

# Computational model of auxin-CUC interaction in the floral meristem

Shuyao Kong, Mingyuan Zhu, David Pan, Brendan Lane, Richard S. Smith, Adrienne H. K. Roeder

12/07/2023

## Summary

The floral meristem is modeled using a 2D disk of cells that are growing and dividing, based on two previous models, GridAuxin and MassSpringCells. Mechanics are modeled using a mass-spring model where cell walls (springs) are pulling cell wall junctions (masses). Each cell has auxin, CUC, and PIN, that are produced, degraded, and interacting, calculated using a differential equations solver. The model is implemented in C++, and can be run inside the VLab environment<sup>1</sup> and the MorphoDynamX software<sup>2</sup>. The model can be run either one simulation at a time, or batch-run using a python code.

## System requirements

The model requires a Linux machine with NVIDIA CUDA (version 11.40) and the Thrust library. MorphoDynamX ([www.MorphoDynamX.org](http://www.MorphoDynamX.org); available upon request) and VLab ([http://algorithmicbotany.org/virtual\\_laboratory/](http://algorithmicbotany.org/virtual_laboratory/)) are required for running the simulations. We suggest using R for downstream data parsing.

For the demo, we used vlab (version 5.0, build #3609) and MorphoDynamX (version 2.0, revision 2-1395, CUDA version 11.40) in an Ubuntu 20.04.6 LTS system, equipped with an Intel Core i9-10900X 3.7 GHz 10-Core Processor, G.Skill Trident Z RGB 256 GB DDR4-3600 CL18 Memory, and a NVIDIA GeForce RTX 4090 24 GB Graphics Card. For downstream data processing, we used RStudio (R version 4.3.1 (2023-06-16) -- "Beagle Scouts").

## The model folder

The folder "MassSpringAuxin" contains the source files for the model.

- "specifications.txt" lists all the files within the folder to be included with the model
- "CellDisk.hpp" and "CellDisk.cpp" are the model source files, and can be edited using a text editor or IDEs such as VS Code.
- "CellDisk.mdv" can be edited using a text editor, and contain metadata of the model, including which initial condition to use.
- "Init\_blank\_72cells\_2023-09-07.mdxm" is the initial condition used in the model. This can be changed to another MDX mesh file.

As detailed below, the model folder can be opened and compiled in VLab Browser, and run in MorphoDynamX, so no installation is needed.

### Running the model one simulation at a time

1. Open the VLab Browser and select the oofs repository.
2. Navigate to the model, and double click on the window to open it.
3. Right click on the model icon, and select “run”. The model would now open in MDX.
4. Set the desired parameters, mainly in Process -> Model -> 01 Cell Disk and Process -> Model -> Auxin Gradient Model. If output is desired, set “Run name”, “Output directory”, and “Output frequency”.
5. Double-click Process -> Model -> 01 Cell Disk to run it. Click on the stop button to stop when desired. Typical run time for one simulation is around 2-5 minutes, depending on the performance of the computer and how many iterations are desired.

### Running the model in batch mode

1. Using a python code, such as the provided “create\_run\_20231010\_demo.py”, create a run file, such as the provided “run\_20231010.py”. The former file specifies the parameter combinations to be tested, while the latter file contains the commands to be fed into MDX.
2. Open the model in MDX following steps 1-4 in the previous section.
3. Under Process -> Tools -> Python -> Python Script, select the run file created in step 1 (“run\_20231010.py”). The model would now run automatically.
4. Screenshots are automatically saved in “output directory/run name/Screenshots/\*.jpg”. Cell complex data are automatically saved in “output directory/run name/CCData/\*.txt”. The following data are saved in these files: Cell index (CCIndex), x and y coordinates, cell area, and auxin and CUC concentrations. Each row represents a cell.
5. In the demo, 5 simulations each of WT, *drmy1*, *cuc1*, and *drmy1 cuc1*, with growth rates 0.4, 0.8, and 1.2 were run, and the results were saved in the directory “20231010” (in Github, these are uploaded as five zip files, 20231010 \*.zip). The simulations were named as follows: “Run\_CUC Prod\_SD Auxin Prod\_PIN low CUC\_PIN high CUC\_Plasticity\_Unchanging Noise\_sim number\_date”.

### Parsing and analysis of CCData

1. The code “Parse\_output\_2023-10-10\_demo.R” parses these data files, identifies auxin maxima, tracks them over time, and summarizes them in “RData\_20231010.RData”. It also plots how auxin maxima are assigned and tracked. These plots are saved in “output directory/run name/CCData/\*.png”.
2. The data frames in the output RData file can be analyzed as desired. For example, the provided code “Modeling\_analysis\_2023-10-10\_demo.R” reads in the RData file and generates three plots in the “Plots” folder.

## References

1. Federl, and Prusinkiewicz (1999). Virtual laboratory: an interactive software environment for computer graphics. In 1999 Proceedings Computer Graphics International, pp. 93–100. 10.1109/CGI.1999.777921.
2. Strauss, S., Runions, A., Lane, B., Eschweiler, D., Bajpai, N., Trozzi, N., Routier-Kierzkowska, A.L., Yoshida, S., Da Silveira, S.R., Vijayan, A., et al. (2022). Using positional information to provide context for biological image analysis with MorphoGraphX 2.0. *eLife* 11, e72601. 10.7554/eLife.72601.