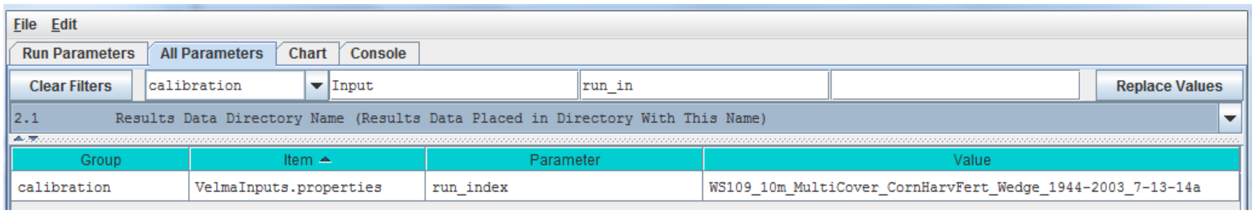
Parameter Definitions

`initializeOutputDataLocationRoot`	Root directory under which to place per-simulation-run directories. If this property is left unspecified per-simulation-run directories for output files will be placed underneath a default root created in the ../data subdirectory of the VELMA simulator's installation directory. When specified this property must be a fully-qualified path name. Whitespace in the name does not have to be escape-delimited or double-quoted and the "/" can (and should!) be used as the path separator for both Unix and Windows paths. EXAMPLE: initializeOutputDataLocationRoot = C:/My VelmaResults/VelmaResultsRoot For the above example subdirectories for each VELMA simulator run are placed

underneath the directory "C:/My VelmaResults/VelmaResultsRoot/"

2.1 – Results Data Directory Name (Results data placed in directory with this name)

Specify a unique simulation filename in the value column for the *run_index* parameter.



calibration | Input | run_in

2.1 Results Data Directory Name (Results Data Placed in Directory With This Name)

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	run_index	WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a

<u>Parameter Definitions</u>

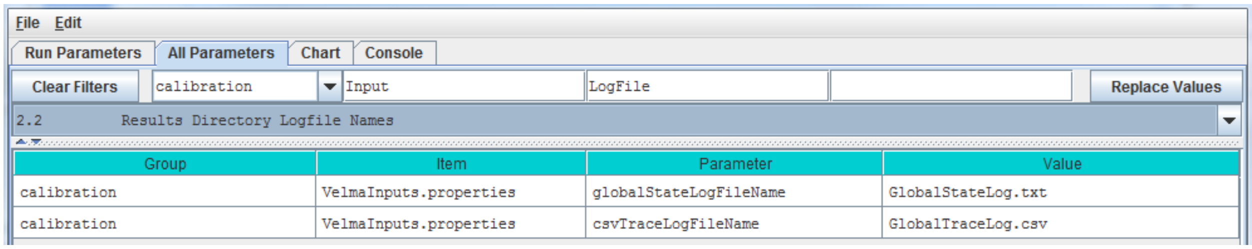
`run_index` Watershed Name (used as core name for the simulation model run data files).

Use this parameter to specify a unique name for each simulation run that you do. For example, a useful run_index name might include the watershed name, date and other identifiers that will make it easier to determine when a simulation configuration was built and how it differs from others. VELMA will use the run_index name to create:

- the *Simulation Run Name* listed in the upper left corner of the *Run Parameters* GUI
- the .XML file name that VELMA writes when the user saves a configuration file (“Ctrl-S”, or “File/Save Configuration To VELMA XML File” using the menu at the top of the *Run Parameters* GUI). This XML file contains all parameter names and values for the saved configuration. VELMA will not save this information if the user does not save it in an XML file;
- the name of the subdirectory (along with the full path) to which model output data files are written.

2.2 – Results Directory Logfile Names

Specify a filename in the Value column for the `globalStateLogFilename` parameter.



calibration | Input | LogFile

2.2 Results Directory Logfile Names

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	globalStateLogFileName	GlobalStateLog.txt
calibration	VelmaInputs.properties	csvTraceLogFileName	GlobalTraceLog.csv

<u>Parameter Definitions</u>

`globalStateLogFilename` The name of the global log file. The Global log file records console messages emitted during a simulation run. When left blank ("") or commented-out no global log file is created.

`csvTraceLogFilename` The name of the trace log file. The trace log file records any low-level debugging messages emitted during a simulation run. When left blank ("") or commented-out no trace log file is created.

VELMA will use the `globalStateLogFilename` to record the stream of runtime console messages (see Console tab) generated during a simulation run. This is optional, but we recommend creating a `globalStateFileName` to help identify any problems with the simulator configuration file. For example, the console messages recorded in the `globalStateLogFilename` provide a complete, sequential log of runtime information about successful or unsuccessful loading of input files, activation of disturbance events (harvest, fertilization, etc.), saving results to output directories/files, and so on.

///// IMPORTANT NOTE /////

For important additional details on configuring, saving and understanding simulator results, please see Appendix 4: Overview of VELMA Simulator Output.

3.0 – Spatial Dimensions and DEM File (link to All Parameters TOC)

The VELMA simulator requires a DEM file in order to run a simulation. The simulator also assumes that all other spatial data files (discussed below) contain spatial data with the same overall (row and column) and cell (pixel) size dimensions as the DEM file. We highly recommend that the DEM used for VELMA simulations be pre-processed using the JPDEM software, rather than by other available "flat-processor" software (see Appendix 7).The VELMA simulation configuration parameter that specifies the name of the DEM file is:

`input_dem` example: `SmallWatershed_10m.asc`

The VELMA simulation configuration parameters that define the DEM file's location are:

`InputDataLocationRootName` example: `CoastalSite`

and

`inputDataLocationDirName` example: `C:\VelmaInputData`

Note that:

- `Input_dem and inputDataLocationRootName` are single file or directory names, while `inputDataLocationDirName` specifies a complete directory path.)
- When a simulation is started, the DEM data is loaded from `inputDataLocationRootName/inputDataLocationDirName/input_dem`
- The DEM File's Dimensions are NOT Set Automatically by the Simulator or GUI
- The VELMA simulator does not read the DEM file's dimensions from the DEM file's header!

As part of parameterizing a VELMA simulation configuration, you must manually determine (usually by viewing the DEM file's header rows with a text editor) the DEM file dimensions, and set the corresponding VELMA simulation parameters (usually by using the JVelma GUI)

This table lists the header rows of a DEM (Grid ASCII) file and the corresponding VELMA simulation parameters:

Grid ASCII Header Item	VELMA Simulation Configuration Parameter
ncols	Ncol
nrows	Nrow
xllcorner	cellOffsetX

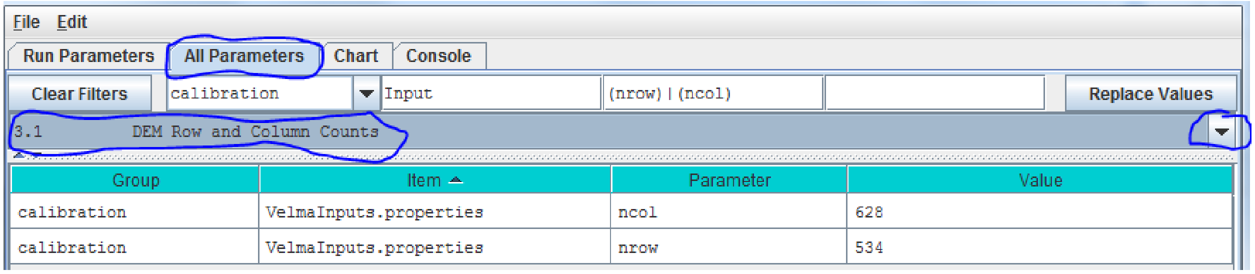
yllcorner	cellOffsetY
cellsize	cell
nodata_value	

Sections 3.1 – 3.3 provide details on configuring these parameters for a DEM file.

Note: if you change the DEM file that a simulation is configured to use (i.e. you change the DEM filename for the input_dem parameter) then you must ensure that all of the parameters above are updated as well.

3.1 – DEM Row and Column Counts

Specify values for the `ncol` and `nrow` parameters based on the DEM grid ASCII header items in the table above.



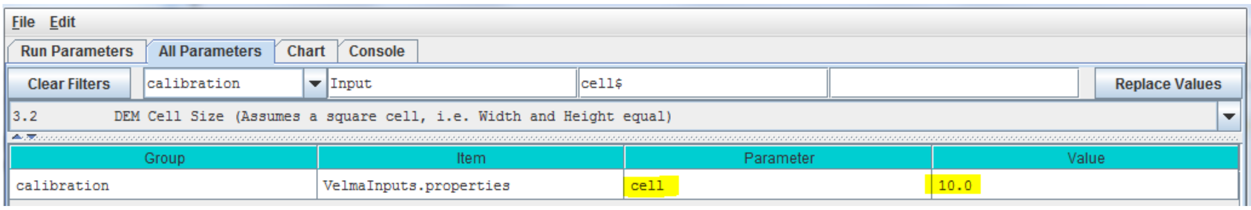
calibration | Input | (nrow)|(ncol)

3.1 DEM Row and Column Counts

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	ncol	628
calibration	VelmaInputs.properties	nrow	534

3.2 – DEM Cell Size (Assumes a square cell, i.e. Width and Height equal)

Specify the value of the `cell` parameter.



calibration | Input | cell$

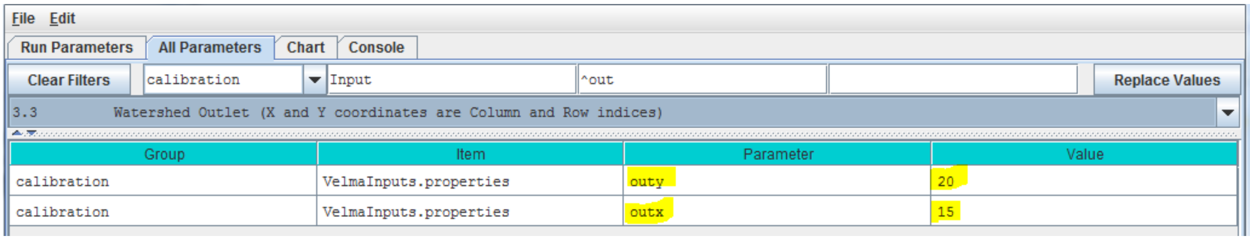
3.2 DEM Cell Size (Assumes a square cell, i.e. Width and Height equal)

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	cell	10.0

This describes the xy-dimension in meters of individual cells in the DEM grid ASCII file. In this example, each cell is 10x10 meters. The choice of DEM cell size will depend upon the set of questions a user wishes to address. For example, applications focusing on the effect of riparian buffer width on nutrient export to streams may require relatively small cells (e.g., 10 meters), compared to applications focusing on ecosystem carbon dynamics (e.g., 100 meters). Obviously, an application's total number of cells (and consequent simulation time) increases with the square of the ratio of alternative cell sizes – e.g., $(100/10)^2$ = 100 times more cells for 10m versus 100m cell sizes. In any case, the specified cell size must be consistent with the DEM cell size. That is, the user cannot arbitrarily adjust the cell size parameter without telling VELMA to load a DEM having the same cell size (see All Parameters 3.0). Furthermore, the DEM used must be "flat-processed" for the specified cell size using the JPDEM tool. The executable JPDEM.jar, a JPDEM user manual, and background information (Pan et al. 2012) can be found here: VELMA Model\JPDEM_DEM Processor. The JPDEM user manual is also included in Appendix 7 of this manual.

3.3 – Watershed Outlet (X and Y coordinates are Column and Row indices)

Specify the value of the `outx` and `outy` parameters. These describe the watershed outlet cell's X and Y coordinates (column index) relative to the DEM upper-left corner. The watershed outlet cell coordinates are determined using the JPDEM tool.



3.3 Watershed Outlet (X and Y coordinates are Column and Row indices)

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	outy	20
calibration	VelmaInputs.properties	outx	15

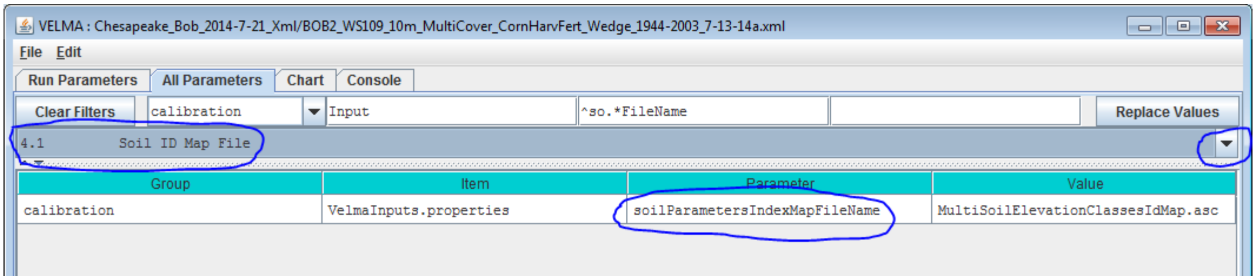
4.0 – Soil Types (link to All Parameters TOC)

VELMA requires information on key soil properties affecting hydrological and biogeochemical processes within a study watershed. Whether single or multiple soil types are modeled within a watershed, you will need to establish a soil type map and specify parameter values for a set of soil properties associated with each soil type. The following steps will accomplish this:

- Establish a soil ID map describing the assignment of soil type IDs for every cell within the delineated watershed. See section 4.1 for details.
- Specify soil ID parameter names and values corresponding to those appearing in the soil ID map. See section 4.2 for details.
- Specify soil depth values for the entire soil column and for each of four soil layers within the column. See section 4.3 for details.
- Specify soil hydraulic conductivity values controlling infiltration rate (Ks), and depth dependent changes in vertical and lateral macropore flow, Ksv and Ksl, respectively. See section 4.4 for details.

4.1 – Soil ID Map File

The Soil ID Map File is specified by the *soilParametersIndexMapFileName* parameter. You can view and set it by selecting "4.1 Soil ID Map File" in the All Parameters outline drop-down selector:



4.1 Soil ID Map File

Group	Item	Parameter	Value
calibration	VelmaInputs.properties	soilParametersIndexMapFileName	MultiSoilElevationClassesIdMap.asc

Parameter Definitions

`soilParametersIndexMapFileName`	The file name of a spatial-explicit map of Soil Parameters uniqueIds. The data in this map file specifies the Soil Parameterization of a given pixel in the DEM map.

The file specified for the `soilParametersIndexMapFileName` is assumed to be a Grid ASCII (".asc") map file with the same row and column dimensions as the simulation configuration's

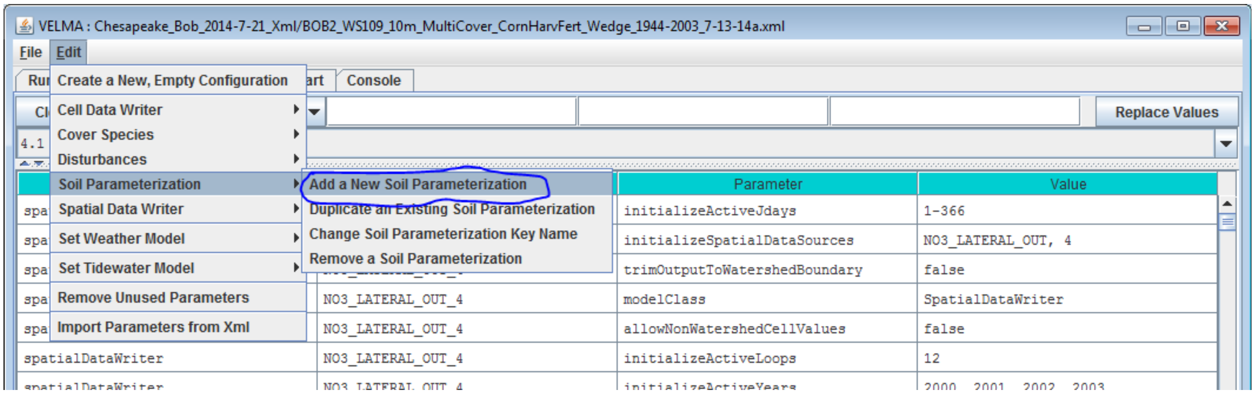
DEM file. Its contents should be integer values. Each cell's integer value should be the ID number of a soil type.

Suppose the specified soil index map file contains three distinct integers for this map: 100, 105 and 110. The VELMA simulation will expect the simulation configuration to include 3 soil parameterizations.

4.2 – Soil IDs and Names

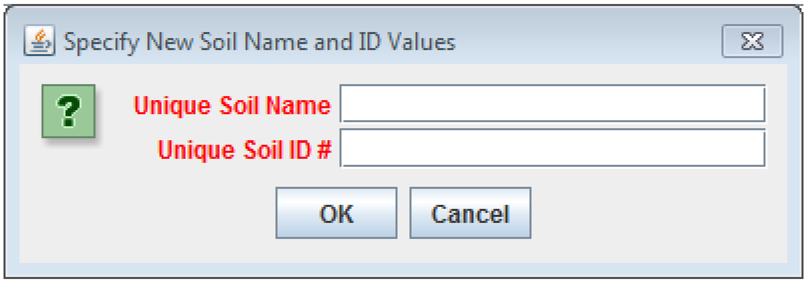
Adding a Soil Parameterization to a simulation configuration using the following steps.

Click the Edit → Soil Parameterizations → "Add a New Soil Parameterization" menu item:


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Create a New, Empty Configuration
Cell Data Writer
Cover Species
Disturbances
Soil Parameterization
Spatial Data Writer
Set Weather Model
Set Tidewater Model
Remove Unused Parameters
Import Parameters from Xml
Add a New Soil Parameterization
Duplicate an Existing Soil Parameterization
Change Soil Parameterization Key Name
Remove a Soil Parameterization
Console
Replace Values
4.1

		Parameter	Value
spa		initializeActiveJdays	1-366
spa		initializeSpatialDataSources	NO3_LATERAL_OUT, 4
spa		trimOutputToWatershedBoundary	false
spa	NO3_LATERAL_OUT_4	modelClass	SpatialDataWriter
spa	NO3_LATERAL_OUT_4	allowNonWatershedCellValues	false
spatialDataWriter	NO3_LATERAL_OUT_4	initializeActiveLoops	12
spatialDataWriter	NO3_LATERAL_OUT_4	initializeActiveYears	2000, 2001, 2002, 2003

Clicking "Add a New Soil Parameterization" opens the Soil Parameterization Name and ID dialog which looks like this:



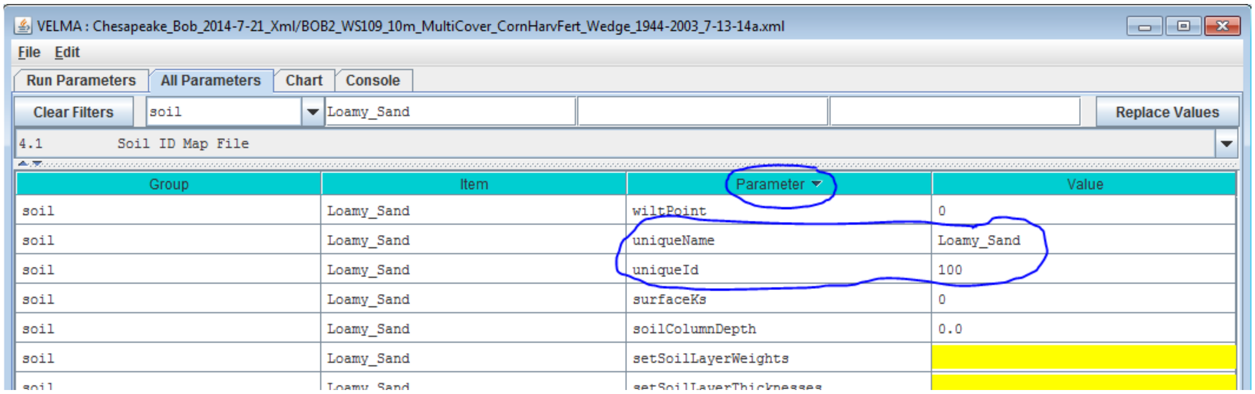
Enter a name and ID number for your new Soil Parameterization.

The name must be unique (i.e. no other Soil Parameterization already specified for this simulation configuration can share the name you specify) and we recommend avoiding whitespace and punctuation characters (e.g. "(" and ")") apart from the underbar ("_") character, which can be used instead of whitespace.
The ID number must be an integer, not already specified for any existing Soil Parameterization, and must match one of the integers that appears in the Soil ID Map File.

As an example: Suppose we are adding the first Soil Parameterization ("100") for the hypothetical Soil ID Map File mentioned above, and that ID "100" identifies cells that contain loamy sand.

Acceptable names (it's guaranteed unique because it's the first Soil Parameterization we're adding) might be: "Loamy_Sand" or "Soil_100_LoamySand", or something similar. However, the only acceptable ID value is "100" – because the Unique Soil ID # of this parameterization is what keys it to those particular cells in the Soil ID Map File.

Once you've entered valid name and ID values in the Name and ID dialog, click the OK button. When you click OK, the VELMA GUI adds a new Soil Parameterization to the simulation configuration, and sets the All Parameters tab's filters to display only the parameters of the newly-added Soil Parameterization. Assuming we named our new Soil Parameterization "Loamy_Sand", with ID = "100", this is what the All Parameters tab would look like after clicking OK:



VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml

soil | Loamy_Sand

4.1 Soil ID Map File

Group	Item	Parameter	Value
soil	Loamy_Sand	wiltPoint	0
soil	Loamy_Sand	uniqueName	Loamy_Sand
soil	Loamy_Sand	uniqueId	100
soil	Loamy_Sand	surfaceKs	0
soil	Loamy_Sand	soilColumnDepth	0.0
soil	Loamy_Sand	setSoilLayerWeights	
soil	Loamy_Sand	setSoilLayerThicknesses	

We did one other thing to get the above display: notice that the "Parameter" column is circled, and has a "down-arrow" indicating "descending-sort". After clicking OK to add the new Soil Parameterization, we clicked the "Parameter" header field (twice), which sorts the table of properties on that column.

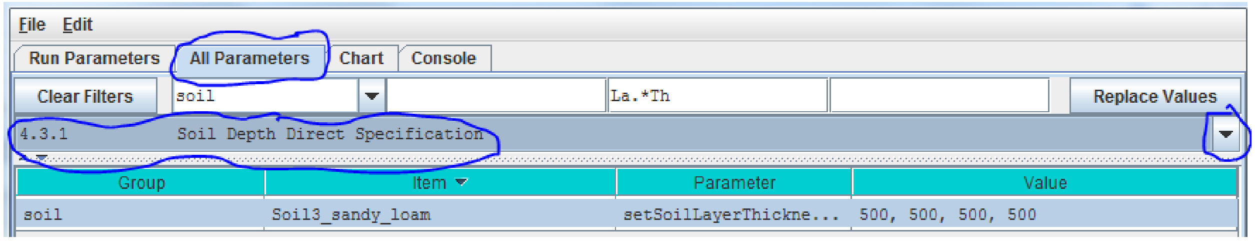
Now go to *All Parameters 4.3* and *4.4* where parameter values for each soil name/ID will need to be specified for total soil column depth, thickness of each of four soil layers, and hydrologic parameters affecting vertical and lateral flow.

4.3 – Soil Depth Values

There are two methods, direct or indirect, that can be used to specify soil layer depths. The direct and indirect methods have their own distinct set of parameters that are available to each soil type specified for a given simulation configuration. Each method has its advantages. Thus, it is up to the user to decide which one to use. Either the direct or indirect set of parameters can be applied to a specified soil type, independent of which set is applied in other specified soil types.

4.31 – Soil Depth Direct Specification

The direct method (optional) for setting a soil type's layer thicknesses uses a comma-separated list of values for the `setSoilLayerThicknesses` parameter in millimeters for of each of 4 soil layers. For example: "500, 500, 500, 1500" (without the quote-marks) specifies the soil type's 4 layers as being 500 mm, 500 mm 500 mm, and 1500 mm thick, from layer 1 to layer 4, respectively.



soil | | La.*Th

4.3.1 Soil Depth Direct Specification

Group	Item	Parameter	Value
soil	Soil3_sandy_loam	setSoilLayerThickne...	500, 500, 500, 500

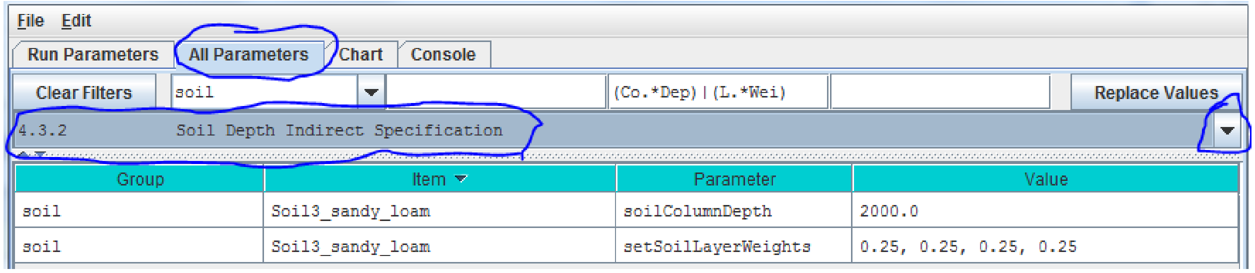
Parameter Definitions

`setSoilLayerThicknesses` A comma-separated list of values specifying the thickness of each soil layer in millimeters. For example: "500 750 1000 1500" (without the quote-marks) specifies the soil's 4 layers as being 500 mm 750 mm 1000 mm and 1500 mm thick from layer 1 to layer 4 respectively. NOTE: This list of values may be left unspecified. When `setSoilLayerThickness` is unspecified the soil's layer thicknesses are set by the combination of the `soilColumnDepth` and `setSoilLayerWeights` parameters. When `setSoilLayerThickness` is specified the `soilColumnDepth` and `setSoilLayerWeights` parameters are ignored.

Thus, for the direct method, VELMA internally calculates the value for the *soilColumnDepth* parameter from the `setSoilLayerThicknesses` values specified for layers 1, 2, 3 and 4. When `setSoilLayerThicknesses` is not specified (left blank), the indirect method must be used to specify the soil layer thicknesses and soil column depth (see All Parameters 4.32).

4.32 – Soil Depth Indirect Specification

As an alternative to the direct method described under *All Parameters 4.31*, you can instead use the indirect method for specifying a soil type's total soil column depth (parameter = `soilColumnDepth`, in millimeters) and soil layer depth (parameter = `setSoilLayerWeights`, which is equal to the fraction "soil layer thickness / soil column depth".



Group	Item	Parameter	Value
soil	Soil3_sandy_loam	soilColumnDepth	2000.0
soil	Soil3_sandy_loam	setSoilLayerWeights	0.25, 0.25, 0.25, 0.25

In this example the soil type named "SoilWedge1" has `soilColumnDepth` = 2000 mm, and `setSoilLayerWeights` = 0.25, 0.25, 0.25, 0.25 for layers 1, 2, 3 and 4, respectively. VELMA uses these values during a simulation to compute each soil layer's thickness (in millimeters) as:

`layerThickness`[iLayer] = `soilColumnDepth` * `soilLayerWeight`[iLayer].

Thus, the parameter values shown in the above example translate to soil layer thicknesses of 500 mm, 500 mm, 500 mm, and 500 mm for each layer.

When `setColumnDepth` and `setSoilLayerWeights` are not specified (left blank), the direct method must be used to specify the soil layer thicknesses and soil column depth (see All Parameters 4.31).

4.4 – Soil Ks Values

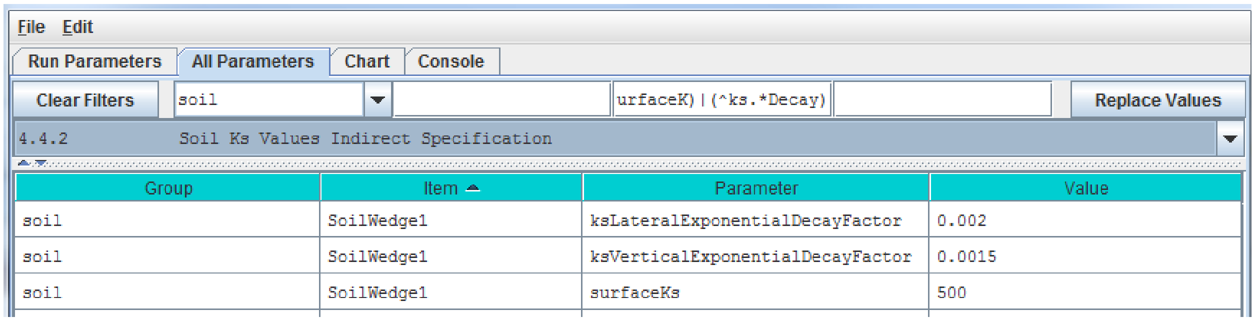
VELMA simulates vertical and lateral water drainage using a logistic function that is intended to capture the breakthrough characteristics of soil water movement. The logistic function provides a simple way (3 parameters) to capture the fast "switching" from low to high flow as water storage within a soil layer approaches field capacity. VELMA offers two alternative methods, direct or indirect, for specifying these parameters. However, because the direct method is still under development, we recommend that users use the indirect method (see *All Parameters 4.4.2*).

4.4.1 – Soil Ks Values Direct Specification

Under development – please use the indirect method described in the next section.

4.4.2 – Soil Ks Values Indirect Specification

We refer to this method as an “indirect" because VELMA internally uses the specified soil Ks parameter values to calculate the actual soil lateral and vertical hydraulic conductivity (Ks) values used during the simulation run. You must specify values for 3 parameters for each soil type (in this example, “SoilWedge1” refers to a particular soil type / map unit).



soil

urfaceK)|(^ks.*Decay)

4.4.2 Soil Ks Values Indirect Specification

Group	Item ⏶	Parameter	Value
soil	SoilWedge1	ksLateralExponentialDecayFactor	0.002
soil	SoilWedge1	ksVerticalExponentialDecayFactor	0.0015
soil	SoilWedge1	surfaceKs	500

<u>Parameter Definitions</u>

`surfaceKs`	saturated hydraulic conductivity (Ks, mm/day) at the soil surface. For equations and parameterization information see Appendix 6.1 (Abdelnour et al. 2011).
`ksVerticalExponentialDecayFactor`	vertical Ks exponential decay factor (unitless) controlling the rate of decrease in vertical flow with depth. For equations and parameterization information see Appendix 6.1 (Abdelnour et al. 2011).
`ksLateralExponentialDecayFactor`	lateral Ks exponential decay factor (unitless) controlling the exponential rate of decrease in lateral flow with depth. . For equations and parameterization information see Appendix 6.1 (Abdelnour et al. 2011).

<u>Calibration Notes</u>

As a starting point, we recommend using the texture-specific values listed in Table 1 for “Surface Ks” and other physical characteristics of soil. Note that all the values in Table 1can vary significantly as a result of compaction, presence of hard pans, and other factors.

Based on the parameter values specified, VELMA computes the Ks lateral and Ks vertical flows (mm/day) per each soil layer as

$$\texttt{ksLateral[ilayer] = surfaceKs} * e^{-ksLateralExponentialDecayFactor * \sum layers[iLayer]*0.5}$$

$$\texttt{ksVertial[ilayer] = surfaceKs} * e^{-ksVerticalExponentialDecayFactor * \sum layers[iLayer]*0.5}$$

Iterative calibration of the user-specified parameter values is generally required to obtain a good fit between simulated and observed stream discharge (mm/day) at the watershed outlet. At the end of each simulation, VELMA calculates a Nash-Sutcliffe efficiency coefficient to assess the goodness-of-fit and reports the results in a simulation output file named “NashSutcliffeCoefficients.txt”.

See http://en.wikipedia.org/wiki/Nash%E2%80%93Sutcliffe_model_efficiency_coefficient for a general overview of the Nash-Sutcliffe efficiency coefficient and its application. Quoting that website:

> “Nash–Sutcliffe efficiencies can range from −∞ to 1. An efficiency of 1 (E = 1) corresponds to a perfect match of modeled discharge to the observed data. An efficiency of 0 (E = 0) indicates that the model predictions are as accurate as the mean of the observed data, whereas an efficiency less than zero (E < 0) occurs when the observed mean is a better predictor than the model.”

In general, a Nash-Sutcliffe coefficient of 0.7 is considered a good fit between simulated and observed discharge. A value of 0.8 or greater is excellent, particularly for spatially-distributed ecohydrological models like VELMA that simulate the processes and flow paths by which water arrives at the watershed outlet (empirical models do not have this requirement).

Advanced users can consult Appendix 6.1 (Abdelnour et al. 2011) for additional details concerning soil Ks equations, parameters and calibration.

4.5 – Soil Other Hydrologic Parameters

VELMA simulates the effect of several soil physical properties on water retention, rates of drainage and other hydrological processes. These physical properties and their associated parameter names are

- Soil porosity (parameter name = `porosityFraction`): the fraction of void space (v/v) in a soil.
- Soil field capacity (parameter name = `fieldCapacity`): the amount of soil moisture (v/v) held in soil after excess water has drained away and the rate of downward movement has materially decreased
- Wilt Point (parameter name = `wiltPoint`): the minimal point of soil moisture (v/v) the plant requires not to wilt.
- Bulk density (parameter name = `bulkDensity`): the dry weight of soil per unit volume of soil (g/cm^3).

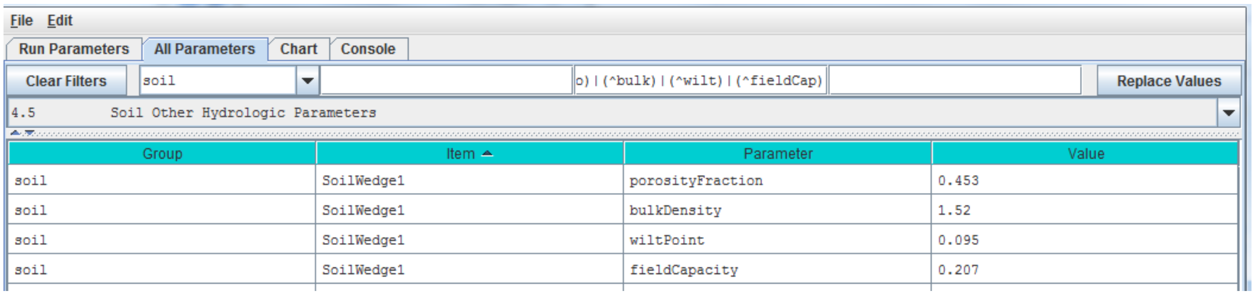
The following table provides typical values for these parameters for 11 soil texture classes (surface Ks, described above in All Parameters 4.4, is included here as well). The values listed are offered here as a starting point for model initialization. However, users should be aware that these values can vary dramatically as a result of compaction, presence of hard pans, and other factors.

Table 1. Physical characteristics of major soil texture classes (Pan et al. 2009, VELMA v1.0 user manual)

Soil Texture	Porosity (v/v)	Field Capacity (v/v)	Wilt Point (v/v)	Bulk Density (g/cm^3)	Surface Ks (mm/day)
sand	0.437	0.091	0.033	1.65	5040
loamy sand	0.437	0.125	0.055	1.65	1466
sandy loam	0.453	0.207	0.095	1.52	622
loam	0.463	0.27	0.117	1.42	317
silt loam	0.501	0.33	0.133	1.42	163
sandy clay loam	0.398	0.255	0.148	1.4	103
clay loam	0.464	0.318	0.197	1.31	55
silty clay loam	0.471	0.366	0.208	1.27	36
sandy clay	0.43	0.339	0.239	1.32	29
silty clay	0.479	0.387	0.25	1.22	22
clay	0.475	0.396	0.272	1.21	14

For each soil type in a watershed (we recommend just one texture type per soil type / mapping unit), you will need to specify values for porosity, field capacity, wilt point and bulk density.

Use the All Parameters drop-down menu to select "4.5 Soil Other Hydrologic Parameters (only one of six soil types is shown in the truncated screenshot below):



soil

o) | (^bulk) | (^wilt) | (^fieldCap)

4.5 Soil Other Hydrologic Parameters

Group	Item	Parameter	Value
soil	SoilWedge1	porosityFraction	0.453
soil	SoilWedge1	bulkDensity	1.52
soil	SoilWedge1	wiltPoint	0.095
soil	SoilWedge1	fieldCapacity	0.207

IMPORTANT INFORMATION FOR SECTIONS 5.0 – 25.0 OF THE ALL PARAMETERS TOC

(link to All Parameters TOC)

VELMA version 2.0 simulates plant biomass and detritus much differently than VELMA version 1.0. Version 1.0 used a simplified Plant Soil Model (PSM, Stieglitz et al. 2006) to simulate plant biomass in each cell in aggregate – that is, leaves, stems and roots were lumped together in a single biomass pool (Abdelnour et al. 2011). This lumped approach greatly simplified the structure and application of the model. Version 1.0 successfully simulates important aspects of catchment hydrology and biogeochemistry (e.g., Abdelnour et al. 2011, 2013), but it does not simulate effects associated with phenological changes and disturbances that disproportionately affect specific plant tissues – for example, leaf growth and senescence, defoliation events, selective removal of high or low C/N tissues via harvest or grazing, etc. These effects can have important implications for assessments involving water quality and quantity, carbon sequestration and other ecosystem services.

To address such effects, VELMA version 2.0 simulates nitrogen pools and fluxes for four plant tissue types: leaves, aboveground stems, belowground stems (>2 mm diameter), and fine roots (< 2 mm diameter). This new Leaf-Stem-Root (LSR) submodel includes live (biomass) and dead (detritus) pools for each of the four plant tissue types.

A carbon/nitrogen (C/N) ratio must be specified for each plant tissue type so that VELMA can estimate carbon pools and fluxes corresponding to modeled nitrogen pools and fluxes. Although the model parameterization allows different C/N values for the biomass and detritus pools of the same tissue type, at present, the plant biomass and detritus pools for each tissue type must have the same C/N ratio to maintain mass balance of C and N pools. Specification of distinct C/N ratios per tissue type will be addressed in future versions.

Several other features have been built into the LSR submodel.

1) A new nonlinear mortality subroutine simulates the loss of live plant biomass to detritus for all four tissues. See All Parameters section 12.0 for details. This replaces the mortality subroutine in VELMA version 1.0 described in Appendix 1, section A1.2.3.

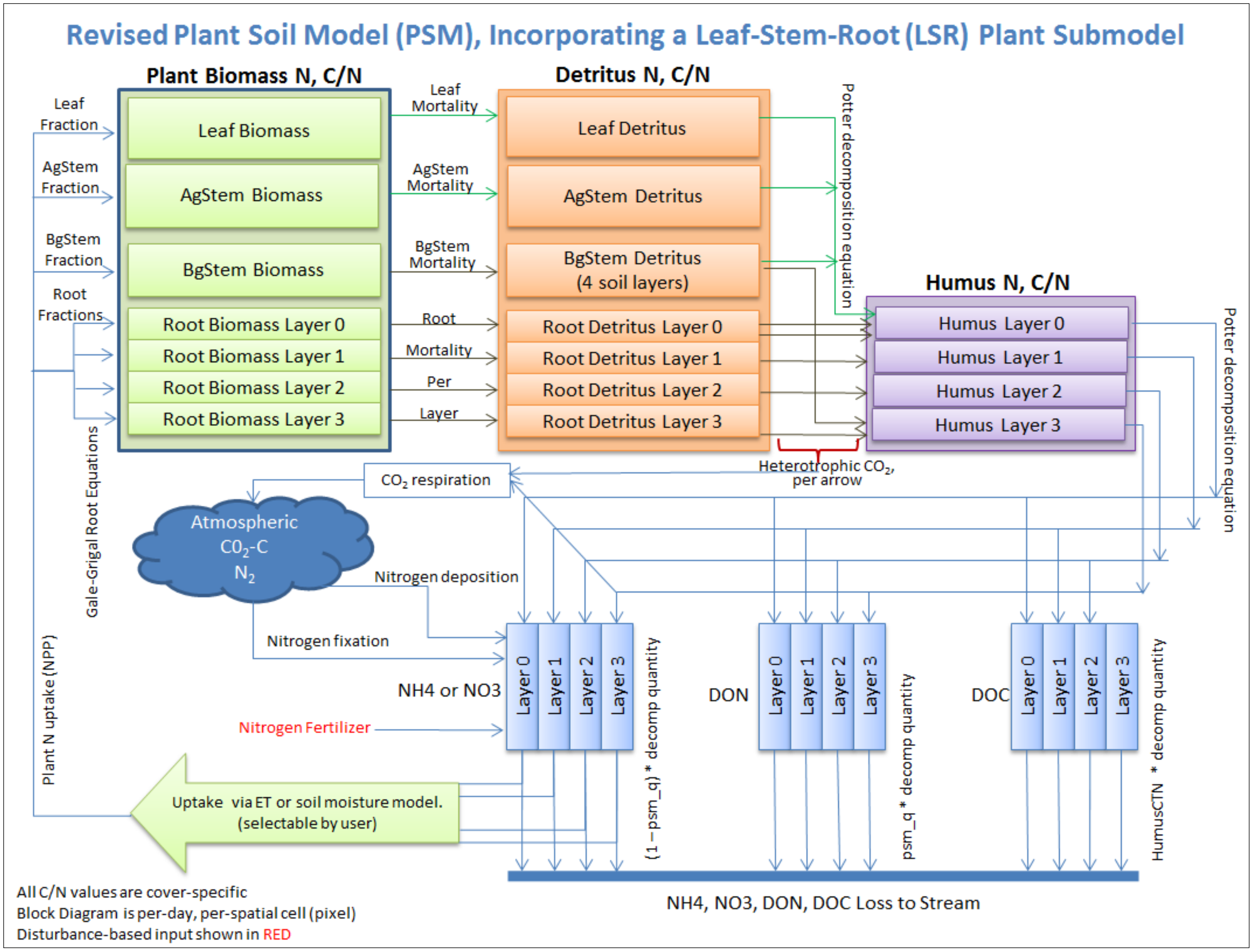
2) A modified version of the decomposition subroutine of Potter et al. (1993) is used to decompose detritus (leaf, stem and root) to humus (unidentifiable dead material). See All Parameters section 13.0 for details. This replaces the decomposition subroutine in VELMA version 1.0 described in Appendix 1, section A1.2.

3) The “ET recovery” subroutine (Abdelnour et al. 2011) has been revised to use leaf biomass, rather than stand age, to simulate effects of forest regrowth on ET and streamflow. This subroutine can be applied to any cover type. See All Parameters section 13.0 for details. This replaces the decomposition subroutine in VELMA version 1.0 described in Appendix 6.2, section B2.

4) A new subroutine describing age-related declines in transpiration of forest stands has been added. Aboveground plant biomass is used as a proxy for stand age. See All Parameters section 10.3 for details.

The figure on the next page illustrates the structure of the new LSR submodel and its integration into the overall Plant Soil Model. Additional Details on the LSR model structure, equations and parameters can be found in "Appendix 1: Overview of VELMA’s Leaf-Stem-Root (LSR) Plant Biomass Submodel”. We recommend reviewing this material before proceeding further, focusing on the LSR model structure and equations describing the various LSR nitrogen and carbon pools and fluxes.

Figure 1.



The User Manual package includes slides for a seminar describing the effectiveness of riparian buffers for protecting water quality in the Chesapeake Bay (please contact mckane.bob@epa.gov if you have questions):

Filename: McKane_VELMA overview_OSU 2April2014_final.pptx

Folder location: VELMA Model\Supporting Documents\VELMA Pubs & Talks

5.0 – Cover Types (link to All Parameters TOC)

VELMA version 2.0 includes two major changes to the vegetation submodel:

(1) Multiple cover types (forest, grassland, agricultural crops, etc.) within a watershed can now be modeled. The version 1.0 convention of one cover type per grid cell is maintained.

(2) Plant biomass for each cover type is modeled using four plant tissues – leaves, aboveground stems, belowground stems, and roots – rather than as a single biomass pool.

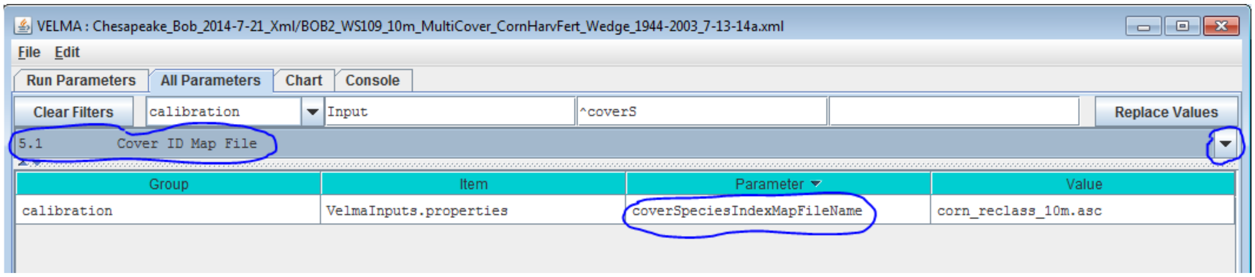
Sections 5.1 – 5.3, below, discuss procedures for initializing a simulator configuration for one or more cover types.

Sections 5.4 through 5.7, below, discuss procedures for initializing a simulator configuration for the four plant biomass pools.

5.1 – Cover ID Map File

The VELMA program requires a set of Cover Species parameters for each Cover ID number that occurs in the Cover ID Map File. Each Cover ID corresponds to a distinct, user-defined Cover Species (cover type) – for example, hardwood forest, grassland, corn cropland, and so forth. This section describes the Cover ID map file and associated parameter specifications.

The Cover ID Map File is specified by the *coverSpeciesIndexMapFileName* parameter. You can view and set it by selecting "All Parameters 5.1 Cover ID Map File" from the All Parameters outline drop-down menu:



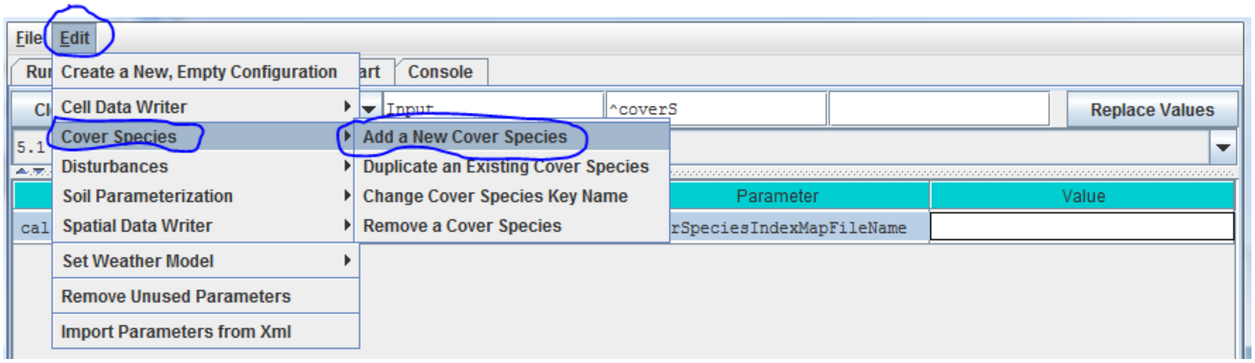
Group	Item	Parameter	Value
calibration	VelmaInputs.properties	coverSpeciesIndexMapFileName	corn_reclass_10m.asc

The file specified for the *coverSpeciesIndexMapFileName* is assumed to be a grid ASCII (".asc") map file with the same row and column dimensions as the simulation configuration's DEM file. Its contents should be integer values. Each cell's integer value should be the ID number of a cover type.

Suppose the specified cover index map file contains three distinct integers: 5, 7 and 9. For this map, the VELMA simulation will expect the simulation configuration to include 3 Cover Species.

You will need to add a new Cover Species to a simulation configuration using the following steps:

(1) Click Edit → Cover Species → "Add a New Cover Species" menu item. This will cause a pop-up menu to appear (see step 2).



5.3 – Cover IDs and Names

[Note: All Parameters 5.2 – Cover Age Map File has been moved below section 5.3 to reflect the required workflow]

Picking up from All Parameters 5.2, step (2)…

(2) Enter a name and ID number for your new Cover Species.
The name must be unique (i.e. no other Cover Species already specified for this simulation configuration can share the name you specify) and we recommend avoiding whitespace and punctuation characters (e.g. "(" and ")") apart from the underbar ("_") character, which can

be used instead of whitespace.

The ID number must be an integer, not already specified for any existing Cover Species, and must match one of the integers that appears in the Cover ID Map File.

As an example:

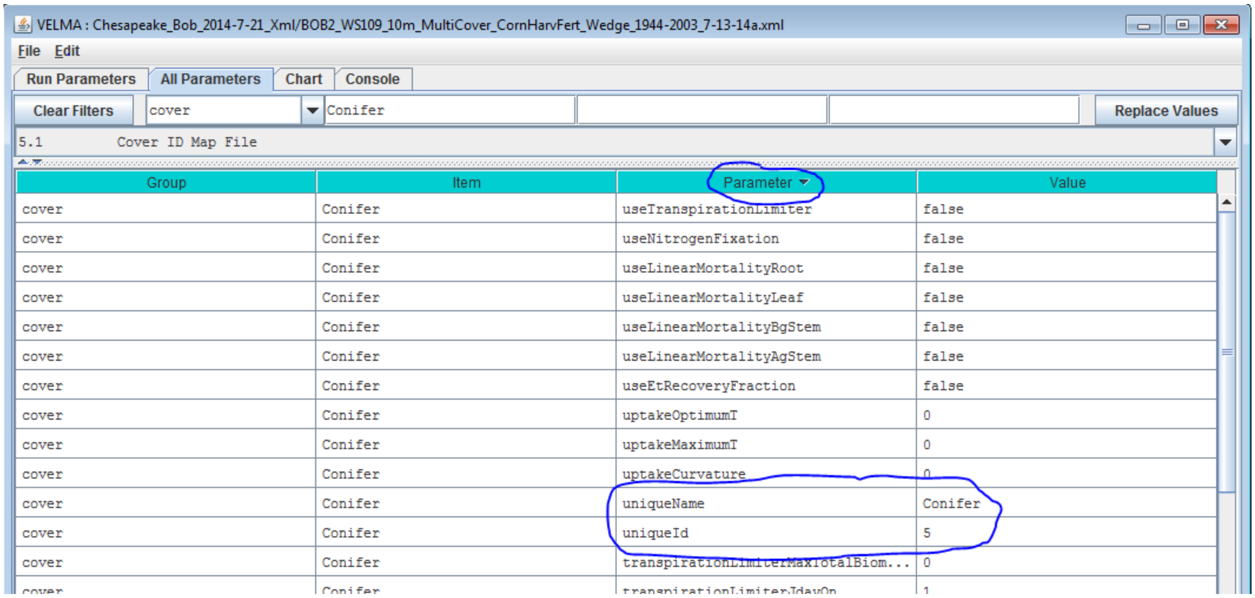
Suppose we are adding the first Cover Species ("5") for the hypothetical Cover ID Map File mentioned above, and that ID "5" identifies cells that contain Conifers.

Acceptable names (guaranteed unique because this is the first Cover Species we're adding) might be: "Conifer" or "Cover_5_Conifer", or something similar.

However, the only acceptable ID value is "5" – because the Unique Cover ID # of this parameterization is what keys it to those particular cells in the Cover ID Map File.

(3) Once you've entered valid name and ID values in the Name and ID dialog, click the OK button.

When you click OK, the VELMA GUI adds a new Cover Species Parameterization to the simulation configuration, and sets the All Parameters tab's filters to display only the parameters of the newly-added Cover Species. Assuming we named our new Cover Species "Conifer", with ID = "5", this is what the All Parameters tab would look like after clicking OK:


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Run Parameters
All Parameters
Chart
Console
Clear Filters
Replace Values

cover | Conifer

5.1 Cover ID Map File

Group	Item	Parameter ▼	Value
cover	Conifer	useTranspirationLimiter	false
cover	Conifer	useNitrogenFixation	false
cover	Conifer	useLinearMortalityRoot	false
cover	Conifer	useLinearMortalityLeaf	false
cover	Conifer	useLinearMortalityBgStem	false
cover	Conifer	useLinearMortalityAgStem	false
cover	Conifer	useEtRecoveryFraction	false
cover	Conifer	uptakeOptimumT	0
cover	Conifer	uptakeMaximumT	0
cover	Conifer	uptakeCurvature	0
cover	Conifer	uniqueName	Conifer
cover	Conifer	uniqueId	5
cover	Conifer	transpirationLimiterMaxTotalBiom...	0
cover	Conifer	transpirationLimiterJdayOn	1

We did one other thing to get the above display: notice that the "Parameter" column is circled, and has a "down-arrow" indicating "descending-sort". After clicking OK to add the new Cover Species, we clicked the "Parameter" header field (twice), which sorts the table of properties on that column.

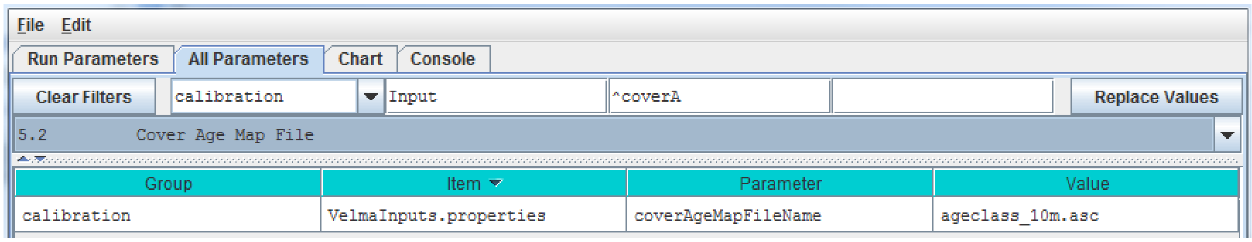
5.2 – Cover Age Map File

[Note: section 5.2 appears after section 5.3 to reflect the required workflow for developing a Cover Age Map file and associated parameter values]

To enable scheduling of disturbance events during a simulation (per All Parameters section 24.0), VELMA needs to keeps track of changes in the age (years) of vegetation from one calendar year to another, or as a result of disturbances that can reset stand age to 0 (e.g., by clearcutting) or to some other age (e.g., by selective thinning of old trees that establishes understory vegetation as the new dominant age class).   Note that cover age maps can be can be applied to cover types other than forests, e.g., rangeland vegetation that may have a more diverse mixt of species than vegetation recovering from a recent fire.

After you have established a Cover ID Map file (section 5.1) and specified Cover IDs and names for each Cover Species (section 5.3, above!), you will need to set up a Cover Age Map file and specify the file name.

The Cover Age Map file name is specified by the `coverAgeMapFileName` parameter. You can view and set it by selecting "All Parameters 5.2 Cover Age Map File" from the All Parameters outline drop-down menu:

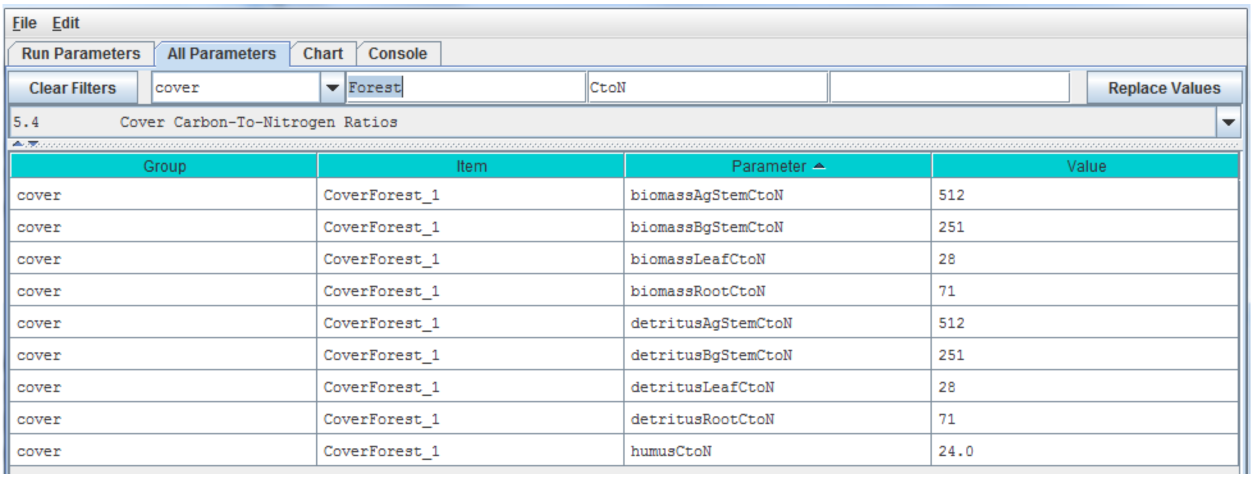


Group	Item	Parameter	Value
calibration	VelmaInputs.properties	coverAgeMapFileName	ageclass_10m.asc

The file specified for the `coverAgeMapFileName` is assumed to be a grid ASCII (".asc") map file with the same row and column dimensions as the simulation configuration's DEM file. Its contents should be integer values reflecting the age of vegetation (Cover Species) occupying each grid cell. Each cell's integer value should be the ID number of a Cover Species. For example, an age of 0 could be assigned to row crops (corn, soybeans, etc.). Cells in a forest landscape might range in age from 0 (newly burned or planted) to many centuries. The VELMA team has used biomass-to-age and tree height-to-age relationships to establish Cover Age Maps.

5.4 – Cover Carbon-To-Nitrogen Ratios

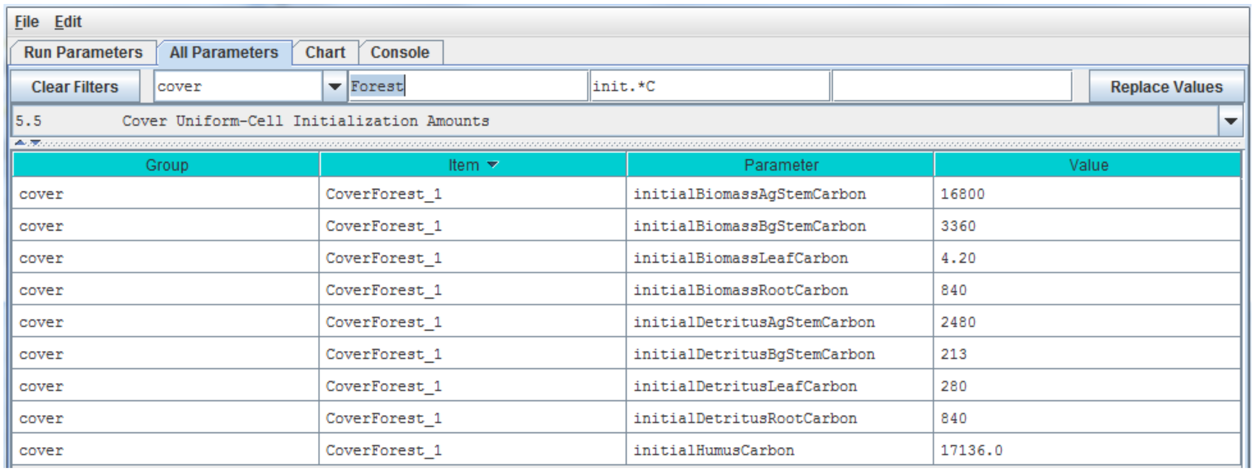
This section is concerned with initializing plant tissue C/N ratios for plant biomass and detritus associated with leaves, aboveground stems (AgStem), belowground stems (BgStem) and roots. Thus, a total of eight C/N (CtoN) ratios must be specified for each cover type. Only one cover type (CoverForest_1) is displayed for the example below. In addition, a C/N ratio must be specified for the humus pool.



Group	Item	Parameter	Value
cover	CoverForest_1	biomassAgStemCtoN	512
cover	CoverForest_1	biomassBgStemCtoN	251
cover	CoverForest_1	biomassLeafCtoN	28
cover	CoverForest_1	biomassRootCtoN	71
cover	CoverForest_1	detritusAgStemCtoN	512
cover	CoverForest_1	detritusBgStemCtoN	251
cover	CoverForest_1	detritusLeafCtoN	28
cover	CoverForest_1	detritusRootCtoN	71
cover	CoverForest_1	humusCtoN	24.0

5.5 – Cover Uniform-Cell Initialization Amounts

Using the *All Parameters* drop-down menu, select "5.5 Cover Uniform-Cell Initialization Amounts". The parameters shown are used to specify initial pool sizes (**g C / m^2**) that VELMA requires at the start of each simulation run. Values must be specified for all plant biomass pools, detritus pools and humus pool associated with each cover type. The example below shows just one cover type, but the same list of 9 parameters repeats for each additional cover type that has been has been for the simulation configuration.



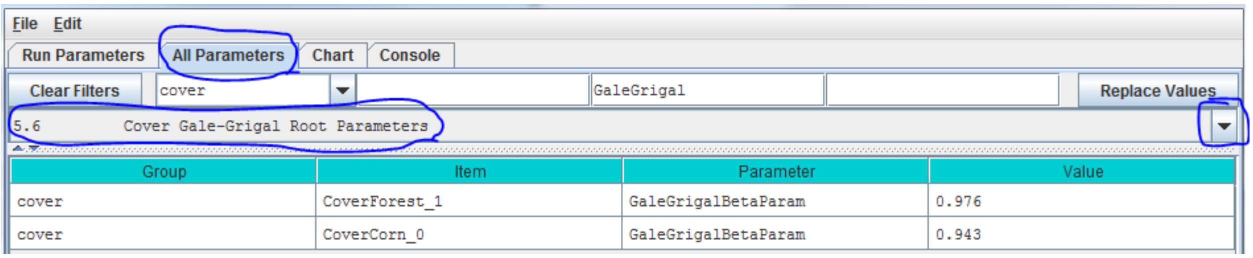
5.5 Cover Uniform-Cell Initialization Amounts

Group	Item	Parameter	Value
cover	CoverForest_1	initialBiomassAgStemCarbon	16800
cover	CoverForest_1	initialBiomassBgStemCarbon	3360
cover	CoverForest_1	initialBiomassLeafCarbon	4.20
cover	CoverForest_1	initialBiomassRootCarbon	840
cover	CoverForest_1	initialDetritusAgStemCarbon	2480
cover	CoverForest_1	initialDetritusBgStemCarbon	213
cover	CoverForest_1	initialDetritusLeafCarbon	280
cover	CoverForest_1	initialDetritusRootCarbon	840
cover	CoverForest_1	initialHumusCarbon	17136.0

“Cover Uniform-Cell Initialization Amounts” refers to the fact that all grid cells assigned to a particular cover type will have the same initial values for plant biomass, detritus and humus. Obviously this method will not work for watersheds where there is significant spatial variability in pool sizes (e.g., where forest stands vary in age). In these situations, you can use All Parameters menu item “24.3 Disturbance Items Spatial Specifiers” to establish spatially variable initialization amounts for any or all pools.

5.6 – Cover Gale-Grigal Root Parameter

Select section 5.6 from the All Parameters drop-down menu to specify Gale-Grigal root parameter values for each cover type in your watershed:



5.6 Cover Gale-Grigal Root Parameters

Group	Item	Parameter	Value
cover	CoverForest_1	GaleGrigalBetaParam	0.976
cover	CoverCorn_0	GaleGrigalBetaParam	0.943

Parameter Definitions

GaleGrigalBetaParam	The Beta term for this cover type’s root vertical distribution. For the root distribution function of Gale Grigal (1987): roots[D] = 1 - B^D, where
	B = GaleGrigalBetaParam value (assigned here)
	D = layer depth

References

Gale, M. R., & Grigal, D. F. (1987). Vertical root distributions of northern tree species in relation to successional status. Canadian Journal of Forest Research, 17(8), 829-834.

Jackson, R. B., Canadell, J., Ehleringer, J. R., Mooney, H. A., Sala, O. E., & Schulze, E. D. (1996). A global analysis of root distributions for terrestrial biomes. Oecologia, 108(3), 389-411.

Calibration Notes:

VELMA uses the method of Gale and Grigal (1987) to vertically distribute a cover type's specified total root biomass. This results in an exponentially decreasing amount of root biomass per soil layer. See Appendix 6.2 section A1.2.6 (Abdelnour et al. 2013) for a description of VELMA's implementation of the Gale-Grigal root distribution function.

Jackson et al. (1996) used the Gale-Grigal method to characterize rooting patterns for terrestrial biomes globally. In the absence of measured data to calibrate `GaleGrigalBetaParam` for your site's cover types, we recommend consulting Jackson et al.'s Table 1and Figure 1:

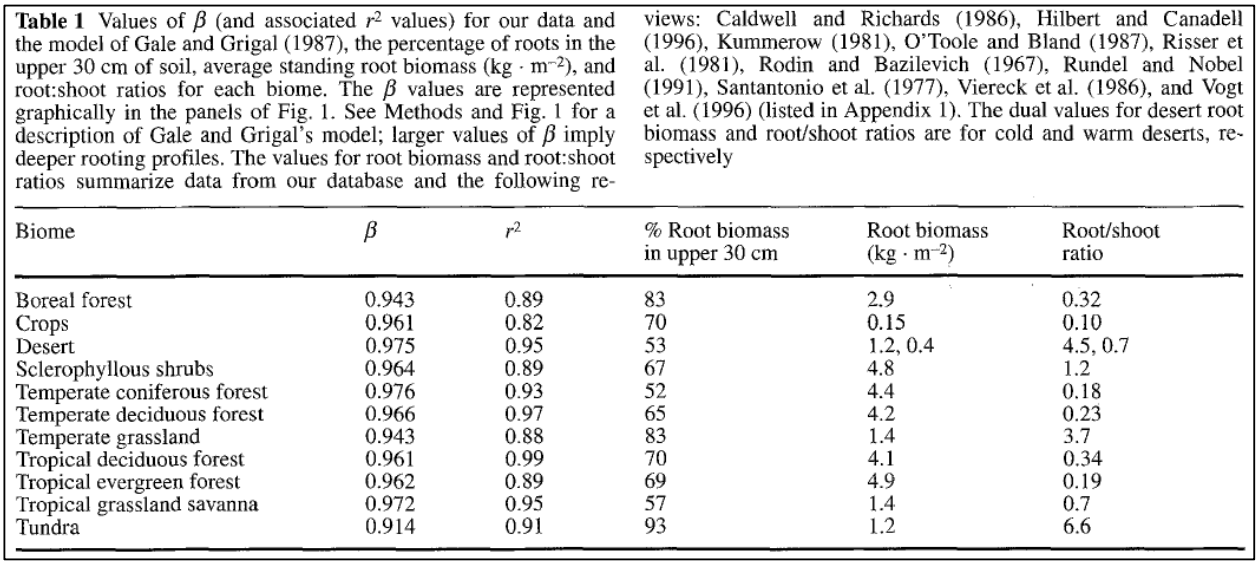


Table 1 Values of β (and associated r^2 values) for our data and the model of Gale and Grigal (1987), the percentage of roots in the upper 30 cm of soil, average standing root biomass (kg · m^{-2}), and root:shoot ratios for each biome. The β values are represented graphically in the panels of Fig. 1. See Methods and Fig. 1 for a description of Gale and Grigal's model; larger values of β imply deeper rooting profiles. The values for root biomass and root:shoot ratios summarize data from our database and the following reviews: Caldwell and Richards (1986), Hilbert and Canadell (1996), Kummerow (1981), O'Toole and Bland (1987), Risser et al. (1981), Rodin and Bazilevich (1967), Rundel and Nobel (1991), Santantonio et al. (1977), Viereck et al. (1986), and Vogt et al. (1996) (listed in Appendix 1). The dual values for desert root biomass and root/shoot ratios are for cold and warm deserts, respectively

Biome	β	r^2	% Root biomass in upper 30 cm	Root biomass (kg · m^{-2})	Root/shoot ratio
Boreal forest	0.943	0.89	83	2.9	0.32
Crops	0.961	0.82	70	0.15	0.10
Desert	0.975	0.95	53	1.2, 0.4	4.5, 0.7
Sclerophyllous shrubs	0.964	0.89	67	4.8	1.2
Temperate coniferous forest	0.976	0.93	52	4.4	0.18
Temperate deciduous forest	0.966	0.97	65	4.2	0.23
Temperate grassland	0.943	0.88	83	1.4	3.7
Tropical deciduous forest	0.961	0.99	70	4.1	0.34
Tropical evergreen forest	0.962	0.89	69	4.9	0.19
Tropical grassland savanna	0.972	0.95	57	1.4	0.7
Tundra	0.914	0.91	93	1.2	6.6

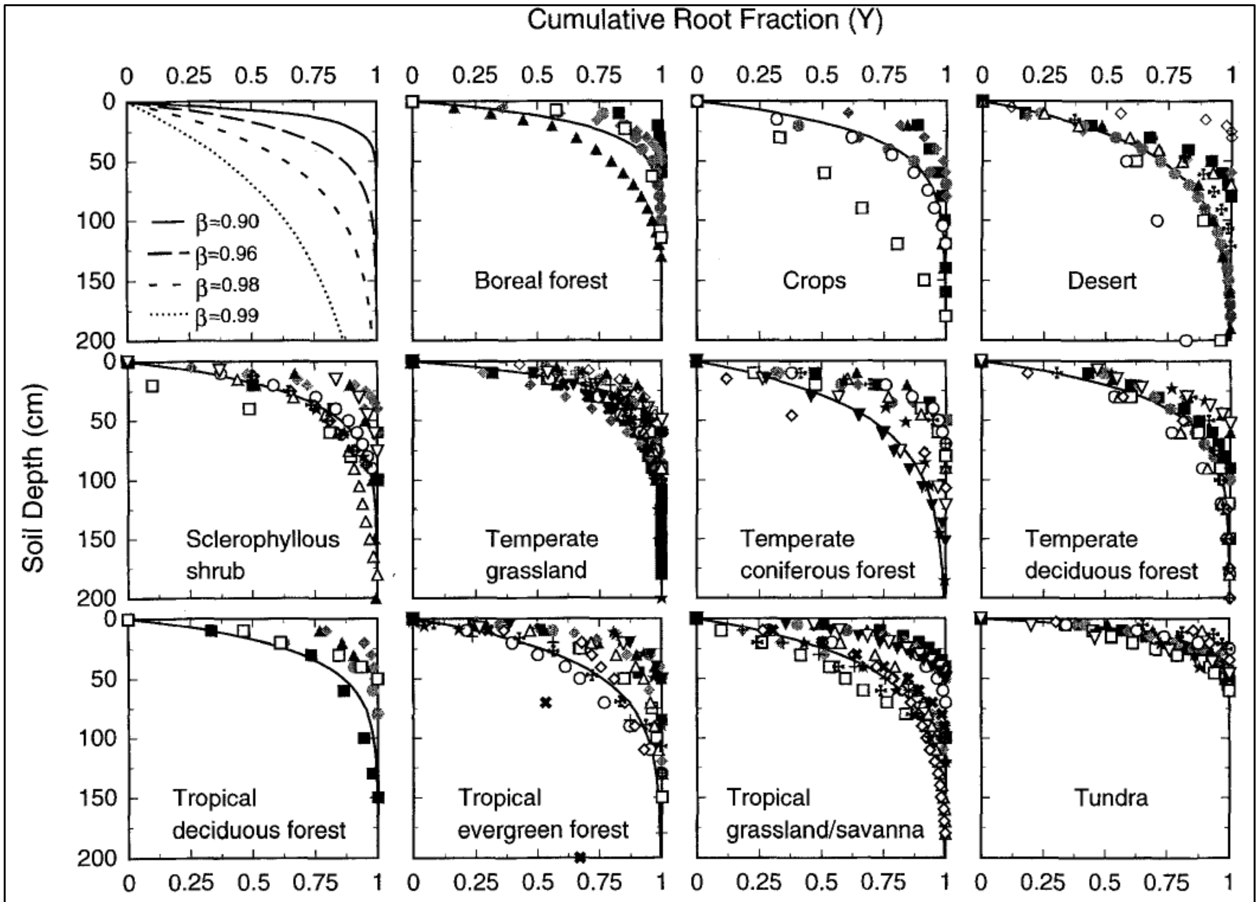
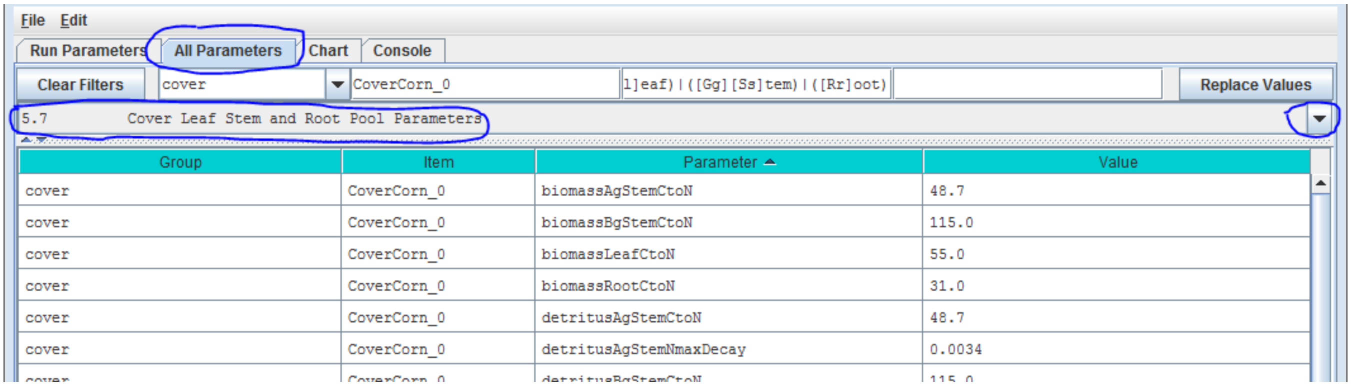


Fig. 1 Cumulative root distribution (cumulative proportion) as a function of soil depth for eleven terrestrial biomes and for the theoretical model of Gale and Grigal (1987). The curve in each biome panel is the least squares fit of β for all studies with data to at least 1 m depth in the soil. The specific β values and the associated r^2 values can be found in Table 1 and the key to the *symbols* in each panel is in Table 2. Gale and Grigal's equation is of the form $y=1-\beta^d$, where *Y* is the cumulative root fraction with depth (a proportion between 0 and 1), *d* is soil depth (in cm), and β is the fitted parameter. Larger values of β imply deeper rooting profiles

5.7 – Cover Leaf Stem and Root Pool Parameters

Select section 5.7 from the All Parameters drop-down menu to specify leaf stem and root pool parameter values for each cover type in your watershed (do this for each cover type):



cover | CoverCorn_0 | l]eaf)|([Gg][Ss]tem)|([Rr]oot)

5.7 Cover Leaf Stem and Root Pool Parameters

Group	Item	Parameter ▲	Value
cover	CoverCorn_0	biomassAgStemCtoN	48.7
cover	CoverCorn_0	biomassBgStemCtoN	115.0
cover	CoverCorn_0	biomassLeafCtoN	55.0
cover	CoverCorn_0	biomassRootCtoN	31.0
cover	CoverCorn_0	detritusAgStemCtoN	48.7
cover	CoverCorn_0	detritusAgStemNmaxDecay	0.0034
cover	CoverCorn_0	detritusBgStemCtoN	115.0

Parameters in this section cover the Leaf, Stem and Root pool C/N ratios, initial values, mortality parameterizations and uptake fractions. Because each biomass or detritus leaf, stem and root pool has its own set of parameters for these subjects, this list of parameters is relatively extensive.

Parameter Definitions

`biomassLeafCtoN` The Carbon-to-Nitrogen ratio of the cover type's Leaf biomass pool.

`biomassAgStemCtoN` The Carbon-to-Nitrogen ratio of the cover type's AgStem biomass pool.

`biomassBgStemCtoN` The Carbon-to-Nitrogen ratio of the cover type's BgStem biomass pool.

`biomassRootCtoN` The Carbon-to-Nitrogen ratio of the cover type's Root biomass pool. (All layers of this pool use the same CtoN ratio value.)

`detritusLeafCtoN` The Carbon-to-Nitrogen ratio of the cover type's Leaf detritus pool.

`detritusAgStemCtoN` The Carbon-to-Nitrogen ratio of the cover type's AgStem detritus pool.

`detritusBgStemCtoN` The Carbon-to-Nitrogen ratio of the cover type's BgStem detritus pool. (All layers of this pool use the same CtoN ratio value.)

`detritusRootCtoN` The Carbon-to-Nitrogen ratio of the cover type's Root detritus pool. (All layers of this pool use the same CtoN ratio value.)

`detritusLeafNmaxDecay` Maximum decay rate for Potter decomposition of Leaf detritus pool's Nitrogen amount.

`detritusAgStemNmaxDecay` Maximum decay rate for Potter decomposition of AgStem detritus pool's Nitrogen amount.

`detritusBgStemNmaxDecay` Maximum decay rate for Potter decomposition of BgStem detritus pool's Nitrogen amount.

`detritusRootNmaxDecay` Maximum decay rate for Potter decomposition of Root detritus pool's Nitrogen amount.

`initialBiomassLeafCarbon` The amount of biomass (in grams of Carbon) placed in the Leaf biomass pool of each cell of this cover type at initialization start.

`initialBiomassAgStemCarbon` The amount of biomass (in grams of Carbon) placed in the AgStem biomass pool of each cell of this cover type at initialization start.

`initialBiomassBgStemCarbon` The amount of biomass (in grams of Carbon) placed in the BgStem biomass pool of each cell of this cover type at initialization start.

`initialBiomassRootCarbon` The amount of biomass (in grams of Carbon) placed in the Root biomass pool of each cell of this cover type at initialization start.

`initialDetritusLeafCarbon` The amount of detritus (in grams of Carbon) placed in the Leaf detritus pool of each cell of this cover type at initialization start.

`initialDetritusAgStemCarbon` The amount of detritus (in grams of Carbon) placed in the AgStem detritus pool of each cell of this cover type at initialization start.

`initialDetritusBgStemCarbon` The amount of detritus (in grams of Carbon) placed in the BgStem detritus pool of each cell of this cover type at initialization start.

`initialDetritusRootCarbon` The amount of detritus (in grams of Carbon) placed in the Root detritus pool of each cell of this cover type at initialization start.

`etRecoveryFractionMinimumLeafBiomassC` The minimum amount of leaf biomass (in gC/m^2) required to compute an ET recovery fraction. Whenever leaf biomass is less than this amount, the ET recovery fraction is forced to the `etRecoverFractionMinimumValue`.
See Section 10.2.1 ET Recovery Parameters.

`mortalityLeafAnnualFraction` The fraction of the Leaf biomass Pool annual mortality amount actually subtracted from that pool per day of senescence. The default value for this fraction is 1.0 (i.e. the full per-day amount) and the valid range is [0.0 to 1.0]. Warning! Users unfamiliar with this parameter's behavior should leave it set to the default value of 1.0.

`mortalityLeafDenominatorCoefficient` The mortality equation’s denominator coefficient value for the biomass Leaf pool.

`mortalityLeafExponentialCoefficient` The mortality equation’s exponential coefficient value for the biomass Leaf pool.

`mortalityLeafNumeratorCoefficient` The mortality equation’s numerator coefficient value for biomass Leaf pool.

`mortalityAgStemAnnualFraction` The fraction of the AgStem biomass Pool annual mortality amount actually subtracted from that pool per day of senescence. The default value for this fraction is 1.0 (i.e. the full per-day amount) and the valid range is [0.0 to 1.0]. Warning! Users unfamiliar with this parameter's behavior should leave it set to the default value of 1.0.

`mortalityAgStemDenominatorCoefficient` The mortality equation’s denominator coefficient value for the biomass AgStem pool.

`mortalityAgStemExponentialCoefficient` The mortality equation’s exponential coefficient value for the biomass AgStem pool.

`mortalityAgStemNumeratorCoefficient`	The mortality equation’s numerator coefficient value for the biomass AgStem pool.
`mortalityBgStemAnnualFraction`	The fraction of the BgStem biomass Pool annual mortality amount actually subtracted from that pool per day of senescence. The default value for this fraction is 1.0 (i.e. the full per-day amount) and the valid range is [0.0 to 1.0]. Warning! Users unfamiliar with this parameter's behavior should leave it set to the default value of 1.0.
`mortalityBgStemDenominatorCoefficient`	The mortality equation’s denominator coefficient value for the biomass BgStem pool.
`mortalityBgStemExponentialCoefficient`	The mortality equation’s exponential coefficient value for the biomass BgStem pool.
`mortalityBgStemNumeratorCoefficient`	The mortality equation’s numerator coefficient value for the biomass BgStem pool.
`mortalityRootAnnualFraction`	The fraction of the Leaf biomass Pool annual mortality amount actually subtracted from that pool per day of senescence. The default value for this fraction is 1.0 (i.e. the full per-day amount) and the valid range is [0.0 to 1.0]. Warning! Users unfamiliar with this parameter's behavior should leave it set to the default value of 1.0.
`mortalityRootDenominatorCoefficient`	The mortality equation’s denominator coefficient value for the biomass Root pool.
`mortalityRootExponentialCoefficient`	The mortality equation’s exponential coefficient value for the biomass Root pool.
`mortalityRootNumeratorCoefficient`	The mortality equation’s numerator coefficient value for the biomass Root pool.
`useLinearMortalityLeaf`	When true, leaf mortality is computed using a simple linear function. When false (default) mortality is computed via a logistic function.
`useLinearMortalityAgStem`	When true, AgStem mortality is computed using a simple linear function. When false (default) mortality is computed via a logistic function.
`useLinearMortalityBgStem`	When true, BgStem mortality is computed using a simple linear function. When false (default) mortality is computed via a logistic function.
`useLinearMortalityRoot`	When true, Root mortality is computed using a simple linear function. When false (default) mortality is computed via a logistic function.
`nppToBiomassLeafNfraction`	The fraction of total daily NPP for a cell of this cover type allotted to the biomass Leaf pool.
`nppToBiomassRootNfraction`	The fraction of total daily NPP for a cell of this cover type allotted to the biomass Root pool.
`nppToBiomassAgStemNfraction`	The fraction of daily NPP ALOTTED TO STEM POOLS for a cell of this cover type allotted to the

biomass AgStem pool. The remaining NPP allotted to stem pools goes to the biomass BgStem pool. NOTE: Total daily NPP is reduced by the fractions for leaf and root pools first. Any remainder is the daily NPP allotted to stem pools.

Calibration Notes

At this stage (section 5.7) of your simulation configuration, don't try to specify values for all the parameters listed above. For now, just focus on the initial C and N pools for plant biomass and detritus (highlighted in yellow, above). To specify initial C and N stocks (g/m^2) for these pools, we recommend that you consult Appendix 1 for an overview of VELMA's LSR submodel and it C and N pools and fluxes. Filling in C and N values for the various pools shown would be a good start. Remember to document data sources as you go.

Methods for specifying parameter values for NPP (plant uptake) allocation, plant biomass mortality, and detritus decay (decomposition) rates, and NPP are discussed in subsequent sections:

- N uptake (NPP): Section 11.0
- Mortality: Section 12.0
- Decomposition: Section 13.0

5.7.1 – Cover Leaf Pool Parameter

Select section 5.7.1 from the All Parameters drop-down menu to specify cover-specific parameter values for leaf biomass and detritus. See 5.7 above for the parameter descriptions.

5.7.2 – Cover Above-Ground Stem Pool Parameters

Select section 5.7.1 from the All Parameters drop-down menu to specify cover-specific parameter values for aboveground stem biomass and detritus. See 5.7 above for the parameter descriptions.

5.7.3 – Cover Below-Ground Stem Pool Parameters

Select section 5.7.1 from the All Parameters drop-down menu to specify cover-specific parameter values for belowground stem biomass and detritus. See 5.7 above for the parameter descriptions.

5.7.4 – Cover Root Pool Parameters

Select section 5.7.1 from the All Parameters drop-down menu to specify cover-specific parameter values for root biomass and detritus. See 5.7 above for the parameter descriptions.

6.0 – Weather Model (link to All Parameters TOC)

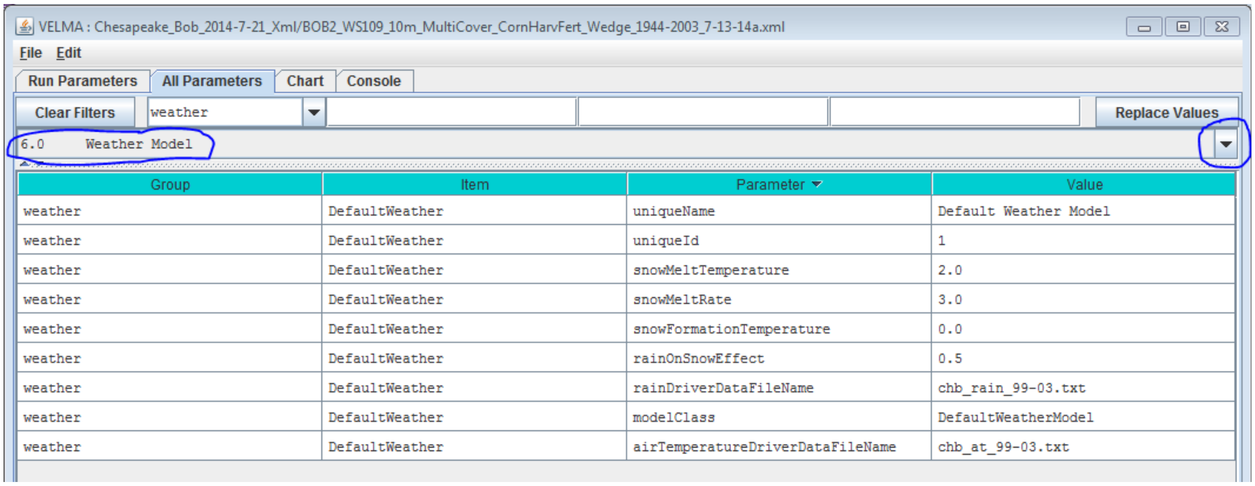
The Default Weather Model provides uniform precipitation and air temperature values for all cells of a simulation's delineated watershed for each day of a simulation run, and includes a simple snow model.

A Default Weather Model is Provided Automatically in New Simulation Configurations

When you create a new simulation configuration in the VELMA GUI (Edit → "Create a New, Empty Configuration"), it automatically contains parameters for the default weather model. This weather model's `modelClass`, `uniqueName` and `uniqueId` parameters are preset to valid, default values. You may change those values if you wish, but there is no need to do so and we advise you to leave them alone. You must provide the model with the names of data files for precipitation and air temperature driver data, and parameterize the snow model.

You can focus the All Parameters tab's parameters table to various aspects of the simulation configuration's weather model by selecting them in the tab's drop-down selector.

To view the complete core weather model parameterization, select "6.0 Weather Model":


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File Edit
Run Parameters
All Parameters
Chart
Console
Clear Filters
Replace Values

weather

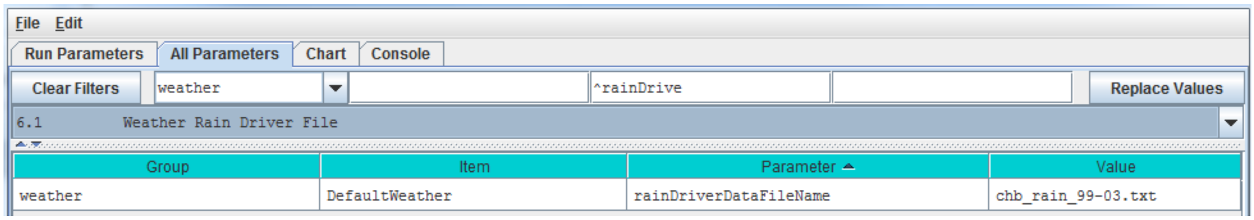
6.0 Weather Model

Group	Item	Parameter	Value
weather	DefaultWeather	uniqueName	Default Weather Model
weather	DefaultWeather	uniqueId	1
weather	DefaultWeather	snowMeltTemperature	2.0
weather	DefaultWeather	snowMeltRate	3.0
weather	DefaultWeather	snowFormationTemperature	0.0
weather	DefaultWeather	rainOnSnowEffect	0.5
weather	DefaultWeather	rainDriverDataFileName	chb_rain_99-03.txt
weather	DefaultWeather	modelClass	DefaultWeatherModel
weather	DefaultWeather	airTemperatureDriverDataFileName	chb_at_99-03.txt

You Must Provide the Weather Model with Rain and Air Temperature Driver Files. The rain and air temperature driver files share the same format requirements: one floating-point value per line, with exactly one line for each day between the first day of the year specified by the simulation configuration's forcing_start value, and the last day of the year specified by the simulation configuration's forcing_end value. For example, if a forcing start/end range of [2000 to 2001] would require each driver file to have 731 lines of data (corresponding to 366 + 365 days per year, accounting for the leap year), with one value per line.

You may omit the path of the driver file names you specify. The Velma simulator then assumes the file's location is specified by the simulation configuration's input data location settings (set the outline selection to "1.0 Input Data Selection" to review or set the input location). You may also provide a fully-qualified path + name for a driver file. If you do, be sure to use "/" (forward-slash) characters as path-separators.

Select "6.1 Weather Rain Driver File" to focus the parameters table on the weather model's `rainDriverDataFileName` parameter.



weather

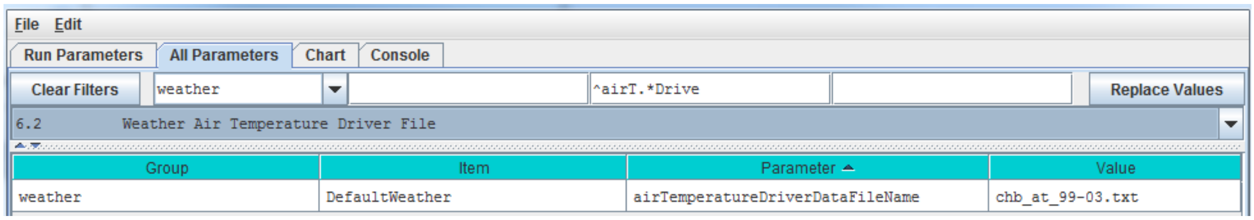
^rainDrive

6.1 Weather Rain Driver File

Group	Item	Parameter	Value
weather	DefaultWeather	rainDriverDataFileName	chb_rain_99-03.txt

Each value in this file represents a day's precipitation (in millimeters) at each cell in simulation's the delineated watershed. Because of the snow model, some of this precipitation may be rain, and some may be snow. The parameterization of the snow model determines the amounts of both.

Select "6.2 Weather Air Temperature Driver File" to focus the parameters table on the weather model's `airTemperatureDriverFileName`.



weather

^airT.*Drive

6.2 Weather Air Temperature Driver File

Group	Item	Parameter ▲	Value
weather	DefaultWeather	airTemperatureDriverDataFileName	chb_at_99-03.txt

Each value in this file represents a day's average air temperature (in degrees Centigrade) at each cell in the simulation's delineated watershed.

The Snow Model Determines the Rain / Snow Mixture for the Precipitation Driver Value

The four parameters of the snow model determine how much of a given day's precipitation falls as rain or as snow, and how fast snow on the ground is converted to surface water.

`snowFormationTemperature` The air temperature (in degrees C) below which the day's precipitation is counted as snow. At and above this temperature, it is counted as rain.

`snowMeltTemperature` The temperature above which any available snow begins to melt.

`snowMeltRate` The rate at which melting snow becomes surface water.

`rainOnSnowEffect` Used in conjunction with the `snowMeltRate` to determine the amount of snow that melts into surface water.

A Summary Description of the Snow Model's Behavior

For each day of a simulation run, the daily air temperature and precipitation values are taken together with the above snow model parameters and daily values are calculated for the following: `Rain, Snow, Snow Depth` and `Snow Melt`.

When air temperature is at or above the `snowFormationTemperature`, all the day's precipitation is counted as rain and becomes the day's Rain value. If the air temperature is also greater than the `snowMeltTemperature`, the amount of snow that melts is calculated using the following equation:

```
        snowMeltRate * (airT - snowMeltTemperature) + rainOnSnowEffect * rain
```

where `airT` and `rain` are the day's air temperature and precipitation values respectively.

If there has been snow deposited into the Snow Depth value on previous simulation days (see below) the amount of snow melted is "deducted" from that value and becomes additional "rain" for the day. If the melted amount that is computed exceeds the amount of snow depth, the entire amount (but no more) is "deducted".

When air temperature is below the `snowFormationTemperature` parameter, all the day's precipitation is counted as snow and becomes the day's Snow value. The day's Snow mount is also added to the Snow Depth value.

6.3.1 – Weather Model Coefficients Files (Only Spatial Weather)

VELMA's default weather model (described under section 6.0 – 6.2) does not address spatial variations in temperature and precipitation associated with topographic features – elevation, slope, aspect and cold air drainages.

VELMA’s optional ***Spatial Weather Model*** provides differing precipitation and air temperature values for each cell of a simulation’s delineated watershed for each day of a simulation run. It includes the same simple snow model as the Default Weather Model.

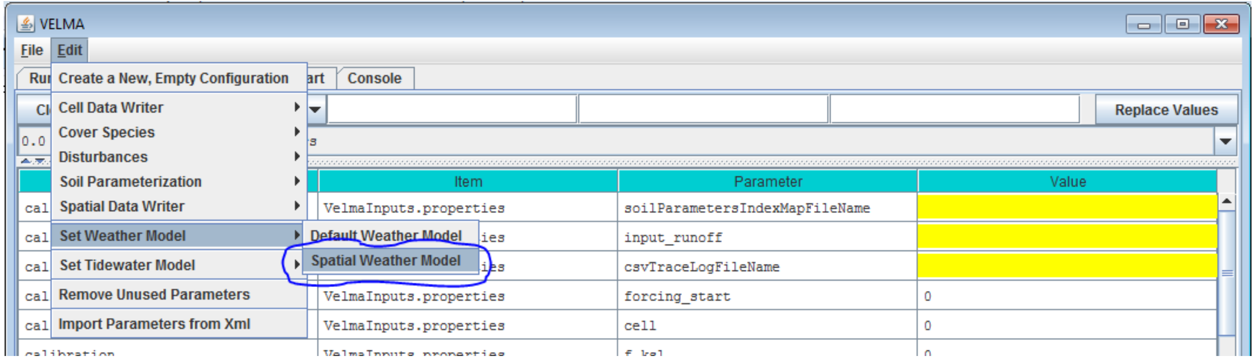
CAUTION!
The Spatial Weather Model is still somewhat experimental and undergoing testing and refinement.

Also, it requires more effort to configure/parameterize than the Default Weather Model.

For both of these reasons, we currently recommend using only the Default Weather Model unless you *must* have the more complex behavior that the Spatial Weather Model provides, e.g, for mountainous terrain.

You Must Explicitly Specify Activation of the Spatial Weather Model

When you create a new simulation configuration, the VELMA GUI automatically adds a Default Weather Model parameters group to it. If you wish to use the Spatial Weather Model instead, Click the Edit → “Set Weather Model → “Spatial Weather Model” menu item:



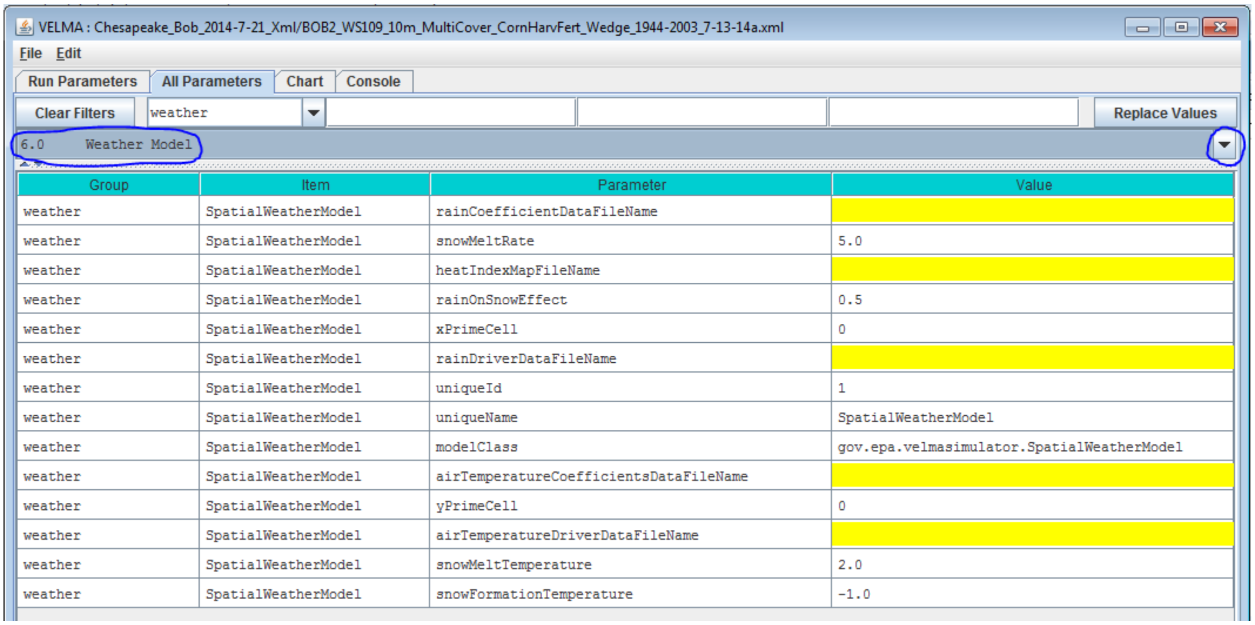
Item	Parameter	Value
VelmaInputs.properties	soilParametersIndexMapFileName	
ies	input_runoff	
ies	csvTraceLogFileName	
VelmaInputs.properties	forcing_start	0
VelmaInputs.properties	cell	0

When you set the weather model to “Spatial Weather Model”, the VELMA GUI removes any pre-existing weather model parameters and adds the parameters group for the Spatial Weather Model.

You can focus the All Parameters tab’s parameters table to various aspects of the simulation configuration’s weather model by selecting them in the tab’s outline drop-down selector.

The VELMA GUI automatically provides valid default values for the Spatial Weather Model’s `uniqueId`, `uniqueName` and `modelClass` parameters. You may leave these as they are initially set.

To view the complete core weather model parameterization, select “6.0 Weather Model”:



VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml

weather

6.0 Weather Model

Group	Item	Parameter	Value
weather	SpatialWeatherModel	rainCoefficientDataFileName	
weather	SpatialWeatherModel	snowMeltRate	5.0
weather	SpatialWeatherModel	heatIndexMapFileName	
weather	SpatialWeatherModel	rainOnSnowEffect	0.5
weather	SpatialWeatherModel	xPrimeCell	0
weather	SpatialWeatherModel	rainDriverDataFileName	
weather	SpatialWeatherModel	uniqueId	1
weather	SpatialWeatherModel	uniqueName	SpatialWeatherModel
weather	SpatialWeatherModel	modelClass	gov.epa.velmasimulator.SpatialWeatherModel
weather	SpatialWeatherModel	airTemperatureCoefficientsDataFileName	
weather	SpatialWeatherModel	yPrimeCell	0
weather	SpatialWeatherModel	airTemperatureDriverDataFileName	
weather	SpatialWeatherModel	snowMeltTemperature	2.0
weather	SpatialWeatherModel	snowFormationTemperature	-1.0

You Must Provide the Weather Model with Rain and Air Temperature Driver Files

The Spatial Weather Model derives daily rain and air temperature values for every cell in the simulation watershed by applying functions to the daily rain and air temperature values of one specific cell (the "Primary" or "Prime" Cell) within the watershed. You must provide the data file names for rain and air temperature data observed for this Prime Cell, as well as its x-and-y coordinates as part of the Spatial Weather Model's parameterization.

Set the `xPrimeCell` and `yPrimeCell` parameter's values to the Prime Cell's x (column) and y (row) location.

Set the `rainDriverDataFileName` and `airTemperatureDriverDataFileName` to the file names of the precipitation and air temperature files, respectively.

You may omit the path of the driver file names you specify. The Velma simulator then assumes the file's location is specified by the simulation configuration's input data location settings (set the outline selection to "1.0 Input Data Selection" to review or set the input location). You may also provide a fully-qualified path + name for a driver file. If you do, be sure to use "/" (forward-slash) characters as path-separators.

The rain and air temperature driver files share the same format requirements: one floating-point value per line, with exactly one line for each day between the first day of the year specified by the simulation configuration's `forcing_start` value, and the last day of the year specified by the simulation configuration's `forcing_end` value. For example, if a forcing start/end range of [2000 to 2001] would require each driver file to have 731 lines of data, with one value per line.

You Must Provide Files of Coefficients for Precipitation and Air Temperature Calculations

The Spatial Weather Model calculates the day's precipitation and air temperature values for a given cell location by applying a precipitation or air temperature function to the day's driver precipitation or air temperature value. These functions map/modify the "seed" value at the Prime Cell to derive a value for the other cells. The functions take into account the cell's elevation, flow accumulation and heat index values, and the equations involved require per-month coefficients for each term of their equations.

You must provide the file names (or optionally full path + names) of one file for rain (precipitation) equation coefficients and one file of air temperature coefficients.

NOTE

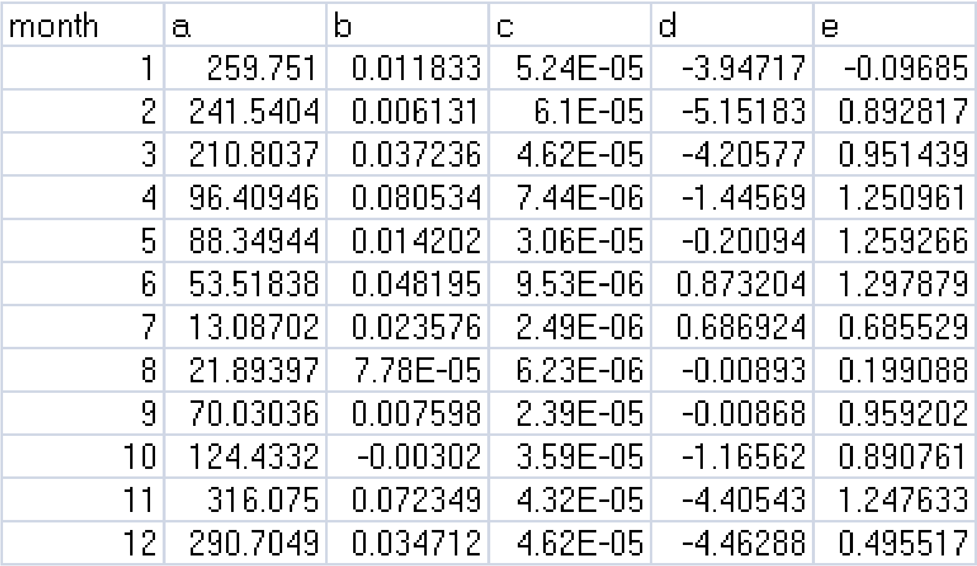
Determining the values for precipitation and air temperature coefficients requires detailed knowledge of the equations defining the Spatial Weather Model's behavior.

For additional information concerning theory and procedures for developing statistical coefficients for VELMA's spatial climate model see Appendix 5: Generating daily temperature and precipitation grids for running VELMA: development of statistical models based on monthly PRISM data.

The Precipitation Coefficients File

A comma-separated values (.csv) file containing a table of coefficients for the spatial weather model's precipitation equation. The file must contain 13 rows of 6 comma-separated values (columns) each.

The first row is a header row, and the remaining 12 rows provide values for the equation's a, b, c, d and e coefficients for each month of the year.
Here is an example table of values:



month	a	b	c	d	e
1	259.751	0.011833	5.24E-05	-3.94717	-0.09685
2	241.5404	0.006131	6.1E-05	-5.15183	0.892817
3	210.8037	0.037236	4.62E-05	-4.20577	0.951439
4	96.40946	0.080534	7.44E-06	-1.44569	1.250961
5	88.34944	0.014202	3.06E-05	-0.20094	1.259266
6	53.51838	0.048195	9.53E-06	0.873204	1.297879
7	13.08702	0.023576	2.49E-06	0.686924	0.685529
8	21.89397	7.78E-05	6.23E-06	-0.00893	0.199088
9	70.03036	0.007598	2.39E-05	-0.00868	0.959202
10	124.4332	-0.00302	3.59E-05	-1.16562	0.890761
11	316.075	0.072349	4.32E-05	-4.40543	1.247633
12	290.7049	0.034712	4.62E-05	-4.46288	0.495517

and here are the lines from the corresponding .csv data file:

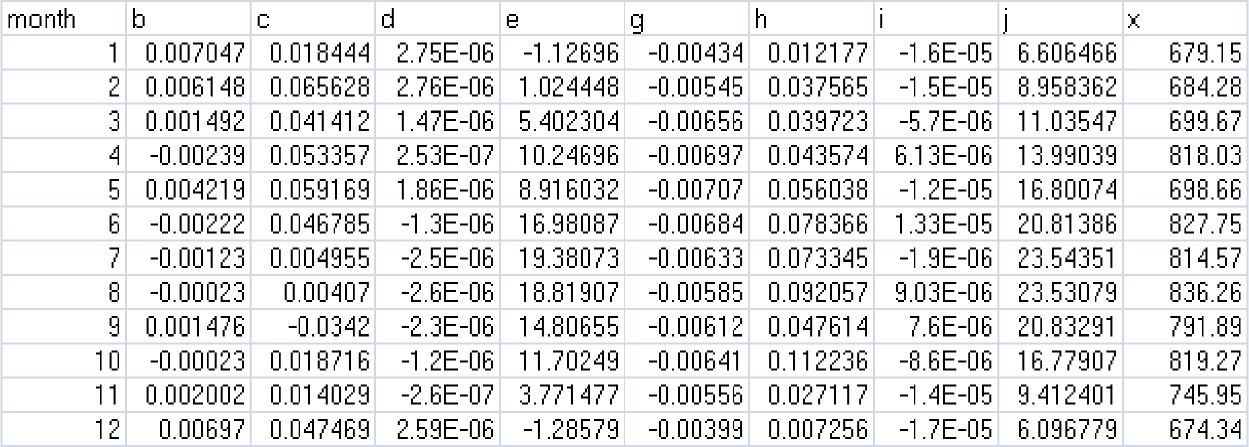
```
month,a,b,c,d,e
1,259.751,0.011833,5.24E-05,-3.94717,-0.09685
2,241.5404,0.006131,6.10E-05,-5.15183,0.892817
3,210.8037,0.037236,4.62E-05,-4.20577,0.951439
4,96.40946,0.080534,7.44E-06,-1.44569,1.250961
5,88.34944,0.014202,3.06E-05,-0.20094,1.259266
6,53.51838,0.048195,9.53E-06,0.873204,1.297879
7,13.08702,0.023576,2.49E-06,0.686924,0.685529
8,21.89397,7.78E-05,6.23E-06,-0.00893,0.199088
9,70.03036,0.007598,2.39E-05,-0.00868,0.959202
10,124.4332,-0.00302,3.59E-05,-1.16562,0.890761
```

```
11,316.075,0.072349,4.32E-05,-4.40543,1.247633
12,290.7049,0.034712,4.62E-05,-4.46288,0.495517
```

The Air Temperature Coefficients File

A comma-separated values (.csv) file containing a table of coefficients for the spatial weather model's air temperature equation. The file must contain 13 rows of 10 comma-separated values (columns) each.

The first row is a header row, and the remaining 12 rows provide values for the equation's a – j and x coefficients for each month of the year.
Here is an example table of values:



month	b	c	d	e	g	h	i	j	x
1	0.007047	0.018444	2.75E-06	-1.12696	-0.00434	0.012177	-1.6E-05	6.606466	679.15
2	0.006148	0.065628	2.76E-06	1.024448	-0.00545	0.037565	-1.5E-05	8.958362	684.28
3	0.001492	0.041412	1.47E-06	5.402304	-0.00656	0.039723	-5.7E-06	11.03547	699.67
4	-0.00239	0.053357	2.53E-07	10.24696	-0.00697	0.043574	6.13E-06	13.99039	818.03
5	0.004219	0.059169	1.86E-06	8.916032	-0.00707	0.056038	-1.2E-05	16.80074	698.66
6	-0.00222	0.046785	-1.3E-06	16.98087	-0.00684	0.078366	1.33E-05	20.81386	827.75
7	-0.00123	0.004955	-2.5E-06	19.38073	-0.00633	0.073345	-1.9E-06	23.54351	814.57
8	-0.00023	0.00407	-2.6E-06	18.81907	-0.00585	0.092057	9.03E-06	23.53079	836.26
9	0.001476	-0.0342	-2.3E-06	14.80655	-0.00612	0.047614	7.6E-06	20.83291	791.89
10	-0.00023	0.018716	-1.2E-06	11.70249	-0.00641	0.112236	-8.6E-06	16.77907	819.27
11	0.002002	0.014029	-2.6E-07	3.771477	-0.00556	0.027117	-1.4E-05	9.412401	745.95
12	0.00697	0.047469	2.59E-06	-1.28579	-0.00399	0.007256	-1.7E-05	6.096779	674.34

And here are the lines from the corresponding .csv data file:

```
month,b,c,d,e,g,h,i,j,x
1,0.007047,0.018444,2.75E-06,-1.12696,-0.00434,0.012177,-1.60E-05,6.606466,679.15
2,0.006148,0.065628,2.75E-06,1.024448,-0.00545,0.037565,-1.50E-05,8.958362,684.28
3,0.001492,0.041412,1.47E-06,5.402304,-0.00656,0.039723,-5.70E-06,11.03547,699.67
4,-0.00239,0.053357,2.53E-07,10.24696,-0.00697,0.043574,6.13E-06,13.99039,818.03
5,0.004219,0.059169,1.86E-06,8.916032,-0.00707,0.056038,-1.20E-05,16.80074,698.66
6,-0.00222,0.046785,-1.30E-06,16.98087,-0.00684,0.078366,1.33E-05,20.81386,827.75
7,-0.00123,0.004955,-2.50E-06,19.38073,-0.00633,0.073345,-1.90E-06,23.54351,814.57
8,-0.00023,0.00407,-2.60E-06,18.81907,-0.00585,0.092057,9.03E-06,23.53079,836.26
9,0.001476,-0.0342,-2.30E-06,14.80655,-0.00612,0.047614,7.60E-06,20.83291,791.89
10,-0.00023,0.018716,-1.20E-06,11.70249,-0.00641,0.112236,-8.60E-06,16.77907,819.27
11,0.002002,0.014029,-2.60E-07,3.771477,-0.00556,0.027117,-1.40E-05,9.412401,745.95
12,0.00697,0.047469,2.59E-06,-1.28579,-0.00399,0.007256,-1.70E-05,6.096779,674.34
```

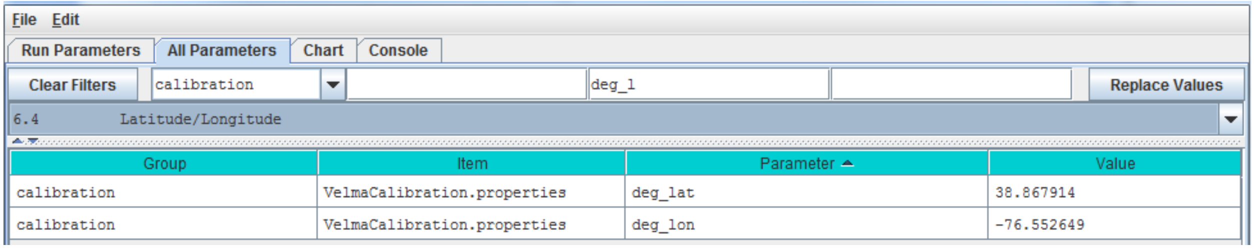
The Spatial and Default Weather Model Both Employ the Same Snow Model

Parameterization of the Spatial Weather Model's snow (sub)model is exactly the same as that of the Default Weather Model's. See the Default Weather Model's documentation for the details.

The difference between the two is that the Spatial Weather Model first calculates the precipitation and air temperature values *for the specific cell* before applying the snow model's mechanism to determine the day's rain, snow, snow depth and snow melt values. In contrast, the Default Weather Model presents the same (driver) precipitation and air temperature values to the snow model at every cell location in the simulation's watershed.

6.4 – Latitude/Longitude

The VELMA simulator computes a daily solar radiation value for the latitude and longitude specified by the `deg_lat` and `deg_lon` parameter values. Set the values of these parameters to the latitude and longitude of your simulation configuration's location. Use decimal degrees, NOT "degrees, minutes, seconds" format for the values. For example:



calibration

deg_l

6.4 Latitude/Longitude

Group	Item	Parameter ▲	Value
calibration	VelmaCalibration.properties	deg_lat	38.867914
calibration	VelmaCalibration.properties	deg_lon	-76.552649

6.5 – Solar Radiation

In addition to latitude and longitude, VELMA's solar radiation equation requires values for the following parameters:

`isc`	Solar Constant.
`sr1`	Constant parameter 1 of solar radiation uptake function.
`sr2`	Constant parameter 2 of solar radiation uptake function.
`srth`	Solar radiation threshold.

The VELMA GUI automatically provides default values for each of these parameters; these may be acceptable for your simulation location.

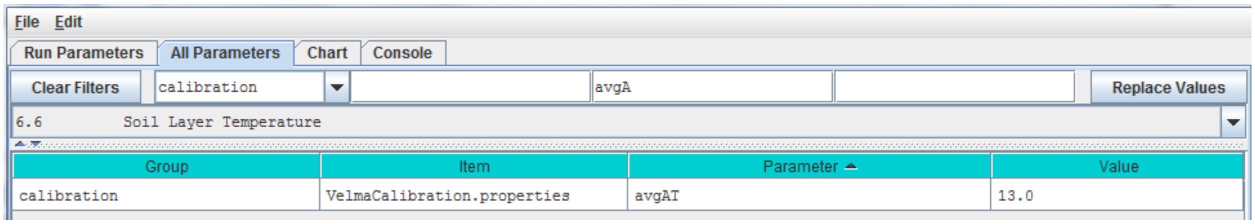
NOTE: Solar radiation parameters are intended for use in computing plant nitrogen uptake and snow melt. Although these modeling capabilities are under development, VELMA 2.0 currently does not include them. That being the case you may leave the solar radiation parameters set to their default values without affecting your simulation run results.

6.6 – Soil Layer Temperature

Soil temperature is used in many of VELMA's subroutines, for example, to compute rates of soil organic matter decomposition, plant nitrogen uptake, nitrification, and many other temperature sensitive processes. Documentation describing the equations and methods for computing subsurface temperature per VELMA's four soil layers can be found in Appendix 6.2 section A1.1.

For the soil temperature subroutine to work properly for your simulation site, you need to:

- provide a daily mean air temperature file (see "Required Input Data Files for VELMA Simulations")
- specify values for total soil depth and soil bulk density (see All Parameters sections 4.3 and 4.5, respectively)
- and specify the average annual air temperature for your site (avgAT), as follows:



calibration | avgA

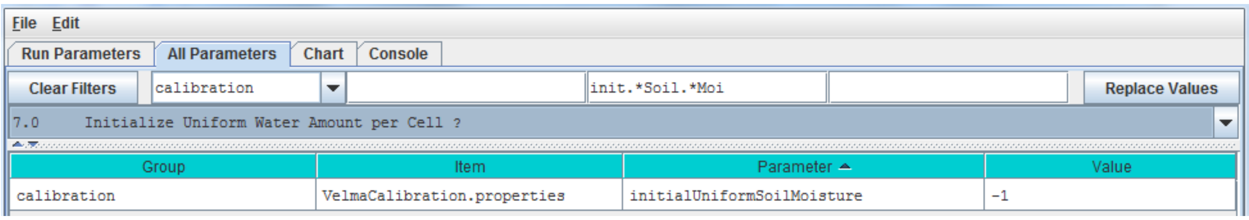
6.6 Soil Layer Temperature

Group	Item	Parameter ▲	Value
calibration	VelmaCalibration.properties	avgAT	13.0

7.0 – Initialize Uniform Water Amount per Cell (link to All Parameters TOC)

Because soil moisture can vary significantly in space and time, it would be extremely difficult to specify realistic soil moisture amounts for every soil layer in every cell within a watershed. A more practical method is to specify a uniform amount of initial soil moisture for all layers and cells, and then allow these initial values to adjust to realistic levels during a “spin up” period of years prior to the simulated years of interest.

By default, VELMA set the initial water content of all cells and layers to the field capacity value specified for each soil type. , i.e., when the parameter `initialUniformSoilMoisture` is set to its default value of -1 (if it is set to any other value, reset it to -1).



calibration | init.*Soil.*Moi

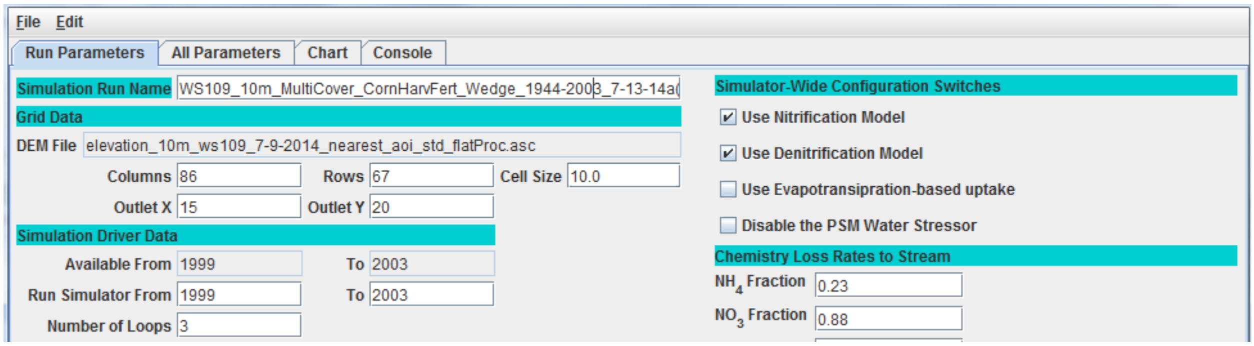
7.0 Initialize Uniform Water Amount per Cell ?

Group	Item	Parameter ▲	Value
calibration	VelmaCalibration.properties	initialUniformSoilMoisture	-1

<u>Calibration Notes</u>

We recommend using this default option, and that you also discard the first year of simulation results, and possibly more, until soil moisture levels have sufficient time to “spin up” to your site’s observed conditions. Shallow soils may adjust within a year or two, whereas deep soils may require decades before simulated groundwater tables reach observed levels. You will need to experiment with this to determine how many spin-up years are required.

To set up a spin-up period, you can instruct VELMA to perform one or more simulation “loops” for the set of years included in the climate input files. To do this, specify the desired “Number of Loops” in Run Parameters tab of the simulator’s GUI. For the example shown below, 3 loops are specified, meaning that the 5 years (1999 – 2003) of available climate driver data will be repeated (looped) 3 times for this simulation.



Simulation Run Name: WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a(

Grid Data

DEM File: elevation_10m_ws109_7-9-2014_nearest_aoi_std_flatProc.asc

Columns: 86 Rows: 67 Cell Size: 10.0

Outlet X: 15 Outlet Y: 20

Simulation Driver Data

Available From: 1999 To: 2003

Run Simulator From: 1999 To: 2003

Number of Loops: 3

Simulator-Wide Configuration Switches

☑ Use Nitrification Model

☑ Use Denitrification Model

☐ Use Evapotranspiration-based uptake

☐ Disable the PSM Water Stressor

Chemistry Loss Rates to Stream

NH_4 Fraction: 0.23

NO_3 Fraction: 0.88

Thus, VELMA will generate 3 loops x 5 years = 15 years of model output, for which the first 2 loops (10 spin-up years) can be discarded.

An alternative method to accomplish the same thing would be to edit the temperature and precipitation driver files, such that a copy of the years you want to use for spin-up purposes appears at the beginning of the original climate record (remember to rename the edited climate driver files to reflect the spin-up number of years).

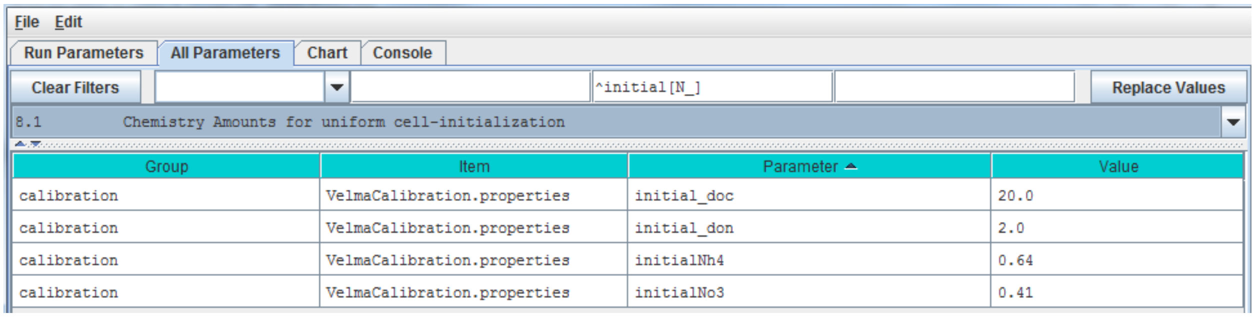
8.0 – Chemistry Pools (Dissolved C and N) (link to All Parameters TOC)

VELMA simulates ecosystem inputs, outputs and internal cycling of four pools of dissolved nitrogen and carbon in soil: ammonium (NH_4), nitrate (NO_3), dissolved organic nitrogen (DON) and dissolved organic carbon (DOC).

Sections 8.1 and 8.2, below, describe the specification of initial pools and calibration of parameters controlling leaching losses to streams. Subsequent sections describe additional processes affecting inputs and outputs from these dissolved C and N pools.

8.1 – Chemistry Amounts for uniform cell-initialization

Initial soil pools (g N / m^2 or g C / m^2) of NH4, NO3, DON and DOC can be viewed and specified by selecting “8.1 Chemistry Amounts for uniform cell-initialization” from the All Parameters drop-down menu. The values specified should represent the sum of all 4 soil layers (total soil column) averaged across the watershed.



^initial[N_]

8.1 Chemistry Amounts for uniform cell-initialization

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	initial_doc	20.0
calibration	VelmaCalibration.properties	initial_don	2.0
calibration	VelmaCalibration.properties	initialNh4	0.64
calibration	VelmaCalibration.properties	initialNo3	0.41

Parameter Definitions

initial_doc Initial DOC pool amount (per cell) in gC/m^2

initial_don Initial DON pool amount (per cell) in gN/m^2.

initial_Nh4 Initial NH4 pool amount (per cell) in gN/m^2.

initial_No3 Initial NO3 pool amount (per cell) in gN/m^2.

Calibration Notes

On day 1 of a simulation, VELMA will use the specified parameter values to calculate the fraction of each nutrient’s total pool to allocate to each of the four soil layers. This is done uniformly for all grid cells on day 1.

With some effort, VELMA can be calibrated to simulate spatial variability in C, N, H_2O pools and fluxes. With a well-calibrated parameter set, initially uniform nutrient pool values will begin to adjust and, after a sufficient spin-up time, more closely match spatial and temporal patterns of observed nutrient pools (McKane 2014).

Section 7.0 describes methods for setting up an appropriate spin-up period for allowing initialized soil moisture levels to adjust to a site’s biophysical conditions. The same spin-up method applies for all model pools and fluxes, including stream chemistry.

Alternatively, you can use a more advanced method for specifying initial nutrient pool values that consider spatial variability in soil types, and cover species type and age. Details can be found in Appendix 3: Creating Initial ASCII Grid Chemistry Spatial Data Pools.

References

The User Manual package includes slides for a seminar describing the effectiveness of riparian buffers for protecting water quality in the Chesapeake Bay (please contact mckane.bob@epa.gov if you have questions):

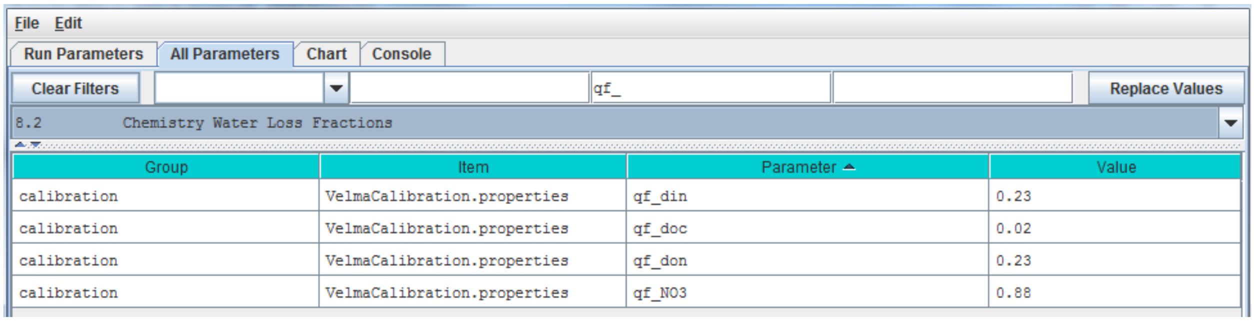
Filename: McKane_ORD GI webinar_7-30-14 Final.pptx

Folder location: VELMA Model\Supporting Documents\VELMA Pubs & Talks

8.2 – Chemistry Water Loss Fractions

In VELMA, transport of dissolved nutrients within the soil column, and ultimately to the stream, is a function of vertical and lateral water drainage, the size of the nutrient pool, and a specified loss fraction (qf) parameter.

The nutrient loss fraction parameters can be viewed and specified by selecting "8.2 Chemistry Water Loss Fractions" from the All Parameters drop-down menu.



qf_

8.2 Chemistry Water Loss Fractions

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	qf_din	0.23
calibration	VelmaCalibration.properties	qf_doc	0.02
calibration	VelmaCalibration.properties	qf_don	0.23
calibration	VelmaCalibration.properties	qf_NO3	0.88

Parameter Definitions

qf_din NH4 loss due to horizontal and vertical water flow range [0.0 1.0].

qf_doc DOC loss due to horizontal and vertical water flow range [0.0 1.0].

qf_don DON loss due to horizontal and vertical water flow range [0.0 1.0].

qf_NO3 NO3 loss due to horizontal and vertical water flow range [0.0 1.0].

Calibration Notes

The "qf" parameters for specifying nutrient loss fractions are very important for achieving a good fit between simulated and observed stream chemistry data (Abdelnour et al. 2013). However, when you are calibrating these and any other parameters, keep in mind that it is fairly easy to specify qf values that provide a good fit between simulated and observed stream chemistry data, even when the source soil nutrient pools are poorly calibrated (too large or too small.

Achieving a well-calibrated ecosystem model is an iterative process, requiring numerous simulation runs until a set of parameters values is identified that simultaneously provides a good fit between simulated and observed data for plants, soils and streams, i.e., all C, N and H_2O pools and fluxes for which good quality observed data are available (recognizing that complete validation is an ideal never reached).

9.0 - 9.1.1 – Nitrogen Deposition (link to All Parameters TOC)

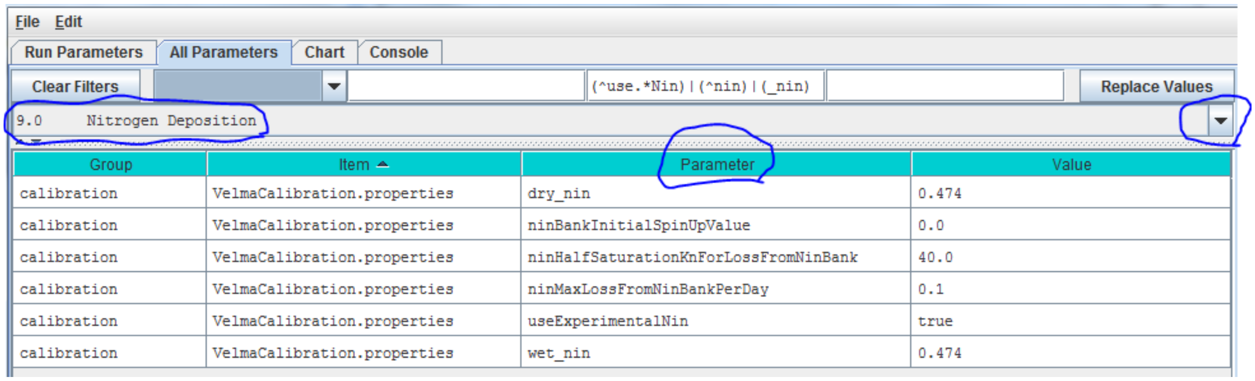
Atmospheric inputs of wet and dry nitrogen deposition can make a significant contribution to ecosystem nitrogen budgets, especially in regions influenced by industry and agricultural. It is important to estimate the timing and amounts of wet and dry nitrogen deposition, and its availability for plant uptake and other ecohydrological processes. Such estimates are complicated by time lags associated with dry deposition of nitrogen on leaves and other surfaces, and subsequent flushing into the soil during precipitation events.

VELMA version 1.0 modeled daily atmospheric inputs of wet and dry deposition as a function of the annual atmospheric input and the fraction of annual rain that occurs at a given day.

VELMA version 2.0 models atmospheric nitrogen deposition based on the following assumptions:

1) The daily rate of atmospheric N deposition (wet + dry) is the same every day of the year, i.e., annual total N deposition/365 = daily N deposition
2) Daily N deposition (`Nin`) accumulates on leaves & other surfaces (`NinBank`) until a liquid precipitation (rain + snowmelt) event.
3) Precipitation events wash a fraction of `NinBank` (throughfall N) into the surface soil layer's inorganic N pool. This fraction (0-1) varies as a nonlinear (asymptotic) function of leaf biomass and daily precipitation amount, such that grid cells having higher leaf biomass have disproportionately lower throughfall N at lower amounts of precipitation. Furthermore, N stored in `NinBank` becomes depleted past a certain upper threshold (asymptote) of daily precipitation. Thus, the main difference between this method and VELMA version 1.0 is that it can account for observed nonlinear threshold behavior of throughfall as a function of leaf area and precipitation amount.
4) Currently, daily `Nin` is composed of NH4 + NO3 deposition, and `Nin` transferred from the `NinBank` to the surface soil layer is to the ammonium pool in `Layer0` (NH4 and NO3 will be modeled separately in a future version of VELMA).

Parameters for the experimental (but preferred) nitrogen deposition subroutine can be viewed and specified by selecting "9.0 Nitrogen Deposition" from the All Parameters drop-down menu:



(^use.*Nin)|(^nin)|(_nin)

9.0 Nitrogen Deposition

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	dry_nin	0.474
calibration	VelmaCalibration.properties	ninBankInitialSpinUpValue	0.0
calibration	VelmaCalibration.properties	ninHalfSaturationKnForLossFromNinBank	40.0
calibration	VelmaCalibration.properties	ninMaxLossFromNinBankPerDay	0.1
calibration	VelmaCalibration.properties	useExperimentalNin	true
calibration	VelmaCalibration.properties	wet_nin	0.474

Parameter Definitions

`wet_nin` Wet nitrogen deposition factor in gNm^2 per year

`dry_nin` Dry nitrogen deposition factor in gNm^2 per year

`ninBankInitialSpinUpValue` N-deposition initialization value. When this initialization value is zero a simulation run's first year is used to "spin up" the Michaelis-Menten-based N-deposition model. The N-deposition values computed for that first year (and subsequent values calculated from them) are unlikely to be accurate. However if a correct value is known this initialization parameter may be set to it. The value should be the daily average N-deposition value for a single cell of the current simulation. If the value is unknown or cannot be reliably deduced ahead of simulation start leave this value zero and incur the first-year spin-up (see Calibration notes, below)

`ninHalfSaturationKnForLossFromNinBank` N-deposition Michaelis-Menten kinetics Kn constant.

`ninMaxLossFromNinBankPerDay` N-Deposition Michaelis-Menten kinetics maximum

`useExperimentalNin` When set to “true” N-deposition is computed using a banked Michaelis-Menten-based equation. When set to “false” N-deposition is computed using the VELMA version 1.0 linear equation. Use of the older equation is deprecated. The default value for this parameter is “true”.

Calibration Notes

For the reasons mentioned under assumption #3, above, we recommend using the experimental (VELMA version 2.0) nitrogen deposition subroutine rather than that used in VELMA version 1.0.

We have prepared a DRAFT Excel spreadsheet that you may choose to use to parameterize the experimental nitrogen deposition subroutine (please contact mckane.bob@epa.gov if you have questions):

Filename: VELMA 2.0_Nitrogen Deposition Calibrator_7-26-13 v3.xlsx

Folder location: VELMA Model\Supporting Documents\Excel Calibration Files

9.1.2 Old (DEPRECATED) Nitrogen Deposition Parameters

Use of the VELMA version 1.0 nitrogen deposition subroutine is discouraged. Please see section 9.0.

10.0 – Transpiration (link to All Parameters TOC)

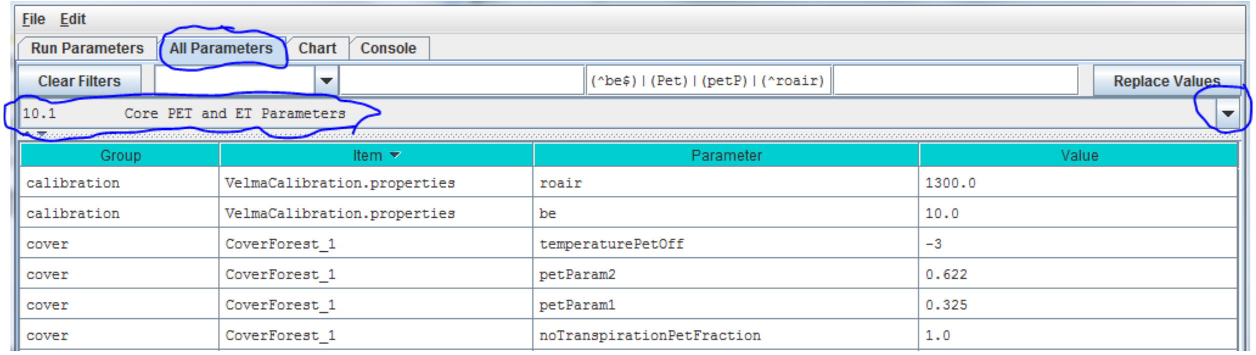
Evapotranspiration (ET) is the sum of evaporation and plant transpiration. ET equations and parameters in VELMA version 2.0 are the same as for version 1.0 (see Abdelnour et al. 2011), except for the following:

- The ET recovery function has been replaced with a new subroutine that models post-disturbance changes in ET based on leaf biomass dynamics.
- A new subroutine models age-related declines in transpiration for forest systems.

See details in subsections 10.1 – 10.3.1.

10.1 – Core PET and ET Parameters

Core parameters for calibrating VELMA's PET and ET submodels can be found by selecting section 10.1 from the All Parameters drop-down menu:



(^be$)|(Pet)|(petP)|(^roair)

10.1 Core PET and ET Parameters

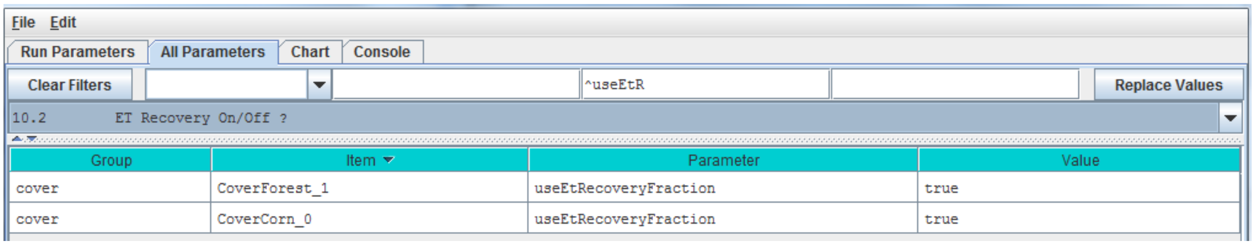
Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	roair	1300.0
calibration	VelmaCalibration.properties	be	10.0
cover	CoverForest_1	temperaturePetOff	-3
cover	CoverForest_1	petParam2	0.622
cover	CoverForest_1	petParam1	0.325
cover	CoverForest_1	noTranspirationPetFraction	1.0

Parameter definitions (see Abdelnour et al. 2011 for equations and additional explanation):

`roair`	Air density in g/m^3
`be`	ET coefficient used in the logistic equation that computes ET from PET
`temperaturePetOff`	PET is only active when air temperature is > than this value (in degrees C)
`petParam1`	First term of PET Hamon Equation: petParam1 * petParam2 * gS.roair * (esat / 1000.0f)
`petParam2`	Second term of PET Hamon Equation: petParam1 * petParam2 * gS.roair * (esat / 1000.0f)
`notranspirationPetFraction`	The fraction of PET available outside of this a cover species/type's growing season range [0.0 - 1.0].

10.2 – ET Recovery On/Off ?

Select section 10.2 from the All Parameters drop-down menu to turn the ET recovery function on (true) or off (false):



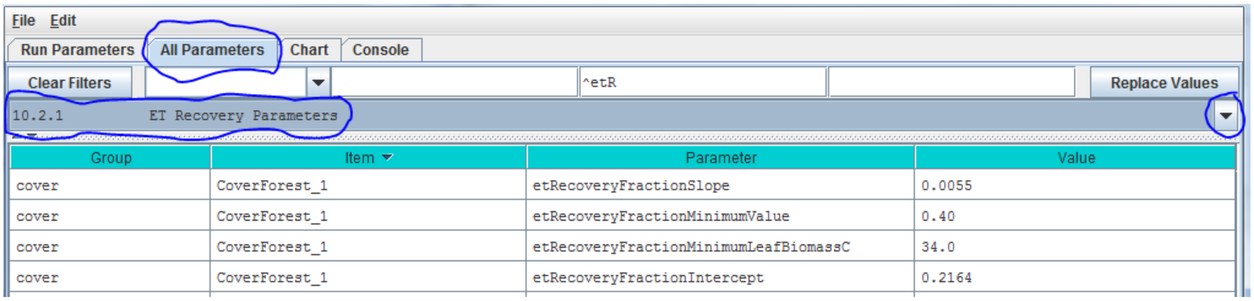
^useEtR

10.2 ET Recovery On/Off ?

Group	Item	Parameter	Value
cover	CoverForest_1	useEtRecoveryFraction	true
cover	CoverCorn_0	useEtRecoveryFraction	true

10.2.1 – ET Recovery Parameters

Select section 10.2.1 from the All Parameters drop-down menu to specify ET recovery parameter values for each cover type in your watershed (just one cover shown below):



^etR

10.2.1 ET Recovery Parameters

Group	Item	Parameter	Value
cover	CoverForest_1	etRecoveryFractionSlope	0.0055
cover	CoverForest_1	etRecoveryFractionMinimumValue	0.40
cover	CoverForest_1	etRecoveryFractionMinimumLeafBiomassC	34.0
cover	CoverForest_1	etRecoveryFractionIntercept	0.2164

Parameter Definitions

`etRecoverFractionSlope` The value of the ET recovery fraction's linear equation slope component.

`etRecoverFractionMinimumValue` The ET recovery fraction is forced to this value when the current leaf biomass is less than the value specified for `etRecoveryFractionLeafBiomassMinimum` or when the calculated ET recovery fraction is less than zero.

`etRecoverFractionMinimumLeafBiomassC` The minimum amount of leaf biomass (in gC/m^2) required to compute an ET recovery fraction. Whenever leaf biomass is less than this amount the ET recovery fraction is forced to the etRecoverFractionMinimumValue.

`etRecoverFractionIntercept` The value of the ET recovery fraction's linear equation intercept component.

Calibration Notes

We have prepared a DRAFT Excel spreadsheet that you may choose to use to calibrate ET recovery parameters for your site (please contact mckane.bob@epa.gov if you have questions):

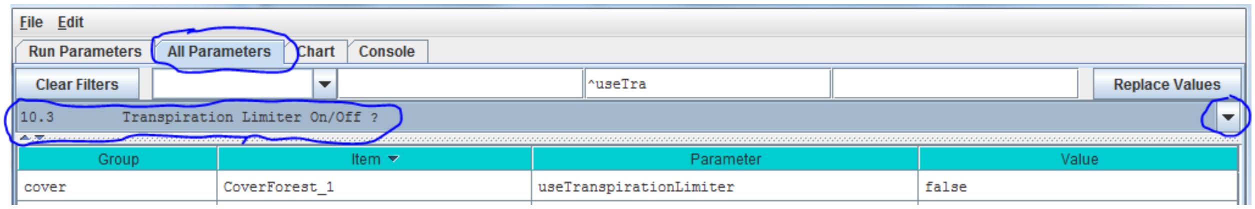
Filename: VELMA 2.0_ET Recovery Calibrator_HJA_12-1-13.xlsx

Folder location: VELMA Model\Supporting Documents\Excel Calibration Files

10.3 – Transpiration Limiter On/Off?

Studies in the Pacific Northwest have shown that young forests transpire significantly more water than old forests (McDowell et al. 2002; Moore et al. 2004). VELMA version 2.0 has a new subroutine to address age-related declines in transpiration by forest vegetation.

Select section 10.3 from the All Parameters drop-down menu to turn the Transpiration Limiter subroutine on (true) or off (false):



^useTra

10.3 Transpiration Limiter On/Off ?

Group	Item	Parameter	Value
cover	CoverForest_1	useTranspirationLimiter	false

Parameter definition:

`useTranspirationLimiter` When set false (default value) the transpiration limiter fraction value is forced to 1.0 (Which effectively means "No age-related limitation" i.e. the limiter is OFF.) When set true the transpiration limiter fraction is computed based on its jday and coefficients settings.

See section 10.4 for parameters and calibration information.

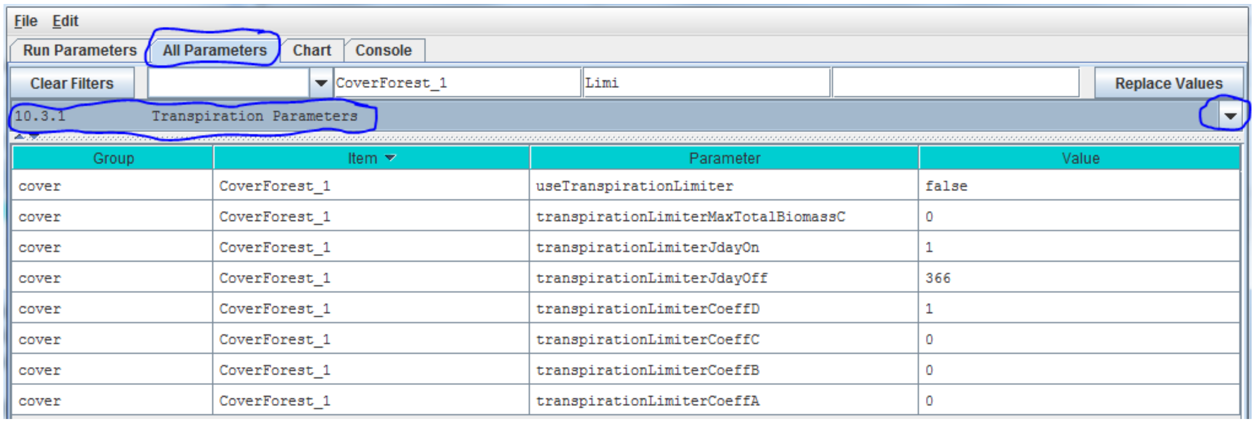
References

McDowell, N. G., Phillips, N., Lunch, C., Bond, B. J., & Ryan, M. G. (2002). An investigation of hydraulic limitation and compensation in large, old Douglas-fir trees. Tree Physiology, 22(11), 763-774.

Moore, G. W., Bond, B. J., Jones, J. A., Phillips, N., & Meinzer, F. C. (2004). Structural and compositional controls on transpiration in 40-and 450-year-old riparian forests in western Oregon, USA. Tree physiology, 24(5), 481-491.

10.3.1 – Transpiration Parameters

Select section 10.3.1 from the All Parameters drop-down menu to specify parameter values for the Transpiration Limiter subroutine used to describe age-related declines in transpiration by forest vegetation:



CoverForest_1 | Limi

10.3.1 Transpiration Parameters

Group	Item	Parameter	Value
cover	CoverForest_1	useTranspirationLimiter	false
cover	CoverForest_1	transpirationLimiterMaxTotalBiomassC	0
cover	CoverForest_1	transpirationLimiterJdayOn	1
cover	CoverForest_1	transpirationLimiterJdayOff	366
cover	CoverForest_1	transpirationLimiterCoeffD	1
cover	CoverForest_1	transpirationLimiterCoeffC	0
cover	CoverForest_1	transpirationLimiterCoeffB	0
cover	CoverForest_1	transpirationLimiterCoeffA	0

Parameter Definitions

`useTranspirationLimiter` When set false (default value) the transpiration limiter fraction value is forced to 1.0 (Which effectively means "No limiting at all" i.e. the limiter is OFF.) When set true the transpiration limiter fraction is computed based on its jday and coefficients settings.

`transpirationLimiterMaxTotalBiomassC` Upper limit of total biomass amount (in gC/m^2) for transpiration limiter fraction computation. When total biomass amount in a cell exceeds this max value the transpiration limiter faction value is forced to the transpirationLimiterCoeffD parameter's value.

`transpirationLimiterJdayOn` Transpiration reduction occurs on the Julian day specified by this "on" parameter and continues on every day of the year up to and including the day

	specified by the paired "off" parameter. I.e. transpiration reduction occurs in the range [transpirationLimiterJdayOn transpirationLimiterJdayOff]
`transpirationLimiterJdayOff`	Transpiration reduction occurs on the Julian day specified by this "off" parameter and does not occur for any days of the year remaining after it. I.e. transpiration reduction occurs in the range [transpirationLimiterJdayOn transpirationLimiterJdayOff]
`transpirationLimiterCoeffA`	The A coefficient in the Transpiration Limiter Fraction equation: A / (1 + B * exp(C * totalBiomassC)) + D
`transpirationLimiterCoeffB`	The B coefficient in the Transpiration Limiter Fraction equation: A / (1 + B * exp(C * totalBiomassC)) + D
`transpirationLimiterCoeffC`	The C coefficient in the Transpiration Limiter Fraction equation: A / (1 + B * exp(C * totalBiomassC)) + D
`transpirationLimiterCoeffD`	The D coefficient in the Transpiration Limiter Fraction equation: A / (1 + B * exp(C * totalBiomassC)) + D

Calibration Notes

The transpiration limiter subroutine is experimental and we do not recommend using it at this time.

11.0 Plant Uptake (link to All Parameters TOC)

The VELMA v1.0 subroutine for plant nitrogen uptake is described by Abdelnour et al. (2011). VELMA v2.0 generally follows the same subroutine, but includes important changes to several components affecting nitrogen uptake:

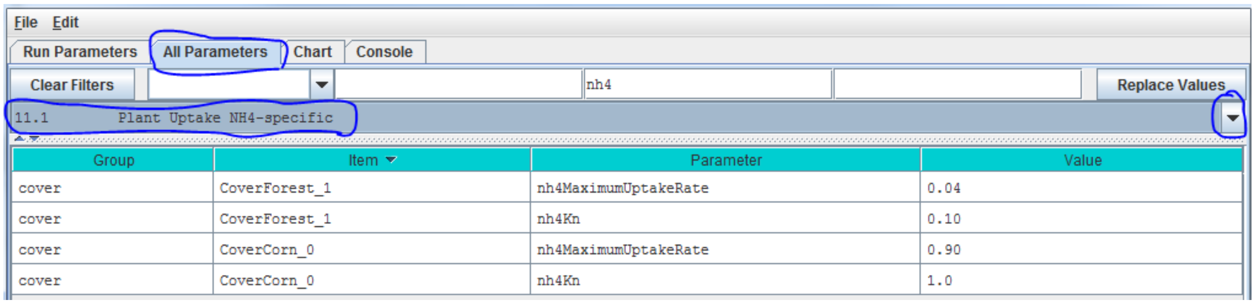
- Uptake of NH_4 and NO_3 is now modeled explicitly, using pool-specific Michaelis-Menten equations.
- The temperature response function for plant N uptake is now based on a quadratic equation described by Rastetter et al. (1991).
- In VELMA v1.0 a constant root fraction parameter was applied to total plant biomass to indirectly simulate effects of successional changes in root biomass on N uptake. VELMA v2.0 includes a leaf-stem-root (LSR) subroutine that explicitly models root biomass dynamics and consequent effects on NH_4 and NO_3 uptake.
- Simulated total N uptake per day is allocated to leaf, stem and root tissues according to specified fractions per tissue.
- Phenological controls on plant nitrogen uptake constrain when nitrogen uptake can occur.
- The nonlinear (Weibull) function in VELMA v1.0 for simulating the effect of forest stand age on N uptake has been removed. This and other age-related effects in VELMA v2.0 are now modeled as a consequence of successional changes in the availabilities of nitrogen and water.

- The VELMA v2.0 plant N uptake subroutine must be set up and calibrated for each cover species/types within a watershed.

Details are provided in subsections 11.1 – 11.5.

11.1 – Plant Uptake NH4-specific

Select “11.1 Plant Uptake NH4-specific” from the All Parameters drop-down menu to specify parameter values controlling NH_4 uptake per cover species/type. Note that two cover types (forest and corn) are shown here to emphasize that parameters must be specified for each cover type within a watershed (true for all of section 11).



nh4

11.1 Plant Uptake NH4-specific

Group	Item	Parameter	Value
cover	CoverForest_1	nh4MaximumUptakeRate	0.04
cover	CoverForest_1	nh4Kn	0.10
cover	CoverCorn_0	nh4MaximumUptakeRate	0.90
cover	CoverCorn_0	nh4Kn	1.0

Parameter Definitions

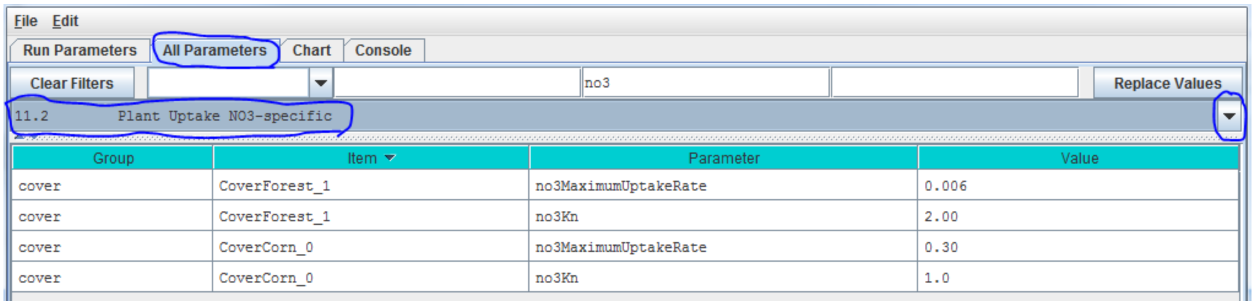
`nh4MaximumUptakeRate` Cover-specific maximum rate of NH_4 uptake for biomass uptake equation.

`nh4Kn` Half saturation constant kn for Michaelis Menton NH_4 uptake in gN/m^2.

See calibration notes for section 11.2.

11.2 – Plant Uptake NO3-specific

Select “11.2 Plant Uptake NO3-specific” from the All Parameters drop-down menu to specify parameter values controlling NO_3 uptake per cover species/type:



no3

11.2 Plant Uptake NO3-specific

Group	Item	Parameter	Value
cover	CoverForest_1	no3MaximumUptakeRate	0.006
cover	CoverForest_1	no3Kn	2.00
cover	CoverCorn_0	no3MaximumUptakeRate	0.30
cover	CoverCorn_0	no3Kn	1.0

Parameter Definitions

`no3MaximumUptakeRate` Cover-specific maximum rate of NO_3 uptake for biomass uptake equation.

`no3Kn` Half saturation constant, kn, for Michaelis-Menten NO_3 uptake in gN/m^2.

Calibration Notes

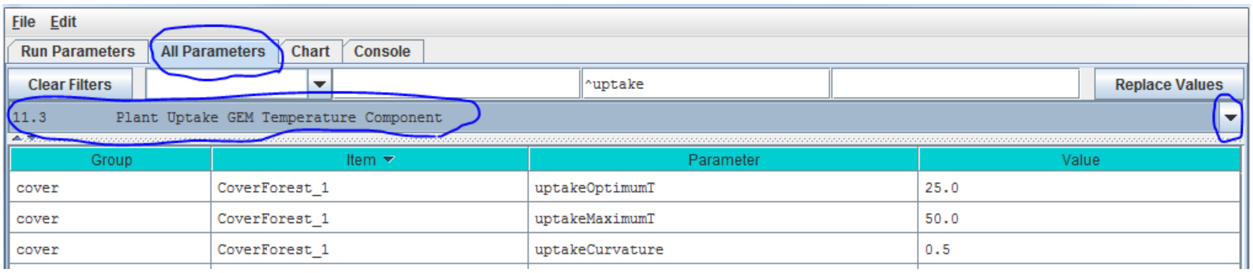
We have prepared a DRAFT Excel spreadsheet that you may choose to use to calibrate NH4 and NO3 plant uptake parameters for your site (please contact mckane.bob@epa.gov if you have questions):

Filename: VELMA 2.0_Nitrogen Uptake Calibrator (CHB Forest)_9-24-13.xlsx

Folder location: VELMA2.0Software/SupportingDocuments/ExcelCalibrationFiles

11.3 – Plant Uptake GEM Temperature Component

Select "11.3 Plant Uptake GEM Temperature Component" from the All Parameters drop-down menu to specify parameter values for the temperature response function affecting NH_4 and NO_3 uptake (specify per cover type):

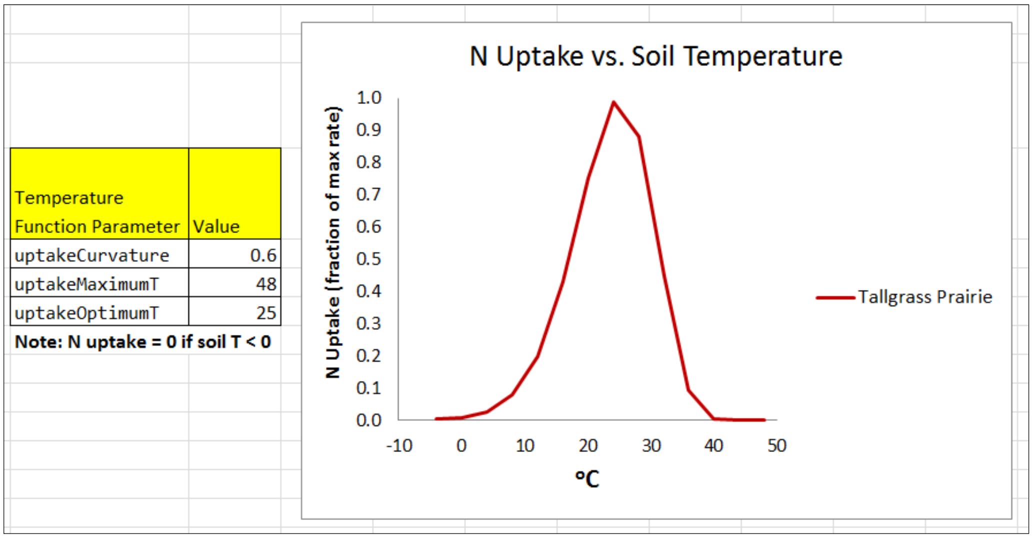


Group	Item	Parameter	Value
cover	CoverForest_1	uptakeOptimumT	25.0
cover	CoverForest_1	uptakeMaximumT	50.0
cover	CoverForest_1	uptakeCurvature	0.5

Parameter definitions (see Rastetter et al. (1991) for a description of the General Ecosystem Model (GEM) temperature function and parameters for simulating plant nitrogen uptake):

`uptakeOptimumT`	Optimal temperature (°C) plant uptake of NH_4 and NO_3
`uptakeMaximumT`	Maximum temperature (°C) plant uptake of NH_4 and NO_3
`uptakeCurvature`	Curvature parameter of GEM temperature function.

An example:



Temperature Function Parameter	Value
uptakeCurvature	0.6
uptakeMaximumT	48
uptakeOptimumT	25

Note: N uptake = 0 if soil T < 0

Calibration Notes

We have prepared a DRAFT Excel spreadsheet that you may choose to use to calibrate temperature response functions for plant N uptake and other temperature sensitive processes in VELMA. (please contact mckane.bob@epa.gov if you have questions):

Filename: VELMA 2.0_Temperature & Moisture Response Functions_9-8-14.xlsx

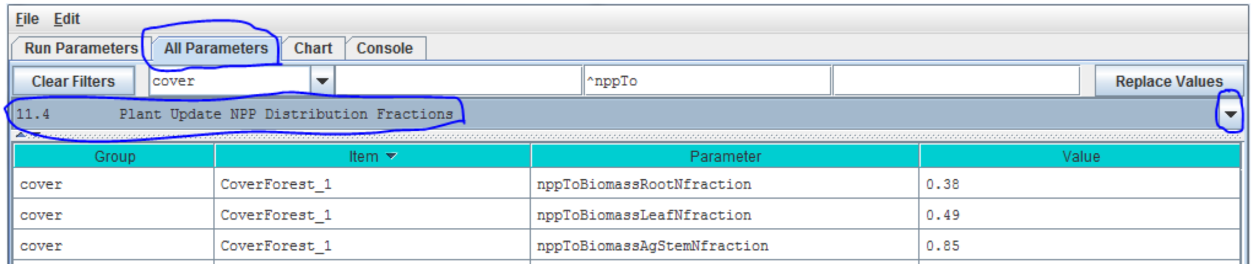
Folder location: VELMA Model\Supporting Documents\Excel Calibration Files

References

Rastetter, E. B., Ryan, M. G., Shaver, G. R., Melillo, J. M., Nadelhoffer, K. J., Hobbie, J. E., & Aber, J. D. (1991). A general biogeochemical model describing the responses of the C and N cycles in terrestrial ecosystems to changes in CO2, climate, and N deposition. Tree Physiology, 9(1-2), 101-126.

11.4 – Plant Uptake NPP Distribution Fractions

Select “11.4 Plant Uptake NPP Distribution Fractions” from the All Parameters drop-down menu to specify parameter values controlling how daily total N uptake (a.k.a, “NPP” in VELMA) is allocated to leaf, aboveground stem, belowground stem, and root biomass:



cover

^nppTo

11.4 Plant Update NPP Distribution Fractions

Group	Item	Parameter	Value
cover	CoverForest_1	nppToBiomassRootNfraction	0.38
cover	CoverForest_1	nppToBiomassLeafNfraction	0.49
cover	CoverForest_1	nppToBiomassAgStemNfraction	0.85

Parameter Definitions

`nppToBiomassRootNfraction` The fraction (0.0 -1.0) of total plant uptake of nitrogen (NH_4 + NO_3) per day allocated to the root biomass N pool.

`nppToBiomassLeafNfraction` The fraction (0.0 -1.0) of total plant uptake of nitrogen (NH_4 + NO_3) per day allocated to the leaf biomass N pool.

`nppToBiomassAgStemNfraction` The fraction (0.0 -1.0) of total plant uptake of nitrogen (NH_4 + NO_3) per day allocated to the AgStem biomass N pool.

Note that “nppToBiomassBgStemfraction” does not need to be specified because VELMA internally calculates it by difference.

Calibration Notes

Figure 1 and Appendix 1 provides conceptual diagrams and equations describing how N uptake (NPP) is allocated among the four LSR plant tissues.

To calibrate the N allocation parameters described in this section, we recommend constructing an annual plant biomass N budget(s) for all four LSR tissues per cover type. Observed data and estimates can usually be found in the literature for this. For example:

– Leaf litterfall data can be used to estimate annual leaf NPP, usually as dry weight or carbon, so you will also need C/N data to calculate annual N uptake.

- Data for above- and belowground stem biomass growth can be obtained from tree-ring data and allometric biomass equations for forest cover types, or from “destructive” biomass harvest methods in grasslands or agricultural plots.
- Production estimates for fine roots exist for many cover types, and are based on a variety of methods – e.g., see Nadelhoffer, K. J., & Raich, J. W. (1992).

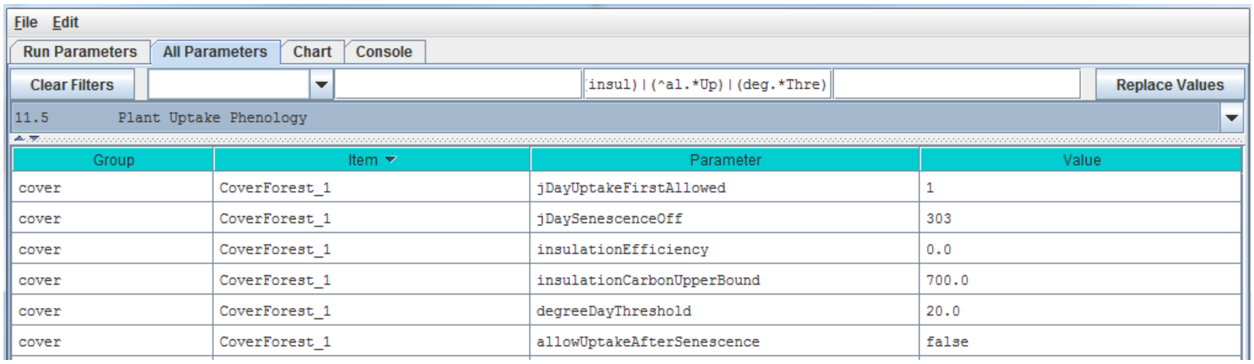
Please contact mckane.bob@epa.gov if you have questions.

References

Nadelhoffer, K. J., & Raich, J. W. (1992). Fine root production estimates and belowground carbon allocation in forest ecosystems. Ecology, 1139-1147

11.5 – Plant Uptake Phenology

Select “11.5 Plant Uptake Phenology” from the All Parameters drop-down menu to specify parameter values affecting when plant N uptake begins and ends during the year (per cover type):



insul)|(^al.*Up)|(deg.*Thre)

11.5 Plant Uptake Phenology

Group	Item	Parameter	Value
cover	CoverForest_1	jDayUptakeFirstAllowed	1
cover	CoverForest_1	jDaySenescenceOff	303
cover	CoverForest_1	insulationEfficiency	0.0
cover	CoverForest_1	insulationCarbonUpperBound	700.0
cover	CoverForest_1	degreeDayThreshold	20.0
cover	CoverForest_1	allowUptakeAfterSenescence	false

Parameter Definitions

`jDayUptakeFirstAllowed` The first Julian day of the year on which uptake is possible for this Cover. Uptake will NEVER happen before the specified day. Use this parameter to delay uptake until the daily photoperiod is sufficient for the given Cover. This parameter specifies the first day of the year when uptake can occur not necessarily the first day when uptake will occur. On and after the day specified by this parameter uptake will occur whenever the degree day parameterization allows it to begin.

`jDaySenescenceOff` The Julian Day of the year when this cover species biomass stops senescing (Note: any value >= 367 results in biomass NOT senescing through the end of the year.`nppToBiomassAgStemNfraction`The fraction (0.0 -1.0) of total plant uptake of nitrogen (NH_4 + NO_3) per day allocated to the AgStem biomass N pool.

`insulationEfficiency` Insulating efficiency of available litter range [0.0 1.0] (this affects soil temperature and, therefore, plant N uptake). NOTE: Set this value to zero to disable insulation effects on degree-day calculation.

`insulationCarbonUpperBound` Quantity of above-ground litter beyond which there is no additional insulating effect in gC/m^2 (this affects soil temperature and, therefore, plant N uptake).

`degreeDayThreshold`	Allow biomass uptake only when accumulated degree day value >= degreeDayThreshold.
`allowUptakeAfterSenescence`	When set true (default) uptake takes place before jDaySenescenceOn and resumes after jDaySenescenceOff. When set "false" uptake occurs ONLY before jDaySenescenceOn

12.0 Plant Mortality (link to All Parameters TOC)

VELMA v.2.0 has a new plant biomass mortality function, replacing the linear mortality function in v1.0 (Abdelnour et al. 2011). The new mortality function was initially developed for forest vegetation but we have found that it applies just as well to other cover types, including nonwoody species such as grasslands and agricultural crops.

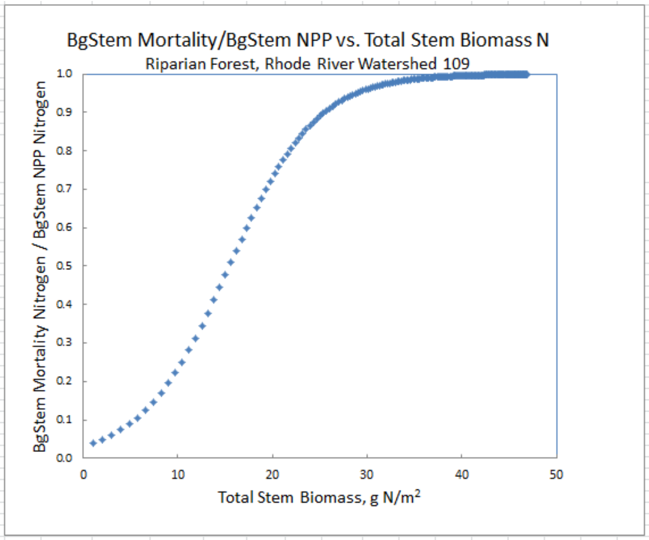
The mortality function is based on the following observations and assumptions:

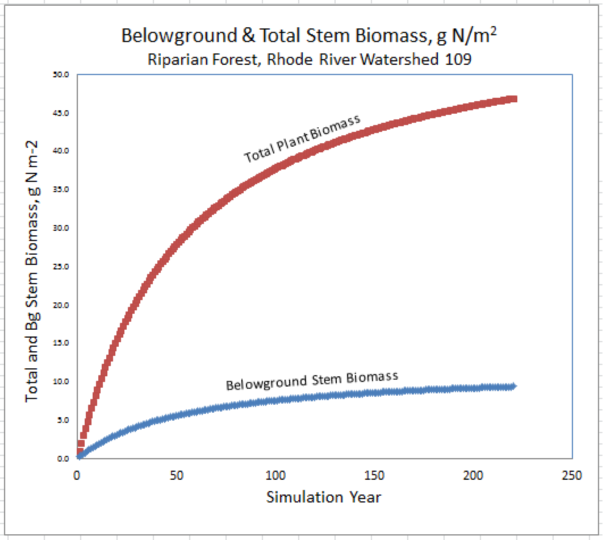
- The ratio of mortality/NPP during succession is nonlinear. For example, in regrowing forest stands, observed increases in stem biomass require that stem NPP exceeds stem mortality until steady state is reached.
- A logistic function provides sufficient flexibility to capture temporal changes in the relationship between mortality and NPP during succession. Specifically, a logistic function is used to describe nonlinear changes in a tissue's litterfall/NPP ratio vs. stand age, where "tissue" can represent leaf, aboveground stem, belowground stem, or root biomass. In practice, total plant biomass (or aboveground stem biomass) serves as a proxy for stand age.
- For example, using VELMA's parameter naming convention,

  mortalityAgStemCarbon / nppAgStemCarbon = C / (1 + A * exp^(-B * plantBiomassCarbon))
- Rearranging,

  mortalityAgStemCarbon = nppAgStemCarbon * C / (1 + A * exp^(-B * plantBiomassCarbon))

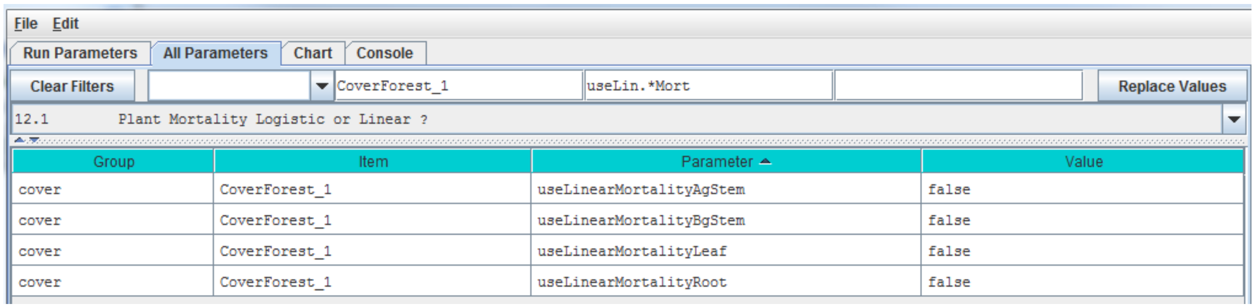
For example, the figures below show results obtained by applying the logistic mortality function to a riparian hardwood forest. The left figure shows changes in the ratio of belowground stem mortality/NPP vs. total stem biomass (proxy for stand age). The right figure shows simulated temporal changes in simulated belowground stem biomass and total plant biomass. Adjustment of the mortality function parameters can be used to flatten or steepen the asymptotic trends in biomass, according to chronosequence data. The mortality function must be applied to all four of VELMA's plant tissues (results for leaf, aboveground stem and fine root biomass are not shown here).





12.1 – Plant Mortality Logistic or Linear?

Select “12.1 Plant Mortality Logistic or Linear ?” from the All Parameters drop-down menu to specify whether you wish to use the logistic (recommended) or linear functions for simulating plant biomass mortality:



CoverForest_1

useLin.*Mort

12.1 Plant Mortality Logistic or Linear ?

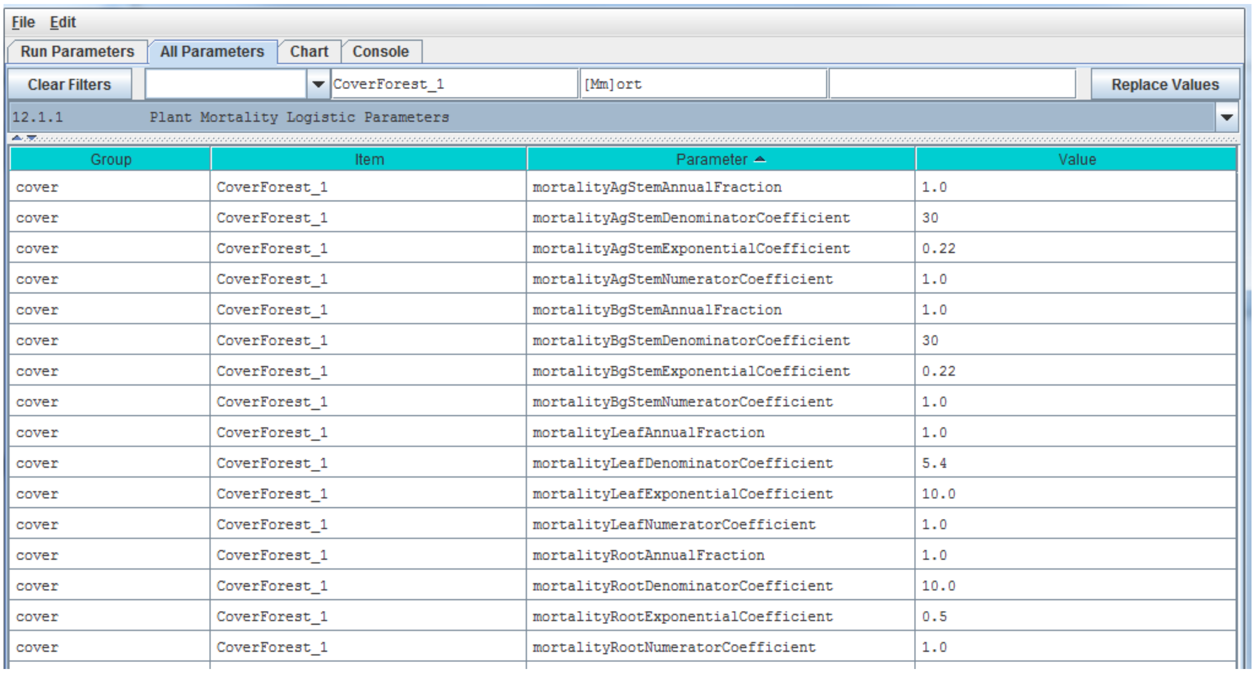
Group	Item	Parameter	Value
cover	CoverForest_1	useLinearMortalityAgStem	false
cover	CoverForest_1	useLinearMortalityBgStem	false
cover	CoverForest_1	useLinearMortalityLeaf	false
cover	CoverForest_1	useLinearMortalityRoot	false

///// IMPORTANT NOTE /////

We do not recommend using the old linear model for simulating plant biomass mortality. Therefore, make sure all four of the parameters shown above are set = false. When that is the case, plant biomass mortality is computed using the logistic function. You will need to do this for all cover types included in your watershed.

12.1.1 – Plant Mortality Logistic Parameters

Select “12.1.1 Plant Mortality Logistic Parameters” from the All Parameters drop-down menu to specify parameter values affecting biomass mortality for each of VELMA’s four plant tissue types. Parameter specifications are required for each cover type.



CoverForest_1 [Mm]ort

12.1.1 Plant Mortality Logistic Parameters

Group	Item	Parameter	Value
cover	CoverForest_1	mortalityAgStemAnnualFraction	1.0
cover	CoverForest_1	mortalityAgStemDenominatorCoefficient	30
cover	CoverForest_1	mortalityAgStemExponentialCoefficient	0.22
cover	CoverForest_1	mortalityAgStemNumeratorCoefficient	1.0
cover	CoverForest_1	mortalityBgStemAnnualFraction	1.0
cover	CoverForest_1	mortalityBgStemDenominatorCoefficient	30
cover	CoverForest_1	mortalityBgStemExponentialCoefficient	0.22
cover	CoverForest_1	mortalityBgStemNumeratorCoefficient	1.0
cover	CoverForest_1	mortalityLeafAnnualFraction	1.0
cover	CoverForest_1	mortalityLeafDenominatorCoefficient	5.4
cover	CoverForest_1	mortalityLeafExponentialCoefficient	10.0
cover	CoverForest_1	mortalityLeafNumeratorCoefficient	1.0
cover	CoverForest_1	mortalityRootAnnualFraction	1.0
cover	CoverForest_1	mortalityRootDenominatorCoefficient	10.0
cover	CoverForest_1	mortalityRootExponentialCoefficient	0.5
cover	CoverForest_1	mortalityRootNumeratorCoefficient	1.0

Reference to mortality coefficients in the parameter definitions, below, refer to the following plant mortality logistic function. Using AgStem as an example:

AgStem Mortality = AgStem NPP * C / (1 + A * exp^(-B * Tissue N)),

where,

AgStem Mortality & AgStem NPP are computed internally by VELMA in g N m^{-2} d^{-1}

A = mortalityAgStemDenominatorCoefficient

B = mortalityAgStemExponentialCoefficient

C = mortalityAgStemNumeratorCoefficient

Parameter Definitions

mortalityAgStemAnnualFraction
The fraction of a tissue’s simulated annual mortality that gets subtracted from the corresponding biomass pool per day during the specified period of senescence. The default value for this fraction is 1.0 (i.e. the full per-day amount) and the valid range is [0.0 to 1.0] Warning! Users unfamiliar with this parameter's behavior should leave it set to the default value of 1.0.

mortalityAgStemDenominatorCoefficient
Parameter “A” in the equation above.

mortalityAgStemExponentialCoefficient
Parameter “B” in the equation above.

above.mortalityAgStemNumeratorCoefficient
Parameter “C” in the equation above.

Note: the 4 preceding parameters repeat for the other tissues – i.e., “Leaf”, “BgStem” or “Root” appear in place of “AgStem” in the parameter names above. These

Calibration Notes

We have prepared a DRAFT Excel spreadsheet that you may choose to use to calibrate plant logistic mortality parameters for your site (please contact mckane.bob@epa.gov if you have questions):

Filename: VELMA 2.0_Biomass Mortality Calibrator_CHB_9-24-13 v3.xlsx

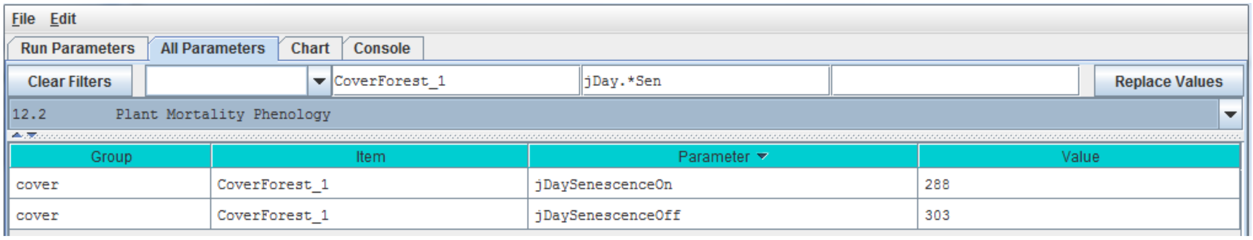
Folder location: VELMA2.0Software/SupportingDocuments/ExcelCalibrationFiles

12.1.1.1 through 12.1.1.4 – Plant Mortality Logistic Parameters for Leaf, AgStem, BgStem and Root Biomass

See section 12.1, above, for mortality parameter definitions and calibration information. Although 12.1 complete list   However, to avoid mistakes, we recommend that you use sections 12.1.1.1 through 12.1.1.4 in the All Parameters drop-down menu to specify tissue-specific mortality parameters. For example, use the All Parameters drop-down menu to select section 12.1.1.1 to display and specify leaf logistic mortality parameters. Parameters for aboveground stem, belowground stem and roots appear under the following 3 subsections.

12.2 – Plant Mortality Phenology

Select “12.2 Plant Mortality Phenology” from the All Parameters drop-down menu to specify parameter values for the beginning and end dates for plant biomass mortality.



CoverForest_1

jDay.*Sen

12.2 Plant Mortality Phenology

Group	Item	Parameter	Value
cover	CoverForest_1	jDaySenescenceOn	288
cover	CoverForest_1	jDaySenescenceOff	303

Parameter Definitions

`jDaySenescenceOn` The Julian Day of the year when this cover species biomass begins senescing. (Note: Set >= 367 to "deactivate" senescence for this cover species.)

`jDaySenescenceOff` The Julian Day of the year when this cover species biomass stops senescing (Note: any value >= 367 results in biomass NOT senescing through the end of the year.

13.0 Decomposition (link to All Parameters TOC)

Decomposition of soil organic matter is a key process in VELMA, controlling the rate at which dead organic C and N pools (detritus and humus) are converted to inorganic C and N pools (CO_2, NH_4, NO_3, DON and DOC).

VELMA v2.0 has a new decomposition model based on Potter et al. (1993). In the Potter model, microbially-mediated decomposition of plant and soil organic residues produces CO_2, such that

$CO2_i = C_i * k_i * Ws * Ts * (1\text{-}Me)$, where

- i = a specified soil organic matter pool (in VELMA, plant tissue detritus or humus)
- $CO2_i$ = carbon dioxide released by the decomposition of pool i
- C_i = carbon content (gC/m^2) of pool i
- k_i = maximum decay rate constant of pool i
- Ws = scalar constant (0-2.8) for the effect of soil moisture content on decomposition

- Ts = scalar constant (0-1) for the effect of temperature on decomposition
- Me = carbon assimilation efficiency (0 – 1) of microbes

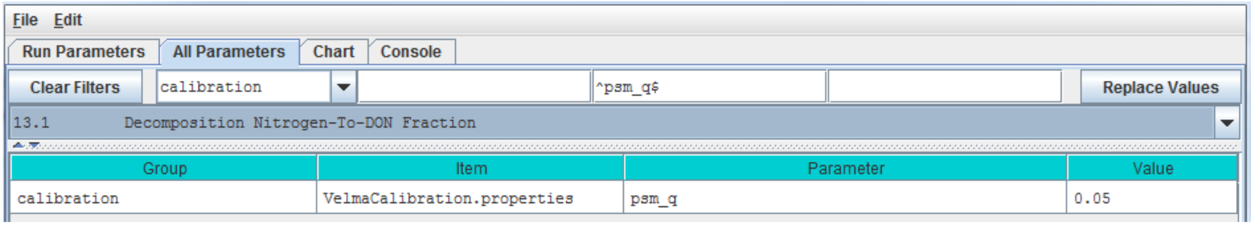
We made several modifications to the Potter model for VELMA v2.0:

- We modified the Potter decomposition equation, above, to apply it to detritus and humus N pools rather than C pools. The difference in the C/N ratios of donor and recipient pools are used to estimate total carbon decomposed. Specified fractions of the total carbon decomposed are converted to CO_2 and DOC. Similarly, a specified fraction of total decomposed N is converted to DON.
- We replaced Potter's scalar constant for soil temperature (Ts) with a nonlinear function describing the fraction (0 – 1) of the maximum decay rate (K_i) versus soil temperature (^{o}C, per layer). See section 11.3 for a description of this nonlinear temperature function, which is also used in the plant N uptake subroutine.
- We replaced Potter's scalar constant for soil moisture (Ws) with a nonlinear function describing the fraction (0 – 1) of the maximum decay rate (K_i) versus layer-specific soil moisture (v/v, per layer). Our soil moisture decomposition function uses a nonlinear Weibull function, which is the same as that applied to the soil moisture N fixation function (see section 14.0).

"Appendix 1: Overview of VELMA's Leaf-Stem-Root (LSR) Plant Biomass Submodel" presents a conceptual diagram and equations describing pools and fluxes involved in the decomposition subroutine for VELMA v2.0.

13.1 – Decomposition Nitrogen-To-DON Fraction

Select "13.1 Decomposition Nitrogen-To-DON fraction Plant Mortality Phenology" from the All Parameters drop-down menu to specify a parameter value for controlling DON production:



calibration

^psm_q$

13.1 Decomposition Nitrogen-To-DON Fraction

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	psm_q	0.05

Parameter Definitions

psm_q The fraction of decomposed nitrogen converted to DON range [0.0 – 1.0]

Calibration Notes

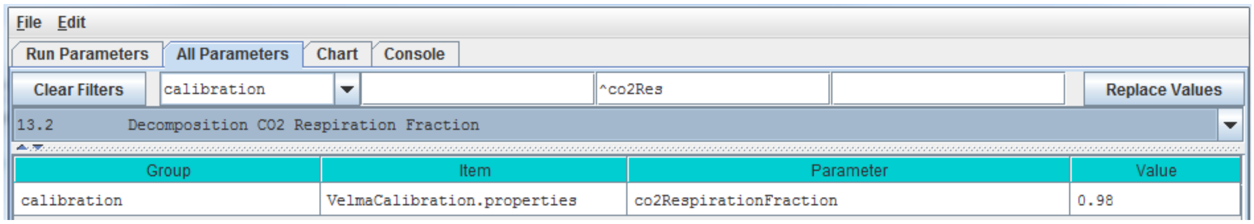
You will need to iteratively calibrate "psm_q" against observed stream DON concentrations (Abdelnour et al. 2013). This is most easily accomplished using VELMA's runtime visualization option for "Calibration Annual Nutrients". After annual results are in the ballpark, switch to "Calibration Daily Nutrients".

Reference:

Potter, C. S., Randerson, J. T., Field, C. B., Matson, P. A., Vitousek, P. M., Mooney, H. A., & Klooster, S. A. (1993). Terrestrial ecosystem production: a process model based on global satellite and surface data. Global Biogeochemical Cycles, 7(4), 811-841.

13.2 – Decomposition CO2 Respiration Fraction

Select “13.2 Decomposition CO2 Respiration Fraction” from the All Parameters drop-down menu to specify a parameter value controlling CO_2 production during decomposition:



calibration | ^co2Res

13.2 Decomposition CO2 Respiration Fraction

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	co2RespirationFraction	0.98

Parameter Definitions

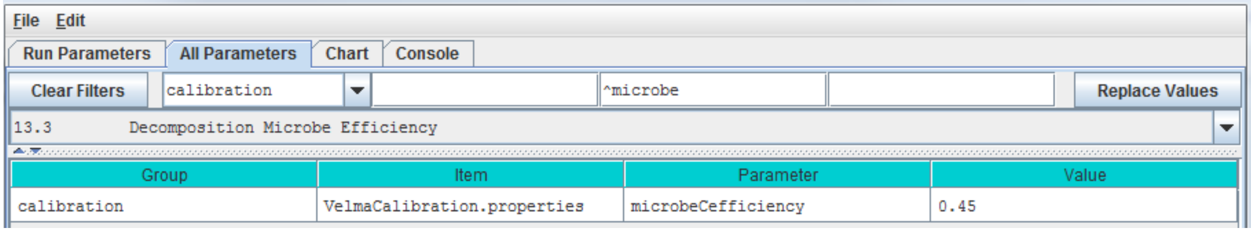
`co2RespriationFraction` The fraction of humus decomposition (in Carbon) lost to the atmosphere as CO2.

Calibration Notes

You will need to iteratively calibrate “co2RespriationFraction” in order to obtain a good fit between simulated and observed stream DOC concentrations, since DOC produced during decomposition = 1 - co2RespriationFraction. This is most easily accomplished using VELMA’s runtime visualization option for “Calibration Annual Nutrients”. After annual results are in the ballpark, switch to “Calibration Daily Nutrients”.

13.3 – Decomposition Microbe Efficiency

Select “13.3 Decomposition Microbe Efficiency” from the All Parameters drop-down menu to specify the microbial efficiency for carbon assimilation during decomposition:



calibration | ^microbe

13.3 Decomposition Microbe Efficiency

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	microbeCefficiency	0.45

Parameter Definitions

`microbeCefficiency` The carbon assimilation efficiency (0 – 1) of microbes. This is a globally-applicable term of the Potter equation.

As an example, an assimilation efficiency = 0.45 means that for every gram of carbon decomposed, 0.45 g is assimilated into the humus carbon pool (microbes are not explicitly modeled), and the remaining 0.55 g is respired as CO_2. However, a small fraction (1 – co2RespriationFraction) of this 0.55 g goes to the DOC pool.

Consult the references below for insight into typical values for microbeCefficiency.

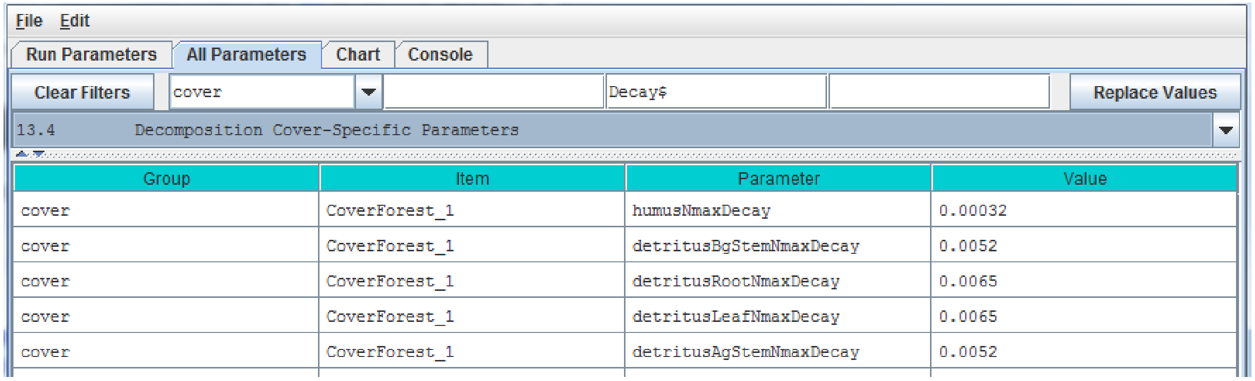
References:

Potter, C. S., Randerson, J. T., Field, C. B., Matson, P. A., Vitousek, P. M., Mooney, H. A., & Klooster, S. A. (1993). Terrestrial ecosystem production: a process model based on global satellite and surface data. Global Biogeochemical Cycles, 7(4), 811-841.

Herron, P. M., Stark, J. M., Holt, C., Hooker, T., & Cardon, Z. G. (2009). Microbial growth efficiencies across a soil moisture gradient assessed using ^{13}C-acetic acid vapor and ^{15}N-ammonia gas. Soil Biology and Biochemistry, 41(6), 1262-1269.

13.4 – Decomposition Cover-Specific Parameters

Select “13.4 Decomposition Cover-Specific Parameters” from the All Parameters drop-down menu to specify the microbial efficiency for carbon assimilation during decomposition. These parameters must be specified for each cover type.



cover | Decay$

13.4 Decomposition Cover-Specific Parameters

Group	Item	Parameter	Value
cover	CoverForest_1	humusNmaxDecay	0.00032
cover	CoverForest_1	detritusBgStemNmaxDecay	0.0052
cover	CoverForest_1	detritusRootNmaxDecay	0.0065
cover	CoverForest_1	detritusLeafNmaxDecay	0.0065
cover	CoverForest_1	detritusAgStemNmaxDecay	0.0052

Parameter Definitions

`humusNmaxDecay` The maximum rate of decay (0 – 1) for the humus N pool.

`detritusBgStemNmaxDecay` The maximum rate of decay (0 – 1) for the belowground stem (BgStem) N pool.

`detritusRootNmaxDecay` The maximum rate of decay (0 – 1) for the root N pool.

`detritusLeafNmaxDecay` The maximum rate of decay (0 – 1) for the leaf N pool.

`detritusAgStemNmaxDecay` The maximum rate of decay (0 – 1) for the aboveground stem (AgStem) N pool.

Calibration Notes

You will need to iteratively calibrate these decay constants, for each cover type, in order to obtain a good fit between simulated and observed detritus pool data. It is almost always best to calibrate for steady state (equilibrium) conditions, e.g., for mature cover types.

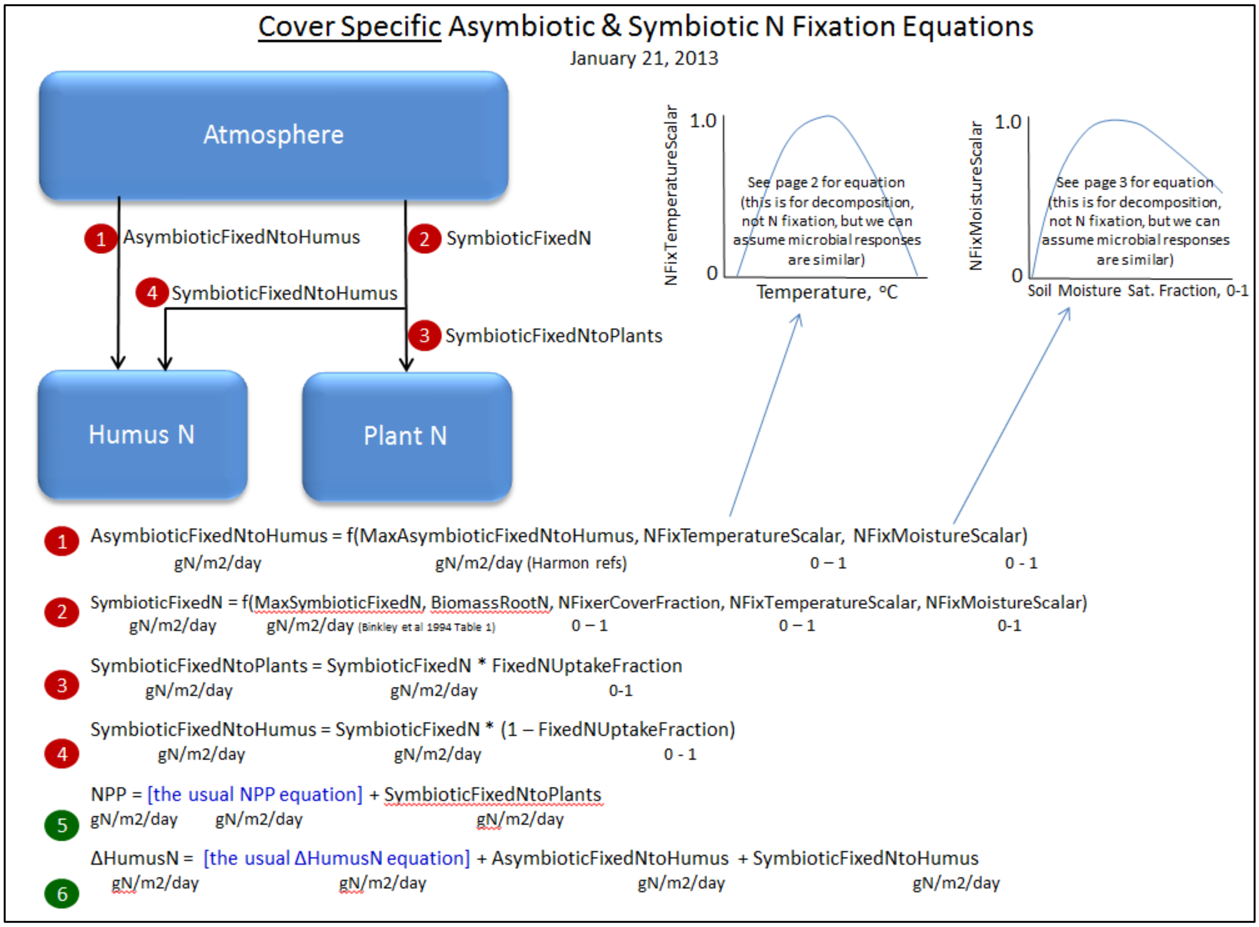
14.0 Nitrogen-Fixation (link to All Parameters TOC)

Nitrogen fixation is a process in which nitrogen (N_2) in the atmosphere is converted into ammonium (NH_4). In biological systems, N fixation can be carried out by free-living (asymbiotic) microorganisms, or by plant species that have a symbiotic relationship with N-fixing bacteria (*Frankia* and *Rhizobia*). In symbiotic N fixation, plants provide carbohydrates as an energy source to bacteria that in turn supply NH_4 to the host plant.

Rates of symbiotic N fixation for soybeans, red alder and other species can sometimes exceed 100 kg ha^{-1} yr^{-1}, an amount similar to agricultural fertilizer applications. Rates for asymbiotic N fixers tend to be much lower – for example, <1 kg ha^{-1} yr^{-1}in some PNW coniferous forests. However, it is important to capture even small N additions, which are important for maintaining and even increasing ecosystem N stocks over successional time scales.

We added a nitrogen fixation subroutine to VELMA v2.0 that addresses symbiotic and asymbiotic fixation for any cover type (agricultural, grassland, forest, etc.)

The conceptual diagram below shows VELMA’s modeled pools and fluxes. Note that parameter names in this diagram do not strictly follow those used in VELMA, but nonetheless convey the overall structure of the N fixation subroutine.



Cover Specific Asymbiotic & Symbiotic N Fixation Equations

January 21, 2013

Atmosphere
1 AsymbioticFixedNtoHumus
2 SymbioticFixedN
4 SymbioticFixedNtoHumus
3 SymbioticFixedNtoPlants
Humus N
Plant N
NFixTemperatureScalar
1.0
0
Temperature, °C
See page 2 for equation (this is for decomposition, not N fixation, but we can assume microbial responses are similar)
NFixMoistureScalar
1.0
0
Soil Moisture Sat. Fraction, 0-1
See page 3 for equation (this is for decomposition, not N fixation, but we can assume microbial responses are similar)

1 AsymbioticFixedNtoHumus = f(MaxAsymbioticFixedNtoHumus, NFixTemperatureScalar, NFixMoistureScalar)
gN/m2/day gN/m2/day (Harmon refs) 0 – 1 0 - 1

2 SymbioticFixedN = f(MaxSymbioticFixedN, BiomassRootN, NFixerCoverFraction, NFixTemperatureScalar, NFixMoistureScalar)
gN/m2/day gN/m2/day (Binkley et al 1994 Table 1) 0 – 1 0 – 1 0-1

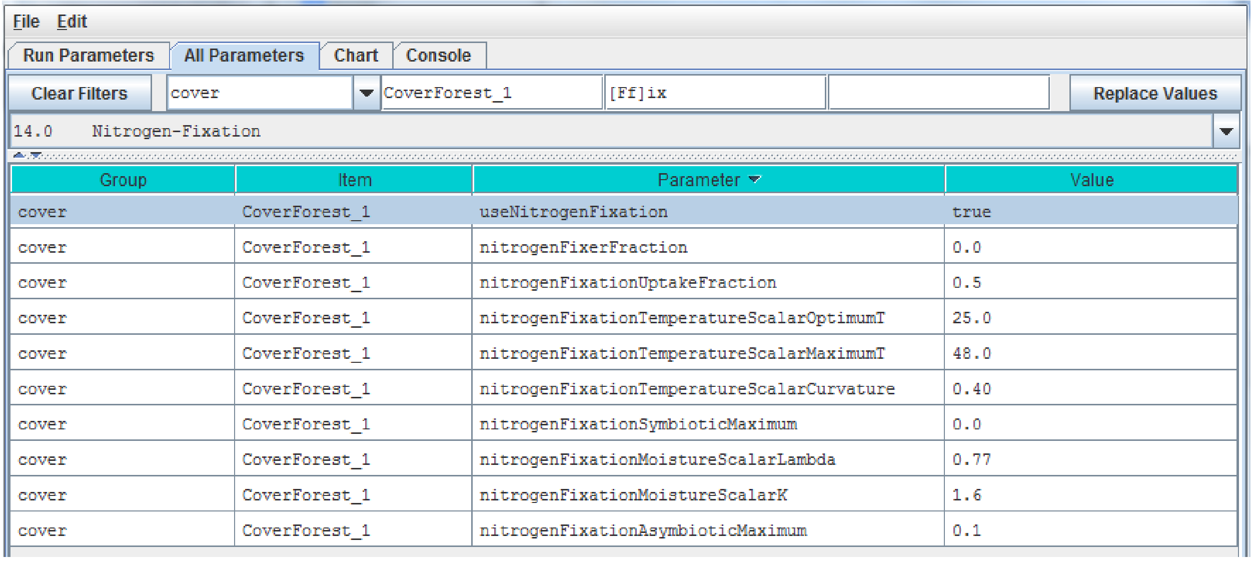
3 SymbioticFixedNtoPlants = SymbioticFixedN * FixedNUptakeFraction
gN/m2/day gN/m2/day 0-1

4 SymbioticFixedNtoHumus = SymbioticFixedN * (1 – FixedNUptakeFraction)
gN/m2/day gN/m2/day 0 - 1

5 NPP = [the usual NPP equation] + SymbioticFixedNtoPlants
gN/m2/day gN/m2/day gN/m2/day

6 ΔHumusN = [the usual ΔHumusN equation] + AsymbioticFixedNtoHumus + SymbioticFixedNtoHumus
gN/m2/day gN/m2/day gN/m2/day gN/m2/day

Select "14.0 Nitrogen-Fixation" from the All Parameters drop-down menu to specify parameter values for asymbiotic and symbiotic N fixation.



cover CoverForest_1 [Ff]ix

14.0 Nitrogen-Fixation

Group	Item	Parameter	Value
cover	CoverForest_1	useNitrogenFixation	true
cover	CoverForest_1	nitrogenFixerFraction	0.0
cover	CoverForest_1	nitrogenFixationUptakeFraction	0.5
cover	CoverForest_1	nitrogenFixationTemperatureScalarOptimumT	25.0
cover	CoverForest_1	nitrogenFixationTemperatureScalarMaximumT	48.0
cover	CoverForest_1	nitrogenFixationTemperatureScalarCurvature	0.40
cover	CoverForest_1	nitrogenFixationSymbioticMaximum	0.0
cover	CoverForest_1	nitrogenFixationMoistureScalarLambda	0.77
cover	CoverForest_1	nitrogenFixationMoistureScalarK	1.6
cover	CoverForest_1	nitrogenFixationAsymbioticMaximum	0.1

Parameter Definitions

`useNitrogenFixation` When using NitrogenFixation = false (default), no nitrogen fixation is computed for this cells of this cover. When useNitrogenFixation = true, nitrogen fixation is computed using this cover's other nitrogen fixation parameters.

`nitrogenFixerFraction` The fraction of a cell's plant matter (in the range [0.0 1.0]) that performs Nitrogen fixation.

`nitrogenFixationUptakeFraction` The fraction (in the range [0.0 1.0]) of the symbiotic fixed Nitrogen that is available for plant NPP. Plant symbiotic fixed nitrogen is: Total symbiotic fixed Nitrogen * nitrogenFixationUptakeFraction The remainder of the total is allocated as Humus symbiotic fixed nitrogen.

`nitrogenFixationTemperatureScalarOptimumT` Optimum temperature (in degrees C) parameter of GEM temperature scalar function for Nitrogen fixation.

`nitrogenFixationTemperatureScalarMaximumT` Maximum temperature (in degrees C) parameter of GEM temperature scalar function for Nitrogen fixation.

`nitrogenFixationTemperatureScalarCurvature` Curvature parameter ("qx") of GEM temperature scalar function for Nitrogen fixation

`nitrogenFixationSymbioticMaximum` The maximum amount of Nitrogen converted from the atmosphere by symbiotic processes in gNfixed/gFixerFineRootBiomassN per m2 per day.

`nitrogenFixationMoistureScalarLambda` The Lambda value of the Weibull equation used to compute a water scalar for Nitrogen fixation.

`nitrogenFixationMoistureScalarK` K value of the Weibull equation used to compute a water scalar for Nitrogen fixation.

`nitrogenFixationAsymbioticMaximum` The maximum amount of Nitrogen converted from the atmosphere by asymbiotic processes in gN/m2/day.

Calibration Notes

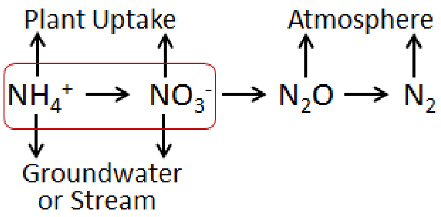
For initial calibration of the N fixation subroutine's temperature and moisture functions, we recommend starting with the values shown in the screenshot above. The cover-specific parameters will require iterative calibration against published data.

14.1 – Nitrogen-Fixation On/Off ?

See section 14.0, above, to specify whether to turn the N fixation subroutine on (true) or off (false). We've placed the on/off parameter there.

15.0 Nitrification (link to All Parameters TOC)

Nitrification is the microbially-mediated conversion of ammonium (NH_4^+) to nitrate (NO_3^-). As the diagram below indicates, nitrification (red box) is an important step in the soil nitrogen cycle (partially shown here).

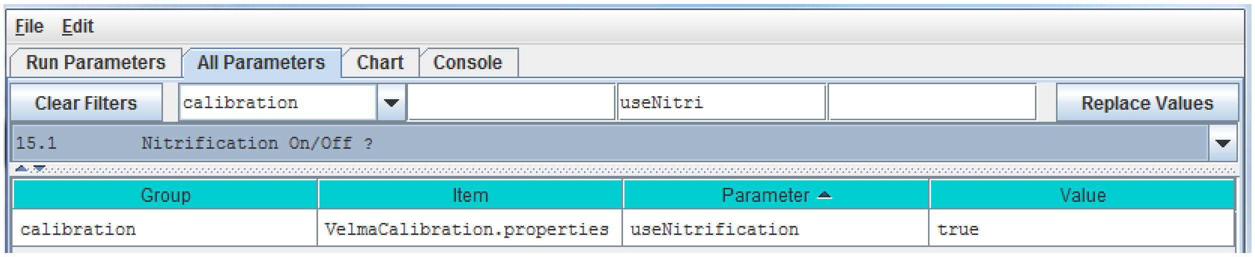


Compared to the positively charged NH_4^+ ion, the negatively charged NO_3^- ion is highly mobile in soil (also negatively charged) and more readily leached to groundwater and streams. Nitrification also affects rates of nitrogen uptake by plant species that differ in their affinities for NH_4^+ and NO_3^- (McKane et al. 2002). Nitrification is also a preliminary step for the process of denitrification, in which NO_3^- is microbially transformed to nitrous oxide (N_2O), a potent greenhouse gas, and nitrogen gas.

The default (recommended) version of the nitrification subroutine in VELMA v2.0 is based on Parton et al. 2001, with recent modifications made by Dr. Stephen Del Grosso (steve.delgrosso@ars.usda.gov) and coworkers. Dr. Del Grosso's modifications are described in his NGAS spreadsheet model.

15.1 – Nitrification On/Off

Select "15.1 Nitrification On/Off ?" from the All Parameters drop-down menu to specify whether to turn the Nitrification subroutine on (true) or off (false).



calibration | useNitri

15.1 Nitrification On/Off ?

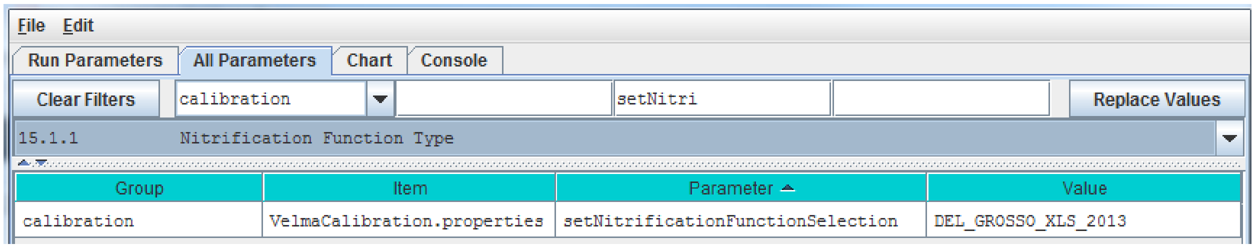
Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	useNitrification	true

Parameter Definitions

`useNitrogenFixation` When set false daily nitrification amounts are ignored and the PSM model behaves as if the nitrification amount is zero When set true (default) nitrification occurs normally

15.1.1 – Nitrification Function Type

Select "15.1.1 Nitrification Function Type" from the All Parameters drop-down menu to specify the



calibration | setNitri

15.1.1 Nitrification Function Type

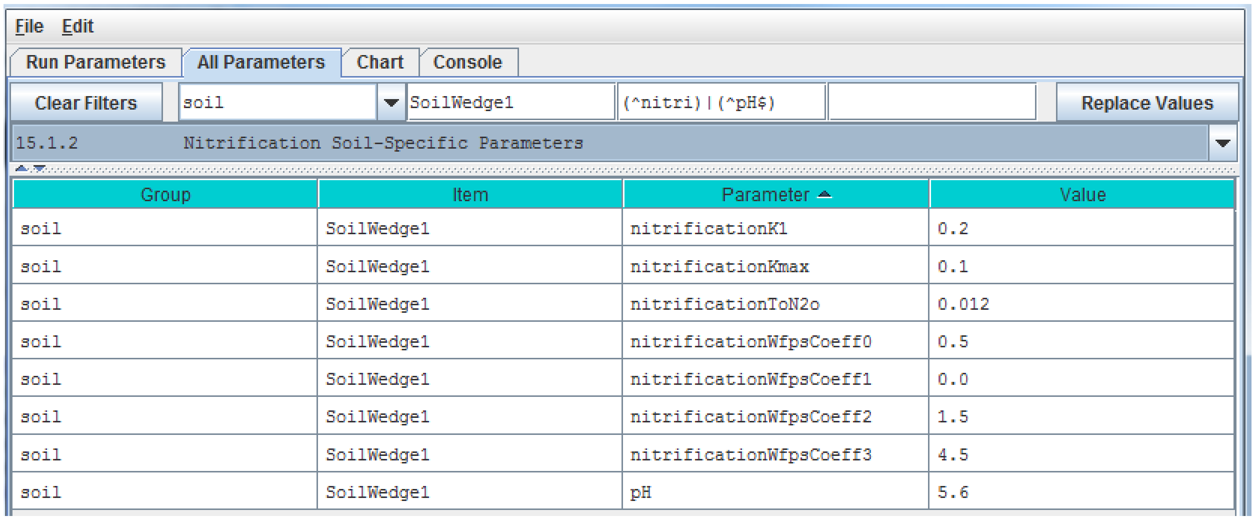
Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	setNitrificationFunctionSelection	DEL_GROSSO_XLS_2013

Parameter Definitions

`setNitrogenFunctionSelection` Specify "PARTON_1996" "PARTON_2001" or "DEL_GROSSO_XLS_2013" as the value of the setNitrificationFunctionSelection property. Warning: be sure to use UPPERCASE text! The simulator engine will compute nitrifcation amounts using the specified function. PARTON_1996 = Function based upon equation given in Figure 2 (d) in [Parton et al. 1996] from Global Biochemical Cycles v.10. PARTON_2001 = Function based upon equation (2) in [Parton et al. 2001] from Journal of Geophysical Research v.106. **DEL_GROSSO_XLS_2013 = (Default / RECOMMENDED)** function implementing the Excel 2013 spreadsheet model provided by Dr. Stephen Del Grosso.

15.1.2 – Nitrification Soil-Specific Parameters

Select "15.1.2 Nitrification Function Type" from the All Parameters drop-down menu to specify nitrification subroutine parameters. The parameters shown below must be specified for each soil type in your watershed (just one soil type shown in the screenshot below).



soil | SoilWedge1 | (^nitri)|(^pH$)

15.1.2 Nitrification Soil-Specific Parameters

Group	Item	Parameter ▲	Value
soil	SoilWedge1	nitrificationK1	0.2
soil	SoilWedge1	nitrificationKmax	0.1
soil	SoilWedge1	nitrificationToN2o	0.012
soil	SoilWedge1	nitrificationWfpsCoeff0	0.5
soil	SoilWedge1	nitrificationWfpsCoeff1	0.0
soil	SoilWedge1	nitrificationWfpsCoeff2	1.5
soil	SoilWedge1	nitrificationWfpsCoeff3	4.5
soil	SoilWedge1	pH	5.6

Parameter Definitions

nitrificationK1
Fraction of net Nitrogen mineralized that is nitrified per day range in [0.0 1.0]. [Parton et al. 2001].

nitrificationKmax
Maximum fraction of NH4 nitrified per day in units of gN/m2/day range in [0.0 1.0]. [Parton et al. 2001]

nitrificationToN2o
N2O flux associated with nitrification in units of gN/m2/day

nitrificationWfpsCoeff0
Soil-specific coefficient for Water Filled Pore Space (WFPS) effect on nitrification equation.

nitrificationWfpsCoeff1
Soil-specific coefficient for Water Filled Pore Space (WFPS) effect on nitrification equation.

nitrificationWfpsCoeff2
Soil-specific coefficient for Water Filled Pore Space (WFPS) effect on nitrification equation.

nitrificationWfpsCoeff3
Soil-specific coefficient for Water Filled Pore Space (WFPS) effect on nitrification equation.

pH	This soil's pH value.

Calibration Notes

The nitrification model has been extensively applied to agricultural systems by the NGAS model developers (Parton et al. 2001; Del Grosso et al. 2001 and 2006). We recommend starting with the default parameter values provided by Stephen Del Grosso in his NGAS spreadsheet model.

Default Parameter Values:

nitrificationK1 = 0.20 (Parton et al. 2001; Del Grosso NGAS spreadsheet model)

nitrificationKmax = 0.1 (Parton et al. 2001; Del Grosso NGAS spreadsheet model)

nitrificationToN2o = 0.02 (Parton et al. 2001; Del Grosso NGAS spreadsheet model = 0.012)

	Soil Texture		
Nitrification Parameter	Coarse	Fine	Volcanic ash
nitrificationWfpsCoeff0	0.5	0.65	0.75
nitrificationWfpsCoeff1	0	0	0
nitrificationWfpsCoeff2	1.5	1.2	1.2
nitrificationWfpsCoeff3	4.5	2.5	2.5

References for Sections 15.0 – 15.1.2

Del Grosso, S. J., Parton, W. J., Mosier, A. R., Ojima, D. S., Kulmala, A. E., & Phongpan, S. (2000). General model for N2O and N2 gas emissions from soils due to dentrification. Global Biogeochemical Cycles, 14(4), 1045-1060.

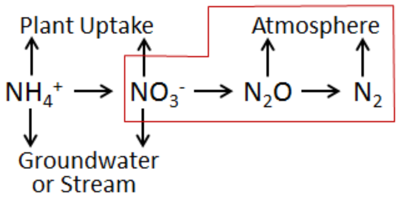
Del Grosso, S. J., Parton, W. J., Mosier, A. R., Walsh, M. K., Ojima, D. S., & Thornton, P. E. (2006). DAYCENT national-scale simulations of nitrous oxide emissions from cropped soils in the United States. Journal of environmental quality, 35(4), 1451-1460.

McKane, R. B., Johnson, L. C., Shaver, G. R., Nadelhoffer, K. J., Rastetter, E. B., Fry, B., ... & Murray, G. (2002). Resource-based niches provide a basis for plant species diversity and dominance in arctic tundra. Nature, 415(6867), 68-71.

Parton, W. J., Holland, E. A., Del Grosso, S. J., Hartman, M. D., Martin, R. E., Mosier, A. R., ... & Schimel, D. S. (2001). Generalized model for NO x and N2O emissions from soils. Journal of Geophysical Research: Atmospheres (1984–2012), 106(D15), 17403-17419.

16.0 Denitrification (link to All Parameters TOC)

Denitrification is the microbially-mediated conversion of nitrate (NO_3^-) to nitrous oxide (N_2O) and nitrogen gas (N_2). This process is represented in the red-outlined area in the diagram below.

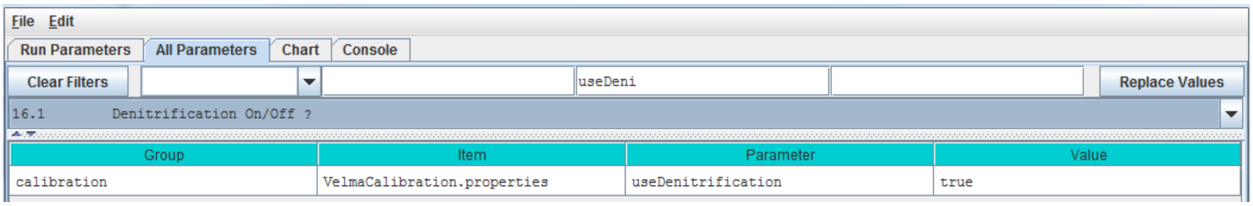


Nitrate that is not denitrified or taken up by plants can be readily transported to ground and surface waters (the high mobility of NO_3^- in soil is because both are negatively charged). Thus, denitrification can be important for protecting water quality. On the other hand, if denitrification does not go to completion (stops at N2O instead of N2) it can increase atmospheric inputs of N_2O, a potent greenhouse gas.

The denitrification subroutine in VELMA v1.0 and v2.0 is based on Del Grosso et al. 2001. However, VELMA v2.0 includes recent updates to equations and correction to parameter values described in Dr. Stephen Del Grosso's (steve.delgrosso@ars.usda.gov) NGAS spreadsheet model.

16.1 – Denitrification On/Off?

Select "16.1 Denitrification On/Off ?" from the All Parameters drop-down menu to specify whether to turn the denitrification subroutine on (true) or off (false).



useDeni

16.1 Denitrification On/Off ?

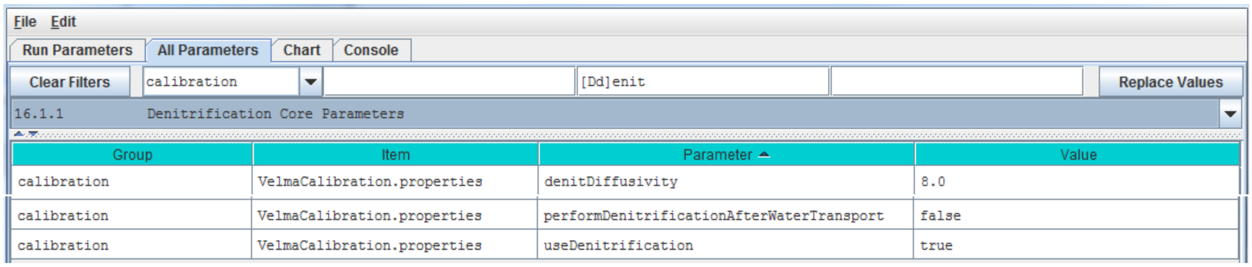
Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	useDenitrification	true

Parameter Definitions

`setNitrogenFunctionSelection` When set "false" daily denitrification amounts are ignored and VELMA's biogeochemistry submodel behaves as if the denitrification amount is zero. When set "true" (default) denitrification occurs normally.

16.1.1 – Denitrification Core Parameters

Select "16.1.1 Denitrification Core Parameters" from the All Parameters drop-down menu to specify denitrification subroutine parameter values. These global parameters are applicable to all soil types.



calibration

[Dd]enit

16.1.1 Denitrification Core Parameters

Group	Item	Parameter ▲	Value
calibration	VelmaCalibration.properties	denitDiffusivity	8.0
calibration	VelmaCalibration.properties	performDenitrificationAfterWaterTransport	false
calibration	VelmaCalibration.properties	useDenitrification	true

Parameter Definitions

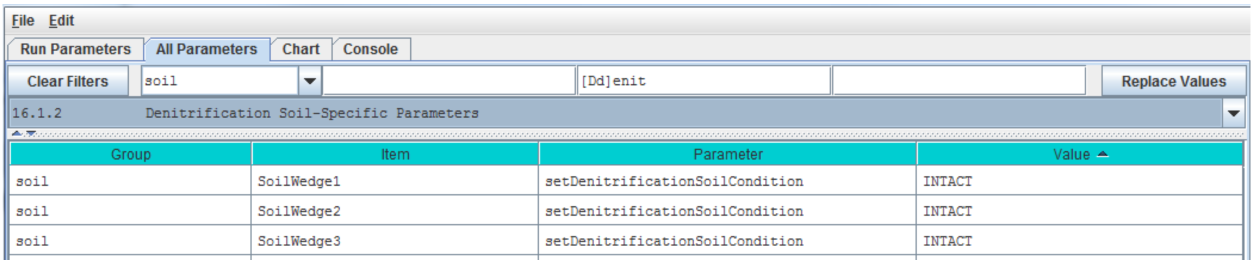
`denitDiffusivity` This is the value of the "a" parameter in Del Grosso et al. (2000) fig.7(a). It controls the shape of the logistic function that defines the response of "relative denitrification" (0.0 – 1.0) to water filled pore space (WFPS %). See calibration notes at the end of section 16.0, below.

`performDenitrificationAfterWaterTransport` When set "true" daily denitrification amounts are calculated AFTER vertical and lateral transportation of chemicals by water occurs. When set "false" (default) daily denitrification occurs BEFORE water transport.

`useDenitrification` When set “false” daily denitrification amounts are ignored and the PSM model behaves as if the denitrification amount is zero. When set “true” (default) denitrification occurs normally

16.1.2 – Denitrification Soil-Specific Parameters

Select “16.1.2 Denitrification Soil-Specific Parameters” from the All Parameters drop-down menu to specify the following set of denitrification parameters for each soil type in your watershed. There is just one parameter per soil type (3 soil types are shown in the screenshot below).



Group	Item	Parameter	Value
soil	SoilWedge1	setDenitrificationSoilCondition	INTACT
soil	SoilWedge2	setDenitrificationSoilCondition	INTACT
soil	SoilWedge3	setDenitrificationSoilCondition	INTACT

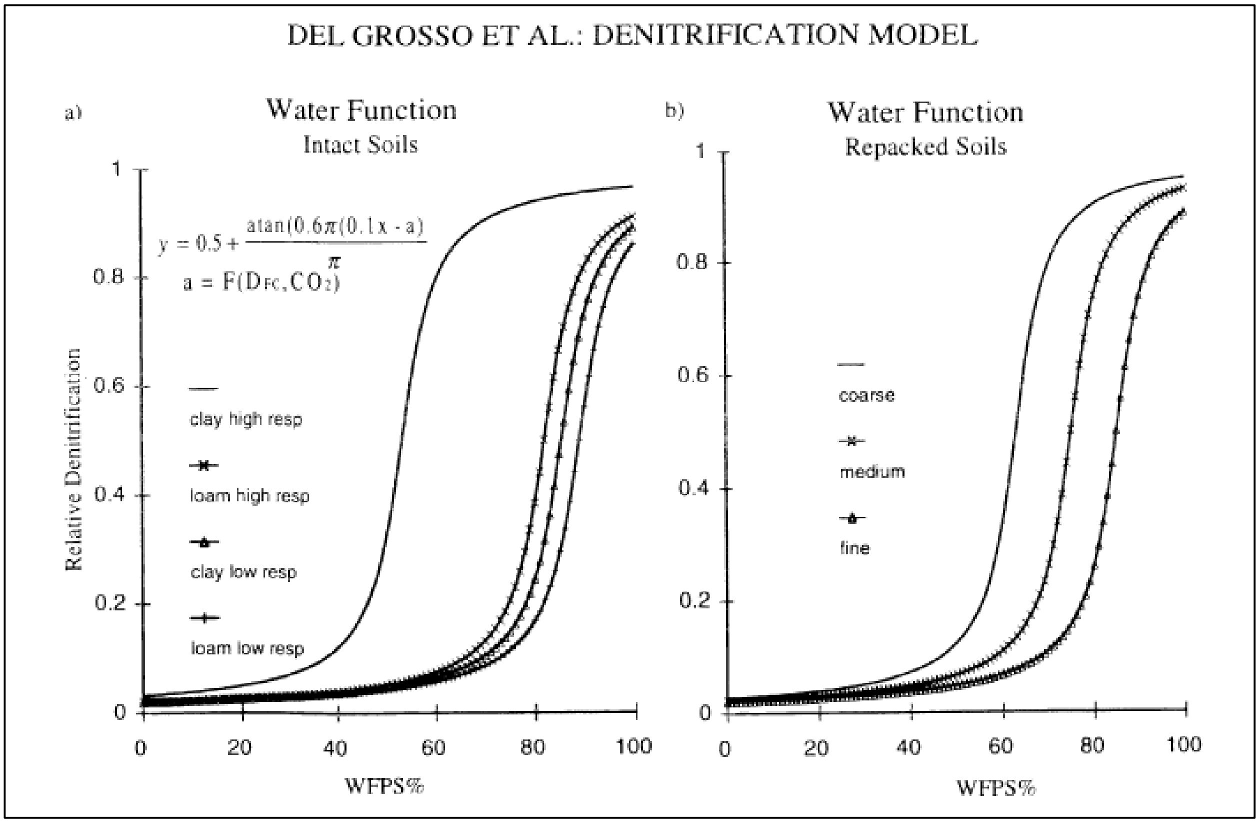
Parameter Definitions

`setDenitrificationSoilCondition` Defines soil condition with respect to denitrification calculations. Value must be one of the following: "INTACT" (the default value) or "REPACKED_COARSE" or "REPACKED_MEDIUM" or "REPACKED_FINE" The value is case-sensitive and must appear exactly as given here (but with the surrounding double-quotes). In practice, Del Grosso’s NGAS spreadsheet uses “INTACT” = undisturbed soils, and "REPACKED_MEDIUM” and "REPACKED_FINE" = medium or fine textured soil samples, respectively, that have been repacked in the lab after field collection.

Calibration Notes:

The denitrification subroutine has been extensively applied to agricultural systems by the NGAS model developers (Parton et al. 2001; Del Grosso et al. 2000 and 2006). The parameter values used in the model seem to be fairly robust across the wide range of conditions reported by the authors. When calibrating the denitrification subroutine for a new site, we recommend that you initially focus on just two parameters (assuming all the true/false options are set up correctly). These are the yellow-highlighted parameters above: `denitDiffusivity and setDenitrificationSoilCondition.`

The `denitDiffusivity` parameter controls the shape of the logistic function that defines the response of “relative denitrification” (0.0 – 1.0) to water filled pore space (WFPS %). Figure 7 in Del Grosso et al. (2000) illustrates this:



Thus, you will need to adjust the `denitDiffusivity` parameter ("a" in Figure 7) to obtain a response for the soil texture that is most appropriate for your watershed. You will also need to specify "INTACT" for undisturbed soils. "REPACKED_MEDIUM" or "REPACKED_FINE" refers to soil columns that have been reconstructed in the laboratory (Del Grosso et al. 2000). You will need to decide if the REPACKED choices are appropriate for compacted field soils, e.g., under high tillage practices.

References for section 16.0 (all subsections sections)

Del Grosso, S. J., Parton, W. J., Mosier, A. R., Ojima, D. S., Kulmala, A. E., & Phongpan, S. (2000). General model for N2O and N2 gas emissions from soils due to dentrification. Global Biogeochemical Cycles, 14(4), 1045-1060.

Del Grosso, S. J., Parton, W. J., Mosier, A. R., Walsh, M. K., Ojima, D. S., & Thornton, P. E. (2006). DAYCENT national-scale simulations of nitrous oxide emissions from cropped soils in the United States. Journal of environmental quality, 35(4), 1451-1460.

Parton, W. J., Holland, E. A., Del Grosso, S. J., Hartman, M. D., Martin, R. E., Mosier, A. R., ... & Schimel, D. S. (2001). Generalized model for NO x and N2O emissions from soils. Journal of Geophysical Research: Atmospheres (1984–2012), 106(D15), 17403-17419.

17.0 – Simulation Run Schedule (link to All Parameters TOC)

To let VELMA know when to start and stop a simulation, you specify need to specify values for the following parameters:

`forcing_start` First year of driver (forcing) data available for the simulation model run

`forcing_end`	Last year of driver (forcing) data available for the simulation model run
`syear`	Starting year of simulation model run
`eyear`	Ending year of simulation model run
`numberOfLoops`	The number of times (loops) to repeat a simulation (`syear` to `eyear` will repeat for the specified number of loops)

The Forcing Start and End Years Specify When a Simulation CAN Run

The VELMA Simulator expects every driver and observed data file specified in the simulation configuration to have 1 line of data for each day between the first day of the year specified for the `forcing_start` parameter and the last day of the year specified for the `forcing_end` parameter.

For example, if `forcing_start` = 2000 and `forcing_end` = 2001, each driver or observed data file must have 366 + 365 lines of data (one for each of the 731 days of years 2000 through 2001). The value specified by `forcing_start` must be less than or equal to the value of `forcing_end`.

The Simulation Start and End Years Specify When a Simulation WILL Run

The years that the VELMA Simulator actually runs a simulation for are specified by the `syear` (start year) and `eyear` (end year) simulation configuration parameters. It is an error for `syear`'s value to be greater than `eyear`'s, and for either value to lie outside the bounds of the years specified by `forcing_start` and `forcing_end`, however `syear` and `eyear` may specify a sub-range within that range.

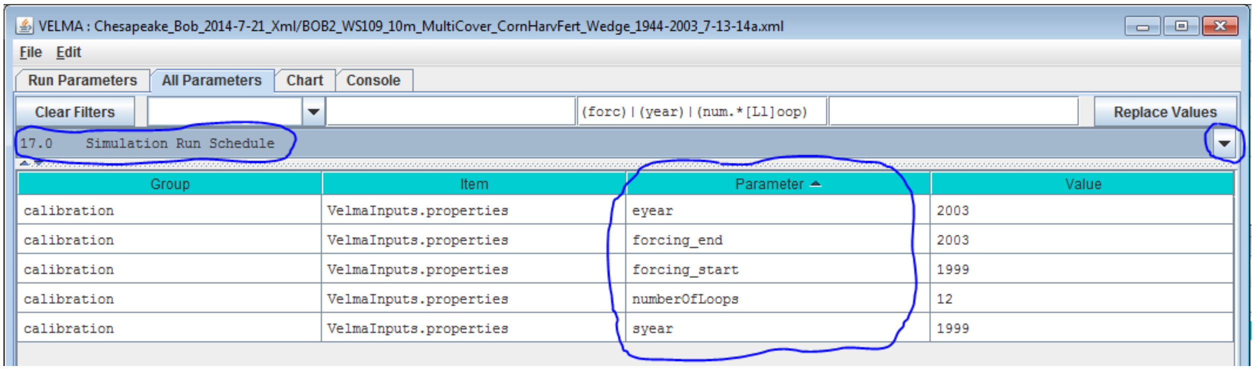
For example, if `[forcing_start, forcing_end]` = [1995, 2010], `[syear, eyear]` may be [1997, 2010], or even a single year, like [1998, 1998].

The Loops Parameter Specifies How Many Times the Simulation Repeats a Run

The `numberOfLoops` parameter defaults to "1", which tells the VELMA Simulator to run the specified simulation from start year to end year once, and then stop. Setting the `numberOfLoops` parameter to *n*, (with > 1) tells the VELMA simulator to run the simulation from start year to end year n times in a row, but to only initialize the simulation before the first start-to-end run. In this way, the `numberOfLoops` parameter provides a (simplistic) spin-up mechanism. The second start-to-end run after the first will begin with the simulation's data state where it was at the end of the first start-to-end run, and so on.

Use the All Parameters Outline Selector to Focus on the Simulation Run Schedule Parameters

You can filter the All Parameters tab's parameters table to focus on the simulation run schedule by selecting item "17.0 Simulation Run Schedule" in the outline dropdown selector:


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Run Parameters
All Parameters
Chart
Console
Clear Filters
Replace Values

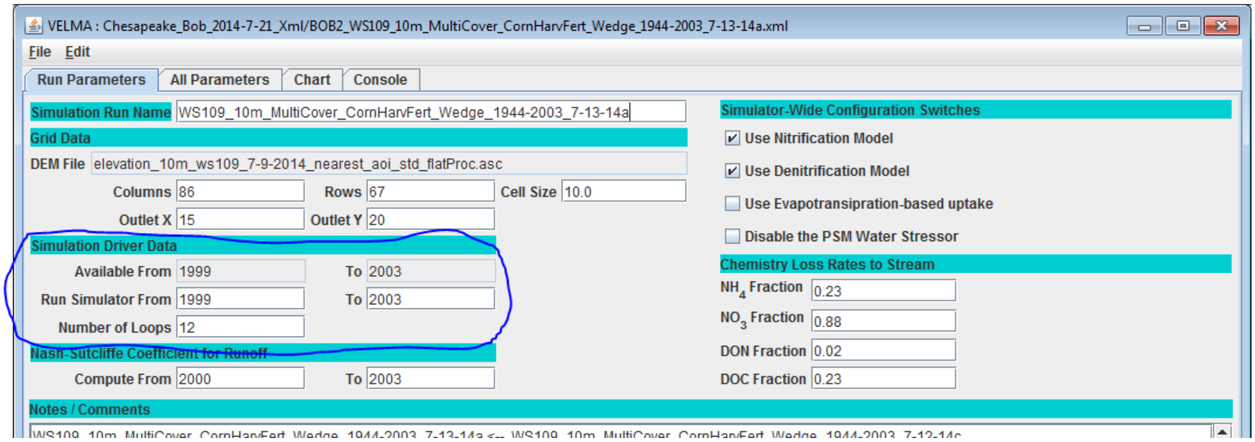
(forc)|(year)|(num.*[Ll]oop)

17.0 Simulation Run Schedule

Group	Item	Parameter ▲	Value
calibration	VelmaInputs.properties	eyear	2003
calibration	VelmaInputs.properties	forcing_end	2003
calibration	VelmaInputs.properties	forcing_start	1999
calibration	VelmaInputs.properties	numberOfLoops	12
calibration	VelmaInputs.properties	syear	1999

You Can Also Review and Set Some of these Values in the Run Parameters Tab

Compare circled fields in the Run Parameters tab below with the parameters circled in the All Parameters table above. Setting the “Run Simulator From” and “To” fields in the Run Parameters tab sets the values of the syear and eyear parameters. Setting the Run Parameters “Number of Loops” value sets the value of the numberOfLoops parameter. Note however, that you can only set the forcing_start and forcing_end values in the All Parameters table. These values are read-only fields in the Run Parameters tab. The screenshot above happens to specify 12 loops – i.e. run the simulation [1999, 2003] x12 – that’s a significant spin up. Generally speaking, find out how long one simulation loop takes to run before embarking on that many multiple loops.


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Run Parameters
All Parameters
Chart
Console

Simulation Run Name: WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a

Grid Data

DEM File: elevation_10m_ws109_7-9-2014_nearest_aoi_std_flatProc.asc

Columns: 86 Rows: 67 Cell Size: 10.0

Outlet X: 15 Outlet Y: 20

Simulation Driver Data

Available From: 1999 To: 2003

Run Simulator From: 1999 To: 2003

Number of Loops: 12

Nash-Sutcliffe Coefficient for Runoff

Compute From: 2000 To: 2003

Simulator-Wide Configuration Switches

☑ Use Nitrification Model

☑ Use Denitrification Model

☐ Use Evapotranspiration-based uptake

☐ Disable the PSM Water Stressor

Chemistry Loss Rates to Stream

NH_4 Fraction: 0.23

NO_3 Fraction: 0.88

DON Fraction: 0.02

DOC Fraction: 0.23

Notes / Comments

WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a <-- WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-12-14c

18.0 – Years to Compute Nash-Sutcliffe for Runoff (link to All Parameters TOC)

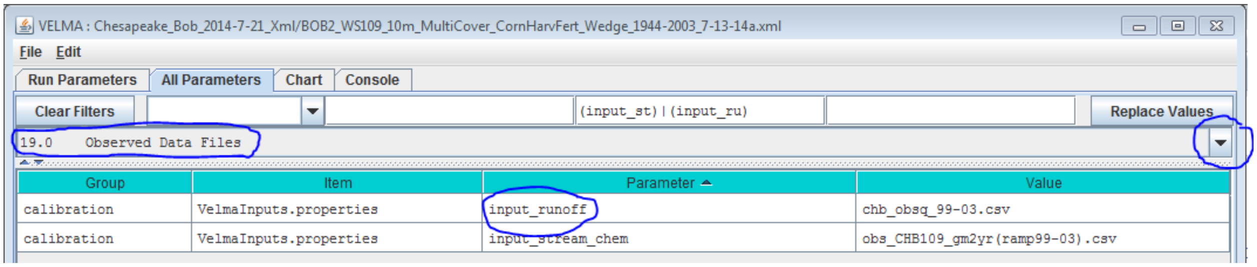
The VELMA Simulator Calculates a Nash-Sutcliffe Value for Runoff

Whenever possible, the VELMA simulator automatically computes a Nash-Sutcliffe Coefficient for observed vs. simulated annual runoff values. In order to calculate the runoff Nash-Sutcliffe, the simulator must know which span of years to include in the calculation, and have observed runoff data for those years available.

To ensure that the VELMA simulator is able to compute a meaningful Nash-Sutcliffe value for your simulation run, provide values for the following configuration parameters.

Set the Observed Runoff Source File

Click the All Parameters tab’s configuration outline dropdown selector and select item “19.0 Observed Data Files”. The All Parameters table should then look like this:


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Run Parameters
All Parameters
Chart
Console
Clear Filters
Replace Values

(input_st)|(input_ru)

19.0 Observed Data Files

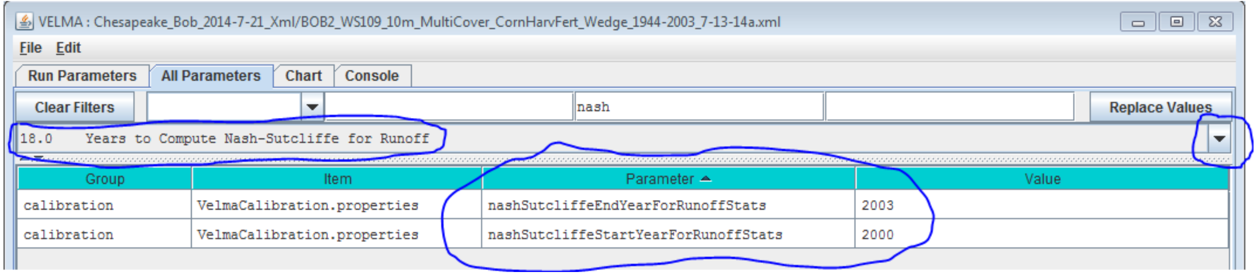
Group	Item	Parameter ▲	Value
calibration	VelmaInputs.properties	input_runoff	chb_obsq_99-03.csv
calibration	VelmaInputs.properties	input_stream_chem	obs_CHB109_gm2yr(ramp99-03).csv

Set the input_runoff parameter to the name of a .csv file containing valid observed runoff data. A valid file has a single column of floating-point data, one day’s observed runoff value per row. There must be exactly as many rows in the file as there are days in the simulation configuration’s

`forcing_start` to `forcing_end` parameter's span of years. (E.g. if `forcing_start` = 2000 and `forcing_end` = 2001, then there must be 366 + 365 or 731 rows of data in the file.)
The specified file's location is assumed to be the directory specified by the `inputDataLocationRootName` + `inputDatalocationDirName`.

Set the Years to Simulation the Nash-Sutcliffe

Click the All Parameters outline dropdown selector again, and this time select item "18.0 Years to Compute Nash-Sutcliffe for Runoff". The All Parameters table should then look like this:



VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Run Parameters
All Parameters
Chart
Console
Clear Filters
nash
Replace Values
18.0 Years to Compute Nash-Sutcliffe for Runoff

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	nashSutcliffeEndYearForRunoffStats	2003
calibration	VelmaCalibration.properties	nashSutcliffeStartYearForRunoffStats	2000

The `nashSutcliffeStartYearForRunoffStats` and `nashSutcliffeEndYearForRunoffStats` specify the first and last years of the simulation run that are used to compute the Nash-Sutcliffe coefficient. Obviously, these start and end year values must be within the range of the simulation run's start and end years as a whole. For example, if your simulation runs from 1995 to 2005, you cannot specify years 1980 to 2009 as the span for computing the Nash-Sutcliffe.

The VELMA Simulator Writes the Nash-Sutcliffe Coefficient for Runoff to an Output File

The VELMA simulator writes the Nash-Sutcliffe coefficient for runoff to a file named "NashSutcliffeCoefficients.txt". This is a simple text file, created in the directory specified by the simulation configuration's `intializeOutputDataLocationRoot` parameter.
The file contains a single line, reporting the Nash-Sutcliffe result like this:

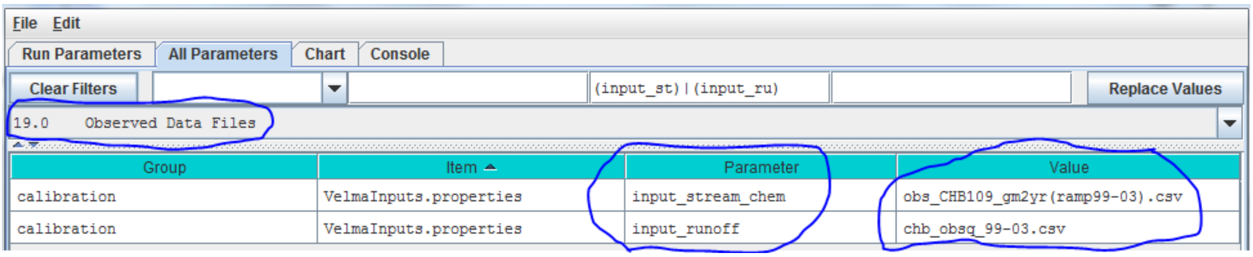
```
Nash-Sutcliffe Coefficient=0.8045351825866978 Loop=1 Years=[1969 to 2008]
```

The Computed Nash-Sutcliffe Value is Only as Accurate as the Observed Data

The auto-computation of the Nash-Sutcliffe coefficient for runoff is convenient, but it is only as accurate as the observed data and year's span allow it to be.

19.0 – Observed Data Files (link to All Parameters TOC)

If you want VELMA to graphically display observed stream chemistry and runoff data along with simulated values during a simulation (section 25.0), and calculate the Nash-Sutcliffe efficiency coefficient (section 18.0), you will need to prepare input files and specify their names in the All Parameters outline drop-down menu for section "19.0 Observed Data Files":

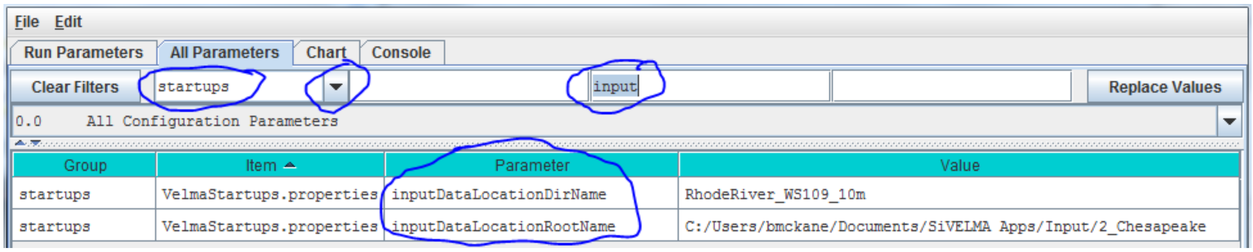


Group	Item	Parameter	Value
calibration	VelmaInputs.properties	input_stream_chem	obs_CHB109_gm2yr(ramp99-03).csv
calibration	VelmaInputs.properties	input_runoff	chb_obsq_99-03.csv

For the `input_runoff` parameter, specify the name of a file (.csv format) containing daily observed streamflow (runoff) values in mm/day. The data must be formatted as one value per line in the file, and the file must have as many lines as there are days between the simulator configuration's specified `forcing_start` and `forcing_end` parameter values (see "All Parameters 17.0 – Simulation Run Schedule")

For the `input_chem` parameter, specify the name of a file (.csv format) containing observed daily stream chemistry values for DON, NH_4, DOC and NO_3, **in that order**! Units are g N m^{-2} day^{-1}, or g C m^{-2} day^{-1}.

Both input files must be placed in the simulation configuration's input file directory location (`inputDataLocationDirName`) and name (`inputDataLocationRootName`). These input data parameters can be found by selecting "Startups" under the drop-down Filter menu tab (type "input" in the filter for the "Parameter" column if you wish to focus the selection on just these two parameters:



Group	Item	Parameter	Value
startups	VelmaStartups.properties	inputDataLocationDirName	RhodeRiver_WS109_10m
startups	VelmaStartups.properties	inputDataLocationRootName	C:/Users/bmckane/Documents/SiVELMA Apps/Input/2_Chesapeake

20.0 – Simulation-End Data Capture for Spin-up Initialization Use (link to All Parameters TOC)

Skip to section 21.0 where the procedure for section 20.0 is included as part of the "Simulation-Start Initialization from Spin-up Data" procedure.

21.0 – Simulation-Start Initialization from Spin-up Data (link to All Parameters TOC)

This section outlines a technique for using the ending state of a VELMA simulator run's spatial data pool values as the initial starting state for the spatial data pools of a second simulation run.

(1) Create a common simulation configuration base .xml document

The base .xml simulation configuration is your overall simulation configuration, with DEM, soil, cover, etc. parameters set as you wish them to be for site and conditions you are simulating.

(2) Make two copies of the base simulation configuration .xml document

One of the two copies will be modified to be the spin up configuration. The other will be the actual, final configuration.

(3) Modify the Spin up configuration to save spatial data state at the end of its simulation run

1. Load the Spin up configuration into JVelma.
2. Click to the "All Parameters" tab and select "All startups" from the drop-down list of filters. That is, use the drop-down button ▼ located above the "Group" column, not the "Value" column. You may need to click Clear Filters button first to get rid of any filtering text already present).
3. In the All Parameter "ID Key" column, locate the `setEndStateSpatialDataLocationName` parameter, then click-click into its "Value" field and type in the name of a directory

where the simulation can store spatial data map (".asc") files.
Note that:
A. The directory name may be either a fully-qualified path, or a relative path.
If it's a relative path, JVelma assumes its root is the output directory specified by the simulation configuration's /calibration/VelmaInputs.properties/initializeOutputDataLocationRoot parameter.
B. Use "/" path separators in the directory name.
C. The directory does not need to exist beforehand; JVelma will try to create it when it's needed during the simulation run.

4. Click to the "Run Parameters" tab and set the "Run Simulator From" and "To" values to the years you want the Spin up to run. (You may also wish to set the Nash-Sutcliffe coefficient "Compute From" and "To" values at this point.)

Save the Spin up configuration .xml file.

(4) Modify the Actual configuration to initialize its spatial data state at simulation start.

1. Load the Actual configuration into JVelma.
2. Click the "All Parameters" tab and select "All Startups" from the drop-down list of filters (you may first need to click the "Clear Filters" button).
3. In the All Parameters "ID Key" column, locate the `setStartStateSpatialDataLocationFileName` parameter, then click-click into its "Value" field and type in the name of a directory where the simulation can look for spatial data map (".asc") files.
   Note that:
   A. Presumably, the directory path will be to the directory specified as the `setEndStateSpatialDataLocationFileName` value in the Spin up configuration file.
   B. The directory name may be either a fully-qualified path, or a relative path.
   If it's a relative path, JVelma assumes its root is the output directory specified by the simulation configuration's `inputDataLocationRootName/inputDataLocationDirName` parameters.
   C. Use "/" path separators in the directory name.
   D. The directory MUST exist at the time that the Actual simulation starts running.
4. Click to the "Run Parameters" tab and set the "Run Simulator From" and "To" values to the years you want the Spin up to run. (You may also wish to set the Nash-Sutcliffe coefficient "Compute From" and "To" values at this point.)

Save the Actual configuration .xml file.

(5) Run the Spin up simulation to completion.

Start JVelma, load the Spin up configuration .xml and Start the simulation running.
Wait until the Spin up simulation completes its run before proceeding.

(6) Run the Actual simulation to completion.

Start JVelma, load the Actual configuration .xml and Start the simulation running.

When the Actual configuration completes, the Actual simulation run's results will be data based on an initial spatial data state taken from the final spatial data state of the Spin up simulation run.

22.0 – Cell Data Writer Items (link to All Parameters TOC)

Cell Data Writers Allow You to Gather Simulation Results for a Specific Grid Location

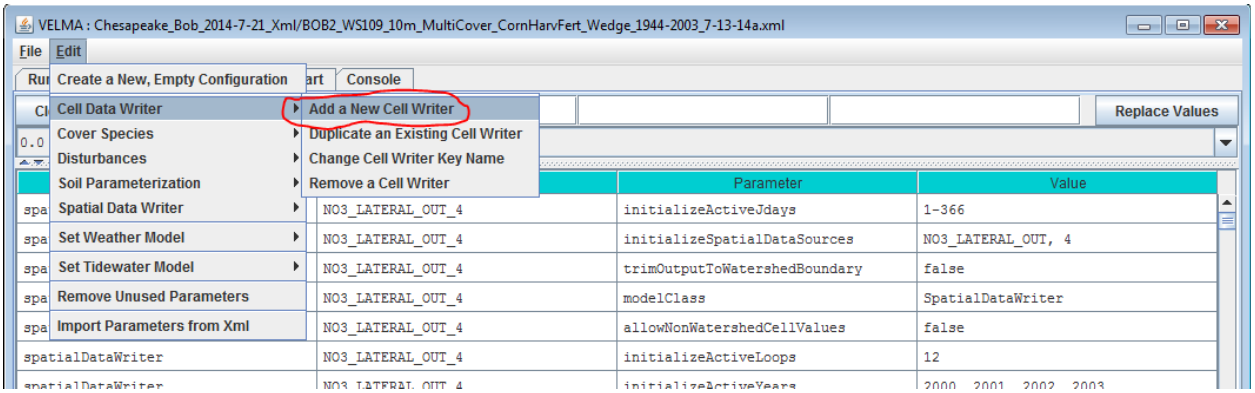
The VELMA simulator automatically provides daily simulation results for various values (e.g. Leaf Biomass), however these daily results are average values that are computed by summing individual cell values and then dividing the sum by the number of cells in the simulation's delineated watershed.

To gather daily simulation results for a single, specific cell, you need to add a Cell Data Writer parameterization for that cell to the simulation configuration.

Adding a Cell Data Writer to a Simulation Configuration

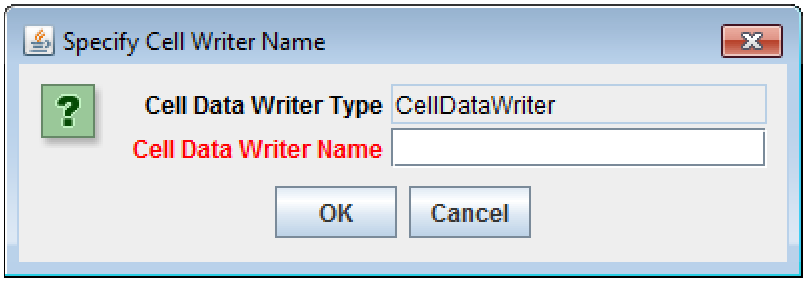
Cell Data Writers are optional. Simulation configurations contain zero instances of them by default. When added, they do not change the simulation computations – they are only involved in reporting results.

To add a Cell Data Writer to your simulation configuration:
Click the "Edit" → "Cell Data Writer" → "Add a New Cell Writer" menu item.


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Create a New, Empty Configuration
Cell Data Writer
Cover Species
Disturbances
Soil Parameterization
Spatial Data Writer
Set Weather Model
Set Tidewater Model
Remove Unused Parameters
Import Parameters from Xml
Add a New Cell Writer
Duplicate an Existing Cell Writer
Change Cell Writer Key Name
Remove a Cell Writer
Console
Replace Values

		Parameter	Value
spa	NO3_LATERAL_OUT_4	initializeActiveJdays	1-366
spa	NO3_LATERAL_OUT_4	initializeSpatialDataSources	NO3_LATERAL_OUT, 4
spa	NO3_LATERAL_OUT_4	trimOutputToWatershedBoundary	false
spa	NO3_LATERAL_OUT_4	modelClass	SpatialDataWriter
spa	NO3_LATERAL_OUT_4	allowNonWatershedCellValues	false
spatialDataWriter	NO3_LATERAL_OUT_4	initializeActiveLoops	12
spatialDataWriter	NO3_LATERAL_OUT_4	initializeActiveYears	2000, 2001, 2002, 2003

Clicking "Add a New Cell Writer" opens the Cell Writer Name dialog, which looks like this:



Cell Data Writer Type: CellDataWriter
Cell Data Writer Name:

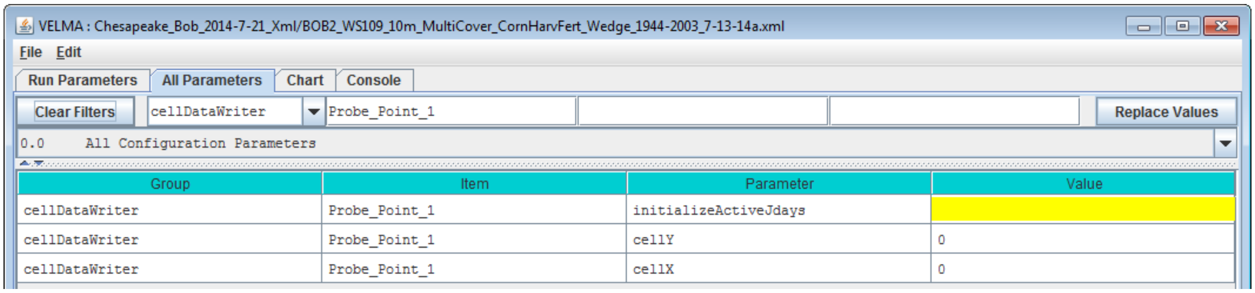
Enter a name for your new Cell Data Writer and click "OK".

The name must be unique (i.e. no other Cell Data Writer already specified for this simulation configuration can share the name you specify) and we recommend avoiding whitespace, and punctuation characters (e.g. "(" and ")"). An acceptable example name (assuming it's not already in use by another Cell Data Writer might be "Probe_Point_1" or maybe "`Outlet_Cell`".

Configuring a Cell Data Writer's Parameters

After you click OK in the Cell Data Writer's naming dialog, the VELMA GUI adds the Cell Data Writer to the simulation configuration, and sets the All Parameters tab's filters to display only the parameters of the newly-added Cell Data Writer.

Assuming we named our new Cell Data Writer "Probe_Point_1" and clicked OK, the VELMA GUI would look like this afterwards:


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Run Parameters
All Parameters
Chart
Console
Clear Filters
Replace Values

cellDataWriter

Probe_Point_1

0.0 All Configuration Parameters

Group	Item	Parameter	Value
cellDataWriter	Probe_Point_1	initializeActiveJdays	
cellDataWriter	Probe_Point_1	cellY	0
cellDataWriter	Probe_Point_1	cellX	0

Notice, in passing, that the configuration outline does *not* automatically get set by the GUI. In the example screen capture above, it displays "0.0 All Configuration Parameters" – which is not what is being displayed in the parameterization table. This behavior is harmless.

A Cell Data Writer has only 3 parameters, but they must all be set correctly: none are optional.

Set the `initializeActiveJdays` parameter to the Julian days that you want data reported for. If you want data reported on every day of the year (this is the most common case) set the `initializeActiveJdays` value to be "1-366". (The 366 avoids missing the last day in any leap years that occur in the simulation run.)

If you only want data reported on specific days of the year, enter the Julian Day values as a comma-separated list (e.g. "1, 90, 180, 270" reports data on days 1, 90, 180 and 270). You may specify an inclusive range of days by separating a start-day value from an end-day value by a dash (e.g. ("1, 90-180, 270" reports data on days 1, days 90 through 180, and on day 270).

Set the `cellX` parameter's value to the X-coordinate (i.e. column value) of the cell you want data reported for. The valid range is from 0 (the leftmost column) to (number of columns – 1).

Set the `cellY` parameter's value to the Y-coordinate (i.e. row value) of the cell you want data reported for. The valid range is from 0 (the topmost row) to (number of rows – 1).

Cell Data Writer Output is A Comma-Separated Values File

The VELMA simulator creates 1 .csv file for each valid Cell Data Writer parameterization in the simulation configuration .xml file.

Cell Data Writer output files are written to the Results Data Location folder. Their filenames always begin with the prefix "Cell_" and contain the linear index, x-coordinate and y-coordinate of the cell they contain data for (e.g. `Cell_i2873_x35_y33.csv`).

Each row of a Cell Data Writer output file contains results data for a specific loop, year and Julian day during the simulation run. The file's columns contain the specific results. The header row of the file specifies the contents of each column. Note that some columns contain data that never varies (e.g. the `DEM_Elevation(m)` column reports the cell's elevation – which is always the same value.)

Cell Data Writers and Spatial Data Writers are Different, but Complementary

A Cell Data Writer reports all the results data available for a specific cell on user-specified days. A Spatial Data Writer reports a specific result for all the cells in the simulation watershed on user-specified days. Both are optional for simulation runs.

23.0 – Spatial Data Writer Items (link to All Parameters TOC)

Spatial Data Writers Report a Simulation Result for Every Cell in the Simulation Grid

Spatial Data Writers are how the VELMA simulation provides spatially-explicit results output.

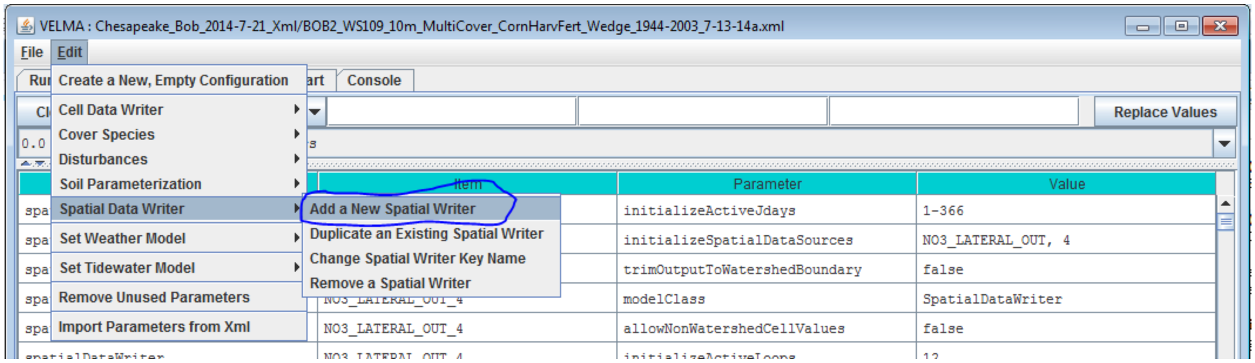
Spatial Data Writers are layer- and result-specific. You must configure separate spatial data writer parameterizations for each result type you will to output, and for layer of that result type. (E.g. to get spatial data output for the Root Biomass pool layers 3 and 4, you would need to add and configure two separate Spatial Data Writer parameterizations to your simulation configuration.

23.1 – Spatial Data Writer Items by Model Type

Adding a Spatial Data Writer to a Simulation Configuration

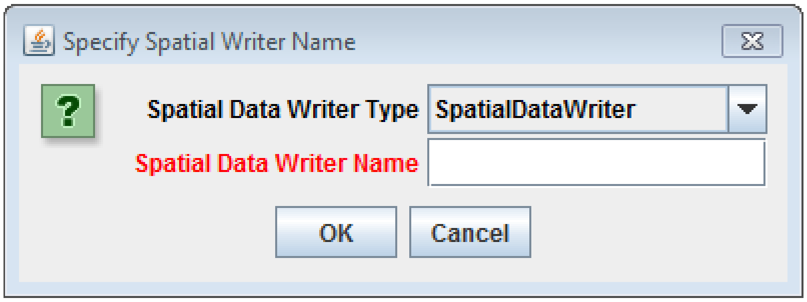
Spatial Data Writers are optional. Simulation configurations contain zero instances of them by default. When added, they do not change the simulation computations – they only report results.

To add a Spatial Data Writer to your simulation configuration:
Click the "Edit" → "Spatial Data Writer" → "Add a New Spatial Writer".


VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml
File
Edit
Create a New, Empty Configuration
Cell Data Writer
Cover Species
Disturbances
Soil Parameterization
Spatial Data Writer
Set Weather Model
Set Tidewater Model
Remove Unused Parameters
Import Parameters from Xml
Add a New Spatial Writer
Duplicate an Existing Spatial Writer
Change Spatial Writer Key Name
Remove a Spatial Writer
Console
Replace Values

Item	Parameter	Value
	initializeActiveJdays	1-366
	initializeSpatialDataSources	NO3_LATERAL_OUT, 4
	trimOutputToWatershedBoundary	false
NO3_LATERAL_OUT_4	modelClass	SpatialDataWriter
NO3_LATERAL_OUT_4	allowNonWatershedCellValues	false
NO3_LATERAL_OUT_4	initializeActiveLoops	12

Clicking "Add a New Spatial Writer" opens the Spatial Writer Type and Name dialog which looks like this:


Specify Spatial Writer Name

Spatial Data Writer Type: SpatialDataWriter

Spatial Data Writer Name:

Leave the Spatial Data Writer Type selector set to "SpatialDataWriter".

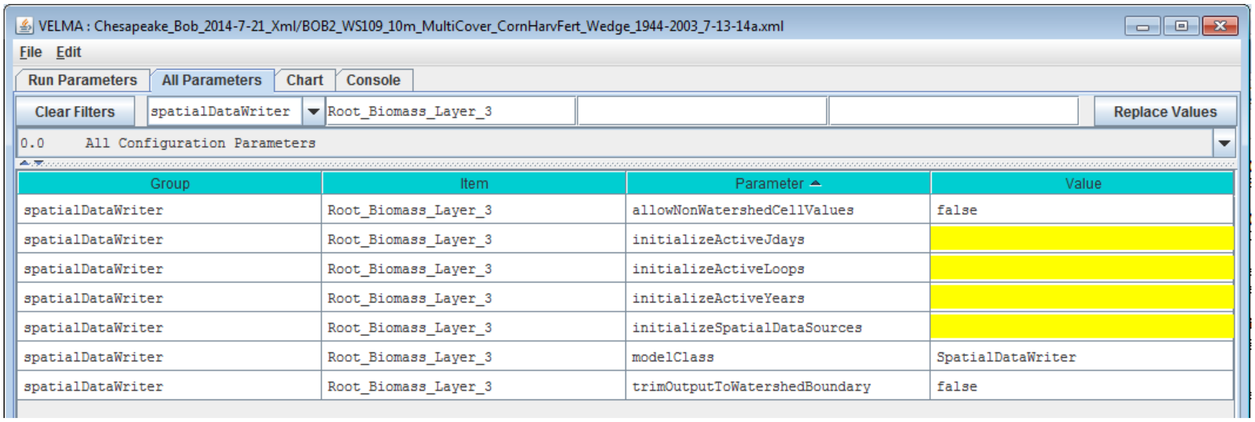
The Spatial Data Writer Name must be unique (i.e. no other Spatial Data Writer already specified for this simulation configuration can share the name you specify). Avoid whitespace and punctuation characters (e.g. "(" and ")"). As an example, here are a couple of acceptable names (assuming they're not already in use by another Spatial Data Writer, and the Spatial Data Writer we're adding will report layer 3 root biomass; "Root_Biomass_Layer_3" and "Root_L3".

23.2 – Spatial Data Writer Items Data Sources

Configuring a Spatial Data Writer's Parameters

After you click OK in the Spatial Data Writer's Type and Name dialog, the VELMA GUI adds the Spatial Data Writer to the simulation configuration, and sets the All Parameters tab's filters to display only the parameters of the newly-added Spatial Data Writer.

Assuming we named our new Spatial Data Writer "Root_Biomass_Layer_3" and clicked OK, the VELMA GUI would look like this afterwards:



VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml

spatialDataWriter ▼ Root_Biomass_Layer_3

0.0 All Configuration Parameters

Group	Item	Parameter ▲	Value
spatialDataWriter	Root_Biomass_Layer_3	allowNonWatershedCellValues	false
spatialDataWriter	Root_Biomass_Layer_3	initializeActiveJdays	
spatialDataWriter	Root_Biomass_Layer_3	initializeActiveLoops	
spatialDataWriter	Root_Biomass_Layer_3	initializeActiveYears	
spatialDataWriter	Root_Biomass_Layer_3	initializeSpatialDataSources	
spatialDataWriter	Root_Biomass_Layer_3	modelClass	SpatialDataWriter
spatialDataWriter	Root_Biomass_Layer_3	trimOutputToWatershedBoundary	false

Notice, in passing, that the configuration outline does not automatically get set by the GUI. In the example screen capture above, it displays "0.0 All Configuration Parameters" – which is not what is being displayed in the parameterization table. This behavior is harmless.

A Spatial Data Writer has 7 parameters. None are optional, but one (`modelClass`) is set automatically by the GUI, and two others (`allowNonWatershedCellValues` and `trimOutputToWatershedBoundary`) default to reasonable values automatically. The remaining 4 parameters specify when this parameterization writes its specified spatial data source to file, and what that spatial data source should be.

The Spatial Data Sources Parameter Specifies What Data Is Written

Set the `initializeSpatialDataSources` parameter to the name of the spatial data source you want to report results for. The name is case-sensitive, and must exactly match a name from the Table of Spatial Data Sources given below. Be careful to avoid leading or trailing whitespace when you enter the spatial data sources name; leading or trailing whitespace can "break" the simulation configuration.

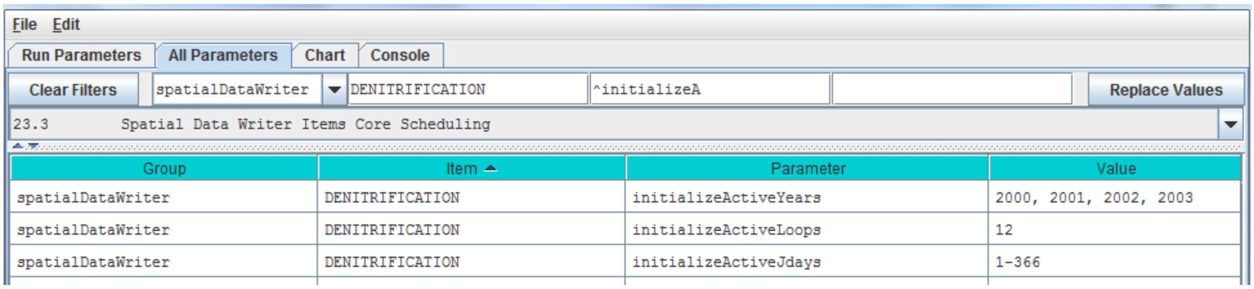
If the spatial data source you choose is multi-layered (see the table), you may (in some instances, *must*) explicitly specify the layer you wish to write. Specify an explicit layer by adding it, comma-separated, after the spatial data source name. The range of valid layer numbers is [1 to 4].

For example: "`BIOMASS_ROOT_N`" specifies the spatial data source name for biomass root, and "`BIOMASS_ROOT_N, 2`" specifies biomass root data, layer 2 (enter these, and all parameterization values, ***without*** the surrounding quotation (") marks).

23.3 – Spatial Data Writer Items Core Scheduling

The Three "InitializeActive…" Parameters Specify When Data Is Written

Select "23.3 Spatial Data Writer Items Core Scheduling" from the All Parameters drop-down outline selector menu:



spatialDataWriter | DENITRIFICATION | ^initializeA

23.3 Spatial Data Writer Items Core Scheduling

Group	Item ▲	Parameter	Value
spatialDataWriter	DENITRIFICATION	initializeActiveYears	2000, 2001, 2002, 2003
spatialDataWriter	DENITRIFICATION	initializeActiveLoops	12
spatialDataWriter	DENITRIFICATION	initializeActiveJdays	1-366

Set the `initializeActiveLoops`, `initializeActiveYears` and `initializeActiveJdays` parameter values to the loop numbers, years and Julian day values that you want the Spatial Data Writer to write results for.

Any specified Loop numbers, years or Julian days that are "beyond bounds" are ignored. I.e. if the simulation configuration specifies 3 loops, and the Spatial Data Writer `initializeActiveLoops` value is set to 4, no data will ever be written. It is an error to specify negative numbers for Loop, Year or Jday parameters.

[TIP: Setting an initializeActiveLoops value greater than the number of loops the simulation configuration specifies run is an unintuitive yet handy way to temporarily "shut off" a spatial data writer's output without removing it from the simulation configuration.]

You may specify more than one value by separating values with commas, and you may specify an inclusive range of values by separating the start and end values by dashes.
(E.g. specifying "`1, 3-7, 12`" for `initializeActiveLoops` would allow the spatial data writer to write data in loops 1, loops 3 through 7, and loop 12 – assuming the simulation configuration's `numberOfLoops` parameter is set to 12 or more.)

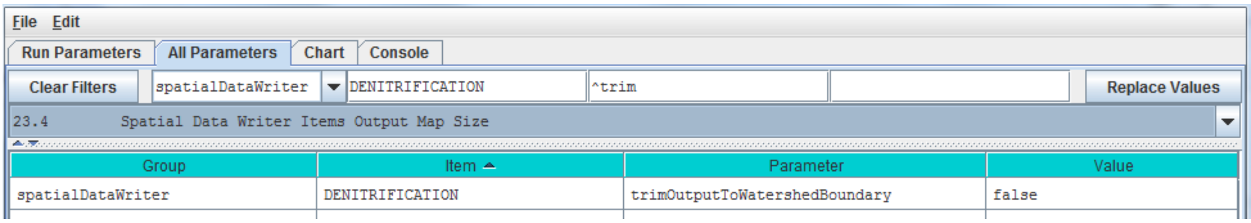
You May Emphasize the Watershed Cells, or Not

When the `allowNonWatershedCellValues` parameter is set `false`, its spatial data writer writes any cell value outside the watershed as NO_DATA (the value "`-9999.0`"). The default value is `false`, which makes it easy to discern the watershed's delineation in the results file. If you want the results file to include every cell value, regardless of whether a given cell is in or out of the watershed delineation, set this parameter to `true`.

23.4 – Spatial Data Writer Items Output Map Size

You May Reduce the Dimensions of the Results Grid File

Select "23.4 Spatial Data Writer Items Output Map Size" from the All Parameters drop-down outline selector menu:



spatialDataWriter | DENITRIFICATION | ^trim

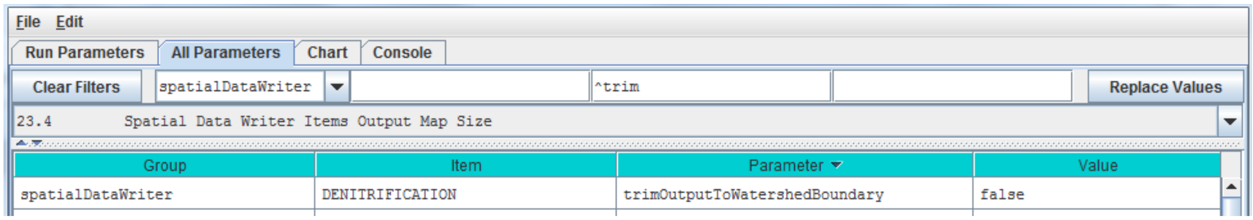
23.4 Spatial Data Writer Items Output Map Size

Group	Item ▲	Parameter	Value
spatialDataWriter	DENITRIFICATION	trimOutputToWatershedBoundary	false

When `trimOutputToWatershedBoundary` is set `false`, the spatial data writer writes a results file with the same grid dimensions at the simulation configuration's DEM file. This (default value) make it easy to overly Spatial Data Writer output results on top of the DEM, cover Id map, or soil Id map files. When set to `true`, the spatial data writer writes a results file that contains only the cells within the bounding rectangle of the delineated watershed. Depending upon the size of the watershed relative to the overall DEM, this may result in Spatial Data Writer files that are smaller, and easier to review "by hand".

[TIP: Spatial Data Writer output results files are not really meant for "manual" review (e.g. via Notepad or a similar simple text editor). The choice of the Grid ASCII format was for ease of use with GIS systems, and so that they could be readily "fed back" into the VELMA Simulator as subsequent initialization input. However, there's nothing wrong with opening a results file in a text editor or spreadsheet for a quick sanity check.]

To reduce the Dimensions of the Results Grid File, click the All Parameters tab's configuration outline dropdown selector and select "23.4 Spatial Data Writer Items Output Map Size".



spatialDataWriter ^trim

23.4 Spatial Data Writer Items Output Map Size

Group	Item	Parameter	Value
spatialDataWriter	DENITRIFICATION	trimOutputToWatershedBoundary	false

Spatial Data Writer Output is a Grid ASCII File

The VELMA simulator creates 1 .asc file each time a given Spatial Data Writer is triggered (based on its active Loops, Years and Jdays parameter settings).

Spatial Data Writer output files are written to the Results Data Location folder. Their filenames always begin with the prefix "Spatial_" and include the name of the spatial data source and layer number of the data they contain. The dimensions of the Grid ASCII file match those of the simulation configuration's DEM .asc file, unless the `trimOutputToWatershedBoundary` parameter was set "true"; in that case, the dimensions of the Spatial Data Writer output file are those of the bounding box of the simulation run's delineated watershed.

Each data value in the output .asc file contains the value for the specified data source for a particular cell, on the particular loop, year and Julian day during the simulation run that the output file was written.

Spatial Data Writers and Spatial Data Writers are Different, but Complementary

A Spatial Data Writer reports a specific result for all the cells in the simulation watershed on user-specified days. A Spatial Data Writer reports all the results data available for a specific cell on user-specified days. Both are optional for simulation runs.

Table of Spatial Data Sources

This table lists the available spatial data source name and their layer requirements.

SpatialDataSource Keyword means:

The spatial data source keyword names that are valid values for the `initializeSpatialDataSources` parameter value of a Spatial Data Writer's parameterization.

"Type" means:

- Accessor = Spatial Data that is not directly stored in a spatial data pool object.
- Direct = Spatial Data that is directly stored in a spatial data pool object.
- Buffered = Spatial Data directly stored in a spatial data pool object, but only when a Spatial Data Writer requires it.

(In practice, you can ignore what type a spatial data source is when you are configuring a Spatial Data Writer.)

“Can Sum Column?” means:

- N/A = Not applicable
- No = The data cannot be summed: if MultiLayer is “YES” for this type of data, then a layer must be explicitly provided in the `initializeSpatialDataSources` parameter value.
- Optional = If MultiLayer is “YES” for this type of data and no explicit layer number is indicated in the initializeSpatialDataSources parameter value, the value reported will be the sum of all layer values at a given cell’s location.
- Always = The spatial data type is MultiLayer, but only summed data is available; no explicit layer value is allowed in the initializeSpatialDataSources parameter value.

Type	SpatialDataSource Keyword	MultiLayer?	Can Sum Column?
Accessor	AirTemperature		N/A
Accessor	CellWriter		N/A
Accessor	CoverAge		N/A
Accessor	CoverId		N/A
Accessor	DenitrificationLayer	YES	No
Accessor	FlowAccumulation		No
Accessor	No3DenitrificationFactorLayer	YES	No
Accessor	QetLayer	YES	No
Accessor	Rain		N/A
Accessor	Snow		N/A
Accessor	SoilId		N/A
Accessor	SurfaceElevation		N/A
Accessor	TotalBiomassCarbon		Always
Accessor	TotalBiomassNitrogen		Always
Accessor	TotalDetritusCarbon		Always
Accessor	TotalDetritusNitrogen		Always
Accessor	TotalHumusCarbon		Always
Accessor	TotalVolumetricSoilMoisture		Always

Accessor	VolumetricSoilMoistureLayer	YES	No
Direct	ACCUMULATED_DEGREE_DAY		N/A
Direct	ACCUMULATED_N_IN		N/A
Direct	ASYMBIOTIC_NITROGEN_FIXED		N/A
Direct	BIOMASS_AG_STEM_N		N/A
Direct	BIOMASS_APEX_TOTAL_STEM_N		N/A
Direct	BIOMASS_BG_STEM_N		N/A
Direct	BIOMASS_DELTA_AG_STEM_N		N/A
Direct	BIOMASS_DELTA_BG_STEM_N		N/A
Direct	BIOMASS_DELTA_LEAF_N		N/A
Direct	BIOMASS_DELTA_ROOT_N	YES	Optional
Direct	BIOMASS_HARVESTED_TO_OFFSITE_C		N/A
Direct	BIOMASS_LEAF_N		N/A
Direct	BIOMASS_ROOT_N	YES	Optional
Direct	CO2	YES	Optional
Direct	DENITRIFICATION	YES	Optional
Direct	DENITRIFICATION_CO2_SCALAR	YES	Optional
Direct	DENITRIFICATION_NO3_SCALAR	YES	Optional
Direct	DENITRIFICATION_WATER_SCALAR	YES	Optional
Direct	DETRITUS_AG_STEM_N		N/A
Direct	DETRITUS_BG_STEM_N	YES	Optional
Direct	DETRITUS_BURNED_C		N/A
Direct	DETRITUS_LEAF_N		N/A
Direct	DETRITUS_ROOT_N	YES	Optional
Direct	DOC	YES	Optional
Direct	DON	YES	Optional
Direct	GROUND_SURFACE_TEMPERATURE		N/A
Direct	GROUND_TEMPERATURE_LAYERS	YES	Optional
Direct	HUMUS	YES	Optional
Direct	LATERAL_INFLOW	YES	Optional
Direct	LATERAL_OUTFLOW	YES	Optional
Direct	NH4	YES	Optional
Direct	NITRIFICATION	YES	Optional

Direct	NO3	YES	Optional
Direct	NPP_C		N/A
Direct	NPP_N		N/A
Direct	STANDING_WATER		N/A
Direct	SURFACE_LATERAL_OUTFLOW		N/A
Direct	SYMBIOTIC_NITROGEN_FIXED_HUMUS		N/A
Direct	SYMBIOTIC_NITROGEN_FIXED_PLANT		N/A
Direct	UPTAKE_N	YES	Optional
Direct	WATER_STORED	YES	Optional
Buffered	DOC_LATERAL_IN	YES	Optional
Buffered	DOC_LATERAL_OUT	YES	Optional
Buffered	DOC_LOSS	YES	Optional
Buffered	DON_LATERAL_IN	YES	Optional
Buffered	DON_LATERAL_OUT	YES	Optional
Buffered	DON_LOSS	YES	Optional
Buffered	FERTILIZATION_HUMUS_N_ADDED		N/A
Buffered	FERTILIZATION_NH4_ADDED		N/A
Buffered	FERTILIZATION_NO3_ADDED		N/A
Buffered	NH4_LATERAL_IN	YES	Optional
Buffered	NH4_LATERAL_OUT	YES	Optional
Buffered	NH4_LOSS	YES	Optional
Buffered	NO3_LATERAL_IN	YES	Optional
Buffered	NO3_LATERAL_OUT	YES	Optional
Buffered	NO3_LOSS	YES	Optional

24.0 – Disturbance Items   (link to All Parameters TOC)

Understanding how disturbances such as harvest, fire and fertilization affect ecosystem services has been a major motivation in the development of VELMA. For example, how do disturbances such as forest harvest or the application of agronomic fertilizers affect hydrological and biogeochemical processes controlling water quality and quantity, carbon sequestration, production of greenhouse gases, etc.? Abdelnour et al. (2011, 2013) have already demonstrated the use of VELMA v1.0 to simulate the effects of forest clearcutting on ecohydrological processes that regulate a variety of ecosystem services.

With the addition of a tissue-specific plant biomass (LSR) simulator and an enhanced GUI, VELMA v2.0 significantly expands the detail, flexibility, and ease of use for simulating disturbance effects. Currently available disturbance models include:

- `BurnDisturbanceModel` → effects of fire
- `GrazeDisturbanceModel` → effects of grazing
- `FertilizeLsrDisturbanceModel` → effects of fertilizer applications
- `HarvestLsrDisturbanceModel` → effects of biomass harvest

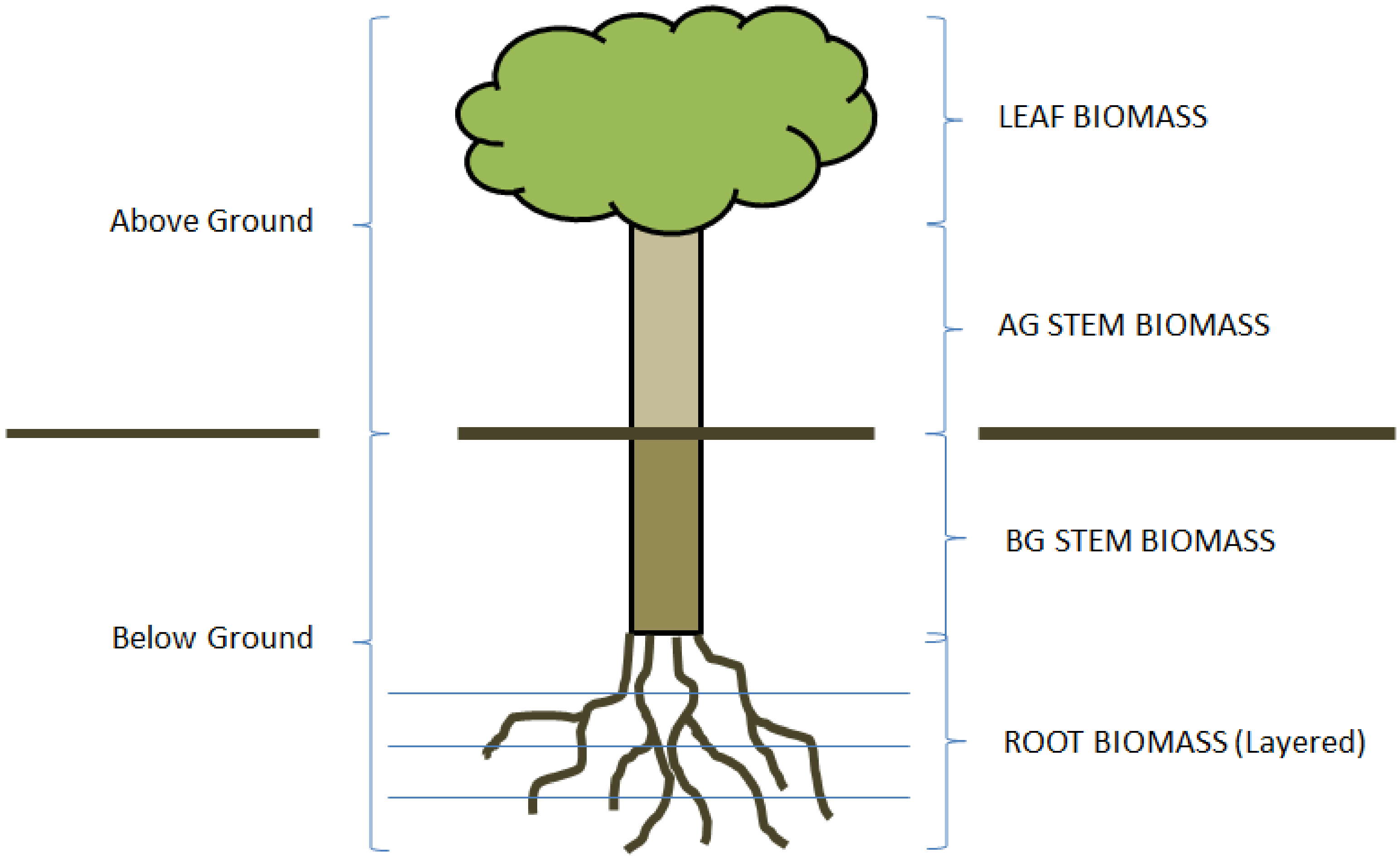
Each of these disturbance models specifies where and when a disturbance event will occur. The Burn, Graze and Harvest models have options for specifying how much of each plant tissue and detritus pool (leaves, stems, roots) will be removed and where it goes (offsite and/or to a specified onsite C and N pools). The Fertilize model has options for applying nitrogen as ammonium, nitrate, urea and/or manure.

Requirements for implementing a disturbance model include (1) a grid ASCII map of your watershed and associated parameter specifications that determine where and when a disturbance will occur, and (2) parameter specifications that determine which plant biomass and detritus pools will be affected, and the fraction of each pool burned, grazed or harvested and its destination. Portions of harvested aboveground stem biomass can be exported offsite or transferred to the aboveground stem detritus pool.

Sections 24.1 – 24.3 describe how to select a disturbance model type and specify parameter values to simulate where, when and how disturbances occur. You can set up as many disturbance types and events as you may require.

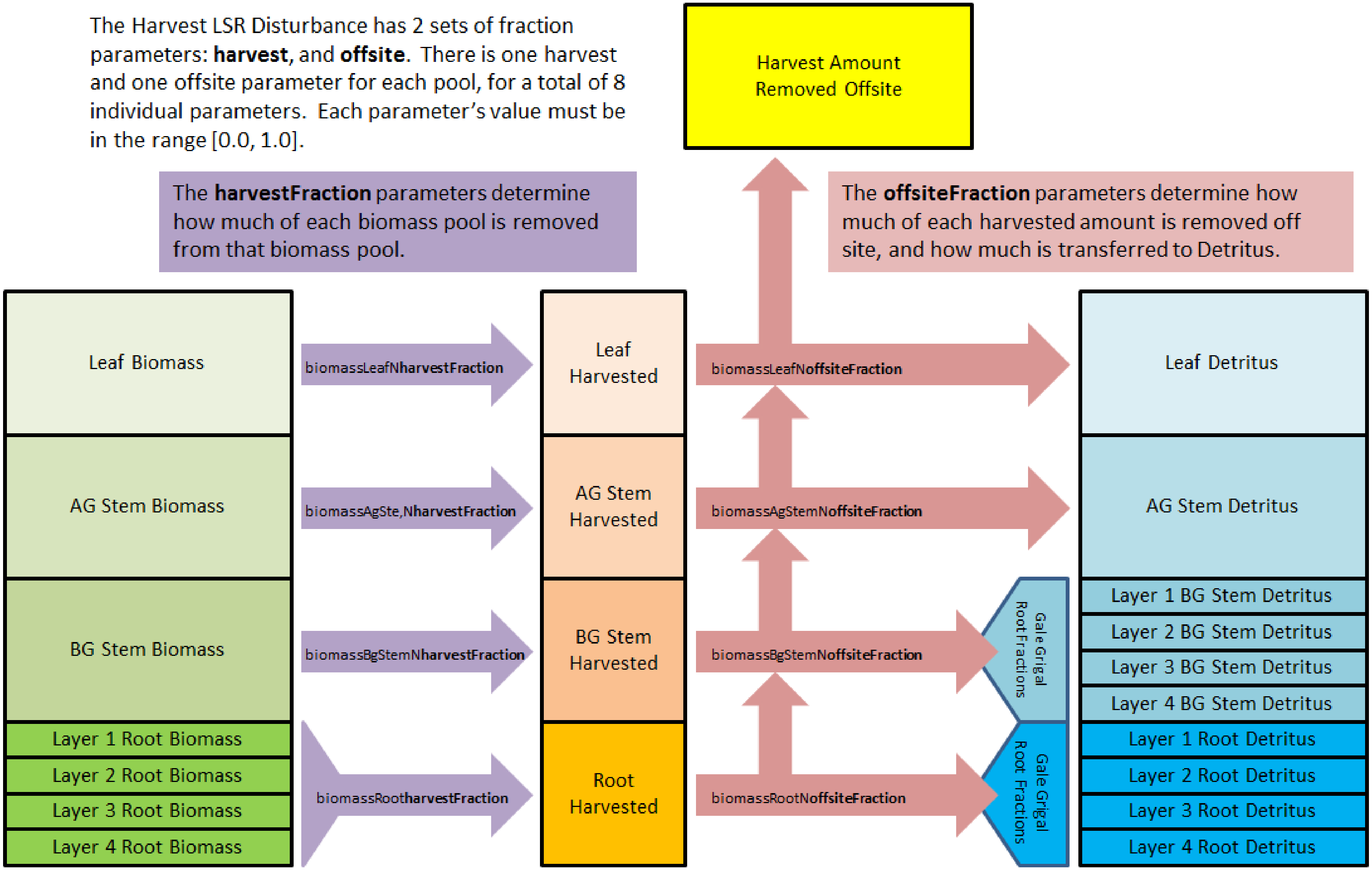
However, we first want to step through an example on the next four pages that illustrate how VELMA's new disturbance models work. We've chosen the Harvest disturbance model for this example because it illustrates key concepts shared by most of the other disturbance models, particularly with regard to how disturbances can be keyed to the LSR (Leaves-Stems-Root) plant biomass model. We recommend that you look over the details of this example before proceeding to sections 24.1 – 24.3.

LEAF, STEM and ROOT Biomass Pool Arrangement



Harvest LSR Disturbance Transfer Diagram

The Harvest LSR Disturbance has 2 sets of fraction parameters: **harvest**, and **offsite**. There is one harvest and one offsite parameter for each pool, for a total of 8 individual parameters. Each parameter's value must be in the range [0.0, 1.0].



Harvest Amount Removed Offsite

The **harvestFraction** parameters determine how much of each biomass pool is removed from that biomass pool.

The **offsiteFraction** parameters determine how much of each harvested amount is removed off site, and how much is transferred to Detritus.

Leaf Biomass

AG Stem Biomass

BG Stem Biomass

Layer 1 Root Biomass

Layer 2 Root Biomass

Layer 3 Root Biomass

Layer 4 Root Biomass

biomassLeafNharvestFraction

biomassAgSte,NharvestFraction

biomassBgStemNharvestFraction

biomassRootharvestFraction

Leaf Harvested

AG Stem Harvested

BG Stem Harvested

Root Harvested

biomassLeafNoffsiteFraction

biomassAgStemNoffsiteFraction

biomassBgStemNoffsiteFraction

biomassRootNoffsiteFraction

Gale Grigal Root Fractions

Gale Grigal Root Fractions

Leaf Detritus

AG Stem Detritus

Layer 1 BG Stem Detritus

Layer 2 BG Stem Detritus

Layer 3 BG Stem Detritus

Layer 4 BG Stem Detritus

Layer 1 Root Detritus

Layer 2 Root Detritus

Layer 3 Root Detritus

Layer 4 Root Detritus

When and Where a Harvest LSR Disturbance Occurs: An Example

Harvest LSR Disturbances occur for each loop, year and Julian day the user specifies
in the `initializeActiveLoops`, `initializeActiveYears` and `initializeActiveJdays` parameters' values.
The user may specify one or more loops, years and Julian days, or ranges of the same.
At least one loop, year and Julian day must be specified.

Here is an example specification for the temporal scheduling parameters:

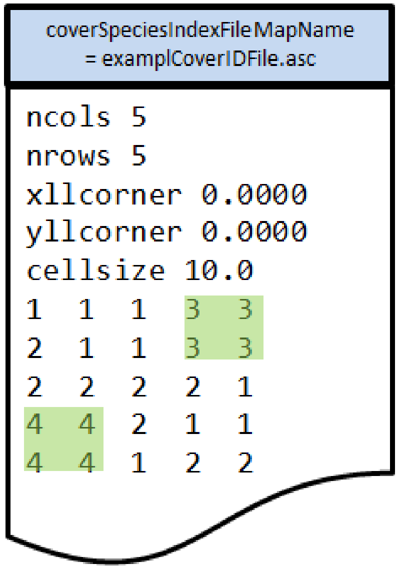
```
initializeActiveLoops = 1, 3                  ( harvest will occur in loops 1 and 3 )
initializeActiveYears =1969-1980, 1985        ( harvest occurs in years 1969 through 1980, and in 1985 )
initializeActiveJdays = 200                   ( harvest occurs on the 200th Julian day )
```

The disturbance only occurs for the cover types the user specifies
in the `initializeCoverIds` parameter's values; multiple cover Id numbers may be specified, but not ranges of numbers.
The disturbance uses the specified cover Ids, plus the data in the cover id map specified by the `coverSpeciesIndexFileMapName` parameter to determine which cells should be harvested.

Here is an example specification for the spatial/cover parameters:

```
initializeCoverIds = 3, 4
coverSpeciesIndexFileMapName = exampleCoverIDFile.asc
```

With the above parameterization, and the exampleCoverIDFile.asc contents at right, harvest will occur in the cells highlighted in green.



coverSpeciesIndexFileMapName
= examplCoverIDFile.asc

```
ncols 5
nrows 5
xllcorner 0.0000
yllcorner 0.0000
cellsize 10.0
1  1  1  3  3
2  1  1  3  3
2  2  2  2  1
4  4  2  1  1
4  4  1  2  2
```

The disturbance occurs only for the specified cover types at or above a specified age.
The user specifies this "lowest age limit" in the `ageThreshold` parameter value. The value must be an positive integer, or zero – zero specifies any age is acceptable.

Here is an example specification for the age limit parameter:

```
ageThreshold = 135              ( harvest will occur only for cells whose cover is 135 or more years old. )
```

Filtering: Additional Control For Where a Harvest LSR Disturbance Occurs

Harvest LSR Disturbances may (optionally) be limited to specific areas by specifying a Filter Map
In order to use filtering, the user must have define a Filter Map: a Grid Ascii map with ID numbers for cell data values.
The user specifies this map's name as the `filterMapFullName` parameter value.
(Leaving the `filterMapFullName` unspecified indicates "No Filtering" to the Harvest LSR Disturbance.)

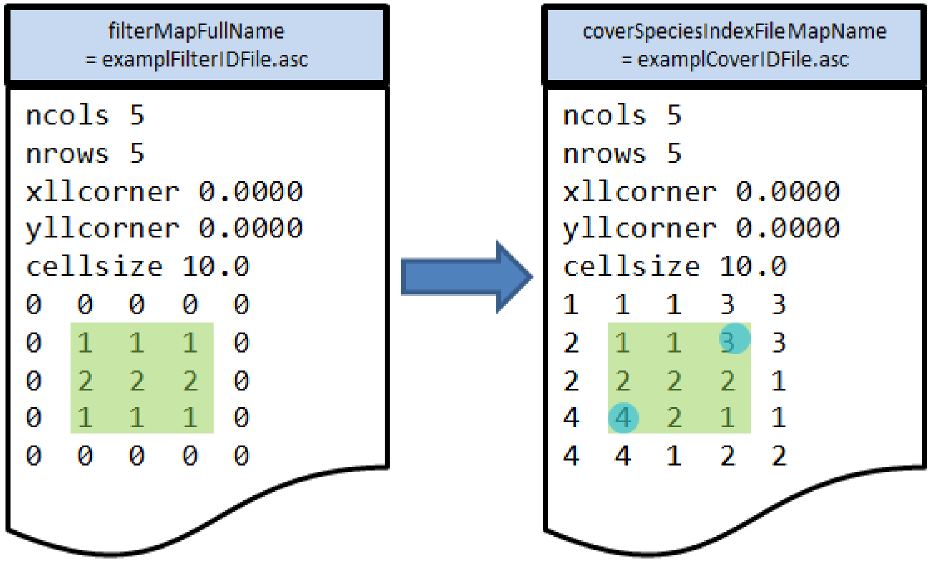
When a filter file is specified the disturbance only occurs for the filter Ids cells the user specifies
in the `initializeFilterIds` parameter's values; multiple filter Id values may be specified, but not number ranges.
The disturbance uses the specified filter Ids, plus the data in the filter map specified by the `filterMapFullName` parameter to determine which cells are eligible for harvested.

Here is an example specification for the spatial/cover parameters:

```
initializeFilterIds = 1, 2
filterMapFullName = exampleFilterIDFile.asc
```

Note: these example filter Id values, and the values in the exampleFilterIDFile have ***no*** *relation to the cover Ids.*
Using this filtering scheme **and the cover ID example from the previous slide**, the selection logic is
(1) Find the subset of cells in the filter map that match the filter ids specified by initializeFilterIds.
(2) Consider other spatial-selection (e.g. cover Ids) based *only on that subset of cells.*

The Map files to the right show the example:
(1) Cells eligible for consideration based on the filter map and specified filter Ids are highlighted in green.
(2) That subset of cells is then examined via the cover Id map and specified cover Ids.
Cells eligible for harvest based on filtering, followed by cover id selection are further highlighted in blue.
Note: Other criteria (i.e. age) might also apply after filtering to the filtered subset of cells.
Note: *Remember!* Id numbers in the filter map are *not* related to Id numbers in the cover map. (i.e. filter id 2 is *not* equal to cover id 2)



filterMapFullName
= examplFilterIDFile.asc

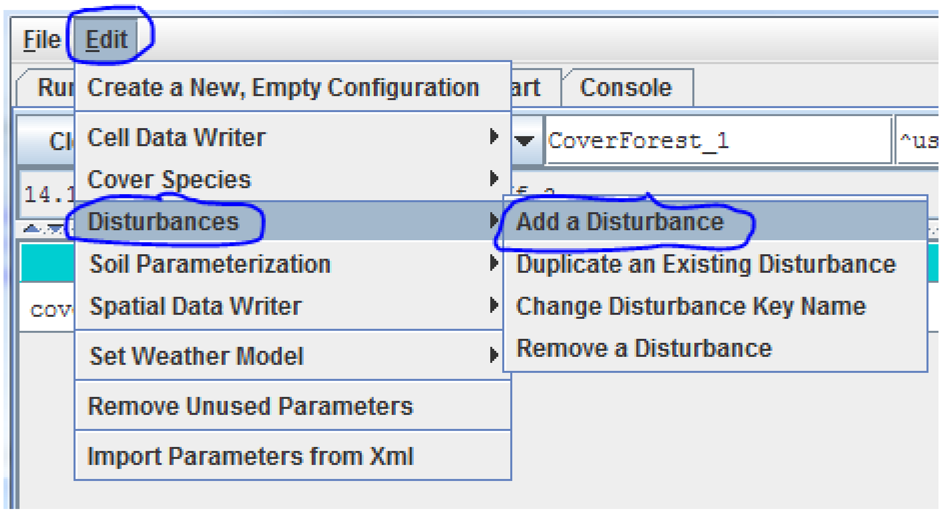
```
ncols 5
nrows 5
xllcorner 0.0000
yllcorner 0.0000
cellsize 10.0
0   0   0   0   0
0   1   1   1   0
0   2   2   2   0
0   1   1   1   0
0   0   0   0   0
```

coverSpeciesIndexFileMapName
= examplCoverIDFile.asc

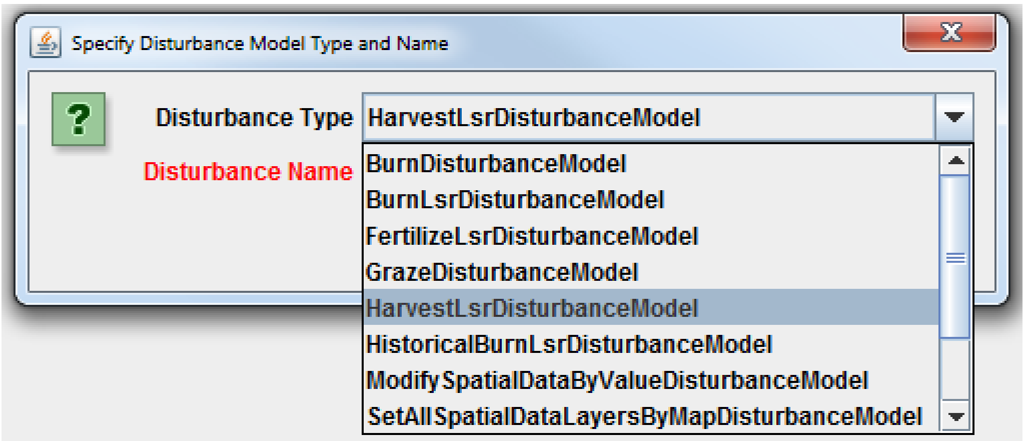
```
ncols 5
nrows 5
xllcorner 0.0000
yllcorner 0.0000
cellsize 10.0
1   1   1   3   3
2   1   1   3   3
2   2   2   2   1
4   4   2   1   1
4   4   1   2   2
```

24.1 – Disturbance Items by Model Type

To begin, you need to select the disturbance model you want by clicking Edit → Disturbances → Add a Disturbance:



You will see the following drop-down menu for selecting a Disturbance Model. To further demonstrate the harvest disturbance example illustrated under section 24.0, we select the HarvestLsrDisturbanceModel:

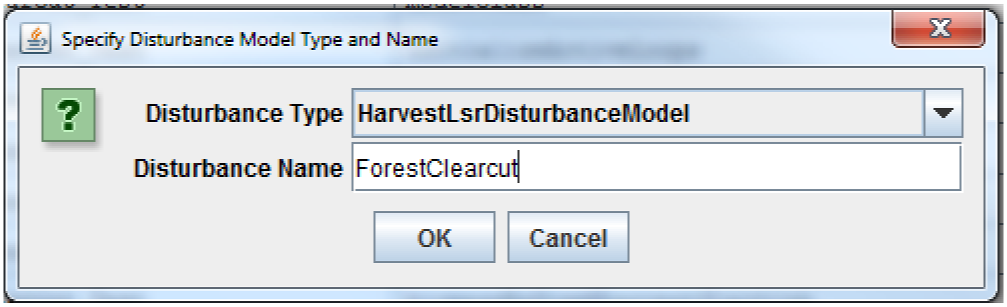

Specify Disturbance Model Type and Name

Disturbance Type: HarvestLsrDisturbanceModel

Disturbance Name

BurnDisturbanceModel
BurnLsrDisturbanceModel
FertilizeLsrDisturbanceModel
GrazeDisturbanceModel
HarvestLsrDisturbanceModel
HistoricalBurnLsrDisturbanceModel
ModifySpatialDataByValueDisturbanceModel
SetAllSpatialDataLayersByMapDisturbanceModel

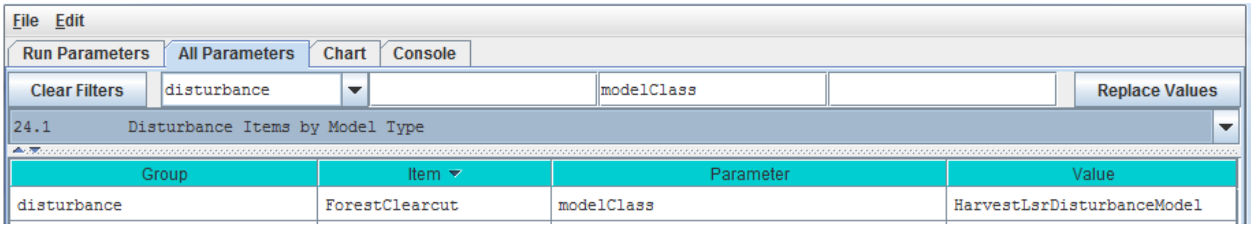
Now choose a name for your `HarvestLsrDisturbanceModel` and type it the "Disturbance Name" dialog box, for example, "ForestClearcut":


Specify Disturbance Model Type and Name

Disturbance Type: HarvestLsrDisturbanceModel

Disturbance Name: ForestClearcut

After selecting the `HarvestLsrDisturbanceModel` and giving it a disturbance name, go to the All Parameters drop-down menu and select "24.1 Disturbance Items by Model Type". You should see the following:

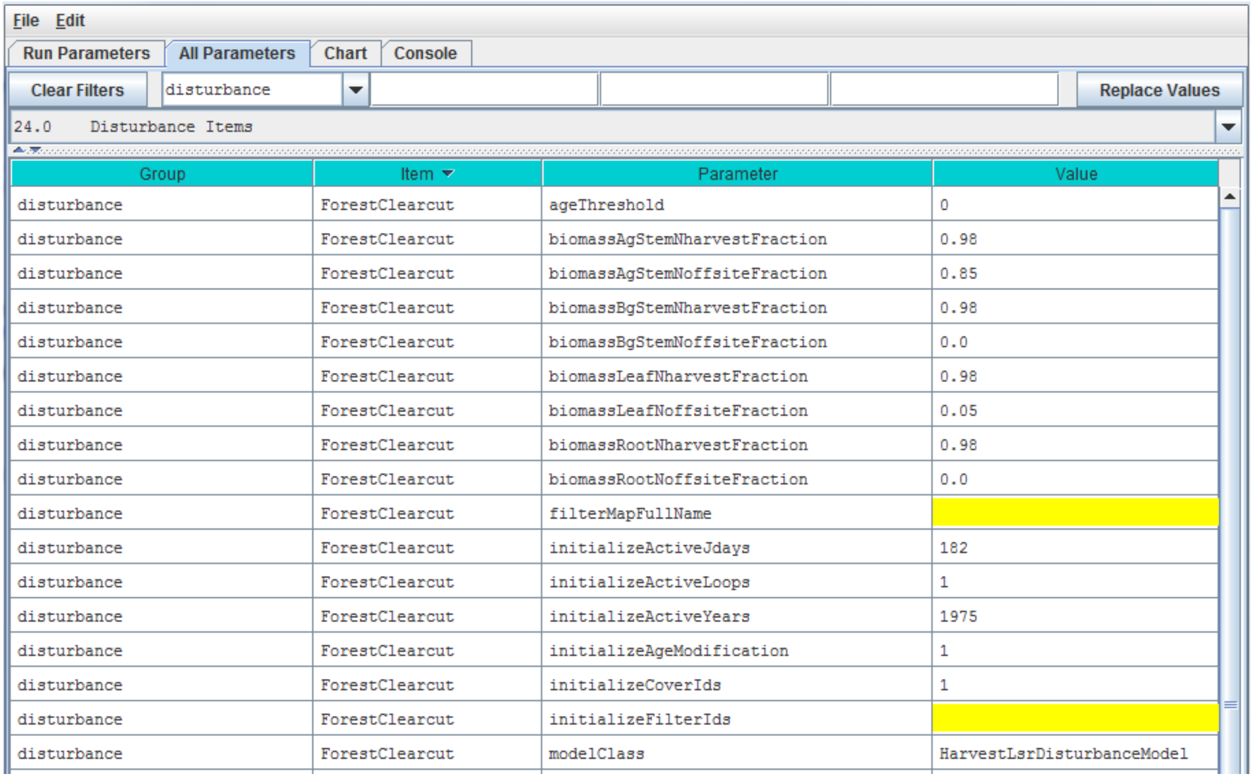


disturbance | modelClass

24.1 Disturbance Items by Model Type

Group	Item	Parameter	Value
disturbance	ForestClearcut	modelClass	HarvestLsrDisturbanceModel

Select "24.0 – Disturbance Items" from the All Parameters drop-down menu to see a list of all parameters associated with the `HarvestLsrDisturbanceModel`. In the screenshot below, we have specified parameter values for the `HarvestLsrDisturbanceModel` disturbance we named "ForestClearcut":



disturbance

24.0 Disturbance Items

Group	Item	Parameter	Value
disturbance	ForestClearcut	ageThreshold	0
disturbance	ForestClearcut	biomassAgStemNharvestFraction	0.98
disturbance	ForestClearcut	biomassAgStemNoffsiteFraction	0.85
disturbance	ForestClearcut	biomassBgStemNharvestFraction	0.98
disturbance	ForestClearcut	biomassBgStemNoffsiteFraction	0.0
disturbance	ForestClearcut	biomassLeafNharvestFraction	0.98
disturbance	ForestClearcut	biomassLeafNoffsiteFraction	0.05
disturbance	ForestClearcut	biomassRootNharvestFraction	0.98
disturbance	ForestClearcut	biomassRootNoffsiteFraction	0.0
disturbance	ForestClearcut	filterMapFullName	
disturbance	ForestClearcut	initializeActiveJdays	182
disturbance	ForestClearcut	initializeActiveLoops	1
disturbance	ForestClearcut	initializeActiveYears	1975
disturbance	ForestClearcut	initializeAgeModification	1
disturbance	ForestClearcut	initializeCoverIds	1
disturbance	ForestClearcut	initializeFilterIds	
disturbance	ForestClearcut	modelClass	HarvestLsrDisturbanceModel

<u>Parameter Definitions</u>

`ageThreshold` — Applicable cover must be greater or equal to the specified ageThreshold (in integer years) to be harvested. Defaults to 0 -- i.e., any otherwise-applicable cell is harvested regardless of age.

`biomassAgStemNharvestFraction` — The fraction [0.0 1.0] of available above-ground stem biomass harvested i.e. converted from biomass above-ground stem to above-ground stem detritus and/or removed from the simulation.

`biomassAgStemNoffsiteFraction`

The fraction [0.0 1.0] of harvested above-ground stem biomass removed offsite, i.e., removed from the cell and the simulation.

`biomassBgStemNharvestFraction`

The fraction [0.0 1.0] of available below-ground biomass harvested i.e. converted from biomass below-ground to below-ground detritus and/or removed from the simulation.

`biomassBgStemNoffsiteFraction`

The fraction [0.0 1.0] of harvested below-ground stem biomass removed offsite. i.e. removed from the cell and the simulation.

`biomassLeafNharvestFraction`

The fraction [0.0 1.0] of available leaf biomass harvested i.e. converted from biomass leaf to leaf detritus and/or removed from the simulation.

`biomassLeafNoffsiteFraction`

The fraction [0.0 1.0] of harvested leaf biomass removed offsite. i.e. removed from the cell and the simulation.

`biomassRootNharvestFraction`

The fraction [0.0 1.0] of available root biomass harvested i.e. converted from biomass root to root detritus and/or removed from the simulation.

`biomassRootNoffsiteFraction`

The fraction [0.0 1.0] of harvested root biomass removed offsite. i.e. removed from the cell and the simulation.

`filterMapFullName`

The fully-qualified (path/name.ext) name of a Grid ASCII (.asc) file containing (integer) filter Ids. When left unspecified (default) indicates that filtering will be turned off. Note: if the value for this parameter is not a fully-qualified name the simulator engine will prefix the specified input data location root and name to this parameter's value to create a default fully-qualified name.

`initializeActiveJdays`

The comma-separated list of Julian days on which the disturbance occurs. Entries in the list may be either single Julian days or start-end (dash-separated) ranges. A valid Julian day is any positive whole number between 1 and 366 (however day 366 is ignored in non-leap years). For example this text triggers output on days 1 365 and between 30 and 90 inclusive: 1 30-90 365

`initializeActiveLoops`

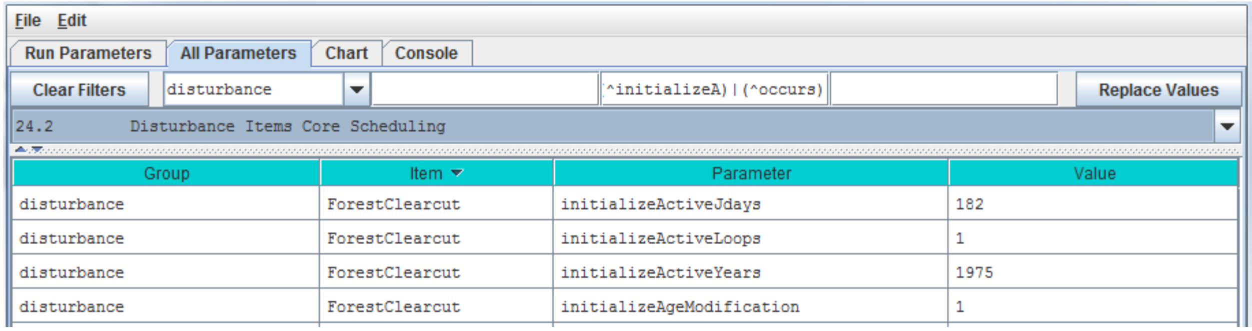
The comma-separated list of loop index numbers (range [1 nth-Loop]) in which the disturbance occurs. Entries in the list may be either single loop numbers or start-end (dash-separated) ranges. A valid loop index number is any positive whole number between 1 and the number of loops specified for the simulation run. For example in a simulation run for 10 loops this text triggers output in loops 1 3 through 5 and 10 inclusive: 1 3-5 10

`initializeActiveYears`

The comma-separated list of Years in which the disturbance occurs. Entries in the list may be either

	single years or start-end (dash-separated) ranges. A valid year is any positive whole number (but the user is expected to choose years within the simulation run!). For example in a simulation run between 1995 and 2010, this text triggers output in 2000 through 2005 inclusive: 2000-2005
`initializeAgeModification`	(Optional) Specifies the number of years to subtract or add to the age of the cover in affected cells when the disturbance occurs. To add n years to a cell's cover age prefix a plus (+) sign to the number n. To subtract n years to a cell's cover age prefix a minus (-) sign to the number n. To set the cell's cover age to a specific number of years specify a number n without any sign prefix. Subtraction halts at zero (e.g. a cover age of 7 given -10 as the modifier will result in 0 not -3 as the modified age).
`initializeCoverIds`	The Cover Species Id numbers that are affected by this disturbance model. Specify a comma-separated sequence of Cover Species uniqueId numbers.
`modelClass`	The disturbance model's actual Java class name

Select "24.2 – Disturbance Items Core Scheduling" to focus the All Parameters table on the parameters for specifying when disturbance events occur:



disturbance

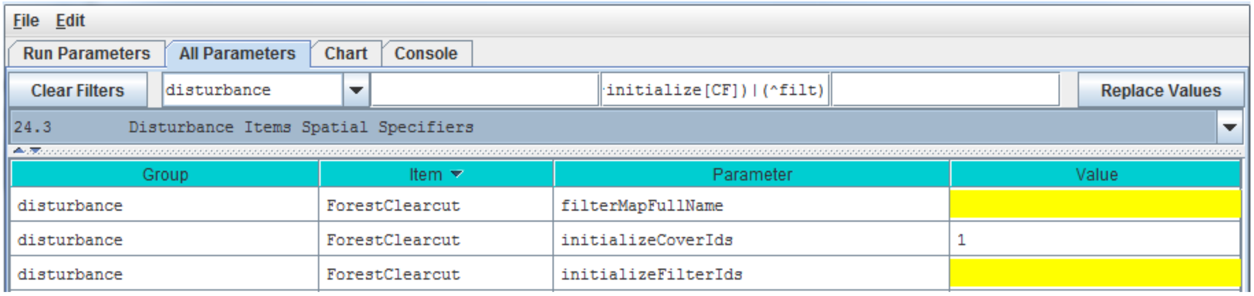
^initializeA)|(^occurs)

24.2 Disturbance Items Core Scheduling

Group	Item	Parameter	Value
disturbance	ForestClearcut	initializeActiveJdays	182
disturbance	ForestClearcut	initializeActiveLoops	1
disturbance	ForestClearcut	initializeActiveYears	1975
disturbance	ForestClearcut	initializeAgeModification	1

Parameter Definitions: these parameters are defined under the All Parameters screenshot for Select "24.0 – Disturbance Items", above.

Select "24.3 -- Disturbance Items Spatial Specifiers" to focus the All Parameters table on the parameters for specifying where disturbance events occur:



disturbance

initialize[CF])|(^filt)

24.3 Disturbance Items Spatial Specifiers

Group	Item	Parameter	Value
disturbance	ForestClearcut	filterMapFullName	
disturbance	ForestClearcut	initializeCoverIds	1
disturbance	ForestClearcut	initializeFilterIds	

Parameter Definitions: these parameters are defined under the All Parameters screenshot for Select "24.0 – Disturbance Items", above.

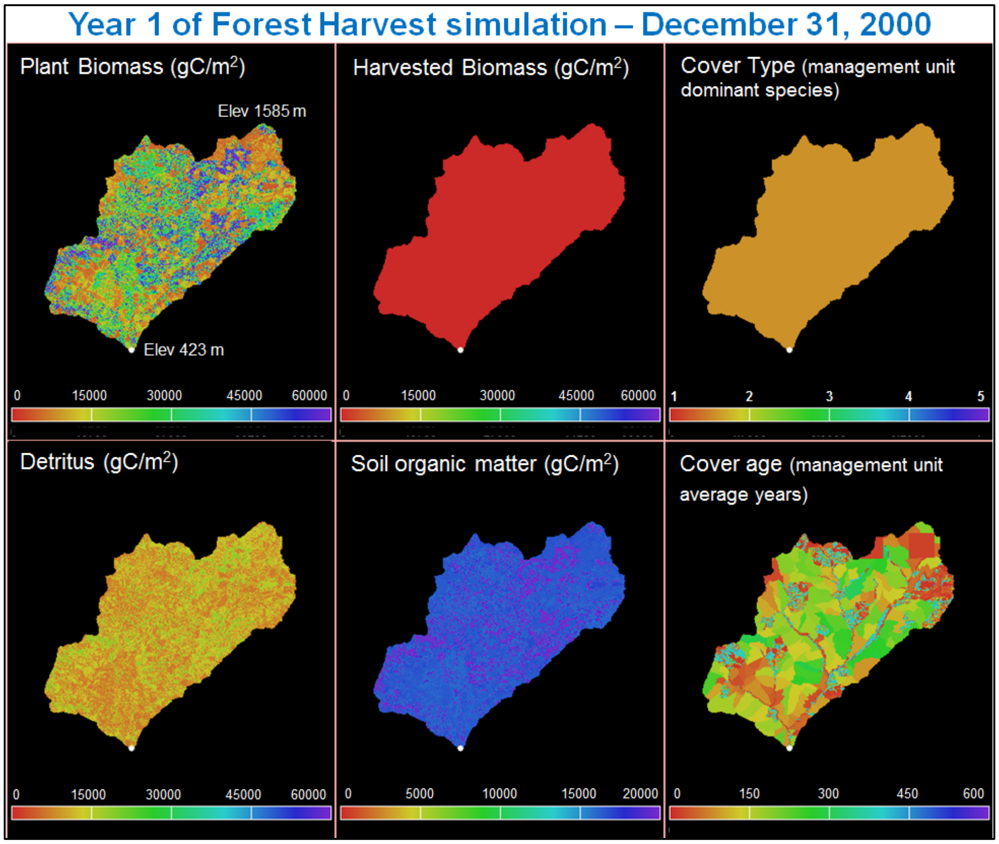
Calibration Notes

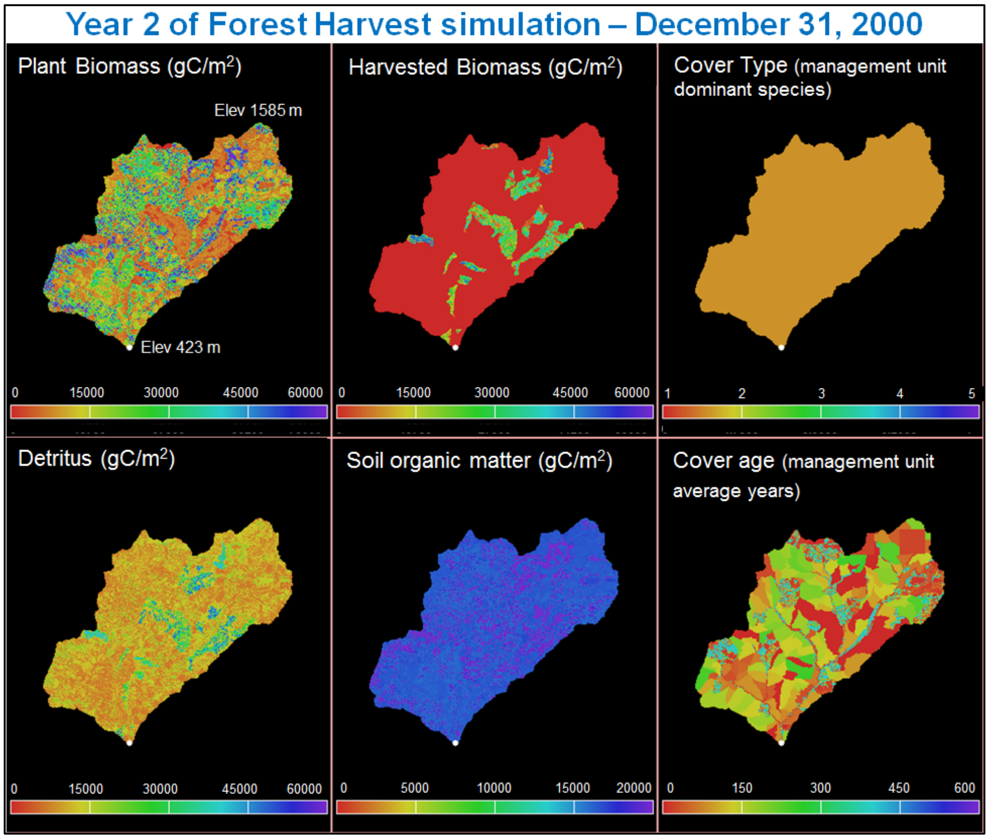
Make sure that the disturbance models you have set up are working correctly. Are disturbance events occurring when and where you want them to, are the correct amounts of biomass being harvested/burned/grazed, and are the correct amounts and forms of fertilizer being applied. For a first check, click VELMA's GUI "Chart" tab at any time during a simulation, then click the drop-down menu button located near the lower left corner of the screen, then select an appropriate spatial data display option (see section "25.0 – Runtime Chart Display Scale" for details").

For example, by selecting the "Harvest Spatial Data" menu item you can see daily and year-to-date summaries of harvest events and their effects on various biomass tools. The two figures below show year-end summaries of harvest events for consecutive years, 2000 and 2001. In 2000 there were no harvests (top figure: top row, middle frame). In 2001 there were several fairly large harvest events (bottom figure: top row, middle frame). A comparison of the other frames in top and bottom figures reveals changes in plant biomass, detritus, soil organic matter, and cover age following harvest events.

An examination of VELMA's daily and annual output files (csv format) provides a more quantitative check of disturbance events.

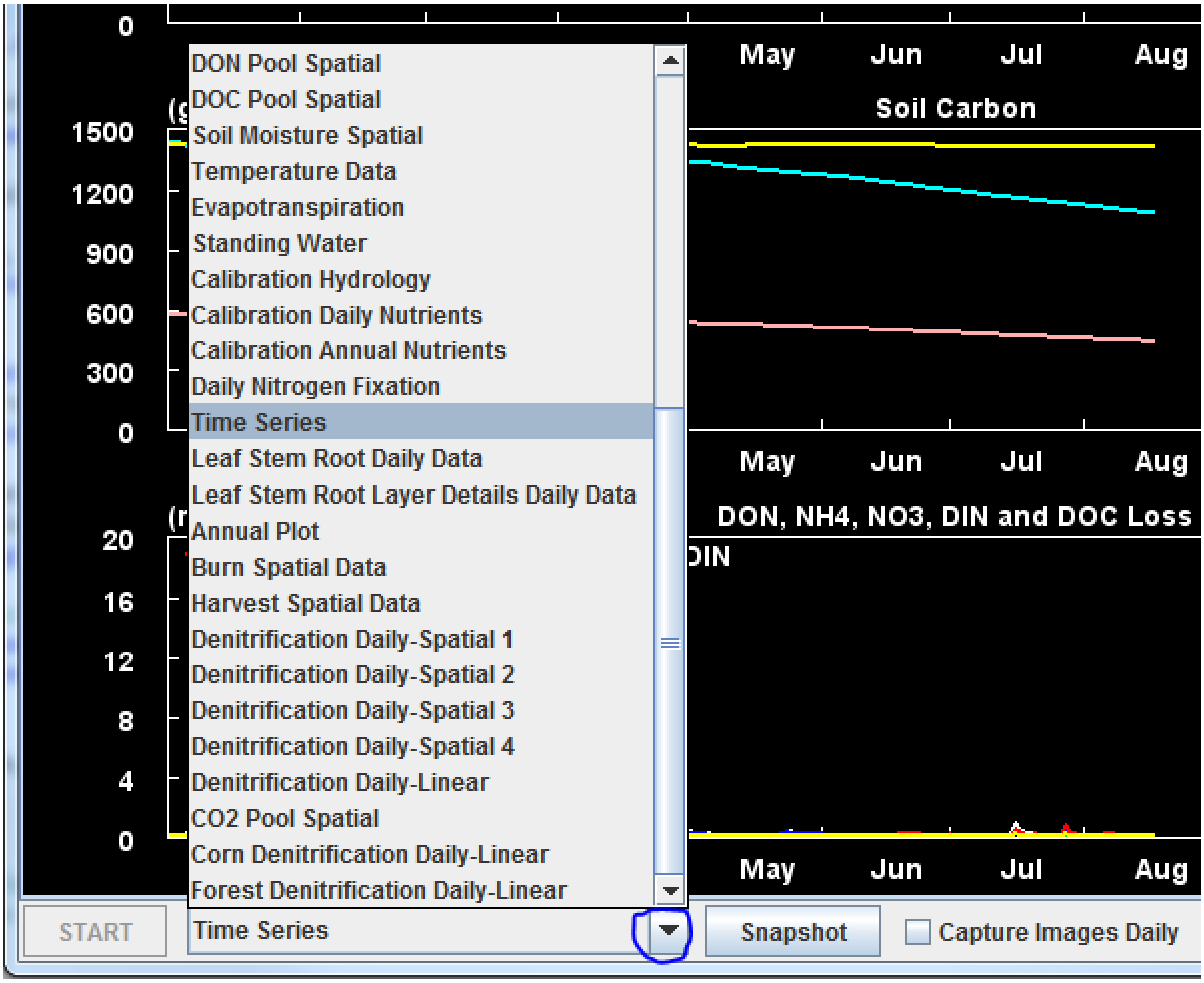
Screenshot of runtime “Harvest Spatial Data” display.


Year 1 of Forest Harvest simulation – December 31, 2000
Plant Biomass (gC/m2)
Elev 1585 m
Elev 423 m
0 15000 30000 45000 60000
Harvested Biomass (gC/m2)
0 15000 30000 45000 60000
Cover Type (management unit dominant species)
1 2 3 4 5
Detritus (gC/m2)
0 15000 30000 45000 60000
Soil organic matter (gC/m2)
0 5000 10000 15000 20000
Cover age (management unit average years)
0 150 300 450 600

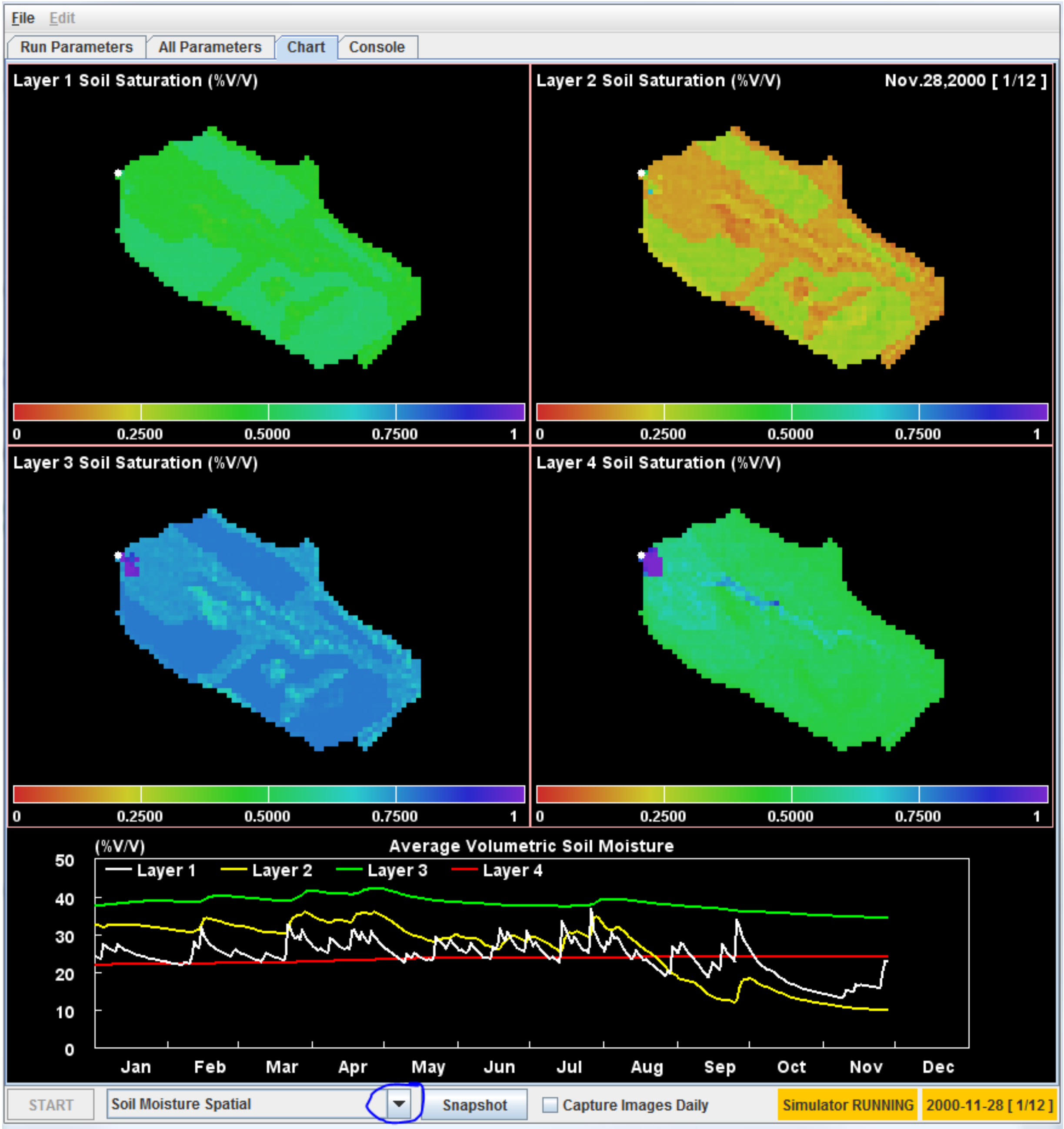

Year 2 of Forest Harvest simulation – December 31, 2000
Plant Biomass (gC/m2)
Elev 1585 m
Elev 423 m
0 15000 30000 45000 60000
Harvested Biomass (gC/m2)
0 15000 30000 45000 60000
Cover Type (management unit dominant species)
1 2 3 4 5
Detritus (gC/m2)
0 15000 30000 45000 60000
Soil organic matter (gC/m2)
0 5000 10000 15000 20000
Cover age (management unit average years)
0 150 300 450 600

25.0 – Runtime Chart Display Scale (link to All Parameters TOC)

VELMA has powerful visualization capabilities built into it, hence the model's name – Visualizing Ecosystem Land Management Assessments. To see a list of visualization options, select the VELMA GUI "Chart" tab at any time during a simulation, then click the drop-down menu button (▼) located near the lower left corner of the screen, and select a display option. Each option dynamically displays daily changes in one or more output variables. The screenshot below shows the lower half of possible visualization options:

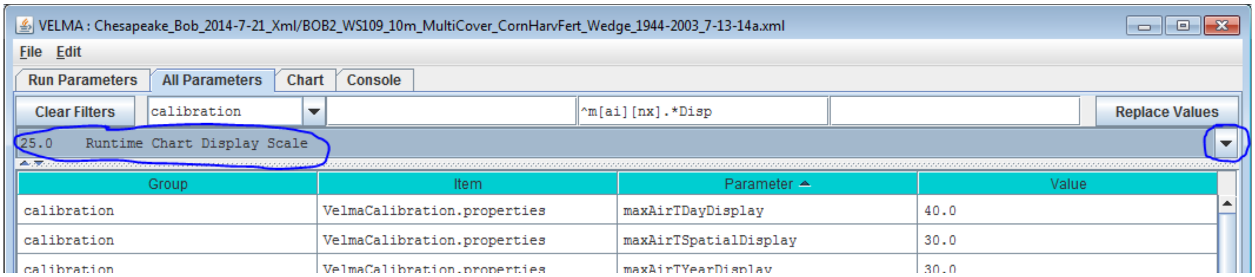

0
May
Jun
Jul
Aug
Soil Carbon
(g
1500
1200
900
600
300
0
May
Jun
Jul
Aug
DON, NH4, NO3, DIN and DOC Loss
DIN
(r
20
16
12
8
4
0
May
Jun
Jul
Aug
DON Pool Spatial
DOC Pool Spatial
Soil Moisture Spatial
Temperature Data
Evapotranspiration
Standing Water
Calibration Hydrology
Calibration Daily Nutrients
Calibration Annual Nutrients
Daily Nitrogen Fixation
Time Series
Leaf Stem Root Daily Data
Leaf Stem Root Layer Details Daily Data
Annual Plot
Burn Spatial Data
Harvest Spatial Data
Denitrification Daily-Spatial 1
Denitrification Daily-Spatial 2
Denitrification Daily-Spatial 3
Denitrification Daily-Spatial 4
Denitrification Daily-Linear
CO2 Pool Spatial
Corn Denitrification Daily-Linear
Forest Denitrification Daily-Linear
START
Time Series
Snapshot
□ Capture Images Daily

For example, the "Soil Moisture Spatial" visualization displays runtime daily changes in spatial patterns of soil moisture (% saturation) for four soil layers for a 20-ha watershed on the day currently being simulated. A screenshot for a particular day is shown below. The lower third of display is a running graphical summary of daily changes in soil moisture per soil layer from January 1 up to the date currently simulated. A new view is generated at the beginning of each calendar year). A variety of hardwired display formats are used for different output variables.



You can save daily screenshots as .png files by clicking the “Snapshot” button (located near the middle of the menu bar at the bottom of the screenshot, above). You can also save a series of daily images by clicking the “Capture Images Daily” checkbox. After a simulation is completed, third-party software can be used to stitch together a time series of saved .png screenshots to make a time-lapse movie. However, note that saving daily .png images for long simulations can fill large amounts of disk space.

You can use the VELMA GUI’s All Parameters tab to review and change the settings for display parameters controlling minimum (min) and maximum (max) values for the y-axis and range color bar. To locate all display parameters, select the “25.0 Runtime Chart Display Scale” item from the All Parameters drop-down menu button:



VELMA : Chesapeake_Bob_2014-7-21_Xml/BOB2_WS109_10m_MultiCover_CornHarvFert_Wedge_1944-2003_7-13-14a.xml

calibration | | ^m[ai][nx].*Disp |

25.0 Runtime Chart Display Scale

Group	Item	Parameter	Value
calibration	VelmaCalibration.properties	maxAirTDayDisplay	40.0
calibration	VelmaCalibration.properties	maxAirTSpatialDisplay	30.0
calibration	VelmaCalibration.properties	maxAirTYearDisplay	30.0

Different runtime display charts use different subsets of the display parameters. Unfortunately, nothing listed under “All Parameters 25.0 Runtime Chart Display Scale” indicates which display parameters belong to a particular runtime display chart. Therefore, we have constructed the table below so that users can associate runtime display chart names with the display parameters used by that chart.

For example, if you want to adjust VELMA’s “Annual Plot” display for soil humus, you would find “Annual Plot” under the “Display Selector Name” column in the table below, copy “`maxHumusDayDisplay`” and paste it in the “Parameter” search window in the VELMA GUI’s All Parameters view. Hit your keyboard’s “Enter” key and you should see “`maxHumusDayDisplay`” under “Parameter column. Specify the value you want. Repeat for the process for the “`minHumusDayDisplay`” parameter. [Note: you cannot change the display parameterizations during a simulation run. Currently, the VELMA GUI does not allow runtime display scales to change dynamically during simulation runs. Prior to starting a simulation, you must set the display parameters for any chart you will use during a simulation.]

Display Selector Name	MIN Display Setting	MAX Display Setting
<cover> + Denitrification Daily-Linear	`( always zero )`	`maxDenitrificationDaySpatialDisplay`
	`( always zero )`	`maxNo3DenitDaySpatialDisplay`
	`( always zero )`	`maxCo2DenitDaySpatialDisplay`
Annual Plot	`( always zero )`	`maxDonLossYearFinalDisplay`
	`( always zero )`	`maxDocLossYearFinalDisplay`
	`( always zero )`	`maxNppYearSumDisplay`
	`minBiomassYearFinalDisplay`	`maxBiomassYearFinalDisplay`
	`minHumusDayDisplay`	`maxHumusDayDisplay`
	`minLitterDayDisplay`	`maxLitterDayDisplay`
	`minNepYearSumDisplay`	`maxNepYearSumDisplay`
Burn Spatial Data	`minBiomassAgStemDayDisplay`	`maxBiomassAgStemDayDisplay`
	`minBiomassLeafDayDisplay`	`maxBiomassLeafDayDisplay`
	`minDetritusAgStemDayDisplay`	`maxDetritusAgStemDayDisplay`
	`minDetritusLeafDayDisplay`	`maxDetritusLeafDayDisplay`
	`minHumusDayDisplay`	`maxHumusDayDisplay`
	`minLitterDayDisplay`	`maxLitterDayDisplay`

Calibration Annual Nutrients	(always zero)	maxNh4LossYearSumDisplay
	(always zero)	maxNo3LossYearSumDisplay
	(always zero)	maxDonLossYearSumDisplay
	minDocLossYearSumDisplay	maxDocLossYearSumDisplay
Calibration Daily Nutrients	(always zero)	maxNh4LossDayDisplay
	(always zero)	maxNo3LossDayDisplay
	(always zero)	maxDonLossDayDisplay
	(always zero)	maxDocLossDayDisplay
Calibration Hydrology	(always zero)	maxRainDayDisplay
	(always zero)	maxRunoffDayDisplay
	(always zero)	maxSubRunoffDayDisplay
	(always zero)	maxEtDayDisplay
	minHydroYearSumDisplay	maxHydroYearSumDisplay
CO2 Pool Spatial	(always zero)	maxCo2DaySpatialDisplay
Daily Nitrogen Fixation	(always zero)	maxNinDayDisplay
	(always zero)	maxPlantUptakeSpatialDisplay
	(always zero)	maxSymbioticNfixedDayDisplay
	(always zero)	maxAsymbioticNfixedDayDisplay
	(always zero)	maxDonLossDayDisplay
	(always zero)	maxTotalNLossDayDisplay
	(always zero)	maxNitrificationDaySpatialDisplay
	(always zero)	maxDenitrificationDaySpatialDisplay
Default Layer 1 - 4	(always zero)	maxDonLossDayDisplay
	(always zero)	maxDocLossDayDisplay
	(always zero)	maxRainDayDisplay
	(always zero)	maxRunoffDayDisplay
Denitrification Daily-Spatial	(always zero)	maxDenitrificationDaySpatialDisplay
Denitrification Daily-Spatial 1 - 4	(always zero)	maxDenitrificationDaySpatialDisplay
	(always zero)	maxNo3DenitDaySpatialDisplay
	(always zero)	maxCo2DaySpatialDisplay
DOC Pool Spatial	(always zero)	maxDocSpatialDisplay
DON Pool Spatial	(always zero)	maxDonSpatialDisplay
Evapotranspiration	(always zero)	maxEtDayDisplay
Harvest Spatial Data	minBiomassDayDisplay	maxBiomassDayDisplay
	minHumusDayDisplay	maxHumusDayDisplay

	minLitterDayDisplay	maxLitterDayDisplay
Leaf Stem Root Daily Data	minBiomassAgStemDayDisplay	maxBiomassAgStemDayDisplay
	minBiomassBgStemDayDisplay	maxBiomassBgStemDayDisplay
	minBiomassLeafDayDisplay	maxBiomassLeafDayDisplay
	minBiomassRootDayDisplay	maxBiomassRootDayDisplay
	minDetritusAgStemDayDisplay	maxDetritusAgStemDayDisplay
	minDetritusBgStemDayDisplay	maxDetritusBgStemDayDisplay
	minDetritusLeafDayDisplay	maxDetritusLeafDayDisplay
	minDetritusRootDayDisplay	maxDetritusRootDayDisplay
Leaf Stem Root Layer Details Daily Data	minBiomassRootLayersDayDisplay	maxBiomassRootLayersDayDisplay
	minDetritusBgStemLayersDayDisplay	maxDetritusBgStemLayersDayDisplay
	minDetritusRootLayersDayDisplay	maxDetritusRootLayersDayDisplay
	minHumusLayersDayDisplay	maxHumusLayersDayDisplay
LSR Summary Spatial	(always zero)	maxNppDayDisplay
	minBiomassDayDisplay	maxBiomassDayDisplay
	minHumusDayDisplay	maxHumusDayDisplay
	minLitterDayDisplay	maxLitterDayDisplay
NH4 Pool Spatial	(always zero)	maxNh4SpatialDisplay
Nitrification Daily-Spatial	(always zero)	maxNitrificationDaySpatialDisplay
Nitrogen Fixation Daily-Spatial	(always zero)	maxSymbioticNfixedDayDisplay
	(always zero)	maxAsymbioticNfixedDayDisplay
NO3 Pool Spatial	(always zero)	maxNo3SpatialDisplay
Plant Uptake Spatial	minPlantUptakeSpatialDisplay	maxPlantUptakeSpatialDisplay
Productivity Spatial	(always zero)	maxNppDayDisplay
	(always zero)	maxNh4SpatialDisplay
	(always zero)	maxNo3SpatialDisplay
	minBiomassDayDisplay	maxBiomassDayDisplay
Snow Dynamics Spatial	(always zero)	maxRainSpatialDisplay
	(always zero)	maxSnowSpatialDisplay
	(always zero)	maxSnowDepthSpatialDisplay
	(always zero)	maxSnowMeltSpatialDisplay
	minAirTSpatialDisplay	maxAirTSpatialDisplay
Standing Water	(always zero)	maxRainDayDisplay
	(always zero)	maxRunoffDayDisplay
	(always zero)	maxStandingWaterDayDisplay
Temperature Data	minAirTDayDisplay	maxAirTDayDisplay

Time Series	(always zero)	maxDonLossDayDisplay
	(always zero)	maxDocLossDayDisplay
	(always zero)	maxRainDayDisplay
	(always zero)	maxRunoffDayDisplay
	minAirTDayDisplay	maxAirTDayDisplay
	minBiomassDayDisplay	maxBiomassDayDisplay
	minHumusDayDisplay	maxHumusDayDisplay
	minLitterDayDisplay	maxLitterDayDisplay
VSA Spatial and Hydro Linear	(always zero)	maxRainDayDisplay
	(always zero)	maxRunoffDayDisplay
Weather Summary Spatial	(always zero)	maxRainDayDisplay
	(always zero)	maxRainSpatialDisplay
	(always zero)	maxSnowSpatialDisplay
	(always zero)	maxSnowMeltSpatialDisplay
	minAirTDayDisplay	maxAirTDayDisplay
	minAirTSpatialDisplay	maxAirTSpatialDisplay

APPENDICES

VELMA Version 2.0

Appendix 1:
Overview of VELMA's Leaf-Stem-Root (LSR) Plant Biomass Submodel

(Link to All Parameters Table of Contents)

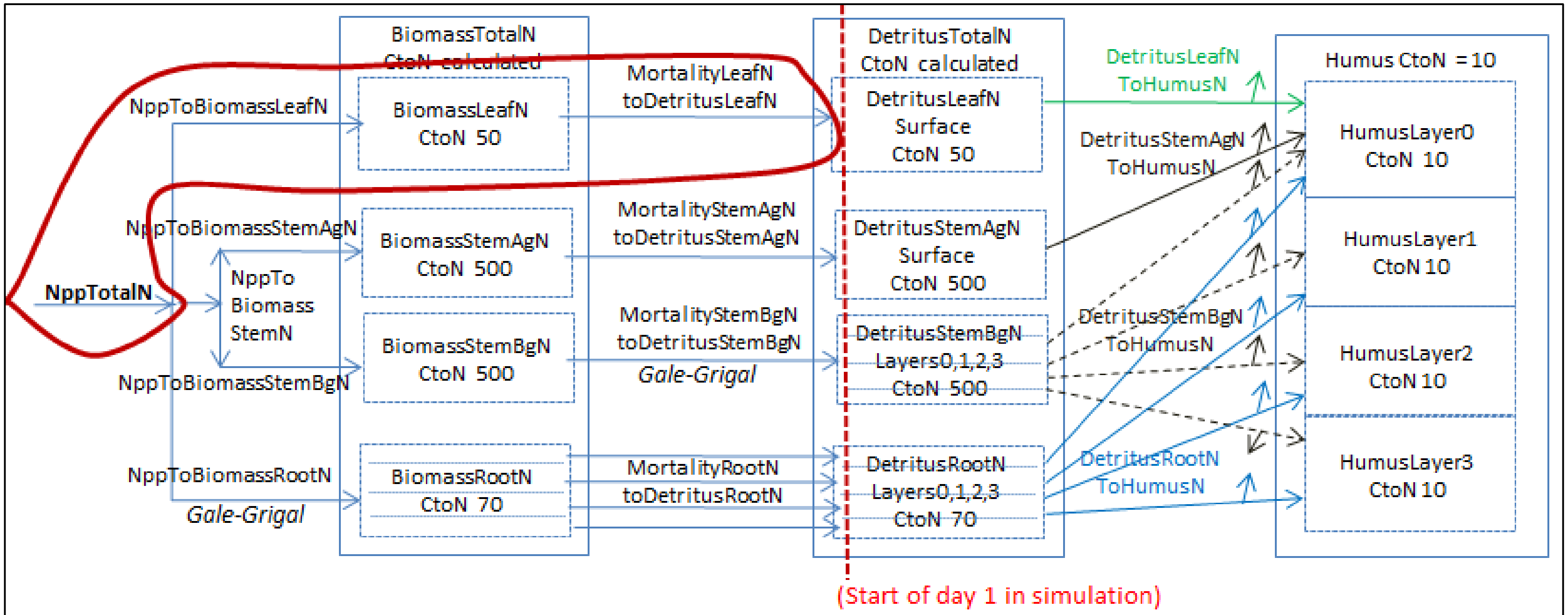
In VELMA version 1.0, plant biomass in each cell was simulated in aggregate. That is, leaves, stems and roots were lumped together in a single biomass pool (Abdelnour et al. 2011). This lumped approach greatly simplified the structure and application of the model. Although version 1.0 successfully simulates important aspects of catchment hydrology and biogeochemistry (e.g., Abdelnour et al. 2011, 2013), it does not simulate effects associated with phenological changes and disturbances and that disproportionately affect specific plant tissues, for example, leaf growth and senescence, defoliation events, selective removal of high or low C/N tissues via harvest, grazing, fire, etc.

To address such effects, version 2.0 simulates four plant biomass pools: leaves, aboveground stems, belowground stems, and fine roots. Main features of this leaf-stem-root (LSR) submodel include:

- Live plant biomass pools for four tissue types (also see Figure 1 and explanation preceding "All Parameters Section 5.0" of the User Manual)
  - Leaves (annual or perennial grasses, deciduous or coniferous trees, etc.)
  - Aboveground stems (trees, grasses, etc.)
  - belowground stems (<2 mm diameter, not involved in water and nutrient uptake)
  - fine roots (<2 mm diameter, involved in water and nutrient uptake)
- Dead (detritus) plant biomass pools for each of the four plant tissue types
- A user-specified C/N ratio for each plant tissue type. In version 2.0, a tissue's C/N ratio is applied to both the live and dead biomass pools. Simulation of separate live and dead C/N ratios will be addressed in future versions
- A new plant tissue mortality submodel simulates the loss of live biomass to detritus (see Appendix 1, Figures A1.1 – A1.4)
- A new decomposition model based on Potter et al. (1993) simulates the decomposition of detritus to CO_2 and humus (Appendix 1, Figures A1.5 – A1.8)
- A specified fraction of total plant nitrogen uptake is allocated to each plant tissue type, such that the fractions for all tissues = 1.0 (see Appendix 1, Figures A1.1 – A1.4)

Figures A1.1 – A1.11, below, graphically summarize the LSR submodel structure and model equations. These figures portray the pools (boxes) and fluxes (arrows) of nitrogen and/or carbon from live to dead biomass and from dead biomass to the humus pool and atmosphere. Pools and fluxes involving VELMA's four soil layers are also shown. In each successive figure, a red encircling line delineates a subset of the LSR system for which text is provided to describe associated model parameters and equations that govern simulated C and N pools and fluxes. Note that parameter names in the VELMA version 2.0 program Java code and GUI do not necessarily conform to Figures A1.1 – A1.11 parameter names, which were modified using more rigorous naming rules during the VELMA programming process.

Figure A1.1



BiomassLeafN pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial state value must be specified

NppTotalN = calculated as before using Michaelis-Menten equation

NppToBiomassLeafN = NppTotalN * (NppTotalNFractionToBiomassLeafN, 0-1)

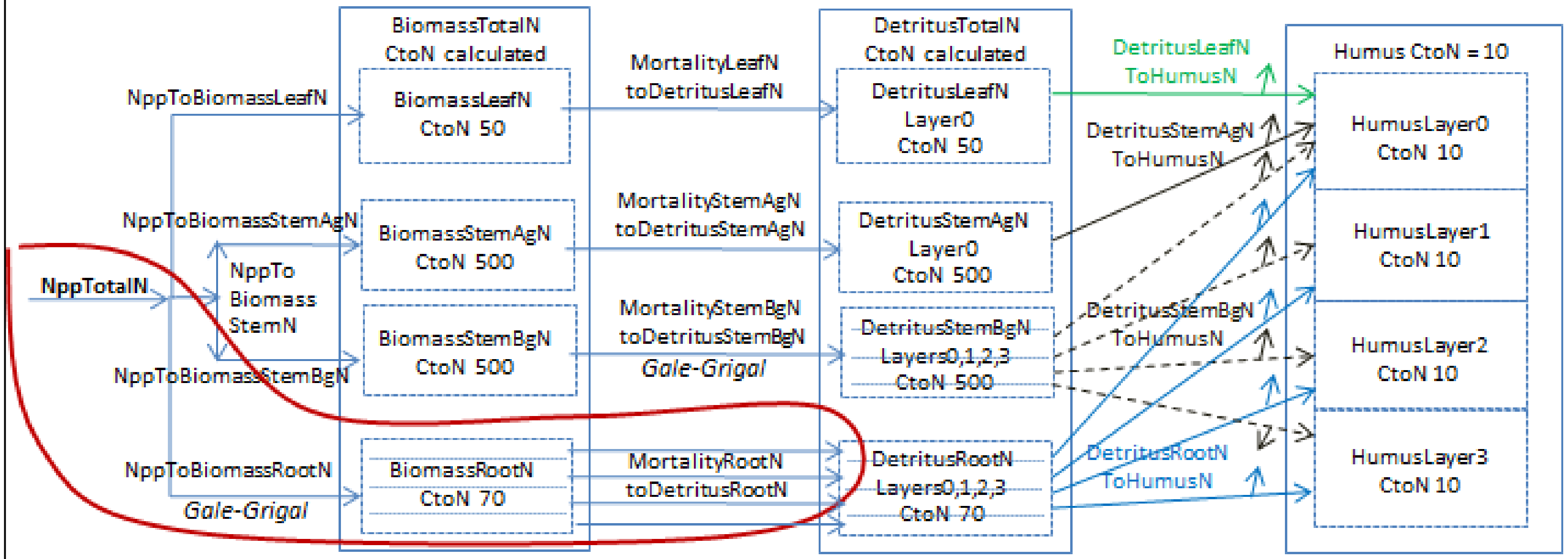
MortalityLeafNtoDetritusLeafN = BiomassLeafN * (MortalityLeafNFraction, 0-1)

BiomassLeafN = BiomassLeafN + NppToBiomassLeafN – MortalityLeafNtoDetritusLeafN
where BiomassLeafN vs time must reflect successional accumulation

BiomassLeafC pools & fluxes in gC/m^2

BiomassLeafC = BiomassLeafN * BiomassLeafCtoN

Figure A1.2



BiomassRootN pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial state value must be specified

NppTotalN = calculated as before using Michaelis-Menten equation

NppToBiomassRootN = NppTotalN * (NppTotalNFractionToBiomassRootN, 0-1)

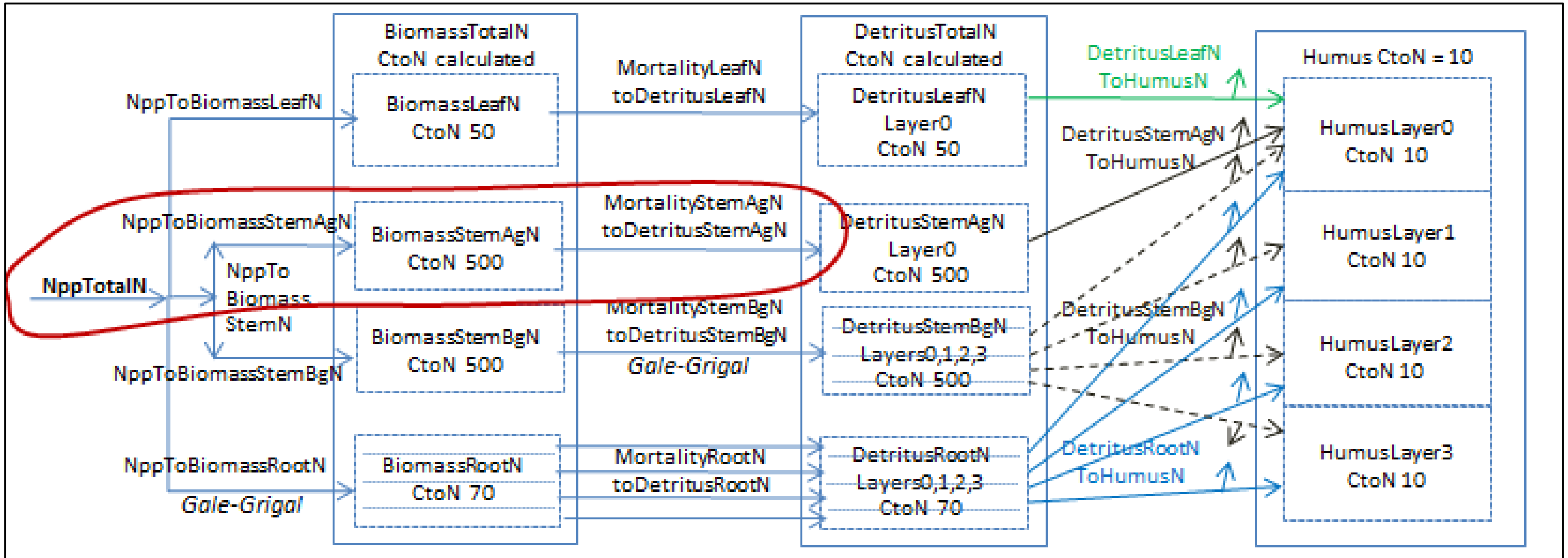
MortalityRootNtoDetritusRootN = BiomassRootN * (MortalityRootNFractiontoDetritusRootN, 0-1)

BiomassRootN = BiomassRootN + NppToBiomassRootN – MortalityRootNtoDetritusRootN
where BiomassLeafN vs. time must reflect successional accumulation

BiomassRootC pools g C m^{-2}

BiomassRootC = BiomassRootN * BiomassRootCtoN

Figure A1.3



BiomassStemAgN pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial state value must be specified

NppTotalN = calculated as before using Michaelis-Menten equation

NppToBiomassStemN = NppTotalN - NppToBiomassLeafN – NppToBiomassRootN), e.g., 6stem gN/m2 = 12 g total – 3 g leaf – 3 g root

NppToBiomassStemAgN = NppToBiomassStemN * (NppTotalNFractionToStemAgN, 0-1), e.g., 4.8 gN/m2 = 6 g stem * 0.8

Cover-specific StemAg mortality:

(1) Non-woody cover types (grasslands, crops, etc.):

MortalityStemAgNtoDetritusStemAgN = BiomassStemAgN * (MortalityStemAgNFractiontoDetritusStemAgN, 0-1)

(2) Woody cover types (forests, shrubs, etc.):

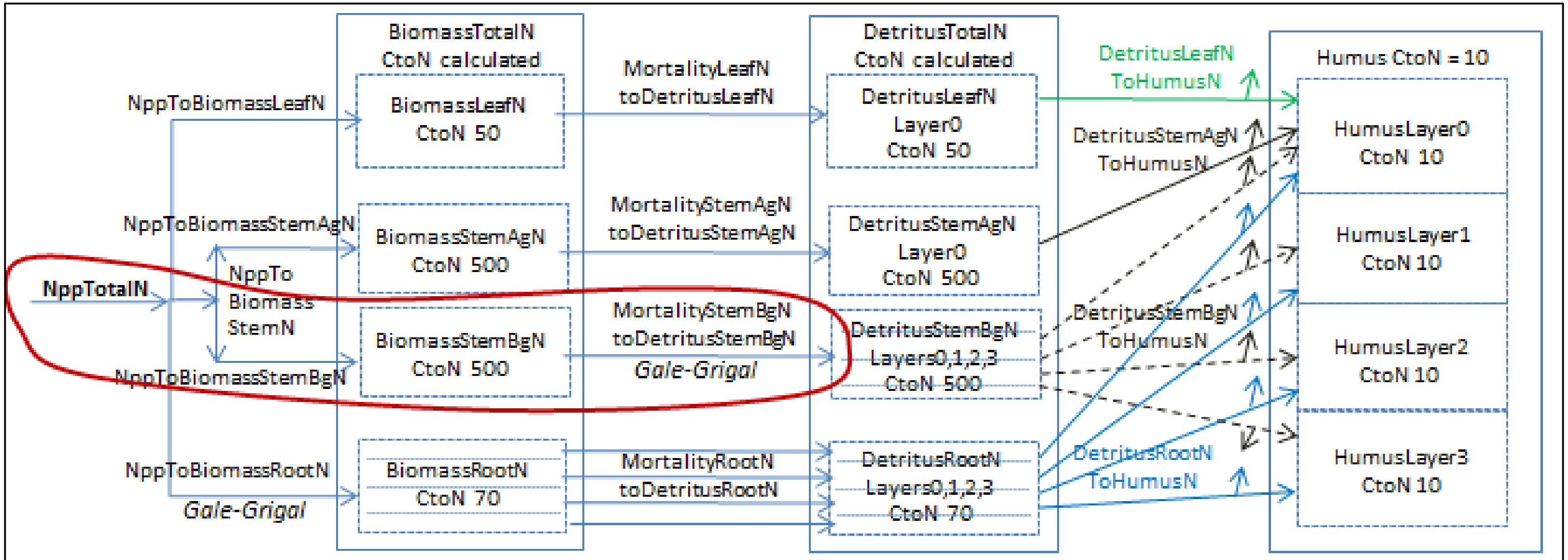
MortalityToDetritusStemAgN = NppToBiomassStemAgN * C / (1 + A * exp^(-B * BiomassStemAgN))
(see Excel mortality file for estimating parameters A, B and C

BiomassStemAgN = BiomassStemAgN + NppToBiomassStemAgN – MortalityStemAgNtoDetritusStemAgN

BiomassStemAgC pools & fluxes in gC/m^2

BiomassStemAgC = BiomassStemAgN * BiomassStemAgCtoN

Figure A1.4



BiomassStemBgN pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial state value must be specified

NppTotalN = calculated as before using Michaelis-Menten equation

NppToBiomassStemBgN = NppToBiomassStemN * (1 - NppTotalNFractionToStemAgN)

Cover-specific StemBg mortality:

(1) Non-woody cover types (grasslands, crops, etc.):

MortalityStemBgNtoDetritusStemBgN = BiomassStemBgN * (MortalityStemBgNFractiontoDetritusStemBgN, 0-1)
where mortality fraction ~ 0.33, such that 0.33 mortality fraction → 1/0.33 = 3 yr residence time (Potter 1999 Table 1)

(2) Woody cover types (forests, shrubs, etc.):

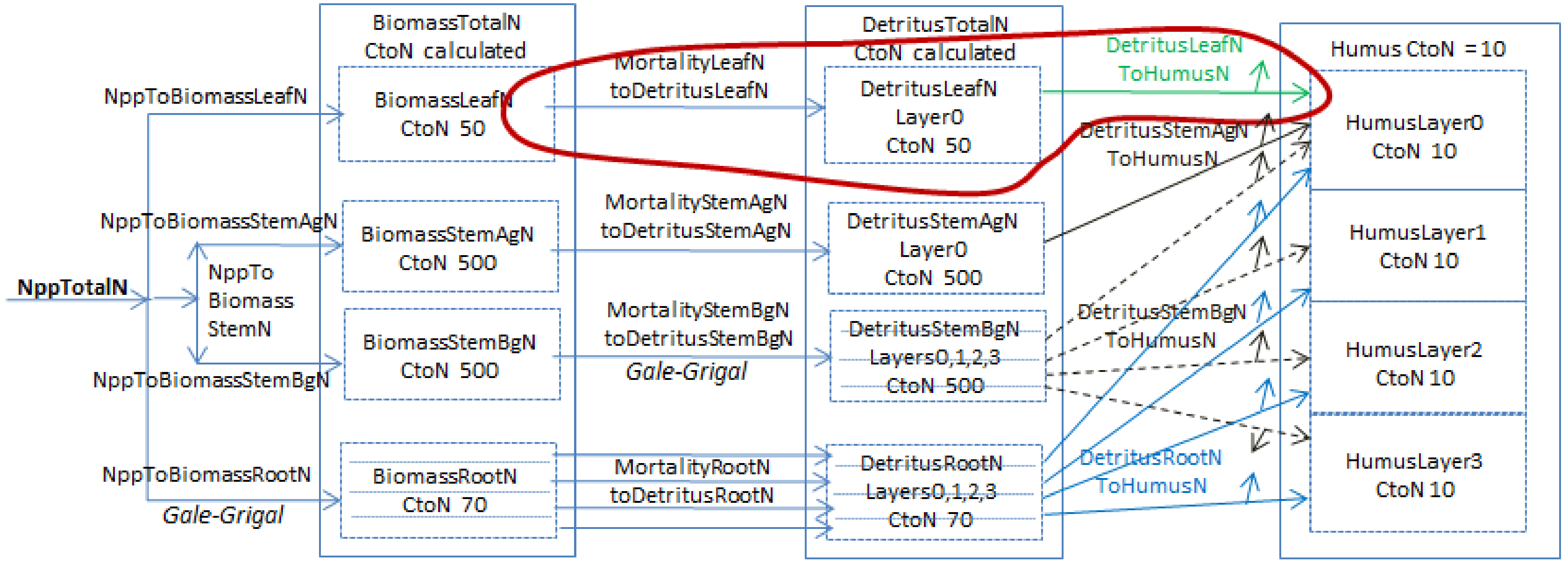
MortalityToDetritusStemBgN = NppToBiomassStemBgN * C / (1 + A * exp^(-B * BiomassStemBgN))
(see Excel mortality file for estimating parameters A, B and C)

BiomassStemBgN = BiomassStemBgN + NppToBiomassStemBgN – MortalityStemBgNtoDetritusStemBgN

BiomassStemBgC pools & fluxes in gC/m^2

BiomassStemBgC = BiomassStemBgN * BiomassStemBgCtoN

Figure A1.5



NOTE: If VELMA PSM calculations for a given day are started here, then you will need to delete “MortalityLeafNtoDetritusLeafNLayer0” from the from the 2nd equation, below. For example,

DetritusLeafNLayer0 = DetritusLeafNLayer0 – DetritusLeafNtoHumusNLayer0.

This means that the deleted mortality term will need to be added to the affected detritus pool at the end of the day, such that: DetritusLeafNLayer0 = DetritusLeafNLayer0 + MortalityLeafNtoDetritusLeafNLayer0.

These steps will need to be applied to all of the Detritus pools (Leaf, Stem, Root).

DetritusLeafNLayer0 pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial state value must be specified

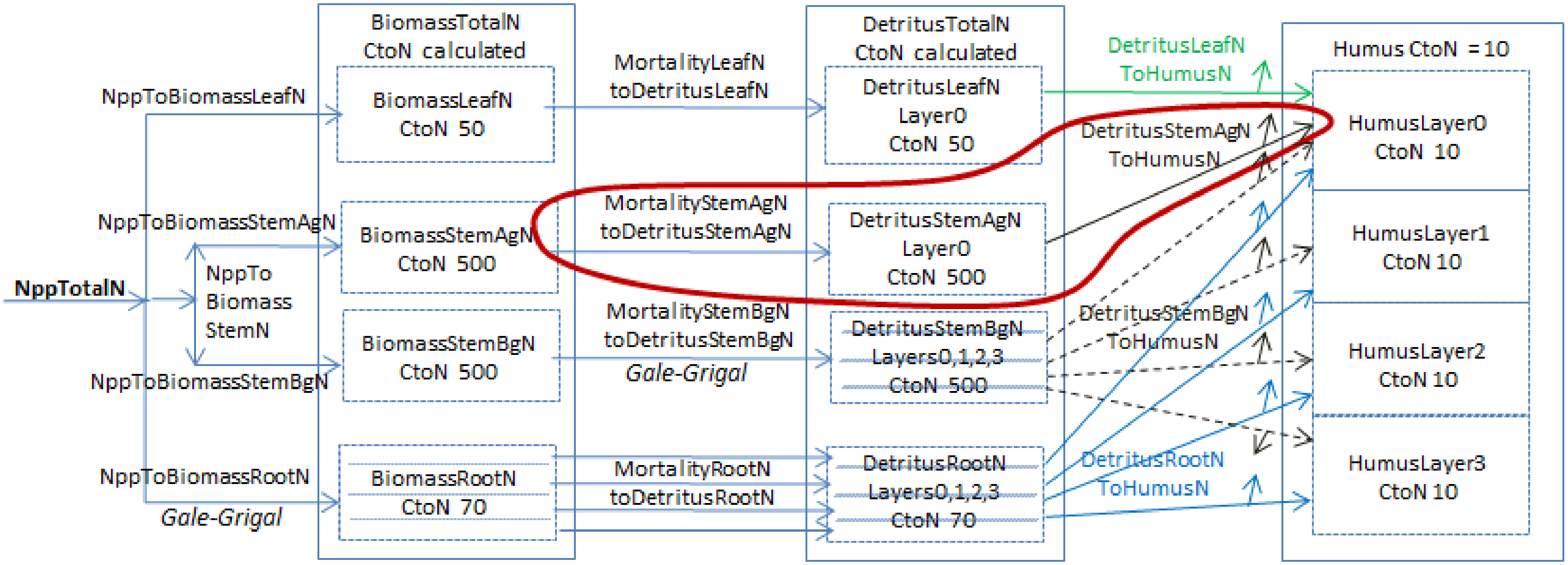
DetritusLeafNtoHumusN = Psm4d.getPotterDecompositionAmount(DetritusLeafNT1[Layer0], curCover.DetritusLeafNMaxDecay, wfps,groundT, microbeEfficiency); *Notes: (1) this is a copy of the current PSM4d code, but with edited variable names; (2) in the case of DetritusLeafN and DetritusStemAg,* all DetritusStemAgNtoHumus goes to Layer0 (i.e., no GaleGrigal partitioning).

DetritusLeafNLayer0 = DetritusLeafNLayer0 + MortalityLeafNtoDetritusLeafNLayer0 - DetritusLeafNtoHumusNLayer0

DetritusLeafCLayer0 pools & fluxes in gC/m^{2}

DetritusLeafCLayer0 = DetritusLeafNLayer0 * DetritusLeafCtoN

Figure A1.6



DetritusAgNLayer0 pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial state value must be specified

DetritusStemAgNtoHumusN = Psm4d.getPotterDecompositionAmount(DetritusStemAgNT1[Layer0], curCover.DetritusStemAgNMaxDecay, wfps, groundT, microbeEfficiency)

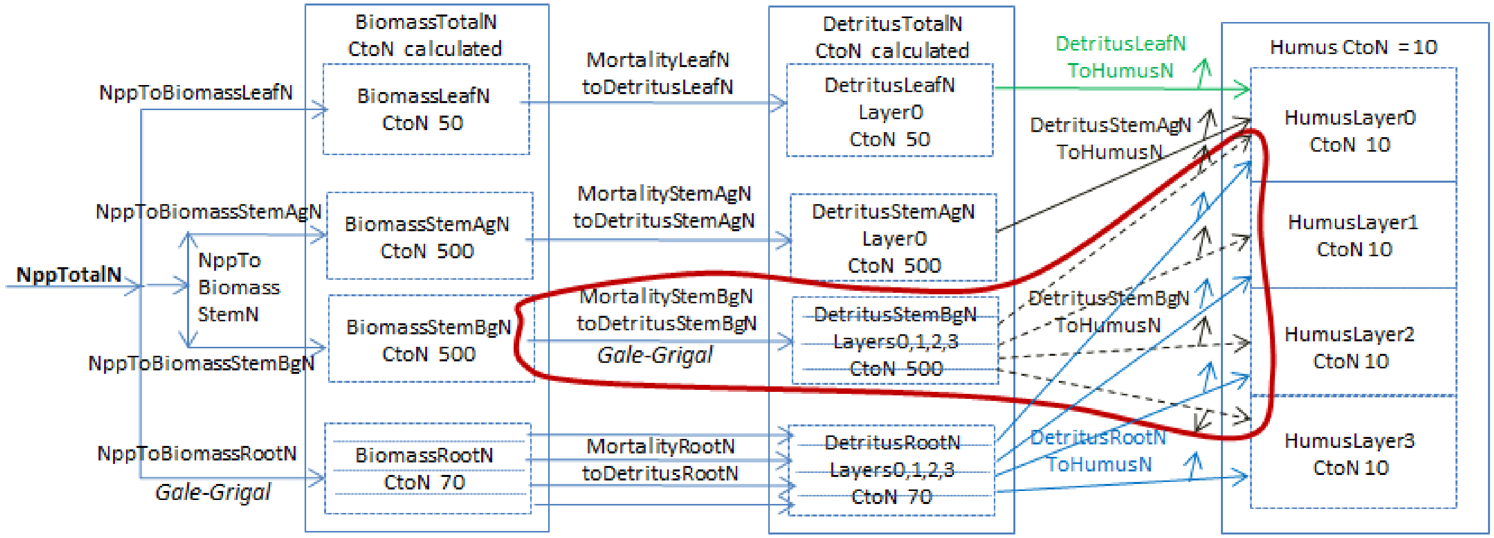
Notes: (1) this is a copy of the current PSM4d code, but with edited variable names; (2) in the case of DetritusLeafN and DetritusStemAg, all DetritusStemAgNtoHumus goes to Layer0 (i.e., no GaleGrigal partitioning).

DetritusStemAgNLayer0 = DetritusStemAgNLayer0 + MortalityStemAgNtoDetritusLeafNLayer0 - DetritusStemAgNtoHumusNLayer0

DetritusStemAgCLayer0 pools & fluxes in gC/m^2

DetritusStemAgCLayer0 = DetritusStemAgNLayer0 * DetritusStemAgCtoN

Figure A1.7



DetritusStemBgNLayer*i* (for *i* = 0 , 1, 2 or 3) pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial pool value must be specified

DetritusStemBgNtoHumusN = Psm4d.getPotterDecompositionAmount(DetritusStemBgNT1[Layeri], curCover.DetritusStemBgNMaxDecay, wfps, groundT, microbeEfficiency)
Notes: (1) this is a copy of the current PSM4d code, but with edited variable names; (2)
Psm4d.getPotterDecompositionAmount(DetritusStemBgNT1[Layer*i*] is used to calculate total (all layer) decomposition, which is then partitioned among Layers0,1,2,3 according to GaleGrigal parameter.

DetritusStemBgNLayer0 = DetritusStemBgNLayer0 + MortalityStemBgNtoDetritusStemBgNLayer0 - DetritusStemBgNtoHumusNLayer0

DetritusStemBgNLayer1 = DetritusStemBgNLayer1 + MortalityStemBgNtoDetritusStemBgNLayer1 – DetritusStemBgNtoHumusNLayer1

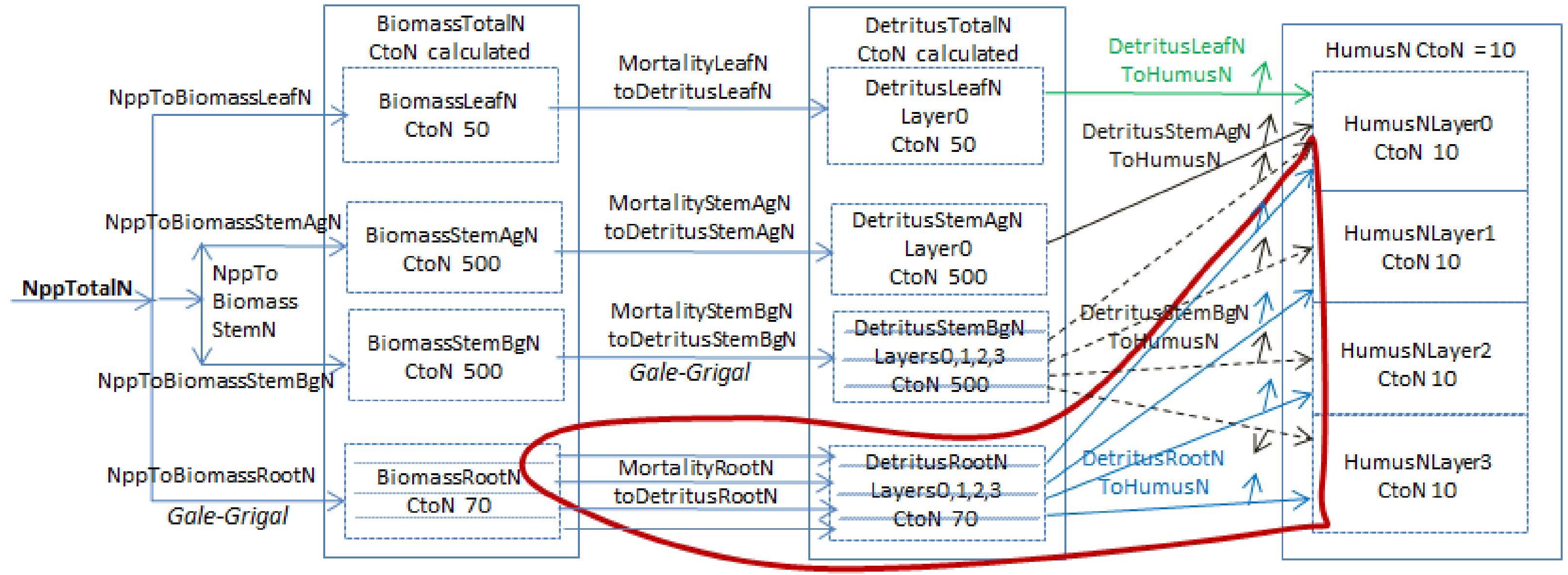
DetritusStemBgNLayer2 = DetritusStemBgNLayer2 + MortalityStemBgNtoDetritusStemBgNLayer2 - DetritusStemBgNtoHumusNLayer2

DetritusStemBgNLayer3 = DetritusStemBgNLayer3 + MortalityStemBgNtoDetritusStemBgNLayer3 – DetritusStemBgNtoHumusNLayer3

DetritusStemBgCLayer*i* (for *i* = 0 , 1, 2 or 3) pools & fluxes in gC/m^2

DetritusStemBgCLayeri= *DetritusStemBgNLayeri* * DetritusStemBgCtoN

Figure A1.8



DetritusRootNLayer*i* (for *i* = 0 , 1, 2 or 3) pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial pool value must be specified

DetritusRootNtoHumusN = Psm4d.getPotterDecompositionAmount(DetritusRootNT1[Layeri], curCover.DetritusRootNMaxDecay, wfps, groundT, microbeEfficiency)

Notes: (1) this is a copy of the current PSM4d code, but with edited variable names; (2) Psm4d.getPotterDecompositionAmount(DetritusRootNT1[Layer*i*] is used to calculate total (all layer) decomposition, which is then partitioned among Layers0,1,2,3 according to GaleGrigal parameter.

DetritusRootNLayer0 = DetritusRootNLayer0 + MortalityRootNtoDetritusRootNLayer0 - DetritusRootNtoHumusNLayer0

DetritusRootNLayer1 = DetritusRootNLayer1 + MortalityRootNtoDetritusRootNLayer1 – DetritusRootNtoHumusNLayer1

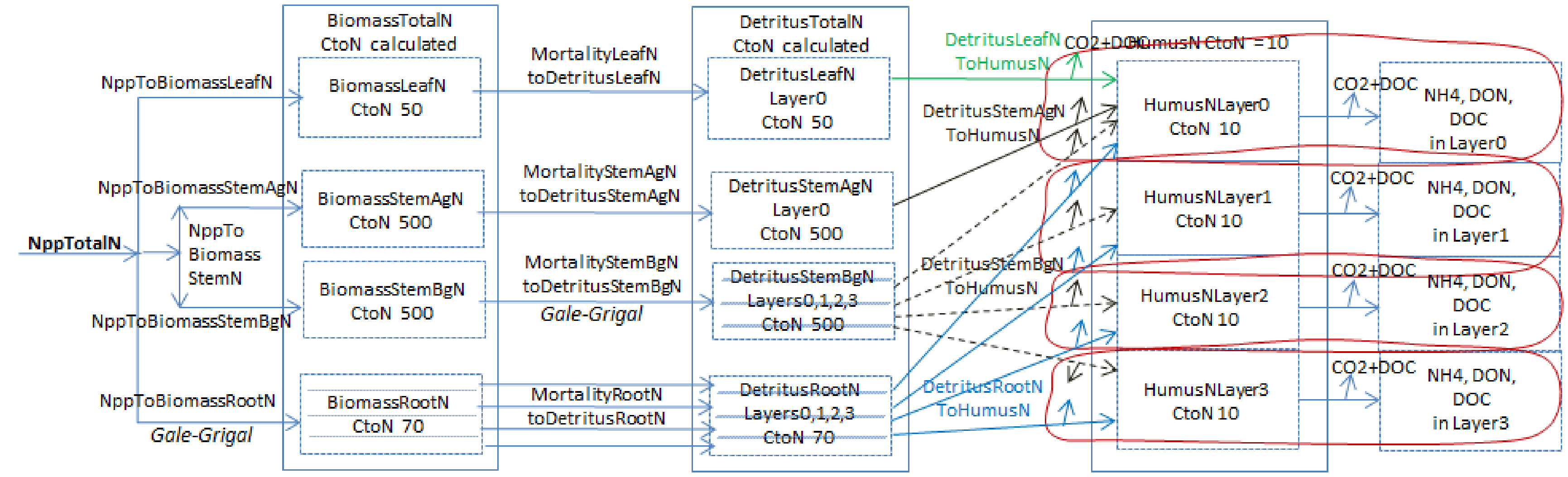
DetritusRootNLayer2 = DetritusRootNLayer2 + MortalityRootNtoDetritusRootNLayer2 - DetritusRootNtoHumusNLayer2

DetritusRootNLayer3 = DetritusRootNLayer3 + MortalityRootNtoDetritusRootNLayer3 – DetritusRootNtoHumusNLayer3

DetritusRootCLayer*i* (for *i* = 0 , 1, 2 or 3) pools & fluxes in gC/m^2

DetritusRootCLayeri = *DetritusRootNLayeri* * DetritusRootCtoN

Figure A1.9



HumusNLayer0,1,2,3 pools (g N m^{-2}) and fluxes (g N m^{-2} d^{-1}), where initial state value must be specified

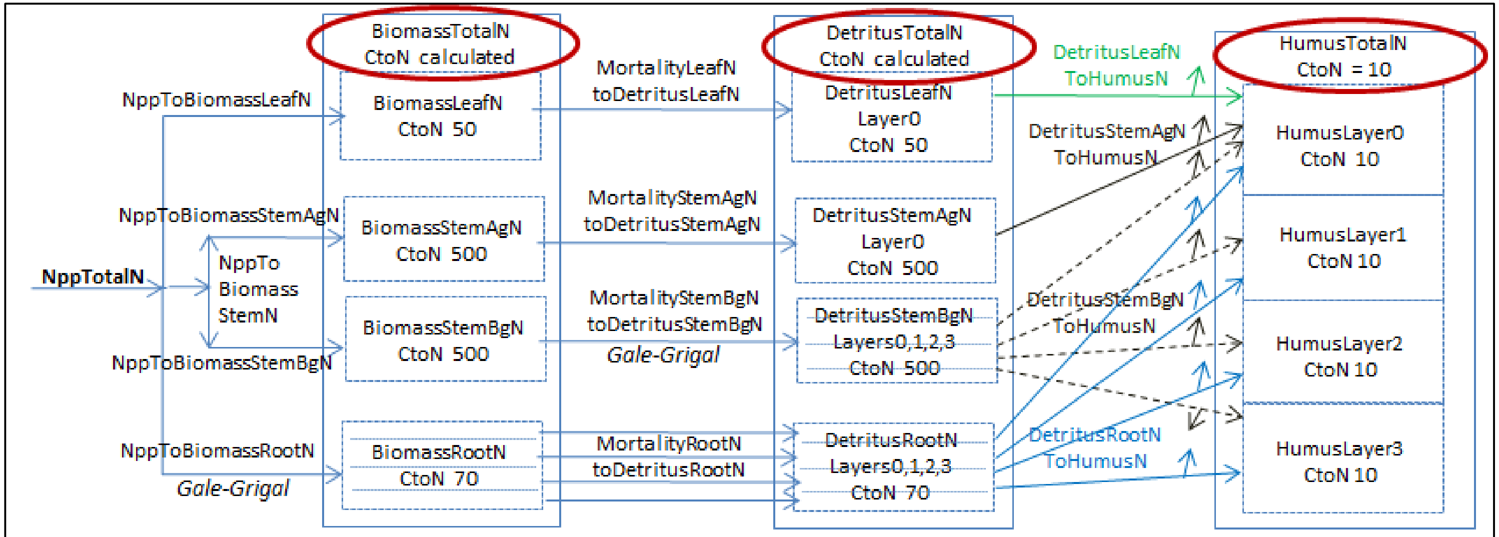
HumusNLayer0 = HumusNLayer0 + (DetritusLeafNtoHumusNLayer0 + DetritusStemAgNtoHumusNLayer0 + DetritusStemBgNtoHumusNLayer0 + DetritusRootNtoHumusNLayer0 – NH4 and DON loss from Layer0, *where NH4, DON and DOC fluxes are calculated per usual*

HumusNLayer1 = HumusNLayer1 + DetritusStemBgNtoHumusNLayer1 + DetritusRootNtoHumusNLayer1 – NH4 and DON loss from Layer1, *where NH4, DON and DOC fluxes are calculated per usual*

HumusNLayer2 = HumusNLayer2 + DetritusStemBgNtoHumusNLayer2 + DetritusRootNtoHumusNLayer2 – NH4 and DON loss from Layer2, *where NH4, DON and DOC fluxes are calculated per usual*

HumusNLayer3 = HumusNLayer3 + DetritusStemBgNtoHumusNLayer3 + DetritusRootNtoHumusNLayer3 – NH4 and DON loss from Layer3, *where NH4, DON and DOC fluxes are calculated per usual*

Figure A1.10



Total Nitrogen Pools for Biomass, Detritus and Humus in g N m^{-2} d^{-1}

BiomassTotalN = BiomassLeafN + BiomassStemAgN + BiomassStemBgN + BiomassRootN

DetritusTotalN = DetritusLeafN + DetritusStemAgN + DetritusStemBgN + DetritusRootN

HumusTotalN = HumusLeafN + HumusStemAgN + HumusStemBgN + HumusRootN

Total Carbon Pools for Biomass, Detritus and Humus in g C m^{-2} d^{-1}

BiomassTotalC = BiomassLeafC + BiomassStemAgC + BiomassStemBgC + BiomassRootC

DetritusTotalC = DetritusLeafC + DetritusStemAgC + DetritusStemBgC + DetritusRootC

HumusTotalC = HumusLeafC + HumusStemAgC + HumusStemBgC + HumusRootC

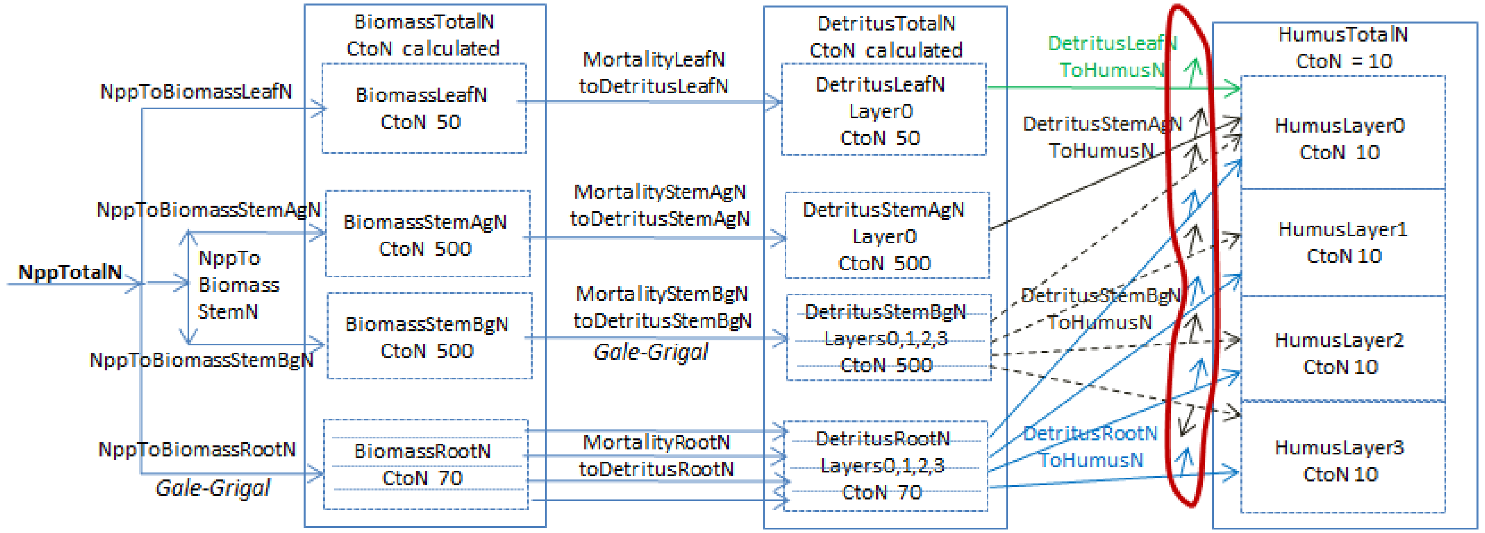
Total CtoN for Biomass, Detritus and Humus (unitless ratio)

BiomassTotalCtoN = (BiomassLeafC + BiomassStemAgC + BiomassStemBgC + BiomassRootC) / (BiomassLeafN + BiomassStemAgN + BiomassStemBgN + BiomassRootN)

DetritusTotalCtoN = (DetritusLeafC + DetritusStemAgC + DetritusStemBgC + DetritusRootC) / (DetritusLeafN + DetritusStemAgN + DetritusStemBgN + DetritusRootN)

HumusTotalCtoN = (HumusLeafC + HumusStemAgC + HumusStemBgC + HumusRootC) / (HumusLeafN + HumusStemAgN + HumusStemBgN + HumusRootN)

Figure A1.11



DetritusCtoCO2+DOC fluxes (g C m^{-2} d^{-1})

DetritusLeafCtoCO2+DOC = (DetritusLeafNtoHumusN *DetritusLeafCtoN) - (DetritusLeafNtoHumusN *HumusCtoN)

DetritusAgStemCtoCO2+DOC = (DetritusAgStemNtoHumusN *DetritusAgStemCtoN) - (DetritusAgStemNtoHumusN *HumusCtoN)

DetritusBgStemCtoCO2+DOC = (DetritusBgStemNtoHumusN *DetritusBgStemCtoN) - (DetritusBgStemNtoHumusN *HumusCtoN)

DetritusRootCtoCO2+DOC = (DetritusRootNtoHumusN *DetritusRootCtoN) - (DetritusRootNtoHumusN *HumusCtoN)

DetritusTotalCtoCO2+DOC = DetritusLeafCtoCO2+DOC + DetritusAgStemCtoCO2+DOC + DetritusBgStemCtoCO2+DOC + DetritusRootCtoCO2+DOC

DetritusCtoCO2 flux (g C m-2 d-1)

DetritusLeafCtoCO2 = DetritusLeafCtoCO2+DOC * DecomposedCarbonFractionToCO2

DetritusAgStemCtoCO2 = DetritusAgStemCtoCO2+DOC * DecomposedCarbonFractionToCO2

DetritusBgStemCtoCO2 = DetritusBgStemCtoCO2+DOC * DecomposedCarbonFractionToCO2

DetritusRootCtoCO2 = DetritusRootCtoCO2+DOC * DecomposedCarbonFractionToCO2

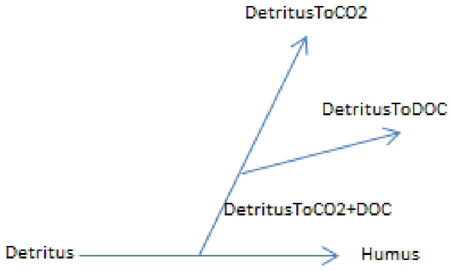
DetritusCtoDOC flux (g C m-2 d-1)

DetritusLeafCtoDOC = DetritusLeafCtoCO2+DOC * (1 - DecomposedCarbonFractionToCO2)

DetritusAgStemCtoDOC = DetritusAgStemCtoCO2+DOC * (1 - DecomposedCarbonFractionToCO2)

DetritusBgStemCtoDOC = DetritusBgStemCtoCO2+DOC * (1 - DecomposedCarbonFractionToCO2)

DetritusRootCtoDOC = DetritusRootCtoCO2+DOC * (1 - DecomposedCarbonFractionToCO2)



Appendix 2: Initializing a Spatial Data Pool Using an ASCII Grid

(Link to All Parameters Table of Contents)

VELMA Simulator spatial data pools may be initialized by specifying “Set Spatial Data By Map” disturbances as part of a simulator run’s simulation configuration. Each spatial data pool (and each layer of that pool for multi-layer pools) that is initialized must have a separate disturbance specified for it, and its own Grid ASCII (.asc) map file available for the disturbance to use as a data source[3].

Assuming an otherwise already configured simulation configuration is loaded into JVelma, here is how to initialize a spatial data pool (using the leaf biomass pool as our example) by adding a Set Spatial By Map disturbance event to the configuration:

Note

It’s assumed that before configuring the simulation, you’ve already created a Grid ASCII map file (.asc) for the disturbance to use. How you create the map is up to you, but it must in standard Grid ASCII format, and have the same dimensions as the Grid ASCII map of DEM data that the simulation will use.
For this example, each cell value in the Grid ASCII map file is assumed to be an amount of leaf biomass (in grams per square meter).

Add the New Disturbance Parameters to the Simulation Configuration

1. In JVelma, with the simulation configuration loaded, Click/open the “Edit” menu and Click/Select “Disturbances “ -> “Add a Disturbance”.
2. In the Disturbance Model Selector dialog, use the drop-down selector to find the “SetSpatialDataByMapDisturbanceModel” and Click/Set it as the Disturbance Type.
3. In the text box next to the “DisturbanceName” prompt, type in a name for this particular disturbance. The name should be unique (the dialog will warn you if you enter a name already in use) and it is better to avoid whitespace and non-alphanumeric characters.
   For this is example, the name will be “Init_Leaf_Biomass”.
4. Click the “OK” button. The dialog closes, the parameters for a new Set Spatial Data By Map disturbance are added to the simulation configuration, and the view should shift to these new parameters in JVelma’s “All Parameters” tab window.

Configure the New Disturbance Parameters

The parameters for the Set Spatial Data By Map disturbance tell the disturbance When it should occur, What its data source is, and Which Spatial Data Pool to apply that data source to.

[3] Two or more Set Spatial Data By Map disturbances may be set to use the same Grid ASCII map file as their input data source, but it’s more likely that a given spatial data pool to have its own unique input data source.

The initializeActiveLoops, initializeActiveYears” and initializeActiveJdays parameters, plus the occursAtStepStart parameter specify when this disturbance should occur. To initialize the leaf biomass pool before any simulation steps occur, set them as follows:

Parameter Name	Set Value As …	Comment
initializeActiveLoops	1	Occurs during the first loop
initializeActiveYears	2000	Assumes 2000 is the simulation’s start year
initializeActiveJdays	1	Occurs during the first Julian day
occursAtStepStart	true	Disturbance will occur before any simulation work is performed on the day when it occurs

Set the spatialDataFileFullName parameter to the fully-qualified path and name of the Grid ASCII file that you plan to use as the input data source for biomass leaf values.
For this example, suppose that file is named “LeafBiomassData.asc” and resides in directory “C:\My Data\”. The spatialDataFileName value should then be set to: “C:/My Data/LeafBiomassData.asc” (without the double-quotes).
Note
Instead of Windows backslash “\” path separators, use forward slash “/” characters as path separators.
Also, be very careful with whitespace; if there is any, it must *exactly* match whatever whitespace is present in the actual path and filename.

Set the spatialDataName parameter to the Spatial Data Pool that will receive data from the file.
For our example of leaf biomass, the pool name is “BIOMASS_LEAF_N” (without the double-quotes).
Note
A table containing the names of the Velma Simulator’s spatial data pools is available at the end of this document.

Set the spatialDataLayer parameter to the (one-based) layer index number of the spatial data pool layer that will receive data from the file.
For single-layer spatial data pools this value MUST be set to “1” see the table of spatial data at the end of this document for a summary of how many layers each pool has.

In summary, here is the full parameterization for our example Set (Leaf Biomass) Spatial Data Pool By Map disturbance:

ID Key	Value
/disturbance/Init_Leaf_Biomass/initializeActiveJdays	1
/disturbance/Init_Leaf_Biomass/initializeActiveLoops	1
/disturbance/Init_Leaf_Biomass/initializeActiveYears	2000
/disturbance/Init_Leaf_Biomass/modelClass	SetSpatialDataByMapDisturbanceModel
/disturbance/Init_Leaf_Biomass/occursAtStepStart	true
/disturbance/Init_Leaf_Biomass/spatialDataFileFullName	C:/My Data/LeafBiomassData.asc
/disturbance/Init_Leaf_Biomass/spatialDataLayer	1
/disturbance/Init_Leaf_Biomass/spatialDataName	BIOMASS_LEAF_N

Configured as specified by the parameters above, this SetSpatialDataByMapDisturbanceModel disturbance occurs once during the simulation; in the first loop, on the first day of 2000 (assumed to be the first year of simulation for our example) and before any simulation work is done on that day.
When the disturbance occurs, it will open the Grid ASCII spatial data file
"C:/My Data/LeafBiomassdata.asc", and set each cell of the BIOMASS_LEAF_N spatial data pool in layer 1 to the value for that cell read from the .asc spatial data file.

That's all there is to it!
Although SetSpatialDataByMapDisturbanceModel disturbances may be used for other purposes, by varying when they occur, this example shows the commonest use-case; initializing specific non-uniform values into a spatial data pool at the start of a simulation.

Appendix 3:
Creating Initial ASCII Grid Chemistry Spatial Data Pools

(Link to All Parameters Table of Contents)

Initial chemistry or nutrient pools are created through a series of transformations of above ground Biomass to each of the respective spatial data pools. These transformations are defined by relationship ratios (fractions) each pool has to "real" or estimated measurements of Biomass (g/m²) for the study area.
37 VELMA Simulator spatial data pools will be created from this one raster grid Biomass layer. See Table 1 for the complete list of initial layers that will be created. Each spatial data pool layer will have its own corresponding Grid ASCII (.asc) map file upon completion of the Python script. These Grid ASCII (.asc) map files will be used as inputs to VELMA for initializing the map disturbance events. Figure 1 describes the internal processing flow, inputs, and outputs of the script. Figure 2 describes the calculations and quality control of the internal Python processing.

Software Requirements

- ArcGIS 10.0 or higher with "Spatial Analyst" license
- NumPy package installed, for the Python version used by ArcGIS 10.X
- IDLE – Python's Integrated Development Environment installed

File Requirements for Data Processing

- Raster above ground biomass of the study area
  - Unit type – Carbon grams / m²
  - ".tif" extension
  - Raster has been clipped and snapped to same dimensions as the Grid ASCII map of DEM data that the simulation will use.
- Summary Biomass to ratio pools file
  - ".csv" comma delaminate format
  - No headers
  - Biomass values (stand age / mass) is in the first column
  - Pool ratios values in subsequent columns (quantity 37)
  - Column arrangement /order is known for the ratios

Setting up Python Script for Data Processing

The script is set up in two general sections, "User defined inputs" and "Processing functions". "User defined inputs" is where inputs, outputs, and column order will be defined or edited. Changes to the "Processing functions" section of the Python code should be avoided.
Note
While editing "User defined inputs"

- *All file and folder paths need to be enclosed in double quotes: "C:/Temp/Biomass/"*

- *All paths in Python use a "/" instead of the DOS "\"*
- *Folder defined paths need to end in "/" "C:/Temp/Biomass/"*
- *File defined paths need to end with correct extension "C:/Temp/Biomass/C_g_leaf.tif"*
- *Ensure that all final output working directories are the same*

Note
Biomass to ratio pools file

- *Column arrangement must be maintained as noted above*

Editing the Python Script

1. Right click the "lookup_ratio_calculate_biomass_func.py" script and click "Edit With IDLE"
2. Editing within the "User defined inputs" section
3. Edit all path variables for:
    - "biomassLayer" – Biomass file
    - "inputPath" – Main output directory
    - "massRatio" – ".csv" Summary Biomass to ratio pools file
    - Ouput rasters files, for each of the velma attributes - **".tif"** extension
    - Column number of pool ratio to biomass – **Order specific**
    - Carbon to nitrogen ratios
    - Raster attributes
        - xllcorner
        - yllcorner
        - cellsize
        - nodata
4. Pres Ctrl+S to save the file
5. Press F5 to run!

Note
Processing time is very dependent on processing power and the size of your study area. For example the Blue River, in Oregon, had a study grid of ~350,000 cells and took 7-12hours to complete all 37 pools. The Python shell will print updates of pool processing steps, and say "All Done!" when it's done.

Table 1. Initial spatial pools derived from ratios to above ground biomass (C g/m^2)

Spatial Data Pool Name	Layers	Unit Type
BIOMASS_AG_STEM_N	1	N g/m^2
BIOMASS_BG_STEM_N	1	N g/m^2
BIOMASS_LEAF_N	1	N g/m^2
BIOMASS_ROOT_N	4	N g/m^2
DETRITUS_AG_STEM_N	1	N g/m^2
DETRITUS_BG_STEM_N	4	N g/m^2
DETRITUS_LEAF_N	1	N g/m^2
DETRITUS_ROOT_N	4	N g/m^2
DOC	4	C g/m^2
DON	4	N g/m^2
HUMUS	4	N g/m^2
NH4	4	N g/m^2
NO3	4	N g/m^2

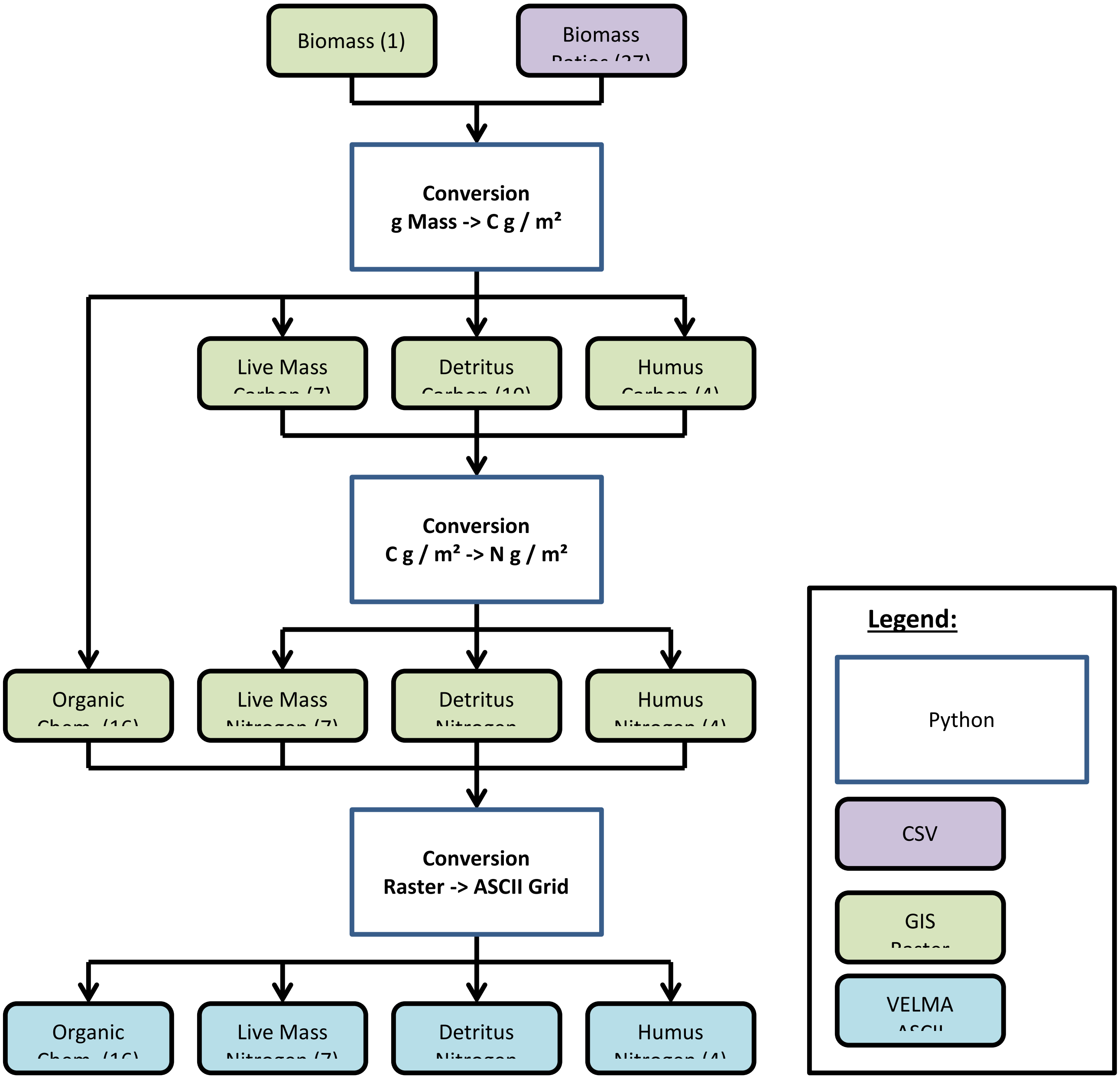
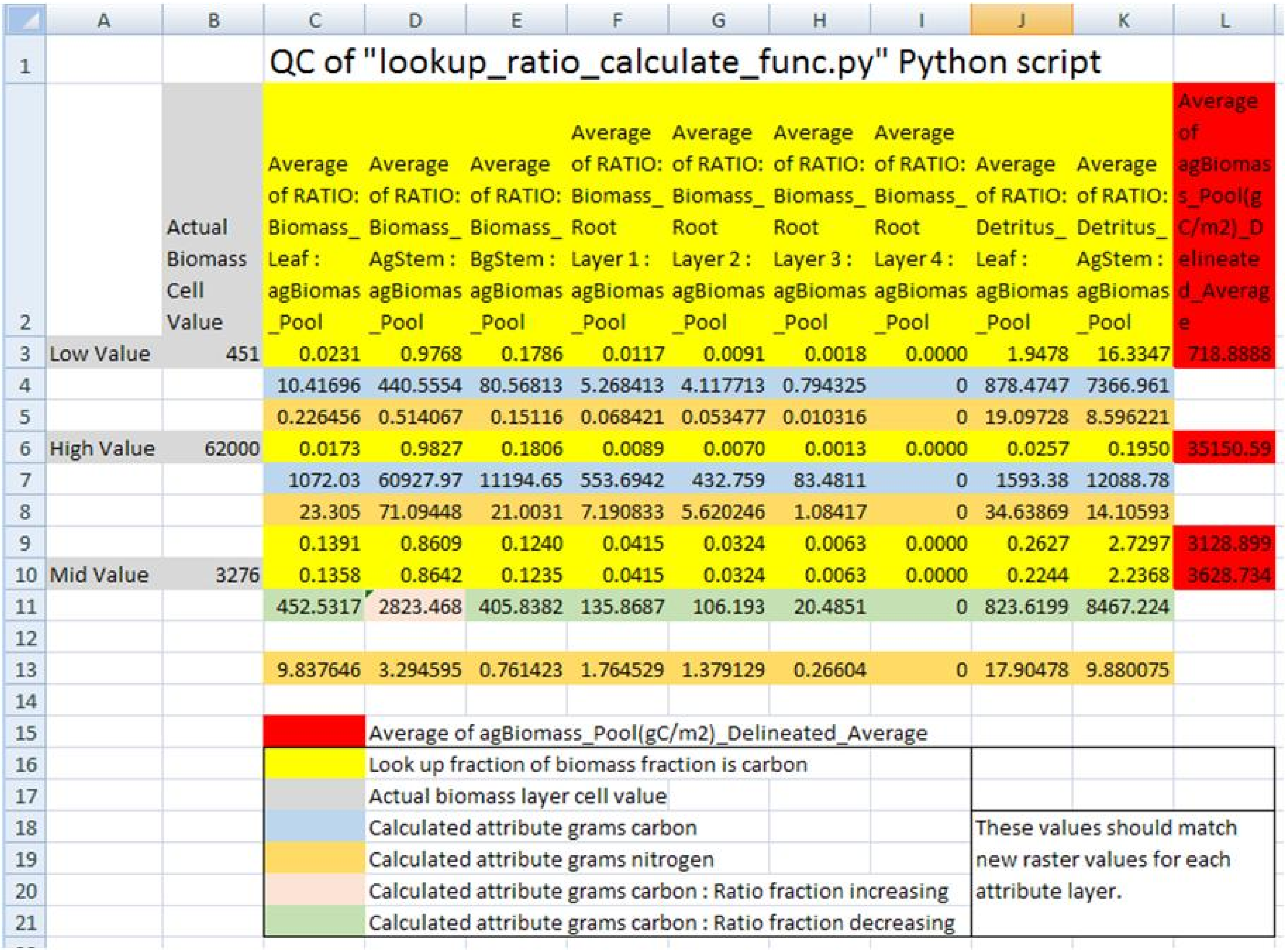


Figure 1. Process flow diagram of ratio (fractions) processing of “observed” above ground biomass (g / m^2) to the final nitrogen (g / m^2) spatial data pools for VELMA’s initialization of disturbance events.



	A	B	C	D	E	F	G	H	I	J	K	L
1			QC of "lookup_ratio_calculate_func.py" Python script									
2		Actual Biomass Cell Value	Average of RATIO: Biomass_Leaf : agBiomas_Pool	Average of RATIO: Biomass_AgStem : agBiomas_Pool	Average of RATIO: Biomass_BgStem : agBiomas_Pool	Average of RATIO: Biomass_Root Layer 1 : agBiomas_Pool	Average of RATIO: Biomass_Root Layer 2 : agBiomas_Pool	Average of RATIO: Biomass_Root Layer 3 : agBiomas_Pool	Average of RATIO: Biomass_Root Layer 4 : agBiomas_Pool	Average of RATIO: Detritus_Leaf : agBiomas_Pool	Average of RATIO: Detritus_AgStem : agBiomas_Pool	Average of agBiomass_Pool(gC/m2)_Delineated_Average
3	Low Value	451	0.0231	0.9768	0.1786	0.0117	0.0091	0.0018	0.0000	1.9478	16.3347	718.8888
4			10.41696	440.5554	80.56813	5.268413	4.117713	0.794325	0	878.4747	7366.961	
5			0.226456	0.514067	0.15116	0.068421	0.053477	0.010316	0	19.09728	8.596221	
6	High Value	62000	0.0173	0.9827	0.1806	0.0089	0.0070	0.0013	0.0000	0.0257	0.1950	35150.59
7			1072.03	60927.97	11194.65	553.6942	432.759	83.4811	0	1593.38	12088.78	
8			23.305	71.09448	21.0031	7.190833	5.620246	1.08417	0	34.63869	14.10593	
9			0.1391	0.8609	0.1240	0.0415	0.0324	0.0063	0.0000	0.2627	2.7297	3128.899
10	Mid Value	3276	0.1358	0.8642	0.1235	0.0415	0.0324	0.0063	0.0000	0.2244	2.2368	3628.734
11			452.5317	2823.468	405.8382	135.8687	106.193	20.4851	0	823.6199	8467.224	
12												
13			9.837646	3.294595	0.761423	1.764529	1.379129	0.26604	0	17.90478	9.880075	
14												
15			🟥	Average of agBiomass_Pool(gC/m2)_Delineated_Average								
16			🟨	Look up fraction of biomass fraction is carbon								
17			⬜	Actual biomass layer cell value								
18			🟦	Calculated attribute grams carbon						These values should match new raster values for each attribute layer.		
19			🟧	Calculated attribute grams nitrogen								
20			🟪	Calculated attribute grams carbon : Ratio fraction increasing								
21			🟩	Calculated attribute grams carbon : Ratio fraction decreasing								

Figure 2. Quality control check of Python script's dynamic conversions of "observed" above ground biomass (g / m²) to the final nitrogen (g / m²) spatial data pools. Conversion of biomass (g / m²) to the final nitrogen (g / m²) were made through an intermediate step of converting mass to carbon (g / m²), and then using different C to N ratios for each pool to calculate the final nitrogen pools. Fractions (ratios) of spatial pools change both in the positive and negative direction as a forest ages or increases in total biomass. Ratio pools can increases or decreases between stand age increments, and the calculations of these intermediate biomass observational values must reflect this dynamic nature of changing ratios between time steps. Intermediate values of biomass, those that land in between time step years, were therefore linearly interpolated between the time point bounds. Spatial map ASCII grid values for each of the 37 pools should then backward match ratio conversions and final C to N calculations to the original "observed" above ground biomass. (Only a partial list of pools is displayed here).

Example: A mid biomass cell value at B10 = 3276 (g / m²) and is part way between stand age values at L9 and L10. It's above ground stem carbon mass fraction increases from 0.8609 (D9) to 0.8642 (D10). So its fraction stem mass (column D) should increase as its percentage difference moves from L9 towards LD10. Conversely, for all the other pool attributes, its percentage difference would increase as its fraction moves the opposite direction from rows 10 to rows 9.

Appendix 4:
Overview of VELMA Simulator Output

[***(Link to All Parameters Table of Contents)***]

Where are VELMA Simulator Results Located?

The results of a VELMA Simulator run are a collection of data files that the simulator writes to a specific directory location. That directory is specified in the simulator configuration by 2 related parameters:

```
/calibration/VelmaInputs.properties/initializeOutputDataLocationRoot
```

and

```
/calibration/VelmaInputs.properties/run_index
```

The `initializeOutputDataLocationRoot`'s value should be a fully qualified path, and the `run_index` should be the name of a directory under that path.

For example, given the following configuration:

ID Key	Value
`/calibration/VelmaInputs.properties/initializeOutputDataLocationRoot`	`C:/MyVELMA_OutputData`
`/calibration/VelmaInputs.properties/run_index`	`LittleMtn_ClearCut`

All results for the simulation run are written to the directory:

```
C:/MyVELMA_OutputData/LittleMtn_ClearCut
```

If the directory name specified as the value for `run_index` does not exist as a subdirectory of the directory path specified for `initializeOutputDataLocationRoot`, the VELMA simulator will create it as a subdirectory before writing any results.

If the `run_index` directory name already exists as a subdirectory of the `initializeOutputDataLocationRoot` directory path, the VELMA simulator does not overwrite existing contents. Instead, it creates a new subdirectory named `<run_index>_<n>`, where `<run_index>` is the value of the `run_index` parameter, and `<n>` is a unique sequence number. This allows you to run the same simulation several times, without losing the results for any of the individual simulation runs.

If you ran the simulation run with the example parameters above 3 times in a row, without removing any results directories, you would end up with 3 complete sets of results data in the 3 following directories:

```
C:/MyVELMA_OutputData/LittleMtn_ClearCut

C:/MyVELMA_OutputData/LittleMtn_ClearCut_0

C:/MyVELMA_OutputData/LittleMtn_ClearCut_1
```

What Results are Available after a VELMA Simulation is Run?

The VELMA simulator writes several different types of results data, in several different file formats:

Image Data

For each year of the simulations runs via the JVelma GUI, the GUI creates a PNG image that shows the current graph/chart state on the last day of the year. These files appear in the run_index directory with the following naming format:

```
out_<run_index>_<loopNumber>_<YYYY>.png
```

Where `<run_index>` is the value of the run_index parameter, `<loop_number>` is the simulation loop number and <YYYY> is the year for the particular image.

For example here is the 2nd loop, year 1979 PNG file for a simulation run of the example configuration used earlier in this document:

```
out_LittleMtn_ClearCut_1_1978.png
```

The actual chart or graph type in the image is whatever the user had selected in the JVelma GUI when the simulator completed the particular year.

Table Data

Table data are files of comma-separated values (.csv) data. They are the primary results output of a VELMA simulator run. The VELMA simulator produces several .csv files; some document what the configuration of the simulation run was, while others report the actual simulation results.

File name	**Contents Type**	**Description**
DemState.csv	Simulation Configuration	Information about the DEM and watershed delineation.
CoverState.csv	Simulation Configuration	Information about the specified cover types.
SoilState.csv	Simulation Configuration	Information about the specified soil parameterizations.
SimulationConfiguration.csv	Simulation Configuration	The full set of parameters used to configure the simulation.
DailyResults.csv	Simulation Results	The primary, per-day results output for the simulation run.
Combined_DailyResults.csv	Simulation Results	A convenience file: contains the combined contents of the DemState, SoilState, CoverState and DailyResults .csv files.
Calibration_DailyResults.csv	Simulation Results	Additional, per-day results data that does not appear in the DailyResults .csv file.
AnnualResults.csv	Simulation Results	Derived from the contents of DailyResults; per year min, max, sum and range values for each column of DailyResults file.

Text Data

Text data files contain simple text summaries of either results data or simulator configuration information for a particular simulation run.

Of these, the most important is the `NashSutcliffeCoefficients.txt,` which contains a Nash-Sutcliffe coefficient value computed for the simulated runoff values. Note that this file only contains useful data if the VELMA simulator was provided with observed runoff data as part of its simulation configuration.

Appendix 5:
Generating daily temperature and precipitation grids for running VELMA

(Link to All Parameters Table of Contents)

Introduction

The VELMA model runs at a daily time step on a spatial grid covering the study area of interest. The grid is scalable but is often desired to be at a fairly fine scale of resolution, e.g., 30-m cells. The model requires daily values of precipitation and mean temperature for each cell. Historic daily temperature and precipitation records can be obtained from weather stations in the National Weather Service or other networks. A simplistic approach would be to assign temperature and precipitation values to each cell equal to those at the nearest weather station (Thiessen polygons). However, this would lead to a very "blocky" spatial data set with sharp discontinuities along lines equidistant from two weather stations. Smoother surfaces could be obtained by using algorithms such as inverse distance weighting to calculate values for each cell depending on their distance from multiple weather stations. Both of these approaches ignore spatial variation in weather variables that are related to physiographic characteristics other than distance from a data source. Elevation is often a strong driver of both temperature gradients due to adiabatic cooling and precipitation gradients due to orographic effects produced as the rising air cools, lowering its moisture-holding capacity and causing precipitation of the excess.

The PRISM model was developed to spatially interpolate climate data from observational networks in a way that accounts for elevation and other physiographic variables of importance. Originally developed for precipitation (Daly et al. 1994), it has since been expanded to include temperature as well (Daly et al. 2008). The model has become the most widely used source of spatial climate data sets in the U.S. and has been extensively used worldwide (Daly and Bryant 2013).

Daily time series of PRISM interpolated temperature and precipitation data are supposed to be available for grids in the conterminous U.S. at a 4-km resolution (PRISM Climate Group 2013; data set AN81d), but as of 3/6/14 this FTP link through the PRISM Climate Group web site (http://prism.oregonstate.edu/) was not functional. Even if available, a 4-km grid scale is very coarse compared to the finer scale resolution (e.g., 30-m) often desired for VELMA runs. Daily PRISM time series at higher resolution more suitable for supporting VELMA runs would need to be purchased, which would add significant cost to the application of VELMA to a study area. Depending on the number of cells in the study area, reading in daily grids of weather data would also significantly slow down the processing time for VELMA runs as well as greatly increasing the memory requirements.

This manual describes an alternative approach for providing VELMA with daily temperature and precipitation data for each cell in a study grid. The desire is to capture the essence of PRISM's consideration of physiographic factors, but without (a) the expense of purchasing custom PRISM high-resolution daily time series products, or (b) the processing time and memory requirements that use of these products would entail. The method involves developing statistical regression relationships between weather variables from PRISM monthly 30-year normal maps and physiographic factors (elevation, etc.). Those relationships are then applied on a cell-by-cell basis, adjusting for the departure of daily weather values from monthly normals at a reference weather station in the study area. The steps involved in developing and applying these regressions are described below, using a VELMA application at the H.J. Andrews Experimental Forest in Oregon (hereafter referred to a HJA) as a specific example. Comments are also given about how this may vary for other application study sites.

Procedure

1. Choose candidate variables and assemble data

PRISM assumes that elevation is the most important factor in the distribution of temperature and precipitation, but it also includes effects of a variety of other physiographic variables; these include topographic facet, proximity to coastlines, cold air drainage, and measures of topographic complexity (Daly and Bryant 2013). The first step in the regression approach is to choose proxy variables, for which data are available on a cell-by-cell basis, that are relevant to these physiographic relationships in PRISM. For the HJA we chose elevation, flow accumulation, and heat load index. Flow accumulation (also known as contributing area) is the area of cells that are upslope of the target cell, into which water would flow. This is commonly used to model hydrologic routing, but is also relevant to cold air drainage modeled in PRISM. Heat load index is a variable that quantifies the heat load from annual incident solar radiation as a function of slope and aspect (McCune and Keon 2002), which relates to the topographic facets in PRISM. In other study areas, other appropriate variables might include distance from coastline if there is a significant coastal to inland gradient, or latitude if there is a sizable north to south gradient, for example.

Once these physiographic variables have been selected, a data set must be assembled that includes values for each of the cells in the study area. In the HJA study, a 30-m grid was used. A 30-m Digital Elevation Model (DEM) provided the data necessary for elevation and calculation of flow accumulation and heat index in a Geographic Information System (GIS) for each 30-m grid cell. In addition to physiographic data, PRISM monthly "normals" for temperature and precipitation must be assembled for each cell. On the PRISM web site (http://prism.oregonstate.edu) data sets are freely available for the conterminous United States that include 30-year monthly temperature and precipitation normals for 1981-2010 at resolutions of 800 m and 4 km (PRISM Climate Group 2013). As an alternative, for the HJA study we used 50-m resolution PRISM data sets from the HJA web site (http://andrewsforest.oregonstate.edu/) that included monthly normals for total precipitation (mm) and daily mean temperature (C; average of daily minimum and daily maximum). The PRISM temperature data were based on 30 years of weather station observation (1971-2000), but 50-m PRISM precipitation data were only available at the time for 1981-1989. The monthly normals were rescaled to the 30-m grid, and this data set was merged with the 30-m physiographic data set. In summary, for each of 73,331 30-m cells the HJA data set included:

Jan – Dec monthly normal precipitation (mm/month)
Jan – Dec monthly normal mean temperature (C; average of daily min and max)
elevation (m)
flow accumulation (unitless)
heat load index (unitless)

Example maps of these HJA data are shown in Fig. 1.

2. Exploratory data analysis

Once the data set has been read in, the distributions of values for the variables should be examined. This serves two purposes. First, it allows checking to see if all values are in an appropriate range and that there have not been any gross data transcription errors. Second, it allows consideration of whether transformations of the original variables might be another choice. In the case of the HJA data, for example, the flow accumulation variable was found to be very highly right skewed. A $\log_{10}$ transformation was used to create an alternative variable for testing in the regression model fitting.

Correlation coefficients can be calculated between the weather variables (temperature and precipitation) and the physiographic variables. As expected, and as assumed by PRISM, in the HJA data set this analysis revealed that the strongest correlation was with elevation. Plots of temperature and precipitation versus elevation should be made for each month to see the form of this relationship. For HJA precipitation, this revealed fairly linear patterns for some months (Fig. 2a), but concave-upwards patterns for other months (Fig. 2b) indicating a possible quadratic relationship with elevation. Consequently, an elevation^2 variable was created for testing alternative regression models. For HJA temperature, there was a linear increase with elevation up to a point, and then a linear decrease with elevation (Fig. 3). This indicated that a spline regression model be used, fitting a piecewise linear model with an inflection point.

3. Regression and model selection

Once the physiographic variables have been selected (step 1), additional polynomial terms have been created if indicated (step 2), and alternative transformations have been created if indicated (step 2), then alternative regression models can be fit for each month and the most appropriate model form chosen. Akaike's information criterion (AIC; Akaike 1974) is a widely used statistic to choose among competing regression models. It represents a compromise between goodness of fit (which increases with more parameters) and parsimony (which decreases with more parameters). In practice, ΔAIC values are computed as the model AIC value minus the minimum AIC value for all models tested. While it is possible that different models might have the lowest AIC value for different months, models with $\Delta AIC \leq 2$ are considered roughly equivalent. For consistency, it is desirable to choose a single model form that applies for all months and has $\Delta AIC \leq 2$.

For HJA precipitation, the dependent variable was the mean precipitation for each month, and the independent variables were elevation (*elev*), elevation^2 (*elevsq*), heat load index (*hli*), and $\log_{10}$ of flow accumulation (*logfacc*). Alternative regression models were fit that included *elev* and various subsets of the other variables. The full model with all the variables was selected as the best model; it had ΔAIC = 0 for 8 of the 12 months and ≤ 2 for the other 4 months (Fig. 4). The regression parameters are shown in Table 1. The regression residuals fell inside an envelope of ±20% (or better) of the monthly estimates over all months (Fig. 5). Sample SAS code for generating and comparing the regression models is given in Appendix 5.1, below.

For HJA temperature, the dependent variable was the mean temperature for each month, and the independent variables were elevation (*elev*), heat load index (*hli*), and either flow accumulation (*facc*) or $\log_{10}$ of flow accumulation (*logfacc*). Alternative regression models were fit that included *elev* and various subsets of the other variables. To do the spline fit, a non-linear regression procedure was used (PROC NLIN in SAS) that seeks an iterative solution to the regression parameters for each of the two linear pieces, as well as the inflection point where they meet. The full model with *elev, hli,* and *facc* was selected as the best model; it had ΔAIC = 0 for 9 of the 12 months, ≤ 2 for 2 months, and 2.6 for the remaining month (Fig. 6). Models incorporating *logfacc* rather than *facc* failed to converge for several of the months, so *facc* was used instead. The regression parameters are shown in Table 2. The regression residuals roughly fell inside an envelope of ±1 °C over all months (Fig. 7). Sample SAS code for generating and comparing the regression models is given in Appendix 5.2, below.

4. Implementation to get daily temperature and precipitation

The regression models give estimates of the long-term normal monthly mean temperature and precipitation totals at each cell, based on its elevation, heat load index, and flow accumulation values. (The monthly precipitation totals must be divided by the number of days in the month to translate this to a daily basis.) However, many VELMA applications seek to do simulations based on historic daily weather time series, or even on future scenarios of daily weather time series (adding projected temperature and precipitation changes to historic weather time series, for example).

One mechanism to translate the regression temperature and precipitation monthly estimates into daily time series for each cell is to adjust the values using daily deviations from the monthly normals for a reference weather station in the study area. Essentially, this assumes that the daily deviations from climate normals are the same across the entire study area. For the HJA study, the PRIMET weather station was used as the reference (Fig. 8). (For future development, methods to use several reference weather stations might be explored, e.g., using inverse distance weighting to determine the deviations to apply to each cell in the study area.) Using the HJA example, the steps for this procedure were as follows:

- Compute expected daily PRIMET temperature (T) and precipitation (P) using the regression equations for that month, and the *elev*, *hli*, and *facc* values for the PRIMET location. (Divide the monthly P total estimates by the number of days in the month to get the daily estimate; this is not necessary for temperature.)
- Using the observed daily PRIMET T and P values, compute T and P deviations (observed – expected).
- Compute expected T and P values for all other cells using their *elev*, *hli*, and *facc* values.
- Add the PRIMET T and P deviations to the expected values for all cells.
- Using P as an example:

Subscript 0 denotes PRIMET, x denotes other location, m denotes month, ndays_m denotes number of days in month m, exp denotes expected value from regression
P_0 = observed P at PRIMET for a day in month m
$P_{0,exp} = (a_m + b_m{*}\mathit{elev}_0 + c_m{*}\mathit{elevsq}_0 + d_m{*}\mathit{hli}_0 + e_m{*}\mathit{logfacc}_0) / \text{ndays}_m$
$\Delta = P_0 – P_{0,exp}$
$P_x' = P_{x,exp} + \Delta$
$= (a_m + b_m{*}\mathit{elev}_x + c_m{*}\mathit{elevsq}_x + d_m{*}\mathit{hli}_x + e_m{*}\mathit{logfacc}_x) / \text{ndays}_m + \Delta$

A test was done to make sure the adjustment was working correctly in the VELMA code by focusing on the 30-m cell in which PRIMET was located. For each day in 2007, the expected daily precipitation and temperature were calculated from the regression equations and then adjusted following the procedure above. As expected, the adjusted simulated results for PRIMET were identical to the observed precipitation and temperature data, indicating that the code correctly implemented these procedures (Fig. 9).

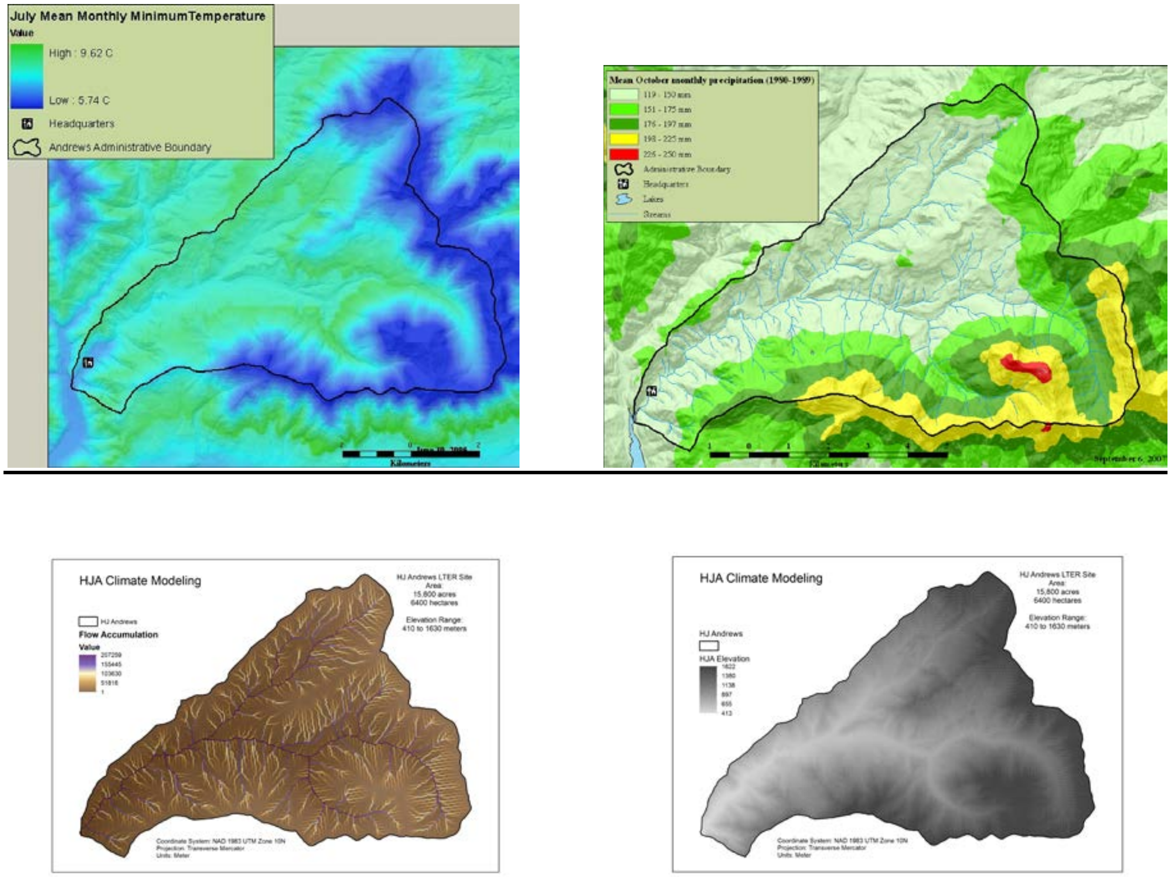


Fig. 1 – Example maps of H.J. Andrew Experimental Forest PRISM climate and physiographic variables.

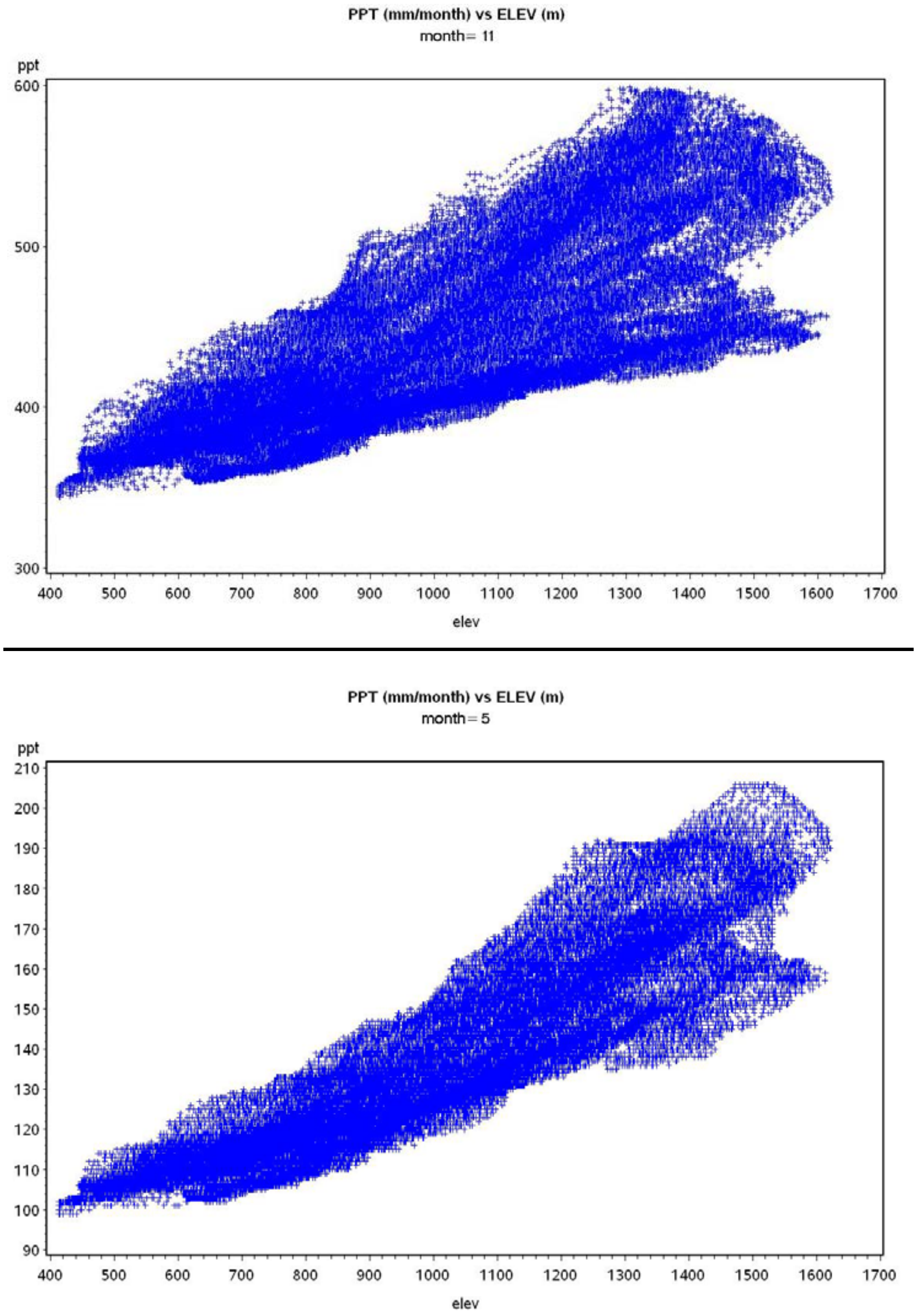


Fig. 2 – Plots of monthly mean precipitation (mm) versus elevation for H.J. Andrews Experimental Forest. (a) November data, showing more or less linear increase. (b) May data, showing a quadratic increase.

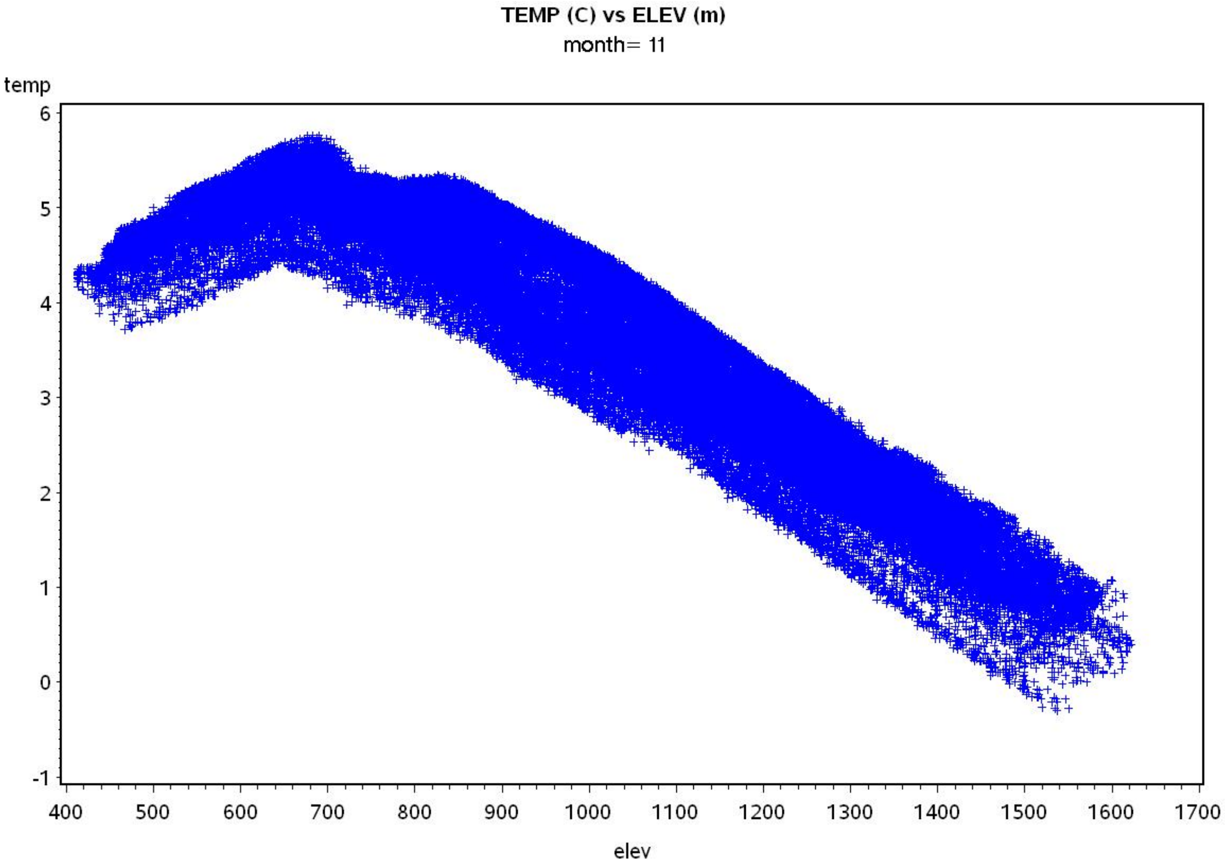


Fig. 3 - Plots of monthly mean temperature (C) for November versus elevation for H.J. Andrews Experimental Forest. Note the piecewise linear relationship, with an inflection point at around 700 m elevation.

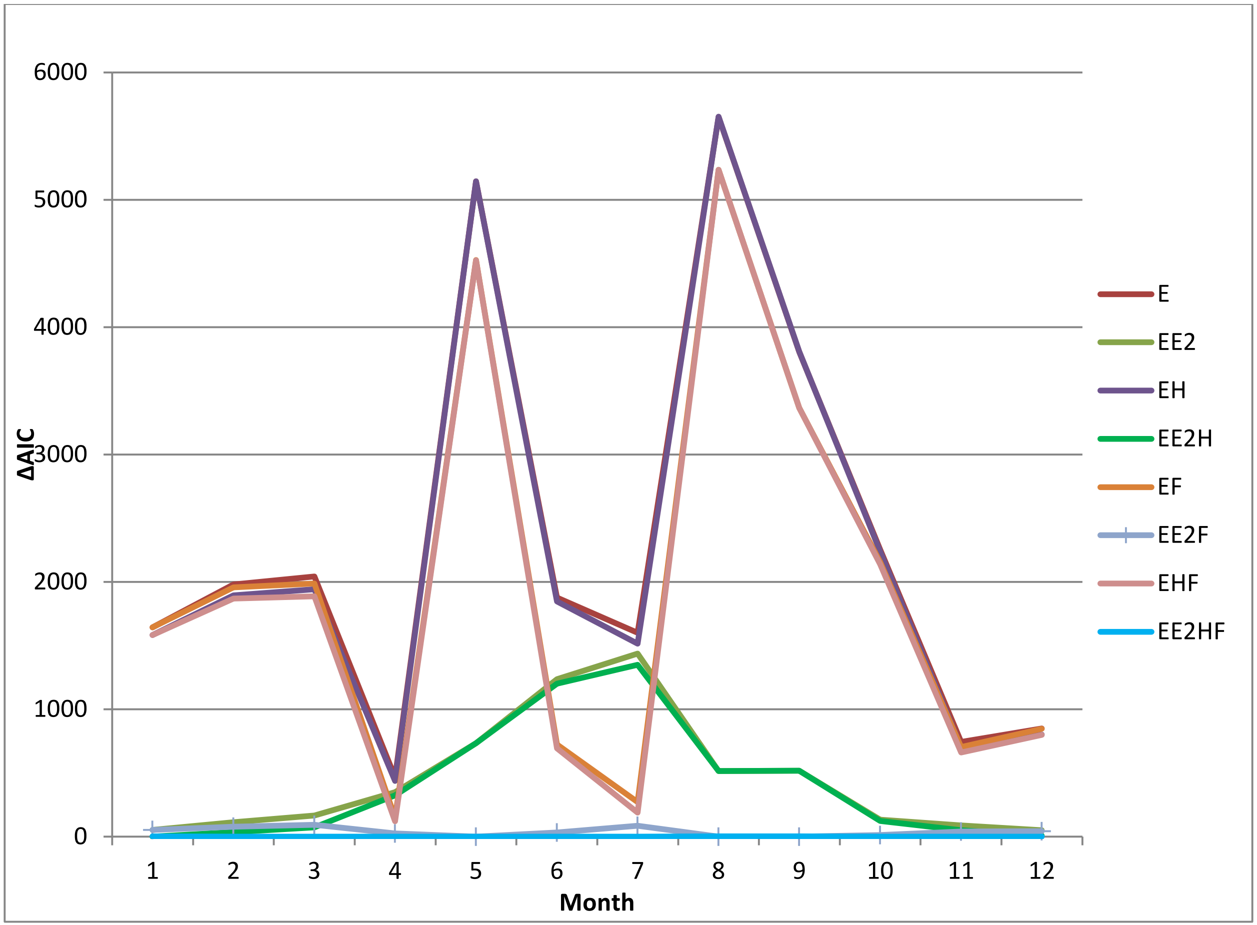


Fig. 4 – Plot of ΔAIC over all months for alternative precipitation regression models at H.J. Andrews Experimental Forest. E = elevation, E2 = elevation^2, H = heat load index, F = log$_{10}$ of flow accumulation.

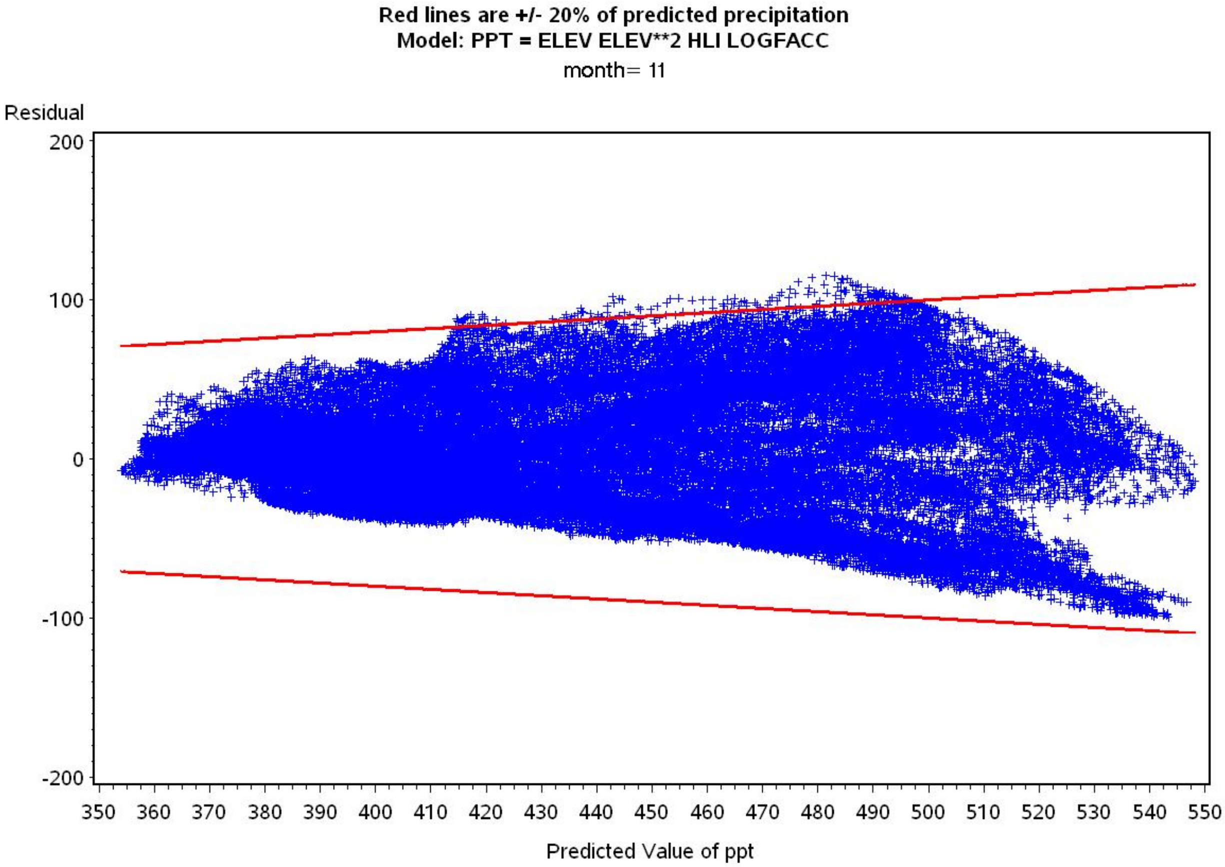


Fig. 5 – Graph of November precipitation regression residuals versus predicted precipitation at H.J. Andrews Experimental Forest. Red bounding lines represent ±20% of the monthly precipitation estimate.

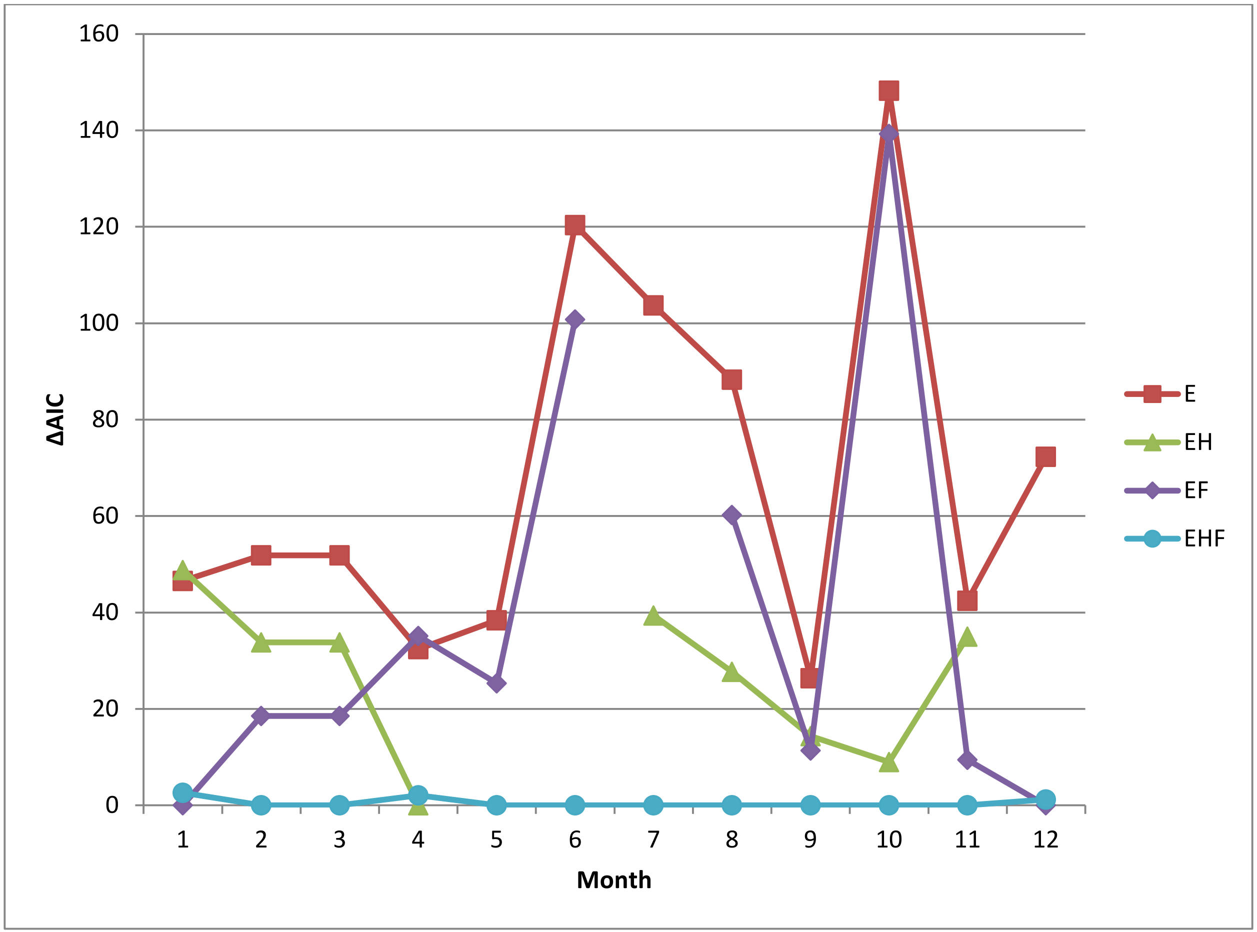


Fig. 6 – Plot of ΔAIC over all months for alternative temperature regression models at H.J. Andrews Experimental Forest. E = elevation, H = heat load index, F = flow accumulation.

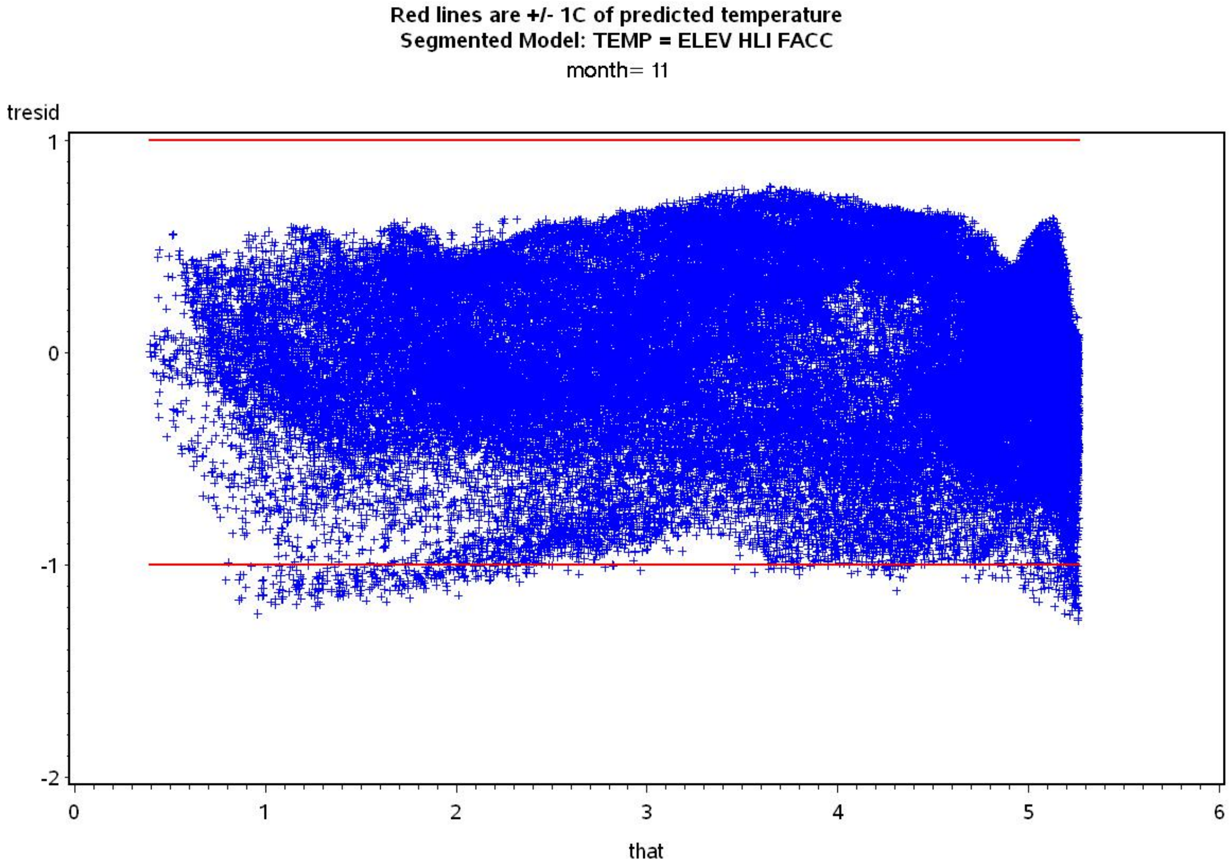


Fig. 7 – Graph of November mean temperature regression residuals versus predicted precipitation at H.J. Andrews Experimental Forest. Red bounding lines represent +1 °C.

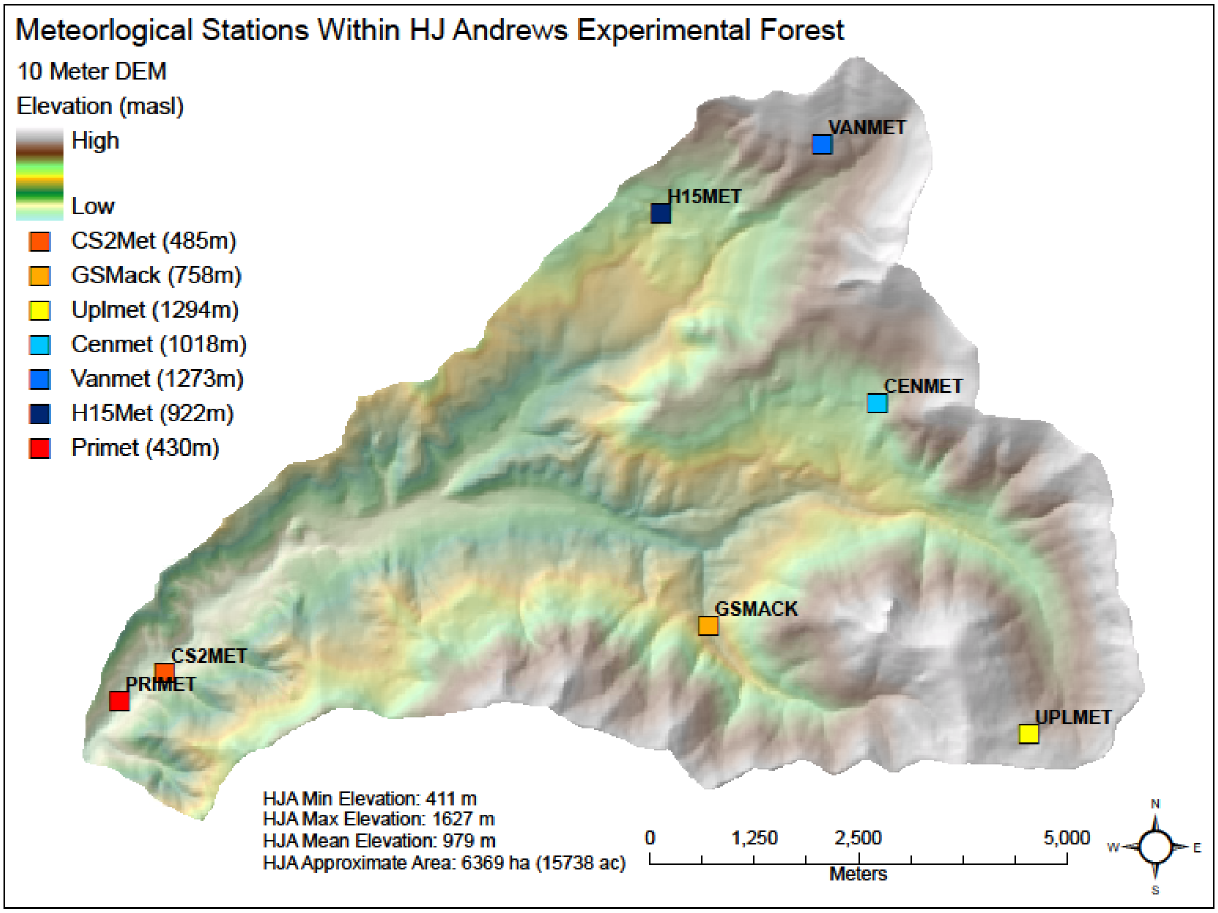


Fig. 8 – Map of weather station locations at the H.J. Andrews Experimental Forest.

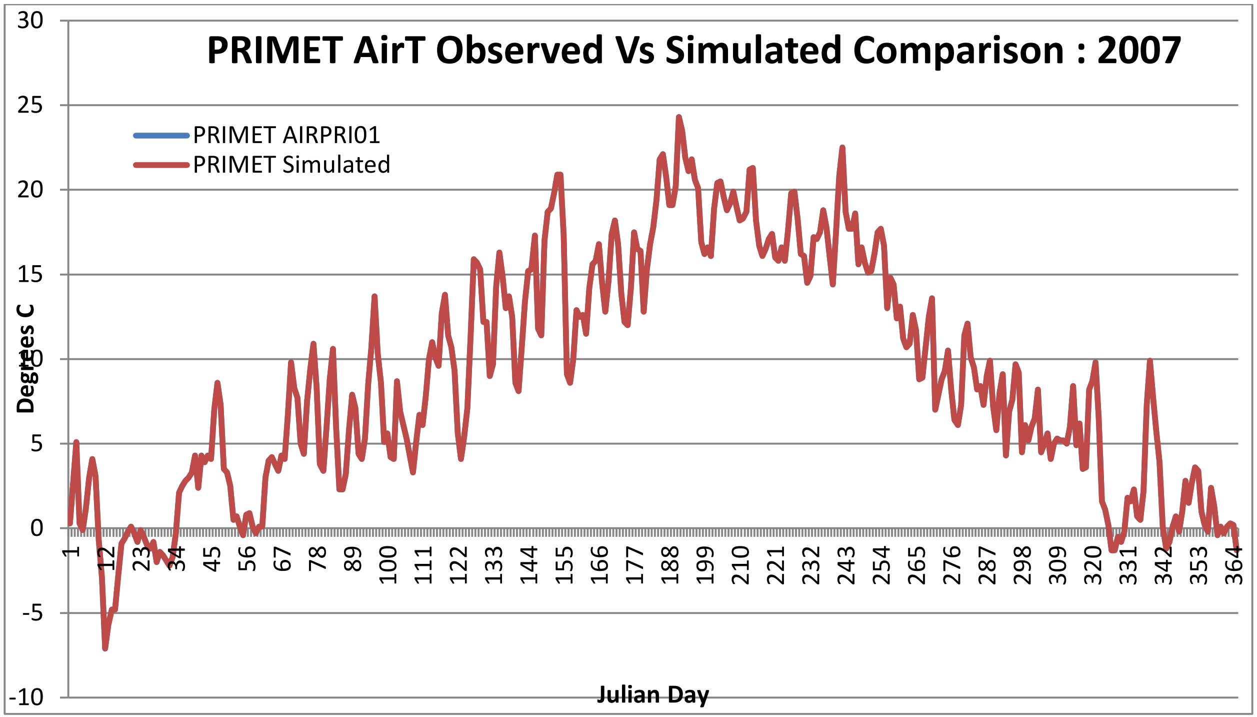


Fig. 9 – Comparison of observed H.J. Andrews Experimental Forest PRIMET weather station daily mean temperature for 2007 and VELMA simulated data for the same location. The VELMA simulated data used the temperature regression equation to estimate monthly normal temperature for the cell containing PRIMET, and then made adjustments based on the observed PRIMET daily mean temperature. This is a check to make sure that the implementation of the temperature regression and adjustment is working correctly, as indicated by the coincidence of the blue and red lines (red is on top).

Month	a	b	c	d	e
1	259.751	0.011833	5.24E-05	-3.94717	-0.09685
2	241.5404	0.006131	6.1E-05	-5.15183	0.892817
3	210.8037	0.037236	4.62E-05	-4.20577	0.951439
4	96.40946	0.080534	7.44E-06	-1.44569	1.250961
5	88.34944	0.014202	3.06E-05	-0.20094	1.259266
6	53.51838	0.048195	9.53E-06	0.873204	1.297879
7	13.08702	0.023576	2.49E-06	0.686924	0.685529
8	21.89397	7.78E-05	6.23E-06	-0.00893	0.199088
9	70.03036	0.007598	2.39E-05	-0.00868	0.959202
10	124.4332	-0.00302	3.59E-05	-1.16562	0.890761
11	316.075	0.072349	4.32E-05	-4.40543	1.247633
12	290.7049	0.034712	4.62E-05	-4.46288	0.495517

Table 1. Monthly mean precipitation regression model coefficients based on PRISM data for 30-m cells in the H. J. Andrews Experimental Forest. The models are of the form:
P = a + b**elev* + c**elevsq* + d**hli* + e**logfacc*
P in mm/month, *elev* in m, *elevsq* in m^2, *hli* is unitless, *logfacc* is unitless

Month	b	c	d	e	g	h	i	j	x
1	0.007047	0.018444	2.75E-06	-1.12696	-0.00434	0.012177	-1.6E-05	6.606466	679.15
2	0.006148	0.065628	2.76E-06	1.024448	-0.00545	0.037565	-1.5E-05	8.958362	684.28
3	0.001492	0.041412	1.47E-06	5.402304	-0.00656	0.039723	-5.7E-06	11.03547	699.67
4	-0.00239	0.053357	2.53E-07	10.24696	-0.00697	0.043574	6.13E-06	13.99039	818.03
5	0.004219	0.059169	1.86E-06	8.916032	-0.00707	0.056038	-1.2E-05	16.80074	698.66
6	-0.00222	0.046785	-1.3E-06	16.98087	-0.00684	0.078366	1.33E-05	20.81386	827.75
7	-0.00123	0.004955	-2.5E-06	19.38073	-0.00633	0.073345	-1.9E-06	23.54351	814.57
8	-0.00023	0.00407	-2.6E-06	18.81907	-0.00585	0.007047	9.03E-06	23.53079	836.26
9	0.001476	-0.0342	-2.3E-06	14.80655	-0.00612	0.047614	7.6E-06	20.83291	791.89
10	-0.00023	0.018716	-1.2E-06	11.70249	-0.00641	0.112236	-8.6E-06	16.77907	819.27
11	0.002002	0.014029	-2.6E-07	3.771477	-0.00556	0.027117	-1.4E-05	9.412401	745.95
12	0.00697	0.047469	2.59E-06	-1.28579	-0.00399	0.007256	-1.7E-05	6.096779	674.34

Table 2. Monthly mean temperature regression model coefficients based on PRISM data for 30-m cells in the H. J. Andrews Experimental Forest. The models are of the form:
For $elev < x$: $T = b*elev + c*hli + d*facc + e$
For $elev \geq x$: $T = g*elev + h*hli + i*facc + j$
T is in °C, *elev* in m, *hli* is unitless, *facc* is unitless

References

Akaike H (1974) A new look at the statistical model identification. IEEE Transactions on Automatic Control 19:716-723

Daly C, Bryant K (2013) The PRISM Climate and Weather System - An Introduction. In. PRISM Climate Group, Oregon State University, Corvallis, Oregon, http://prism.oregonstate.edu/documents/PRISM_history_jun2013.pdf

Daly C, Halbleib M, Smith JI et al. (2008) Physiographically sensitive mapping of climatological temperature and precipitation across the conterminous United States. International Journal of Climatology 28:2031-2064

Daly C, Neilson RP, Phillips DL (1994) A statistical-topographic model for mapping climatological precipitation over mountainous terrain. Journal of Applied Meteorology 33:140-158

McCune B, Keon D (2002) Equations for potential annual direct incident radiation and heat load. Journal of Vegetation Science 13:603-606

PRISM Climate Group (2013) Descriptions of PRISM Spatial Climate Datasets for the Conterminous United States. In. PRISM Climate Group, Oregon State University, Corvallis, Oregon, http://prism.oregonstate.edu/documents/PRISM_history_jun2013.pdf

Appendix 5.1 – Sample SAS code for generating and comparing monthly total precipitation regression models at H.J. Andrews Experimental Forest.

```
proc reg data=hja.hja outest=parmout;
   by month;
   model ppt = elev elevsq hli logfacc / selection=cp aic;
   output out=pptout p=phat r=presid;
```

Notes:

- The input data are read in from the file ‘hja’ in the SAS library (folder) ‘hja’.
- Separate models are fit for each month.
- With the ‘selection’ option, models are fit using all the various sets of independent variables listed.
- By using the ‘aic’ option, Akaike Information Criterion (AIC) values are given in the output for each model. This allows calculation of ΔAIC compared to the model with the lowest AIC value (Fig. 4).
- The parameter estimates for each model are written to the file ‘parmout’ (Table 1).
- Predicted estimates (‘phat’) and residuals (‘presid’) are written to the file ‘pptout’ for plotting residuals vs. predicted values (Fig. 5).

Appendix 5.2 – Sample SAS code for generating and comparing monthly temperature regression models at H.J. Andrews Experimental Forest.

```
* TEMP MODEL WITH ELEV, HLI, FACC.  BEST MODEL - MIN AIC FOR MOST MONTHS,
MINIMAL DELTA AIC FOR OTHER MONTHS;

* Segmented model, linear with elev above and below a join point x;

proc nlin data=hja.hja outest=parmout;
   by month;
   parms b=0.01 c=0 d=0 e=0 g=-0.005 h=0 i=0 j=10;
   file print;
   x = ((h-c)*hli + (i-d)*facc + (j-e))/(b-g);
   if elev < x then do;
      model temp = b*elev + c*hli + d*facc + e;
        der.b = elev;
        der.c = hli;
        der.d = facc;
        der.e = 1;
        der.g = 0;
        der.h = 0;
        der.i = 0;
        der.j = 0;
   end;
   else do;
      model temp = g*elev + h*hli + i*facc + j;
        der.b = 0;
        der.c = 0;
        der.d = 0;
        der.e = 0;
        der.g = elev;
        der.h = hli;
        der.i = facc;
        der.j = 1;
   end;
   if _obs_=1 then do; * PRINT X ESTIMATE TO TRACK CONVERGENCE;
      put x =;
   end;
   output out=tempout predicted=that residual=tresid;
```

Notes:

- The input data are read in from the file ‘hja’ in the SAS library (folder) ‘hja’.
- Separate models are fit for each month.
- This is a spline model, fitting linear models above and below an inflection point at elevation *x*, which is also estimated. This requires use of an iterative non-linear procedure (PROC NLIN). Unlike PROC REG in Appendix 5.1, this does not have the capability to fit multiple alternative models at once and report the AIC values for each. Instead, each alternative model must be specified. The above code specifies the full model with *elev*, *hli*, and *facc*. AIC values are not automatically printed and must be calculated from the sum of squares error (SSE), number of parameters (p; 4 in this case), and number of sample points (n; 73,331 in this case), according to the formula: AIC = n * ln(SSE / n) + 2*p. ΔAIC values are then calculated by subtracting the AIC for the model with the lowest AIC value (Fig. 6).
- The ‘parms’ statement specifies starting values for the parameters to begin the iterative search for best fit.

- *x* represents the elevation at which the two piecewise linear models meet. To make the overall model continuous requires that the estimated temperature must be the same for both piecewise linear models at that point. The expression for *x* is derived by setting the two models equal to each other and solving for *elev* at which they are equal:

  b**elev* + c**hli* + d**facc* + e = g**elev* + h**hli* + i**facc* + j

  b**elev* - g**elev* = h**hli* - c**hli* + i**facc* - d**facc* + j – e

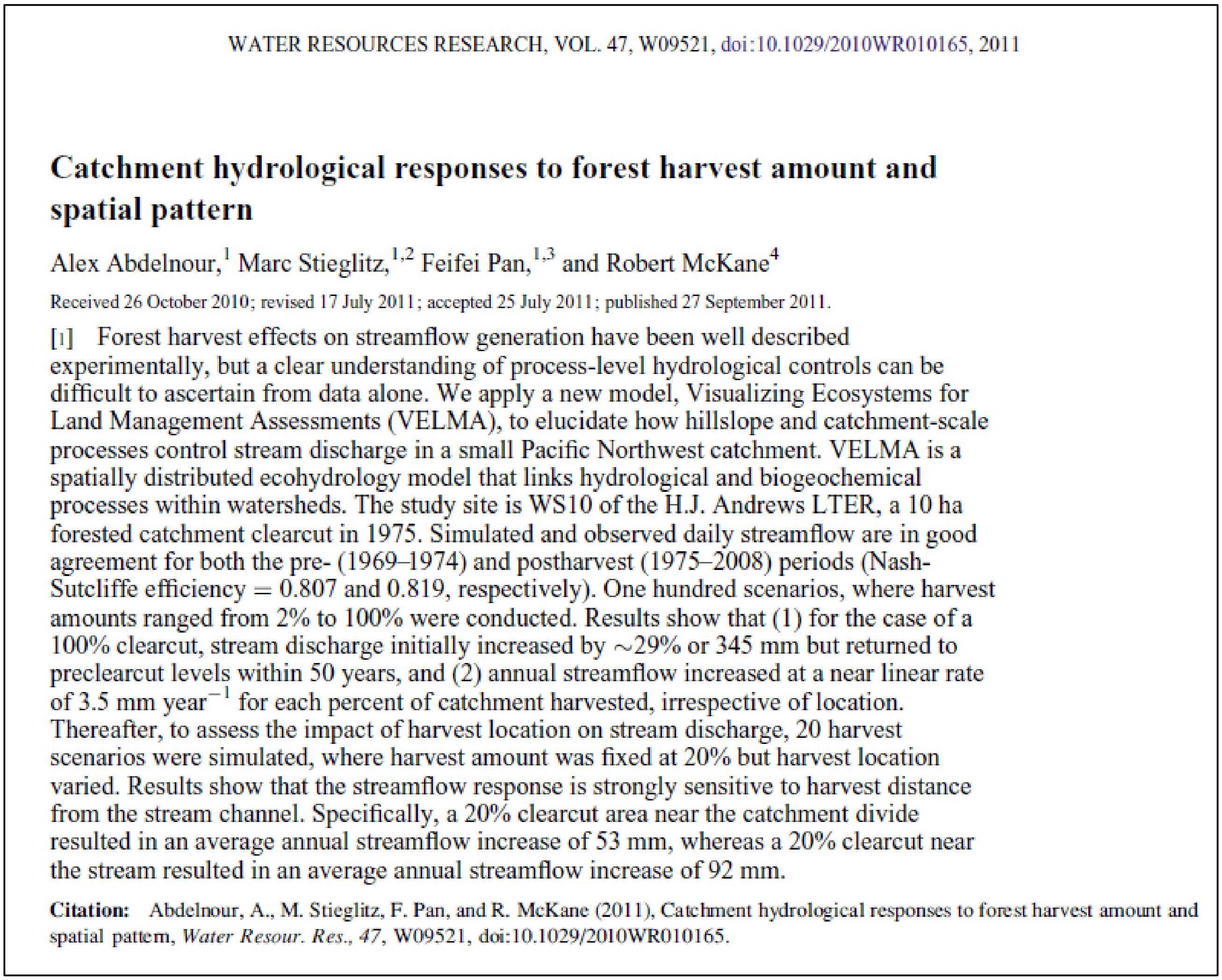
  x = *elev* = ((h – c)**hli* + (I - d)**facc* + (j – e))/(b – g)

- The ‘der’ statements specify the first derivatives of each of the linear models with respect to each of the parameters.
- The parameter estimates for each model are written to the file ‘parmout’ (Table 2).
- Predicted estimates (‘that’) and residuals (‘tresid’) are written to the file ‘tempout’ for plotting residuals vs. predicted values (Fig. 7).

Appendix 6.1:

VELMA version 1.0, Part 1 of 2 HYDROLOGICAL Model Description

(Link to All Parameters Table of Contents)



WATER RESOURCES RESEARCH, VOL. 47, W09521, doi:10.1029/2010WR010165, 2011

Catchment hydrological responses to forest harvest amount and spatial pattern

Alex Abdelnour,[1] Marc Stieglitz,[1,2] Feifei Pan,[1,3] and Robert McKane[4]

Received 26 October 2010; revised 17 July 2011; accepted 25 July 2011; published 27 September 2011.

[1] Forest harvest effects on streamflow generation have been well described experimentally, but a clear understanding of process-level hydrological controls can be difficult to ascertain from data alone. We apply a new model, Visualizing Ecosystems for Land Management Assessments (VELMA), to elucidate how hillslope and catchment-scale processes control stream discharge in a small Pacific Northwest catchment. VELMA is a spatially distributed ecohydrology model that links hydrological and biogeochemical processes within watersheds. The study site is WS10 of the H.J. Andrews LTER, a 10 ha forested catchment clearcut in 1975. Simulated and observed daily streamflow are in good agreement for both the pre- (1969–1974) and postharvest (1975–2008) periods (Nash-Sutcliffe efficiency = 0.807 and 0.819, respectively). One hundred scenarios, where harvest amounts ranged from 2% to 100% were conducted. Results show that (1) for the case of a 100% clearcut, stream discharge initially increased by ~29% or 345 mm but returned to preclearcut levels within 50 years, and (2) annual streamflow increased at a near linear rate of 3.5 mm year^{-1} for each percent of catchment harvested, irrespective of location. Thereafter, to assess the impact of harvest location on stream discharge, 20 harvest scenarios were simulated, where harvest amount was fixed at 20% but harvest location varied. Results show that the streamflow response is strongly sensitive to harvest distance from the stream channel. Specifically, a 20% clearcut area near the catchment divide resulted in an average annual streamflow increase of 53 mm, whereas a 20% clearcut near the stream resulted in an average annual streamflow increase of 92 mm.

Citation: Abdelnour, A., M. Stieglitz, F. Pan, and R. McKane (2011), Catchment hydrological responses to forest harvest amount and spatial pattern, *Water Resour. Res.*, *47*, W09521, doi:10.1029/2010WR010165.

of the spatial distribution of soil texture and soil depth on streamflow dynamics would certainly provide insights into catchment dynamics, it is beyond the scope of this paper.

[43] Finally, it is important to ask whether or not an explicit treatment of the preceding issues would improve model performance and the understanding of process-level controls on streamflow and other ecohydrological processes. For some applications, the explicit treatment of these processes may be needed. However, it must be recognized that such added processes come at the cost of computational efficiency, model complexity, and applicability to larger spatial and temporal scales. The current version of VELMA is an initial attempt at a parsimonious solution to this dilemma.

7. Conclusion

[44] Despite the limitations discussed above, the model presented here provides a relatively simple, spatially distributed framework for assessing the effects of changes in climate, land use, and land cover on ecohydrological processes within watersheds. The WS10 simulations suggest that the model can predict, with reasonable accuracy, the effects of forest harvest on daily, seasonal, and annual changes in streamflow. The simulations describing the effects of harvest amount and spatial pattern provide process-level insights into important hydrological responses to harvest—details that would be difficult or impossible to capture through experimentation or observation alone. Moreover, the model provides an integrated ecohydrological framework for evaluating how alternative climate and forest management scenarios may interact to affect the functioning and health of forest and stream ecosystems. Finally, the simplicity of the model makes it potentially useful for applications across a range of spatial and temporal scales relevant to land managers and policy makers.

Appendix A: Model Description

[45] We include two appendices. Appendix A describes the hydrology model, which includes the equations typically applied across watersheds and ecosystems. Appendix B describes the evapotranspiration recovery function used to mimic postharvest transpiration dynamics.

[46] VELMA is a spatially distributed ecohydrology model that accounts for hydrologic and biogeochemical processes within watersheds. The model simulates daily to century-scale changes in soil water storage, surface and subsurface runoff, vertical drainage, carbon (C) and nitrogen (N) cycling in plants and soils, as well as transport of nutrients from the terrestrial landscape to the streams. VELMA consists of multilayered soil column models that communicate with each other through the downslope lateral transport of water (Figure A1). Each soil column model consists of three coupled submodels: a hydrological model, a soil temperature model, and a plant-soil model. What we describe below is the hydrology component of the model. First, we describe the soil column model and then place this soil column within a catchment framework.

A1. The Hydrological Model

A1.1. Soil Column Framework

[47] We employ a multilayer soil column as the fundamental hydrologic unit. The soil column consists of n soil layers, a standing water layer, and a snow layer (Figure A2). Soil water balance is solved for each model layer (equations (A1)–(A6)). Soil water storage in layer i (s_i), surface standing water (s_{STW}), and snow water equivalent (s_{SWE}), are tracked and updated at each time step. For a four-layer soil model, such as the one used in this work:

$$\frac{ds_{\mathrm{SWE}}}{dt} = P_s - m, \tag{A1}$$

$$\frac{ds_{\mathrm{STW}}}{dt} = P_r - I + m - Q_{s\_\mathrm{out}} + Q_{s\_\mathrm{in}}, \tag{A2}$$

$$\frac{ds_1}{dt} = I - D_1 - ET_1 - Q_{1\_\mathrm{out}} + Q_{1\_\mathrm{in}}, \tag{A3}$$

$$\frac{ds_2}{dt} = D_1 - D_2 - ET_2 - Q_{2\_\mathrm{out}} + Q_{2\_\mathrm{in}}, \tag{A4}$$

$$\frac{ds_3}{dt} = D_2 - D_3 - ET_3 - Q_{3\_\mathrm{out}} + Q_{3\_\mathrm{in}}, \tag{A5}$$

$$\frac{ds_4}{dt} = D_3 - Q_{4\_\mathrm{out}} + Q_{4\_\mathrm{in}}, \tag{A6}$$

where P_r (mm day^{-1}) and P_s (mm day^{-1}) are rain and snow, respectively, ET_i (mm day^{-1}) is the water extracted from soil layer i due to evapotranspiration, s_i (mm) is the soil water storage in layer i; s_{SWE} (mm) is the snow water equivalent due to the accumulation of snow, s_{STW} (mm) is

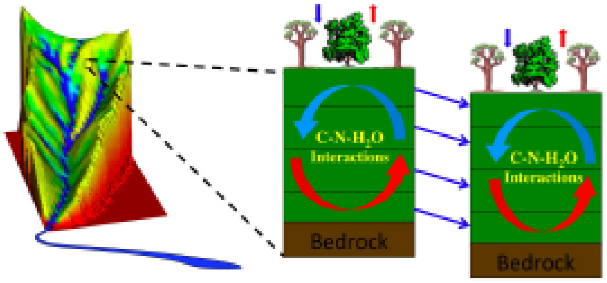


Figure A1. Conceptual catchment modeling framework using multilayered soil columns.

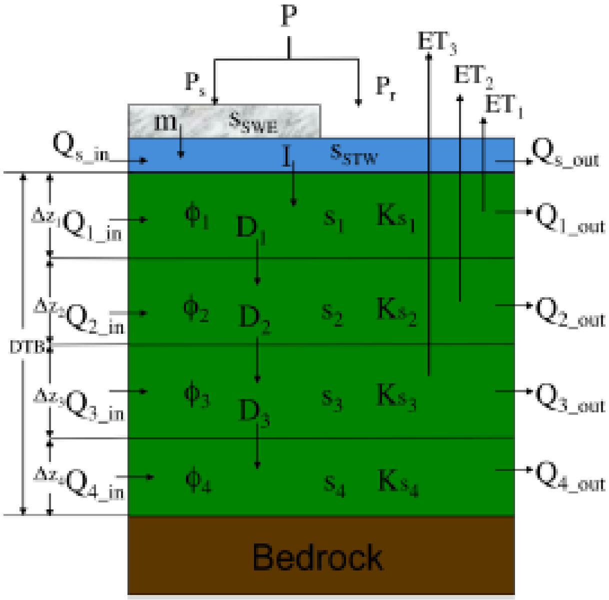


Figure A2. The soil column framework consists of four-layer soil column, a standing water layer, and a snow layer. DTB is the soil column depth to bedrock. z_l, Ks_l, ϕ_l, and s_l are the thickness, the saturated hydraulic conductivity, the soil porosity, and the soil water storage of layer i, respectively.

the standing water amount, m (mm day^{-1}) is the snowmelt that enters the standing water layer, I (mm day^{-1}) is the soil infiltration rate, D_l (mm day^{-1}) is the vertical drainage from layer l to layer $l+1$ within a given soil column, Q_{l_in} (mm day^{-1}) and Q_{l_out} (mm day^{-1}) are the lateral subsurface flow into and out of layer l; Q_{s_in} (mm day^{-1}) is the surface water flow from the s_{STW} pool of an upslope soil column, and Q_{s_out} (mm day^{-1}) is the surface water flow to the s_{STW} pool of a downslope soil column or into the stream.

A1.2. Vertical Water Drainage

[48] The vertical water drainage (D_l) is modeled using a logistic function that is intended to capture the breakthrough characteristics of soil water movement. Specifically, we employ a logistic function $f\left(s_l/s_l^{\max}\right)$ that permits for fast "switching" from low to high flow as layer water storage approaches field capacity:

$$D_l = Ks_l^{\text{vertical}} \times f\left(s_l/s_l^{\max}\right) \qquad l=1,2,\ldots,n, \qquad \text{(A7)}$$

where Ks_l^{vertical} is the vertical saturated hydraulic conductivity in layer l, $\left(s_l/s_l^{\max}\right)$ is the soil degree of saturation in layer l, $s_l^{\max}$ (mm) is the maximum soil water storage in layer l, and $f\left(s_l/s_l^{\max}\right)$ is the logistic function for layer l. The vertical saturated hydraulic conductivity follows a TOPMODEL framework [*Beven and Kirkby*, 1979] and decreases exponentially with depth such that

$$Ks_l^{\text{vertical}} = Ks \times e^{-f_v \times d_l} \qquad l=1,2,\ldots,n, \qquad \text{(A8)}$$

where Ks is the soil surface saturated hydraulic conductivity [*Clapp and Hornberger*, 1978; *Dingman*, 1994], f_v is the vertical decay rate of Ks with depth, and d_l is the soil depth to the center of layer l.

[49] The logistic function is modeled as

$$f\left(s_l/s_l^{\max}\right) = \frac{1+\exp\left[-\left(s_l/s_l^{\max}\right)\right]}{1+a_{1,l}\exp\left[-a_{2,l}\left(s_l/s_l^{\max}\right)\right]} - \frac{2}{\left(1+a_{1,l}\right)} \quad l=1,2,\ldots,n$$

$$\text{with}\begin{cases} a_{1,l} = \exp\left[6+270.562\times\left(Ks_l^{\text{vertical}}\right)^{-0.574}\right] \\ a_{2,l} = 6+144.749\times\left(Ks_l^{\text{vertical}}\right)^{-0.444}. \end{cases} \qquad \text{(A9)}$$

A1.3. Precipitation, Rain, Snow, and Snowmelt

[50] Below a threshold temperature (T_{th}), precipitation (P) falls as snow (P_s), otherwise as rain (P_r). Snow accumulates until air temperature (T_a) warms and reaches melting temperature (T_m). Snowmelt rate (m) follows a degree day approach [*Rango and Martinec*, 1995], and includes for the heat provided by rain on snow [*Harr*, 1981]. Snowmelt (m) enters the s_{STW} pool and from there infiltrates into the top soil layer (or continues as lateral surface flow and enters a downslope s_{STW} pool, section 2.2)

$$P = \begin{cases} P_r & T_a > T_{\text{th}} \\ P_s & T_a \le T_{\text{th}} \end{cases}$$
$$m = \begin{cases} \alpha(T_a - T_m) + \sigma \times P_r \times T_a & T_a > T_m \\ 0 & T_a \le T_m, \end{cases} \qquad \text{(A10)}$$

where α (mm°C^{-1} day^{-1}) is the degree-day factor for melt and σ is the rain on snow factor.

A1.4. Surface Soil Infiltration

[51] Based on the large uncertainties associated in ascribing soil texture, soil structure, and soil properties, we simply

assume that water stored in the s_{STW} pool is allowed to infiltrate (I) the top soil layer such that:

$$\begin{aligned} &I = (P_r + m + Q_{s\_\mathrm{in}} - Q_{s\_\mathrm{out}}) \\ &\quad \text{for } (P_r + m + Q_{s\_\mathrm{in}} - Q_{s\_\mathrm{out}}) < Ks_{i=1}^{\mathrm{vertical}} \\ &I = Ks_{i=1}^{\mathrm{vertical}} \\ &\quad \text{for } (P_r + m + Q_{s\_\mathrm{in}} - Q_{s\_\mathrm{out}}) \geq Ks_{i=1}^{\mathrm{vertical}}. \end{aligned} \tag{A11}$$

A1.5. Evapotranspiration

[52] Evapotranspiration increases exponentially with increasing soil water storage and asymptotically approaches the potential evapotranspiration (PET) rate as water storage reaches saturation [*Davies and Allen*, 1973; *Federer*, 1979, 1982; *Spittlehouse and Black*, 1981]:

$$\begin{aligned} &ET_i = W_{E,i} \times PET \times \left\{1 - \exp\left[-c_{ET} \times \left(s_i / s_i^{\max}\right)\right]\right\} \\ &\quad i = 1, 2, \ldots, n, \end{aligned} \tag{A12}$$

where $W_{E,i}$ is the soil water extraction fraction in layer i and c_{ET} is an ET shape factor to ensure that ET approaches PET near field capacity.

[53] PET is estimated using a simple temperature-based method [*Hamon*, 1963]:

$$\begin{aligned} &PET = K_{PET} \times 0.0138 \times L \times \rho_{\mathrm{vsat}}(T_a) \\ &\text{with} \begin{cases} \rho_{\mathrm{vsat}}(T_a) = 0.622 \times \rho_a \times \left(\dfrac{e_{\mathrm{sat}}(T_a)}{p_{\mathrm{SL}}}\right) \\ e_{\mathrm{sat}}(T_a) = 6.11 \times \exp\left(\dfrac{17.3 \times T_a}{T_a + 273.3}\right) \end{cases}, \end{aligned} \tag{A13}$$

where $\rho_{\mathrm{vsat}}(T_a)$ is the saturation absolute humidity (g m^{-3}) at the mean daily air temperature T_a (°C), ρ_a is the air density (1300 g m^{-3}), $e_{\mathrm{sat}}(T_a)$ is the saturation vapor pressure (kPa) at T_a, p_{SL} is the mean pressure at sea level (101.3 kPa), K_{PET} is a calibration constant, and L is the local day length expressed in hours [*Dingman*, 1994].

[54] The distribution of plant water extraction through the soil profile has a significant impact on the ability of vegetation to access water throughout the growing season [*Bond et al.*, 2008]. Roots, soil macropores, and soil saturated hydraulic conductivity all tend to fall off exponentially with depth [*Beven and Kirkby*, 1979; *Gale and Grigal*, 1987; *Jackson et al.*, 1996; *Sidle et al.*, 2001; *Wigmosta and Perkins*, 2001], which suggests that the ability to extract water from the soil column decreases with soil depth. A number of studies have found that the majority of water uptake is in the shallow soils where water and nutrients are abundant [*Jackson et al.*, 1996; *Warren et al.*, 2005]. However, these studies also suggest that water uptake shifts from shallow to deep layers as near surface soils dry out [*Brooks et al.*, 2006; *Hacke et al.*, 2000; *Warren et al.*, 2005]. To mimic these dynamics, soil water uptake is modeled as follows:

$$\begin{aligned} &\text{for } \left(\frac{s_{i=(d_r-1)}}{s^{\max}_{i=(d_r-1)}}\right) \geq \left(\frac{\theta^{w}_{i=(d_r-1)}}{\varphi_{i=(d_r-1)}}\right) \\ &\begin{cases} W_{E,i} = \dfrac{s_i}{\sum\limits_{j=1}^{d_r-1} s_j} & i = 1, 2, \ldots, (d_r - 1) \\ W_{E,i} = 0 & i = d_r, \ldots, n, \end{cases} \end{aligned} \tag{A14}$$

$$\begin{aligned} &\text{for } \left(\frac{s_{i=(d_r-1)}}{s^{\max}_{i=(d_r-1)}}\right) < \left(\frac{\theta^{w}_{i=(d_r-1)}}{\varphi_{i=(d_r-1)}}\right) \\ &\begin{cases} W_{E,i} = \dfrac{(1 - W_{E,\mathrm{deep}}) \times s_i}{\sum\limits_{j=1}^{d_r-1} s_j} & i = 1, 2, \ldots, (d_r - 1) \\ W_{E,i} = W_{E,\mathrm{deep}} & i = d_r \\ W_{E,i} = 0 & i = (d_r + 1), \ldots, n, \end{cases} \end{aligned} \tag{A15}$$

where θ_i^w and φ_i are the soil wilting point and the soil porosity in layer i, respectively, layer d_r is the deepest layer in which water extraction is possible, $\left(s_{i=(d_r-1)} \big/ s^{\max}_{i=(d_r-1)}\right)$ is the degree of saturation of layer (d_r −1), and $W_{E,\mathrm{deep}}$ is the fraction of water uptake from layer d_r during droughts (a calibrated value). The depth of layer d_r is determined either experimentally (typically taken from rooting depth information) or through calibration. Based on equation (A14), water uptake is limited to shallow layers as long as water storage in these layers is above wilting point. When the water storage in the shallow layers is below wilting point, equation (A15) permits for deep soil water extraction.

A2. Watershed Framework

[55] To place the above described soil column framework within a catchment framework, the catchment topography is gridded into a number of pixels (dependent upon the available DEM, e.g., 30 m), with each pixel consisting of one soil column model (Figure A1). Soil columns communicate with each other through the downslope lateral transport of water. For simplicity, lateral subsurface flow Q_i (equation (A16)) from one soil column pixel to a downslope neighbor is from layer i of the upslope pixel to layer i of the downslope pixel. Lateral surface flow Q_s (equation (A18)) is from the s_{STW} pool of an upslope pixel to the s_{STW} pool of a downslope pixel, where it can then either infiltrate into the top soil layer of the downslope pixel, or continue its downslope movement as lateral surface flow. A multiple flow direction method is used where flow from one pixel to its eight neighbors is fractionally allocated according to terrain slope [*Freeman*, 1991; *Quinn et al.*, 1991].

A2.1. Lateral Subsurface Runoff

[56] Lateral downslope flow (Q_i) is triggered near field capacity using the logistic function presented in equation (A9) but (1) corrected for the local slope and (2) Ks_i^{vertical} in $a_{1,i}$ and $a_{2,i}$ is replaced by Ks_i^{lateral} such that

$$Q_i = Ks_i^{\mathrm{lateral}} \times SL \times f\left(s_i / s_i^{\max}\right) \qquad i = 1, 2, \ldots, n \tag{A16}$$

and

$$Ks_i^{\mathrm{lateral}} = Ks \times e^{-f_l \times d_i} \qquad i = 1, 2, \ldots, n, \tag{A17}$$

where Ks_i^{lateral} is the lateral saturated hydraulic conductivity in layer i, SL is the local terrain slope, and f_l is the lateral decay rate of Ks with depth.

A2.2. Surface Runoff

[57] Surface runoff (Q_s) from a pixel is a product of the standing water after infiltration (I) is accounted for, the local terrain slope, and the Chezy "like" coefficient (Che) (1/time) [*Dingman*, 1994]:

$$Q_s = Che \times SL \times s_{STW}. \tag{A18}$$

A2.3. Total Runoff

[58] Total catchment discharge (Q_T) is computed as the sum of the lateral flows into the channel, the rain falling directly on the channel, and the snowmelt from channel pixels. The stream channel is defined as all pixels with a flow accumulation area above a predefined threshold. We assume that all flows entering the channel are directly routed to the outlet such that

$$Q_T = \sum_{j=1}^{cn} \sum_{i=1}^{n} (Q_{s,j} + Q_{i,j}) + \sum_{k=1}^{cr} (P_r + m), \tag{A19}$$

where cn is the number of pixels that are both adjacent to the channel and that have a flow direction into the channel, n is the number of layers in a soil column, and cr is the number of pixels within the channel.

Appendix B: Evapotranspiration Recovery Function Description

B1. Background

[59] Successional changes in forest transpiration are generally consistent with changes in forest Leaf Area Index (LAI), sapwood basal area, and net primary production (NPP) [*Watson et al.*, 1999; *Zimmermann et al.*, 2000]. *Ryan et al.*, [1997] found that forest LAI increases initially after disturbance, reaches a maximum in young stands, and thereafter decreases. *Moore et al.* [2004] found that young Douglas-fir forests in the Pacific Northwest have a higher sapwood basal area and use nearly three times as much water during the growing season as old-growth forests. *Acker et al.* [2002] found that the NPP of young stands in the Pacific Northwest is larger than the NPP of mature and old stands (Figure B1). Furthermore, several experimental studies found that the streamflow in managed forests is reduced to below old-growth values due to rapidly transpiring young vegetation [*Bond et al.*, 2008; *Hicks et al.*, 1991]. Finally, *Yang et al.* [2005] examined conifer development in 153 stands in the Pacific Northwest using interpretation of historic aerial photographs from 1959 to 1997 and found that coniferous forests regenerate quickly and reach closed canopy (defined as >70% tree cover) approximately 50 years after disturbance.

B2. Evapotranspiration Recovery Function

[60] Based on these findings, the ET function given in equation (A12) is modified to account for the (1) reduction in ET due to clearcut, (2) ET recovery during regrowth, (3) high transpiration demands of young forest, and (4) return to old-growth ET values within 50 years. Thus

$$ET_i = f_{TP}(t_d) \times W_{E,i} \times PET \times \left\{1 - \exp\left[-c_{ET} \times \left(s_i/s_i^{\max}\right)\right]\right\}$$
$$i = 1, 2, \ldots, n, \tag{B1}$$

where

$$f_{TP}(t_d) = f_{YF}(t_d) \times \{(1 - r_{ET}) \times [1 - \exp(-t_d/r_T)] + r_{ET}\} \tag{B2}$$

and

$$\text{for } (25 \le t_d \le 45 \text{ yrs } \& \ t_d \in \text{ Jun} - \text{Sept})$$
$$f_{YF}(t_d) = \frac{\Omega}{\{(1 - r_{ET}) \times [1 - \exp(-t_d/r_T)] + r_{ET}\}} \tag{B3}$$
$$\text{else} \quad f_{YF}(t_d) = 1,$$

where $f_{TP}(t_d)$ is the ET recovery function, t_d is time in days after disturbance, r_{ET} is the residual ET immediately after clearcut, r_T is an ET recovery shape factor, and Ω is the percentage increase in transpiration of young stands over

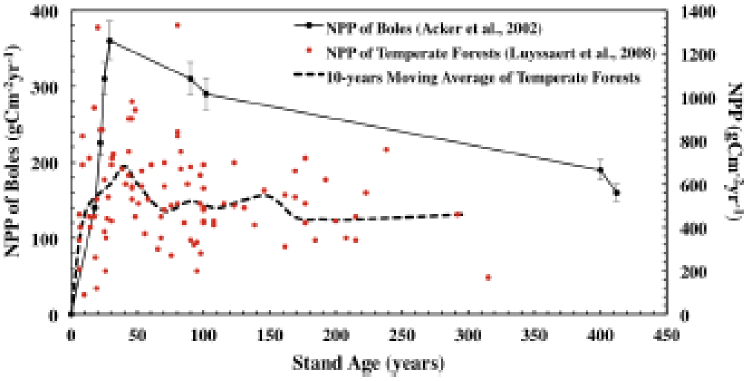


Figure B1. Changes in net primary production (NPP) of temperate forests (red dots are individual forest stands sampled throughout the world; dashed black line is a 10 year moving average) [*Luyssaert et al.*, 2008], and NPP of boles for Pacific Northwest coniferous forests (black circles and solid black line) as a function of stand age (i.e., time after stand-replacing disturbance) [*Acker et al.*, 2002].

Table B1. Model Parameters Values Used to Simulate the Hydrologic Processes of WS10 in the H.J. Andrews Experimental Forest

Parameters	Definition	Value	References
Soil texture	dominant soil texture	loam[a]	*Ranken* [1974]
θ_i^{fc}	field capacity in layer *i*	0.27	*Clapp and Hornberger* [1978]
ϕ_i	porosity in layer *i*	0.463	*Clapp and Hornberger* [1978]
θ_i^{w}	wilting point in layer *i*	0.117	*Clapp and Hornberger* [1978]
	Preharvest Calibrated Model Parameters		
Δz_1	soil surface layer (layer 1) thickness (mm)	300	calibrated
Δz_2	first intermediate soil layer (layer 2) thickness (mm)	750	calibrated
Δz_3	second intermediate soil layer (layer 3) thickness (mm)	750	calibrated
Δz_4	deep soil layer (layer 4) thickness (mm)	200	calibrated
Ks	surface saturated hydraulic conductivity (mm day^{-1})	950	calibrated
f_v	vertical decay rate of *Ks* (1/m)	1.3	calibrated
f_l	lateral decay rate of *Ks* (1/m)	1.55	calibrated
d_r	deepest layer in which water extraction is possible	3	*Santantonio et al.* [1977]
$W_{K,\,deep}$	fraction of water uptake from the deep layers during droughts	0.2	calibrated
c_{ET}	ET shape factor	5	calibrated
K_{PET}	potential evapotranspiration calibration parameter	2	calibrated
α	degree-day factor for melt (mm °C^{-1} day^{-1})	5	calibrated
T_{th}	threshold temperature (°C)	−1	calibrated
T_m	melting temperature (°C)	2	calibrated
σ	rain on snow parameter	0.5	calibrated
Che	Chezy "like" coefficient (1/day)	540,000	calibrated
	Postharvest Calibrated Model Parameters		
r_{ET}	residual evapotranspiration fraction after clearcut	0.3	calibrated
r_T	transpiration recovery shape factor	3000	calibrated
Ω	percentage increase in daily transpiration due to young vigorous forest (%)	10	calibrated

[a] Soil texture in WS10 range from gravelly, silty clay loam to very gravelly clay loam [*Ranken*, 1974].

old-growth during the growing season (assumed here between June and September). The function $f_{TP}(t_d)$ accounts for the increase in ET of young stands over old-growth.

[61] The ET recovery function $f_{TP}(t_d)$ is a modified Chapman-Richards growth function [*Hunt*, 1982; *Ratkowsky*, 1990; *Richards*, 1959] that accounts for the higher transpiration rate of young vigorous stands over old-growth [*Bond et al.*, 2008; *Jones and Post*, 2004]. Specifically, $f_{TP}(t_d)$ increases exponentially (from a clearcut value of $f_{TP}(t_d = 0) = r_{ET}$) and asymptotes to old-growth values within 50 years (i.e., $f_{TP}(t_d = 50 \text{ yrs}) = 1$). This ET recovery function (B2) is an initial attempt to capture the complex successional dynamics associated with canopy recovery. This function has been widely used in studies of trees and stand growth (e.g., at the H.J. Andrews, the Coweeta, and the Hubbard Brook Experimental Forests) [*Bosch and Von Gadow*, 1990; *Christina et al.*, 2011; *Duan*, 1996; *Janisch and Harmon*, 2002; *Khamis et al.*, 2005; *Watchler et al.*, 2005; *Zeide*, 1993, among others]. *Watchler et al.* [2005] used the Chapman-Richards growth function to capture canopy recovery in three watersheds within H.J. Andrews. *Yang et al.* [2005] used the Chapman-Richards growth function to simulate the recovery of shrubs, hardwood trees, conifer trees, and mixed trees successional postdisturbance dynamics.

[62] Calibration of the ET recovery function parameters is conducted as follows: Ω is calibrated to capture the observed 1975–2008 annual, seasonal, and monthly streamflow. r_{ET} is calibrated based on the annual P and Q_T record at WS10 immediately after clearcut and yields a value of 30% (Table B1), which is well within the range of observed values in forests across the United States and the Pacific Northwest [*Spittlehouse*, 2006; *Stednick*, 1996; *Stoy et al.*, 2006; *Winkler et al.*, 2010]. For example, a number of studies in the Pacific Northwest, including H.J. Andrews, have found that the initial ET after clearcut ranged from 280 to 550 mm (30% to 55% of preclearcut ET) [*Bosch and Hewlett*, 1982; *Hibbert*, 1966; *Stednick*, 1996].

[63] **Acknowledgments.** The information in this document has been funded in part by the U.S. Environmental Protection Agency. It has been subjected to the Agency's peer and administrative review, and it has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use. This research was additionally supported in part by the following NSF Grants 0439620, 0436118, and 0922100. We thank Sherri Johnson, Barbara Bond, Suzanne Remillard, Theresa Valentine, and Don Henshaw for invaluable assistance in accessing and interpreting various H.J. Andrews LTER data sets used in this study. Data for streamflow, stream chemistry and climate were provided by the H.J. Andrews Experimental Forest research program, funded by the National Science Foundation's Long-Term Ecological Research Program (DEB 08-23380), U.S. Forest Service Pacific Northwest Research Station, and OR State University.

References

Acker, S., C. Halpern, M. Harmon, and C. Dyrness (2002), Trends in bole biomass accumulation, net primary production and tree mortality in Pseudotsuga menziesii forests of contrasting age, *Tree Physiol.*, *22*(2–3), 213.

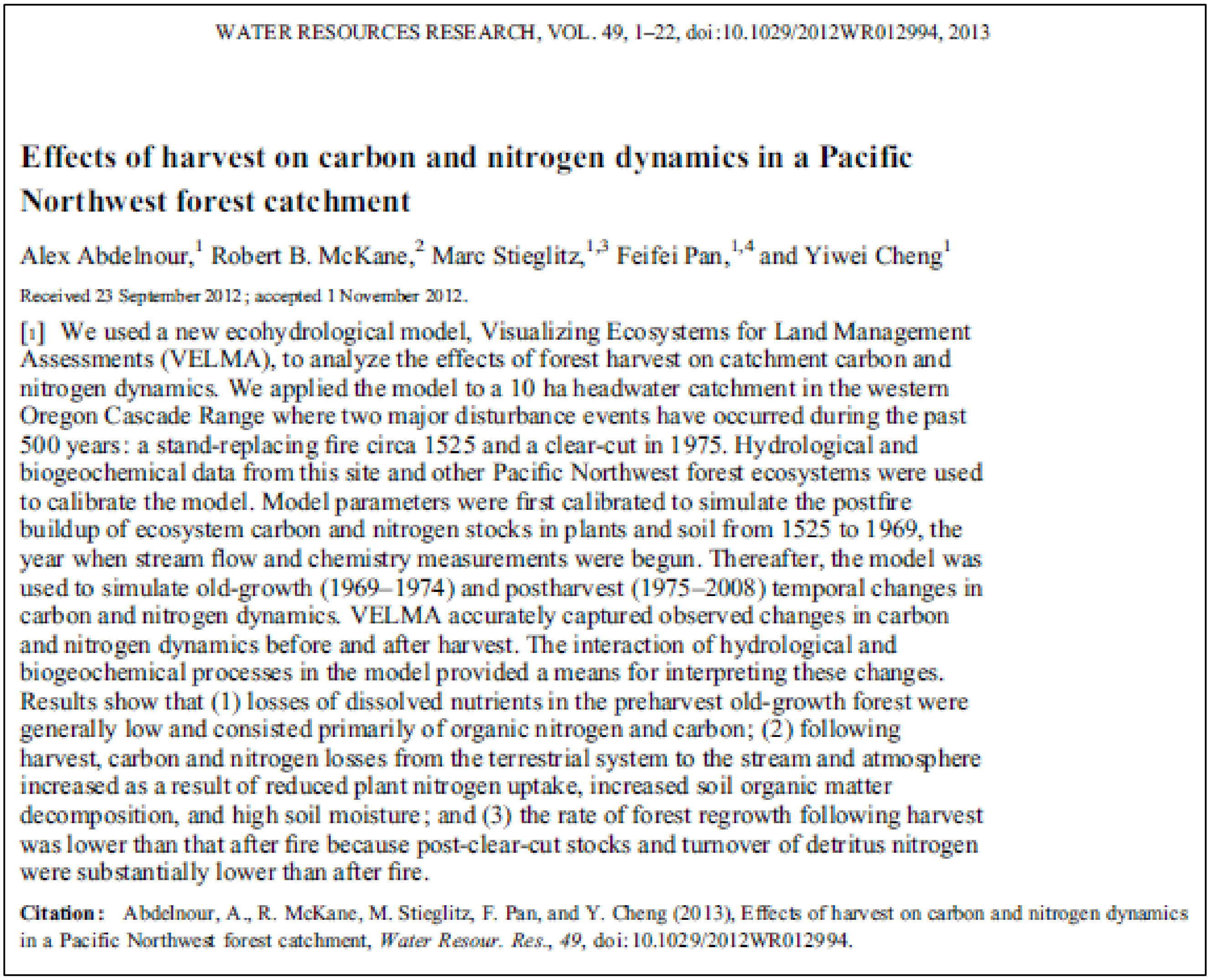
Beckers, J., B. Smerdon, T. Redding, A. Anderson, R. Pike, and A. T. Werner (2009), Hydrologic models for forest management applications: Part 1: Model selection, *Watershed Manage. Bull.*, *13*(1), 35–44.

Beschta, R. L. (1978), Long-term patterns of sediment production following road construction and logging in the Oregon Coast Range, *Water Resour. Res.*, *14*(6), 1011–1016, doi:1010.1029/WR1014i1006p01011.

Beschta, R., M. Pyles, A. Skaugset, and C. Surfleet (2000), Peakflow responses to forest practices in the western Cascades of Oregon, USA, *J. Hydrol.*, *233*(1–4), 102–120.

Appendix 6.2:
VELMA version 1.0, Part 2 of 2 BIOGEOCHEMICAL Model Description

(Link to All Parameters Table of Contents)



WATER RESOURCES RESEARCH, VOL. 49, 1–22, doi:10.1029/2012WR012994, 2013

Effects of harvest on carbon and nitrogen dynamics in a Pacific Northwest forest catchment

Alex Abdelnour,[1] Robert B. McKane,[2] Marc Stieglitz,[1,3] Feifei Pan,[1,4] and Yiwei Cheng[1]

Received 23 September 2012; accepted 1 November 2012.

[1] We used a new ecohydrological model, Visualizing Ecosystems for Land Management Assessments (VELMA), to analyze the effects of forest harvest on catchment carbon and nitrogen dynamics. We applied the model to a 10 ha headwater catchment in the western Oregon Cascade Range where two major disturbance events have occurred during the past 500 years: a stand-replacing fire circa 1525 and a clear-cut in 1975. Hydrological and biogeochemical data from this site and other Pacific Northwest forest ecosystems were used to calibrate the model. Model parameters were first calibrated to simulate the postfire buildup of ecosystem carbon and nitrogen stocks in plants and soil from 1525 to 1969, the year when stream flow and chemistry measurements were begun. Thereafter, the model was used to simulate old-growth (1969–1974) and postharvest (1975–2008) temporal changes in carbon and nitrogen dynamics. VELMA accurately captured observed changes in carbon and nitrogen dynamics before and after harvest. The interaction of hydrological and biogeochemical processes in the model provided a means for interpreting these changes. Results show that (1) losses of dissolved nutrients in the preharvest old-growth forest were generally low and consisted primarily of organic nitrogen and carbon; (2) following harvest, carbon and nitrogen losses from the terrestrial system to the stream and atmosphere increased as a result of reduced plant nitrogen uptake, increased soil organic matter decomposition, and high soil moisture; and (3) the rate of forest regrowth following harvest was lower than that after fire because post-clear-cut stocks and turnover of detritus nitrogen were substantially lower than after fire.

Citation: Abdelnour, A., R. McKane, M. Stieglitz, F. Pan, and Y. Cheng (2013), Effects of harvest on carbon and nitrogen dynamics in a Pacific Northwest forest catchment, *Water Resour. Res.*, *49*, doi:10.1029/2012WR012994.

when N fixing plants and microorganisms tend to be more abundant [*Rastetter et al.*, 2001]. However, this simplification is acceptable for WS10, given the low abundance of N fixers in the young, postharvest forest. *Gholz et al.* [1985] found that post-clear-cut N fixers such as red alder (*Alnus rubra*) and snowbush (*Ceanothus veluntinus*) were sparse and limited to the riparian zone of WS10.

[39] (4) *Controls on dissolved organic matter*: VELMA does not explicitly model processes that control the retention and loss of dissolved organic matter. In a field study conducted in HJA, *Yano et al.* [2004, 2005] found that the net retention of soil dissolved organic matter are controlled by the relative magnitude of hydrophobic and hydrophilic acid fractions as well as by the litter quality. For example, *Yano et al.* [2005] found that DON production from root litters is an order of magnitude higher than other litter types (i.e. needle and wood). Other studies have shown that anions such as sulfate and phosphate can impact retention of DOC in forest soils by competing with DOC for adsorption sites within the soil matrix [*Vance and David*, 1992]. Although the incorporation of the chemical processes discussed above in VELMA is beyond the scope of this paper, doing so would provide a more accurate representation of mechanisms controlling catchment-scale DON and DOC production and export. In its present configuration, VELMA is calibrated to provide a best fit for observed DON and DOC exports. Thus, the chemical processes affecting measured concentrations of DON and DOC are implicitly included in this model calibration. Consequently, an explicit treatment of the chemical processes would require recalibration of the terrestrial processes controlling DON and DOC transport to the stream.

[40] (5) *Soil spatial heterogeneity*: Soil texture and depth vary spatially within WS10 [*McGuire et al.*, 2007; *Ranken*, 1974; *Sayama and McDonnell*, 2009]. However, deriving high-resolution and catchment wide soil texture and depth maps from, typically, a small number of point measurements, is, at best, uncertain. Instead, we assume uniform loam soil texture and uniform depth to bedrock of 2 m to reflect, more or less, average conditions in the catchment [*Ranken*, 1974]. Although a sensitivity analysis on the impact of the spatial distribution of soil texture and soil depth on streamflow dynamics would certainly provide insights into catchment dynamics, it is beyond the scope of this paper.

[41] For some applications, the explicit treatment of these processes may be needed. However, it must be recognized that such added processes come at the cost of increased model complexity, computational efficiency, and applicability to larger spatial and temporal scales. These are important tradeoffs to consider, given that data needed to implement complex models are not generally available.

Appendix A: Model Description

[42] VELMA is a spatially distributed ecohydrology model that accounts for hydrological and biogeochemical processes within watersheds. The model simulates daily to century-scale changes in soil water storage, surface and subsurface runoff, vertical drainage, carbon and nitrogen cycling in plants and soils, as well as transport of nutrients from the terrestrial landscape to the streams. VELMA consists of multilayered soil column models that communicate with each other through the downslope lateral transport of water and nutrients (Figure (A1)). Each soil column model consists of three coupled submodels: a hydrological model, a soil temperature model, and a plant-soil model. Described below are the soil temperature and plant-soil component of the model. The hydrological component was described in a previous manuscript [*Abdelnour et al.*, 2011]. First, we describe the soil column model and then place this soil column within a catchment framework.

A1. Soil Column Framework

[43] We employ a multilayered soil column as a fundamental hydrologic and ecological unit. The soil column consists of *n* soil layers (Figures (A2) and (A3)). Soil water balance, soil subsurface temperature, and soil C and N pools are computed for each layer.

A1.1. Soil Temperature Model

[44] The soil temperature model first simulates the ground surface temperature (GST) from the available mean surface air temperature (T_a) in the presence of snow cover.

[45] The ground surface temperature is calculated as follows:

$$\mathrm{GST}(t) = T_a \times \exp^{\left(\frac{\mathrm{SD}(t)}{\lambda_{\mathrm{snow}}}\right)}, \qquad \text{(A1)}$$

where SD(t) is the snow depth (mm) at time t, and λ_{snow} is the seasonal damping depth for snow that is approximately equal to 670 mm [*Hillel*, 1998] for a snowpack of density 300 kg m^{-3}. In this model, snow is an insulative material that only attenuates the mean surface air temperature signal [*Cheng et al.*, 2010]. The attenuation of the T_a signal is assumed proportional to the depth of the snowpack [*Cheng et al.*, 2010]. As a result, during snow free periods, the ground surface temperature is assumed equal to the mean surface air temperature: SD = 0, GST = T_a.

[46] Subsurface heat transfer is then simulated using the analytical solution of the one-dimensional heat-conduction equation [*Carslaw and Jaeger*, 1959; *Hillel*, 1998].

[47] The subsurface soil temperature in layer i is calculated as follows:

$$T_{s,i}(d_i,t) = \overline{\mathrm{GST}} + \left(\mathrm{GST}(t-\phi(d_i,t)) - \overline{\mathrm{GST}}\right) \times \exp^{\left(-\frac{d_i}{\delta(t)}\right)} \qquad i = 1,2\ldots n, \qquad \text{(A2)}$$

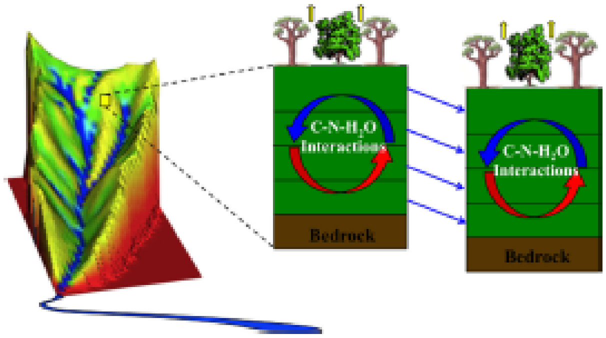


Figure A1. Conceptual catchment modeling framework using multilayered soil columns.

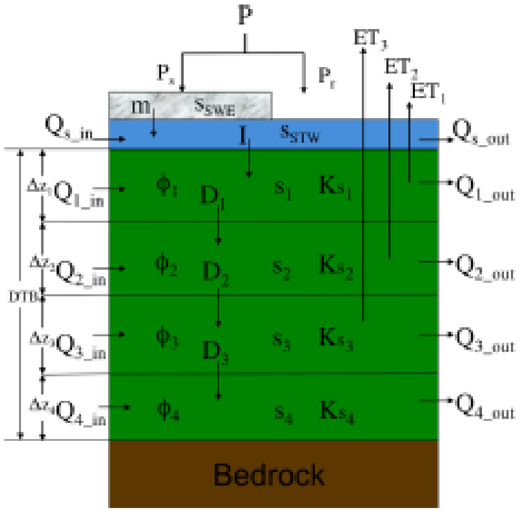


Figure A2. The soil column hydrological framework consists of a four-layered soil column, a standing water layer, and a snow layer. DTB is the soil column depth to bedrock. Δz_i, Ks_i, ϕ_i, and s_i are the thickness, the saturated hydraulic conductivity, the soil porosity, and the soil water storage of layer i, respectively; P, P_s, and P_r are the precipitation, snow, and rain, respectively; m is the snowmelt, and s_{SWE} is the snow water equivalent depth; I is the infiltration, and s_{STW} is the standing water amount; Q_s is the surface runoff; and Q_i, D_i, and ET_i, are the subsurface runoff, the drainage and the ET of layer i, respectively.

where $\overline{\mathrm{GST}}$ is the annual mean soil temperature (°C), d_i is the soil depth to the middle of layer i (mm), $\phi(d_i, t)$ is the phase lag of $T_{s,i}$ relative to GST at depth d_i:

$$\phi(d_i, t) = \left(\frac{d_i}{\lambda(t)} \times \frac{360}{(2\pi)}\right) \qquad i = 1, 2, \ldots, n, \tag{A3}$$

and $\lambda(t)$ is the damping depth of the soil (mm), defined as the characteristic depth at which the temperature signal is attenuated to $1/e$ of the GST. $\lambda(t)$ is a function of the thermal properties of the soil and the frequency of the temperature fluctuation:

$$\lambda(t) = \left(\frac{2D_h(t)}{w}\right)^{\frac{1}{2}}, \tag{A4}$$

where $D_h(t)$ is the time-dependent thermal diffusivity of the soil (mm^2 d^{-1}) and is a function of the simulated soil moisture $\left(\frac{s_i}{s_i^{\max}}\right)$ [*De Vries*, 1975]. For each layer i of the soil column,

$$\begin{aligned} D_h(t) &= (19.45 \times 10^{-3}) \times \left(\frac{S_i}{S_i^{\max}}\right) + 2 \times 10^{-3} \\ &\quad \text{for} \left(\frac{S_i}{S_i^{\max}}\right) < 0.18 \\ D_h(t) &= (-4.055 \times 10^{-3}) \times \left(\frac{S_i}{S_i^{\max}}\right) + 6.23 \times 10^{-3} \\ &\quad \text{for} \left(\frac{S_i}{S_i^{\max}}\right) \geq 0.18 \quad i = 1, 2, \ldots n, \end{aligned} \tag{A5}$$

and w is the frequency of annual temperature fluctuation (d^{-1}):

$$w = \frac{2\pi}{365}. \tag{A6}$$

A1.2. Plant-Soil Model

[48] The plant-soil model simulates ecosystem carbon storage and the cycling of carbon and nitrogen between a plant biomass layer and the active soil pools (Figure (A3)). Specifically, the model simulates the interaction among plant biomass (B), SOC including humus and detritus (SOC), plant available soil nitrogen (N) including DON and DIN, as well as DOC. The DIN pool is divided into an ammonium (NH$_4$) and nitrate (NO$_3$) pool. B, SOC, NH$_4$, NO$_3$, DON, and DOC pools are updated at each time step. For an n-layer soil model ($i = 1, 2, \ldots, n$):

$$\frac{\mathrm{d}B}{\mathrm{d}t} = \left[\left(\sum_{i=1}^{n} \frac{r_i \mu_i \delta_{\mathrm{NH}_4} \mathrm{NH}_{4,i}}{\mathrm{NH}_{4,i} kn}\right)\left(\sum_{i=1}^{n} \frac{r_i \mu_i \delta_{\mathrm{NO}_3} \mathrm{NO}_{3,i}}{\mathrm{NH}_{3,i} kn}\right)\right] \mathrm{WS} \left(\frac{s_i}{s_i^{\max}}\right) B - m, \tag{A7}$$

$$\frac{\mathrm{dSOC}_i}{\mathrm{d}t} = r_i m(B) B - \nu_i(T_{s,i}, s_i)\mathrm{SOC}_i, \tag{A8}$$

$$\begin{aligned} \frac{d\mathrm{NH}_{4,i}}{\mathrm{d}t} = {} & n_{\mathrm{in}} - r_i \mu_i \delta_{\mathrm{NH}_4} f_M(\mathrm{NH}_{4,i}) \mathrm{WS}\left(\frac{s_i}{s_i^{\max}}\right) B \\ & + (1-q)\mathrm{SOC}_i \nu_i(T_{s,i}, s_i) - \mathrm{Nit}_i - \zeta_v(\mathrm{NH}_{4,i}) \\ & + \zeta_v(\mathrm{NH}_{4,i-1}) + \zeta_{l_{\mathrm{in}}}(\mathrm{NH}_{4,i}) - \zeta_{l_{\mathrm{out}}}(\mathrm{NH}_{4,i}), \end{aligned} \tag{A9}$$

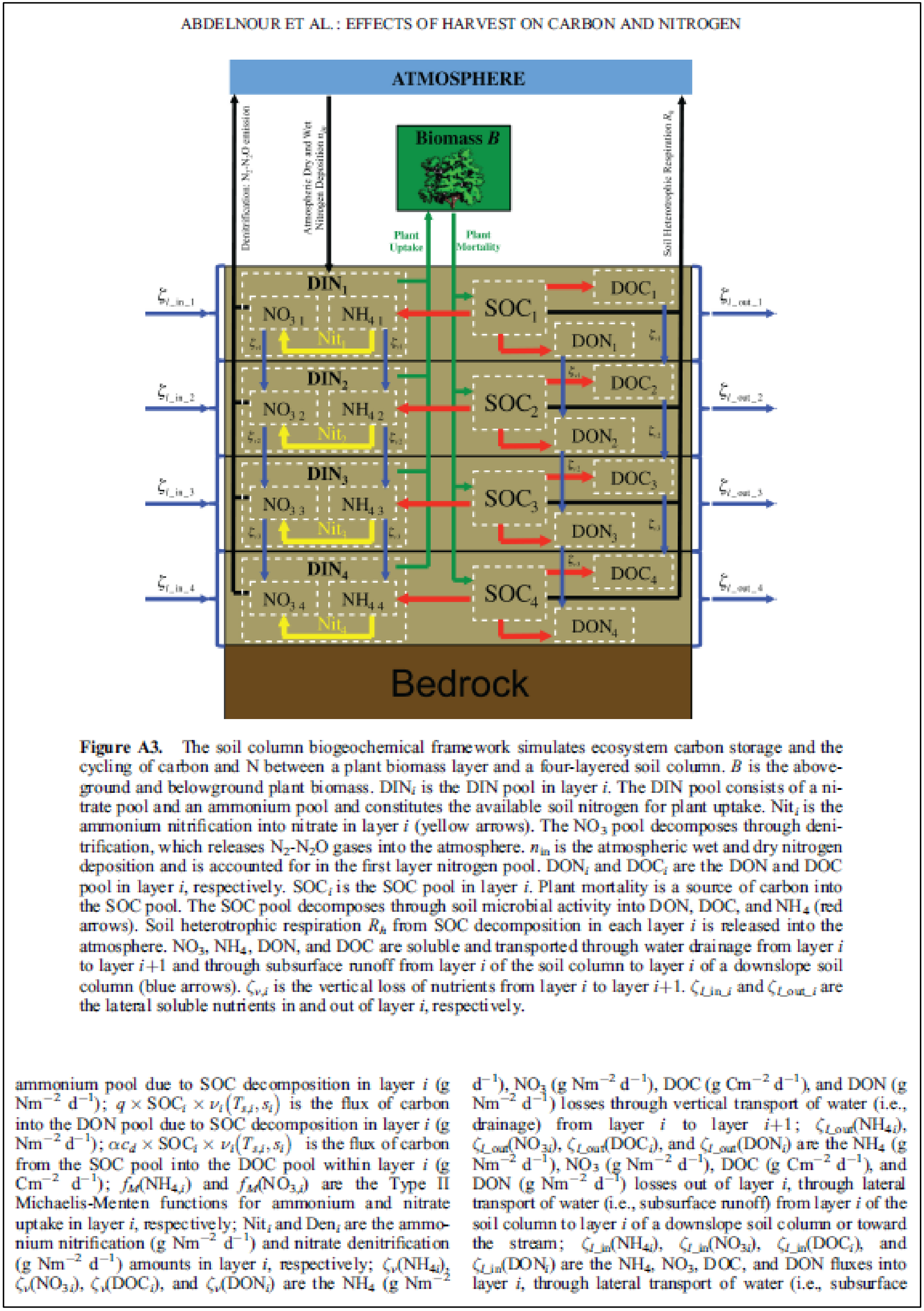
$$\begin{aligned} \frac{d\mathrm{NO}_{3,i}}{\mathrm{d}t} = {} & \mathrm{Nit}_i - r_i \mu_i \delta_{\mathrm{NO}_3} f_M(\mathrm{NO}_{3,i}) \mathrm{WS}\left(\frac{s_i}{s_i^{\max}}\right) B - \mathrm{Den}_i \\ & - \zeta_v(\mathrm{NO}_{3,i}) + \zeta_v(\mathrm{NO}_{3,i-1}) + \zeta_{l_{\mathrm{in}}}(\mathrm{NO}_{3,i}) \\ & - \zeta_{l_{\mathrm{out}}}(\mathrm{NO}_{3,i}), \end{aligned} \tag{A10}$$

$$\begin{aligned} \frac{\mathrm{dDOC}_i}{\mathrm{d}t} = {} & \alpha_{\mathrm{CN}} c_d \mathrm{SOC}_i \ \nu_i(T_{s,i}, s_i) - \zeta_v(\mathrm{DOC}_i) + \zeta_v(\mathrm{DOC}_{i-1}) \\ & + \zeta_{l_{\mathrm{in}}}(\mathrm{DOC}_i) - \zeta_{l_{\mathrm{out}}}(\mathrm{DOC}_i), \end{aligned} \tag{A11}$$

$$\begin{aligned} \frac{\mathrm{dDON}_i}{\mathrm{d}t} = {} & q\mathrm{SOC}_i \ \nu_i(T_{s,i}, s_i) - \zeta_v(\mathrm{DON}_i) + \zeta_v(\mathrm{DON}_{i-1}) \\ & + \zeta_{l_{\mathrm{in}}}(\mathrm{DON}_i) - \zeta_{l_{\mathrm{out}}}(\mathrm{DON}_i), \end{aligned} \tag{A12}$$

where $m(B)$ is the plant mortality rate (d^{-1}); r_i and μ_i are the biomass root fraction and the uptake rate function (d^{-1}) in layer i, respectively; δ_{NH_4} and δ_{NO_3} are the fractions of nitrogen uptake from the ammonium and nitrate pool, respectively; SOC$_i$, NH$_{4,i}$, NO$_{3,i}$, DOC$_i$, and DON$_i$ are the SOC, ammonium, nitrate, DOC, and DON pools in layer i, respectively (g Nm^{-2} or g Cm^{-2}); kn (g Nm^{-2}) is the Michaelis-Menten calibration parameter; $\mathrm{WS}(s_i/s_i^{\max})$ is the water stress function; $\nu_i(T_{s,i}, s_i)$ is a first-order SOC decomposition rate (d^{-1}); n_{in} is the atmospheric input of wet and dry nitrogen deposition (g Nm^{-2} d^{-1}); $(1-q) \times \mathrm{SOC}_i \times \nu_i(T_{s,i}, s_i)$ is the flux of carbon into the

The VELMA v1.0 figure describes simulated interactions of hydrological and biogeochemical processes. See Figure 1 in the Chapter IV of the VELMA v2.0 User Manual for modifications to the PSM portion of the model.



ABDELNOUR ET AL.: EFFECTS OF HARVEST ON CARBON AND NITROGEN

ATMOSPHERE
Denitrification: N_2-N_2O emission
Atmospheric Dry and Wet Nitrogen Deposition n_{in}
Biomass B
Plant Uptake
Plant Mortality
Soil Heterotrophic Respiration R_h
$\zeta_{l\_in\_1}$
$\zeta_{l\_in\_2}$
$\zeta_{l\_in\_3}$
$\zeta_{l\_in\_4}$
DIN_1
$NO_{3\,1}$
$NH_{4\,1}$
Nit_1
ζ_{v1}
SOC_1
DOC_1
DON_1
DIN_2
$NO_{3\,2}$
$NH_{4\,2}$
Nit_2
ζ_{v2}
SOC_2
DOC_2
DON_2
DIN_3
$NO_{3\,3}$
$NH_{4\,3}$
Nit_3
ζ_{v3}
SOC_3
DOC_3
DON_3
DIN_4
$NO_{3\,4}$
$NH_{4\,4}$
Nit_4
SOC_4
DOC_4
DON_4
$\zeta_{l\_out\_1}$
$\zeta_{l\_out\_2}$
$\zeta_{l\_out\_3}$
$\zeta_{l\_out\_4}$
Bedrock

Figure A3. The soil column biogeochemical framework simulates ecosystem carbon storage and the cycling of carbon and N between a plant biomass layer and a four-layered soil column. B is the aboveground and belowground plant biomass. DIN_i is the DIN pool in layer i. The DIN pool consists of a nitrate pool and an ammonium pool and constitutes the available soil nitrogen for plant uptake. Nit_i is the ammonium nitrification into nitrate in layer i (yellow arrows). The NO_3 pool decomposes through denitrification, which releases N_2-N_2O gases into the atmosphere. n_{in} is the atmospheric wet and dry nitrogen deposition and is accounted for in the first layer nitrogen pool. DON_i and DOC_i are the DON and DOC pool in layer i, respectively. SOC_i is the SOC pool in layer i. Plant mortality is a source of carbon into the SOC pool. The SOC pool decomposes through soil microbial activity into DON, DOC, and NH_4 (red arrows). Soil heterotrophic respiration R_h from SOC decomposition in each layer i is released into the atmosphere. NO_3, NH_4, DON, and DOC are soluble and transported through water drainage from layer i to layer $i+1$ and through subsurface runoff from layer i of the soil column to layer i of a downslope soil column (blue arrows). $\zeta_{v,i}$ is the vertical loss of nutrients from layer i to layer $i+1$. $\zeta_{l\_in\_i}$ and $\zeta_{l\_out\_i}$ are the lateral soluble nutrients in and out of layer i, respectively.

ammonium pool due to SOC decomposition in layer i (g Nm^{-2} d^{-1}); $q \times SOC_i \times \nu_i(T_{s,i}, s_i)$ is the flux of carbon into the DON pool due to SOC decomposition in layer i (g Nm^{-2} d^{-1}); $\alpha c_d \times SOC_i \times \nu_i(T_{s,i}, s_i)$ is the flux of carbon from the SOC pool into the DOC pool within layer i (g Cm^{-2} d^{-1}); $f_M(NH_{4,i})$ and $f_M(NO_{3,i})$ are the Type II Michaelis-Menten functions for ammonium and nitrate uptake in layer i, respectively; Nit_i and Den_i are the ammonium nitrification (g Nm^{-2} d^{-1}) and nitrate denitrification (g Nm^{-2} d^{-1}) amounts in layer i, respectively; $\zeta_v(NH_{4i})$, $\zeta_v(NO_{3i})$, $\zeta_v(DOC_i)$, and $\zeta_v(DON_i)$ are the NH_4 (g Nm^{-2} d^{-1}), NO_3 (g Nm^{-2} d^{-1}), DOC (g Cm^{-2} d^{-1}), and DON (g Nm^{-2} d^{-1}) losses through vertical transport of water (i.e., drainage) from layer i to layer $i+1$; $\zeta_{l\_out}(NH_{4i})$, $\zeta_{l\_out}(NO_{3i})$, $\zeta_{l\_out}(DOC_i)$, and $\zeta_{l\_out}(DON_i)$ are the NH_4 (g Nm^{-2} d^{-1}), NO_3 (g Nm^{-2} d^{-1}), DOC (g Cm^{-2} d^{-1}), and DON (g Nm^{-2} d^{-1}) losses out of layer i, through lateral transport of water (i.e., subsurface runoff) from layer i of the soil column to layer i of a downslope soil column or toward the stream; $\zeta_{l\_in}(NH_{4i})$, $\zeta_{l\_in}(NO_{3i})$, $\zeta_{l\_in}(DOC_i)$, and $\zeta_{l\_in}(DON_i)$ are the NH_4, NO_3, DOC, and DON fluxes into layer i, through lateral transport of water (i.e., subsurface runoff) from layer i of an upslope soil column; α is the C:N ratio for plants and soils and is currently assumed constant for the entire simulations; c_d is the fraction of carbon that is not lost to the atmosphere due to soil heterotrophic respiration.

A1.2.1. Atmospheric Nitrogen Deposition

[49] Atmospheric inputs of wet and dry nitrogen deposition are assumed to affect only the first soil layer and to be temporally distributed throughout the year as a function of precipitation:

$$n_{\text{in}} = \overline{n}_{\text{in}} \times \frac{(P_r + m)}{P_{\text{an}}}, \qquad \text{(A13)}$$

where $\overline{n}_{\text{in}}$ is the long-term average annual wet and dry nitrogen deposition (g Nm^{-2} yr^{-1}), P_r is the rain (mm d^{-1}), m is the snowmelt (mm d^{-1}), and P_{ann} is the long-term average annual precipitation (mm yr^{-1}).

A1.2.2. Michaelis-Menten Functions

[50] The Type II Michaelis-Menten functions are used to limit NH_4 and NO_3 uptake.

$$\begin{aligned} f_M(\mathrm{NH}_{4i}) &= \frac{\mathrm{NH}_{4i}}{\mathrm{NH}_{4i} + kn} \\ f_M(\mathrm{NO}_{3i}) &= \frac{\mathrm{NO}_{3i}}{\mathrm{NO}_{3i} + kn} \qquad i = 1, 2 \ldots n, \end{aligned} \qquad \text{(A14)}$$

A1.2.3. Plant Mortality

[51] Plant mortality rate is simulated as a function of plant biomass. *Acker et al.* [2002] found that biomass mortality increases slowly with age for young stand, until it reaches the mortality of mature and old-growth stands. In VELMA, plant mortality is assumed to increase exponentially with biomass and to reach a steady state value for mature/old-growth stands.

$$m(B)\begin{cases} \left(\dfrac{(m_a \times B)_{mb} \times m_c}{B}\right) & \text{for } B < B_{st} \\ m_{st} & \text{for } B \geq B_{st} \end{cases}, \qquad \text{(A15)}$$

where m_a, m_b, and m_c are the mortality rate parameters; m_{st} is the equilibrium mortality rate of old-growth stands (d^{-1}); and B_{st} is the biomass value at equilibrium for an old-growth stand (g Nm^{-2} or g Cm$^{-2}/\alpha$).

A1.2.4. Plant Uptake

[52] Plant uptake rate is assumed to increase with increasing stand age (S_{age}), reach a maximum value for young stand, and then decrease and reach equilibrium value for mature/old-growth stand [*Acker et al.*, 2002; *Waring and Franklin*, 1979].

$$\mu_i = \begin{cases} \mu_{\min} + 1.44 \times \left(\dfrac{W_{k_1}}{W_{\lambda_1}}\right) \times \left(\dfrac{S_{\text{age}}}{W_{\lambda_1} \times S_{\text{age}}^{\max}}\right)^{w_{k_1}-1} \times \exp^{\left(\dfrac{S_{\text{age}}}{W_{\lambda_1} \times S_{\text{age}}^{\max}}\right)^{w_{k_1}}} & \text{for } S_{\text{age}} \leq S_{\text{age}}^{\max} \\ \mu_{st} + \left(\dfrac{W_{k_1}}{W_{\lambda_2}}\right) \times \left(\dfrac{S_{\text{age}}}{W_{\lambda_2} \times S_{\text{age}}^{\max}}\right)^{w_{k_2}-1} \times \exp^{\left(\dfrac{S_{\text{age}}}{W_{\lambda_2} \times S_{\text{age}}^{\max}}\right)} & \text{for } S_{\text{age}} > S_{\text{age}}^{\max} \end{cases} \quad i = 1, 2, \ldots n, \qquad \text{(A16)}$$

where $\mu_{\min}$ is the minimum uptake rate of plant (d^{-1}); μ_{st} is the steady-state/equilibrium value of plant uptake (d^{-1}); $S_{\text{age}}^{\max}$ is the stand age for which plant uptake is the highest (days); and W_{k_1}, W_{λ_1}, W_{k_2} and W_{λ_2} are the Weibull distribution parameters to calibrate.

A1.2.5. Water Stress Function

[53] The water stress function varies between 0 and 1 and is proportional to the soil layer water saturation. The water stress function limits plant growth (i.e., plant nutrient uptake capacity), as soil layer wetness approaches zero or saturation.

$$\mathrm{WS}\left(\frac{S_i}{S_i^{\max}}\right) = \begin{cases} 0.002154 \times \exp^{\left(15.3511 \times \left(\frac{S_i}{S_i^{\max}}\right)\right)} & \text{for } \left(\dfrac{S_i}{S_i^{\max}}\right) < \mathrm{WS}^{\min} \\ 1 & \text{for } \mathrm{WS}^{\min} \leq \left(\dfrac{S_i}{S_i^{\max}}\right) \leq \mathrm{WS}^{\max} \\ 2.44141 \times \exp^{\left(-1.116 \times \left(\frac{S_i}{S_i^{\max}}\right)\right)} & \text{for } \left(\dfrac{S_i}{S_i^{\max}}\right) > \mathrm{WS}^{\max} \end{cases} \quad i = 1, 2, \ldots n, \qquad \text{(A17)}$$

where $WS^{\min}$ and $WS^{\max}$ are the minimum and maximum soil layer water saturation values between which water stress function has no effect on plant nutrient uptake, respectively.

A1.2.6. Biomass Root Fraction

[54] Biomass root fraction distribution with depth follows *Gale and Grigal* [1987] model of vertical root distribution:

$$r_i = 1 - \beta^{d_i} - \sum_{j=1}^{i} r_j, \qquad i = 1, 2, \ldots n, \tag{A18}$$

where β is a fitted "extinction coefficient" that depends on the vegetation type.

A1.2.7. Vertical Transport of Nutrient

[55] Vertical transport of nutrients within the soil column is a function of the vertical water drainage and the size of the nutrient pool in layer i:

$$\varsigma_v(\mathrm{NH}_{4i}) = qf_{\mathrm{NH}_4} \times \frac{D_i}{S_i} \times \mathrm{NH}_{4i} \qquad i = 1, 2 \ldots n, \tag{A19}$$

$$\varsigma_v(\mathrm{NO}_{3i}) = qf_{\mathrm{NO}_3} \times \frac{D_i}{S_i} \times \mathrm{NO}_{3i} \qquad i = 1, 2 \ldots n, \tag{A20}$$

$$\varsigma_v(\mathrm{DON}_i) = qf_{\mathrm{DON}} \times \frac{D_i}{S_i} \times \mathrm{DON}_i \qquad i = 1, 2 \ldots n, \tag{A21}$$

$$\varsigma_v(\mathrm{DOC}_i) = qf_{\mathrm{DOC}} \times \frac{D_i}{S_i} \times \mathrm{DOC}_i \qquad i = 1, 2 \ldots n, \tag{A22}$$

where D_i (mm d^{-1}) is the vertical water drainage from layer i to layer i+1; S_i (mm) is the amount of water in layer i; and qf_{NH_4},qf_{NO_3},qf_{DON}, and qf_{DOC} are the maximum fractions of NH_4, NO_3, DON, and DOC pools, respectively, that can be lost through transport of water.

A1.2.8. SOC Decomposition

[56] SOC decomposition rate varies with environmental factors such as soil temperature [*Katterer et al.*, 1998; *Lloyd and Taylor*, 1994; *Rustad and Fernandez*, 1998] and soil moisture [*Davidson et al.*, 2000] and is based on the process-based TEM presented by *Raich et al.* [1991]. Soil moisture impacts SOC decomposition rate via moisture availability in dry soil and via oxygen availability in wet soil, such as

$$v_i(T_{s,i}, s_i) = k_c \times F_c^{\mathrm{temp}}(T_{s,i}) \times F_c^{\mathrm{moisture}}\left(\frac{s_i}{S_i^{\max}}\right), \quad i = 1, 2, \ldots n, \tag{A23}$$

$$F_c^{\mathrm{temp}}(T_{s,i}) = 0.00442 \times e^{\left(0.0693 \times T_{s,i}\right)}, \quad i = 1, 2, \ldots n, \tag{A24}$$

$$F_c^{\mathrm{moisture}}\left(\frac{S_i}{S_i^{\max}}\right) = 0.8 \times M_{\mathrm{sat}}^{A} + 0.2, \quad i = 1, 2, \ldots n, \tag{A25}$$

$$A = \left(\frac{\left(100 \times \left(\frac{S_i}{S_i^{\max}}\right)\right)^{ma_1} - \left(\frac{S_i}{S_i^{\max}}\right)_{\mathrm{opt}}^{ma_1}}{\left(\frac{S_i}{S_i^{\max}}\right)_{\mathrm{opt}}^{ma_1} - 100^{ma_1}}\right)^2, \qquad i = 1, 2, \ldots n, \tag{A26}$$

where k_c (d^{-1}) is the potential decomposition rate determined by model calibration; $F_c^{\mathrm{temp}}(T_{s,i})$ relates the microbial activity rate to changes in soil temperature (Equation 1.13 [*Raich et al.*, 1991]); $F_c^{\mathrm{moisture}}(S_i/S_i^{\max})$ defines the impact of soil moisture on decomposition (Equation 1.14b [*Raich et al.*, 1991]); M_{sat} is a parameter that determines the value of $F_c^{\mathrm{moisture}}(S_i/S_i^{\max})$ when the soil is saturated (Table A5 [*Raich et al.*, 1991]); ma_1 is a shape parameter defining the skewness of the curve (Table A5 [*Raich et al.*, 1991]); $\left(S_i/S_i^{\max}\right)_{\mathrm{opt}}$ is the optimal soil wetness value for which carbon decomposition is maximal; and $F_c^{\mathrm{temp}}(T_{s,i})$ and $F_c^{\mathrm{moisture}}(S_i/S_i^{\max})$ vary between the values of 0 and 1.

A1.3. Nitrification Rate

[57] Nitrification is the biological oxidation of ammonium into nitrite and subsequently nitrate under aerobic conditions. Nitrification occurs naturally in the environment and is carried out by autotrophic bacteria. Soil nitrification rates depend on a number of environmental factors such as soil ammonium level [*Smart et al.*, 1999], soil moisture [*Davidson et al.*, 1993], soil temperature [*Malhi and McGill*, 1982], and soil pH [*DeGroot et al.*, 1994]. In VELMA, nitrification is simulated using similar equations to the generalized model of N_2 and N_2O production of *Parton et al.* [1996, 2001]. Soil nitrification rate is assumed to (1) increase exponentially with soil temperature ($F_N^{\mathrm{temp}}(T_{s,i})$; Figure 2b [*Parton et al.*, 1996]), (2) increase as soil layer water saturation reaches optimal value for bacterial decomposition and then decrease rapidly as soil layer reaches saturation ($F_N^{\mathrm{moisture}}(S_i/S_i^{\max})$; Figure 2a [*Parton et al.*, 1996]), (3) decrease exponentially as soil layer acidity (pH_i) increases ($F_N^{\mathrm{moisture}}(S_i/S_i^{\max})$; Figure 2c [*Parton et al.*, 1996]), and (4) be limited by the amounted of ammonium available for nitrification.

[58] The nitrification rate in layer i is calculated as follows:

$$\mathrm{Nit}_i = K_N^{\max} \times F_N^{\mathrm{temp}}(T_{s,i}) \times F_N^{\mathrm{acidity}}(\mathrm{pH}_i) \times F_N^{\mathrm{moisture}}\left(\frac{S_i}{S_i^{\max}}\right) \times f_m(\mathrm{NH}_{4i}) \times \mathrm{NH}_{4i} \qquad i = 1, 2 \ldots n, \tag{A27}$$

$$F_N^{\mathrm{temp}}(T_{s,i}) = -0.06 + 0.13 \times e^{\left(0.07 \times T_{s,i}\right)} \qquad i = 1, 2 \ldots n, \tag{A28}$$

$$F_N^{\mathrm{acidity}}(pH_i) = 0.56 + \frac{\arctan(\pi \times 0.45 \times (-5 \times \mathrm{pH}_i))}{\pi} \qquad i = 1, 2 \ldots n, \tag{A29}$$

$$F_N^{\mathrm{moisture}}\left(\frac{S_i}{S_i^{\max}}\right) = \left(\frac{\left(\frac{S_i}{S_i^{\max}}\right) - N_b}{N_a - N_b}\right)^{\left(N_d \times \frac{N_b - N_a}{N_a - N_c}\right)} \times \left(\frac{\left(\frac{S_i}{S_i^{\max}}\right) - N_b}{N_a - N_c}\right)^{N_d} \qquad i = 1, 2 \ldots n, \tag{A30}$$

where $K_N^{\max}$ (d^{-1}) is the maximum nitrification rate determined by model calibration; and N_a, N_b, N_c, and N_d are the soil parameters set according to soil texture and described by *Parton et al.* [1996].

A1.4. Denitrification Rate

[59] Denitrification is the biological reduction of nitrate under anaerobic conditions. During denitrification, heterotrophic microbes contribute to the NO_3 reduction into NO_2, NO, and N_2O intermediates and ultimately into molecular nitrogen N_2 lost to the atmosphere. The denitrification process is controlled by environmental factors such as soil nitrate level, soil oxygen availability, and soil labile carbon availability (e$^-$ donor) [*Weier et al.*, 1993]. In VELMA, denitrification is simulated using the denitrification submodel of N_2 and N_2O production presented by *Parton et al.* [1996, 2001] and *Del Grosso et al.* [2000]. The rate of

denitrification is proportional to the amount of bioavailable SOC level. However, VELMA does not differentiate between labile and nonlabile soil organic matter. Therefore, simulated ecosystem CO_2 loss (soil heterotrophic respiration) is used as a proxy for the amount of bioavailable SOC [*Del Grosso et al.*, 2000; *Parton et al.*, 1996]. The rate of denitrification increases with decreasing oxygen availability. Oxygen availability is another critical factor not simulated by VELMA but assumed as a function of soil moisture, soil gas diffusivity, and oxygen demand. Gas diffusivity is simulated as a function of soil moisture and soil properties, whereas oxygen demand is a function of the simulated soil heterotrophic respiration [*Del Grosso et al.*, 2000; *Parton et al.*, 1996]. As a result, soil denitrification rate is simulated as a function of soil saturation $F_D^{\mathrm{moisture}}(S_i/S_i^{\max})$ (Equation (A1) [*Parton et al.*, 2001]), soil heterotrophic respiration $F_D^{\mathrm{carbon}}(CO_{2,i})$ (Figure 1d [*Del Grosso et al.*, 2000]), and soil available nitrate $F_D^{\mathrm{nitrate}}(\mathrm{NO}_{3i})$ (Figure 1c [*Del Grosso et al.*, 2000]). Currently, VELMA simulates the total denitrification or N_2+N_2O emission without the partition between N_2 and N_2O, such that

$$\mathrm{Den}_i = \min\left[F_D^{\mathrm{carbon}}(CO_{2i}), F_D^{\mathrm{nitrate}}(\mathrm{NO}_{3i})\right] \times F_D^{\mathrm{moisture}}\left(\frac{S_i}{S_i^{\max}}\right) \quad i = 1,2\ldots n, \tag{A31}$$

$$F_D^{\mathrm{moisture}}\left(\frac{S_i}{S_i^{\max}}\right) = 0.5 + \frac{\arctan\left(\pi \times 0.6 \times \left(0.1 \times \left(\frac{S_i}{S_i^{\max}}\right)\right) - \vartheta_a\right)}{\pi} \quad i = 1,2\ldots n, \tag{A32}$$

$$F_D^{\mathrm{carbon}}(CO_{2,i}) = 0.1 \times \left(\alpha \times c_d \times \mathrm{SOC}_i \times v_i(T_{s,i}, S_i)\right)^{1.3} \quad i = 1,2\ldots n, \tag{A33}$$

$$F_D^{\mathrm{nitrate}}(\mathrm{NO}_{3i}) = 0.005 \times (\mathrm{NO}_{3i})^{0.57} \quad i = 1,2\ldots n, \tag{A34}$$

where $F_D^{\mathrm{carbon}}(CO_{2,i})$ and $F_D^{\mathrm{nitrate}}(\mathrm{NO}_{3i})$ represent the maximum possible N gas flux from layer i for a given soil heterotrophic respiration rate and nitrate level, respectively (g Nm^{-2} d^{-1}); ϑ_a is a shape parameter that depends on soil texture; and $F_D^{\mathrm{moisture}}(S_i/S_i^{\max})$ varies between 0 and 1.

A2. Watershed Framework

[60] To place the earlier described soil column framework within a catchment framework, the catchment topography is gridded into a number of pixels, with each pixel consisting of one coupled soil column. Soil columns communicate with each other through the downslope lateral transport of water and nutrients. Surface and subsurface runoff are responsible for this lateral transport and link each soil column to the surrounding downslope soil columns. A multiple flow direction method is used where flow and nutrients from one pixel to its eight neighbors is fractionally allocated according to terrain slope [*Freeman*, 1991; *Quinn et al.*, 1991]. Moreover, nutrients transported downslope from one soil column to another can be processed through the different submodels in that downslope soil column, or continue to flow downslope, interacting with other soil columns, or ultimately discharging water and nutrients to the stream.

A2.1. Lateral Transport of Nutrients

[61] Lateral transport of nutrients from layer i of an upslope soil column to layer i of a downslope soil column or toward the stream is based on the flow routing information and on terrain slope. As with the vertical transport of nutrients, the lateral transport of nutrient is a function of the lateral runoff and the size of the nutrient pool in layer i. For simplicity, we assume that both surface runoff and layer 1 subsurface runoff impact the nutrient pool in layer 1 of the soil column:

$$\zeta_1(\mathrm{NH}_{4i}) = qf_{\mathrm{NH}_4} \times \frac{Q_i + q_{\mathrm{surf}} \times Q_s}{S_i} \times \mathrm{NH}_{4i} \quad i = 1,2\ldots n, \tag{A35}$$

$$\zeta_1(\mathrm{NO}_{3i}) = qf_{\mathrm{NO}_3} \times \frac{Q_i + q_{\mathrm{surf}} \times Q_s}{S_i} \times \mathrm{NO}_{3i} \quad i = 1,2\ldots n, \qquad ((\mathrm{A36})$$

$$\zeta_1(\mathrm{DON}_i) = qf_{\mathrm{DON}_i} \times \frac{Q_i + q_{\mathrm{surf}} \times Q_s}{S_i} \times \mathrm{DON}_i \quad i = 1,2\ldots n, \tag{A37}$$

$$\zeta_1(\mathrm{DOC}_i) = qf_{\mathrm{DOC}_i} \times \frac{Q_i + q_{\mathrm{surf}} \times Q_s}{S_i} \times \mathrm{DOC}_i \quad i = 1,2\ldots n, \tag{A38}$$

with $q_{\mathrm{surf}} = \begin{cases} 1 & \text{for} \quad i = 1 \\ 0 & \text{for} \quad i = 2,3\ldots n' \end{cases}$

where Q_i (mm d^{-1}) is the lateral subsurface runoff from layer i (Equation (A16) [*Abdelnour et al.*, 2011]; and Q_s (mm d^{-1}) is the surface runoff that impact the nutrients pools in layer 1 (Equation (A18) [*Abdelnour et al.*, 2011]).

Appendix B

[62] Appendix B describes the list of parameters used in the model. Table B1 includes the parameters definitions, values and references.

Table B1. Model Parameter Values Used to Simulate the Biogeochemical Processes of WS10 in the HJA

Parameters	Definition	Value	References
λ_{snow}	Seasonal damping depth for snow (mm)	670	*Hillel* [1998]
GST	Annual mean soil temperature (°C)	8.5	*Sollins and McCorison* [1981]
Kn	Michaelis-Menten calibration parameter (g Nm^{-2})	0.1	Calibrated
δ_{NO_3}	Fraction of nitrogen uptake from the nitrate pool	0.3	*Rygiewicz and Bledsoe* [1986], *Kamminga-van Wijk and Prins* [1993]
δ_{NH_4}	Fraction of nitrogen uptake from the ammonium pool	0.7	*Rygiewicz and Bledsoe* [1986], *Kamminga-van Wijk and Prins* [1993]
α_{CN}	C:N ratio for plants and soils	50	*Sollins and McCorison* [1981]
c_d		0.004	Calibrated

Table B1. (continued)

Parameters	Definition	Value	References
	Fraction of carbon that is not lost to the atmosphere due to the soil heterotrophic respiration		
$\overline{\pi_N}$	Annual wet and dry deposition of atmospheric N (g Nm^{-2} yr^{-1})	0.2	*Sollins et al.* [1980]
β	Fitted extinction coefficient	0.976	*Jackson et al.* [1996]
q	Fraction of carbon decomposition that feeds into the DON pool	0.015	Calibrated
m_{ss}	Steady-state average mortality rate of old-growth forest (yr^{-1})	0.0125	*Lutz and Halpern* [2006]
B_{ss}	Average biomass value for an old-growth forest (g Nm^{-2})	42,350	*Harmon et al.* [2004], *Sollins et al.* [1980]
m_a	Mortality rate parameter	1.55	Fixed a priori
m_b	Mortality rate parameter	4	Fixed a priori
m_c	Mortality rate parameter	1E−14	Fixed a priori
μ_{min}	Minimum uptake rate of vegetation after disturbance (yr^{-1})	0.20	Fixed a priori
μ_{ss}	Steady state value of plant uptake (yr^{-1})	0.25	Calibrated
S_{age}^{max}	Stand age for which plant uptake is the highest (years)	35	*Luyssaert et al.* [2008]
W_{k1}	Weibull distribution parameter for plant uptake	1.60	Fixed a priori
$W_{\lambda1}$	Weibull distribution parameter for plant uptake	1.70	Fixed a priori
W_{k2}	Weibull distribution parameter for plant uptake	0.95	Fixed a priori
$W_{\lambda2}$	Weibull distribution parameter for plant uptake	0.85	Fixed a priori
WS^{min}	Lower limit of soil water saturation, below which plant uptake is reduced (%)	40	Fixed a priori
WS^{max}	Upper limit of soil water saturation, above which plant uptake is reduced (%)	80	Fixed a priori
k_c	Potential carbon decomposition rate (vegetation dependent; yr^{-1})	0.45	Calibrated
M_{sat}	Parameter that determines the value $F_N^{moisture}(s_i/s_i^{max})$	0.25	*Raich et al.* [1991]
ma_1	Shape parameter defining the skewness of the $F_N^{moisture}(s_i/s_i^{max})$ curve	0.14	*Raich et al.* [1991]
$\left({}^{s}/{}_{s}{}^{max}\right)_o$	Optimal soil wetness for which decomposition is maximal	40%	*Alexander* [1977]
pH	Average pH value for the soils in WS10	4.5	*Chaer et al.* [2009]
K_N^{max}	Maximum nitrification rate (d^{-1})	0.15	*Parton et al.* [2001]
N_a	Soil moisture function parameter for ammonium nitrification	0.4	*Parton et al.* [1996]
N_b	Soil moisture function parameter for ammonium nitrification	1.7	*Parton et al.* [1996]
N_c	Soil moisture function parameter for ammonium nitrification	3.22	*Parton et al.* [1996]
N_d	Soil moisture function parameter for ammonium nitrification	0.007	*Parton et al.* [1996]
ϑ_a	Soil moisture function shape parameter	5.0	*Del Grosso et al.* [2000]
qf_{NH_4}	Maximum fraction of NH_4 pool that can be lost through transport of water	0.12	Calibrated
qf_{NO_3}	Maximum fraction of NH_4 pool that can be lost through transport of water	0.04	Calibrated
qf_{DON}	Maximum fraction of NH_4 pool that can be lost through transport of water	0.02	Calibrated
qf_{DOC}	Maximum fraction of NH_4 pool that can be lost through transport of water	0.06	Calibrated

[63] **Acknowledgments.** The information in this document has been funded in part by the U.S. Environmental Protection Agency. It has been subjected to the agency's peer and administrative review, and it has been approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use. This research was additionally supported in part by the following NSF grants 0439620, 0436118, and 0922100. We thank Sherri Johnson, Barbara Bond, Suzanne Remillard, Theresa Valentine, and Don Henshaw for invaluable assistance in accessing and interpreting various HJ. Andrews LTER data sets used in this study. Sherri Johnson also provided helpful comments on an earlier draft. Data for streamflow, stream chemistry, and climate were provided by the HJA Research Program, funded by the National Science Foundation's Long-Term Ecological Research Program (DEB 08-23380), U.S. Forest Service Pacific Northwest Research Station, and Oregon State University.

References

Abdelnour, A., M. Stieglitz, F. Pan, and R. McKane (2011), Catchment hydrological responses to forest harvest amount and spatial pattern, *Water Resour. Res.*, *47*, W09521, doi:10.1029/2010WR010165.

Acker, S., C. Halpern, M. Harmon, and C. Dyrness (2002), Trends in bole biomass accumulation, net primary production and tree mortality in Pseudotsuga menziesii forests of contrasting age, *Tree Physiol.*, *22*(2–3), 213.

Agee, J. K. (1990), The historical role of fire in Pacific Northwest forests, edited by J. D.Walstad, S. R. Radosevich, and D. V. Sandberg, Natural and prescribed fire in Pacific Northwest forests, Oregon State University Press, Corvallis, Oregon, USA.

Agee, J. (1994), Fire and weather disturbances in terrestrial ecosystems of the eastern Cascades, Gen. Tech. Rep. PNW-GTR-320, U.S. For. Serv., Pac. Northwest Res. *Stn, Portland, Oreg*.

Alexander, M. (1977), *Introduction to Soil Microbiology*, 2nd ed., Wiley, New York.

Alila, Y., and J. Beckers (2001), Using numerical modelling to address hydrologic forest management issues in British Columbia, Hydrol. *Processes, 15*(18), 3371–3387.

Amaranthus, M., H. Jubas, and D. Arthur (1989), Stream shading, summer streamflow and maximum water temperature following intense wildfire in headwater streams, in *Proceedings of the Symposium on Fire and Watershed Management, Sacramento, California*, edited by N. H. Berg, Gen. Tech. Rep. PSW-109, pp. 75–78, U.S. Dept. of Agric., For. Serv., Pac. Southwest For. and Range Exp. Stn., Berkeley, Calif., 26–28 Oct. 1988.

Bernot, M., and W. Dodds (2005), Nitrogen retention, removal, and saturation in lotic ecosystems, *Ecosystems, 8*(4), 442–453.

Beschta, R. L. (1990), *Effects of fire on water quantity and quality, in Natural and Prescribed Fire in Pacific Northwest Forests*, edited by J. D. Walstad, S. R. Radosevich, and D. V. Sandberg, pp. 219–232, Oreg. State Univ. Press, Corvallis, Oreg.

Beschta, R., M. Pyles, A. Skaugset, and C. Surfleet (2000), Peakflow responses to forest practices in the western Cascades of Oregon, USA, *J. Hydrol., 233*(1–4), 102–120.

Bilby, R., and G. Likens (1980), Importance of organic debris dams in the structure and function of stream ecosystems, *Ecology, 61*(5), 1107–1113.

Binkley, D., P. Sollins, R. Bell, D. Sachs, and D. Myrold (1992), Biogeochemistry of adjacent conifer and alder-conifer stands, *Ecology, 73*(6), 2022–2033.

Bormann, F., G. Likens, D. Fisher, and R. Pierce (1968), Nutrient loss accelerated by clear-cutting of a forest ecosystem, *Science, 159*(3817), 882.

Bosch, J., and J. Hewlett (1982), A review of catchment experiments to determine the effect of vegetation changes on water yield and evapotranspiration, *J. Hydrol., 55*(1–4), 3–23.

Cairns, M., and K. Lajtha (2005), Effects of succession on nitrogen export in the west-central Cascades, Oregon, *Ecosystems, 8*(5), 583–601.

Appendix 7: Creating Flat-Processed DEM Data For The VELMA Simulator and Determining Outlet and Watershed Delineation

The VELMA Simulator requires a Digital Elevation Map (DEM) for any simulation it runs. This DEM contains elevation data for the area simulated.

The area specified must contain no "sinks" or "flat spots".

When VELMA simulates surface waterflow it must be able to flow "off" the map area in some way. Water is not allowed to pool anywhere within the simulation area.

In order to ensure this "no flat spots rule", DEM data used by the VELMA Simulator is first "flat-processed" to remove any sinks or flat spots.

Use JPDEM to prepare DEM data for use by the VELMA Simulator.

NOTE
You must have Java 1.7 installed on your computer to run JPDEM.

I. Creating a Flat-Processed DEM

Start JPDEM

Open a Windows Command Line and enter the command:

```
C:\> java -Xmx1024m -jar C:\full\path\to\JPDEM.jar
```

The `"-Xmx1024m"` tells the Java Runtime Environment (JRE) to let JPDEM have 1024 megabytes (1 gigabyte) of working memory.

If your machine doesn't have that much memory, substitute a smaller value (e.g. "512" instead of "1024"). Alternately, if the .asc file of DEM data you intend to load is very large (and your machine has the memory) you may substitute a larger value (e.g. "2048" instead of "1024").
If the full path to the JPDEM.jar file contains whitespace characters, place double-quotes around the path.
Example:

```
"C:\full\path\with white space\to\JPDEM.jar"
```

Loading DEM Data Into JPDEM

Load a "raw" DEM file into JPDEM (the file must be in standard ESRI Grid .asc format).
Click the "File" menu, then select and click "Load DEM File".
The "Select DEM File to Load" dialog window opens.

Browse to the location of your DEM file, and click the name of the DEM file.
The DEM file's name appears in the "File Name:" text box.
Click "Load File" to load that file's data into JPDEM.
The file is loaded, and an image of the file appears in JPDEM's map display.
If the file is too small (has few rows and columns), click the Image Scale up-arrow button to "zoom in".

Flat Processing DEM Data In JPDEM

Click the "Tools" menu, then select and click "Flat-Process DEM Data (Standard)".
JPDEM will process the DEM, adjusting elevation data to remove flat spots or sinks within the DEM data's area.
If the processing takes more than a second or two, the display will periodically update, showing remaining flat spots in blue.
When the flat-processing algorithm finishes, the display shifts to a grey-scale representation, with darker-colored pixels representing cells with higher flow-accumulation values.
Note:
There are several menu options for Flat-Processing DEM data.
The "(Standard)" menu item uses an algorithm that produces good results, but generally takes more time to produce them. We do not, at this time, recommend using the other two flat-processing algorithms. They are still being developed.

Saving Flat-Processed DEM Data From JPDEM To a File

After JPDEM finishes flat-processes DEM data, click the "File" menu, then click the "Save DEM As ..." submenu, and finally select and click the "Ascii Grid" menu item.
The "Save DEM Grid to File" dialog window opens.
Browse to the location you wish to save the data in, then type a name for the file in the "File Name:" text box.
Finally, click the "Save File" button to save the DEM data to the specified file name and location.
Note
Flat-processed DEM data may be loaded into JDPEM just like "raw" DEM data.
Flat-processed DEM date does not need to be flat-processed again after it is loaded into JPDEM. However, to view the flow-accumulation grey-scale visualization of that flat-processed data, you must determine the flow data information for the flat-processed DEM values.

II. Watershed Delineation

Determining Flow Data For a Flat-Processed DEM

Load a flat-processed DEM .asc file into JPDEM, in the same way you would load a "raw" DEM file.
After loading the DEM data, click the "Tools" menu, then select and click the "Determine DEM Flow Data (Standard)" menu item.
JPDEM will then compute flow data (flow direction and accumulation values) for each cell in the DEM area.

Once the flow data is computed, JPDEM will shift the display to the flow-accumulation grey-scale visualization.

Note

Loading a "raw" DEM file and then immediately trying to determine DEM flow data should not crash JPDEM, but it won't result in any meaningful information either.

Note

The difference in results between the "(Standard)" and "(Alternate)" flow data algorithms should be zero or very small.

You may use either algorithm. The Alternate algorithm is somewhat faster, but often not much faster than the Standard algorithm.

Determining Watershed Delineation

Flat-processed, flow-accumulated DEM data in JDEM may be used to determine the delineation relative to a specified outlet cell (i.e. the set of cells that flow "to" a specified "outlet" cell).

In order to compute the delineation for an outlet cell, you must first specify the coordinates of the outlet cell.

You can do this either by entering the zero-based x (column) and y (row) coordinates of the outlet cell in the "Outlet X=" and "Y =" number boxes, above JPDEM's map display, or by using the mouse.

To select an outlet cell using the mouse, do the following:

Move the mouse pointer over the cell you wish to select.

Press down on either the right or left mouse button.

A context menu should pop-up, containing information about the cell you are pointing at.

Holding the mouse button down, drag the mouse pointer onto the menu item containing cell info (the info should highlight), then release the mouse button.

The menu goes away, and the x and y coordinates for the cell you selected appear in the "Outlet X = " and "Y = " number boxes above the map display.

Note:

"Zoom in" on the map display by clicking the up-arrow button of the "Image Scale" to make selecting a specific cell easier. For example, at "Image Scale 3", for example, every cell is displayed by 3x3 pixels.

However, on very large maps zooming too far in can crash the display, because it may run out of memory.

Once the coordinates of an outlet cell are specified, click the "Tools" menu, then select and click the "Delineate DEM Data (Experimental)" menu item.

JPDEM will then compute the delineation (i.e. the area of cells) for the currently specified "Outlet X =" and "Y =" values.

The resulting area is displayed in a logarithmic-coloring blue-red scale, superimposed over the grey-scale of the flow-accumulation display.

If the delineation looks incorrect, change the outlet cell coordinates and re-run the delineation.

When the delineation looks correct, record the "Outlet X =" and "Y =" values for future reference and use with JVELMA. (JPDEM does not "record" outlet values anywhere.)

Note

There are currently four different JPDEM delineation algorithm implementations to choose from.

We believe they all produce the same results (except possibly for cells on the edge of the DEM area -- which is an acceptable difference). However, they vary greatly in how long they take to run. Use the "(Experimental)" algorithm for delineation: it runs significantly faster than any of the other delineation implementations. In particular, ***do not*** use the "(PDEM-2)" algorithm, which can take a very long time to run, even for small maps.