Manual for GOTM v3.2.0

Tadhg Moore

2019-10-04

Introduction:

General Ocean Turbulence model (GOTM) is a one-dimensional water column model that includes the most important hydrodynamic and thermodynamic processes related to vertical mixing in natural waters (Umlauf et al. 2005). It was initially developed for modelling turbulence in the oceans, but it has been recently adapted for use in hydrodynamic modelling in lakes (Enstad et al., 2008; Sachse et al., 2014). It is made up of state-of-the-art and traditional turbulence parameterisations which allows turbulence intensity to be modelled in a dynamic manner (Burchard et al., 2006).

Typically, GOTM is used as a stand-alone model for studying dynamics of boundary layers in natural waters. Frequent hydrodynamic applications are investigations of air-sea fluxes, surface mixed-layer dynamics, stratification processes in shelf seas, dynamics of bottom boundary layers with or without sediment transport, estuarine and coastal dynamics.

Framework for Aquatic Biogeochemical Models (FABM) is used to couple GOTM with biogeochemical models.

Relatively easy to install and run on most operating systems.

Code freely available on GitHub (https://github.com/gotm-model/code).

Steps for Computer Setup for Running GOTM v.1.0

Note: It is recommended to have admin rights on the computer you are working from as otherwise you might run into difficulties with installing some of the necessary programs and packages.

Quick and easy setup:

If you use R you can set it up in 2 easy steps:

1. Follow instructions to install 'GOTMr' from GitHub. https://github.com/tadhg-moore/GOTMr
2. Follow instructions to install 'gotmtools' from GitHub https://github.com/tadhg-moore/gotmtools
3. Then you will be able to run GOTM from within the R environment.

Computer Setup:

1. Install a text editor (Recommended: Notepad++)
   https://notepad-plus-plus.org/download/v7.5.5.html)

2. Install 'PyNCView' for viewing netCDF files
   https://sourceforge.net/projects/pyncview/

3. Install Python (v.3.7.xx; latest version) [Only needed for 'parsac' tool]
   a. Install into 'C:\...' directory e.g. 'C:\Python37\'
      https://www.python.org/downloads/

4. Add Python to PATH variable in Environmental variables

   Windows 7

   a. On the 'Start' menu, right click 'Computer' and select properties
   b. Click on 'Advanced system settings'
   c. In the 'Advanced' tab click on 'Environmental Varables…'
   d. Under 'User variables for *user*', select PATH and click 'Edit…'
   e. Locate 'python.exe' on your system, it should be in 'C:\Python37' and pip.exe which should be in 'C:\Python37\Scripts'
   f. Append both file paths to the 'Variable value' in PATH by placing a ';' between that and the other file paths i.e. 'C:\Python37;C:\Python37\Scripts'
   g. Click 'OK' and then 'OK' in 'Environmental Variables' and then 'OK' in 'System Properties'

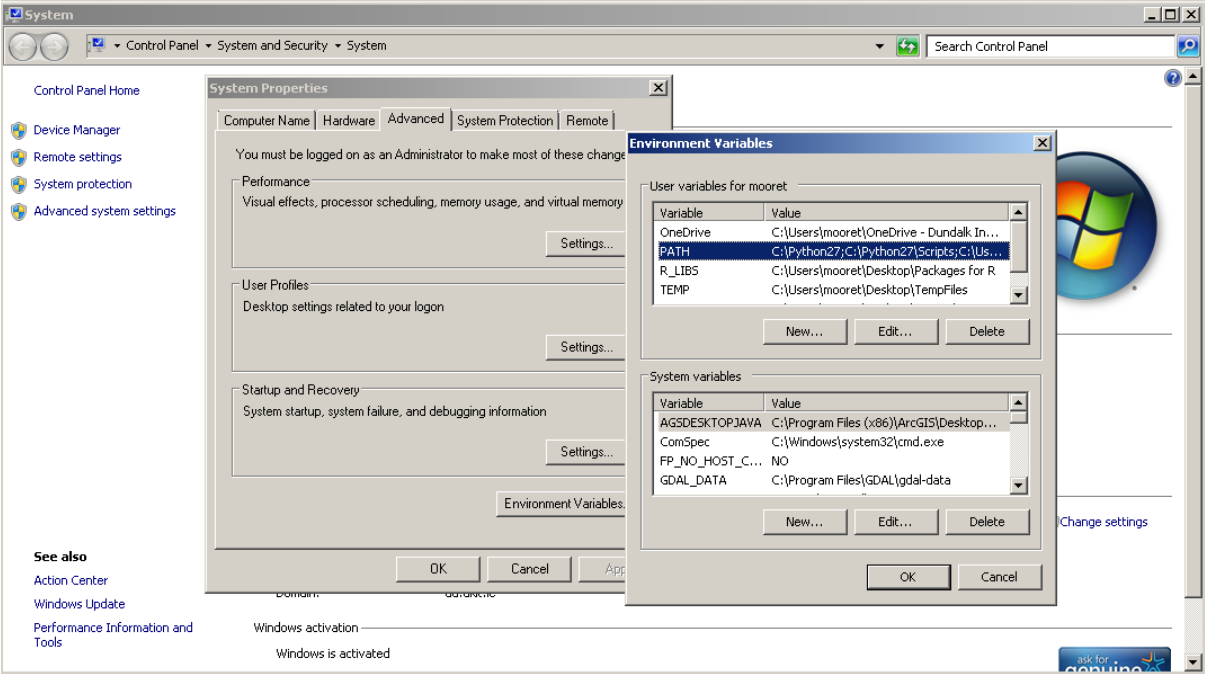

Fig 1. Screeshot showing how to set environment variables in Windows 7.

In Windows 10 follow instructions from here: https://superuser.com/questions/949560/how-do-i-set-system-environment-variables-in-windows-10

Check if Python is setup correctly:

- Open a command prompt and type the command ‘python’
- If you get the screen with a similar message below then your system should be setup (fingers crossed!)

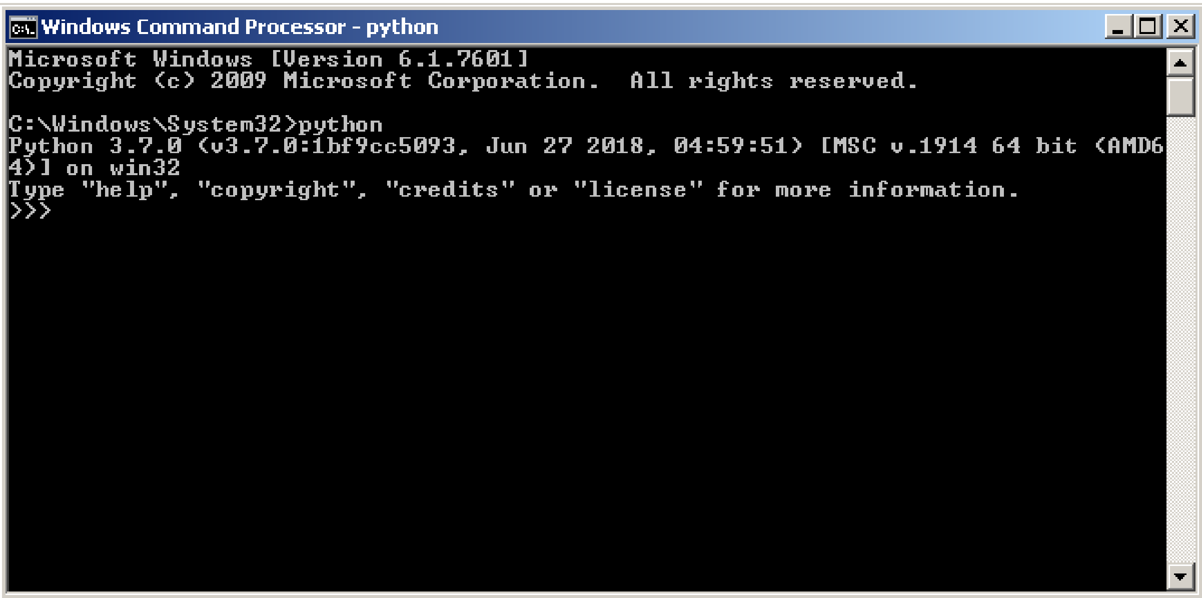

Fig 2. Screenshot of Command Prompt showing the results if Python has been added to Environment variables.

5. Install necessary packages for running parsac – Open Command Prompt and use the command 'pip install package' e.g. 'pip install matplotlib'
    a. parsac – program used to autocalibrate GOTM. See http://bolding-bruggeman.com/portfolio/parsac/
    b. SALib – needed for sensitivity analysis in parsac
    c. matplotlib – needed for generating plots using acpy
    d. numpy
    e. netcdf4 – needed for interacting with GOTM output
    f. pp – allows acpy to be run in parallel
    g. pyyaml – allows interaction with yaml files
    h. xmlplot – allows plotting from xml file via gotmgui
    i. xmlstore – for interaction with xml files
    j. editscenario – used to create and format namelist files for GOTM setup

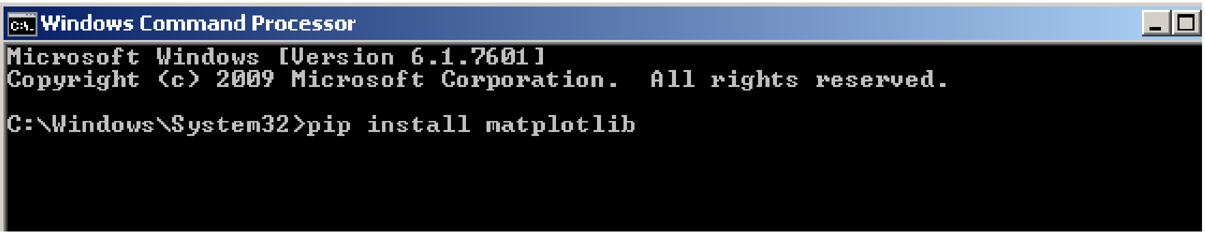

Fig 3. Screenshot showing how to install python libraries.

6. Install GOTM
    a. Download 'gotm.exe' from https://github.com/gotm-model/code/releases and create a new folder 'C:\Program Files\gotm\bin' and place 'gotm.exe' in this folder. [It can be in any folder, but it is good practice to place it here with other programs.]

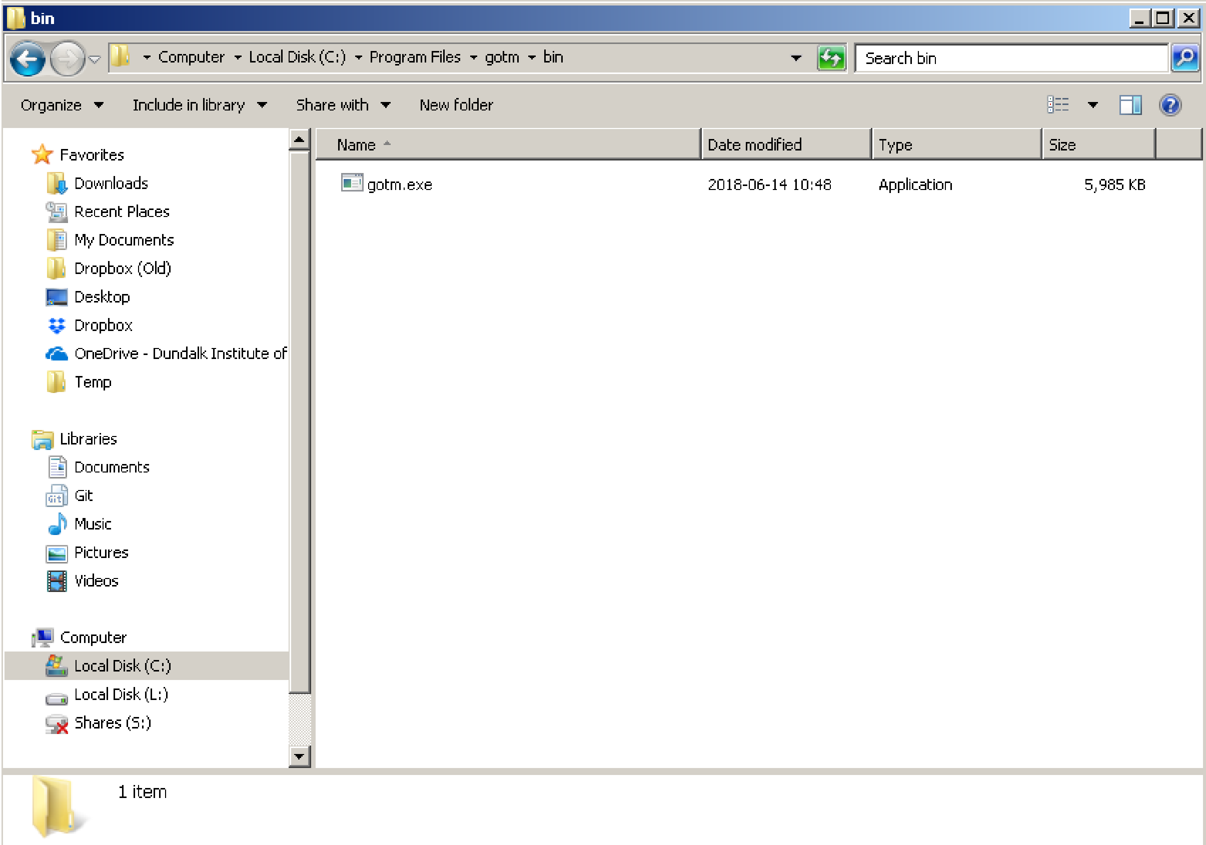

Fig 4. Screenshot showing location of 'gotm.exe'

i. Add this folder to the PATH Environment Variables.
ii. Open a new Command Prompt window and type 'where gotm' and you should get the file path to the 'gotm.exe' file.

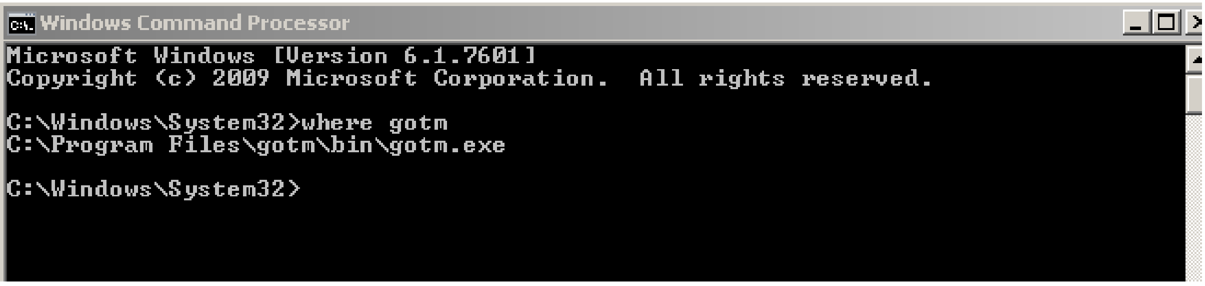

Fig 5. Screenshot showing how to check if 'gotm.exe' has been added to Environment Variables

Section 1. Guide to file setup within ‘GOTM_Example>feeagh’ folder

1. Check file paths in the following files using Notebook++:
    a. ‘run_gotm.bat’ to point to the ‘gotm.exe’ file

Batch files

Batch files (.bat) files are text files that contain commands that can be run at the Windows Command Prompt. They are saved with the extension .bat and when they are double clicked it sends the commands to the Command Prompt. The benefits of these files is that it allows you run the same commands without having to open the Command Prompt.

GOTM input files

These are the most common model problems

- Ensure that the datetime column is in the ISO format ‘YYYY-mm-dd HH:MM:SS’
- Ensure that columns are separated by a space or tab

2. Necessary files:
    a. Meteorology file
        i. Wind speed (either u & v vectors or m/s)
        ii. Mean sea level pressure (MSLP)
        iii. Air temperature
        iv. Relative humidity or wet bulb temperature or dewpoint temperature or specific humidity
        v. Cloud cover
    b. Hypsograph file
    c. File with initial temperature profile
        i. This can either contain either one profile or a time series of profiles
    d. gotm.yaml file
        i. This is the file that you will use to interact with the model. It contains all parameters and switches within the model. The editscenario.bat function uses this file to generate the namelist files (.nml) which are needed to run the model
    e. Fabm.yaml
        i. Needed if you are to run the biogeochemical model
    f. Fabm_input.yaml file
        i. Needed to include nutrient inputs in the inflow data when the biogeochemical model is coupled and is switched on.
    g. Output.yaml
        i. Needed to generate an output file from the model
        ii. Controls the type of output generated by the model

3. Optional files
    a. Precipitation files
    b. Shortwave radiation files – SWR can be calculated from cloud cover, time, and latitude
    c. Inflows/outflows
    d. Lake level
4. Preparing input data
    a. Meteorological input data
        i. Required file –

```
!DateTime        u10      v10      MSLP     AirT     DewT     CC
2005-02-22 00:00:00      -4.3     -3.9     1021.1   2.6      -2.3     0.125
2005-02-22 01:00:00      -4.0     -3.7     1021.4   2.6      -2.7     0.125
2005-02-22 02:00:00      -4.1     -3.5     1021.6   2.3      -2.4     0.125
```

        ii. Shortwave radiation and precipitation are separate files and follow the format of depth-independent variables.
    b. Depth-independent variables e.g. river inflows – datetime formatted correctly and separated by a tab or space

```
1995-01-01 00:00:00   360.63
1996-01-01 00:00:00   362.36
```

    c. Depth-dependent variables e.g. water temperature profiles, O_2 profiles etc.
        i. DateTime formatted correctly, followed by the number of observations, then 1 or 2; 1 indicates bottom to surface, 2 indicates surface to bottom. Ensure depths are negative values.

```
1998-01-01 00:00:00  13 2
-2.5  8.06867027
-7.5  8.08003998
-12.5  8.08532047
-17.5  8.0909996
-22.5  8.09687042
-27.5  8.10340977
-32.5  8.10943031
-37.5  8.11474991
-42.5  8.11933994
-47.5  8.12364006
-55.  8.12893963
-67.5  8.13638973
-85.5  8.14319992
```

Note: Either make sure to remove the headers or block them out by placing an '!' at the start of the line. Make sure to remove all NA's as they will cause GOTM to crash. With regards missing values GOTM does a simple linear interpolation between the values (or a step jump??).

Table 1. Units for the different variables used within GOTM

Variable	**File**	**Unit**
Wind u + v vectors	Meteo_file	m s^{-1}

Mean sea level pressure	Meteo_file	Pa
Air temperature	Meteo_file	°C
Dewpoint temperature	Meteo_file	°C
Relative humidity	Meteo_file	%
Cloud cover	Meteo_file	Fraction (0-1)
Precipitation	Precip_file	m s^{-1}
Short wave radiation	Swr_file	W m^{-2}
Water temperature	T_prof_file/inflow_file	°C
Water discharge	Inflow_file	m^3 s^{-1}
Depth	T_prof_file/hypsograph.dat	m
Lake level	Zeta_file	m
Area	Hypsography.dat	m^{-2}
Inflow concentrations	Inflow_chem.dat	See fabm.yaml file for units

Section 2. Editing model setup and running the model

1. All editing of the model setup takes place within the ‘gotm.yaml’ file.

YAML

YAML Ain’t Markup Language (YAML) is a human-readable data serialization language. It is commonly used for configuration files.

XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

2. Within the ‘gotm.yaml’ there are 20 main sections

Table 2. Description of ‘gotm.yaml’ file

Section	**Description**
location	Name, latitude, longitude, depth and hypsograph
time	Start, stop, time step
grid	Number of layers
temperature	Initial temperature profile
salinity	Initial salinity profile
surface	Flux calculation method, meteorological forcing and ice model
bottom	Physical bottom roughness
light extinction	Method and fraction and e-folding depths for visible and non-visible light
turbulence	Methods for turbulence, turbulent kinetic energy, dissipative length scale, stability functions
mimic 3d	Effects of horizontal gradients
velocities	Observed/prescribed horizontal velocities
o2	Oxygen inout data
streams	Inflow and outflows
physical constants	Physical constants
buoyancy	
fabm	Switch on FABM biogeochemical model
eq state	Equations of state, reference temperature/salinity, thermal expansion coefficient
restart	Hot start facility in the model
output	Controls for output files but is overrules if there is an ‘output.yaml’ present.

Note:

I would recommend having many different output.yaml files (e.g. ‘output_wtemp.yaml’) and then copy and overwrite the ‘output.yaml’ depending on what output you require. This is because you can speed up model runtime by reducing the number of output variables.

When the ‘editscenario.bat’ is run it loads the files from the ‘.xml’ file and creates 8 namelist files (‘.nml’) named according to their section and also data files ‘meteo_file.dat’, ‘precip_file.dat’, ‘swr_file.dat’ and ‘t_prof_file.dat’ are created. The data files are copies of the files that the .xml file points to.

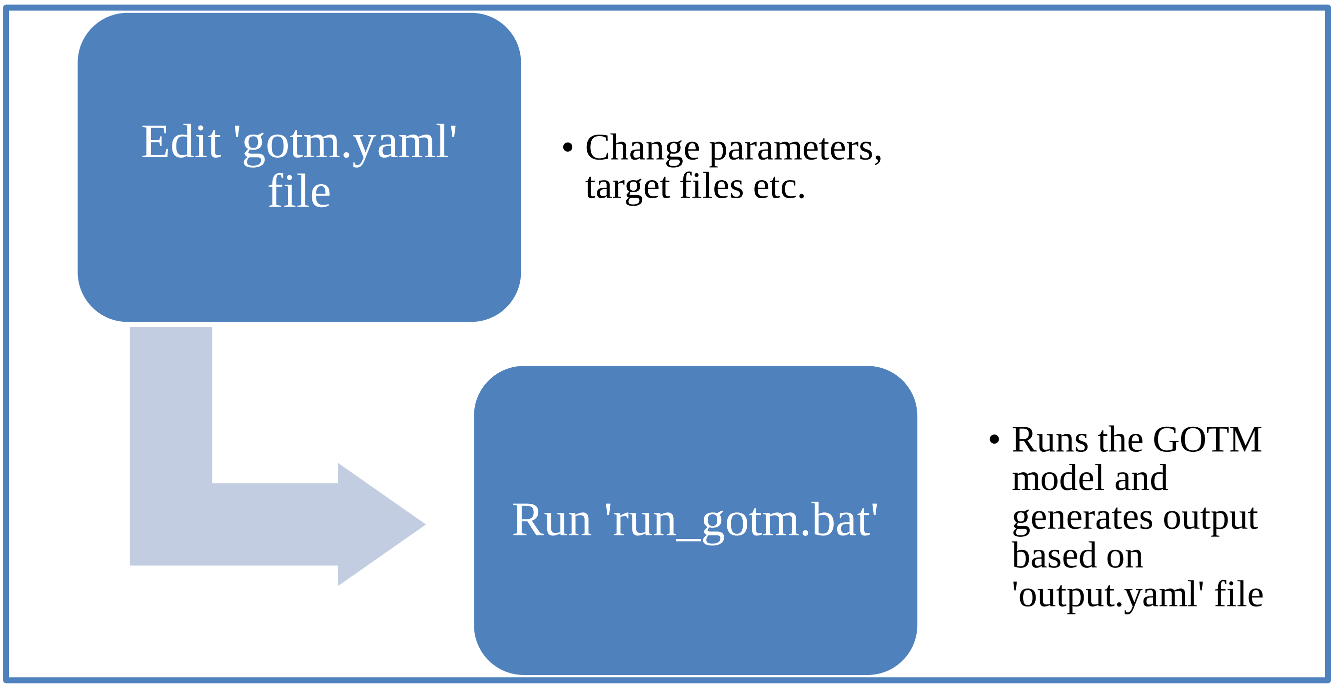

Fig 6. Workflow with current setup of GOTM.

There is a description of each parameter and what it does.within the ‘gotm.yaml’ file,

Section 3. Output from GOTM

The ‘output.yaml’ file controls the output from the GOTM model. If there is no ‘output.yaml’ file present in the folder the model will run and generate output depending on the output section in ‘gotm.yaml’.

Section 4. Viewing Output

The default output is netCDF format.

> NetCDF
>
> NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.

This appears as a file called 'output.nc' within the directory. It can be viewed using PyNcView. It allows the user to view the model output variables across 2 or 3 dimensions.

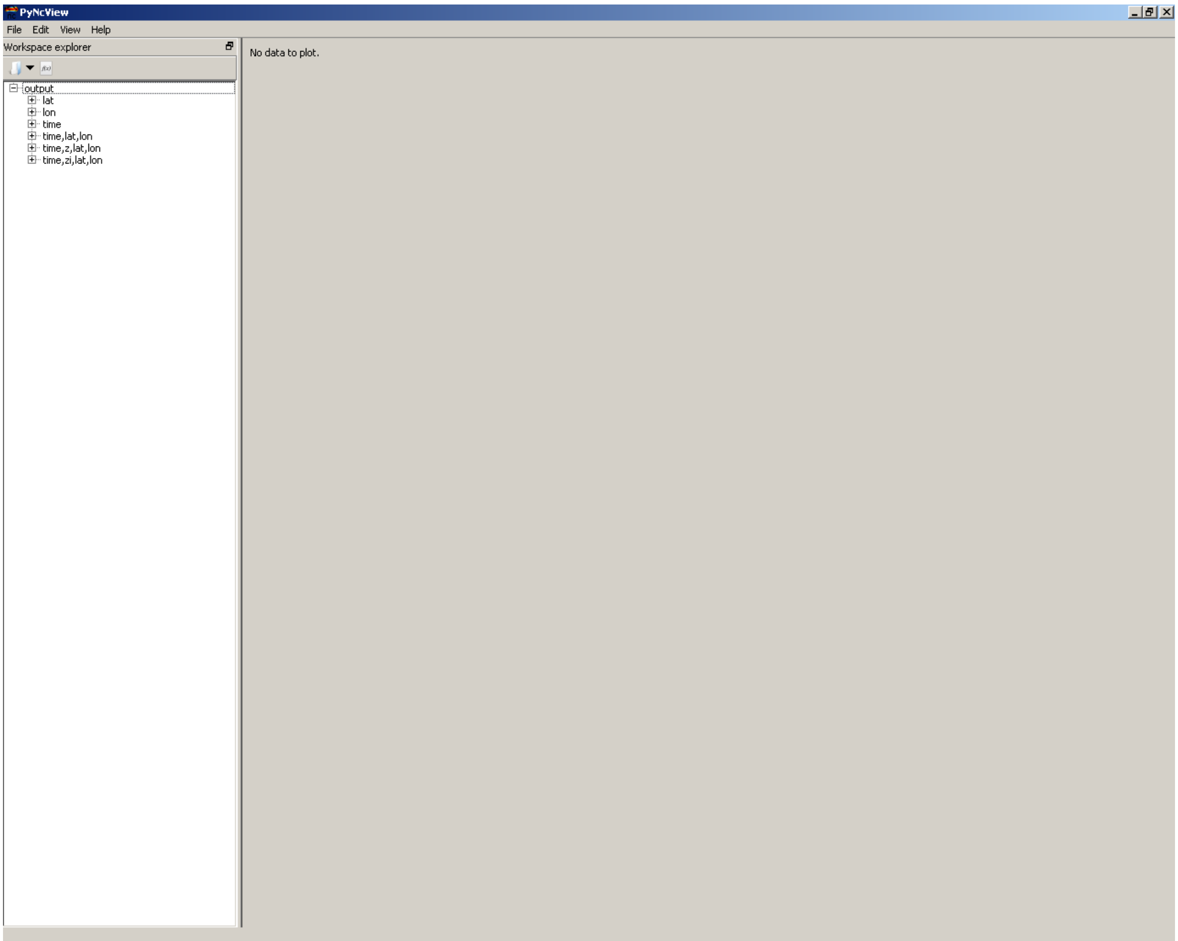

Fig 7. Initial view of netCDF file

The main variables of concern are 'potential temperature' (modelled temperature) and 'observed temperature'. These can be viewed under the 'time,z,lat,lon' drop down menu. To find a variable's code name for the 'output.yaml' file, right click on the variable and click 'Properties…' and the code will be given at the top of the new window which opens.

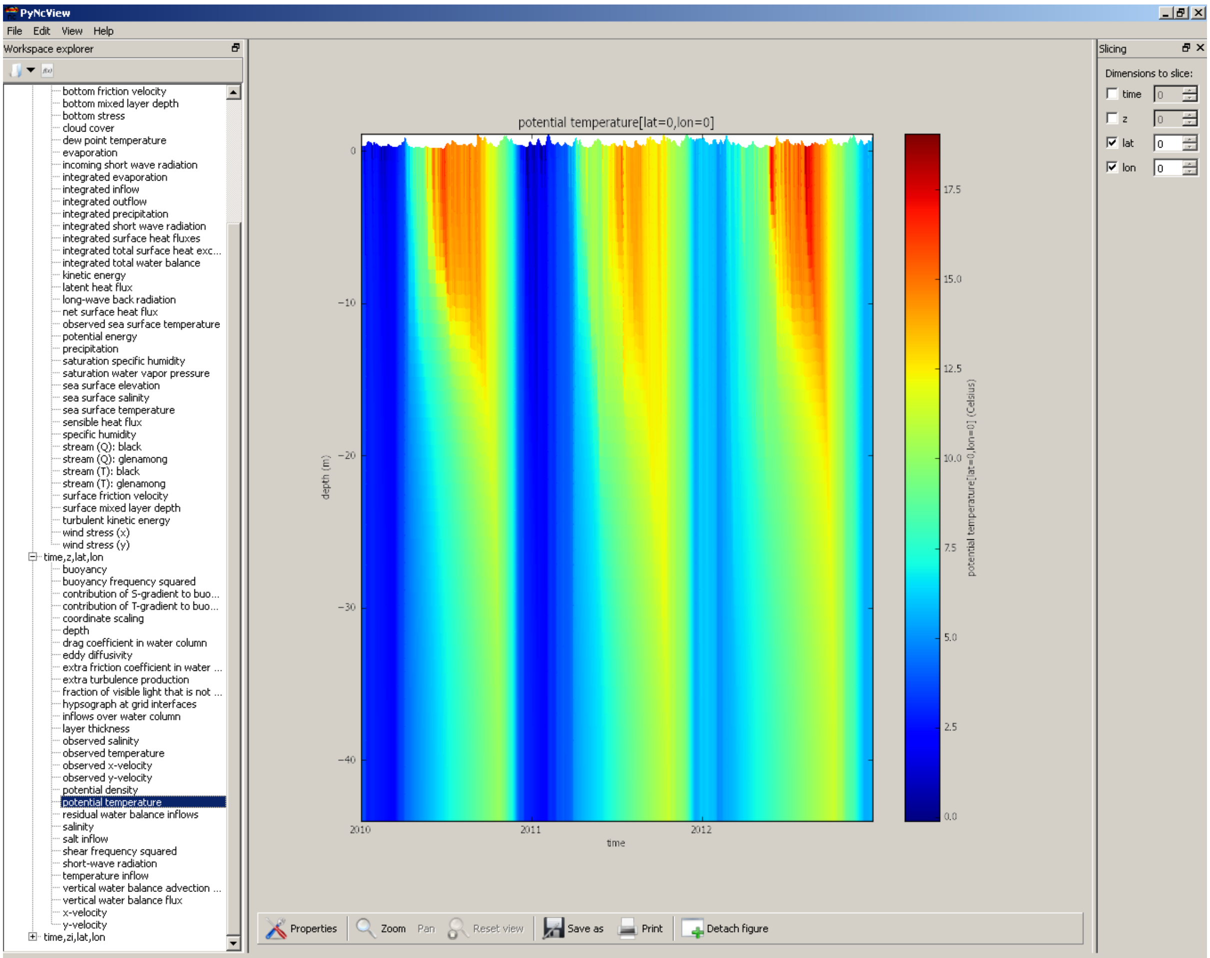

Fig 8. Temperature plot of the modelled water temperature.

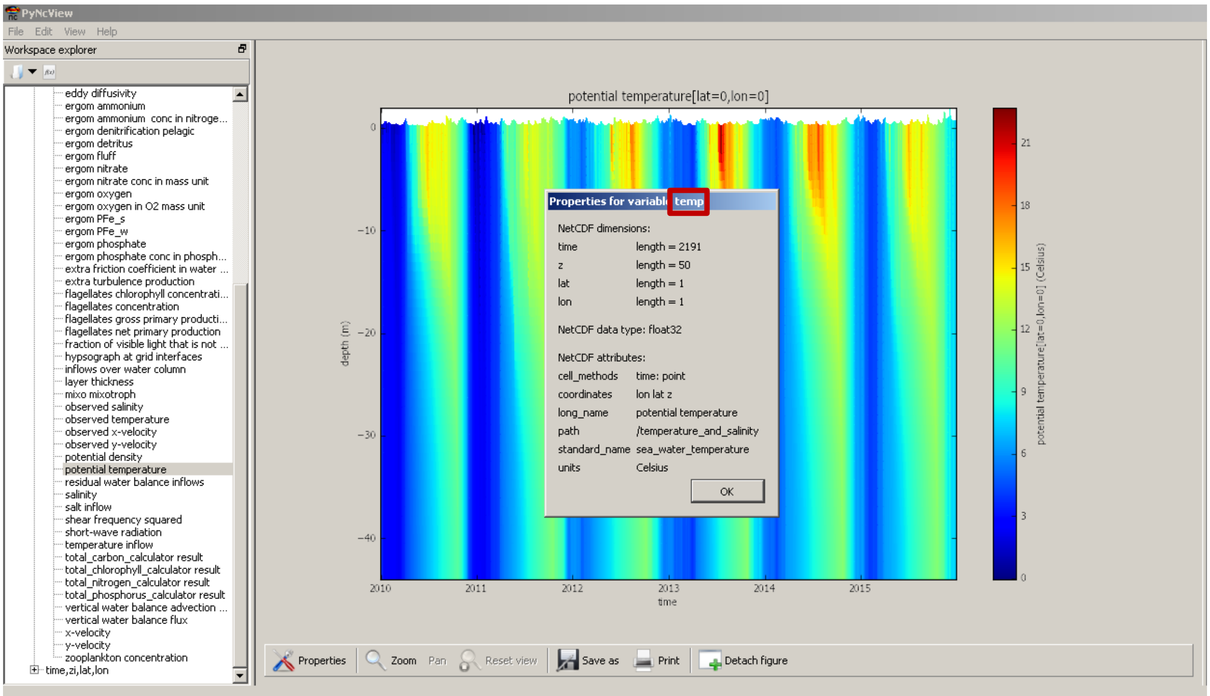

Fig 9. Properties of potential temperature with the red box showing the position of the name to be used in the ‘output.yaml’ file.

An equation can be built within PyNcView by using the equation builder  in the top left corner. To highlight the difference between the modelled and the observed you can use (potential temperature)-(observed temperature).

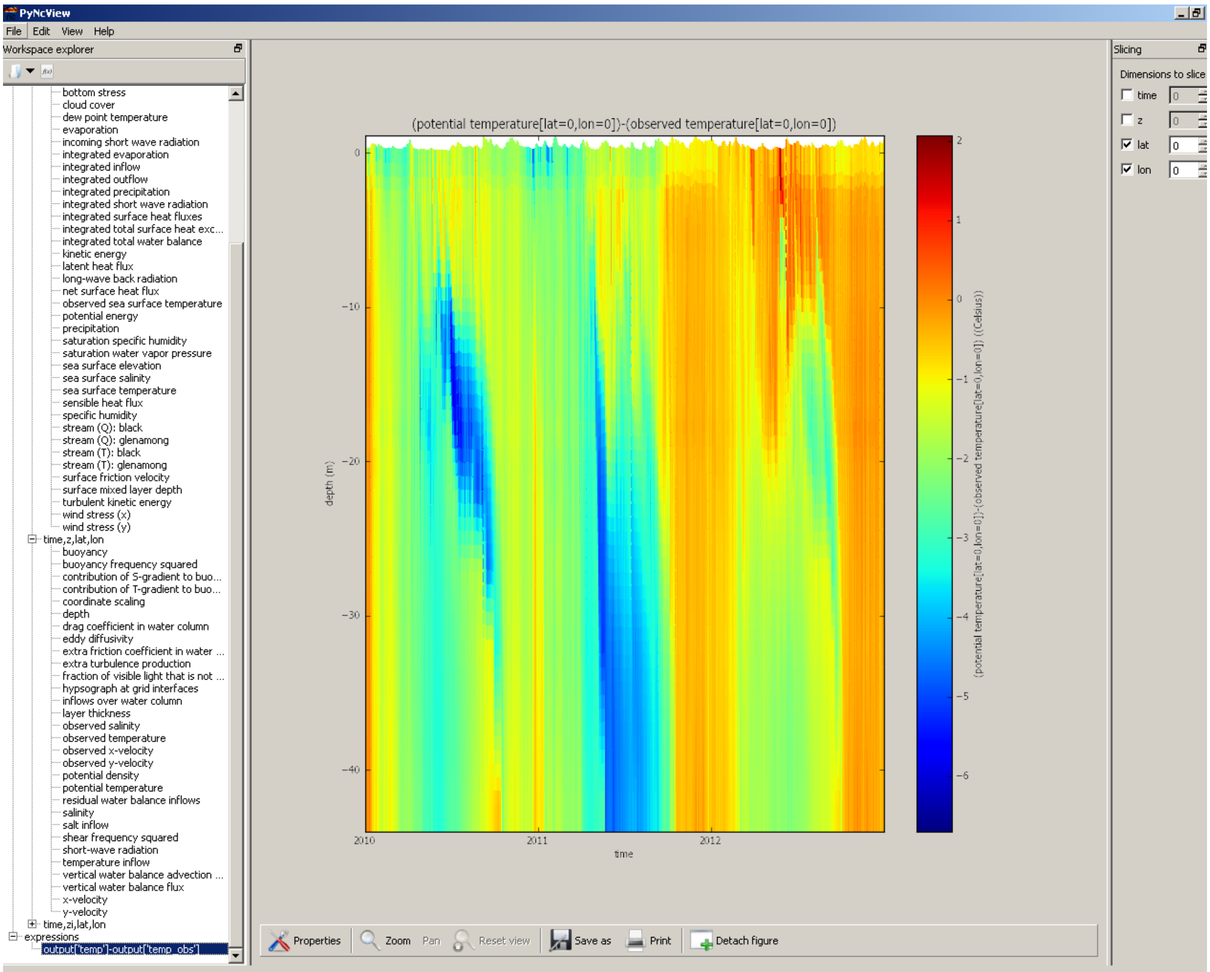

Fig 10. Temperature plot showing the difference between modelled and observed temperature.

This is useful to give an insight into how the model is performing.

Section 5. Extracting Data for further analysis

Note: I work primarily in R so I have written code which enables me to do all my modelling operations within R. This is replicable in Python or MATLAB.

All the model data is stored in the netCDF file. This can be accessed using R but the problem with this is that the netCDF file can be very large especially if you are modelling a long time period. Then it makes sense to only output the files you need i.e. modelled temperature and observed temperature. This can be done within the netCDF or the text files.

output_analysis.yaml | Mod_temp_hr_temp.txt | Mod_temp_hr_z.txt

```
# title: GOTM Simulation
# variable: potential temperature (Celsius)
# dimensions: lon=1,lat=1,z=1:50
# fixed coordinates:
#   longitude: -9.5000000 degrees_east
#   latitude: 53.900000 degrees_north
# rows: time
# columns: depth (for values see output\Mod_temp_hr_z.txt)
# time  1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18
2010-01-01 00:00:00 4.5608220   4.5608220   4.5624174   4.5694374   4.5764573
2010-01-01 01:00:00 4.5608221   4.5608311   4.5624481   4.5694412   4.5764605
2010-01-01 02:00:00 4.5608222   4.5608323   4.5624515   4.5694407   4.5764599
2010-01-01 03:00:00 4.5639787   4.5641603   4.5657416   4.5712817   4.5777102
2010-01-01 04:00:00 4.5658763   4.5660738   4.5693994   4.5737284   4.5791728
2010-01-01 05:00:00 4.5703943   4.5708215   4.5722526   4.5750132   4.5801815
2010-01-01 06:00:00 4.5732646   4.5735291   4.5744437   4.5771160   4.5815871
2010-01-01 07:00:00 4.5758475   4.5760876   4.5767322   4.5786242   4.5820752
2010-01-01 08:00:00 4.5771667   4.5773047   4.5776226   4.5784526   4.5794535
2010-01-01 09:00:00 4.5734847   4.5730935   4.5724708   4.5717197   4.5709165
2010-01-01 10:00:00 4.5642433   4.5635572   4.5631368   4.5628186   4.5625192
2010-01-01 11:00:00 4.5557487   4.5553111   4.5548003   4.5542926   4.5538175
2010-01-01 12:00:00 4.5475210   4.5469913   4.5462513   4.5455307   4.5448494
2010-01-01 13:00:00 4.5387899   4.5381197   4.5372952   4.5365960   4.5359791
2010-01-01 14:00:00 4.5303928   4.5297391   4.5290612   4.5285081   4.5279623
2010-01-01 15:00:00 4.5232177   4.5226669   4.5220761   4.5215866   4.5211005
2010-01-01 16:00:00 4.5169500   4.5164467   4.5158736   4.5153876   4.5149317
2010-01-01 17:00:00 4.5107466   4.5102191   4.5096180   4.5091194   4.5086499
2010-01-01 18:00:00 4.5036912   4.5030785   4.5024088   4.5018675   4.5013319
2010-01-01 19:00:00 4.4950135   4.4942819   4.4935297   4.4928910   4.4922602
2010-01-01 20:00:00 4.4844607   4.4836293   4.4828096   4.4820894   4.4813845
2010-01-01 21:00:00 4.4725269   4.4716552   4.4707922   4.4700373   4.4693010
2010-01-01 22:00:00 4.4601226   4.4592636   4.4583823   4.4576342   4.4569026
2010-01-01 23:00:00 4.4476230   4.4467737   4.4458647   4.4451158   4.4443786
2010-01-02 00:00:00 4.4345242   4.4336277   4.4326418   4.4318417   4.4310506
2010-01-02 01:00:00 4.4215987   4.4207041   4.4197286   4.4189358   4.4181606
2010-01-02 02:00:00 4.4098322   4.4090038   4.4080874   4.4073330   4.4066171
2010-01-02 03:00:00 4.3989355   4.3981368   4.3972252   4.3964594   4.3957576
```

Fig 11. Text modelled water temperature output from GOTM.

Along with the output water temperature a file is output with the depths of each column. It is the same name as the temperature output file with '_z' at the end of the name to indicate depths. This file allows depth values to be attributed to each of the layers. These text files can then be loaded into R and use the package 'rLakeAnalyzer' to analyze the data.