Arctic Thermokarst Model Project Description: Phase 0 – Phase 2

Rawser W Spicer

2017-09-25

# Introduction

The Arctic Thermokarst Model (ATM) models thermokarst disturbances in the Alaskan arctic and boreal forests. This state-and-transition model has been designed to simulate thermokarst initiation and expansion in these ecosystems. The Alaskan landscape is split into grid cells that are defined from raster files (model input). Within each grid cell, the ATM tracks the fractional area of landscape cohorts, which are landscape units mainly characterized by vegetation composition, age, hydrology, soil texture and other environmental factors. The ATM utilizes yearly time steps to update the distribution of the landscape cohorts. See [this presentation on the Arctic Thermokarst Model](http://arcticlcc.org/assets/products/ARCT2010-05/presentations/Bolton_Alaska_Themokarst_Model_20151026.pdf)[1] and [this poster](https://csc.alaska.edu/sites/default/files/Yujin-Zhang_AGUposter_final1.pdf) [2] for more information. The ATM is designed to run over the period of hundreds of years and thus may be utilized to predict the effect of climate change on the ecosystem structure and function. This paper is a brief introduction to the fall 2017 effort to archive, document, and expand the ATM.

# Phase 0: Archiving the ATM

A primary development goal for fall 2017 is to create a frame work that allows the ATM to be expanded to the entire Arctic coastal plain. Before this could be accomplished there were several issues that needed to be addressed; we define these tasks as Phase 0. First, we needed to verify the ATM will run execute on various computing platforms (Linux, Mac). Second, a simple refactoring of the code to separate it from the inputs used and outputs generated would make future calculations easier to track. Third, we needed to outline what code modifications were necessary for the ATM coastal plain expansion with an eye towards the expansion to the boreal forest regime.

The ATM package was developed by Dr. Bob Bolton (IARC) and written in Python. The ATM package contains all of the inputs, current outputs, and ancillary files (such as ATM presentation and documentation). The ATM is easy to run, but to make it portal to any computer, an Anaconda environment was set up to install the dependencies. The ATM is executed by making a simple change to a path in a file *Control*, and running “*python ATM.py Control*” from the parent ATM directory. This file contained the information needed to run the ATM in a two-year test state for Barrow. The outputs are dumped into the *Outputs/Barrow* directory and archived as a time-stamped tarball in *Outputs/Barrow/Archive*. At this stage, an archive of the code was created.

Before any further work commenced, we created a git repository on GitHub to archive code changes (https://github.com/gina-alaska/arctic\_thermokarst\_model). The ATM code was separated from the other files by moving all python files in the code directory provided an atm/ subdirectory. A Changelog, readme, a copy of the *Control* file, and file describing the Anaconda environment were also included in the git repository. There are three items necessary for the ATM to run: The file *Control*, and the directories *Input*, and *Output*. These items were moved to a directory named *atm\_data*. All other files, and directories were moved a directory named *atm\_support\_files*. The *arctic\_thermokarst\_model*, *atm\_data*, and *atm\_support\_files* were all moved to the same top-level directory. The ATM could now be run by changing to the *atm\_data* directory, adjusting the path in the *Control* file, and running “python ../arctic\_thermokarst\_model/atm/ATM.py Control”.

The main structure for running the Barrow test simulation is contained in *run\_barrow.py*. As Barrow was the most complete test case for the ATM, this code would serve as the basis for the expansion of the model to the entire Arctic coastal plain. The structure of this code was linear, and fragments of code were often repeated in a checking apparatus. These checks determine if and how each ATM element changes over time. It was determined that this would be the best place to refactor the code to make it extensible to this larger geographic area.

Finally, areas of the code that could be improved by further refactoring were the overall structure of the ATM object, the internal representation of the landscape being modeled, handling of inputs and outputs, and the execution of the ATM itself. The initial code for ATM is contained in one large object. It would need to be refactored to classes and functions that represent one thing or task. Related to this is the internal representation of the area to model. This is currently very disparate with a separate attribute for each type (cohort) of land. These should be reorganized into a structure that contains and allows access to all of them. This will be a focus of Phase 2. Strategies for improving the other issues are to be determined.

# Phase 1: Expanding to the Arctic coastal plain

Phase 1 of this project is the code refactoring to expand the ATM capabilities to allow for the modeling of the entire Arctic coastal plain. The projects and issues features on GitHub.com were used to track the implementation of this phase. These tools will continue to be used for future development.

Eight issues were identified for this task:

1. Decide how to track ATM output for comparison between changes (git issue 1).
2. Provide a central access point for initial input data.
3. Create a function to run any set of cohort checks, provided to the function as a list of strings (git issue 2).
4. Create no-age cohort checks where necessary, as age data is not available for all of the Arctic coast (git issue 3).
5. Create a metadata object to allow the mapping of the cohort checks to the strings provided to the new run function from #3 (git issue 4).
6. Refactor the cohort checks in to a submodule (git issue 5).
7. Delete any Python files marked as a backup of something (git issue 6).
8. Verify the changes, and deploy as a new version. (git issue 7).

The first two issues were easy to solve. The ATM already archives the outputs, and initially these could be used for comparisons. It was later decided that piping the output from the terminal program could also be used. To accept model revisions (via regression testing, git issue 7), it was also determined that several options in the various input files would need to be changed to create repeatable results (no random generation of processes). The entire initial input archive was added to a central volume for current access to the input data. This may need to be revisited as the ATM is improved.

A copy of *run\_barrow.py* was created as *run\_general.py* - where the new run function was created. This function loops through a list of cohort checks (strings) provided, accessing each of the check functions via the metadata object that maps the stings to functions to run. One check function had to be changed so that its function call would work in this loop (the objective of that function did not change). One file *check\_climate\_event.py* was not included in checks, as it has different functionality, and was causing issues with imports when moved. During the initial test of this new method, using Barrow, it was discovered that an initialization function, unique for each area, was run. The solution was to provide the function as a parameter to the run function. This solution is not ideal and should be changed. Issues #3 and #5 had now been solved.

All of the python files *check\_<something>.py* were moved to a submodule named *checks*. *check\_<cohort>\_NA* functions were added to all of the wetland tundra files. These were created as placeholders for no-age cohort checks, and do nothing. All of the files with backup (and homers confilicted copy) in their name were deleted. This solved #4, #6, and #7. The metadata object from issue #5 was further improved by moving the metadata object to the *\_\_init\_\_.py* file in the checks submodule.

Once all of the other tasks were completed the code was verified. A few areas of ghost code, code that exists because of a copy and pasted but does nothing, were cleaned up. The metadata on the version and authors was moved to the *\_\_init\_\_.py* file, the version number was changed to 0.2.0, and the version number was added to the output report. Once the changes were accepted, the code was merged to master and tagged as 0.2.0. A tag was also added to the initial commit as 0.1.0 to reflect the version initially given to the code.

# Issues Encountered

Some minor issues were encountered during Phase 0 and Phase 1. The most major was that initially the git repo was set to track remotely on githhub.alaska.edu. Bob Bolton does not have access to that, so it was changed to track on github.com. All of the initial issues had to be migrated manually. The other issues encountered documented before include. The initialization issues with the run function, and the *check\_climate\_event.py* issue.

# Phase 2: Moving Forward

Phase 2 of the project will include changing the internal representation of the area being modeled by creating a new object and refactoring the check functions common functionality. Further things to complete are archive a set of the control files, revisiting where we are keeping the data to run the ATM, and dealing with the initialization portion of the run function. It will also need to be determined what checks need to be run for the Arctic coastal plain.

# References

[1] Presentation on ATM: <http://arcticlcc.org/assets/products/ARCT2010-05/presentations/Bolton_Alaska_Themokarst_Model_20151026.pdf>

[2] Poster on ATM: <https://csc.alaska.edu/sites/default/files/Yujin-Zhang_AGUposter_final1.pdf>