

Macrosystems EDDIE:

Climate Change Effects on Lake Temperatures

Instructor's Manual

Module Description

Climate change is modifying the thermal structure of lakes around the globe. In this module, students will learn how to use a lake model to explore the effects of altered weather on lakes, and then develop their own climate scenarios to test hypotheses about how lakes may change in the future. Once the students have mastered running one climate scenario for their lake, they will learn how to use distributed computing software to scale up and run hundreds of different climate scenarios for their lakes. The overarching goal of this module is for students to explore new modeling and computing tools while learning fundamental concepts about how climate change will affect lakes.

Pedagogical Connections

Phase	Functions	Examples from this module
Engagement	Introduce topic, gauge students' preconceptions, call up students' schemata	Short introductory lecture
Exploration	Engage students in inquiry, scientific discourse, evidence-based reasoning	Development of hypotheses of how climate change affects lakes; testing of these hypotheses by forcing lake models with climate scenarios to see how the lakes respond
Explanation	Engage students in scientific discourse, evidence-based reasoning	In-class discussion of the effects of the different climate scenarios
Expansion	Broaden students' schemata to account for more observations	Using the GRAPLEr software to create hundreds of different climate scenarios
Evaluation	Evaluate students' understanding, using formative and summative assessments	In-class discussion of how climate change can affect lake thermal structure

This module was initially developed by: Carey, C.C., S. Aditya, K. Subratie, R. Figueiredo, and K.J. Farrell. 30 June 2017.
 Macrosystems EDDIE: Climate Change Effects on Lake Temperatures. Macrosystems EDDIE Module 1, Version 1.
<http://module1.macrosystemseddie.org>. Module development was supported by NSF DEB 1245707, ACI 1234983, & EF 1702506.

This document last modified: 19 December 2018 17 December 2018 by K.J. Farrell.

Learning Objectives

By the end of this module, students will be able to:

- Set up and run the General Lake Model (GLM) in the R statistical environment to simulate lake thermal structure.
- Understand how GLM configuration files, high-frequency driver data, and output files are organized and used.
- Modify the input meteorological data for one GLM model to simulate the effects of different climate scenarios on lake thermal structure.
- Interpret model output from GLM simulations to understand how changing climate will alter lake thermal characteristics.
- Use the GRAPLEr R package to set up hundreds of model simulations that vary input meteorological data, and run those simulations using distributed computing.
- Explore the application of distributed computing for modeling climate change effects on lakes.

How to Use this Module

This entire module can be completed in one 3-4 hour lab period or three 60-minute lecture periods for senior undergraduate students or graduate students. Activities A and B could be completed with upper level students in two 60-minute lecture periods, with Activity C as a separate add-on activity. We found that teaching this module in one longer lab section with short breaks was more conducive for introductory students than multiple 1-hour lecture periods.

Depending on available class time, you can ask the students to develop their own climate scenarios and modify the driver files for each accordingly. However, this may be quite time-consuming depending on the students' ability level, so if time is limited, we include three pre-made scenarios based on future conditions derived from downscaled climate scenarios that students could select from and run.

Pre-made scenario files are named as follows:

Climate:

- 1) met_hourly_plus2.csv: A climate warming scenario in which air temperatures are +2°C above baseline historical conditions year-round
- 2) met_hourly_plus4.csv: A climate warming scenario in which air temperatures are +4°C above baseline historical conditions year-round
- 3) met_hourly_plus6.csv: A climate warming scenario in which air temperatures are +6°C above baseline historical conditions year-round

Quick overview of the activities in this module

- Activity A: Plotting water temperatures from a lake model
- Activity B: Develop a climate scenario, generate hypotheses, and model how the lake responds
- Activity C: Using distributed computing to run hundreds of lake model simulations

Module Workflow

1. Have students install R and RStudio software on their laptops before class (send them “R You Ready for EDDIE” file for step-by-step directions).
2. Give students their handout ahead of time to read over prior to class, or distribute handouts when they arrive to class.
3. Instructor gives brief PowerPoint presentation on climate change effects on the thermal structure of lakes, an overview of the GLM model, and the GRAPLEr software.
4. After the presentation, the students divide into teams, set up the GLM files and R packages on their computer to run a default lake model and explore the output (Activity A).
5. The instructor then introduces Activity B. Indicate to students whether they should develop their own climate scenario or use one of the pre-made options.
6. The students create hypotheses about how aspects of climate change may affect lakes (e.g., altered precipitation), develop a climate change scenario for their model lake to test their hypotheses, force a model lake with their scenario, and analyze the output to determine how their scenario alters lake thermal structure (Activity B).
7. After the students have analyzed the model output, they create some figures with their partners to present their model simulation and output to the rest of the class.
8. The instructor then moderates a discussion of the scenarios and output presented in Activity B and introduces Activity C.
9. The students go through a demonstration of the GRAPLEr R package and then design and carry out their own simulation ‘experiment’ with their partners. If time permits, the students create additional figures from their experiment results and share them with the class, with the instructor moderating the discussion (Activity C).

Important Note to Instructors:

All R packages used in this module are continually being updated, so these module instructions will periodically change to account for changes in the code. If you find any errors or have other feedback about this module, please contact the module developers (see “We’d love your feedback” below). Visit our website: <http://graple.org/> for the most recent version of the R packages for this module.

Things to do prior to starting the instructor’s presentation

- Make sure that all students have downloaded R and RStudio successfully on their laptops (see “R You Ready for EDDIE?” file for step-by-step directions).
- While checking to make sure that everyone has R downloaded, have students that are ready and waiting type in some basic commands into the R interface (e.g., ‘2+2’) to explore its capabilities.
- Organize student pairs by operating system, such that Windows PC users are working together, and OS X Macintosh users are working together. It also helps for students to work with partners that have the same version of operating system (e.g., El Capitan vs. Yosemite vs. Sierra Mac OSX users; Windows 7 vs. Windows 8 vs. Windows 10 PC users), though it’s not necessary.
- Have the students read through the student handout, especially the ‘Why lake modeling?’ and ‘Today’s focal question’ sections.

Introductory PowerPoint Presentation

Note: the numbers below match the PowerPoint slide numbers. The text for each slide is also in the "Notes" of the PowerPoint, so can be viewed when projecting in Presenter View.

1. Welcome the students to class. It might be helpful to go around the room and briefly discuss if anyone has experience programming or modeling. The point of this is to emphasize that most students are likely novices, and that asking lots of questions is ok because their peers are novices as well.
 - a. It is really important at this point to emphasize that there will be lots of new material covered during this module, and that going slowly and asking for help is very much encouraged!
2. Quick road map of what will be covered in the PowerPoint
3. Why do we want to know how climate change is affecting lakes? Because there is lots of variability in how climate change is occurring globally and lakes provide critical ecosystem services for humans, so we need to explore many different climate scenarios.
4. Today, we are going to focus specifically on lake thermal structure. Question to ask the students on this slide: what is lake thermal structure?
5. Here are lake temperature data from Lake Sunapee, a large dimictic lake in New Hampshire, USA, from 2012. Tell the students how time is on the x-axis, and depth is on the y-axis (from the surface to the sediments at 9 m depth), with color referring to the temperature, from cold 0°C (blue) to very warm 30°C (red). These are called heat maps or thermal plots, and we will be making lots of these figures in the module.
 - a. When we say 'lake thermal structure,' we are referring to both the magnitude of the water temperature in the lake at multiple depths, as well as the stratification pattern.
 - b. If a lake is thermally stratified, it exhibits distinct layers of water on a density gradient. In the summer, warmer water on top is less dense than colder water on bottom. Maximum density of water is at ~4°C: water at 25°C is substantially less dense than colder water, hence why it is at the surface of the lake.
 - c. Talk through the changes in thermal profiles over time in the context of water density differences among lakes.
 - d. Note! We are focusing here on thermal stratification, not chemical stratification. Density gradients due to differences in water chemistry (e.g., salinity) can be a major factor altering stratification in some lakes, but we are focusing on thermal density gradients in this module.
 - e. We are now going to introduce some terminology:
 - i. *Isothermal* means that the lake has the same temperature along the water depth profile, as indicated by the same color from the lake's surface to the bottom at a certain point in time. When the water is isothermal, we assume that it is mixing, bringing oxygen from the surface to the sediments, and nutrients from the sediments to the surface.
 - ii. The *epilimnion* encompasses the zone of the lake from the surface down to the *thermocline*, or the depth of maximum temperature change.
 - iii. The *hypolimnion* is the lake zone below the thermocline.

6. Water temperature is regulated by several factors, namely, solar radiation, air temperature, wind, precipitation, and inflow/outflow streams. All of these will interact to control thermal structure. The depth of the thermocline is regulated by solar radiation and wind-driven mixing.
7. Drivers of lake thermal structure occur at both local (e.g., warm groundwater or cold inflow streams entering a lake) and regional scales (e.g., different air temperature and precipitation patterns). Looking at the effects of drivers operating at multiple scales uses a macrosystems ecology approach, which entails studying ecological dynamics and feedbacks across multiple spatial and temporal scales. Here, we are using a macrosystems approach combining high-frequency sensor data and simulation models to compare the effects of local and regional drivers on lake thermal structure.
8. To study the effects of climate change on lakes, researchers use models, because it is impossible to manipulate factors such as solar radiation and wind on real lakes at the whole-lake scale in a controlled way. However, simulation modeling allows us to test how changes in one or more drivers alters lake structure.
9. The model we are going to use is GLM (the General Lake Model), developed in 2012 as an open-source model by researchers in GLEON, the Global Lakes Ecological Observatory Network. GLM gives us the opportunity to do climate change experiments, in which we modify different climate conditions and study their effects on the lake.
10. GLM is a lake physics model, which uses climate forcing data as input (e.g., high-frequency inflows, snow, wind, temperature, humidity, radiation) and models lake thermal structure, with lake temperatures as output. GLM has a water quality model that also models water chemistry and food webs (AED), but for today, we are going to focus on lake physics.
11. GLM requires a separate new folder/directory on each student's laptop, which will be downloaded as an RStudio project. Within this folder will be: 1) a CSV (comma-separated values) file, which has the climate driver data (also referred to as a 'met' file; or a file with the meteorological data), 2) an .nml file, which can be opened as a text file, that acts as a master script to the GLM model (it contains parameters for how the model should work, tells the model basic info on the lake, such as depth, latitude, time period of the simulation, etc.), and 3) any inflow/outflow CSV files that specify the temperature and flow rate of the connected streams. For the purpose of today, we are only going to have an .nml file and met .CSV file in our directory and assume that our model lake has no inflows or outflows.
12. Here is an example met file, with columns for time step, shortwave radiation, longwave radiation, air temperature, relative humidity, wind speed, rain, and snow. This met file is on an hourly time step. Note the DateTime structure of the time column: GLM requires this exact format of YYYY-MM-DD hh:mm:ss. GLM also requires that the column headers are spelled exactly like what is in this file (capitalization matters!).
13. Here is an example .nml file, which goes through many required pieces of information, such as what the name of the lake being modeled is, its latitude/longitude, the time period being modeled, etc. Fun fact: nml stands for "namelist" in the Fortran programming language.
14. We are going to run GLM in R, a programming language and statistical environment that is used for running statistics, making figures, and doing lots of different analyses. Within R, you can download lots of different software 'packages' for different types of analyses. The benefit to R is that it is free, runs on all operating systems, and is reproducible- i.e., any code that you write can be saved and run later, and you know exactly what you did!

15. There are two packages that we need to run GLM in R: GLM_r and glmtools. GLM_r runs the model and glmtools gives you different functions for analyzing the model. These two packages were written by Jordan Read and Luke Winslow at USGS, and can be downloaded from their GitHub website.
16. Learning objectives! Talk through these with the students one by one: use the embedded animations to sequentially show each of the six bullet points. Most importantly, the goal here is to have students develop their own hypotheses for how climate change can affect lakes, and then test their hypotheses by creating a climate scenario and forcing the lake with that scenario. They will then make mini-presentations to share their model findings with the class before learning new distributed computing technology tools to run hundreds of simulations.
17. Introduce Activity A, which has four objectives (have students work in pairs).
 - a. Download and extract the GLM files and R packages successfully onto your computer
 - b. Explore the example model files- challenge students to figure out where on the globe “Awesome Lake” is!
 - c. Run the model and look at the thermal output
 - d. Examine how your model output compares to the observed high-frequency field data for your lake

At this point, stop the PowerPoint and let the students get started on Activity A.

Activity A: Plotting water temperatures from a lake model

Activity A challenges the students to create a plot of lake temperatures in a default model lake, using real, high-frequency climate forcing data. Ask the students to open the module R script that corresponds to their operating system (Mac OS X or PC). Before you let them work independently in their pairs, open the R script on your computer and project it to show them how to run lines of code, and also what lines of code correspond to Activity A. **Important: Tell the students to read through the detailed annotation corresponding to each line of code before they run the code in R. The most important part of this module is understanding what the code is doing, which is provided in the annotation in extensive detail.** The annotation is the text that follows a line of code behind the # sign.

Common stumbling blocks for Activity A include:

- If a student is using a recent Mac operating system (e.g., El Capitan, Sierra, High Sierra), they may get an error message prefaced with “DYLD: library...” when they try to run GLM. To avoid this error, they need to disable SIP, a Mac program, before starting the module. See the “R You Ready for EDDIE” file for more information and step-by-step directions.
- In Objective 2, students may not know how to find their folder path to set the sim_folder. In RStudio, click on the Files tab in the right hand windows, and the file path should be shown at the top. Otherwise, students can navigate to their lake_climate_change folder on the desktop, then right click and select Properties (**PC**) and see the path under “Location” or right click and select Get Info (**Mac**) and see the path under “Where”. Mac users can copy and paste this path into RStudio.
 - Note that if PC users determine their path this way, they need to make sure that their slashes in R go the *opposite* direction as is shown in the Properties file.

- **Good:** C:/Users/farrellk/Desktop/lake_climate_change
 - **Bad:** C:\Users\farrellk\Desktop\lake_climate_change
- If a student has opened the field_data.csv or met_hourly.csv files in Excel prior to the module, Excel automatically changed the format of the date-time column to a default format that GLM cannot recognize. This can cause R to give a number of errors, including:
 - Problems running the GLM model in Activity A, Objective 3. The command `run_glm(sim_folder, verbose=TRUE)` will start the GLM run, but you will likely get an error similar to: “Day 2451636 (2000-04-01) not found”
 - Problems comparing the modeled GLM data and the observed data from the field_data.csv file. In Activity A, Objective 4, when students attempt to run the command `plot_temp_compare(nc_file, field_file)`, they will get the error message “Error in as.POSIXlt.character(x, tz, ...) : character string is not in a standard unambiguous format”
 - To fix these problems, we recommend that the student delete the folder of module files they have downloaded, and re-download the original zipped version (extracting the folder to the Desktop without opening any files!).
- If a student is getting lots of error messages with GLM_r, it may be because of difficulty with the download. We recommend deleting the package (the commands: `remove.packages('GLMr')` and `remove.packages('glmtools')` are helpful for this in R) and starting over with re-installing the packages, using the provided code in the script in Activity A.
- In Objective 4, students may have difficulty modifying the plotting code to plot water density instead of water temperature. The script to plot water temperature, which can be easily changed to plot water density is below, with key portions that need to be changed highlighted in yellow. The final script to successfully plot water density is pasted several lines below. (Note that the `ylim=c(15,35)` can be deleted, rather than changed—it will just allow R to estimate the y-axis range based on the data values).

Parts of water temperature plot script to modify:

```
water_temp <- compare_to_field(nc_file, field_file, metric="water.temperature",
  as_value=TRUE, na.rm=TRUE)
plot(water_temp$DateTime, water_temp$obs, type="p", col="blue", ylim=c(15,35),
  ylab="Water temperature in C", xlab="Date")
points(water_temp$DateTime, water_temp$mod, col="red")
legend("topleft", c("Observed", "Modeled"), lty=c(1,1), col=c("blue", "red"))
```

Successful changes to plot water density after having modified the code above:

```
water_dens <- compare_to_field(nc_file, field_file, metric="water.density",
  as_value=TRUE, na.rm=TRUE)
plot(water_dens$DateTime, water_dens$obs, type="p", col="blue", ylim=c(994,1000),
  ylab="Water density in kg/m3", xlab="Date")
points(water_dens$DateTime, water_dens$mod, col="red")
legend("bottomleft", c("Observed", "Modeled"), lty=c(1,1), col=c("blue", "red"))
```


- Sometimes you need to close and reopen RStudio if you are having lots of problems with packages installing or loading. Time spent on these errors can be minimized by encouraging students to try to install all packages ahead of time, using the them “R You Ready for EDDIE” file.

Walk around the pairs and make sure that everyone is able to follow along the R script successfully. When they are done with Activity A, they will produce a figure of thermocline depth of observed versus modeled data, as well as a temperature plot of the modeled default lake. Once 90% of the class has finished with Activity A, return to the PowerPoint to introduce Activity B, and then make sure to help the remaining 10% of students finish Activity A after the others have started B.

- A technique that we have found helpful in “equalizing” a classroom with different skill levels and computer experience is to recruit the more advanced students that have finished an activity to assist the pairs that may be moving more slowly and have lots of questions.

Activity B: Develop a climate scenario, generate hypotheses, and model how the lake responds

18. Introduce Activity B, which has two objectives:
 - a. Develop a climate scenario for any region and explore how it will affect lake thermal structure.
 - b. Create figures to share with the class that demonstrate the model output for the climate scenario.
19. Students might find this website helpful for thinking through predicted climate change in their home region, or other areas. For global projections, you may also try visiting <http://climatemodels.uchicago.edu/maps>. Note that on this second site, students will need to first enter the starting year, and press “Download map”, then the target year for their climate scenario, and press “Download map” again, then click “Calculate anomalies” to estimate the temperature offset they should use when manipulating the data file. Also note that the Climate Model Mapper represents the RCP 8.5 (comparatively high emissions) scenario.

At this point, challenge the students to directly create a hypothesis for how some component of climate change may alter lake thermal structure. This can involve how an extreme event (e.g., a tornado! Hurricane! Superstorm blizzard!) or a more gradual change (e.g., +2°C increase of observed air temperature) affects water temperature and thermal stratification. **The important take-home message here is that students need to 1) first discuss how they expect a climate scenario to affect lakes, 2) design a climate scenario to test their hypothesis (or use one of the pre-made scenarios), and 3) explore if the model output from their scenario supports or contradicts their hypothesis.**

This is a challenging activity because it involves hypothesis generation and instantiating their climate scenario into a met file, so going slow, walking around the classroom to check in, and asking the students about their hypotheses is really important here. **Important: tell the students up front that**

they will need to prepare some figures (e.g., plots of their altered met files, the thermal heat maps generated from their model output) to share with the other students.

20. The table below shows the units for the different variables in the met_houly driver data file. Note that if students are changing rain in their climate scenario, they should be particularly mindful of units, as inputs of precipitation in meters will crash the model.

File Name	Variable	Units	Input Format	Example	Conversion
met_hourly.csv	time	hourly	yyyy-mm-dd hh:mm	2000-03-02 00:00	
met_hourly.csv	ShortWave	W m ⁻²	decimal	-2.514	
met_hourly.csv	LongWave	W m ⁻²	decimal	308.887	
met_hourly.csv	AirTemp	°C	decimal	20.05	
met_hourly.csv	RelHum	%	decimal	38.62	
met_hourly.csv	WindSpeed	m s ⁻¹	decimal ≥ 0	1.87	Convert from miles per hour, multiply by 0.447
met_hourly.csv	Rain	m day ⁻¹	decimal ≥ 0	0.000592	To convert from inches per day, multiply by 0.0254
met_hourly.csv	Snow	m day ⁻¹	decimal ≥ 0	0	Keep as 0

At the end of Activity B, spend some time going around the classroom so that each student pair can show what their climate scenario was, and what the output looked like. Ask probing questions and try to initiate a class discussion in which the other students respond to questions, and ask their own.

Questions could include:

- What was your hypothesis and why?
- How likely is this particular scenario in the real world? What part of the world might experience these conditions?
- How are thermal stratification, water temperature, and thermocline depth affected by the scenario? Why do we see these patterns?
- How does the timing of stratification change?
- When and what is the maximum and minimum water temperature?
- Does the output (plot) support or contradict your hypothesis of how the climate scenario would affect the lake?
- Why do warmer air temperatures often generate colder hypolimnetic temperatures in model output?

Common stumbling blocks for Activity B include:

Important: Tell the students to read through the detailed annotation corresponding to each line of code before they run the code in R. We have embedded lots of hints and troubleshooting help within the R script! We encourage instructors to read through and run the R code before teaching the module so that you are familiar with all of the steps of this activity.

- Editing the met_hourly.csv file: ***Note that students do not need to manually edit the met_hourly.csv file if using one of the pre-made scenarios, and can skip to step 4 below.***
Opening the file in Excel will alter the date-time formatting of the file so that GLM cannot recognize it. You will get an error something like this: “Day 2451636 (2000-04-01) not found”. To get around this error, students will need to follow these steps EVERY time they open the met file in Excel. Note that these steps are provided in the students’ R script.
 - 1) Copy and paste an extra version of the met_hourly.csv file in your sim folder so that you have a backup in case of any mistakes. Right click and rename this file something like ‘met_hourly_UNALTERED.csv’ and be sure **not to open it**.
 - 2) Open the met_hourly.csv file in Excel. Manipulate the different input meteorological variables to create your climate/weather scenario. Note: the order of the columns in the met file does not matter- but you can only have one of each variable and they must keep the same header name (i.e., it must always be 'AirTemp', not 'AirTemp+3C'). When you are done editing the meteorological file, highlight all of the 'time' column in Excel (by clicking on the “A” column label), then click on 'Format Cells', then ‘Custom’. In the ‘Type’ or ‘Formatting’ box, type in **YYYY-MM-DD hh:mm** exactly. This is the only time/date format that GLM can read. When you click ok, this should change the format of the 'time' column so that it reads: ‘1999-12-31 00:00’ with exactly that spacing and punctuation. Save this new file under a different name, following how you have created your scenario, e.g., ‘met_hourly_SIMULATEDSUMMERSTORMS.csv’. Close the CSV file, saving your changes.

Now, do NOT open the file in Excel again- otherwise, you will need to repeat this formatting process before reading the altered met file into GLM.

Note! It may be possible that your version of Excel requires different steps to do this, but it should still be possible to alter the column date-time formatting regardless of your Excel version.

- 3) Read in your new met file to run the simulation. You need to run this code:


```
metdata <- read.csv('met_hourly_SIMULATEDSUMMERSTORMS.csv',
header=TRUE) ## Edit the name of the CSV file so that it matches your
new met file name. I used SIMULATEDSUMMERSTORMS here because that was
my particular scenario!
```

Important note: any time you alter the meteorological input file, you will have to repeat these formatting steps to be able to read it into R and run the model in GLM.

- 4) Finally, you need to edit the glm2.nml file to change the name of the input meteorological file so that it reads in the new, edited meteorological file for your climate scenario, vs. the default ‘met_hourly.csv’. In the .nml file, scroll down to the meteorology section, and change the 'meteo_fl' entry to the new met file name (e.g., from 'met_hourly.csv' to 'met_hourly_SIMULATEDSUMMERSTORMS.csv').
- 5) Note- check to make sure that your quotes ' and ' around the met file name in the .nml file are upright, and not slanted- sometimes the .nml default alters the quotes so that the file cannot be read in properly (super tricky!). Once you have edited the .nml file name, you should check to make sure that it is correct with the following

command:

```
nml<-read_nml(nml_file) #read in your .nml file from your new
directory

get_nml_value(nml, 'meteo_fl') #if you have done this correctly, you
should get an output that lists the name of your new meteorological
file altered for your weather/climate scenario.
```

- 6) Finally, you can now run the model with the new edited nml file, following the instructions as described above for Objective 3.

If a student pair has finished much earlier than the other students, ask them to develop a second scenario to compare with their first one, or to help the other student pairs finish. Staggering the student presentations is ok, too!

Activity C: Using distributed computing to run hundreds of lake model simulations

21. Introduce Activity C. On this slide, emphasize that it is feasible to manually edit a met file and .nml code to create one scenario, but what if you want to run hundreds of scenarios that are slightly different? (e.g., a first scenario that is +2°C, a second scenario that is +2.1°C, a third scenario +2.2°C, and so on). It is not feasible to manually edit hundreds of met files, plus each of those simulations take ~5-10 minutes to run on your own computer. Scaling up, 100 simulations would take 500-1000 minutes (8-16 hrs); that is too long for us to efficiently evaluate model output. We need new tools and technologies to create scenarios and run lake models more efficiently.
22. To remedy this problem, we are going to use a new R package called GRAPLER that creates constant offsets (e.g., +2°C added to all observational air temperature data) for GLM met variables, submits the jobs via a web service to run on other computers (distributed computing), and then returns the model output to us.
23. Many ecology students are not familiar with distributed computing, so a little background here may be helpful. The GRAPLER Web service divides the problem into many parts, HTCondor (a computer job scheduler) dispatches them to be solved by different computers, and IPOP provides the network.
24. A schematic of the workflow for one GLM simulation: you submit one GLM simulation to R on your computer, with an .nml and met .CSV file. R (via the GLMr package) runs the model, and then provides output that you can access via glmtools.
25. We now scale up one simulation to many using the GRAPLER. GRAPLER does three things: 1) creates many simulations with unique met_hourly files per your specifications, 2) distributes the simulations to many computers so that hundreds of runs can happen in the time it takes one on your computer, and 3) returns the output to you via R.
26. A schematic of the GRAPLER workflow: you submit one GLM simulation to R on your computer. The GRAPLER creates hundreds of new simulation files following your specifications (each with small tweaks in their met files), based off that one original simulation. The GRAPLER then

distributes those hundreds of simulations to run on other computers, and returns the output to you, which you can query in R. Voilà!

27. We have provided some more computer science information on how GRAPLER works, which is not necessary to know for running the module but might be interesting for advanced students. Otherwise, this slide can be deleted.
28. Tell the students to return to their R scripts to run the demo for Activity C. Depending on the amount of available time left in the teaching period, you could encourage advanced students to create their own simulation experiment beyond the demo using the GRAPLER, which they can share with the other students, similar to Activity B.

Common stumbling blocks for Activity C include:

- Problems loading GRAPLER package. Try re-running the installation using the script below:
 - This may occur if the installation did not work correctly. Try re-running the installation using the script below:

```
library("devtools")
devtools::install_github("GRAPLE/GRAPLER")
```

Then reload GRAPLER with the `library(GRAPLER)` command.
 - Also check that students have the most up-to-date version of GRAPLER installed, as older versions are no longer supported. Check this by running the following code:

```
packageVersion("GRAPLER")
```

If the output does not read '3.1.0' or a higher number, re-install GRAPLER.

- Status seems to be stuck on 0.0%. GRAPLER does not always update small increments of progress, so it may only show 0.0%, 50.0%, and 100.0%. Encourage students to be patient—they can work on updating the presentation they'll make to their classmates, or discuss model output with another team while they wait.
 - The speed of the GRAPLER simulation depends in part on whether other projects are currently using the distributed computing resources. However, if after ~10 minutes the status is still 0.0%, there may be a problem with the simulation submission. **We encourage students to run only batches of jobs with 100 simulations maximum at a time**, vs. all simultaneously submit 1000s of jobs to the GRAPLER system.
- If students run in to other issues using GRAPLER, you might consult the following sites to check for known, current issues with the interface: <https://github.com/GRAPLE/GRAPLER/issues> and <http://grapple.org/grapple-v3.1.0-released/>

Resources and References

Optional pre-class readings:

- Williamson, C.E., J.E. Saros, W.F. Vincent, and J.P. Smol. 2009. Lakes and reservoirs as sentinels, integrators, and regulators of climate change. *Limnology & Oceanography* 54: 2273-2282.
- Sahoo, G. B., S.G. Schladow, J.E. Reuter, and R. Coats. 2011. Effects of climate change on thermal properties of lakes and reservoirs, and possible implications. *Stochastic Environmental Research and Risk Assessment* 25.4: 445-456.

- Hipsey, M.R., L.C. Bruce, and Hamilton, D.P. 2014. GLM - General Lake Model: Model overview and user information. AED Report #26, The University of Western Australia, Perth, Australia. 42 pp.
- Subratie, K. C., S. Aditya, S. Mahesula, R. Figueiredo, C. C. Carey, and P. C. Hanson. 2017. GRAPLER: A distributed collaborative environment for lake ecosystem modeling that integrates overlay networks, high-throughput computing, and WEB services. Concurrency and Computation: Practice and Experience 29: doi: 10.1002/cpe.4139.

Tools and high-frequency data that we will use in this module:

- Hipsey, M. R., L.C. Bruce, and D.P. Hamilton. 2014. GLM- General Lake Model: Model overview and user information. AED Report #26, The University of Western Australia, Perth, Australia. 42 pp.
- Read, J.S., and L.A. Winslow. 2016. glmtools R package v.0.14.6.
- Subratie, K., and R. Figueiredo. 2017. GRAPLER R package v.3.1.0.
- Winslow, L.A., and J.S. Read. GLMr R package v.3.1.15 and GLMr R package default files. GLMr: A General Lake Model (GLM) base package.

Recent publications about EDDIE modules:

- Carey, C. C., R. D. Gougis, J. L. Klug, C. M. O'Reilly, and D. C. Richardson. 2015. A model for using environmental data-driven inquiry and exploration to teach limnology to undergraduates. Limnology and Oceanography Bulletin 24:32–35.
- Carey, C. C., and R. D. Gougis. 2017. Simulation modeling of lakes in undergraduate and graduate classrooms increases comprehension of climate change concepts and experience with computational tools. Journal of Science Education and Technology 26:1-11.
- Klug, J. L., C. C. Carey, D. C. Richardson, and R. Darner Gougis. 2017. Analysis of high-frequency and long-term data in undergraduate ecology classes improves quantitative literacy. Ecosphere 8:e01733.

We'd love your feedback!

We frequently update this module to reflect improvements to the code, new teaching materials and relevant readings, and student activities. Your feedback is incredibly valuable to us and will guide future module development within the Macrosystems EDDIE project. Please let us know any suggestions for improvement or other comments about the module at

<http://module1.macrosystemseddie.org>.

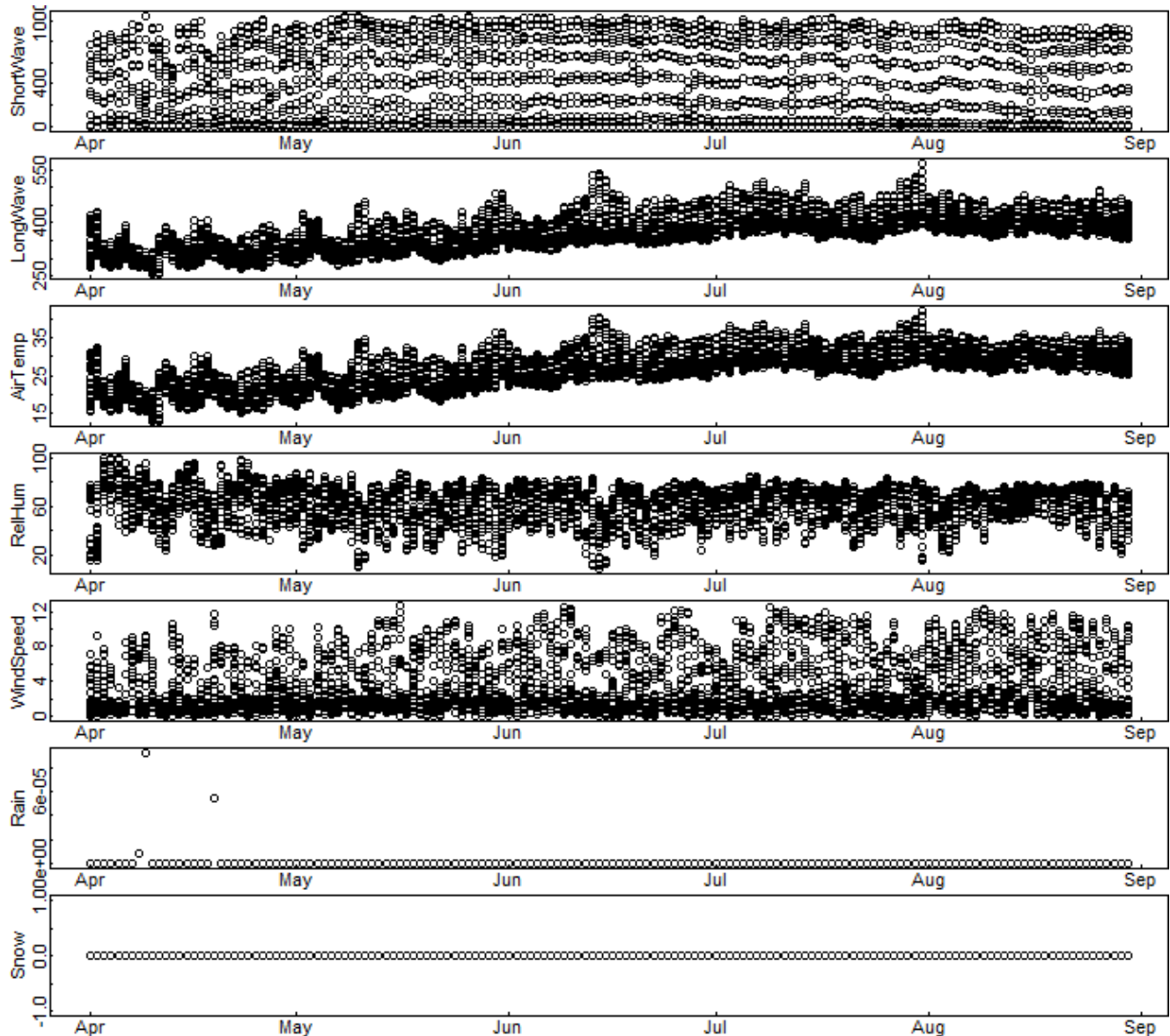
Answer Key

The following plots are indicative of what student model output should look like (approximately), if the module is run correctly.

Activity A:

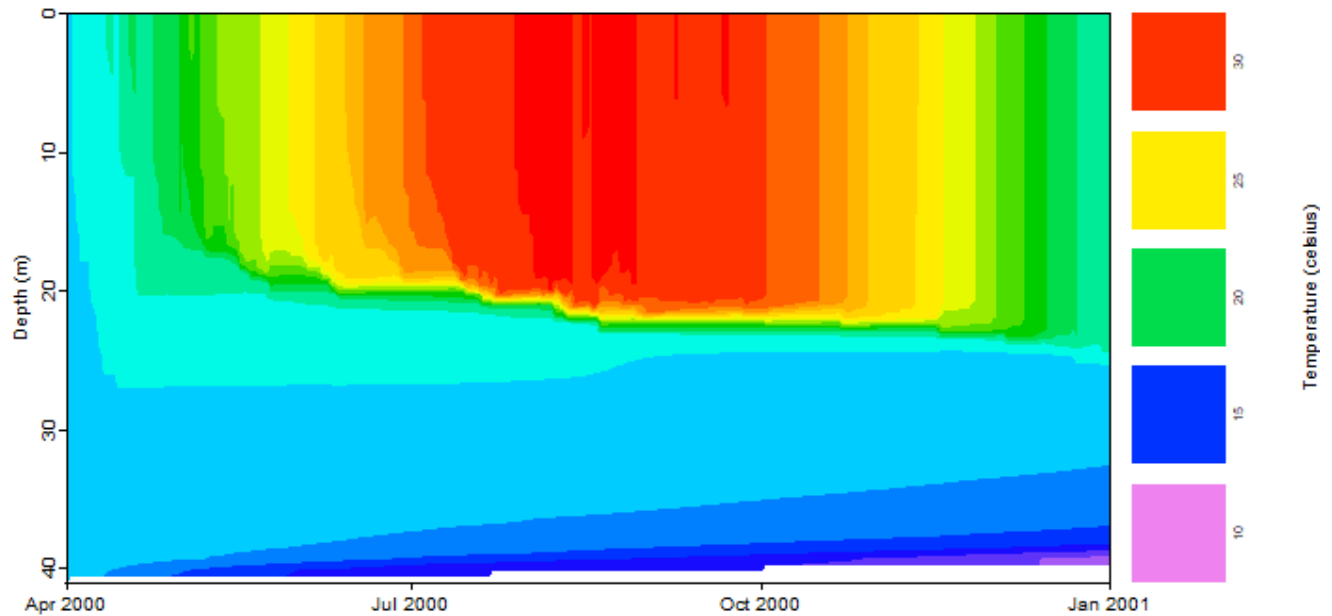
Objective 2: Meteo plot for “Awesome Lake”

Awesome Lake is based on Lake Kinneret, also known as the Sea of Galilee! Students should notice seasonal trends in air temperature and longwave radiation (higher in July & August). No strong seasonal trends in the relative humidity, wind speed, or rain.



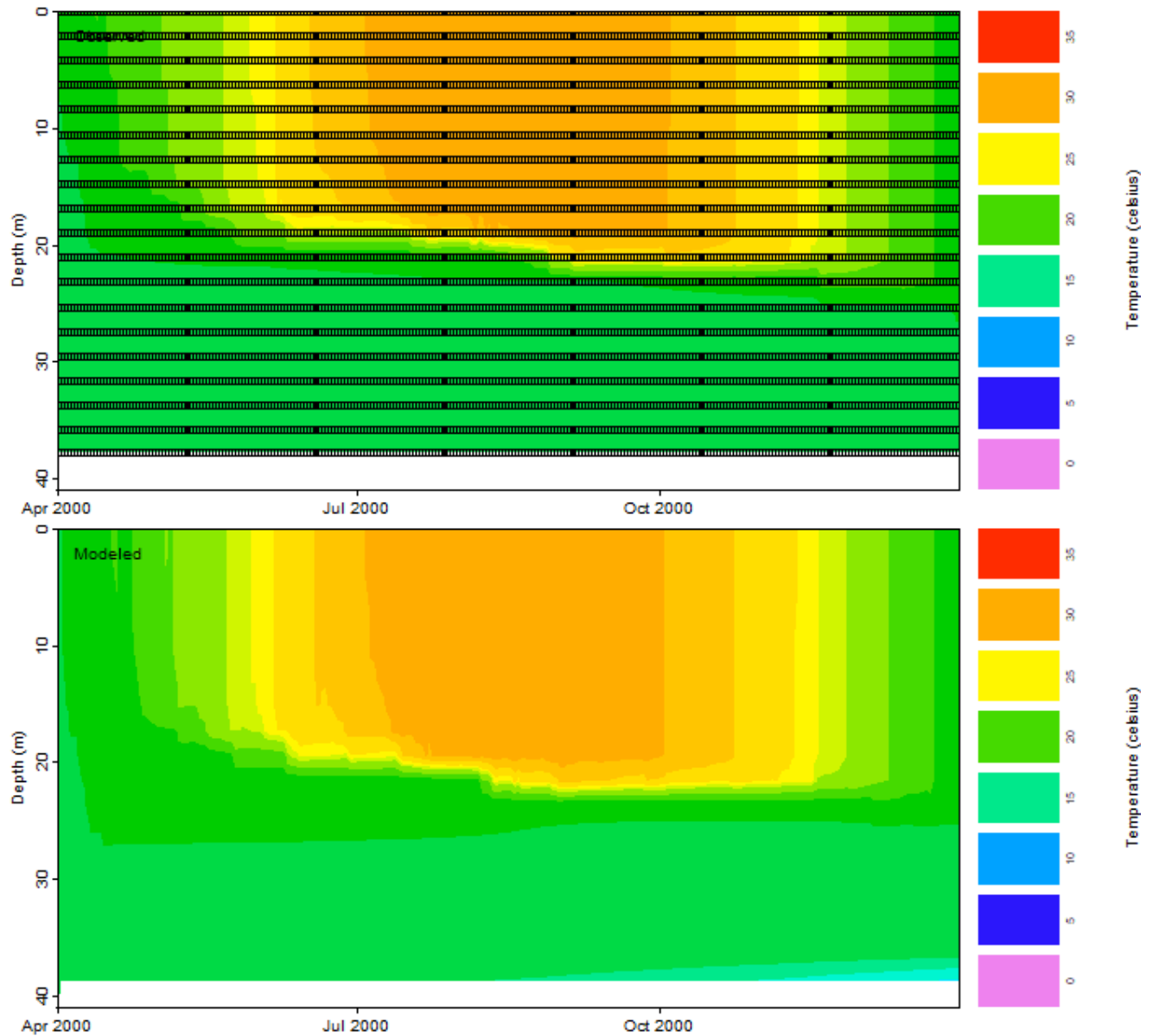
Objective 3: Temperature heatmap from GLM model output

Students should note differences in water temperature with depth, and how that changes as the year progresses. In the beginning of the simulation (April 1st), the whole water column is ~15°C—the lake is not stratified. As the year progresses, the lake begins to stratify, with warmer water near the surface (epilimnion) and cooler water deeper (hypolimnion). The depth at which the temperature changes most rapidly is the thermocline.



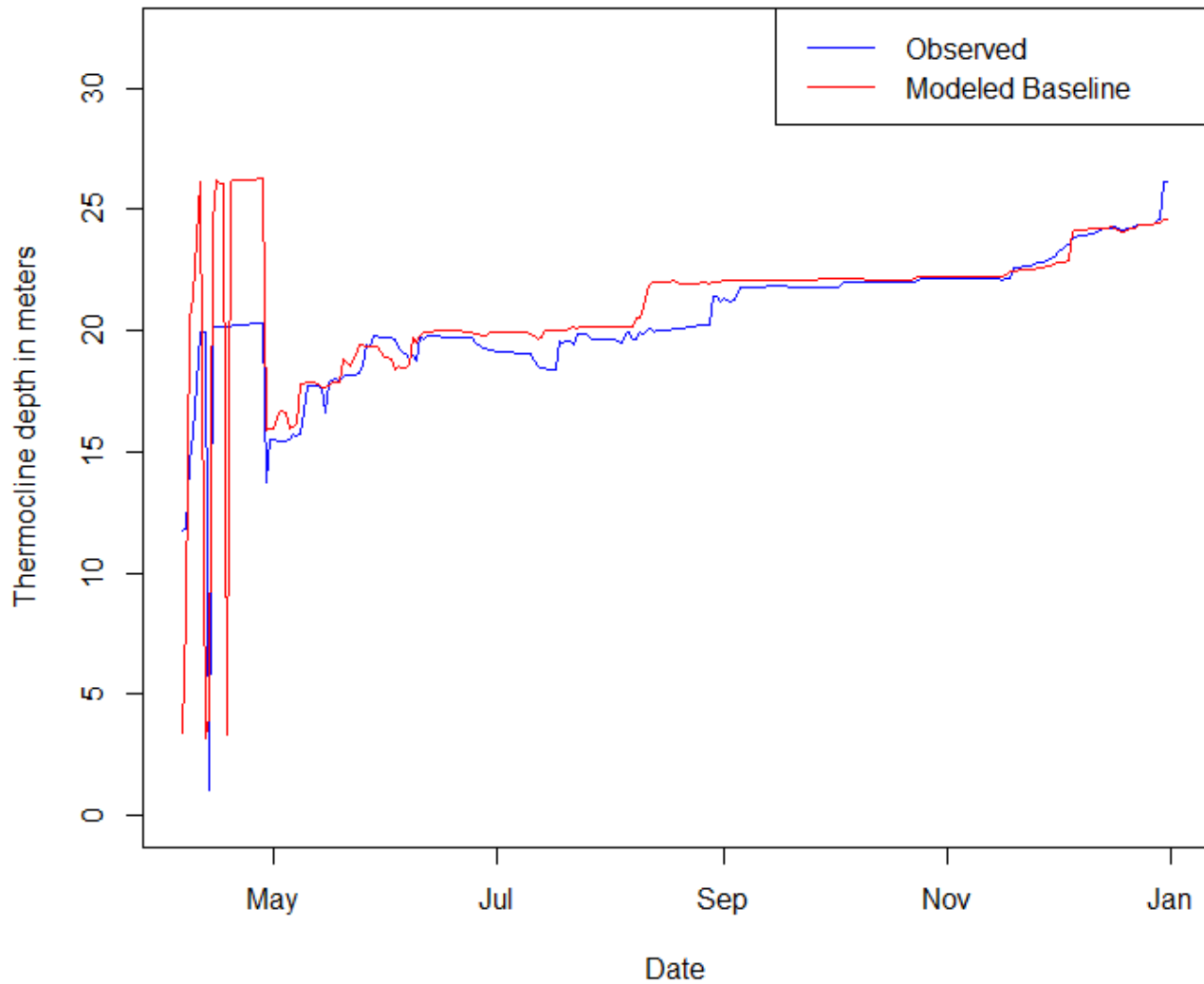
Objective 4: Observed vs. modeled water temperature heatmap

Students should note that the heat maps are *generally* similar, which suggests the model is doing a pretty good job! Note that the color scale here goes from 0-35°C on these plots, so the color scale in this plot is not the same in the plot from Objective 3 above. The black dots represent real high-frequency observations with interpolation between them; their consistency in depth indicates that these were collected from a thermistor chain.



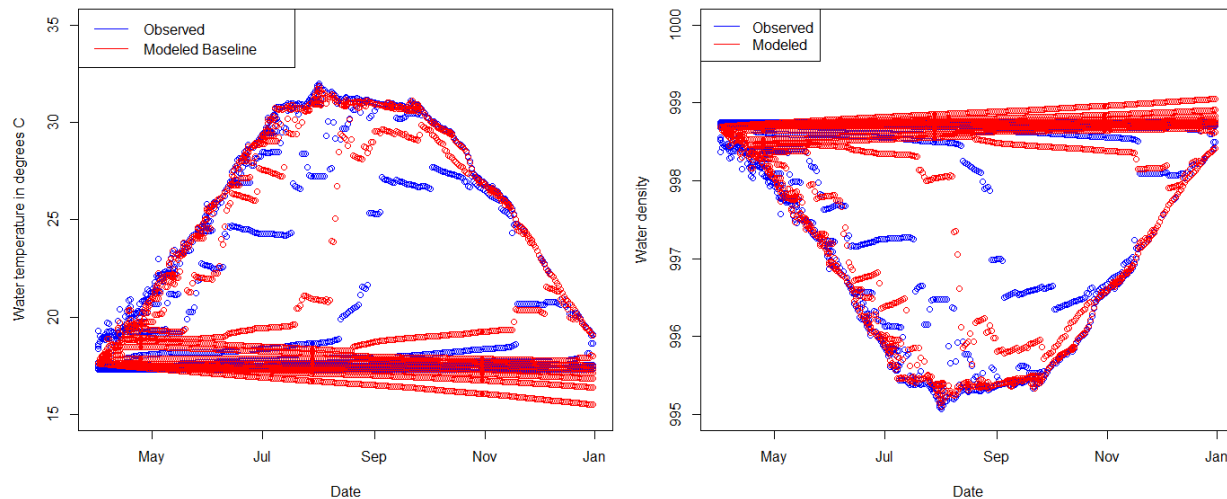
Objective 4: Modeled vs. observed thermocline depth plot

The key take-home here is that again, the model output was *generally* similar to the observed high-frequency data, though students can speculate about why the differences appear.



Objective 4: Modeled vs. observed water temperature (left) and density (right) plots

[Note: the density plot requires students to alter the code, and may be challenging.] Students should think about why there are warm *and* cold temperatures (or higher and lower density) for many days in the simulation. It is because this plot is showing all of the lake water temperatures, so includes the warm epilimnion and the cooler hypolimnion. In the density plot, the differences correspond to cooler water being more dense than warm water—students should note that the plots show the same pattern but are “upside down” from each other. The divergence between the two temperatures (or densities) on the same date (i.e., after stratification sets up in April) is an indicator that the lake is stratifying over the season, as was seen in Objective 3.



Activity B

Objective 5: Create a climate change scenario

Students could take this in lots of different ways, so output plots will vary! The most important part is that students are able to explain: 1) What, in terms of climate, changes in their scenario, 2) How they predicted the change in climate would change lake thermal structure (their hypothesis), and 3) Whether the model output supported their hypothesis (and why they think it did/did not). We recommend that student pairs make a very brief presentation to the rest of the class, explaining these three elements and sharing figures (e.g., heatmaps) from the model output that show how their lake changed. We encourage the instructor to ask questions of the students after their presentation, such as: “Did you expect that the lake would respond in this way to your climate scenario? Why or why not? What was surprising to you in this model output? Do you think that this scenario is realistic?” to stimulate discussion.

Activity C

Objective 7: Run many scenarios using GRAPLER

Students will submit a ‘job’ to GRAPLER that will allow them to run 100s to 1000s of simulations of air temperature changes, and observe how small increments of change in temperature affect lake structure. Students should note differences in the lake thermal profile, in terms of degree and timing of stratification; particularly between the two most extreme scenarios (e.g., -2°C and $+2^{\circ}\text{C}$).