Getting started in RHESSys

Written by: Dexter Howard 17 Nov 2025

Table of Contents

Resources and Readings

Papers

- Tague and Band 2004. RHESSys: Regional Hydro-Ecologic Simulation System—An Object Oriented Approach to Spatially Distributed Modeling of Carbon, Water, and Nutrient Cycling
    - fiesta.bren.ucsb.edu/~rhessys/about/publications/RHESSysTagueEA2004.pdf

RHESSys resources

- Original website (seems archived to github though)
    - fiesta.bren.ucsb.edu/~rhessys/setup/downloads/downloads.html
- **New website:** https://rhessys.github.io/
- **Wiki**: https://github.com/RHESSys/RHESSys/wiki
- **Docker instructions**:
    - https://github.com/RHESSys/RHESSys/wiki/Docker
- rhessys on windows:

    - https://github.com/RHESSys/RHESSys/wiki/RHESSys-on-Windows
- R package github:
    - https://github.com/RHESSys/RHESSysIOinR/tree/main/R
- Training workshop from 2021
    - https://github.com/ryanrbart/rhessys_training_2021/tree/main
- Book (not complete)
    - https://ryanrbart.com/rhessys_book/
- Test case in main github:
    - https://github.com/RHESSys/RHESSys/blob/trunk/Testing/TestCase.Rmd
- Interpreting output resources
    - Github: https://github.com/RHESSys/RHESSys/wiki/RHESSys-Output
    - OG website: fiesta.bren.ucsb.edu/~rhessys/appendices/appendix_c_output/appendixc.html
    - Flow unit conversions
        - https://github.com/ryanrbart/EcoHydroConversions/blob/main/R/cfs_to_mm.R
        - with some proof of concepts
            - https://github.com/ryanrbart/rhessys_training_2021/tree/main/data/streamflow

Getting RHESSys locally

- Clone or fork the tutorial R project. This is where you will keep scripts, parameter and configuration files, maps, and model outputs
  - https://github.com/dwh77/RHESSys_Tutorial
- Set up a docker image of the RHESSys model. This Docker container contains a pre-compiled RHESSys model
  - Create a Docker account and install if you haven't
    - install: https://docs.docker.com/get-started/get-docker/
  - Start the Docker application - Docker desktop
  - Open a shell window (powershell on windows)
  - Pull the RHESSys Docker image from the command line using code below
    - docker pull rhessys/rhessys:develop
  - In Docker Desktop you should now see under images 'rhessys/rhessys'
  - You can follow the first 5 steps of the instructions here for more information
    - https://github.com/RHESSys/RHESSys/wiki/Docker
- Mount docker image to github repo.
  - This allows you to access files and inputs in your working repo through the docker container which contains the model.
  - Run this code below in the powershell
    - docker run -v C:/Users/dwh18/OneDrive/Desktop/R_projects/RHESSys_Tutorial:/RHESSys/repo -it rhessys/rhessys:develop
    - You will have to update 'C:/Users/dwh18/OneDrive/Desktop/R_projects/' To the file path you have stored the RHESSys_Tutorial project
    - The ':/RHESSys/repo' is so that your repo is now mounted within the RHESSys container and named 'repo'
  - To check this worked after running 'docker run -v….'
    - you should go from seeing the command line showing something like 'PS C:\Users\dwh18>' to 'root@4fcf82f7500a:/RHESSys#'
    - If you see root…RHESSys# you can then check that your repo was successfully mounted.
    - Check location and mounting
      - 'ls' : lists file in repo
      - should see both 'rhessys7.5' and 'repo'
      - 'cd repo/' will bring you to your R project repo
        - in here you should see all the folders and files within your repository
      - 'cd /RHESSys' would bring me back to original container point

Input Data

There are two main types of input data needed to get RHESSys running. Spatial landcover data and climate time series data. The next two sections will walk through how to compile this data.

Climate Data

- Using daily Meteo from ROA weather station
    - https://www.ncei.noaa.gov/cdo-web/datasets/LCD/stations/WBAN:13741/detail
    - add data to cart, select daily outputs, NOAA will email you a confirmation you can then download a csv
- Had to match RHESSys format using Roanoke meto data
    - DWH made function that does this in preprocessing scripts
        - ‘ClimateData_preprocess.R’
    - see ‘format for time series data inputs’
        - https://github.com/RHESSys/RHESSys/wiki/Climate-Inputs
    - An R function from them
        - https://github.com/RHESSys/RHESSysIOinR/blob/main/R/IOin_clim.R
    - and function for reading in their style formatted data
        - https://github.com/RHESSys/RHESSysIOinR/wiki/Climate

Spatial Data

In order to run RHESSys you need to have generated several maps to make the world files and flowtables. This section will walk through how to make this maps using ArcGIS but mainly GRASS GIS, following guidance from the RHESSys developers. ArcGIS is used in sections ‘Getting watershed shapefiles’ and ‘How to get DEM data’, but GRASS is used for the rest of tasks. ArcGIS was largely used for changing projections, as this was the software DWH had the most familiarity with, but all of these changes could be done in GRASS or QGIS.

Getting watershed shapefiles

- Use USGS streamstats tool: https://streamstats.usgs.gov/ss/
    - Search for CCR, follow instructions in left side panel
    - use the delineate tool to click on the point of interest. CCR watershed is already on EDI, but can use this to get subbasin watershed areas
- Note that USGS shapefiles from streamstats are in projection ‘WGS 1984’
    - can check in arcGIS by: right-click the shapefile in the Contents pane or Catalog pane, select Properties, and navigate to the Source tab
- To reproject shapefiles to match DEM formation (NAD 1983)
    - Use tool ‘Project (Data Management Tools)’
        - Input: watershed shapefile
        - name the output

        - Output coordinate system: can click on DEM raster or input ‘GCS_North_American_1983’
        - Geographic Transformation: should fill out itself to say: ‘WGS_1984_(ITRF00)_To_NAD_1983’
    - to save shapefile
        - right click new feature
        - Data - export features
            - make sure input is right file
            - for export path - navigate to folder where you want data, I added specific folder name to clarify its NAD83 data
    - To get to UTM (which is preferred RHESSys projection and lets you look at values in square meters instead of square degrees in the GIS programs)
        - for output coordinate systems just use: NAD_1983_UTM_Zone_17N
        - and name output ‘watershed_UTM’ and store in proper folder
        - to check area of shape: right click shape in contents pane - attribute table

How to get DEM data

- DEMs from USGS
    - RHESSys info: https://github.com/RHESSys/RHESSys/wiki/Data-Sources
    - actual data: https://apps.nationalmap.gov/downloader/
        - data description info: https://apps.nationalmap.gov/datasets/
    - Getting data from actual data link:
        - Under data select: ‘Elevation Products (3D Elevation Program Products and Services)’
        - Then click extent button and draw area around CCR watershed
    - Looks like we need two different tiffs to get whole CCR watershed for ⅓ second or 1 second data
        - download ⅓ second data for now
        - data set names: ‘USGS 1/3 Arc Second n38w080 20211229’ AND ‘USGS 1/3 Arc Second n38w081 20230816’
- **Merging two CCR USGS DEM files**
    - In arcGIS Pro: use Geoprocessing tool box - “Mosaic to New Raster’
            - select two USGS tif’s
            - output location: ‘USGS_DEM’
            - name raster
            - pixel type = 32 bit unsigned
            - mosaic operator = Blend
            - Mosaic colormap mode = Match
            - all other blank
        - This gives you the entire region of the two DEM products from USGS as one merged file
        - This file is named: ‘joined_usgs_dem2.tif’
    - To trim to general area of CCR watershed area

        - You could trim to exact watershed but I'm finding edge effect issues in GRASS when doing this
            - read in shapefile of ccr watershed in NAD83 projection
        - geoprocessing tool - "Extract by Mask"
            - Input Raster - select DEM (joined_usgs_dem2.tif)
            - Feature mask: click pencil icon and draw rectangle outside of CCR watershed area
                - or could have choosen CCR watershed shapefile
            - output raster: pick a name
        - This gives you the DEM trimmed to the CCR general watershed area
    - To reproject to UTM 17N from NAD 83
        - Use tool 'Project Raster (Data Management Tools)'
            - Input: dem that is trimmed to ccr watershed
            - name the output
            - Output coordinate system: 'NAD_1983_UTM_Zone_17N'
        - Now you have a DEM of CCR in projection NAD 1983 UTM 17N
        - Export this raster as a tif
            - right click raster - data - export raster
            - select output folder and give relevant name (ie., usgs_dem_ccr_UTM.tiff)

Creating input maps in GRASS GIS

GRASS GIS was used as this software is both open source and the GIS used by RHESSys developers, in turn having the most resources to follow.

Downloading GRASS

- Getting GRASS, DWH is using version 7.8.8
    - https://grass.osgeo.org/download/windows/

Launch and setup GRASS session

- Open GRASS Software icon
- Set database directory link: I've just been making a folder in my main repo folder
    - ex. RHESSys Tutorial
- Following instructions are based on steps here from RHESSys github
    - see "reproject DEM in GRASS" for how to start project at the correct projection for the DEM
        - https://github.com/RHESSys/RHESSys/wiki/Troubleshooting#reproject-dem-in-grass-gis
- for '2.Select GRASS location' select 'New' button
    - name project: "GRASSccr" or "GRASShbp"
    - leave location empty
    - click next (don't select either box)
    - In Window 'Choose method for creating a new location':

        - Read projection and datum terms from a georeferenced data file
        - Click Next
    - Window 'Select georeferenced file':
        - browse to directory/file (of the georeferenced file, i.e. the usgs_dem_ccrWSbox_UTM.tif)
        - Click Next
    - Window 'Summary': click Finish
    - Window 'Import data?': click yes
        - this will have the dem loaded in data
    - click start GRASS
        - select GRASS location should be the location you just mapped
        - 3. should be “PERMANENT”
- Now you’re in a GRASS GIS session and can begin creating needed files to run RHESSys

Create needed input maps for RHESSys

Starting the GRASS session above will have imported the DEM as well. In the data tab you can right click on the raster and select display. But you’ll see there’s shading outside the raster area.

Mask project to area of interest

- The steps above have already added the joined DEM to the GRASS data tab
- Read in the watershed shapefile of interest this will be used to mask all calculations to our area of interest
    - first read in the shapefile of the watershed
    - File - import vector data - data of common format (v.in.org)
        - select file location of shapefile and select ‘.shp’ file
- Next convert the vector to a raster (this is helpful since we need a raster of the basin area anyways for RHESSys inputs)
    - ‘Vector’ tab - ‘Map type conversions’ - ‘Vector to raster’ (v.to.rast)
    - input name = name of vector (i.e., hpb_ws@PERMANENT)
    - name for output = ‘hpb_ws_raster
    - source = ‘cat’
    - how command line looks for this
        - v.to.rast = hpb_ws@PERMANENT output = hpb_ws_raster use = cat
- Set Region and Mask: To Export rasters and avoid having lots of NA areas outside of watershed region from original DEM
    - set region: g.region -p vector=hpb_ws@PERMANENT
        - Settings - computational region - set region
        - This allows all exports to be of just the region of interest
    - Mask region
        - Raster - Mask (r.mask)
        - select watershed raster as input
            - r.mask raster=hpb_ws_raster@PERMANENT
        - This sets a mask for displaying the output maps

- Now export this raster of the watershed since we have it now
    - file - export raster map - r.out.gdal
        - name of raster map to export = the raster you just made (ie. hpb_ws_raster@PERMANENT)
        - name for output raster file: Click Browse and navigate to the file path you want to store these (I'm making a GRASS outputs folder with folders for each basin)
            - Give informative name and be sure to add .tiff at end! (ie. hpb_watershed.tif)
        - raster data format = GTiff
        - command line
            - r.out.gdal input=hpb_ws_raster@PERMANENT output=C:\Users\dwh18\OneDrive\Desktop\R_Projects\CCR_RHESSys\GRASS_outputs\HPB\hpb_watershed.tiff format=GTiff
- Export the DEM trimmed to your watershed of interest since the mask is set as well
    - following steps in previous step but using 'usgs_dem_ccr_UTM' as the input and naming the output 'hpb_dem'

Make other needed maps for RHESSys (slope, aspect, etc)

Spatial input needs: https://github.com/RHESSys/RHESSys/wiki/Spatial-Input-Requirements
RHESSys GRASS guides:
https://github.com/RHESSys/RHESSys/wiki/grass-tutorial-for-creating-necessary-RHESSys-maps

Slope and aspect maps

- Raster - Terrain Analysis - Slope and Aspect (opens r.slope.aspect window)
    - select dem, in outputs tab set names for slope and aspect maps
        - r.slope.aspect elevation=usgs_dem_ccrWSbox_UTM@PERMANENT slope=slope aspect=aspect
    - Then export these maps as rasters like above
- check for NAs in maps
    - This was an issue I ran into when generating world files before
    - How to check for NAs
        - Raster - Reports and Stats - r.univar
            - input as raster of interest
    - Raster - interpolate surfaces - Fill NULL values 'r.fillnulls'
        - r.fillnulls input=slope@PERMANENT output=slope_fillnull method=rst
        - if rst doesn't work can try billinear

East and West Horizons

- Raster - Terrain Analysis - Horizon Angle (r.horizon)
    - see horizon map info here: https://github.com/RHESSys/RHESSys/wiki/East-and-West-Horizon-Maps

        - NOTE: I’m using GRASS v7.8.8 so horizon output is already in radians, so don’t need to do conversion steps
    - r.horizon elevation=usgs_dem_ccrWSbox_UTM@PERMANENT direction=0 output=east
    - r.horizon elevation=usgs_dem_ccrWSbox_UTM@PERMANENT direction=180 output=west
    - Then export these two rasters like above

Patches, stream network, and dividing watershed maps
Watershed basins and stream network

- Raster - Hydrological Modeling - Watershed Analysis (r.watershed)
    - input = DEM (usgs_dem_ccr_UTM)
    - min size of exterior watershed basin (threshold = integer)
        - will likely have to tinker with this, starting with 5000
    - then name outputs for: accumulation, drainage, basin, stream, half_basin
        - ie. basin_5000
    - To get stream raster that only shows stream path as one color
        - r.mapcalc expression=streams_5000_1 = (stream_5000@PERMANENT >0)*1
            - Note the 5000 is just the naming convention I used when threshold was 5000
    - check for NAs in maps like before using r.univar
        - likely NAs in both basin and half-basin maps
        - but using r.fillnulls will make interpolate linearly and not allow for categorical values
        - Use r.neighbors for this: raster - neighborhood analysis - moving windows
            - r.neighbors input=basin_5000@PERMANENT output=basin_5000_neighbors7 method=mode size=7
            - took some tweaking on size (3 is default but didn’t fill all nulls)
        - to check that values didn’t become continuous r.stats with raster as input
            - should just be 2,4,6 or more integers but no groups with decimals
    - export , streams1, basin, and halfbasins

- Exporting files NA notes
    - For different map exports they NA values outside of watershed boundary are categorized differently, keeping record here
        - hpb_watershed, streams = 255
        - DEM = 65535
        - slope, aspect, west and east horizons, basin and halfbasin fillnulls = nan

Generate worldfiles and flowtables

To run RHESSys you need to generate several configuration files for the spatial hierarchy to work. Using the maps generated above move to the R repo in script HPB_RHESSys_PreProcess.Rmd

- There are several configuration files needed to run RHESSys including
    - Template
    - Worldfile
    - Flowtables
    - Header
    - def files
    - Tec files
    - climate files
- The Template will need to be made first in order to generate worldfile and flowtables
    - There is currently a template in the tutorial, which was adapted from the RHESSys testing code.
        - more info on templates can be found here: https://github.com/RHESSys/RHESSys/wiki/Template
- Next you will need to run 'HPB_RHESSys_PreProcess.Rmd' to generate the worldfile and flowtables
    - This script sets up the needed inputs for a RHESSys function to generate the world and flow files
    - You need to be careful about setting appropriate file paths here
    - You'll need to make sure all the correct maps are both available to this script and referenced in the template above.
    - You may find when running this script that some rasters have NAs, if so, go back to the GRASS instructions for how to fill those
- Additionally a header will be needed to run the model. This was adapted from testing as well. https://github.com/RHESSys/RHESSys/wiki/Worldfile-Header
    - This is used to point the model to various parameter files and climate data
    - The different def files referenced in the header are used to set parameter values
        - More guidance on parameters can be found here:
            - https://github.com/RHESSys/RHESSys/wiki/Parameter-Default-Files
            - https://github.com/RHESSys/ParameterLibrary
- Tec files, control the temporal events in the model. Main function of these currently is to set when model starts.
    - but these can also be used to determine model output filtering and when events happen
- Climate files is a folder that contains needed climate inputs

Run RHESSys

Mount repo to docker

- Code to run in powershell

    -
      ```
      docker run -v C:/Users/dwh18/OneDrive/Desktop/R_projects/RHESSys_Tutorial:/RHESSys/repo -it rhessys/rhessys:develop
      ```

- Check location and mounting
    - ‘ls’ : lists file in repo
        - should see both ‘rhessys7.5’ and ‘repo’
    - ‘cd repo/’ will bring you to your R project repo
    - ‘cd /RHESSys’ would bring me back to original container point

Launch R and run model

- ‘R’ to open an R session within the docker container in powershell
- Navigate to the runHPB.Rmd script in your local R project
    - from here follow the instructions but in short:
        - you’ll set the working directory in the container and run a few lines of code
        - The model will output the results in the ‘out’ folder of the repo
        - You can then interact with these outputs within the Rmd script
-