Plant and Soil **188:** 139–151, 1997.

SimRoot: Modelling and visualization of root systems

Jonathan P. Lynch[1], Kai L. Nielsen[1], Robert D. Davis[2] and Andrei G. Jablokow[2]
[1] *Department of Horticulture, The Pennsylvania State University, 221 Tyson Building, University Park, PA16802, USA** *and* [2] *Department of Mechanical Engineering, The Pennsylvania State University, University Park, PA16802, USA*

Received 19 February 1996. Accepted in revised form 22 June 1996

Key words: computer modelling, C programming language, data structure, fractal geometry, kinematics, root system, visualization

Abstract

SimRoot, a geometric simulation model of plant root systems, is described. This model employs a data structure titled the Extensible Tree, which is well suited to the type of data required to model root systems. As implemented on Silicon Graphics workstations, the data structure and visualization code provides for continuous viewing of the simulated root system during growth. *SimRoot* differs from existing models in the explicit treatment of spatial heterogeneity of physiological processes in the root system, and by inclusion of a kinematic treatment of root axes. Examples are provided of the utility of the model in estimating the fractal geometry of simulated root systems in 1, 2, and 3 dimensional space. We envision continued development of the model to incorporate competition from neighboring root systems, linkage with crop simulation models to simulate root-shoot interactions, explicit treatment of soil heterogeneity, and plasticity of root responses to soil factors such as presence of mycorrhizal associations.

Introduction

Simulation models are useful tools in studies of root architecture, because of the difficulty of observing and quantifying the architecture of actual roots, and the complexity and plasticity of roots as geometric objects (for a recent review see Lynch and Nielsen, 1996). Simulation modelling has heuristic value in helping the modeler hypothesize relevant processes and interactions, in assessing the impact of single variables on system performance through sensitivity analysis, and in suggesting issues and hypotheses for experimentation (Wullschleger et al., 1994). Experimentation can help refine a model, which then can suggest further experimentation, in an iterative process that may eventually lead to a model of sufficient validity for use in predictive scenarios or as a module of more integrative models.

The earliest computer simulations of root architecture considered the branching patterns of root axes in 2 dimensions, according to simple growth rules (Hackett and Rose, 1972a,b; Lungley, 1973; for a more recent extension of this approach see Pachepsky et al., 1993). The first model to explicitly consider root architecture in 3 dimensions was ROOTMAP, which generated information about the age, position, and orientation of root segments over time as a function of root elongation rate and branching intensity, which were treated as temperature-dependent variables (Diggle, 1988). This model did not consider morphological features such as radial growth of root segments. Pages et al. (1989) developed an architectural model of maize roots that had both geometric and kinetic rules for root branching, based on empirical observations. This model permitted the visualization of the entire root system in a 3 dimensional 'wire' view, or cross sections of the root systems as would be seen by trench excavation. Rather than simulate the growth of a root system that approximates natural root systems,

* FAX No: +1814 863-6139.
E-mail: Jonathan_Lynch@agcs.cas.psu.edu

Berntson (1994) digitized actual root systems excavated from the soil and used growth rules to simulate the native architecture prior to excavation. This approach was particularly useful in evaluating root system topology, which would not be changed by excavation (for a glossary distinguishing such terms as 'morphology' and 'topology' from 'architecture', see Lynch, 1995). Fitter et al. (1991) described a simulation model of root architecture in 3 dimensions that considered topology, link length and radii, and branching angles. This model estimated zones of P depletion from surrounding soil and the construction 'cost' of tissue volume to estimate the exploitation efficiency of contrasting root architectures. Clausnitzer and Hopmans (1995) developed a spatially explicit model that is capable of simulating dynamic interactions between root growth and soil water status. It combines a three-dimensional root growth model with a transient soil water flow model and a shoot growth and dynamic assimilate allocation model.

The models developed by Fitter et al. (1991) and Clausnitzer and Hopmans (1995) demonstrate the potential utility of geometric models in analyzing the functional implications of root architecture. Both models permit the simulation and visualization of the shape of root systems with considerable precision, and they also link the spatial deployment of the root system with the acquisition of soil resources distributed in space and subject to local depletion. The model by Fitter et al. (1991) considers a 'cost' function of root deployment through estimation of tissue volume, while the model of Clausnitzer and Hopmans (1995) considers the 'gain' and feedback functions of water uptake on shoot production of assimilate for further root growth. The resulting quantitative estimation of the efficiency of resource acquisition with varying root architectures has implications for plant growth and survival in stressful environments that could be empirically validated.

Existing models of root architecture do not explicitly consider the spatial heterogeneity of root processes. Processes such as water uptake (Brouwer, 1954; Hayward et al., 1942), nutrient uptake (Clarkson, 1996; Huang et al., 1992; Waisel and Eshel, 1992), exudation (McDougall and Rovira, 1970), and respiration (Nielsen et al., 1994; Wanner, 1944) vary greatly along a root axis and among differing roots within a root system (for a recent review see Eshel and Waisel, 1996). Such variation must be incorporated into geometric models of root architecture in order to treat more accurately the relationship of spatial deployment of the root system with the spatial deployment of root functions. In this paper we describe a new geometric model of root growth that is designed to model realistically the variation of root functions in the root system, by incorporiting a kinematic context for spatial variation of physiological characteristics along root axes.

Model description

Root modelling and data structures

The basic information needed to model geometrically a single root axis is the direction of growth and the length of the axis. Each root can be thought of as a collection of axial segments which change direction according to the growth model. It is logical to have each segment consist of the incremental growth of the root during a single time step of the simulation (Pages et al., 1989). For solid modelling of a single root the only additional information needed is radius along the root axis. This radius will also be determined by the growth model and may be based on position along the root axis, order of the root, nutrient concentrations, and other empirical data.

Root branching geometry

Root growth will eventually lead to branching. In terms of geometry, two parameters are sufficient to determine the new direction of the branching root; the axial and radial branching angle, β and α respectively (Pages et al., 1989), as shown in Figure 1a. The directional vector $\vec{u}$ of the parent root is rotated through the vertical plane by the axial branching angle β to form $\vec{u}'$. This vector is then rotated about directional vector $\vec{u}$ by the radial branching angle α resulting in vector $\vec{u}''$; the direction of the newly branching root. Calculation of the vector $\vec{u}'$ consists of finding the perpendicular normal to $\vec{u}$ in the horizontal plane and rotating $\vec{u}$ about this perpendicular normal through an angle β as shown in Figure 1b. The perpendicular normal $\vec{n}$ can be found by taking the cross product with the unit direction vector $\hat{j}$.

$$\vec{n} = \vec{u} \times \hat{j} \qquad (1)$$

This calculation yields a null result if the vectors $\vec{u}$ and $\hat{j}$ are colinear. When this is the case; the perpendicular normal $\vec{n}$ may be taken as $-\hat{i}$. The final calculations necessary to find $\vec{u}''$; rotating $\vec{u}$ about $\vec{n}$ by β and $\vec{u}'$ about $\vec{u}$ by α can be accomplished through the use of a transformation matrix.

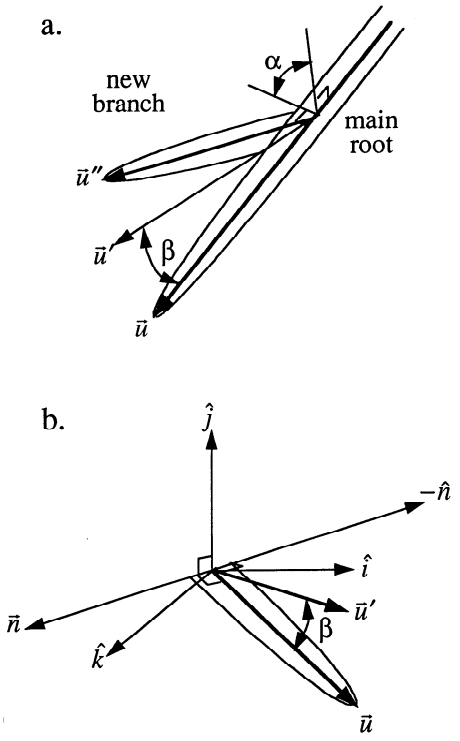

Figure 1. Parameters used to describe root branching geometry. **a**) In order to determine the new direction of a branching root two branching parameters are necessary, the axial (β) and radial (α) branching angle. **b**) Axial branching vector diagram, illustrating the rotation of the directional vector $\vec{u}$ through the vertical plane by the axial branching angle β to form $\vec{u}'$.

In general form, the transformation matrix rotates a vector $\vec{r}$ about a vector $\vec{v}$ by an angle ϕ; provided $\vec{v}$ passes through the origin of the local coordinate system. The resultant rotated vector $\vec{r}'$ is computed consistent with the right hand rule for vector manipulations, as:

$$\vec{r}' = [R(\vec{v}, \phi)]\vec{r} \qquad (2)$$

where $[R\vec{v}, \phi)$ is a finite rotation tensor which is a function of $\vec{v}$ and ϕ and is given by:

$$[R(\vec{v}, \phi] = \begin{bmatrix} v_x^2(1-\cos\phi)+\cos\phi & v_x v_y(1-\cos\phi)+v_z\sin\phi & v_x v_z(1-\cos\phi)+v_y\sin\phi \\ v_x v_y(1-\cos\phi)+v_z\sin\phi & v_y^2(1-\cos\phi)+\cos\phi & v_y v_z(1-\cos\phi)+v_x\sin\phi \\ v_x v_z(1-\cos\phi)+v_y\sin\phi & v_y v_z(1-\cos\phi)+v_x\sin\phi & v_z^2(1-\cos\phi)+\cos\phi \end{bmatrix} \qquad (3)$$

Hence, the direction of the newly branching root axis is computed as:

$$\vec{u}'' = [R(\vec{u}, \alpha)]\vec{u}' = [R(\vec{u}, \alpha)][R(\vec{n}, \beta)]\vec{u} \qquad (4)$$

The mathematics for computing the direction of a newly branching root are thus established in terms of the lateral and radial branching angles. The angles to be used, and where and when branching will occur, are determined by the specified growth model.

Data structure - description and analysis

The data structure is the conceptua1 object in which data is stored and also refers to the organization of such objects. The data structure must also preserve the various relationships inherent in the data. The Extensible Tree data structure presented herein provides the required capabilies. In order to avoid confusion, a few definitions are presented before the discussion of the data structure. A *tree* is a collection of vertices and edges that follows a predefined set of rules. A *vertex* is the object of the tree which stores information. Vertices may a1so be identified as *nodes*. An *edge* is a connection between two vertices. One node of the tree is designated as the *root* of the tree (the start of the root system). Except for the root, each node has one node directly above it. This is its *parent*. The nodes directly below a given node are it's *children*. A *leaf* is a node of the tree with no children (a root tip). The "C" programming language declaration of the data structure has, for each "root" entity, pointers to its parent, the first in a linked list of children, and both its left and right *siblings*. A roots siblings are those roots which share the same parent. A schematic diagram of the Extensible Tree is presented in Figure 2. Further, there is exactly one *path* between each node of the tree and the root; where a path is a list of vertices where the vertices are connected by edges of the tree. The data structure developed to store the root information has been dubbed the Extensible Tree, i.e. a tree wherein each node has an arbitrary number of children. The structure was developed out of the necessity to handle widely varying root architectures. In practice, the data structure mimics the topological structure of the botanical root system being modeled, as will be shown later. Despite the dynamic nature of this structure it has been designed in such a way as to be readily traversed using the same recursive algorithm regardless of the perturbation introduced by mirroring the architecture

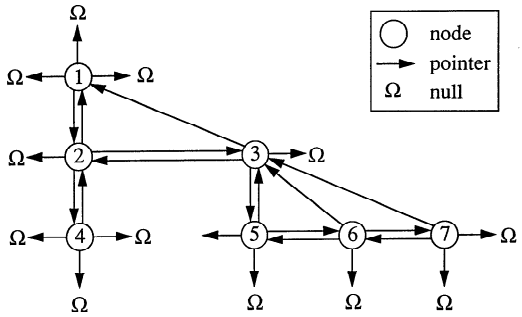

Figure 2. Typical Extensible Tree. This data structure was developed to store the root information from widely varying root architectures. Each root entity has pointers to its parent, the first in a linked list of children, and to its right and left sibling. This allows much faster and simpler implementation of operations, such as root length calculations.

of the root system being modeled. The terms *node* and *vertex* are used interchangeably here and refer to the information stored in the data structure and associated with its respective root entity.The pointers to the parent and the left sibling serve to allow for much faster and simpler implementation of certain operations. For example, finding the path length from a root tip (a leaf) to the start of the system (the root) becomes a linear time algorithm (in the height of the tree) with the addition of the pointer to the parent node. Each node has a pointer to the first in a linked list of segments. Each segment in this list corresponds to the section of root grown during consecutive time steps of the simulation. A direction of growth and length are stored for each segment along the root axis. The spatial origin of the segment is also computed and stored. The direction and length of each segment are generated by the growth model. This axial data is all that is needed to construct a wireframe depiction of the root system architecture. In order to construct a so1id model, radial information must be made available. Radius data will depend on the root model implemented and has the potential to change with each time step. For these reasons radial data are not stored. They are computed as needed using a function that returns the radius for the appropriate position along the current root. This routine for computing the radii is a component of the growth model. It may call on the properties of the root genotype or other data as defined by the growth model. The Extensible Tree, as described above, provides a versatile data structure for the storage of root model data. The algorithms for its manipulation are straightforward, and its demands on computing power, while not optimum, are reasonable.

Growth data needed for modelling botanical roots

Once the data structure is in place it becomes necessary to determine what information will be stored for each root axis, that is, what will be the fields in each node of the data structure. It is also necessary to store the variable parameters of the growth model. Each root entity has a number associating it with a root type. This number corresponds to an array index. The array, in turn, contains a pointer to a structure (of type *rttype*) containing growth data for a specific root type. This is the function parameter data that is consulted by the root growth model, for each root entity, at each time increment. A specific growth model may have multiple root types with different growth data. This allows for differing growth rates in the same root system for differing root types; primary, secondary, etc. Note that the *rttype* structure is not part of the Extensible Tree. The *rttype* structure is a stand alone record which stores growth function parameters. The appropriate *rttype* structure is accessed by the growth function based on the type number associated with a given root axis. The additional fields stored for each node of the Extensible Tree depend entirely on the functional form and necessary data of the growth model being employed to drive the simulation. As an example, time data such as the creation time, age, and time since last branching may be stored for each root and recalled by the growth model when needed.

The same structure is used for all nodes regardless of root type and storage is always allocated for it. This is done in the interests of simplicity; the routines which allocate memory for a newly created branch are simplified by using a universal data structure for each root axis. The functional form of the growth model is currently supplied as one or more C functions. The root growth routines perform computations such as the branching control, length of growth increments, application of tropisms, changes in radius, and nutrient depletion zones based on the parameters set in the *rttype* structure. It is the enhancement of these routines that will enable root biologists to study and explore different growth models representing various phenotypes within the developed system.

Data structure - algorithms

We present some of the basic algorithms used to manipulate the Extensible Tree data structure, as well as some of those necessary to model root systems. Of particular interest will be treatment of algorithms using the

Extensible Tree as a storage medium for the root model data. The algorithms which allow the growth model to determine the structure of the Extensible Tree are also explained.

Traversal of a data structure is the act of visiting each node of the tree in an ordered manner. Traversal of the Extensible Tree data structure is accomplished by a simple recursive algorithm. When Traverse is called for a given node (root entity) a set of predefined calculations are carried out using the data stored in that node. As these calculations are completed the routine calls itself to process the first in its list of children. Hence, the routine is executed for the sub-tree starting at the first child. When this is done, a return to the original call is made and the routine again calls itself for the sub-tree starting with the right sibling. When Traverse is called for the root vertex every node in the tree will be visited. As an example, Traverse will visit the nodes of the tree in Figure 2 in the order: 1, 2, 4, 3, 5, 6, 7.

Branch creation

When a root system is allowed to develop, the branching process will create new root axes. The addition of a root axis necessitates the addition of a node to the Extensible Tree in which to store the associated information. As outlined above, two angles are sufficient to describe the branching geometry. When a new branch is necessary, as determined by the root model, the CreateBranch routine is called with a pointer to the parent root, branch type, branching angles, and branch length, as the parameters. The first few steps of the algorithm initialize the pointers contained in the new node. This includes setting the pointer to the parent node and initializing the pointer to the list of children to null. The new node is also identified with the proper root type for the branch being created. Next, the origin of the new root is placed in the data field. Then the CreateBranch routine calls a subroutine called ComputeNewDirection that uses the radial and lateral branching angles to calculate the direction of the new root axis using the calculations presented above. It then places this direction in the correct field of the new node. Finally the new node is placed in the appropriate position in the list of children of the parent node.

Growth increment algorithm - segment creation

To achieve linear growth, the system is traversed at each time step and each root axis is incremented in length. As mentioned above, the length increment generated at each time step is stored as a separate segment. The algorithm used to perform the IncrementLength operation computes the growth increment as a function of the root growth parameters for the current root. The routine then allocates memory for a new root segment and appends it to the last segment of the current root. The length of the new segment is computed according to the root growth function. The direction of the new segment will always be based on the direction of the previous segment. It may, however, be modifed in order to take into account geotropism and/or a randomness factor. Geotropism is the downward growth tendency of roots. The randomness factor takes into account interspersion of rocks and other obstacles which will cause random variation in the directions of successive root segments.

Calculation of volume and surface area

Volume and surface area are two important quantities for calculating correlates of root system efficiency. A single axis root may be approximated by a series of truncated cones as shown in Figure 3. The volume of a root system is then simply the volume of these truncated cones summed over all segments of all root axes in the system. With l as the segment length, R_1 the radius at the beginning of the segment, and R_2 at the end, the volume of the root system may be computed as:

$$V = \sum_{\text{roots}} \Big(\sum_{\text{segments}} \frac{1}{3}\pi(R_1^2 + R_2^2 + R_1R_2) \Big) \qquad (5)$$

The surface area may similarly be computed, without circular ends, as:

$$S = \sum_{\text{roots}} \Big(\sum_{\text{segments}} (R_1 + R_2)\sqrt{(R_1 - R_2)^2 + l^2)} \Big) \qquad (6)$$

A recursive algorithm based on Traverse may be used to perform these calculations for volume and surface area with the proper routine used for processing each node. As the curvature of the root surface increases along the axis of the root nearer the tip, good approximations of volume and surface area require that the last segments be subdivided for the calculations. That is, more than one truncated cone is used to model those segments near the tip of the root. Also, the very tip can be seen to deteriorate into a cone as the radius at the tip is zero.

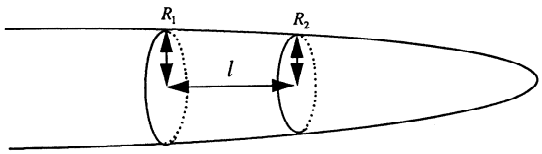

Figure 3. Truncated cones were applied to approximation of volume and surface area of single root axes. The volume and surface area of a root system were calculated as the integrated volume or surface area, respectively, of all segments of all roots.

Visualization of root growth

Graphic visualization provides the most straightforward means of comparing a system model to a real root system. The methods used to visualize the model must provide basic geometrical information. More advanced three dimensional visualization has been applied to enhance the realism of the simulation output. Furthermore, the importance of three dimensional rendering increases as three dimensional data are incorporated into the model. The data structure and visualization code provides for continuous viewing of the axial (wireframe) structure of the root as well as for continuous solid representation of the root as it grows. In addition, simple calculations such as surface area and volume are provided, as well as a number of root structure indices useful for the comparison of botanical root systems. The root growth model implemented is primitive at this point in the sense that the growth functions are linear. However, the data structure provides ease of expansion. A small number of routines embody the basic parts of the root growth model that must be supplied. Growth parameters comprising different root growth functions may be easily added or deleted from the set of fields in the data structure. For the wireframe rendering mode the tree is recursively traversed, drawing all segments of a root. The routine then proceeds to call itself for the child and right sibling in turn. The option to draw a multi-colored wireframe is also provided as a means of examining patterns of daily growth by assigning a distinct color to each days growth. Since days are integers, the day of growth may serve as the index in a color table. The concept of the truncated cone is carried into the routines which render the three dimensional solid image of the root system. The routine for rendering a shaded image is identical in layout to that for the wireframe. Instead of a single line however, each segment is rendered as a truncated cone using a collection of shaded polygons in the form of a triangular mesh. The vertices of the polygons are

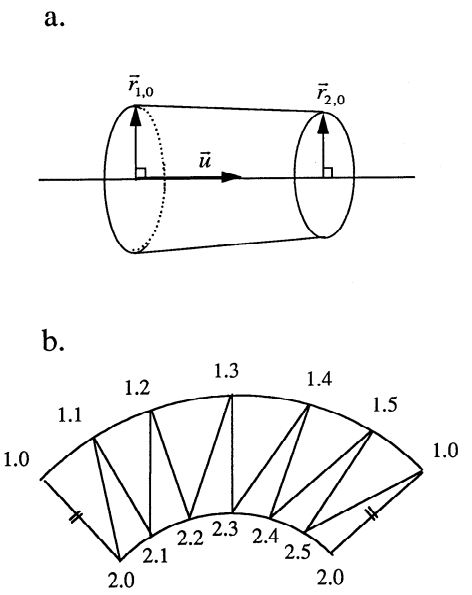

Figure 4. Visualization of a three dimensional solid image of the root system is made possible by an approximation to truncated cones using a collection of shaded polygons, as truncated cones in the form of a triangular mesh. **a**) The vertices of the polygons are formed by finding the perpendicular unit normal $\vec{n}$ to the central axis $\vec{u}$ of the segment being rendered. $\vec{r}_{1,0}$ and $\vec{r}_{2,0}$ are then formed by multiplying unit normal $\vec{n}$ by the magnitude of the radius at the respective end positions. **b**) A set of vertices are then created by rotating $\vec{r}_{1,0}$, and $\vec{r}_{2,0}$ about the segment directional vector at $\vec{u}$ regular intervals.

formed by finding the perpendicular unit normal $\vec{n}$ to the central axis $\vec{u}$ of the segment being rendered. This normal may be calculated using Equation 1. Note that the normalized perpendicular is the same at both ends of the segment, and that the calculated result will not necessarily be in the orientation shown in Figure 4. This unit normal $\vec{n}$ is then multiplied by the magnitude of the radius at the respective end positions to form $\vec{r}_{1,0}$ and $\vec{r}_{2,0}$ depicted in Figure 4. Rotating these position vectors about the segment direction vector at regular intervals will produce the set of vertices necessary to delineate the polygonal construction shown. These rotations are accomplished using the method outlined in Equation 2. Specifically, the position vectors are given as:

$$\vec{r}_{i,j} = \left[R(\vec{u}, j\,\frac{\pi}{3}\right]\vec{r}_{i,o} \quad I = 1,2 \quad j = 1,2,3,4,5 \quad (7)$$

Note from the "unfolded" view that the original points $\vec{r}_{1,0}$ and $\vec{r}_{2,0}$, are also used at 2π to prevent gaps in the shell otherwise brought about by numerical error in the calculations of Equation 7.

If the growth model generates random directional variations from segment to segment, the radial points generated about the end of one segment will not be identical to those generated for the next segment. This mismatched condition will cause gaps in the rendered image. To close such gaps, the second set of points in a given segment are used as the first set in the following segment, effectively eliminating the gaps. Again, as with the volume and surface area calculations, segments near the tip are subdivided for rendering, and the representation will degenerate to a cone at the tip of the root. These rendered images serve well to quickly verify the shape of root system model in real time. Figure 5 shows an example of a bean root displayed in wire, color coded wire, and solid display modes. The viewer provides for rendering of wireframe and solid representations of the root system being simulated. These images may be rotated in three dimensions through mouse control and a wireframe bounding box or cylinder may be superimposed over the image. A depth scale may be presented next to the root system on request. The results of functional analysis of the root system are also available in the viewing window.

Description of root axes processes using a kinematic approach.

A kinematic approach to root axes allows the distinction of changes that occur along a root axis as a function of growth, from those changes that occur independently of growth processes, thus making it possible to address both spatial and temporal dynamics of root response to various environmental stimuli (Silk, 1984; Silk and Erickson, 1979). The Extensible Tree data structure presented here is available at any time during a simulation, thus permiting computation of age and theoretical velocity along the root axis of any segment. The following equation can be used to describe total strain rate, L ($mm^3\ mm^{-1}\ h^{-1}$), a measure of the local relative rate of volume change along a root axis (Silk et al., 1986):

$$L = \frac{\delta v_z}{\delta z} + \frac{2}{R}\frac{\delta R}{\delta z} v_z \qquad (8)$$

where R is root radius (mm), z is distance from the root tip (mm), and v_z is the longitudinal growth veloc-

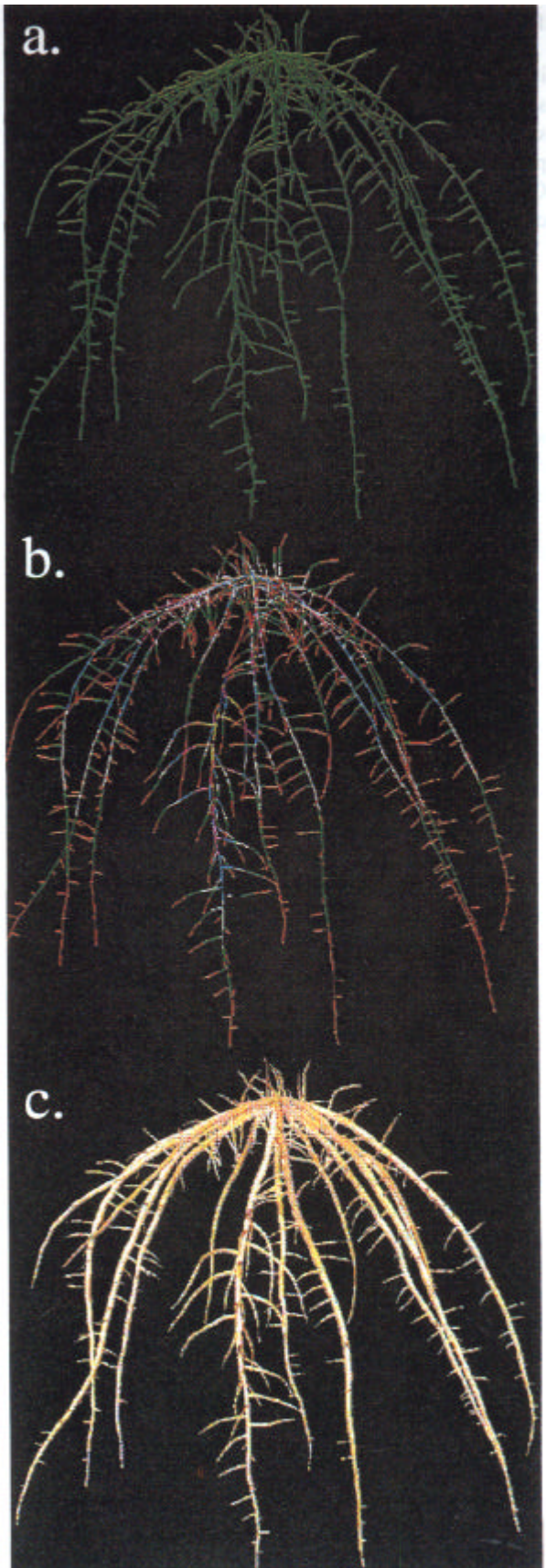

Figure 5. a) Wire, b) color coded wire, where each color represents 24 hours of root growth, and c) solid root display modes for a bean root.

ity (mm h^{-1}). Hence, the first term on the right hand side represents the local relative elongation rate and the second term represents the radial and tangential components of the relative growth rate. Nutrient uptake, CO_2 respiration, exudation, carbohydrate deposition can be described as by Silk et al. (1986):

$$D = \frac{\delta S}{\delta t} + \frac{\delta (Sv_z)}{\delta z} \qquad (9)$$

where D represents the local rate of e.g. CO_2 respiration, exudation, carbohydrate deposition, nutrient uptake (mol mm^{-1} root length h^{-1}) and S is the local cumulative amount of substance e.g. respired CO_2. Hence, $\delta S/\delta t$ represents the local rate of change (e.g. CO_2 respiration rate) and $\delta (Sv_z)/\delta z$ represents the growth associated rate of change. According to Sharp et al. (1990) the growth associated rate of change can be expanded to:

$$\frac{\delta (Sv_z)}{\delta z} = S\frac{\delta v_z}{\delta z} + v_z\frac{\delta S}{\delta z} \qquad (10)$$

where the first term on the right hand side represents the dilution due to growth (change in cumulative amount of e.g. respired CO_2 due to tissue expansion) and the second term represents the convective term (change in e.g. CO_2 respiration rate due to movement of cells away from the apex).

Hard- and software environment

SimRoot is implemented in ANSI C on a Silicon Graphics INDY workstation with 64 Mb RAM and the IRIS GL graphics language (Neider, 1993). Visualization is substantially slower on 32 Mb machines. A sample of *SimRoot* visual output, the Extensible Tree data structure C declaration, the segment structure C declaration, the Extensible Tree traversal routine, as well as the branch creation algorithm is available on the World Wide Web at http://azalea.lscpe.psu.edu/faculty/lynch.html. The algorithms are presented in a pseudo-code that is recognizably based on the C programming language. We will provide a compiled version (approximately 500 kb) of *SimRoot* to interested parties upon request.

Example applications

Variable root system topology

The root systems shown in Figure 6 demonstrate the ability to handle single axis or multiple axes roots as well as a purely dichotomous branching pattern. In general, utilizing the Extensible Tree data structure any branching pattern which occurs in nature can be simulated on the *SimRoot* platform presented herein.

Data structure topology vs. root system topology

Using examples similar to those shown in Figure 6, but much earlier in development, Figure 7 shows the topology of the resultant Extensible Tree data structure for each respective root system. Note that in all cases the Extensible Tree assumes a topology directly related to that of the root system being modeled, thereby preserving the topological relationships of the independent root axes. Because of the amorphous nature of the Extensible Tree it will often take forms similar to other well known data structures. Note in particular the case of purely dichotomous branching where the Extensible Tree degenerates to a form very similar to that of a binary tree.

Varying parameters

The *SimRoot* platform has the ability to change drastically the geometric aspects of a simulated root system by changing the functional form of the growth model or simply the parameters of that model. Using Figure 8 as an example, a root system of herringbone topology, illustrates root models varying from a well-defined topological model, to a model incorporating variation in growth direction, to a model with stochastic elements applied to both topology and direction of growth. Similar effects can be obtained for any root growth model incorporated into the system. The models in Figure 8 are not biologically based. The variations are based on modifying the unit direction vector of each segment randomly, within a predefined range. The standard C normal distribution random number generator is used to produce a random unit direction vector. The sum of the original direction vector and the random vector, its length reduced to a predefined percentage of a unit vector, are then normalized to produce the modified direction vector.

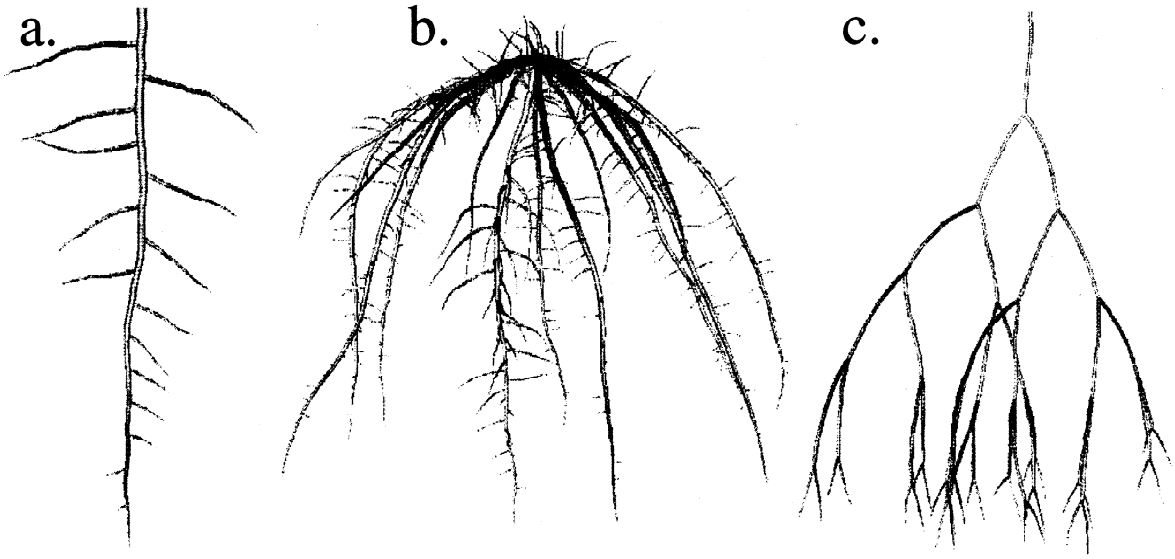

Figure 6. Examples of contrasting root architecture. **a**) Herringbone, **b**) bean, and **c**) dichotomous branching patterns.

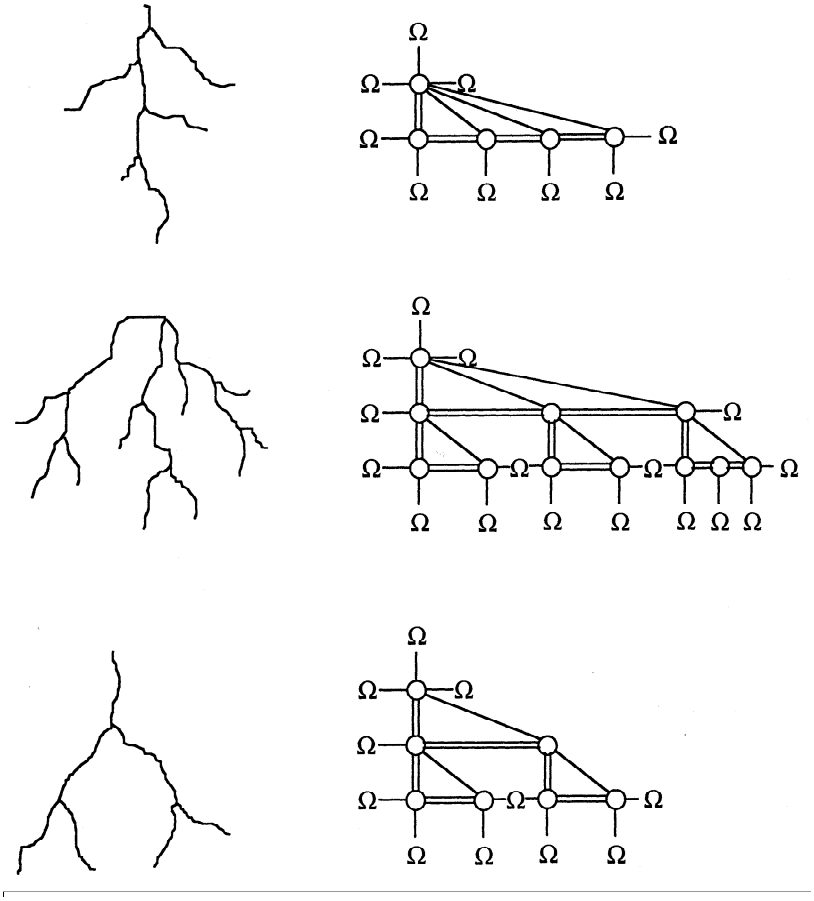

Figure 7. Data structure topology as a function of root system topology. Topology are similar to examples shown in Figure 6, although much earlier in development.

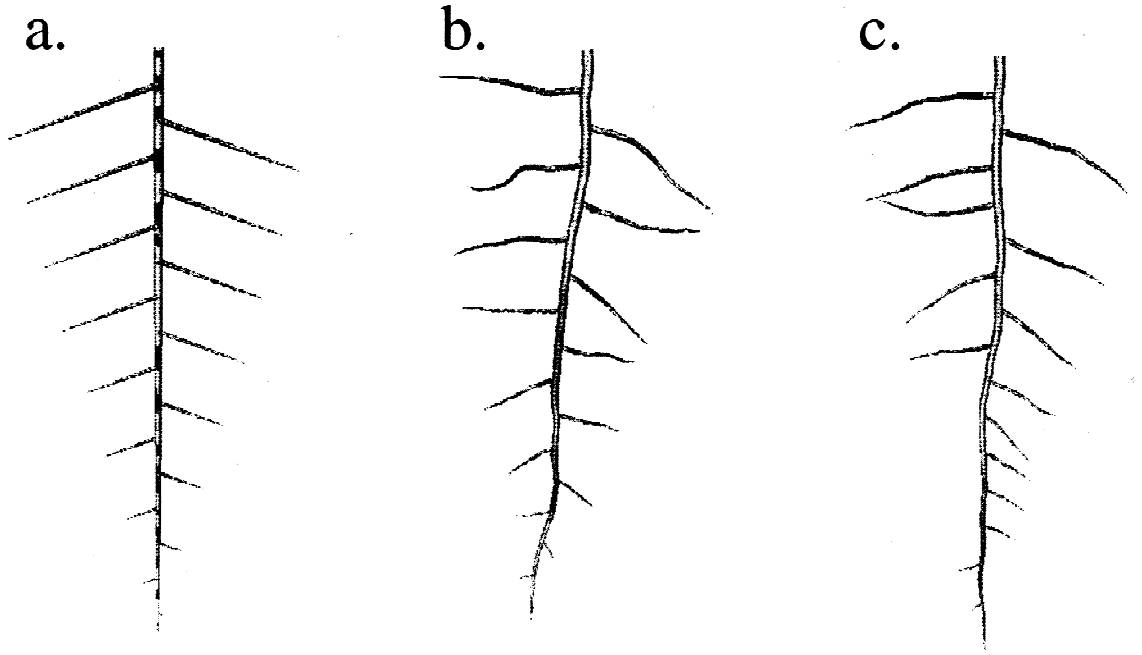

Figure 8. The effect of model variation. **a**) No variation applied to a well defined herringbone topology, **b**) variation applied to direction of growth, and **c**) variation applied to both topology and direction of growth.

Estimation of carbon consumption (cost function)

Since further physiological work is needed at high spatial resolution before models can really benefit from a kinematic approach, a much simpler technique has been used to describe cost:benefit functions in *SimRoot* so far. Respiration, C exudation, and biomass deposition along root axes were measured on *Phaseolus vulgaris* seedlings under laboratory conditions (Nielsen et al., 1994). Arrays with values along the root axes were generated based on these measurements and total respiration, C exudation, and biomass deposition were recursively computed by traversal of the Extensible Tree data structure.

Estimation of depletion volume (benefit function)

Depletion volume was calculated based on diffusion of a given nutrient in the rhizosphere (Nielsen et al., 1994). The collective nutrient depletion zones of all root segments from the *SimRoot* program were converted into a volume element location presentation, where each element represented a cubi-form volume (voxel) within the depletion cylinder (Figure 9). Duplicate voxels, representing inter-root competition, were then eliminated and the depletion volume without the overlap was estimated. This procedure was similar to that described by Fitter et al. (1991).

Fractal geometry analysis

Estimation of three-dimensional fractal dimension (FD) - *SimRoot* and the subprogram Fractal visualized and estimated the number of boxes that intercept the simulated root system in a similar manner as the volume element locations. In the case of box intercepts the zone around each root segment is equal to half the distance between each volume element location. In accordance with the procedures described by Tatsumi et al. (1989), log number of boxes intercepted by the simulated root system can be plotted against log box size. By applying linear regression, a fitted line with slope (D) can be obtained. By definition FD =-D.

Estimation of planar fractal dimension - From the output data structure used in *SimRoot* a point list file describing the positions where the root system was growing through a user defined set of horizontal planes was produced for the estimation of planar FD. For the calculation of planar FD and projection FD, a two-dimensional grid was superimposed on the planes and the number of squares intercepted by the simulated root

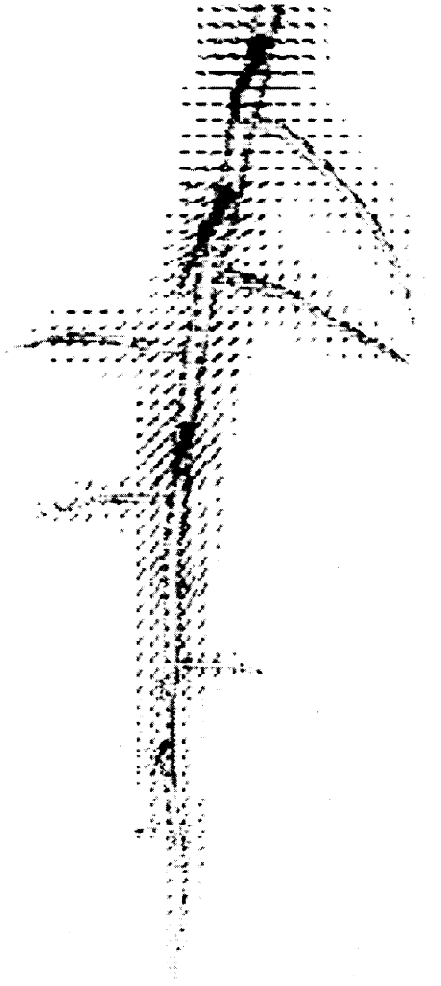

Figure 9. Depletion volume around a root represented by volume element locations (voxels).

system on each plane was estimated for various square sizes (e.g. 0.4, 0.8, and 1.6 cm).

Estimation of linear fractal dimension - Another point list file describing the positions where the root system was intercepting a user defined set of vertical lines was produced for the estimation of linear FD. For the calculation of linear FD a one dimensional grid was superimposed on the lines and the number of line segments intercepted by the simulated root system on each plane was estimated for various segment lengths.

Estimation of projected fractal dimension - Each segment of the root system was described by a starting point (x, y, z), and a vector describing length and growth direction of the segment (dx, dy, dz). A total of nine different projection planes, were produced by rotation of the root system around two coordinate axes before the point lists were generated. The angles of rotation θ were $\pi/6$ or $\pi/3$, each creating three projection planes. Hence, the starting points after rotation (x', y', z') were computed as:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} (x * \cos\phi) - (y * \sin\phi) \\ (((x * \sin\phi) + (y * \cos\phi)) * \cos\phi) - z * \sin\phi) \\ (((x * \sin\phi) + (y * \cos\phi)) * \sin\phi) - z * \cos\phi \end{bmatrix} \quad (11)$$

The vector describing length and growth direction after rotation (dx', dy', dz') are computed as:

$$\begin{bmatrix} dx' \\ dy' \\ dz' \end{bmatrix} = \begin{bmatrix} (dx * \cos\phi) - (dy * \sin\phi) \\ (((x * \sin\phi) + (dy * \cos\phi)) * \cos\phi) - dz * \sin\phi \\ (((dx * \sin\phi) + (dy * \cos\phi)) * \sin\phi) - dz * \cos\phi \end{bmatrix} \quad (12)$$

In order to estimate the projection FD, a point list of the <x,y,z> positions of boxes intercepted by the root system was generated. Sequentially, one of the coordinates for each point in the point list was removed (x, y, or z) and duplicate points were eliminated, giving the positions of squares on the three coordinate planes (<x,y>, <x,z>, and <y,z>) that had been intercepted by the projected root system.

Outlook

SimRoot generates a realistic solid model of root geometry that includes root radius in addition to the skeletal or 'wire-frame' views of earlier models. This has utility in geometric analyses of root architecture, as we demonstrate in this paper as regards fractal geometry, and as we have described earlier (Nielsen and Lynch, 1994). Our fractal analyses indicate that geometric features such as root radius can have significant effects on the overall geometric properties of the root system. Because of the difficulty in analyzing and quantifying the geometry of objects as complex as plant root systems, a geometric simulation of root architecture may be useful in developing analytical approaches that may be useful in evaluating actual root systems. In this regard fractal geometry is particularly appealing as it condenses a great deal of information about the space-filling properties of an object in a quantitative form (see discussion in Berntson et al., 1996).

One of the most important features of this model as well as the models of Fitter et al. (1991) and Clausnitzer and Hopmans (1995), is the ability to model the relationship between the form of the root system and its functional properties such as resource acquisition or efficiency. In this respect the incorporation

of kinematic functions in *SimRoot* may permit the explicit treatment of the well-known and substantial physiological variation within and between root axes of the same root system. The kinematic framework permits the discrimination of spatial patterns arising from differences in growth processes, from patterns arising through changes in the physiology of cells as they mature. A limitation to the application of this approach in modelling is that relatively few data are available concerning root processes at fine spatial resolution (typically 1 mm). Another limitation is that much published literature is based on studies of primary roots of seedlings, excised roots, or of roots of undefined class, whereas in a spatially explicit model information for each root class is needed. Our own work with the variation in C 'cost' functions along root axes is a step towards a data set that could be employed in *SimRoot* in its development from a purely geometric model to a functional, or physiological model.

We foresee the continued development of *SimRoot* in several aspects in order for it to become more useful as a model of organismic and agroecological processes. At the organismic level an interaction with shoot processes as sinks for soil resources and sources for carbohydrate is needed, as implemented for the model of Clausnitzer and Hopmans (1995). Our work with *SimRoot* so far has focused on the common bean (*Phaseolus vulgaris* L.) as a model system, and well-validated shoot growth models for bean and other grain legumes exist (Hoogenboom, 1992). Another shortcoming is that although *SimRoot* is capable of considering spatial variation in root processes, it treats the soil as a uniform media, which is of course a gross simplification. Clausnitzer and Hopmans (1995) dealt with this through finite-element modelling of the soil environment. This is a promising approach, but for immobile resources such as P (an interesting resource from the perspective of root architecture, since it is relatively immobile), a finite-element approach would require many thousands of nodes, requiring supercomputer processing capability with present technology. Another feature that should be incorporated in the model in the future is the action of multiple resource constraints on root form and function, as elegantly demonstrated for models of shoot architecture by Farnsworth and Niklas (1995). Water and P may be useful starting points for this analysis, since water acquisition is related to the acquisition of many water-soluble nutrients, and is essential for root growth, whereas P represents the most immobile primary soil resource. As a step towards consideration of ecological processes, it would be fairly easy to simulate a root system surrounded by other root systems, in order to simulate competition between plants for resources. A problematic feature of actual roots that is difficult to consider in a spatial model is the nearly ubiquitous presence of mycorrhizal associations, which alter root topology (Schellenbaum et al., 1991) and root function, but whose geometric features are complex and poorly characterized.

Acknowledgements

We thank Amram Eshel and Howie Weiss for helpful discussions. This research was supported by USDA/NRI grant 94371000311 to JPL.

References

Berntson G M 1994 Modelling root architecture: are there tradeoffs between efficiency and potential of resource acquisition? New Phytol. 127, 483–493.

Berntson G M, Lynch J P, and Snapp S 1996. Fractal geometry and plant root systems: current perspectives and future applications. *In* Fractals in Soil Science. Eds. P Baveye, J Y Parlange and B A Stewart. Lewis Publishers, New York, USA. *(In press)*.

Brouwer R 1954 Water absorption by the roots of *Vicia faba* at various transpiration strengths. I. Analysis of the uptake and the factors determining it. Proc. K. Ned. Akad. Wet. C54, 106–115.

Clarkson D T 1996 Root structure and sites of uptake. *In* Plant Roots: The Hidden Half. 2nd ed. Eds. Y Waisel, A Eshel and U Kafkafi. pp 483–510. Marcel Dekker, New York.

Clausnitzer V and Hopmans J W 1995 Simultaneous modelling of transient three-dimensional root growth and soil water flow. Plant Soil 164, 299–314.

Diggle A J 1988 ROOTMAP - a model in three-dimensional coordinates of the growth and structure of fibrous root systems. Plant Soil 105, 169–178.

Eshel A and Waisel Y 1996 Multiform and multifunction of various constituents of one root system. *In* Plant Roots: The Hidden Half. 2nd edition. Eds. Y Waisel, A Eshel and U Kafkafi. pp 175–192. Marcel Dekker, New York.

Farnsworth K D and Niklas K J 1995 Theories of optimization, form and function in branching architecture in plants. Func. Ecol. 9, 355–363.

Fitter AH, Strickland T R, Harvey M L and Wilson G W 1991 Architectural analysis of plant root systems. 1. Architectural correlates of exploitation efficiency. New Phytol. 119, 375–382.

Hackett C and Rose D A 1972a A model of the extension and branching of a seminal root of barley, and its use in studying relations between root dimensions. I. The model. Aust. J. Biol. Sci. 25, 669–679.

Hackett C and Rose D A 1972b A model of the extension and branching of a seminal root of barley, and its use in studying relations between root dimensions. Il. Results and inferences from manipulation of the model. Aust. J. Biol. Sci. 25, 681–690.

Hayward H E, Blair W M and Skalling P E 1942 Device for measuring entry of water into roots. Bot. Gaz. 104, 152–160.

Hoogenboom G, Jones J W and Boote K J 1992 Modeling growth, development and yield of grain legumes using SOYGRO, PNUTGRO, and BEANGRO: A review. Trans. ASAE 35, 2043–2056.

Huang J W, Grunes D L and Kochian L V 1992 Aluminum effects on the kinetics of calcium uptake into cells of the wheat root apex. Planta 188, 414–42I.

Lungley D R 1973 The growth of root systems - a numerical computer simulation model. Plant Soil 38, 145–159.

Lynch J 1995 Root architecture and plant productivity. Plant Physiol. 109, 7–13.

Lynch J P and Nielsen K L 1996 Simulation of root system architecture. *In* Plant Roots: The Hidden Half. 2nd ed. Eds. Y Waisel, A Eshel and U Kafkafi. pp 247–258. Marcel Dekker, New York, NY, USA.

McDougall B M and Rovira A D 1970 Sites of exudation of ^{14}C-labelled compounds from wheat roots. New Phytol. 69, 999–1003.

Neider J 1993 Open GL Programming Guide: The Official Guide to Learning Open GL, release 1. OpenGL Architecture Review Board. Addison-Wesley Publishing Co., Reading, MA, USA. 516 p.

Nielsen K L, Lynch J P, Jablokow A G and Curtis P S 1994 Carbon cost of root systems: an architectural approach. Plant Soil 165, 161–169.

Pachepsky Y, Timlin D, Acock B Lemmon H and Trent A 1993 2DSOIL - A New Modular Simulator of Soil and Root Processes. Release 02 (April 1993) USDA-ARS, Systems Research Laboratory, Beltsville, MD, USA. 209 p.

Pages L, Jordan M O and Picard D 1989 A simulation model of the three-dimensional architecture of the maize root system. Plant Soil 119, 147–154.

Schellenbaum L, Berta G Ravolanirina F, Tisserant B, Gianinazzi S and Fittter A H 1991 Influence of endomycorrhizal infection on root morphology in a micropropagated woody plant species (*Vitis vinifera* L.). Ann. Bot. 68, 135–141.

Sharp R E, Hsiao T C and Silk W K 1990 Growth of the maize primary root at low water potentials. II. Role of growth and deposition of hexose and potassium in osmotic adjustment. Plant Physiol. 93, 1337–1346.

Silk W K 1984 Quantitative descriptions of development. Ann. Rev. Plant Physiol. 35, 479–518.

Silk W K and Erickson R O 1979 Kinematics of plant growth. J Theor. Biol. 76, 481–501.

Silk W K, Hsiao T C, Diedenhofen U and Matson C 1986 Spatial distribution of potassium; solutes, and their deposition rates in the growth zone of the primary corn root. Plant Physiol. 82, 853–858.

Tatsumi J, Yamauci A and Kono Y 1989 Fractal analysis of root systems. Ann. Bot. 64, 499–503.

Waisel Y and Eshel A 1992 Differences in ion uptake among roots of various types. Plant Nutr. 15, 945–958.

Wanner H 1944 The zonal graduation of respiratory intensity in the root. Arkiv Bot. 31A, 1–9.

Wullschleger S D, Lynch J P and Berntson G M 1994 Modeling the belowground response of plants and soil biota to edaphic and climatic change - What can we expect to gain? Plant Soil 165, 149–160.

Section editor: B E Clothier