Understanding the canopy growth model

By Flann Chambers

The "readme_USER.pdf" tutorial is a pre-requisite to this tutorial.

This tutorial is destined to users of the canopy growth model, who wish to modify the source code of the program or perform the simulations over a region of their choice. Knowledge of python, the arcpy module, ArcGIS Pro tool creation, and GAML programming will be required for this tutorial.

A detailed report of the work done on this project is available ("CanopyGrowthModel_Report.pdf"). It may help you understand how the model source code works.

The model requires a shapefile of the location of the trees to model, and for each tree, the following attributes are required. The name of the attribute will have to correspond to the string in parentheses, in order for the model to detect it (model source code line 113).

- Plantation date ("DATE_PLANT")
- Category ("Categorie")
- Canopy diameter ("DIAMETRE_C")
- Total height ("HAUTEUR_TO")
- Trunk height ("HAUTEUR_TR")

The height of the tree is only relevant for canopy volume computations.

The model also requires a basemap (.png file) of your chosen region of interest (for instance, a screenshot from ArcGIS Pro), and a CSV file of the parameters (named "allCalibrationParameters.csv") generated by the calibration process. Model source code lines 102 to 110 let the program know the value of the world area based on the chosen region of interest. It is mandatory to update these lines whenever a new region of interest is chosen.

The model comes with a collection of python scripts that will help you generate these files.

1) Generating the category attribute in your shapefile.

We will now work under the assumption that each tree in your shapefile has a species attribute, and that you possess an Excel or CSV file linking the species of tree with a category : small, medium or large tree. This file needs to adopt the following format : the first column contains the name of the species, and the second column contains an integer that is either 1, 2 or 3; 1 corresponding to a large tree, 2 to a medium tree, and 3 to a small tree. The file "SpeciesCategoryCorrespondanceTableExample .csv" is an example of such a file. It contains all the data for tree species over the whole canton of Geneva.

The script "appendCategoriesToAttributeTable.py" will add a Category field to your shapefile, and will fill it in for each tree based on its species. Its inputs include the workspace, the CSV file previously mentioned, the layer corresponding to your shapefile in ArcGIS and the field corresponding to the species inside your layer. Create a new tool with these inputs in ArcGIS Pro, then link this tool to the "appendCategoriesToAttributeTable.py" script and run the tool with the specified inputs.

2) Selecting trees for calibration.

For calibration, the following attributes are required.

- Category
- Plantation date
- Observation date
- Total height
- Trunk height
- Crown diameter*

*if using LIDAR data, instead of crown diameter, the calibration may also be done directly on canopy area. We recommend using the formula for the area of a circle if need be.

The script "selectCalibrationTrees.py" will select all trees from your shapefile which possess data for all of these attributes. It will return a CSV file of all the selected trees with only the aforementioned attributes. Its inputs include the following :

- Your workspace path
- The layer containing your trees
- The category field name inside your layer
- The plantation date field name
- The observation date field name
- The total height field name
- The trunk height field name
- The crown diameter* field name
- The name of your chosen region of interest.

Create a new tool with these inputs in ArcGIS Pro, then link this tool to the "selectCalibrationTrees.py" script and run the tool with the specified inputs. At the end of the script, the selected trees are highlighted for visualization purposes. Do not forget to clear the selection once you have finished working. The following CSV files are a direct product of this python script, and are available as a reference :

- CalibrationCSVdata_Bastions.csv
- CalibrationCSVdata_ParcDesRois.csv
- CalibrationCSVdata_Plainpalais.csv

3) Calibration process.

The python script "CalibrateModel.py" is intended to be run as a standalone program. The script will ask for the file generated at step 2. It will also create a folder named "CalibrationOutput" in the same directory as the script. Successively for each category of tree, the script will brute force through all combinations of values in a given range for each one of these following key parameters :

- Maximum canopy diameter
- Maximum canopy height
- Maximum lifetime
- r ratio
- Age at plantation date
- Growth stress multiplier

You will immediately recognize which part of the program this corresponds to, "thanks" to the nested *for* loops. It is a good idea to re-write this code using the *itertools* module, if time allows it.
The range of each parameter can be directly edited inside the *for* loop declarations.

For each combination of parameters and for each tree, the script calculates the canopy diameter and canopy height corresponding to the age of the actual tree inside the calibration CSV file generated at step 2. The script calculates the absolute value of the percentage difference between the computed values, and the values recorded inside the calibration CSV file. Then, these percentages are averaged and this average will be used to evaluate the performance of the current combination of parameters.

The calibration script outputs three different CSV files, one for each category, inside the newly created "CalibrationOutput" folder. Each output file has 7 columns :

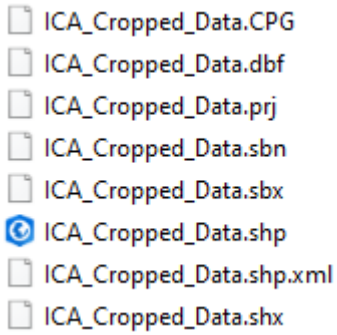
- The average error on canopy diameter
- The average error on canopy height
- Maximum canopy diameter
- Maximum canopy height
- Maximum lifetime
- *r* ratio
- Age at plantation Date
- Growth stress multiplier

The rows correspond to the 20 best experiments, i.e. combinations of parameters that minimize the average error on canopy diameter, ranked from best to worst. The user may then select the combination of parameters of their choice, and copy-paste them into the file "allCalibrationParameters.csv" under the relevant category row. This is to be done for each category.

4) Transferring all the relevant files to the "includes" folder.

The model file (.gaml extension) is located inside the "models" folder. Its parent folder also contains a folder named "includes". This is where all the relevant files for the model are to be added.
The following files are needed. Their name needs to correspond to the name in parentheses.

- CSV file of the calibration parameters ("allCalibrationParameters.csv")
- Basemap image ("basemap_Cropped.png")
- Shapefile containing your tree data ("ICA_Cropped_Data.shp" and associated files, see below).



You should now be all set to run the model on the region of your choice.

5) Output formatting.

During the simulation, two columns of data are saved at every cycle to a CSV file located inside the "includes" folder of your model, and named "saved_canopy_rates.csv". This file is automatically created if it does not exist yet. The first column represents the year of the simulation and the second column represents the simulated canopy rate.

You can use python to produce graphs of canopy rate in function of time easily and in an aesthetically-pleasing manner. The python script "drawOutputCharts.py" does exactly that, and is intended to be run as a standalone program. It will prompt you for "saved_canopy_rates.csv" files. The program lets you draw multiple canopy rate curves on the same graph. Don't hesitate to edit this script in order to fit your needs.