

ROOT GROWTH SIMULATION USING L-SYSTEMS

DANIEL LEITNER * AND ANDREA SCHNEPF †

Abstract. Dynamic root growth simulation is an important tool when analysing the mechanisms within the rhizosphere. The concentration of nutrients in the soil as well as water content is strongly dependent on the root structure. On the other hand root growth is effected by the nutrient concentration and water supply as well as other soil parameters. As a result realistic root growth models are often coupled with models describing the plant-soil interactions. In this paper we present an L-System algorithm which makes it possible to easily create 3-dimensional geometries of growing plant root systems. Furthermore we discuss possibilities of coupling these root growth system models with arbitrary models describing plant-soil interaction. The model is implemented in Matlab, which makes it easy to couple it to existing Matlab or Comsol models.

Key words. Modelling and simulation, L-Systems, root growth dynamics, plant-soil interaction

1. Introduction. In modelling and simulation of plant-soil interactions accurate root growth models are of major importance. In many cases those models are coarse approximations of growing root systems. Many models directly compute root length densities without including root architecture or branching structure. These densities then provide the basis of source and sink terms to nutrient uptake or exudation models ([1], [2]).

Considering discrete 3-dimensional time dependent root growth models has certain advantages. Firstly, it makes it easier to describe the developmental process such as root branching, root diameter or root growth angles. Secondly, it enables us to link the discrete model to a model which describes plant-soil interactions. This coupling makes it possible to describe changes of plant morphology due to water content or distributions of nutrients in the soil. Finally, such models can be used to analyse the effects of root growth on plant nutrient uptake from soil. This is done by comparing the total nutrient uptake to coarser models obtained by averaging or homogenisation.

There exist some powerful discrete root growth models ([3], [4], [5], [6]). However it is hard to completely understand their behaviour and couple them to arbitrary uptake and exudation models. Our aim is therefore to develop tools that can easily set up root growth models and link them to arbitrary uptake or exudation models implemented in Matlab or Comsol. It is important that global root system parameters from literature or experiments can be included in the model. As a result the code is kept simple by using the notation of parametrized L-Systems, and therefore it can be easily adapted to several applications.

2. L-Systems. In computer graphics the complex geometry of plants is often described by L-systems. L-systems achieve this very elegantly and derive amazing pictures from simple production rules (see e.g. [7]). In this work time-dependent L-systems are used to describe the growth of root systems. In this application it is most

*Institute of Soil Science, Department of Forest and Soil Sciences, BOKU - University of Natural Resources and Applied Life Sciences, Vienna, Peter Jordan Str. 82, A-1190, Vienna, Austria (daniel.leitner@boku.ac.at).

†Institute of Soil Science, Department of Forest and Soil Sciences, BOKU - University of Natural Resources and Applied Life Sciences, Vienna, Peter Jordan Str. 82, A-1190, Vienna, Austria (andrea.schnepf@boku.ac.at)

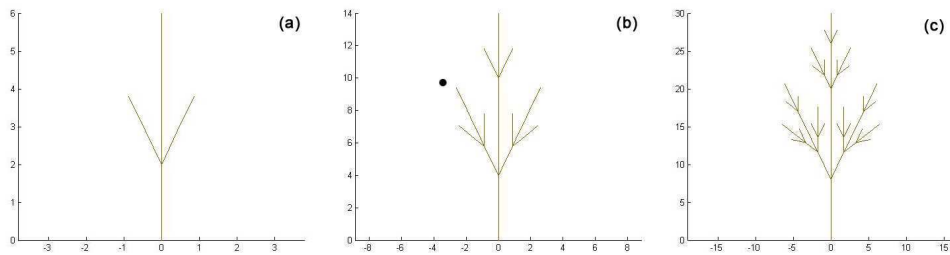


FIG. 2.1. The 2nd, 3rd and 4th generation of an L-System (reproduced from [7])

important that parameters that have been derived from literature or experiments can be used and the time dependent growth is described accurately.

L-systems are rewriting systems. All letters of an initial word usually denoted as ω are replaced according to production rules. The production rules are applied to each letter of the word, this is performed recursively n times to achieve a L-system of the n^{th} generation. The resulting string is interpreted graphically. For example the initiator ω and the production rule for X

$$\omega = X \tag{2.1}$$

$$X \rightarrow F - [[X] + X] + F [+ F X] - X \tag{2.2}$$

create figure 2.1. Note that the letters $F, +, -, [,]$ are replaced by themselves and are used for the graphical interpretation of the string (see appendix A for a list of turtle graphic commands).

This is a very elegant description for a complex branched geometry. In root growth simulations we want to describe a time dependent continuous growth, thus a time step Δt is needed. It must be ensured that the simulation result is independent of the time step Δt . Introducing time is achieved using parametrized L-Systems, where one parameter is denoted as the time.

3. Time dependent root growth simulation. Time dependent L-Systems are used to describe the growth of the root system. Note that in such an L-System most of the letters are replaced by themselves. Only the tips of the root system, the region where growth occurs, are replaced by new letters.

3.1. A single root. A single root which grows with a certain speed to a certain length according to a continuous growth function $\lambda(t)$ can be described with the following production rule:

$$G(t, l) \rightarrow \begin{cases} l + \Delta x \leq \lambda(t + \Delta t) & : R F_{\Delta x} G(t, l + \Delta x) \\ \text{otherwise} & : G(t + \Delta t, l), \end{cases} \tag{3.1}$$

where t represents the current time and Δt is the time step. The length l is the approximation of the actual root length $\lambda(t)$ with $|l - \lambda(t)| < \Delta x$. The letter R describes a rotation (i.e.: describing a small turn of the root tip). The letter $F_{\Delta x}$ produces a segment of the root with length Δx . Note that the notation is recursive, in one time step numerous segments can be produced (equation 3.1, upper expression). After that the time is increased to $t + \Delta t$ (equation 3.1, lower expression).

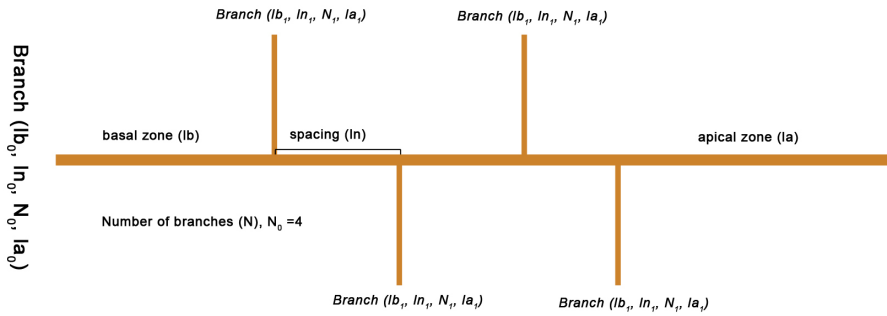


FIG. 3.1. An illustration of the self similar character of plant roots

Every root grows according to an predetermined growth function $\lambda_i(t)$. In our simulations we choose (following [8]):

$$\lambda_i(t) = k_i \left(1 - e^{-\frac{r_i}{k_i}t} \right), \tag{3.2}$$

where i denotes the order of the root, k_i is the maximal length of the root and r_i is initial growth speed. For roots of the highest order, which do not produce branches, the rule given in 3.1 is sufficient. However, in general, branching has to be considered.

3.2. The branched root system. Considering a root system, it is important to exploit its self similar structure (see figure 3.1). Every single root of a certain order produces branches of successive order and is divided into three zones: the basal and apical zones near the base and the tip of the root where no branches are produced, and the branching zone where new roots of the next order are created. The corresponding parameter values depicted in figure 3.1 determine the length of each zone predetermined for every order i .

According to this branching structure, our L-System model has a production rule for the basal zone, which is followed by the production rules for the branching zone and the apical zone. When new branches are created they begin to grow after a certain time delay in order to wait for the apical zone to develop. The production rule to create a new branch sets up the production rules describing the different root zones. In doing so more basic production rules are used. A hierarchy of production rules is given in the following table:

ruleCreateRoot	basal zone	ruleSegmentGrowth
	branching zone	ruleSegmentGrowth ruleBranching ruleDelay ruleCreateRoot ruleGrowth
	apical zone	ruleSegmentGrowth

It would go beyond the scope of this article to describe every production rule in

detail, but we emphasise that the Matlab code is freely available¹ and can be easily analysed and adapted. Every production rule has an Matlab file with corresponding name. In the following we discuss the parameters which are used within the production rules.

3.3. Parameters. For every order of the root system several parameters are needed. In our model for every parameter the mean and standard deviation must be predetermined. If the standard deviation of a value is not known it can be simple set to zero. The parameters describing the dynamic, length and interconnection of the root system are given by:

Mean	SD	Description	Dimension
r	r_s	initial growth speed	[cm/s]
l_b	l_{bs}	length of basal zone	[cm]
l_a	l_{as}	length of apical zone	[cm]
l_n	l_{ns}	spacing between branches	[cm]
n_{ob}	n_{obs}	number of branches	-

To obtain a 3-dimensional geometry the following geometrical parameters are needed:

Mean	SD	Description	Dimension
a	a_s	root radius	[cm]
θ	θ_s	branching angle from predecessor	radian
$color$	-	colour of the root	RGB

Note that the root radius can be a function of root length, e.g. getting smaller towards the root tip. However for many applications it is sufficient to assume a constant radius per root order because variation of root radius is small compared to plant scale. Moreover quantitative values are often not available.

With these basic parameters an idealized root system is described, where every root tip grows straight ahead. To include effects that arise from gravitation, soil properties or nutrient availability more parameters are included. This parameters are discussed in the following section.

4. Plant morphology. The root system has a temporal resolution (Δt) and spatial resolution along a root (Δx). The spatial resolution means that no root contains a segment which is longer than Δx . In front of every segment $F_{\Delta x}$ is rotation R_i (see for example equation 3.1, the other rules are defined in a similar way). This rotation changes the growth direction of the root tip. The subscript i indicates that the mechanism how the rotation is created can be different for each order i . In this way different mechanisms like geotropism or chemotaxis can be included into root growth model.

4.1. Labelling the root tips. If complex mechanisms like geotropism or chemotaxis are implemented, every root tip has to know its position and growth direction. Therefore additional parameters are used within the production rules of the growing tips to take note of the tips position, growth direction and radius. Thus if a tip is

¹A link can be found on www.boku.ac.at/rootmodel/further

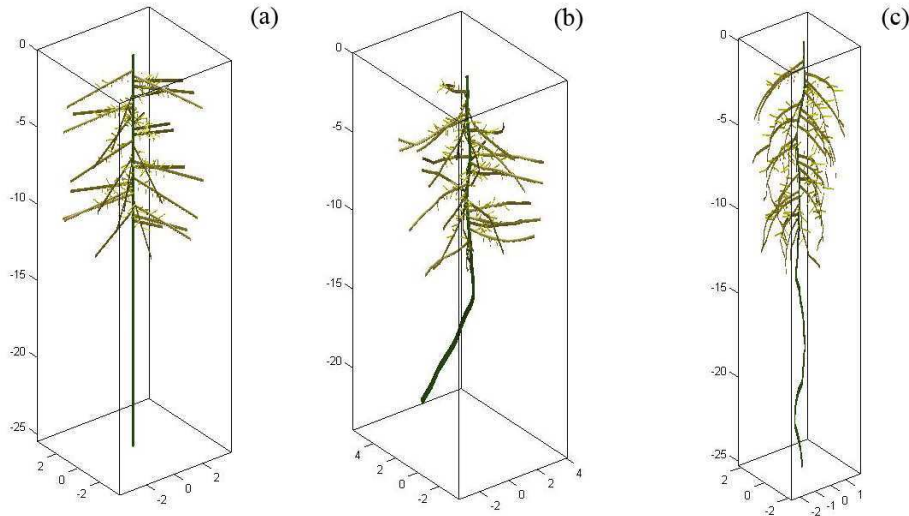


FIG. 4.1. A single 0^{th} order branch of a 25 days old maize root system: (a) without any rotation (b) random rotation ($\sigma_0 = \pi/20, \sigma_{1,2} = \pi/10$) (c) gravitropism ($N_0 = 7, N_1 = 3, N_2 = 4$)

replaced by a rule like

$$\text{tip} \rightarrow \underbrace{\dots R_i F_{\Delta x}}_{\text{transformations } T} \text{ tip}, \quad (4.1)$$

the relevant parameters have to be updated corresponding to the transformation T .

4.2. Gravitropism. If the rotations R_i do not change the heading of the root tips they just grow straight ahead, resulting in figure 4.1(a).

The simplest way to modify the direction of the growth of a root tip is to do it randomly. First the root tip rolls left or right with a uniform random angle δ_1 between 0 and π . Next the tip turns to the left with a normally distributed random number δ_2 with standard deviation σ . The parameter σ controls how fast the root can bend per Δx . This yields in a root system as represented in figure 4.1(b).

The simplest way to implement gravitropism is to favour rotations which will lead to a downward movement of the root tip. Therefore the parameter N is introduced denoting the number of trials for pairs of angles δ_1 and δ_2 . The best pair of angles is then used for the rotation. Figure 4.1(c) shows the effect of gravitropism implemented this way.

4.3. Coupling with a soil model. In many applications it is important to couple the root growth model to a soil model, because they mutually affect each other. For example root growth affects the nutrient distribution and water content of soil, while the growth itself is dependent on these parameters. The coupling of the root growth model with a model describing the mechanisms within the rhizosphere can be achieved by calculating both models in turn with an sufficiently small time step Δt .

The root growth model supplies the 3-dimensional geometry of the root system and therefore all resulting parameters (e.g. total length or root densities). The soil model uses the parameters of interest and supplies the root system model with results

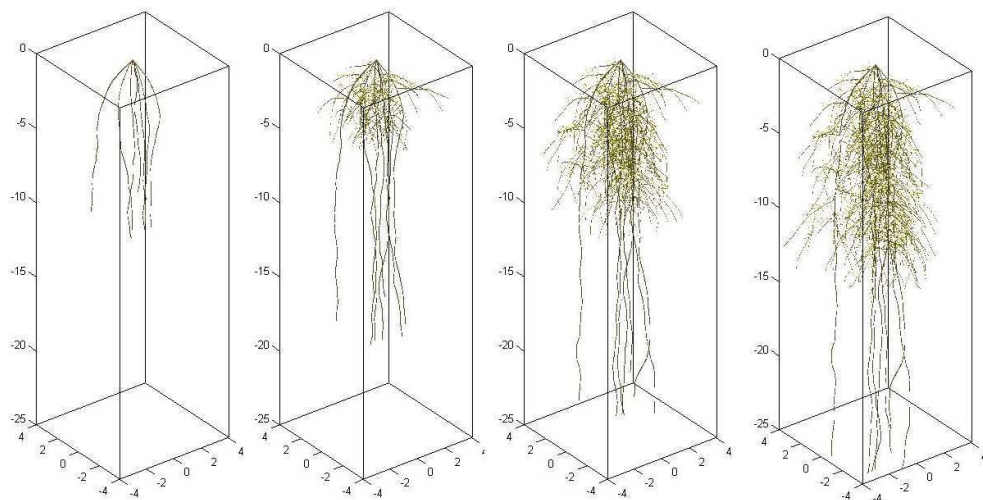


FIG. 4.2. *The time dependent growth of a 7, 14, 21 and 28 days old maize root system*

about nutrient concentration or water content. Note that using the 3-dimensional geometry in a mechanistic soil model is tricky since the structures are highly branched. An introduction and discussion how to mesh these structures along with a simulation of the effect of this root system on nutrient concentration in soil and overall plant nutrient uptake can be found in [9].

If a nutrient concentration is simulated, chemotaxis can be easily implemented in analogy to gravitropism but in contrast by favouring pairs of δ_1 and δ_2 so that the root tip heads to the highest nutrient concentration.

4.4. Example: Maize root systems. As an example how to set up a root growth model a maize root system is simulated. This has already been done by [8] and [10]. In this simulation parameters are used from these two works. Note that it is very easy to include parameters from literature or experiments in the model. Parameters that are not known have to be guessed or fitted to existing data.

The model is set up by basically writing all parameters into a Matlab file. After the choice of time step Δt and spatial resolution along the root Δx the initiator of the L-System is created:

```

p(1).r = 2; % 0st order
p(1).rs = 0;
p(1).lb = 1;
p(1).lbs = 0;
p(1).la = 15;
p(1).las = 0;
p(1).ln = 0.35; % by [10]
p(1).lns = 0;
p(1).nob = 50;
p(1).nobs = 0;
p(1).a = 0.05;

p(1).as = 0;
p(1).theta = 0;
p(1).thetas = 0;
p(1).color = [110/255,100/255,50/255];
p(1).N = 6;
p(1).sigma = pi/20;
p(1).initialNumber = 7; %# 0 roots
p(1).initialAngle = 50/360*2*pi;

p(2).r = 6.4; % 1st order
p(2).rs = 0;

```

```

p(2).lb = 0.4;          p(3).k = 0.5;
p(2).lbs = 0;          p(3).ks = 0;
p(2).la = 2;           p(3).a = 0.01;
p(2).las = 0;          p(3).as = 0;
p(2).ln = 0.24; % by [10] p(3).theta = 68/360*2*pi; % by [10]
p(2).lns = 0;          p(3).thetas = 3/360*2*pi; % by [10]
p(2).nob = 8;          p(3).color = [200/255,200/255,10/255];
p(2).nobs = 0;         p(3).N = 0;
p(2).a = 0.02;         p(3).sigma = pi/10;
p(2).as = 0;
p(2).theta = 68/360*2*pi; % by [10] numparameters.dx = 0.5; %[cm]
p(2).thetas = 0/360*2*pi; numparameters.dt = 7; %[days]
p(2).color = [140/255,125/255,40/255];
p(2).N = 3;           rules = ruleDefinition();
p(2).sigma = pi/10;
                                initiator = createRootSystem(p,...
p(3).r = 2; % 2nd order numparameters,rules);
p(3).rs = 0;

```

After the initiator is created the production rules are applied four times (the time step Δt is set to seven days). The resulting 3-dimensional maize root system is represented in figure 4.2.

5. Conclusion. In this article we have presented a dynamic root architecture model that is based on time-dependent L-Systems. It has been shown that it is easy to set up arbitrary models of a growing root systems and to derive their 3-dimensional geometries. This is done by using parameters found in literature or obtained by experiments. Further we have discussed the possibility of coupling the model to plant-soil interaction models, which are implemented in Matlab or Comsol. In this way it is possible to include effects of plant morphology which are dependent on the environment of the roots. The Matlab code is freely available¹ and can be easily adapted and extended by changing parameters or by adding or rewriting of production rules.

Acknowledgments. Acknowledgments. This work was supported by the Vienna Science and Technology Fund (WWTF, Grant No.: MA07-008) and the Austrian Science Fund (FWF, Grand No.: T341-N13). Andrea Schnepf is a Hertha-Firnberg Research Fellow.

REFERENCES

- [1] T. ROOSE and A. C. FOWLER *A mathematical model for water and nutrient uptake by plant root systems* Journal of Theoretical Biology, 2004. 228(2): p. 173-184.
- [2] T. ROOSE and A. C. FOWLER *A model for water uptake by plant roots* Journal of Theoretical Biology, 2004. 228(2): p. 155-171.
- [3] C. JOURDAN and H. REY *Modelling and simulation of the architecture and development of the oil-palm (elaeis guineensis jacq.) root system: I. The model* Plant and Soil, 1997. 190(2): p. 217-233.
- [4] L. PAGES *Root system architecture: From its representation to the study of its elaboration* Agronomie, 1999. 19(3-4): p. 295-304.
- [5] B. FORDE and H. LORENZO *The nutritional control of root development* Plant and Soil, 2001. 232(1-2): p. 51-68.

¹A link can be found on www.boku.ac.at/rootmodel/further

- [6] L. PAGES ET AL., *Root Typ: A generic model to depict and analyse the root system architecture*, Plant and Soil, 2004. 258(1-2): pp. 103-119.
- [7] P. PRUSINKIEWICZ AND A. LINDENMAYER *The algorithmic beauty of plants* Springer-Verlag, New York, 1990.
- [8] L. PAGES, M. O. JORDAN AND D. PICARD *A Simulation-Model of the 3-Dimensional Architecture of the Maize Root-System* Plant and Soil, 1989. 119(1): p. 147-154.
- [9] A. SCHNEPF AND D. LEITNER *FEM simulation of nutrient uptake from soil by a 3-dimensional root system using DistMesh and Comsol Multiphysics* Proceedings of ALGORITHMY (2009).
- [10] SHIBUSAWA, S. *Modelling the branching growth fractal pattern of the maize root system* Plant and Soil, 1994. 165(2): p. 339-347.

Appendix A. The resulting string of a parametric L-System can be graphically represented using a Turtle graphics. The letters of the string represent the following commands (after [7]):

<i>Letter</i>	<i>Description</i>	<i>Parameter names</i>
F	Draws a segment and moves to new position	length
#	Sets the width of the following segments	width
C	Sets the colour of the following segments	color
+, -	Turns left or right	delta
&, ^	Pitch down or up	delta
\, /	Roll left or right	delta
	Turn around	-
[,]	Pushes and pops the turtle state	-

The segments are normally straight cylindrical tubes or lines. The results can be either plotted to a Matlab figure (`drawString3D.m` or `drawString3Dtube.m`), or to DXF or STL files (`drawString3DDXF.m`, `drawString3DSTL.m`). Further COMSOL Multiphysics geometry can be produced directly (`drawString3DComol.m`) or by first meshing the geometry externally (`ImportMeshComsol.m`) and then importing resulting the mesh.