

An Agent-based, Interactive Simulation Model of Root Growth

Marc Mußmann¹, Daniel Nicolas Hofstadler², and Sebastian von Mammen¹

¹Julius Maximilian University, Germany

²University of Graz, Austria ma_musmann@posteo.eu

Abstract

Plants are complex organisms, showing collective adaptive behavior. Plant behavior is often defined by shoot growth, yet root systems exhibit equally complex, less visible, behaviors. Roots have to navigate in a particular environment, while optimizing nutrient and water uptake as well as avoiding exposure to harmful elements. In this paper, we introduce an interactive, agent-based simulation model of root growth. It supports the exploration of the interplay of different root models within different environments. To this end, we resort to Swarm Grammars (SGs) which combine the interactivity of spatial agents with the generative perspectives of L-Systems. We pursue a point-based representation of the environment due to its versatility with respect to modeling and rendering possibilities. SGs and point-based environments are combined in a simulation that enables interactions between a user, agents and the environment at runtime. We validate the model by recreating and analysing several established root model configurations, and validate the benefits of the interactive simulation by an expert interview.

Introduction

The development of plants is an example of collective adaptive behavior, where the shoot and also the roots react to changes in their environment. Both parts of the plant are growing in very different environments. While the complex behavior of the shoot is easily observable, the ways in which the root system of a plant finds its way through the soil environment is almost always obscured. Root systems adapt their architecture and change their growth direction to optimize nutrient and water uptake, while avoiding exposure to harmful elements. We created a model that allows for interactively exploring the ensuing dynamics. To this end, we extend SGs which combine the interactivity of spatial agent-based models like Boids, which retrace the flocking behavior of birds, with the reproductive generative perspectives of L-Systems, which simulate generative reproduction based on rewriting symbolic strings (Leitner et al., 2010; Hamon et al., 2012). SGs have previously been used as the basis for interactive simulations of plant growth (von Mammen et al., 2017). We enhance the deployed SG agents to interact with a point-based environment model. Point clouds are often

used to display digitised versions of real environments and provide great modeling flexibility because of their simplicity (Schaufler and Jensen, 2000). Next to agents that represent the plant’s root system, we also introduce water agents that interact with their surroundings by colliding with obstacles and emitting water, which also allows to model absorption and usage of nutrients. The combination of agents and points allows one to manipulate both the root growth model and the environment, thus affording rather large degrees of freedom of interactivity. Game engines integrate various functionalities that support the development of interactive, real-time software applications. Therefore, we used the popular game engine Unity for this research project (Unity Technologies, 2024). Like preceding root models, we provide means to track and export model results and to analyse them at runtime. We deem these functionalities crucial especially in interactive contexts, as they document one’s explorations’ results. We validate the model by recreating and analyzing several root systems of plants using the implemented simulation. The implemented simulation, especially with respect to its interactive capabilities, is evaluated from an expert’s perspective. The source code of the simulation is published under the MIT license on a Github repository¹.

In this paper, we first outline related work and relevant background information. Here, we highlight how root systems develop and in which ways they react to their surroundings. Additionally, different existing root simulation models are introduced. Next, we detail the developed simulation context and root growth simulation model. Subsequently, we present and analyze four different model configurations of plant root systems generated by our model. We evaluate the introduced model and consider its benefits of interactive simulation by means of an expert interview, which we report next. We conclude with a summary and a brief outline of potential future work.

Related Work

In this section, we first elaborate on several factors that affect the growth of root systems. Second, we present different

¹<https://github.com/mamusmann/RGS>

existing root simulation models, before we outline SGs and their underlying concepts.

Root Behavior

The distribution of nutrients in the soil can be uneven and change over time. As a result, the root system architecture (RSA) is modulated by plants that can alter root morphology at millimeter scales in response to fine-scale variability in nutrient availability. Plant roots take up several elements from the soil such as phosphorus (P) and iron (Fe). The availability of these elements as nutrients to the roots is not only subjected to some spatial distribution but it also depends on water conditions. This results in dynamic changes of nutrient availability which the plant's root system has to accommodate. The heterogeneity of nutrient concentration generally affects the length, number, angle and diameter of the primary root (PR), lateral roots (LR) and the development of root hairs (Aibara and Miwa, 2014). Additionally, root growth exhibits tropistic responses, or tropisms, that are defined as directional growth towards (positive) or away from (negative) a directional stimulus (Muthert et al., 2020). These responses happen at the tip of the root during growth. Roots can show several tropisms, including reactions to gravity (gravitropism), salt gradients (halotropism), chemical signals (chemotropism), magnetic fields (magnetotropism), touch (thigmotropism), water (hydrotropism) and even sound (phonotropism). While gravitropism is often the dominant tropism, it can be overridden by other tropisms. In general, the existence and strength of tropisms differ between plant types (Muthert et al., 2020).

Simulation models

Modeling plant root system development is inherently complex and there can be major differences between plant species. As a result, several root models emerged to drive computational simulation. In the following, we outline two recently published, extensive simulation models that build upon earlier versions.

The OpenSimRoot software is an extended, open-source re-implementation of the SimRoot model. SimRoot is the most feature-rich and highly cited functional-structural root simulation model, its last full description was published in 1997 (Lynch et al., 1997). OpenSimRoot is implemented in C++ and uses an object-oriented design. By introducing the concept of mini models, the implementation encapsulates the simulation of a single state variable embedded in a simple hierarchy. The implemented model comprises the simulation of RSA, the shoot, carbon (C), water and nutrient acquisition and utilization, root growth plasticity and geometric descriptors (Postma et al., 2017).

CRootBox is a flexible framework to model root architecture and its interactions with soil environments. CRootBox is based on an earlier model named RootBox (Leitner et al., 2010) and is implemented in C++, providing flexi-

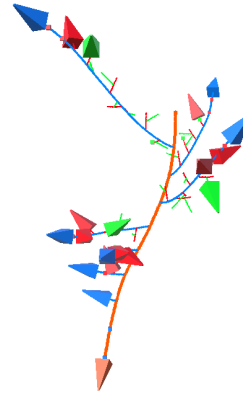


Figure 1: Each active agent is represented as a pyramid shape pointing into its movement direction. Its color depends on its type.

ble extendability through Python bindings (Schnepf et al., 2018). CRootBox supports gravitropism and chemotropism and calculates the tropism direction at the root tip by means of a stochastic optimization process. This process takes one of n vectors, randomly distributed in a cone, that minimizes the related tropism function at the tip of a root. In addition to tropisms, responses to environmental conditions are simulated. CRootBox's flexibility extends to user-defined scaling functions, allowing fine-tuned simulations of root properties such as elongation rate, branching angle, and density within varied soil environments. In CRootBox, each root that can have LRs is divided into three zones: a basal zone, a branching zone and an apical zone. After the basal and apical zones are developed, the LRs can emerge in the branching zone. The CRootBox software implementation supports data export in various formats, including the Root System Markup Language (RSML) (Lobet et al., 2015; Schnepf et al., 2018).

Swarm Grammar

SGs (von Mammen et al., 2017) are an extension of L-Systems (Prusinkiewicz and Lindenmayer, 2012) and bring together their abilities of reproduction and structural generation with the agent-based interactivity of Boids (Reynolds, 1987). Boid agents retrace self-organised group transport as seen in schools of fish or flocks of birds. Each agent merely reacts to its perceived neighbors and environment. More concretely, each agent accelerates at each simulation time step $t \in \mathbb{N}$ considering a weighted sum of different urges towards or away from its neighbors (Reynolds, 1987):

$$\vec{f}_t(b_i) = w_s \vec{s}_t(b_i) + w_a \vec{a}_t(b_i) + w_c \vec{c}_t(b_i) \quad (1)$$

where b_i is the i^{th} Boid agent of a total number of n agents. The urges \vec{s}_t , \vec{a}_t , \vec{c}_t determine an agent's separation from its peers, its alignment with its peers, and its cohesion towards the geometric center of the neighborhood, respec-

tively. These urges are weighted by the respective coefficients $w_s, w_a, w_c \in \mathbb{R}$.

L-Systems were originally developed to capture growth in bacteria and yeast cells (Lindenmayer, 1968). Later, Lindenmayer and Prusinkiewicz developed and extended this language to model plants (Prusinkiewicz and Lindenmayer, 2012). L-Systems can be defined as a tuple (V, w, P) , with V defining an alphabet of symbols, w the axiomatic symbol, and P the set of production rules. A single production rule is composed of (p, s) , with p being the predecessor and s the successor (Hamon et al., 2012). For context-free, so-called 0L-Systems, $p \in \Omega$ is a symbol of an alphabet Ω and $s \in \Omega^*$ represents a word over Ω or an empty symbol. At each step of the derivation process, the predecessors are replaced by their successors. The resulting L-strings can be geometrically interpreted. SGs extend L-Systems in such a way that individual symbols represent spatially interacting agents, whereas the production rules determine their reproduction and construction behavior (Jacob and von Mammen, 2007). In the original L-Systems, structural topologies and interactions among neighbors are embedded into the production rules. In contrast, in SGs, the arising dynamics, the structure and form of the generated artefacts also heavily depend on the agents' individual behaviors and the interplay with the simulated environment, both of which can quickly be changed and be affected by the agents' activities themselves (Raies and von Mammen, 2021). In addition, the agent-based approach with its threefold design of sensing/processing/acting allows for interactivity with the environment and the user's input (von Mammen et al., 2017).

Root Growth Simulation Model

We embed the SG-based root model into a point cloud model environment that supports heterogeneous material distributions and fast interaction of nutrient and water uptake. This further facilitates the interactive modeling and simulation efforts by the user of the system. In this section, we describe the point cloud environment, introduce water agents that simulate dissipation and uptake, and finally we detail the root agents' properties and behaviors.

The points that represent the environment can have sets of type/value-tuples attached representing root material, soil particles, nutrients, obstacles or other elements such as harmful metals that can be found in the soil. In this way, we can have rather detailed heterogeneous spatial configurations, whereas the deployed model agents can react to specific information relevant to their behavior. Soil particles can store water, that is lost over time using a simplified evaporation model with loss of w_{evap} water over time. The deployed agents can generally be classified as model-based reflex agents which have sensors, actuators and an internal state (Russell, 2010, p.50). The agent simulation is discretized into time steps defined by the variable dt . Water agents represent the simplest agent instances in our model.

They are not driven by any production rules (different to water agents in (Raies and von Mammen, 2021)) and only react to environmental point data, which they perceive within a given field of view defined by a perception angle α_w and a radius r_w . All agents in our model accelerate based on perceived stimuli. In analogy to the stochastic direction selection implemented in CRootBox (Schnepf et al., 2018), we also calculate a number of n_{tries} possible directions in a cone given by angle α around the agent's current direction. From these directions one that minimizes/maximizes a specific function is followed that is not leading to a collision. If necessary, the process is repeated considering the search space from α to $\alpha + d\alpha$ until a maximum angle of 180° is reached. At this point, the agent would simply stop moving. Water agents always select from a downward cone, whereas the maximum angle is smaller than $(\frac{3\pi}{4} rad)$ to prevent traveling upwards. The optimal vector for the water agents is the direction with the largest dot product against the gravity vector. All agents use basic kinematic variables velocity and acceleration that are bound by the initial velocity v_{init} , minimum and maximum velocities v_{min} and v_{max} , and a maximum acceleration value a_{max} . Water agents use a fixed acceleration value a_w , whereas the root agents' acceleration value is calculated based on different urges or tropisms. In terms of state, the water agents carry a specific amount of water defined by $w_{content}$. During the simulation, the water is emitted to the soil points within a radius w_{radius} . The exact amount is given by:

$$loss_w(dist) = \min\{w_{loss}dt(1.0 - \frac{dist}{w_{radius}}), w_{content,i}\} \quad (2)$$

where $dist$ is the distance from soil point to agent and $w_{content,i}$ is the current agent i 's available water amount. Soil points can take water until saturated, which is specified by a constant set to 1.0 in our implementation. Water agents are deleted, when depleted or the simulation boundaries are reached.

The root agents are located at the tip of a root branch, as shown in Figure 1. As seen in Figure 1, the SG agents place points with color c_r and radius r_r which depend on their respective type. Agent types are represented by a unique symbol Δ and indicate a specific type of root branch. Root agents are deleted when the maximum root branch length l_{max} or lifetime value rlt is reached. For more realistic variation, these values can be assigned a standard deviation. Their field of view extends by radius o_{radius} beyond the radius of the root geometry r_r . Another radius, r_{sound} , defines the maximal distance for sensing sounds that the root agents can react to (phonotropic response). Data from the environment points is used for calculating collisions, scaling factors, and tropisms.

Next to steering variables, the root agents' internal state holds information about the root branch, including the current root branch length in form of the travel distance, delta

travel distance, root agent type (Δ), parent emergence distance, and branching distance. This information is required for the zone-based branching behavior. Lateral roots (LRs) only emerge at distances of l_n within the interbranching zone, which lies in-between the basal (l_b) and apical (l_a) zones, and emerges after the branch segment has outgrown the first basal and respective, subsequent apical zone. Branching directions are determined using an insertion angle θ and a radial angle r , similar to the CRootBox and OpenSimRoot models. Our model incorporates a gravitropic set-point angle (GSA) parameter gsa , enabling the simulation of LRs that grow at an angle to the gravity vector, a behavior observed in real plants (Muthert et al., 2020). The GSA calculates the optimal growth direction relative to gravity for each root type. The production rules that trigger branching are modulated by a probability br_p as in a probabilistic L-System. In contrast to the standard L-System convention, our model uses probabilities to determine whether an agent's type remains unchanged or is switched to a different type. In the latter case, one applicable production rule is selected based on an associated frequency value, as proposed in (Raies and von Mammen, 2021). In analogy to the type-switching production rules, there are rules that can create new lateral branches. These rules are governed by two factors. The first is a general branching probability br_p , which controls overall branching likelihood. The second is the production rule frequency $freq$. This is part of the tuple $(w, freq)$ contained in a list of possible production rules p . The chosen approach allows having multiple possible production rules and a single value that can be modified to specify whether and how much lateral branching may occur. Agents are replaced by the empty word and deleted, if (1) a branch reaches its maximal length, (2) an agent exceeds its lifespan, or (3) a parent branch is deleted. An agent's activity can also be paused until a certain condition is met. For example, production rules may only be applied, if certain nutrients are available in the plant as a whole. Production rule nutrient costs are defined by the $req_{nutrients}$ parameter, present for each agent type.

Our model provides scaling factors computed by linear scaling functions of the form presented in Eqn. 3, where x is the average point cloud point type value around the root agent. These scaling functions simulate the plasticity of root growth and branching strategies. The implemented factors can be divided into two types: (1) the GSA ones that scale the root length and (2) the ones that scale based on the surrounding points. The GSA factors rely on a scaling function taking the *slope*, a minimum length and the current root length as parameters. The result is clamped between a given s_{min} and s_{max} . The other types include scaling of root length, branching angle, branching density, branching probability, elongation, gravitropism, and GSA. Each scaling factor is computed using the average of multiple scaling functions. A scaling function represents the reaction to a

specific point type. Each function uses the agent's surrounding average value of the specified point type x , a *slope* and minimum concentration c_{min} value. The average is used to combine multiple scaling function results into a single scaling factor value. The values are clamped between s_{min} and s_{max} . The effects of scaling functions and tropisms on the RSA are evident when comparing Figures 2 and 4. Note the root system's differing responses to the uneven and uniform environments.

$$f(x) = \max\{s_{min}, \min\{a, s_{max}\}\},$$

$$\text{with } a = \begin{cases} \text{slope}(x - c_{min}) + 1, & \text{if } x \geq c_{min} \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

The model includes additional tropisms that allow roots to respond to various environmental stimuli, such as halotropism w_{halo} , chemotropism w_{chem} , magnetotropism w_{magn} , thigmotropism w_{thig} , hydrotropism w_{hydr} , and phonotropism w_{phon} . Tropism directions are calculated based on the random optimization technique used by all agents, with a tropism specific function determining the best of n_{tries} directions. Each tropism direction is normalized to a length given by the unit tropism vector magnitude t_{mag} . The specific implementation of each tropism function can be found in the provided source code. Collectively, these acceleration directions, along with the w_{sepa} -weighted separation urge for collision avoidance, determine the growth direction chosen by individual root agents within our simulation model.

An agent's specific growth is accelerated with vector \vec{a}_t of magnitude $a_{mag,t}$ as defined in Eqns. 4 and 5. It is recalculated at each time step t as the sum of all considered, relatively weighted tropisms as in Eqn. 6.

$$a_{mag,t}(b_i, \Delta_i) = \max(0, \min(|\vec{a}_t(b_i, \Delta_i)|, a_{max}(\Delta_i))) \quad (4)$$

$$\vec{a}_t(b_i, \Delta_i) = \frac{\vec{a}_t(b_i, \Delta_i)}{|\vec{a}_t(b_i, \Delta_i)|} a_{mag,t}(b_i, \Delta_i) \quad (5)$$

$$\begin{aligned} \vec{a}_t(b_i, \Delta_i) = & w_{grav}(\Delta_i) \overrightarrow{grav}(b_i, \Delta_i) \\ & + w_{halo}(\Delta_i) \overrightarrow{halo}(b_i, \Delta_i) \\ & + w_{chem}(\Delta_i) \overrightarrow{chem}(b_i, \Delta_i) \\ & + w_{magn}(\Delta_i) \overrightarrow{magn}(b_i, \Delta_i) \\ & + w_{thig}(\Delta_i) \overrightarrow{thig}(b_i, \Delta_i) \\ & + w_{hydr}(\Delta_i) \overrightarrow{hydr}(b_i, \Delta_i) \\ & + w_{phon}(\Delta_i) \overrightarrow{phon}(b_i, \Delta_i) \\ & + w_{sepa}(\Delta_i) \overrightarrow{sepa}(b_i, \Delta_i) \end{aligned} \quad (6)$$

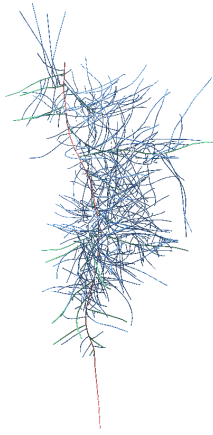


Figure 2: Rendered image of a simulated *Helianthus annuus* (common sunflower) root system, showing effects of tropisms and scaling functions.

In the equations, $\Delta_i \in \Omega$ is the type, b_i a concrete instance of root agent i , and the vectors represent the respective, abbreviated tropisms. The resulting velocity is limited by an agent’s general kinematic variables as outlined above, whereas a root agent’s acceleration is bound by a maximum value $a_{max,r}$.

Next to agent-specific parameters, there are control parameters at the plant-level. They include the axiom w that specifies the initial root agents of a plant, the nutrient parameters of its seed n_{seed} , encoded as tuples of point type and value, as well as a plant’s nutrient storage n_{req} , encoded as triplets of point type, the minimum storage amount required for the plant, and the storage capacity of each root segment. Furthermore, the parameter n_{shoot} defines the nutrient exchange via tuples of point type and nutrient transfer per time step.

Since the presented simulation model approximates root growth, we have certain limitations. Similar to other models, we only consider primary and lateral roots. We only simulate primary, not secondary growth in terms of root cambial thickening. The model abstracts the absorption of nutrients and water in variables n_{absorb} and w_{absorb} , respectively, and water transport and storage in $rW_{capacity}$. We define the nutrient uptake to be limited to a radius of $n_{overlap}$ around the root segment and the stored water as the uptake of nutrients requires water defined by $n_{reqWater}$. Distances are represented in centimeters, angles in degrees. Values for time and nutrient uptake have, for now, not been aligned with real-world scales but fixed to represent reasonable relationships and to support the desired interactivity of the system.

Model Configurations

In this section, we present model configurations designed to validate and illustrate the modeling flexibility of the

model introduced above by recreating and analyzing diverse plant root systems. Our configuration parameters are informed by values published by Forschungszentrum Juelich GmbH (2024) and refined through visual comparison with CRootBox-generated root systems. Root systems were generated in similar environments without simulating nutrients absorption and growth costs.

Table 1 shows an overview of point types deployed in the simulated environment. Table 2 lists the parameters of the water agents, Table 3 the environmental parameters. Tables 4 and 5 reveal plant-specific parameters for two example specimens, whereas Table 6 lists the respective root agents configurations.

Index	Name	PC	NC	Absorbable by Plant
0	Root			
1	Soil			X
2	Obstacle			
3	NaCl			X
4	Phosphorus	X		X
5	Harmful Metals		X	
6	Carbon			X
7	Fe	X		X

Table 1: Configuration of point data available in the simulation. Positive chemotropism (PC) and negative chemotropism (NC) are used to determine tropistic reaction. Absorbable by Plant defines if this point is taken into account when simulating nutrient absorption. While Carbon is not used directly as a point in the point cloud, it is used as a resource being transferred from the shoot.

Symbol	Value	Symbol	Value
r_w	2	$oradius$	2
$v_{init,w}$	50	t_{mag}	1
$v_{max,w}$	50	dt	0.01
a_w	50	$n_{overlap}$	1
$n_{tries,w}$	5	n_{absorb}	0.001
α_w	25	w_{absorb}	0.0001
$d\alpha_w$	15	$n_{reqWater}$	0.001
w_{radius}	7.5	w_{evap}	0.1
w_{loss}	1	$rW_{capacity}$	1
$w_{content}$	250		

Table 2: Configuration of water agent parameters.

Table 3: Environment configuration values.

We selected four contrasting plant species to demonstrate our model’s ability to simulate RSAs reflecting fundamental evolutionary differences (monocot vs. dicot) and ecological pressures (domesticated crop vs. wild plant). For compara-

Symbol	<i>Anagallis foemina</i>
w	H
n_{seed}	$\{(4, 0.8), (6, 0.8), (7, 0.8)\}$
n_{req}	$\{(4, 0.05, 0.001), (6, 0.1, 0.001), (7, 0.05, 0.001)\}$
n_{shoot}	$\{(4, -0.001), (6, 0.25), (7, -0.001)\}$

Table 4: *Anagallis foemina* plant parameter values.

Symbol	<i>Helianthus annuus</i>
w	A
n_{seed}	$\{(4, 1), (6, 1), (7, 1)\}$
n_{req}	$\{(4, 0.05, 0.001), (6, 0.1, 0.001), (7, 0.05, 0.001)\}$
n_{shoot}	$\{(4, -0.001), (6, 0.25), (7, -0.001)\}$

Table 5: *Helianthus annuus* plant parameter values

bility in time and space, all are herbaceous annual flowering plants. Within the angiosperms (flowering plants), monocotyledons (monocots) form a distinct group encompassing grasses and grass-like plants. Our selection includes two monocot representatives: the wild grass *Crypsis aculeata* and the domesticated maize (*Zea mays*). Monocots are characterized by their seeds having only a single embryonic leaf (a cotyledon), ultimately leading to a fundamentally different RSA from the dicotyledons (dicots), which have two cotyledons. Our dicot representatives are the wildflower blue pimpernel (*Anagallis foemina*) and the domesticated crop, the common sunflower (*Helianthus annuus*). Below, we provide brief descriptions of each species and their specific root model configurations.

***Anagallis foemina* (recently renamed *Lysimachia foemina*)** A low-growing wild flower commonly known as blue pimpernel, *A. foemina* is found in diverse habitats, thriving particularly on dry yet nutrient-rich soils. As a dicot, its root system centers around a prominent primary taproot (PR) from which LRs branch out. In our model, the 1st order LRs have a high GSA ($gsa_I = 90^\circ$), leading to a relatively even distribution around the primary axis established by the taproot. Compared to the other modeled species, *A. foemina* has a relatively small root system dominated by GSA-bound 1st order LRs. Figure 3 provides a visual representation of the root distribution in a homogeneous environment, while Figure 5 illustrates the temporal dynamics of root growth, including the time delay induced by the apical zone. Tables 6 and 4 provide the main parameter values used to generate this RSA.

Symbol	<i>Anagallis foemina</i>	<i>Helianthus annuus</i>
Δ	H, I, J, K	A, B, C
c_r	$(1, 0.35, 0), (0, 0.52, 1), (1, 0, 0.9), (0, 1.0, 0.16)$	$(1, 0.49, 0.49), (0.45, 0.87, 0.63), (0.4, 0.64, 0.98)$
r_r	0.05, 0.03, 0.02, 0.02	0.023, 0.023, 0.008
r_{sound}	50	50
r_{lt}	1e9	1e9
br_p	1	1
p	$H \xrightarrow{1} I, I \xrightarrow{1} J, J \xrightarrow{1} K$	$A \xrightarrow{3} B, A \xrightarrow{7} C, B \xrightarrow{1} C$
$v_{init,r}$	1, 1, 0.2, 1	1, 1, 0.2
$v_{min,r}$	0.1, 0.1, 0.2, 0.1	0.1, 0.1, 0.2
$v_{max,r}$	1	1
$a_{max,r}$	1, 2	1, 2
α_r	90, 80, 45, 18	70, 45, 50
$n_{tries,r}$	10, 10, 10, 1	4
$d\alpha_r$	10, 20, 25, 25	10
θ	0, 70, 68, 68	0, 74.48, 70
l_b	1, 0.4, 0.1, 0	1, 0
l_n	0.8, 0.5, 0.24, 0	0.4, 0.4, 0
l_a	1.2, 0.8, 0.1, 0.4	10, 5, 10
l_{max}	33.4, 7.7, 0.44, 0.4	101.46, 15, 10
w_{grav}	1, 1, 0.1, 0	1, 1, 0
gsa	0, 90, 0	0, 74.48, 0
w_{halo}	1, 2, 2, 1	1, 2, 1
w_{chem}	1, 2, 2, 0	1, 2, 1
w_{magn}	1, 0.25, 0	1, 0.25, 0.25
w_{thig}	1	1
w_{hydr}	0.2, 0.1, 0	0.2, 0.5, 0.5
w_{phon}	0.1, 0	0.1, 0
w_{sepa}	0.25, 0.25, 0	0.5, 0.25

Table 6: Parameter values for *Anagallis foemina* and *Helianthus annuus* without standard deviation values. These and the additional scaling and $req_{nutrients}$ parameter values can be found in the source code. Values are shown in the order of root agent types Δ . Duplicate entries at the end of value sequences are omitted.

Helianthus annuus The root system of the common sunflower, a domesticated crop thriving in managed, fertile soils, shares similarities with *A. foemina* as a dicot. Both feature a prominent PR and first-order laterals with a high GSA ($gsa_B = 74.48^\circ$ in our model). However, the sunflower model features a unique characteristic: non-gravitropic lateral roots (ng-LRs) that can emerge directly from the primary root (as 1st order laterals) or from typical, GSA-bound 1st order laterals (as 2nd order laterals). Indeed, ng-LRs

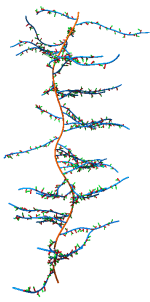


Figure 3: Rendered result of the *Anagallis foemina* root system.

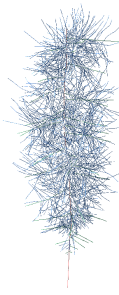


Figure 4: Rendered result of the *Helianthus annuus* root system.

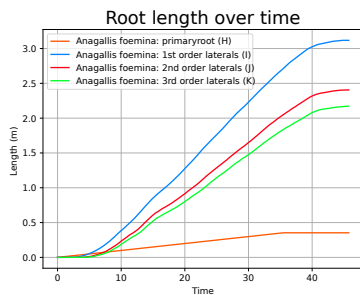


Figure 5: Accumulated length for each root type of the *Anagallis foemina* plant over time.

make up the majority of first-order laterals due to the production rule for the PR, which favors their generation with a 7 : 3 ratio over GSA-bound ones. This leads to an overall dominance of ng-LRs around the central axis laid out by the PR in the *H. annuus* model (Figure 6). Unconstrained by gravitropism, these ng-LRs respond primarily to the combined influence of other tropisms, like thigmo-, hydro- or chemotropism. Figure 2 shows an *H. annuus* root system in a complex environment, highlighting the influence of non-gravitational tropisms on the ng-LRs' growth trajectory. Additionally, Figure 4 provides a visual representation of the RSA in a homogeneous environment, allowing for comparison between the two scenarios.

***Crypsis aculeata* (recently renamed *Sporobolus aculeatus*)** Pricklegrass is a hardy wild annual grass forming dense, prostrate tufts. It thrives in less-than-ideal soils, including those that are seasonally dry and saline. Contrary to its unassuming above-ground appearance, *C. aculeata* boasts the most extensive root system among the species in our models. As a monocot (like all grasses and grass-related species), its root system features a fibrous network of thin roots rather than a dominant taproot. To model this, our configuration uses an axiom with multiple symbols, representing both basal and shoot-borne roots. Initially, these

roots exhibit a high GSA value, promoting shallow growth for accessing nutrients near the surface. However, a scaling function causes the GSA to decrease with root length, enabling the plant to curve and eventually reach deeper soil layers (up to 2 m deep). The simulated root system is visualized in Figure 7 and accompanying length plot in Figure 6.

Zea mays Maize is a globally important domesticated crop known for its extensive root system. A staple food across cultures, maize thrives in warm, well-drained soils with adequate nutrients. Our *Z. mays* model configuration seen in Figure 8 draws upon the *Zea mays* 3 parameter values published in Forschungszentrum Juelich GmbH (2024). Like *C. aculeata*, our model employs a multi-symbol axiom to represent the complexity of the monocot root system with its distinct root types. Shoot-borne roots exhibit an initial high GSA value that is gradually modulated by a scaling function, resulting in the characteristic curvature. These roots, along with others originating from the base of the plant, enable maize to reach substantial depths and explore a wide horizontal area in search of resources. Figure 6 visualizes the growth dynamics of different root types over time.

Evaluation

To validate the benefits of the interactive simulation, a botany expert supported the formative process of the model and software development as an external stakeholder. His expertise and formulation of requirements has greatly influenced this work, which is why, after concluding the evaluations, we asked him to become a co-author of this paper.

During the development we conducted three meetings with this expert. All meetings were taking place via video-conference and used screen sharing to present the progress. The first meeting was held a month prior to the start of development. One and a half months after the work started, a second meeting was held. The third and last meeting took place around three months later and one month before the model and software development was concluded.

The first meeting took about one hour. Here, the expert provided valuable insights into different aspects of root development. He also provided several introductory and advanced scientific references, which informed the model design. In particular the expert highlighted the tropisms (i.e. directed growth responses) of root systems reacting to salt and harmful substances in the soil, as well as the emergence of branches after the root tip has grown a certain length. This directly influenced our model requirements.

The second meeting included a screen-sharing demonstration of an early prototype, showcasing simulated root growth dynamics. Meeting length was around two hours and the implementation showed an early root SG implementation that incorporated basic branching, halotropism and thigmotropism. The expert gave feedback on several aspects.

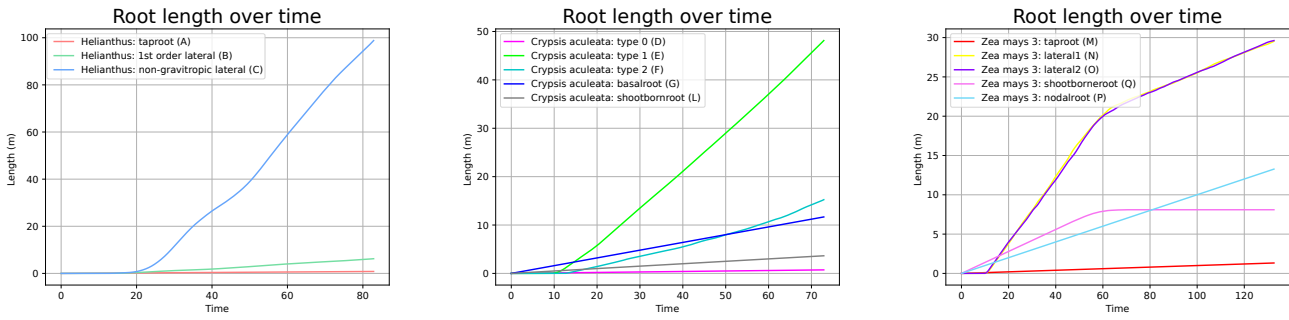


Figure 6: Accumulated length for each root type of *Helianthus annuus*, *Crypsis aculeata* and *Zea mays* over time.

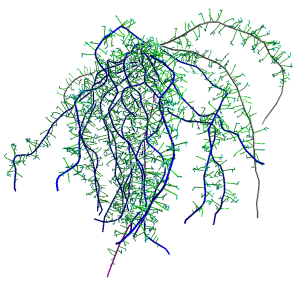


Figure 7: Rendered result of the *Crypsis aculeata* root system.



Figure 8: Rendered result of the *Zea mays* root system.

The expert observed that LRs were not responding to gravity. To address this, we introduced the gravity set-point angle (GSA) into the model. Further questions in the meeting directly informed following improvements. The seeds have initial nutrient values and plants require water for nutrient absorption. Plant's nutrient availability is abstracted to the plant as a whole. Assumptions about only one LR branching level have been corrected by the expert.

The final meeting took about two hours, with a live demo of around one and a half hours. In the demo, the expert first tested the simulation on their own PC and then we showed several simulation aspects, with the simulation running on our hardware. In both cases the screen was shared. Apart from minor improvements, the tested simulation model was in the same state as the provided source code. The expert found the interactions and the implemented menus to be easy to understand and intuitive. He stated, that it didn't take long for him to know what he could do in the simulation. The implemented interactions, inspection features, and export possibilities provided the expert with a range of experimentation options within the simulation. During the live demo, the expert tested diagram export possibilities, generating various examples. Concerning limitations, the expert asked about the possible levels of lateral roots. This is not limited in the simulation and can be defined by the production rules. One

major aspect of their feedback focused on how the plant's internal state, particularly available nutrients and potential harm from high concentrations of substances like iron, might influence growth direction. This is not the case in the current simulation model, but could be implemented with the current basis. Additionally, he inquired about the potential to simulate plants within diverse environments and investigate inter-plant competition for space and nutrients. While the simulation incorporates root separation behavior and nutrient absorption capabilities, the current configurations might not be optimized to model intense competition scenarios.

Conclusion & Future Work

This paper presents an agent-based, interactive simulation model of root growth. It combines SGs with point cloud environments to provide for a large interaction space—between the growing root systems and their heterogeneous, dynamic environments, but also between the various model components and parameters and the researchers driving the simulation and exploration of the model. To validate the model and to demonstrate its versatility, we configure and present four distinct plant root systems.

Overall, the paper encompasses the resulting model specification, its validation based on retracing the growth of well-known example specimen, the formative development process of both model and software which involved the iterative feedback from a botany expert. This process was concluded with a summative evaluation and interview which confirmed the accessible interactivity of the developed software and its benefits: It yielded numerous concrete opportunities for further empirical exploration of various model configurations, which confirms that the intended reasoning support of the interactive simulation setup works in principle.

More human-centered efforts could be pursued to increase the value for scientific exploration. Yet, botanic research can already benefit from the interactivity of the system by quick turn-around cycles of model testing and refinement and, ideally, to trigger novel questions and directions of empirical inquiry.

References

- Aibara, I. and Miwa, K. (2014). Strategies for optimization of mineral nutrient transport in plants: multilevel regulation of nutrient-dependent dynamics of root architecture and transporter activity. *Plant and Cell Physiology*, 55(12):2027–2036.
- Forschungszentrum Juelich GmbH (2024). *CRootBox*. <https://plantmodelling.shinyapps.io/shinyRootBox/> [Accessed: 2024-03-24].
- Hamon, L., Richard, E., Richard, P., Boumaza, R., and Ferrier, J.-L. (2012). Rtil-system: a real-time interactive l-system for 3d interactions with virtual plants. *Virtual Reality*, 16:151–160.
- Jacob, C. and von Mammen, S. (2007). Swarm grammars: growing dynamic structures in 3d agent spaces. *Digital Creativity*, 18(1):54–64.
- Leitner, D., Klepsch, S., Bodner, G., and Schnepf, A. (2010). A dynamic root system growth model based on l-systems: Tropisms and coupling to nutrient uptake from soil. *Plant and soil*, 332:177–192.
- Lindenmayer, A. (1968). Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of theoretical biology*, 18(3):280–299.
- Lobet, G., Pound, M. P., Diener, J., Pradal, C., Draye, X., Godin, C., Javaux, M., Leitner, D., Meunier, F., Nacry, P., et al. (2015). Root system markup language: toward a unified root architecture description language. *Plant physiology*, 167(3):617–627.
- Lynch, J. P., Nielsen, K. L., Davis, R. D., and Jabllokow, A. G. (1997). Simroot: modelling and visualization of root systems. *Plant and Soil*, 188:139–151.
- Muthert, L. W. F., Izzo, L. G., Van Zanten, M., and Aronne, G. (2020). Root tropisms: Investigations on earth and in space to unravel plant growth direction. *Frontiers in plant science*, 10:1807.
- Postma, J. A., Kuppe, C., Owen, M. R., Mellor, N., Griffiths, M., Bennett, M. J., Lynch, J. P., and Watt, M. (2017). Opensim-root: widening the scope and application of root architectural models. *New Phytologist*, 215(3):1274–1286.
- Prusinkiewicz, P. and Lindenmayer, A. (2012). *The algorithmic beauty of plants*. Springer Science & Business Media.
- Raies, Y. and von Mammen, S. (2021). A swarm grammar-based approach to virtual world generation. In *Artificial Intelligence in Music, Sound, Art and Design: 10th International Conference, EvoMUSART 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 10*, pages 459–474. Springer.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34.
- Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc.
- Schaufler, G. and Jensen, H. W. (2000). Ray tracing point sampled geometry. In *Rendering Techniques 2000: Proceedings of the Eurographics Workshop in Brno, Czech Republic, June 26–28, 2000 11*, pages 319–328. Springer.
- Schnepf, A., Leitner, D., Landl, M., Lobet, G., Mai, T. H., Morandage, S., Sheng, C., Zörner, M., Vanderborght, J., and Vereecken, H. (2018). Crootbox: a structural–functional modelling framework for root systems. *Annals of botany*, 121(5):1033–1053.
- Unity Technologies (2024). *Unity*. <https://unity.com> [Accessed: 2024-03-21].
- von Mammen, S., Wagner, D., Knote, A., and Taskin, U. (2017). Interactive simulations of biohybrid systems. *Frontiers in Robotics and AI*, 4:50.