ODD Model Description of Monodominance Model

1. Purpose

Tropical forests are highly diverse ecosystems. However, within many of such forests large regions of monodominance of a single tree species occur. It remains an open question whether a generic mechanism might exist that might explain monodominance across systems. With this model we want to explore whether two basic mechanisms (large seeds and poor dispersal of the monodominant species) are sufficient to describe the emergence and maintenance of clusters of monodominance within a diverse forest.

2. Entities, state variables, and scales

Entities of the model are trees, grid cells and seeds. State variables of trees were: species id, x- and y coordinates of its grid cell, seed mass, and dispersal range. State variables of grid cells were: x- and y coordinates, id of the tree which is on grid cell or, if the grid cell is not occupied by a tree, the number and identity of seeds. Seeds were characterized by species id and seed mass.

A grid cell could either be occupied by a tree or be empty, i.e. a gap in the canopy which occurred when a tree died in the previous time step. Trees belonged to different species, seven of them being identical and one being the candidate for monodominance. Species where distinguished by seed mass and dispersal range.

Seed mass: 'Seed mass' (*SM*) was defined as the energy (in arbitrary energy units) a species invests per grid cell within its dispersal range. In our simulations the seed mass of the non-monodominant species was kept constant and that of the monodominant species was varied. So in all resulting figures "seed mass" refers to the monodominant species.

Dispersal range: Dispersal range was defined as the radius of a circle around a tree. The tree can drop its seeds in all grid cells within this circle, but only seeds landing on empty grid cells were considered. Grid cells had four neighbors, so that for dispersal radius one four neighbor cells were within the range, while a dispersal radius of 20 equaled 1256 cells in range.

The monodominant species had a low dispersal range of 1 grid cell and 4 seeds with high seed mass, while the other seven species were long-distance dispersers with a default dispersal radius of 20 grid cells and seeds with poor seed mass.

The model world consisted of a grid of grid cells. Default grid size was 512x512 grid cells, and we used toroidal, or periodic, boundary conditions, i.e. the model world was wrapped into a torus. Time step length represented the time between death of a tree and the establishment of the next adult tree on that patch. Simulations were usually run for 10,000 time steps.

3. Process overview and scheduling

Each time step the following three steps were performed:

(1) Disperse seeds into empty patches:
Seed input to each empty grid cell was determined. For this, seed energy input from all trees within the dispersal range from the empty cells was calculated (Fig. 1). Parameters were scales so that each tree in the relevant neighbourhood could contribute only one seed per empty cell.

(2) Recruitment:
For all empty grid cells the species to establish was chosen via lottery competition, i.e. the chance for each species to establish was proportional to the sum of their seed energy on this grid cell.

(3) Mortality:
All trees not established in this time step could die randomly according to a mortality rate which was constant and the same for all species.

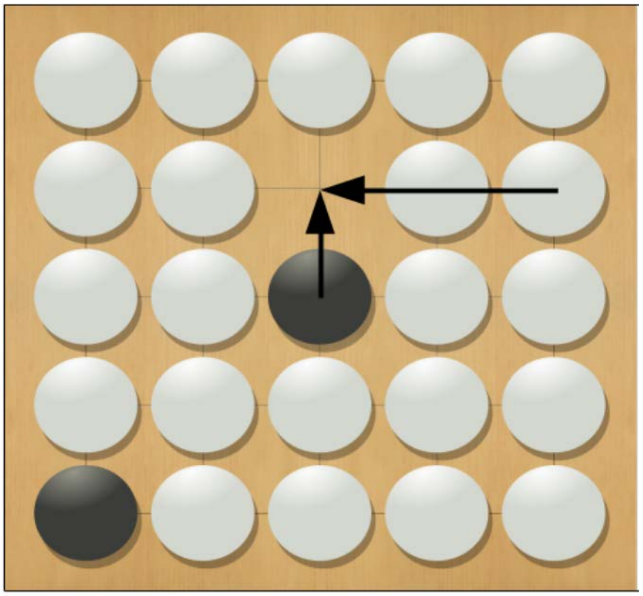


Figure 1: White pebbles symbolize trees of one of the 7 non-monodominant species, black pebbles are monodominant trees. The gap in the middle is to be colonized. The species to establish here is chosen via lottery competition, i.e. the chance for each species to establish was proportional to the sum of their seed energy in this grid cell. For default parameterization, the monodominant species has an energetic advantage over its non-monodominant competitors:
white: dispersal range = 20 grid cells, 1256 grid cells in range, default seed mass = 20 per grid cell
→ total seed mass produced per tree = 25120
black: dispersal range = 1 grid cell, 4 grid cells in range, default seed mass = 9450 per grid cell
→ total seed mass produced per tree = 37800

4. Design concepts

Basic principles
We base the representation of the species-rich forest matrix, within which monodominance can emerge, on Neutral Theory (Hubbel 2001): all non-monodominant species were identical. All have the same possibility of dying, the same growth rate etc. We did not, however, apply the concept of a meta-community, as new recruits are drawn from the seedbanks of the empty patches and these seedbanks are filled by all trees having the patch within their dispersal range. Therefore, if a species goes extinct during the simulation, it cannot re-establish via immigration as it is the case for default neutral models.

Emergence
In the course of the simulation, clusters of monodominant species occurred. The phenomenon of clustering can be seen as a consequence of the small dispersal range of the monodominant species, however whether and where clusters actually emerge and if they can persist over time and to which size they grow depends on the parameter set chosen and is not a deterministic process.

Interaction
Interaction among trees only happens in empty grid cells where new trees are recruited. The interaction – or competition – between trees there is indirect because trees only interact via their seeds.

Stochasticity
Colonization of an empty grid cell is decided by lottery. The sum of the recruitment probabilities PR_k for all species with $k \in [1,8]$ always equals 1 and the probability for each tree species is given an interval I_k , where

.

Note, that if e.g. k = 1, the interval is $I_1 = [0...PR_1]$. Using this, a random number *r* (between 0 and 1) is drawn and if $r \in I_k$ the species *k* is recruited.

Observation
The results of the simulations were analysed after 10.000 years. We have recorded the following properties:

- the relative abundance of the monodominant species
- the number and sizes of monodominant clusters
- whether or whether not there is a percolating cluster (a system spanning cluster, either occupying all positions in x- or in y- direction or both).

To characterize cluster, the following metrics were used (for details, see section "Metrics for characterizing monodominant clusters", which follows this model description):

- the cluster size
- the cluster's position (the position can either be given by listing the sum of all positions of patches being part of the cluster or by calculating the center of gravity)
- the fractal dimension d_F, measuring the complexity of the cluster's surface
- the radius of gyration, measuring the mean distance of the cluster's elements (hence the grid cells) to its center of gravity.

Additionally, spatial snapshots of the model world were taken at regular intervals to visualize cluster dynamics.

5. Initialization

At the beginning of a simulation, all eight species were distributed randomly over the whole grid, so the proportion of each species was 12,5%. All parameter values of a species are set equal, there is no small stochastic fluctuation within individuals of one species. Furthermore, the seven non-monodominant species are identical but the monodominant species differs in dispersal range and seed mass. Initialization was always the same for all simulations performed.

6. Input data

No external data sources were used as input to the model.

7. Submodels

The submodels are comprehensively described in the section “Process overview and scheduling” above, except for the recruitment submodel:

Recruitment of new trees

In an empty grid cell, the next tree is recruited out of a seed bank which contains all seeds of all trees having the current grid cell within their dispersal range. The mass of the seeds is taken into account, such that the recruitment probability of one tree species is given by the proportion of seed mass of the current species versus the sum of all seed masses of all species within the seed bank.

The recruitment probability PR_k for each PFT was calculated for each cell in the grid by dividing the sum of the seed input from all adjacent monodominant trees by the seed input from all trees having the respective cell within their dispersal range:

$$PR_k = \frac{seed\ mass \cdot count\left[species_k \in dispersal\ range\left(species_k\right)\right]}{\sum_{i=1}^{maxspecies} seedenergy\left(species_i\right) \cdot count\left[species_i \in dispersal\ range\left(species_i\right)\right]} \quad ,$$

with *k* being the number of the concerning species.

Metrics for characterizing monodominant clusters

Cluster algorithm

This algorithm is used to count the number and the sizes of clusters at the current time step. It is applied every 50th time step.

For doing so, first the grid which normally consists of grid cells given numbers from 1-7 (if occupied by a non-monodominant tree), 8 (if occupied by a monodominant tree) or 0 (if being a gap) is transformed into a grid only possessing grid cells with values of 0 (if a monodominant tree is present) or -1 (everything else).

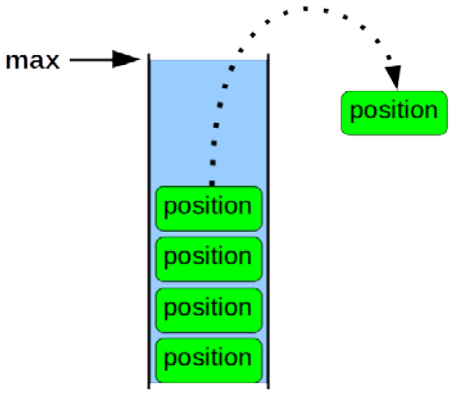
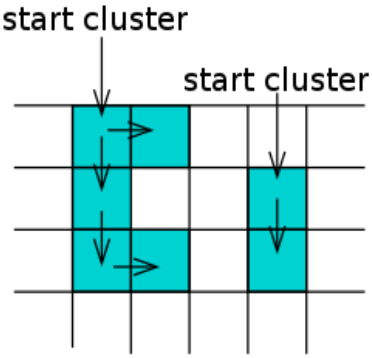


Figure 1: A stack is used as data structure in the cluster algorithm. Here, the „first in, first out“-principle is used. When scanning over all grid cells, every monodominant cell adjacent to one of the current cluster is put onto the stack and is given the number (starting from 1) of the current cluster. To empty the stack, the grid cell, situated on top is taken off the stack.

algorithm cluster

```
begin
        while there is a not treated grid cell t do
        begin
                put t onto stack
                start new cluster with t
                while stack not empty do
                begin
                        take grid cell c from stack
                        for all  neighbours n of c do
                                if n monodominant and not treated then
                                put n onto stack and add to cluster
                                give n the number of the cluster (“treated”)
                end
        end
end
```

Figure 2 shows the algorithm: the whole grid is scannend from top-left to bottom right. When first a monodominant grid cell (a grid cell having the number „0“) is found, a new cluster is started and given the number „1“. The monodominant grid cell is given the cluster's number.



All adjacent monodominant grid cells are put onto the stack of this cluster and as well given the cluster's number of „1“ instead of „0“.

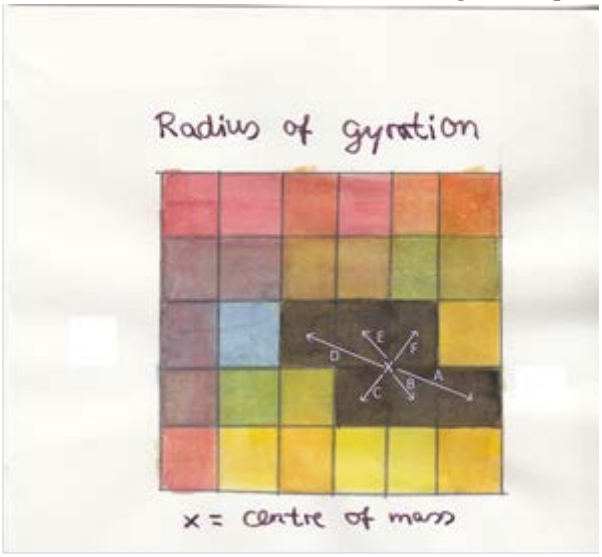
After doing so the first element is drawn off the stack and all grid cells adjacent to the actual patch are checked: are there monodominant patches? If yes, they are put onto the cluster's stack and are as well given the cluster's number. When all neighbouring grid cells are treated, the next patch is taken off the stack.

This process is continued until the stack is empty, thus all monodominant neighbours are given the cluster's number. Then, the routine of scanning all grid cells from top-left to bottom right is continued until the next untreated monodominant grid cell is detected, the clustering algorithm starts again, now with the next cluster number („2“).

These processes are continued until the whole grid is scanned and thus all monodominant grid cells are defined as being part of one cluster.

Radius of Gyration

In physics, the radius of gyration is used to describe the spatial extension of irregular objects, like for example a polymer chain. We here use the radius of gyration to calculate both, the correlation length (correlation length equals the radius of gyration of the biggest, non-percolating cluster, which serves as a good estimate for SM_c as the correlation length is expected to diverge at SM_c) and as well the fractal dimension, being the slope of the size of each cluster over its radius of gyration.



The formulas used to calculate the radius of gyration can be found in the Supplementary material. Here, we only point out, how we dealt with the grid structure and periodic boundaries to estimate the radius of gyration correctly.
We used the cluster algorithm described above to consecutively calculate the mean of all x- and y-positions of all grid cells belonging to one cluster. This equals the center of mass, marked by an „x“ in Figure 3.

As all positions of trees were discrete (trees were placed in the center of each grid cell), the euclidean distance between two trees was calculated by using the row and column numbers of each grid cell (integers) as coordinates (Figure 3). Thus, all calculations were performed with integer values.

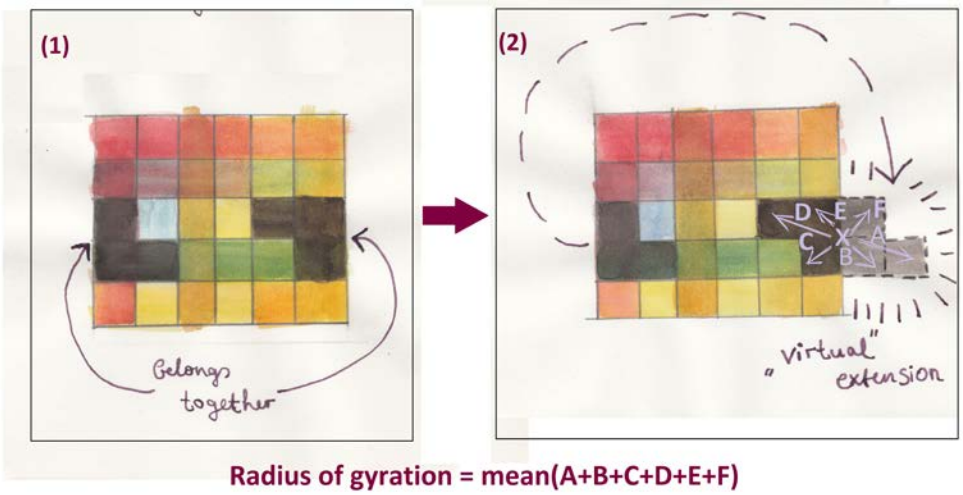


Figure 4: Challenges of the periodic boundaries: if clusters are placed at the border of the grid, the radius of gyration cannot be calculated by using the grid cell's row and column variables to calculate the euclidean distance. Instead, one part of the cluster has to be „virtually flipped“ from the one side to the other.

If a cluster was placed at the border of the grid (in x- or y-direction) and therefore separated into two (or more) pieces among the grid, the euclidean distance could not be calculated by simply using the row and column numbers as coordinates for each grid cell. Therefore, one part of the cluster had to be „virtually flipped“ to the other side, using negative coordinates or coordinates above the world-size. The grid cell was put onto the stack (as described for the cluster algorithm above) and aditionnally it was marked that the current position had been flipped (by using two different variables to distinguish between flipping in x- or in y direction).

When the „flipped“ grid cell was taken off the stack again to search for monodominant neighbours, all neighbours of the „flipped“ grid cell, were automatically flipped in the same direction, too, unless one grid cell had to be flipped back in the opposite direction (thus, when it previously had been marked with „-x“ and now it should be flipped in „+x“ direction). In this case, the marking „flipped“ was removed for this grid cell. However, if a grid cell was e.g. first marked „-x“ and then flipped in y-direction „+y“, both markings persisted, the „-x“ direction could only be deleted by „+x“.
Using this method we first calculated the center of mass (which could like this also be situated outside of the grid) and then the mean of all euclidean distances between center of mass and all grid cells belonging to the cluster.