Proceedings of ALGORITMY 2009
pp. 321–330

FEM SIMULATION OF BELOW GROUND PROCESSES ON A 3-DIMENSIONAL ROOT SYSTEM GEOMETRY USING DISTMESH AND COMSOL MULTIPHYSICS

ANDREA SCHNEPF* AND DANIEL LEITNER†

Abstract. Understanding the underlying below-ground processes and the dynamics of plant-soil interactions is of major importance in areas such as agriculture or phytoremediation. The contribution of individual plant roots to the overall effect of a whole plant root system on the soil and vice-versa is however difficult to assess and mathematical models using different upscaling methods are currently developed. This paper presents a numerical approach for plant and soil interaction models on a 3-dimensional root system geometry. It is based on L-system representations of root systems, the Matlab algorithm `DistMesh` for meshing the geometry and the finite element solver COMSOL Multiphysics for solving the diffusion equation with both linear and nonlinear flux boundary conditions at the root surfaces. Although computational limits are to be expected with regard to the complexity of the root branching structure, we propose that this approach is useful for evaluating different homogenisation and averaging methods by comparing effective solutions to the results of the explicit 3-dimensional solution.

Key words. 3-dimensional model, COMSOL Multiphysics, DistMesh, finite element method, nutrient uptake, root system

1. Introduction. Root architecture and root branching structure have a major impact on the functioning of root systems, for example on the plant water and nutrient uptake from soil. In particular in low-input agricultural systems, where nutrients are not supplied as mineral fertilizers, plant adaptations to their soil environment is crucial for successful plant nutrition. Models of water and nutrient uptake from soil by root systems have been based on simple empirical equations of root length densities in soil and have recently included more sophisticated models of root architecture [2, 5, 6]. However, these models also create volumetric sink terms for nutrient uptake from soil that are based on root length densities. To our knowledge, these models have never been tested against truly 3-dimensional simulations. One of the reasons for this is the complicated fibrous geometry of root systems. At increasing complexity, 3-dimensional representations of root systems become challenging and suitable upscaling methods are necessary. Currently, more rigorous upscaling methods such as averaging and homogenisation are developed for root systems; they need to be assessed by comparing against experimental data or 3-dimensional simulations, or both. In this paper, we use a 3-dimensional L-system representation of a simple root system [3], and model both nutrient uptake from soil and release of organic compounds into the soil by this root system. The numerical solution of this model is based on the finite element method (FEM) [7] and computed using COMSOL Multiphysics.

2. Methods. Starting point is the output of an L-sytem model of a root system [3], a string which is interpreted by a turtle graphic. The result is a polyline

*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).

†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).

along the axis of all branches in the root system with additional information for each segment such as corresponding root radius. Importing this geometry directly into COMSOL and meshing this geometry has proven to be difficult due to the complexity of the branched structure. This was particularly true when the axes of the individual branches were not straight lines but piecewise linear approximations of bent branches that were allowed to change direction while growing. Therefore, a finite element mesh was first created in MATLAB and then imported into COMSOL where the finite element model was solved.

2.1. Finite element mesh generation. The MATLAB algorithm of the mesh generator `DistMesh` [4] was used for meshing the region of soil around a root system. Signed distance and step size functions are required inputs of `DistMesh`.Outputs are two matrices, one containing the coordinates of the mesh nodes and one containing the indices of the nodes which form the tetrahedrons. The geometry is represented implicitly using signed distance functions which are negative inside the domain that needs to be meshed. The position of the mesh nodes is determined based on a force equilibrium in the element edges. For triangulation, the Delaunay algorithm is used. The MATLAB m-files of the signed distance function representing the geometry of a soil around a root system and the corresponding step size function are shown in appendix A and are described in the following paragraphs. An example of a resulting mesh is shown in figure 3.1.

2.1.1. Signed distance function for root systems. The signed distance function representing the geometry of a soil around a root system was created in the following way: The root system created by the L-system algorithm was represented by a polyline along the axes of the roots. All segments of the polyline were stored in a list, where each item had the following format:

Notation	Sample value	Meaning
r	0.1	root radius (cm)
p	[0 0 0]	starting point of line segment
u	[0.015836 -0.076214 -0.99697]	vector between start and end point of line segment

The main idea was to calculate the distance of a given point from the closest polyline segment and subtract the corresponding root radius. If the result is negative, the point lies inside the root and otherwise it lies outside the root. For each arbitrary point q, we check whether the base of the perpendicular from q to the extended line is inside the range of the segment (Fig. 2.1(a)), in which case the distance to the line segment is given by $d = \frac{|(q-p)\times u|}{|u|}$. Otherwise, it is given by the direct distance between the two points p and q or $p + u$ and q (Fig. 2.1(b)). The corresponding MATLAB function `distLinePoint` is given in appendix A. The points of the resulting discrete description of the domain lie inside the roots. However, we would like to describe the influence of nutrient fluxes across the root surfaces on the surrounding soil. Hence, we need to build a signed distance function where the points of the domain lie outside of the roots. This geometry was created by using another signed distance function which describes a box. Then, the difference between the two sets of points (`DistMesh` function `ddiff`, [4]) represents those points which lie outside the root system but within a given bounding box. Our MATLAB function `distTree` creates this domain and is given in appendix A.

(a) Base of the perpendicular from q lies on the line segment

(b) Base of the perpendicular from q lies outside the line segment

FIG. 2.1. *Distance between a point of the domain and a line segment of the root system*

2.1.2. Varying step size function. By default, a uniform step size is used by `DistMesh`. In our problem, we expect most changes to happen close to the surface of the roots where nutrient uptake occurs. Therefore, a nonuniform stepsize function `myfh` (appendix A) was created where the step size near the root surface was set to a smaller value than in the rest of the domain. Since the initial mesh is iteratively improved by solving for a force equilibrium in the element edges, the relative step size at the bounding box boundaries was set to 0.7. This acts as a counterforce to the small element sizes near the root surface boundaries.

2.1.3. Mesh quality. The quality of each element in the mesh is determined by the ratio between the largest inscribed sphere and the smallest circumscribed sphere, i.e.,

$$q = 3\frac{r_{in}}{r_{out}}. \tag{2.1}$$

The factor 3 ensures that the maximum quality is equal to 1 (equilateral tetrahedron); the lowest quality is equal to zero.

2.2. Solution of the finite element model in COMSOL Multiphysics. We consider two scenarios. In the first example, we study the effect of a branched root system which takes up nutrients everywhere on the root surfaces. The initial nutrient concentration in soil is assumed to be homogeneous. Nutrients are taken up at the root surface according to Michaelis-Menten kinetics. Furthermore, nutrients in soil solution are assumed to be in equilibrium with the nutrients adsorbed to the soil solid phase and to move within the soil liquid phase due to diffusion only. The model thus is based on the diffusion equation with non-linear flux boundary conditions at the root surfaces and zero Neumann boundary conditions at the bounding box as well

as homogeneous initial conditions (equations (2.2)-(2.5)).

$$(b+\theta)\frac{\partial c}{\partial t} = \nabla \cdot (D\theta f \nabla c)\,, \tag{2.2}$$

$$D\theta f \nabla c = \frac{F_{max} c}{K_m + c} \text{ when } \boldsymbol{x} \text{ is on a root surface,} \tag{2.3}$$

$$D\theta f \nabla c = 0 \text{ when } \boldsymbol{x} \text{ is on the bounding box,} \tag{2.4}$$

$$c = c_0 \text{ at } t = 0, \tag{2.5}$$

where c is the nutrient concentration in soil solution, b is the buffer power which describes the partition of nutrient between solution and solid phases of the soil, D is the nutrient diffusion coefficient in free water, θ is the volumetric water content, f is the impedance factor, F_{max} is the maximal nutrient influx into the root, K_m is the Michaelis-Menten constant, the concentration at which half the maximum influx is reached, c_0 is the initial nutrient concentration in soil solution, $\boldsymbol{x}$ is the space variable and t is the time.

In the second example, we study the effect of a branched root system which releases photosynthate carbon as root exudates into the soil everywhere on the root surfaces. As in the previous example, exudates in soil solution is assumed to be in equilibrium with the exudates adsorbed to the soil solid phase and to move within the soil liquid phase due to diffusion only. The difference to the previous example is a different boundary condition for the root surfaces where we assume a constant exudation flux. Furthermore, root exudates are decomposed by micoorganisms, thus there is an additional first order reaction term in the equation. The model is based on a diffusion-reaction equation with linear flux boundary conditions at the root surfaces and zero Neumann boundary conditions at the bounding box. Initially, we assume that there are no exudates present in the soil. The model is given by equations (2.6)-(2.9).

$$(b_{ex}+\theta)\frac{\partial c_{ex}}{\partial t} = \nabla \cdot (D_{ex}\theta f \nabla c_{ex}) - k c_{ex}, \tag{2.6}$$

$$D_{ex}\theta f \nabla c_{ex} = F_{ex} \text{ when } \boldsymbol{x} \text{ is on a root surface,} \tag{2.7}$$

$$D_{ex}\theta f \nabla c_{ex} = 0 \text{ when } \boldsymbol{x} \text{ is on the bounding box,} \tag{2.8}$$

$$c_{ex} = 0 \text{ at } t = 0, \tag{2.9}$$

where c_{ex} is the exudate concentration, b_{ex} is the buffer power of the exudate, D_{ex} is the exudate diffusion coefficient in free water and F_{ex} is the constant flux of exudates per unit surface area of root surface.

Equations (2.6)-(2.9) were solved using the finite element solver and analysis package COMSOL Multiphysics and using the same mesh as in the first example. Simulation time was 1 day.

3. Results.

3.1. Finite element mesh. Finite element meshes of a soil boxes surrounding a plant root systems were created with `DistMesh` (e.g. see figure 3.1). The algorithm generally produced meshes of high quality. The quality of each element was calculated according to equation (2.1). In table 3.1, the arithmetic mean and variance of all element qualities is shown for different initial step sizes and for different relative sizes near the root boundaries. The results also show that the difference between the relative step size at the outer boundaries, the faces of the box, and the root system

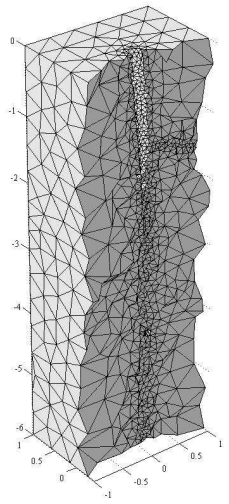


FIG. 3.1. *Finite element mesh of a soil surrounding a root system (split view)*

	h_0=0.05 h_{root}=0.1	h_0=0.03 h_{root}=0.1	h_0=0.03 h_{root}=0.05	h_0=0.021 h_{root}=0.1
# nodes	5,845	25,731	11,177	39,784
# tetrahedrons	27,377	132,263	50,162	213,957
mean quality	0.817	0.860	0.777	0.877
variance	0.018	0.015	0.021	0.014

TABLE 3.1

Meshing the region of soil surrounding a root system - Quality for different initial step sizes h_0 and relative sizes near the root boundaries h_{root}

boundaries should not be too big. Otherwise, both the mesh quality and number of nodes are decreased. This appears to be due to the force equilibrium method used by `DistMesh` to calculate the positions of the nodes such that many of the initial nodes of the equilateral mesh of the bounding box will be drawn outside the domain and removed. Further improvement of the step size function is required. In this work, the step size function was varied by specifying a smaller relative step size near the root surface boundaries. However, the complicated structure of the geometry requires more sophisticated distance functions which are based on curvature or feature size adaptations and/or gradient limiting methods [4]. However, the meshes were of sufficient quality in order to solve the diffusion-reaction equation with nonlinear boundary conditions in COMSOL . This mesh is suitable to solve arbitrary parabolic equations in any finite element solver. As a result, it is possible to solve state of the art models for plant and soil interactions on realistic root geometries.

3.2. Example 1: Nutrient uptake. The classical model of nutrient uptake by a single cylindrical root is applied to this 3-dimensional geometry of a root system with

bent roots. Nutrient uptake from soil is not accounted for by an averaged sink term but by considering nutrient flux into each individual root. Figure 3.2 shows the nutrient concentration in a soil around a root system according to equations (2.2)-(2.5). The parameter values where chosen such that they represent uptake of phosphate from soil by maize. The initially homogeneous concentration is decreased near the roots due to nutrient uptake which leads to the formation of, partly overlapping, depletion zones. Figure 3.3(a) shows the nutrient depletion zone around an individual root which is not influenced by a neighbouring root. The result is the same as usually predicted by 1-D axial symmetric calculations [5]. Figure 3.3(b) shows the effect when depletion zones overlap. The nutrient concentration between two roots is plotted along the line shown in the upper right corner. It can be seen that the concentration remains at its initial value far away from the root when there is no competition. Otherwise, when roots are competing for nutrients, the concentration between two roots is much lower than the initial value. This situation can only be investigated in detail using full 3-dimensional simulation of the root system. Depending on the shape of the geometry and the extent of the depletion zones, this may have a major influence on simulated overall nutrient uptake by the root system.

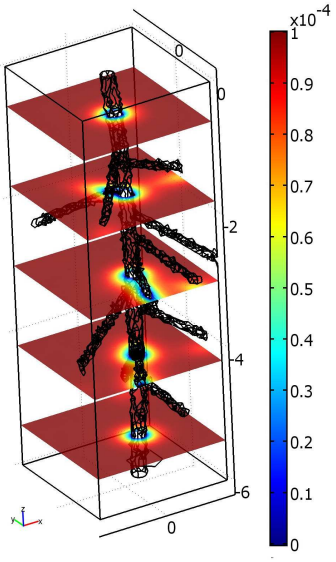
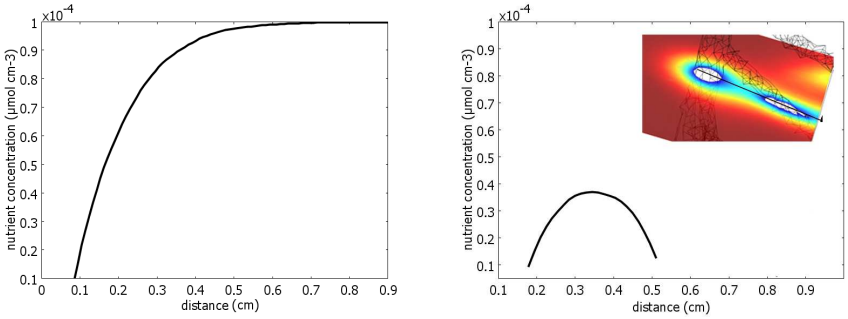


FIG. 3.2. *Nutrient depletion near a root system*

3.3. Example 2: Release of organic compounds from root into the soil. Roots can release a significant part of their photosynthate carbon into the soil (exudation). This mechanism supports a higher microbial community near the root surface and effects the availability of poorly mobile nutrients to the root. In this example, we assume a constant flux of exudates across the surfaces of all roots within the root system into the soil. Figure 3.4 shows the exudate concentration in the soil around the root system according to equations (2.6)-(2.9). The exudates accumulate near the root surface. Figure 3.5(a) shows the exudate concentration around an individual root which is not influenced by a neighbouring root. The result is again the same as usually predicted by 1-D axial symmetric calculations. Figure 3.5(b) shows the effect



(a) no overlapping depletion zones (b) overlaping depletion

FIG. 3.3. *Exudation concentration around a single root*

when accumulation zones overlap. The exudate concentration between two roots is plotted along the line shown in the bottom right corner. Again, this situation can only be investigated in detail using full 3-dimensional simulation of the root system. Depending on the shape of the geometry and the extent of the depletion zones, this may have a major influence on simulated overall effect on nutrient availability or microbial populations.

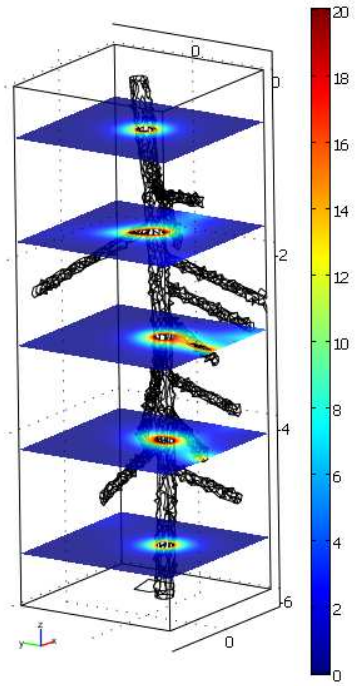
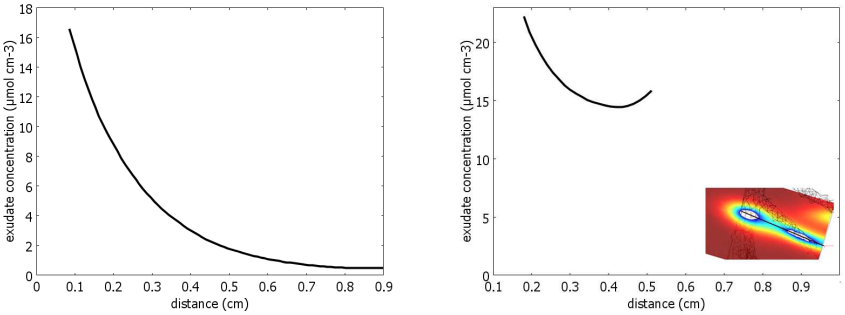


FIG. 3.4. *Release of organic compounds by a root system*

4. Summary. In this work we presented a method to generate finite element meshes of soil around plant roots in three dimensions. The root system geometry was



(a) No overlapping accumulation zones (b) Overlapping accumulation of exudates

FIG. 3.5. *Exudate concentration around a single root*

created with a root growth model using an L-system algorithm. Meshing of highly branched structures is not trivial, e.g. it was not possible to perform meshing with the COMSOL Multiphysics meshing tool. Our approach is based on the `DistMesh` algorithm. The geometry is described implicitly by creating a signed distance function. The resulting mesh is suitable for arbitrary parabolic equations and therefore it is possible to solve state of the art models for plant and soil interactions on realistic root geometries. As examples we presented both nutrient uptake and root exudation by a three-dimensional root system. Both cases highlight the possibly significant competition between the individual roots within a root system. This approach provides a tool to compare and assess different averaging methods.

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] *COMSOL Multiphysics Scripting Guide*, COMSOL Multiphysics Scripting Guide, 2007. Version 3.4. COMSOL AB, Stockholm, Sweden, 190 pp.

[2] V. DUNBABIN, *Simulating the role of rooting traits in crop-weed competition*, Field Crop Res., 104 (2007), pp. 44-51.

[3] D. LEITNER AND A. SCHNEPF, *Root growth simulation using L-systems*, Proceedings of ALGORITMY (2009).

[4] P.-M. PERSSON, *Mesh generation for implicit geometries*, SIAM Review, 46 (2004), pp. 329–345.

[5] T. ROOSE, A. C. FOWLER AND P. R. DARRAH, *A mathematical model of plant nutrient uptake*, J. Math. Biol., 42 (2001), pp. 347-360.

[6] F. SOMMA, V. CLAUSNITZER AND J. W. HOPMANS, *Modeling of transient three-dimensional soil water and solute transport with root growth and water and nutrient uptake*, Plant Soil, 202 (1998), pp. 281–293.

[7] O. C. ZIENKIEWICZ, R. L. TAYLOR AND J. Z. ZHU, *The finite element method: Its basis and fundamentals, 6^{th} edition*, Butterworth-Heinemann, 2005.

Appendix A. Code of MATLAB functions.

```
%
% Distance between line and point
%
% line is given by [p,p+u], point q;
%
function d = distLinePoint(p,u,q)

p0 = ones(size(q,1),1)*p;
t = ((q-p0)*u')/(norm(u)^2);

i0 = t<0;
i1 = t>1;
ie = t>=0 & t<=1;

u0 = ones(size(q,1),1)*u;
d(i0) = sqrt(sum( (p0(i0,:)-q(i0,:)).^2, 2));
d(i1) = sqrt(sum( (p0(i1,:)+u0(i1,:)-q(i1,:)).^2, 2));
d(ie) = sqrt(sum( cross(q(ie,:)-p0(ie,:),u0(ie,:)).^2, 2))/norm(u);
```

```
function D = distTree(p)
%
global tree;        %list of segments making up the root system

D = distBox(p); % p ... vector of mesh nodes

for i = 1 : length(tree)

    d = distLinePoint(tree{i}.p, tree{i}.u,p) - tree{i}.r;
    D = ddiff(D,d);

end
```

```
function h = myfh ( p, varargin )

global tree;      %list of segments making up the root system

h = min(abs(distBox(p)) + ones(size(p,1),1)*0.7,1);

for i = 1: length(tree)

    h = min(abs(distLinePoint(tree{i}.p, tree{i}.u, p) - tree{i}.r) ...
                                        ... + ones(size(p,1),1)*0.1, h);

end
```

Commands are described in [1].

```
% m-file Mesh2Comsol
% Taken from COMSOL Scripting Guide, p. 129
%1  Load the external mesh data to the command line:

load nodes.txt
load tetra.txt

%2  Create an initial mesh object from the external mesh data:

el = cell(1,0);
%tet = treeMesh_t+1; % only needed if lowest mesh point index is zero
el{1} = struct('type','tet','elem',tetra');
m = femmesh(nodes',el);
%(Note the transpositions of the matrices tet and coord.)

%3  Enrich the initialized mesh object with information required for
%   forming a valid mesh, and finally visualize the mesh:

m = meshenrich(m);
meshplot(m);
% Taken from COMSOL Scripting guide, p. 128
%4 Import mesh into COMSOL GUI
clear fem
fem.mesh = m;
```