Arctic Thermokarst Model Project Description: Phase 4 – Final Report

Rawser W Spicer

2017-12-29

# Introduction

The Arctic Thermokarst Model (ATM) models thermokarst disturbances in the Alaskan arctic and boreal forests, as discussed in more detail in the introduction to the report on [Phase 1](https://github.com/gina-alaska/arctic_thermokarst_model/blob/master/documents/Arctic_Thermokarst_Model_Phase_0_1.docx) [1]. Phase 1 and 2 focused on the expanding the capabilities, and improving the data structurers for the ATM. Phase 3 tackled control file structure of the model. The original goal of phase 4 was to integrate all of these pieces into an updated more streamlined code base. While this goal was accomplished many other issues that were over looked earlier had solved. This made phase 4 the longest and most comprehensive part of the project to date.

# Phase 4: Integration

The first 3 phases of this project worked toward expanding the potential capability and improving the code base by designing better data structures to represent the model and its inputs. These were fairly successful on their own but for the most part they did not work together. Phase 4 was originally designed to solve this problem in a very straight forward way. Many issues arose that made it more complicated than originally intended, but that ultimately produced a better product. The GitHub [project board](https://github.com/gina-alaska/arctic_thermokarst_model/projects/4?)[2] and [issues list](https://github.com/gina-alaska/arctic_thermokarst_model/projects/4?)[3] are available online.

One of the first issues to arise was that the data to run the model for the entire Arctic Coastal Plain(ACP) needed to be gathered and processed in to input formats the ATM could use. This involved finding the temperature data for the area, and the input surface area information. The data was easy to find, the surface area input exists as a shape file that was included in the initial model code base, and the historic state wide temperature data was available from the Senarios Network for Arctic + Alaska Planning (SNAP) [Data Portal](http://ckan.snap.uaf.edu/dataset/historical-monthly-temperature-1-km-cru-ts)[4]. Both of these data sets were in the wrong format to work with the model, so tools had to be developed to convert them. For converting the shape file to a set of raster files, the Geospatial Data Abstraction Library [GDAL](http://www.gdal.org/)[5] was used, and a python utility was developed that can be found at *atm/tools/generate\_raster.py*.

For the temperature data several tools had to be developed. The first one which also used GDAL clipped the state wide raster files to the proper bounds for the ACP. This tool is found at *atm/tools/raster\_clip.py* contains functions that can be called by a user. The original version of the model was able to use temperature data in this format to calculate the Thawing and freezing degree days used in the ATM. The new frame work lacked this ability. In fact it lacked any type of structure for handling the temperature of degree day data (meteorological or met data). In order to speed up the development of the model I decided that the new framework would initially only handle preprocessed degree day data. A tool was created to preform this preprocessing. The tool can be called as a utility and be viewed at *atm/tools/calc\_degree\_days.py*. This utility takes advantage of multiprocessing and Numpy memory mapped arrays to speed up processing. The tool generates two data files one which freezing and thawing degree day data.

Creating the input degree day data revealed that there was no new object to represent any met data the model may use. This data is similar to the other data used in the model in that it is still spatial, but differs in that it is all known, or pre-calculated. A base class was designed to support any kind of historical met data that the model may need to use. This class can take a saved Numpy array type and load it as a Numpy memory map that the class then allows the model to access though the \_\_getitem\_\_ function using the year desired as the index. The input file type may be a single file with the data (new style as generated by the calc\_degree\_days tool), or a list of data files (the old-style degree data as seen in the barrow input). Child classes were created for the freezing and thawing degree data. Developing this object showed that it was necessary to differentiate between the initialization year (when the input area rasters represent) and the start year (the first year the model runs from based on met data start year).

In addition to adding the new met grids, several other new grids had to be developed during the integration. The first was a grid to represent the Lake pond depth. This grid (*atm/grids/lake\_pond\_grid.py*) contains information on how deep lakes and ponds are in the grid elements and how thick the ice freezes each year. This grid is important the lake to pond and pond to lake transitions. The next grid was a grid to represent the drainage efficiency. This gird (*atm/grids/drainage\_grid.py*) contains weather each grid cell is above or below the drainage threshold, which is used in the POI based transitions. The final grid added was for simulating climate events. This grid (*atm/grids/climate\_event\_grid.py*) divides the main grid into climate blocks and determines whether a climate event occurs in the block or not for each year. The climate events are used the lake pond expansion functions.

With all the grids squared away it was possible to develop the transition functions. Three transition functions were developed: POI based, Lake to pond, and Pond to lake. These can be seen In *atm/checks.* These transition were based on code by Bob Bolton. I took Bob’s transition check code and found that there were three common sections. I used these sections to create the three transition methods. To accomplish this several fields had to be added to the cohort control files. These are Transition\_check\_type which should be POI, lake\_to\_pond, or pond\_to\_lake, and transitions\_to, which should be the canon cohort name that the cohort becomes. The cohorts transitions can now be easily specified in the control files with out major changes being needed in the code.

After the main transitions were finished 3 special case transitions had to be rewritten to use the new objects. These were the lake drainage by climate event, lake pond expansion and pond infill. This code is in *atm/climate\_events.py* and *atm/lake\_pond\_expansion.py.* These methods are streamlined versions of Bob’s original code. The lake pond expansion function was the most tricky and had several iterations before the final was decided upon.

Finally *atm/ATM.py* had to be updated. The code in this file was easy to updated, but several other changes had happed to stream line the code. The first was converting all of the control files to yaml files. This allowed transition order to be easily be defined as a list. It also allowed further refinement to the function parameters for the POI based transition functions. It also means that all of the yes or no values in the control files were automatic converted to true and false. The next was that the figures and movies had to be add to the grids. Each figure or movie needed was added to the proper object and then called by the ATM model class if the output was requested. The final was rewriting the text log and results archiving feature. The text log is was created by using a common string to write or save the results, and the results archive was rewritten with in the main ATM class.

Once every thing was completed the code was cleaned up by removing any old files and checking that each file had been properly documented. The model was tested using the original Barrow test area, and the new Arctic costal plain area. The control files for theses setups are archived in *example\_control\_files.* The readme and [wiki](https://github.com/gina-alaska/arctic_thermokarst_model/wiki)[6] have been updated to include more information on the model.

# Conclusion and Moving Forward

During the course the ATM framework was updated and expanded using computer science principals for improved maintainability and scalability. Some new features include the ability to generate inputs for new model areas, proper object-oriented programming, use of functions and other methods to reduce duplication, and better documentation. I believe that all of these things greatly improve the model. The Input data and Initial Results can be accessed on [google drive](https://drive.google.com/drive/folders/1LIWLwiMtuuMxhaZwV8LztJngzJIB1YHf)[7] by anyone with a UA email address

There are still some improvements the ATM could use as I see it. The first would be better user documenting. This could include better documentation of the control files, and better documentation of how to run the model. A tool to aid in generating the control files would also be useful. Improvements to the tools for the model is another thing that could make the model easier to use. These could include making more of the tools in to stand alone command line tools, or integrating the tools in to the model workflow. A third thing the model could use is an expanded selection of transition functions. Bob has already developed new streamlined transition workflow that could be added as a new transition type, or just adding further functions for the POI based transition. These are just some of the possible improvements I see for the ATM, other can be found at the [Future Tasks project board](https://github.com/gina-alaska/arctic_thermokarst_model/projects/5?) [8].

# Authors Note

I hope that my improvements to the code will prove useful, and I really enjoyed working on this project. I look forward to opportunities to work on this model in the future

– Rawser Spicer.

# References

[1] Phase 1 status report: <https://github.com/gina-alaska/arctic_thermokarst_model/blob/master/documents/Arctic_Thermokarst_Model_Phase_0_1.docx>

[2] Phase 4 project: <https://github.com/gina-alaska/arctic_thermokarst_model/projects/4>?

[3] Issues: <https://github.com/gina-alaska/arctic_thermokarst_model/milestone/4?closed=1>

[4] State Wide Historical Temperature data From SNAP: <http://ckan.snap.uaf.edu/dataset/historical-monthly-temperature-1-km-cru-ts>

[5] GDAL: http://www.gdal.org

[6] wiki: <https://github.com/gina-alaska/arctic_thermokarst_model/wiki>

[7] model data: <https://drive.google.com/drive/folders/1LIWLwiMtuuMxhaZwV8LztJngzJIB1YHf>

[8] Future Tasks: <https://github.com/gina-alaska/arctic_thermokarst_model/projects/5>?