Chapter 14: Example Problems

Dennis Timlin and Yakov Pachepsky

Four examples are presented in this section. The purpose of the examples is (1) to illustrate the kinds of two-dimensional root and soil processes that can be simulated and (2) to give examples of data preparation for users of 2DSOIL.

14.1 Example 14.1: Alternate and furrow irrigation: root development

Alternate furrow irrigation has been proposed as a water conserving irrigation management practice. In this example, the model output is used to compare water movement, root development, and root water uptake for alternate and regular furrow irrigation. The modules that are involved are **SetSur02**, **WatUptake01**, and **WaterMover**. Figure 14.1 gives a graphical representation of the finite element mesh utilized for the numerical simulations. Note that Figure 14.1 shows a gradual increase in spacing between vertical grid lines. A detail of the shape of the grid that represents the two furrows and one ridge of the soil surface is shown in Fig. 14.1b. The row spacing is 50 cm and the furrow depth is 10 cm. The soil hydraulic properties are characteristic for a loamy sand. For the sake of clarity, we assumed a uniform soil profile. The soil water retention and soil hydraulic conductivity relationships are shown in Fig. 14.2. The initial soil water potential was maintained at -300 cm at the 2-m depth and the initial soil moisture content was 0.175 cm^{-3} cm^{-3}. The plant had an initial height of 3 cm and was placed in the middle of the ridge with the stem base at 10 cm in depth. The parameters for the root system corresponded to those for soybean. Soil temperature and soil oxygen concentration were constant over the soil profile. The simulation time began on May 10 (DOY 130) and ended on July 19 (DOY 200).

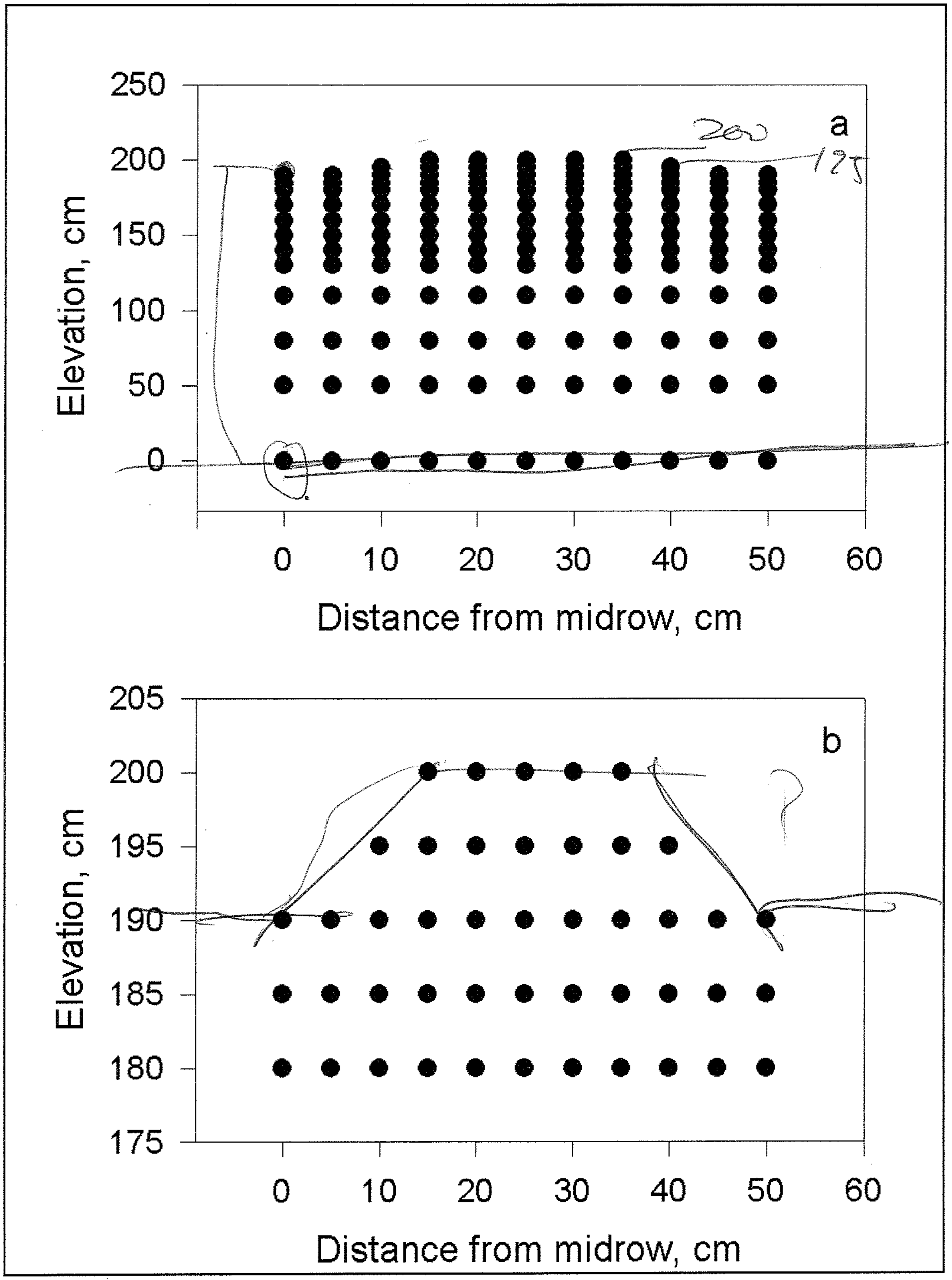

Figure 14.1 Nodes of the grid that covers the soil domain for Example 14.1; a - general pattern, b - upper part. Elevation is measured from the reference plane at 200 cm depth.

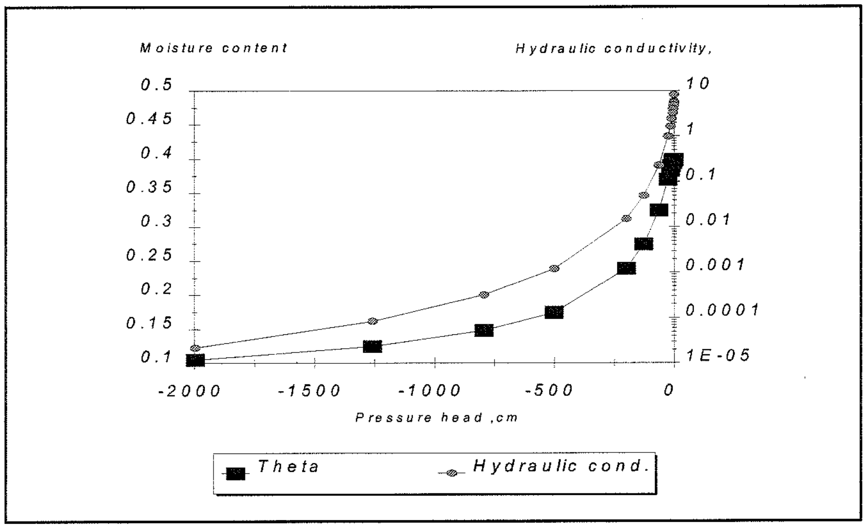

Figure 14.2 Soil hydraulic parameters for Example 14.1.

The weather and irrigation data are summarized in Fig. 14.3. The total amount of rainfall was 3 cm and every two weeks 9.9 cm of irrigation water was applied as flood irrigation giving a total irrigation of 49.5 cm for the season. The irrigation water for regular irrigation was applied in equal amounts to both furrows and only to the left furrow in alternate irrigation simulations. The same total amount of water was supplied for both scenarios.

The calculated root density distributions for July 10 (60 days after planting) are given in Fig. 14.4. The root pattern was nearly symmetrical for regular irrigation. The root pattern was significantly asymmetrical for alternate furrow irrigation because there was less root proliferation under the dry furrow. The depth of the root system was about 50 cm under the ridge. However total root mass was 10 % larger in regular irrigation than in alternate irrigation. The patterns of soil moisture distribution for July 10 are shown in Fig. 14.5. The soil

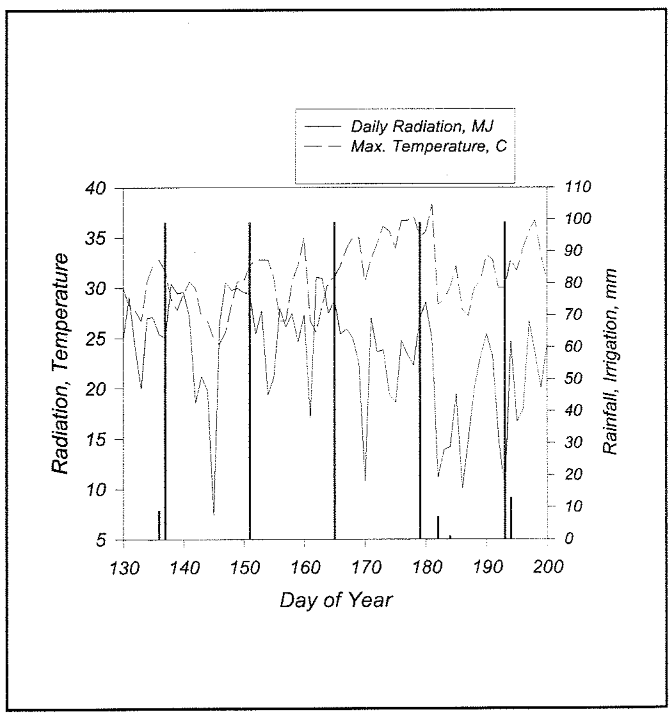

Figure 14.3 Daily weather and irrigation data for Example 14.1.

is drier in the root zone under the stem in the case of regular irrigation than in the case of alternate irrigation. This meant that more carbon was sent to grow roots in the case of regular irrigation because water availability was less. The soil moisture content is lower under the non-irrigated furrow than under the irrigated furrows. The difference is not large mainly because soil hydraulic conductivity is low and decreases very steeply as moisture content diminishes (Fig. 14.2). For regular irrigation, the simulated amount of soil water in the upper 55 cm at the end of the simulation is 13 % less than for alternate irrigation. The data shown here are not sufficient to compare the efficiency of two irrigation techniques for plant yield because a very simple plant model was used. If a more comprehensive plant model was used in place of the simple shoot imitator, the yield efficiency could be easily compared.

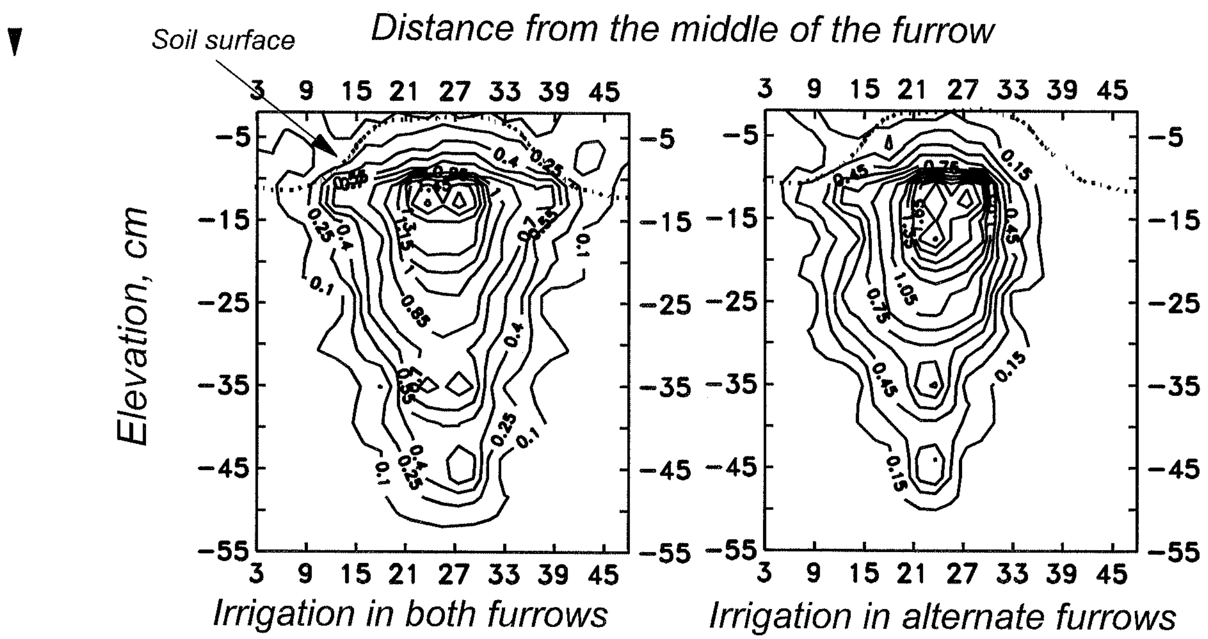

Figure 14.4 Root Density (10^4 g cm^{-3}) distributions for regular (left) and alternate (right) irrigation after 60 days of plant development. Elevation is measured relative to the surface of the ridge and the horizontal coordinate is distance from the middle of the left furrow, cm. Water was applied to the left furrow in the case of alternate furrow irrigation.

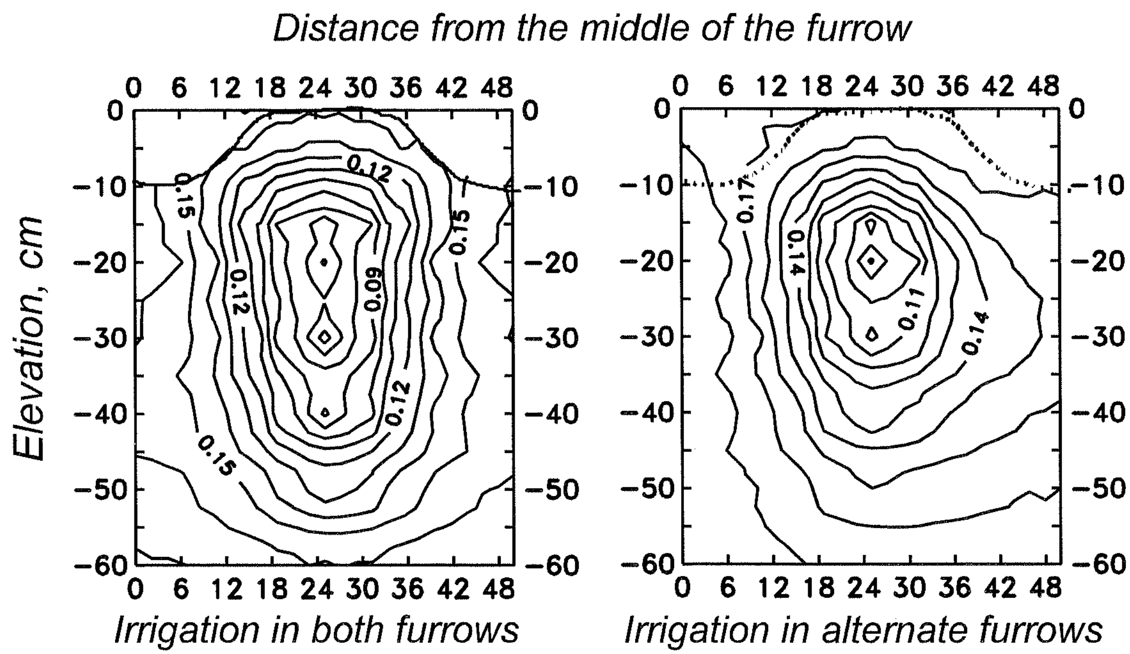

Figure 14.5 Soil moisture content (cm^{-3} cm^{-3}) distributions for regular (left) and alternate (right) irrigation after 60 days of plant development. Elevation is measured relative to the ridge surface and the horizontal coordinate is distance from the middle of the left furrow, cm. Water was applied to the left furrow in the case of alternate furrow irrigation.

14.2 Example 14.2: Soil temperature regime in ridge and interridge zones as affected by plant development

This example is a simulation of soil temperatures in a ridged soil for a three-month growing season. Ridge tillage is a common practice in areas where cool, wet soil conditions occur in the spring. Soil in ridges is generally drier and warms faster in the spring. In this example, however, the effect of ridges under conditions of low moisture availability will be described. The modules that are used are **SetSurf02**, **Heat Mover**, **WatUptake01**, and **WaterMover**.

Figure 14.6 is a graphical representation of the finite element grid at the surface. Below 190 cm the numbering of nodes and elements continues as in the row at 190 cm. The nodes within the ridge are more closely spaced than in the interridge zone. The soil consists of two layers, a sandy loam texture over a loam and the soil hydraulic functions are in Fig. 14.7. Temperature and soil matric potential were held constant at the bottom boundary. The value for temperature was 25 °C and matric potential was -300 cm. The atmospheric data are similar to those presented in Fig.14.3; however, precipitation was less. The details are in the file **Weather.Dat**. The plant was placed at the center of the furrow five cm deep and soybean root parameters were used. Soil oxygen was constant over the profile and solute movement was not simulated. Simulated time began at DOY 130 and ended at DOY 168.

Soil temperatures for the first several days (134-136) of the simulation are shown in Fig. 14.8. Because the plant shades the soil on the ridge, there is no evaporation from the soil surface. As a result, soil temperatures remain near air temperature (30.6 and 32.2 °C) for days 134 and 135. In the interridge zone, however, the soil dries to air-dry and the actual evaporation rate becomes less than the potential rate. A portion of solar radiation now goes to heat the soil. As a result, the additional heat from solar radiation heats the soil in the interridge zones to a temperature greater greater than the air temperature. At 10 cm from the center, the plant shade first covers this node

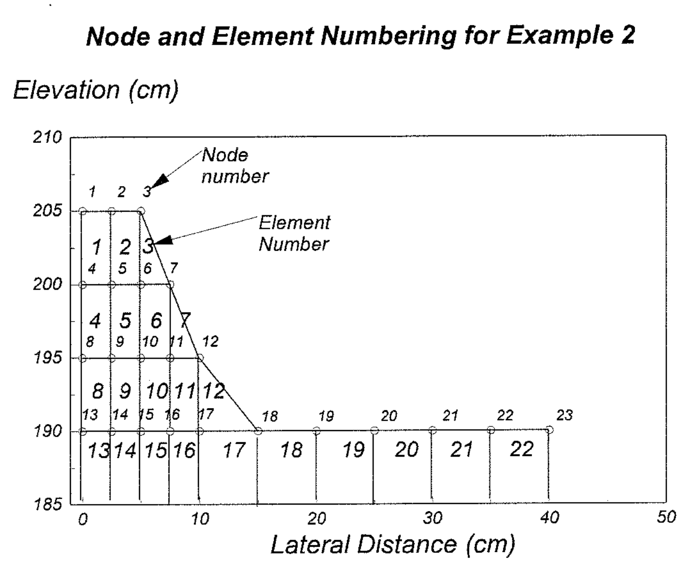

Figure 14.6 Node and element numbering for ridges in Example 14.2

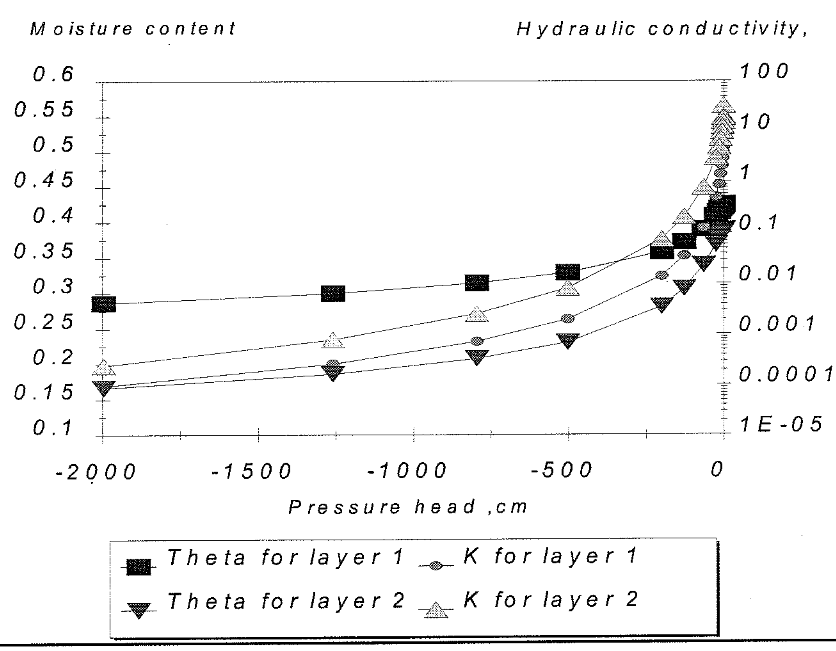

Figure 14.7 Hydraulic functions for the layered soil used in Example 14.2. The surface soil texture is sandy loam, the subsurface texture is loam.

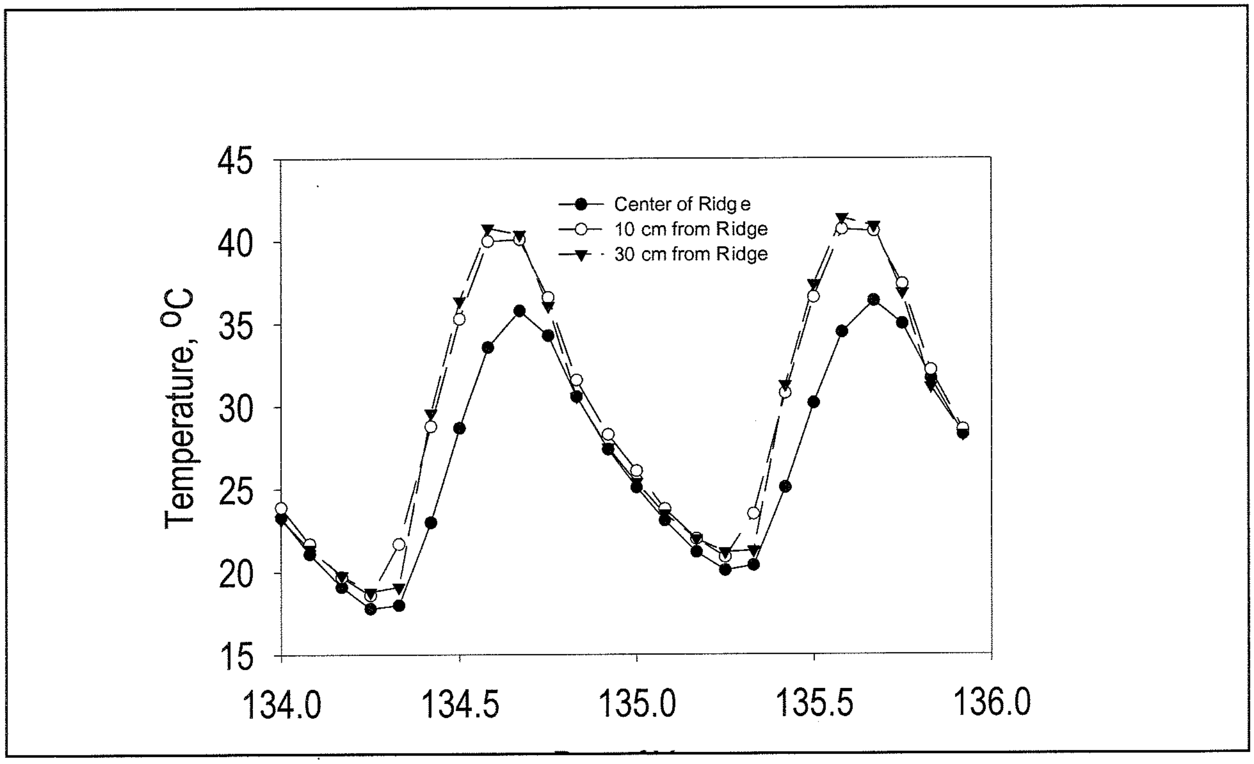

Figure 14.9 Hourly temperature distribution of surface soil in ridge and interridge zones during days of year 134 and 135, before canopy closure.

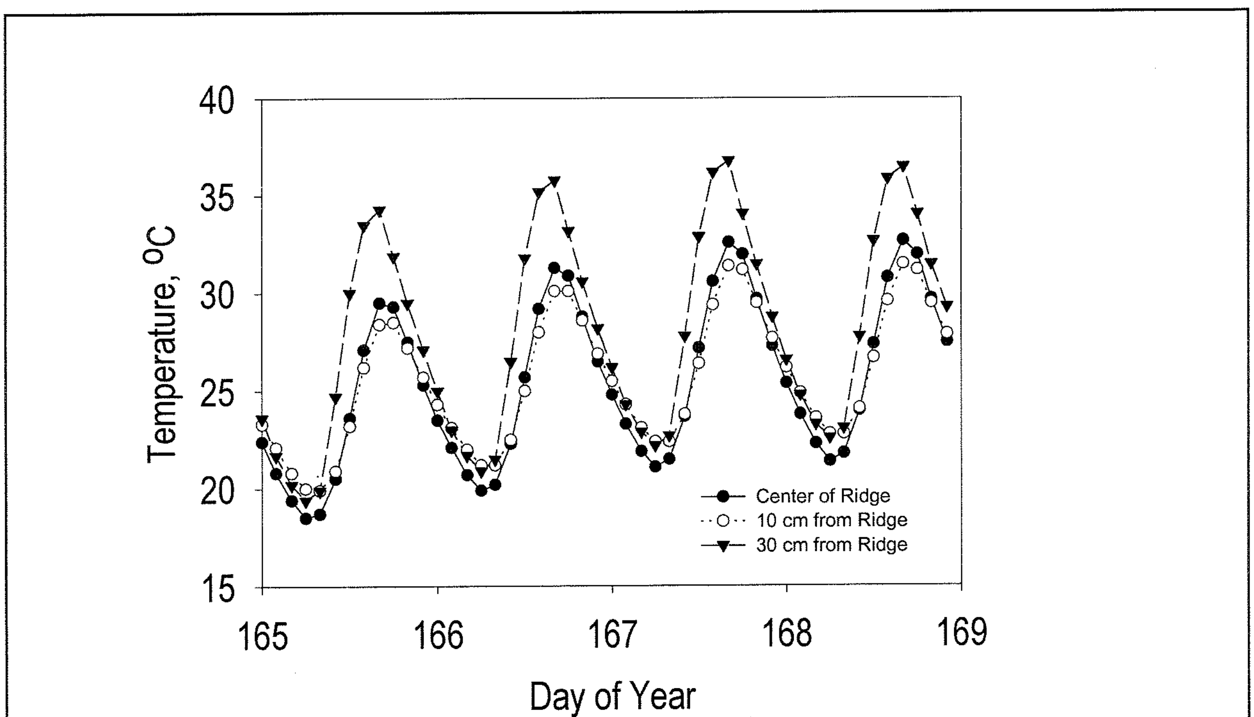

Figure 14.8 Hourly temperature distribution of surface soil in the ridge and interridge zones during days of year 165 through 169, after canopy closure.

near the morning of day 135. This results in the soil cooling to ambient temperature because the sun is no longer heating the soil at this node. Note that this example is given to illustrate the effects of shade on evaporation of water from soil, and on soil temperature. Only a very simple plant model was used.

After day 145 there was no more precipitation. The soil is again allowed to dry by transpiration but plant shade covers all of the interridge zone. As shown in Fig. 14.9 it can be seen that the surface soil in the ridge zone becomes warmer than the soil in the interridge zone and has a wider range of temperature fluctuation. This occurs because the ridge is drier than the interridge zone due to greater root mass and more water uptake. Because the soil on the ridge is drier than the soil in the interridge zone, the heat capacity and thermal conductivity are lower. The soil in the interridge zone can conduct more heat to and from the subsoil and retain more heat, hence it remains cooler during the day and warmer at night.

14.3 Example 14.3: Leaching of the saline soil column

The leaching of dissolved substances in soils involves complex chemical interactions among soil phases. This example demonstrates the influence of gypsum on the concentrations of several selected ions in the soil solution during leaching. Water movement, solute movement and chemical interactions are simulated. Here we consider one-dimensional movement of water and solutes. 2DSOIL still requires two columns of nodes to have one element per depth increment, however. The grid shown in Fig. 14.10 has spacing that is dense at the top of the column and gradually increases downwards. We expect steep gradients of water potential and solute concentrations only in the upper part of the column. Since the horizontal spacing does not matter for one-dimensional simulations, a 1-cm width was chosen. The soil in the column is homogeneous. Cation exchange properties of the soil are illustrated in Fig. 14.11 for solutions of $CaCl_2$-$NaCl$-H_2O and $CaCl_2$-$MgCl_2$-H_2O. The equilibrium contents of exchangeable Na depend not only on the Na fraction in solution but also on total concentration of the ions in solution. The smaller the total concentration, the smaller is the exchangeable sodium fraction provided the fraction of total sodium in the solution is the same. The equilibrium contents of magnesium depend only on the magnesium fraction in the solution. The selectivity coefficient of Ca-Mg exchange at 100 meq/L is close to 1 and the corresponding line in Fig. 14.11 is close to the 1:1 line.

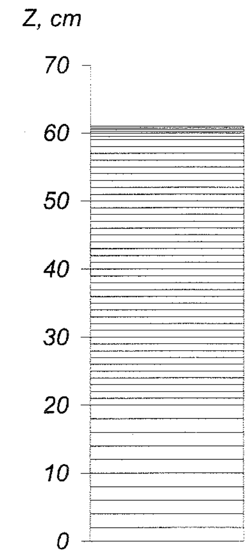

Figure 14.10 Finite element grid for one-dimensional simulations of Example 3.

The initial gypsum and calcite contents were both 0.05 g per g of soil, and the exchangeable cation contents were 0.07, 0.05, and 0.02 eq per kg of soil for Ca^{2+}, Mg^{2+}, and Na^{+}, respectively. The solution was saturated with respect to gypsum and calcite at 0.005 atm partial CO_2 pressure and the partial pressure of CO_2 was held constant in the column.. The sum of cations in the initial soil solution concentration was about 70 meq L^{-1} and the concentrations of cations were 27.4, 20.4, and 22.5 meq L^{-1} for Ca^{2+}, Mg^{2+}, and Na^{+} respectively. The main anion was sulphate, there were 58.2 meq L^{-1} of SO_4^{2-} against 10 meq L^{-1} of Cl^{-}. The water applied for leaching had a very low content of soluble salts: 0.01 meq L^{-1} of Ca^{2+}, Mg^{2+}, Na^{+}, and Cl^{-}, 0.02 meq L^{-1} of SO_4^{2-} and a ponding depth of 2 cm was maintained during the whole run of 40 days simulated time. The initial soil matric potential was -150 cm, and the saturated hydraulic conductivity was 1 cm day^{-1}.

Results of the simulations are summarized in Fig. 14.12 for the top 20-cm layer where noticeable changes occurred during the simulated time. The chloride ion had the simplest behavior as it gradually leached. The main mechanisms for chloride transport included convective transport by moving water and hydrodynamic dispersion due to variations in velocities among different regions of the pore space. The concentration profiles of magnesium and sodium were influenced not only by solute transport mechanisms but also by cation exchange. The concentration of magnesium decreased with time at all depths but the magnesium concentration did not decrease as fast as did the chloride ion's concentration because the solution had a constant source of magnesium in the soil. As gypsum continuously dissolved, calcium ions were exchanged with magnesium. The sodium concentration decreased with time much faster than did magnesium, partly because the Ca-Na cation exchange depended on solute concentration. The graphs in Fig. 14.12 suggest that the leaching of sodium was accelerated by the decrease of total concentration of soluble salts in solution. A small but sharp decrease in sodium and magnesium concentrations during the first day of leaching can be explained by dilution. The infiltration front moved 9 cm by 0.5 day and then to the 16 cm depth after 1 day of simulated time. The decrease in concentration attributed to dilution took place at a depth less than that of the infiltration front.

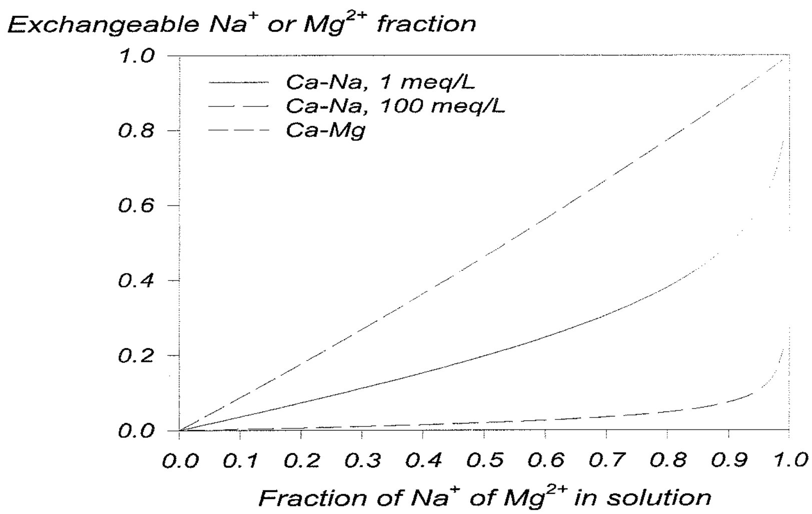

Figure 14.11 Equilibrium contents of cations in the solution and in soil exchangeable complex after binary cation exchange Ca-Mg and Ca-Na from solutions of calcium, magnesium and sodium chlorides.

The sulphate ion concentration was influenced by the dissolution-precipitation of gypsum and solute transport. The sulfate concentration stayed relatively high as long as some gypsum was present in the solid phase. However, the sulfate concentration also gradually decreased with time in the upper part of the column in the presence of gypsum due to the decrease of the total concentration of soluble salts in solution. When the calcium concentration was nearly constant, the sulphate concentration increased as well as the total concentration of soluble salts. At day 40 the gypsum was apparently leached from the upper 2 cm layer and the sulphate ion concentration was decreasing primarily due to convective-dispersive transport without chemical interactions.

The calcium ions demonstrated the most complex behavior as their concentration was dependent on the solubility of gypsum and calcite, cation exchange, and solute transport phenomena. Relatively high levels of calcium were maintained as long as gypsum was present in the solid phase. During the last 30 days, calcium concentration slightly increased with time as a result of the decrease of magnesium sulphate and sodium sulphate in solution. The latter are known to inhibit the solubility of gypsum. After the depletion of gypsum, the calcium concentration began to decrease under the influence of convective-dispersive solute transport. We can expect the calcium concentration to stabilize at the level that corresponds to the solubility of calcite. Because the solubility of calcite is low the calcium concentration had not stabilized during the simulation time.

This simulation illustrates the complex process interactions that can occur in the vicinity of granules of sparingly soluble fertilizers in soil. The importance of soil chemical interactions and solute transport for fertilizer efficiency can be initially examined by means of modeling. In many cases 2DSOIL can be used after addition of an appropriate chemical module.

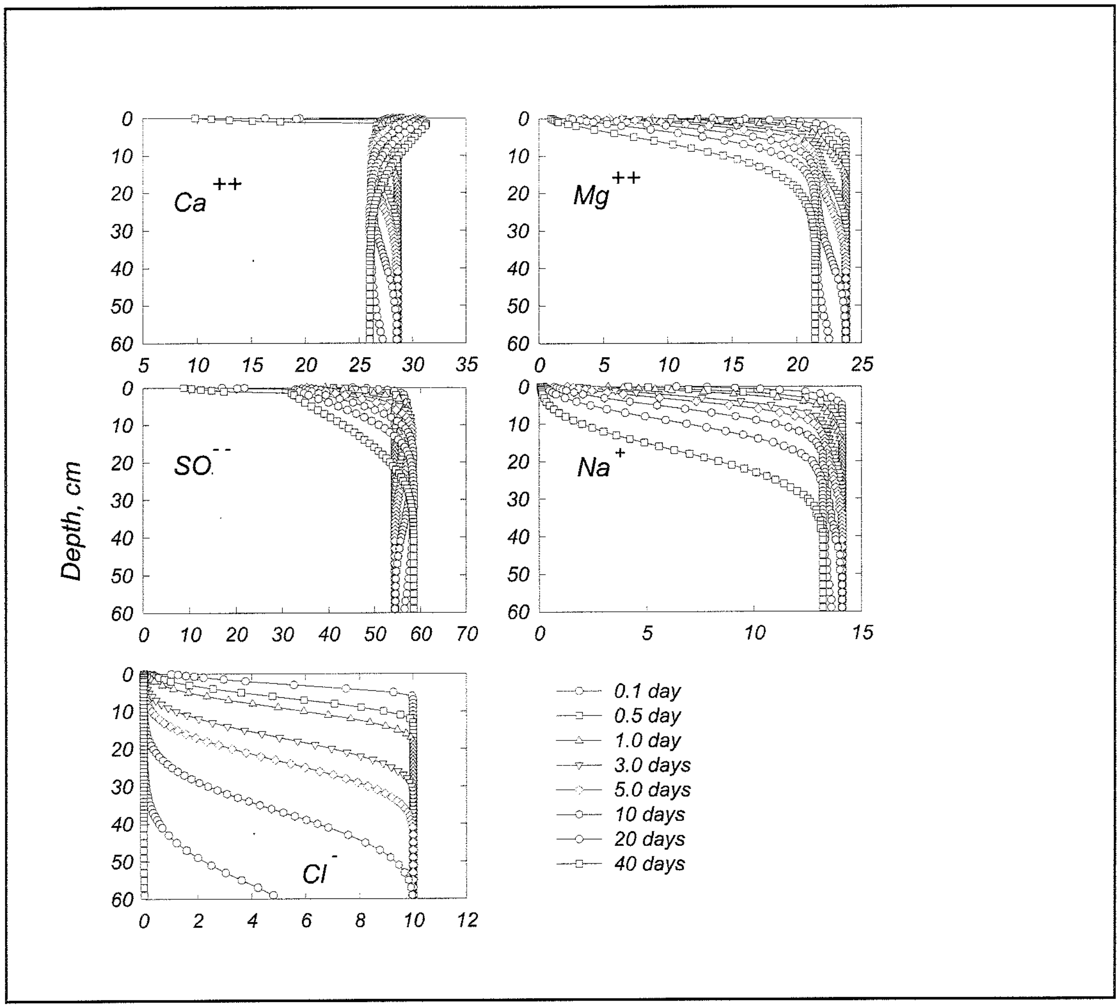

Figure 14.12 Simulated distributions of ions in the soil solution of the soil column at different stages of leaching

14.4 <u>Example 14.4: Nitrogen dynamics in the soil and plant uptake of nitrogen</u>

The purpose of this example is to illustrate the addition and use of the soil nitrogen module **SoilNitrogen**. A new module, **SoluteUptake**, has also been introduced. In this example we present two scenarios, fertilizer banded on the ridge and fertilizer applied evenly all along the soil surface. The other modules used are **Watuptake01**, **Shootimitator01**, **Setsurf02**, **HeatMover**, **SoluteMover**, and **Mngm**. There are two fertilizer applications, one at planting and the other 30 days after planting. For the banded application, fertilizer is initially applied to the top of the ridge. The top dressing is applied at the side of the ridge and buried. In the case of the even application, the top dressed nitrogen is also applied to all the surface nodes. The same amount of fertilizer is applied in both cases. Here, the nodal concentrations of N have been adjusted for the initial fertilization in the initialization file '*Nodal_S.dat*'. A management module has been added to simulate fertilization with additional nitrogen 30 days after germination. This module is described in greater detail in section 15. The grid used in this example has the shape of a ridge-furrow system and is similar to the grid used in Example 14.2. The weather data is shown in Figure 14.13 and the soil hydraulic properties are in Fig 14.7 The initial amount of nitrate-N for the profile in both cases is 169 kg N ha^{-1}. The soil organic matter content is 1.6 kg kg^{-1} soil. It is assumed that 60% of the organic matter is carbon and the ratio of N to C is 1:10.

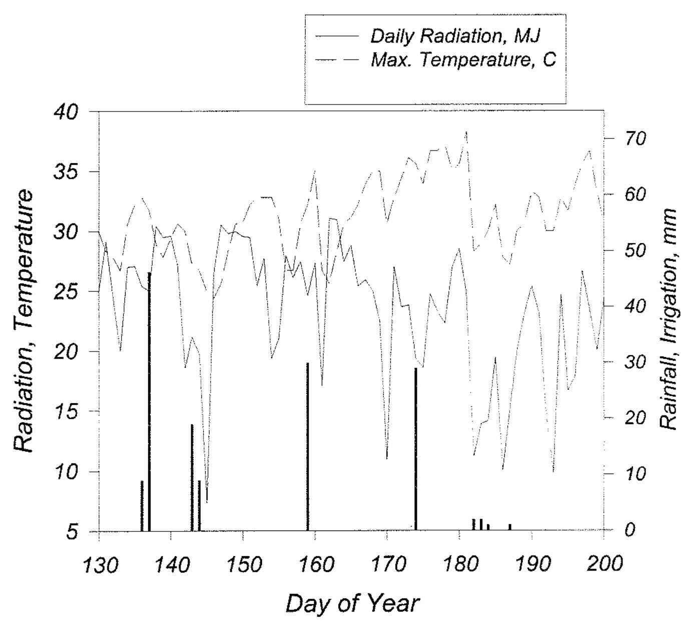

Figure 14.13 Weather data for example 14.4

Nitrogen is added at the rate of 150 kg ha^{-1} for the first fertilization and 100 kg ha^{-1} for the second fertilization. Nitrogen uptake is proportional to the water uptake and the nitrogen concentration in the element from which the water is extracted. The shoot imitator described in Section 10 was used to simulate above-ground plant growth.

Cumulative nitrogen uptake and NO3-N content in the soil as a function of time for both treatments are in Figure 14.14. Total nitrogen uptake for the banded treatment was greater than for the uniform nitrogen application (153 kg ha^{-1} vs 99 kg ha^{-1}). As a result, the NO3-N content of the soil with the uniform N application was greater at the end of the season. Cumulative water uptake for the plant with the banded fertilizer application was also higher (Fig 14.14), this also contributed to the larger N uptake. The plant receiving the banded application also grew taller

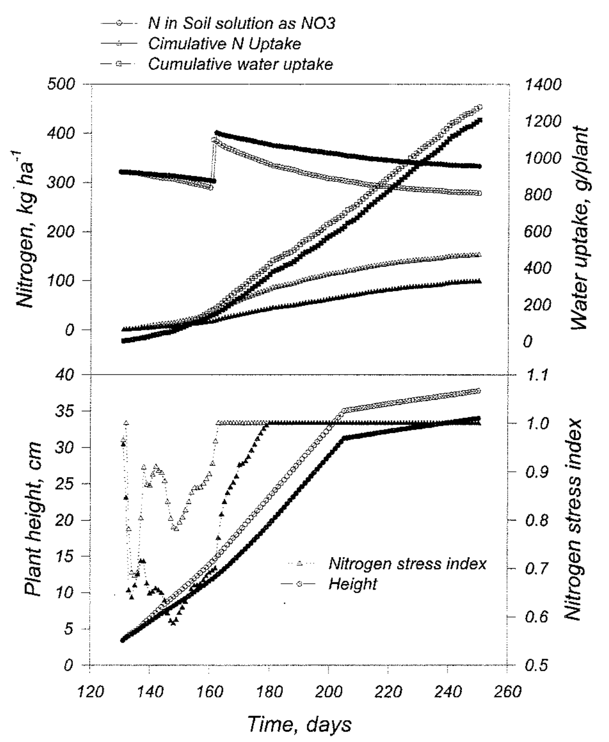

Figure 14.14 Plant uptake of water and nitrogen, nitrogen as NO3 in the soil, nitrogen stress and plant height for Example 14.4 for two methods of nitrogen application, banded and uniformly applied. The open symbols represent data for the banded application and the closed symbols uniform application.

because there was less nitrogen stress (Fig 14.14). This also contributed to the greater uptake of water and hence nitrogen in the banded treatment. The nitrogen stress early in the season is probably unrealistic and is due to the lack of N reserve in seed which usually carries a plant through it's early growth stages. The distribution of NO_3^- in the soil for both treatments was similar (Fig 14.15). The nitrate, however, was more dispersed in the right side of the profile in the banded treatment. This probably reflects distribution of nitrate under the influence of root activity alone. The zone of high nitrate concentration at the right of the profile is more distinct in the uniform application which reflects the addition of nitrogen at the surface.

The amount of nitrogen remaining in the soil profile at the end of the growing season was higher for the uniform application than for the banded application. This may result in greater leaching losses of N over the winter season from the profile to which the nitrogen was applied uniformly. The banded nitrogen was used more efficiently because it was placed close to where there was the largest density of roots.

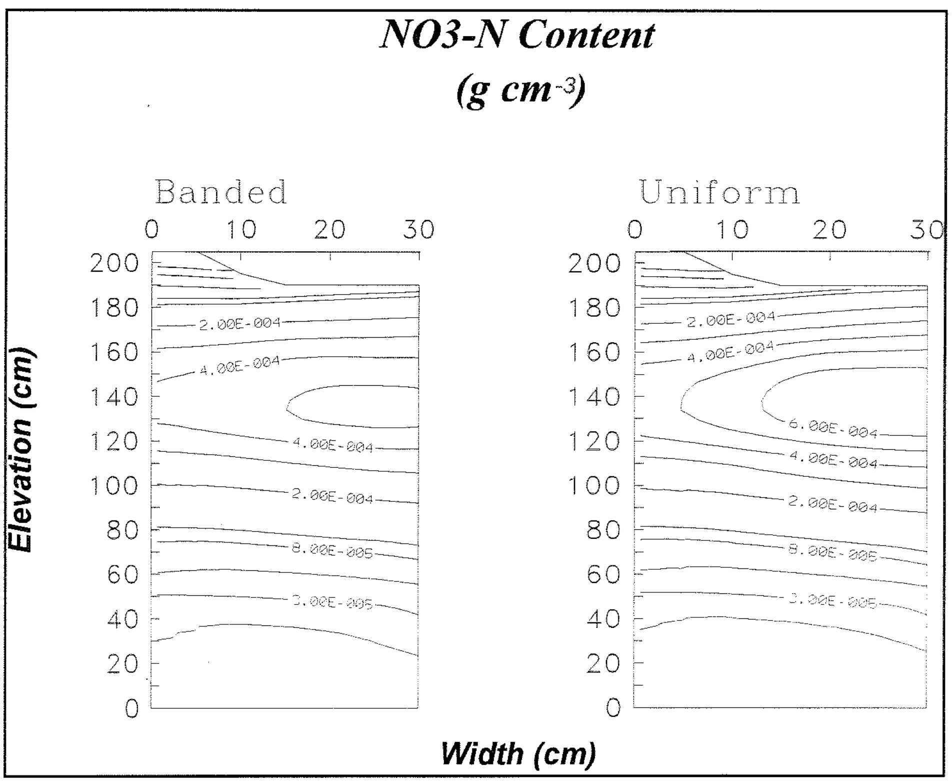

Figure 14.15 Nitrate distributions after last day of simulations for Example 14.4.

Chapter 15: Addition or Replacement of Soil Process Modules.

Dennis Timlin and Yakov Pachepsky

The rules listed below will guide the user to add or replace a process module in 2DSOIL. Illustrations of the application of these rules are also given in this section.

15.1 Rules for introducing a new module

The modules of 2DSOIL interact by sharing public variables that may be used or updated by any other process module. The list of public variables is given in Table 15.1 and are collected in the file **'public.ins'**. This is an 'INCLUDE' file that contains declarations and COMMON statement definitions for all the public variables. A reference to this file name must be at the beginning of the source code for every process module. The user must also be careful not to create any private variables that share the same name as a public variable. The placement of a module relative to other modules is also important because the sequence reflects the method by which transport processes and intrasoil interactions are decoupled during a time step. Transport calculations precede any calculations of intrasoil interactions because transport calculations require time steps and use values of sources and sinks from the previous, or 'old', time level. Intrasoil and surface interactions may use soil state variable values both from the current, or 'new' time step and/or from the previous or 'old' time step. Among transport modules the **WaterMover** must be called first because it produces information, such as water velocities, that other transport modules require.

Rules for introducing a new module are listed below.

1. A new module can use soil state variables that are public and are listed in Table 15.1. Other soil state variables not needed by other modules must only be present in the list of private variables for the new module.

2. There are two special cases for handling variables, these are:

   a. Variables that are used by several modules but not modified by any modules and not calculated using specific parameters. They can be either

      i. GLOBAL, and input or initialized by a particular module or

      ii. LOCAL, and read by each module that uses that variable.

   For example, bulk density is used by more than one module, e.g., the chemistry and the root growth modules. If none of the process modules change bulk density, then each module reads bulk density from it's own data files. Therefore, a new module will use this variable as private if this module does not change bulk density.

   b. Variables that are used by several modules and modified during calculations or calculated using parameters. The following apply for this case:

      i. The variables are GLOBAL.

      ii. The variables should be modified or initialized by only one module though, in practice, this may not always be practical.

      iii. The variable or the parameters used to calculate it must be read in by the module that does the modifications.

      iv. This module that reads a variable or its parameters must be called to initialize the variable before other modules that use the variable are called.

      v. The variable will be stored in a COMMON block in a section with it's own name.

   An example is bulk density. Bulk density can be changed by tillage but is also used in root growth or chemistry routines. Because the tillage module changes bulk density, this module should read it in and be called before the root growth or chemistry modules.

3. A replacement module must update all public variables that were updated by the module being replaced. For example, if a new water mover that simulates steady state water movement and a stationary velocity field is introduced, this water mover must still

produce actual values of boundary fluxes. Table 15.2 shows which variables are updated by each module.

4. Calculation of sinks or modification of soil state variables must be done for the whole grid, if the region of changes is not specified. For example, if a soil denitrification module is introduced, it must update nitrate concentrations in all grid nodes. However, if the module simulates a nitrogen fertilizer application to the soil surface, then nitrate concentrations must be updated only in the designated nodes.
5. If a new module has special time step requirements, it must update certain variables in the '*time_public*' common field (see Table 15.1). If a new module simulates transport processes, it must update the corresponding *DtMx* value at the beginning of simulation and after every time it updates the soil state variables. If a new module has its own time step or it will act only at prescribed times, then it must update the corresponding *tNext* value at the beginning of the simulation, and after every time it is called and updates soil state variables.
6. It is advisable to use Fig. 1.3. as a template for a new module.
7. If a new module uses or introduces plant parameters, it is the responsibility of the user to add plant variables to the '**puplant.ins**' file to be linked with plant and root modules.

Three simple examples that illustrate the addition of a module are given here. These new modules are: 1) a management module to add fertilizer, 2) a root respiration module, and 3) a solute uptake module.

15.1 <u>Overview of the examples</u>

15.1.1 Example 15.1: A management module for chemical application

This example is a simple management module, called **Mngm**, that simulates surface chemical application with instantaneous dissolution at a given time within the growing season. At time *tAppl* the mass of the chemical in (g/L) near the soil surface in nodes *nAppl(1),*

```
            *|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||*
        *
Line #  *  Chemical application module - example 1 from section 15.1  *          *
            *|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||*
        1      Subroutine Mngm()
        2      Include 'public.ins'
        3      Common /Mngmnt/ tAppl, cAppl,NumAp,nAppl(NumBPD),ModNum
        4      If (lInput.eq.1) then
        5        Open(40,file='Mngm.dat')
        6        Read(40,*,ERR=20) tAppl, cAppl,NumAp
        7        Read(40,*,ERR=20) (nAppl(I),I=1,NumAp)
        8        NumMod=NumMod+1
        9         ModNum=NumMod
        10       Close(40)
        11       tNext(ModNum)=tAppl
        12     Endif
        13     If(Abs(Time-tNext(ModNum.lt.0.001*Step) then
        14       Do I=1,NumAp
        15       Conc(nAppl(I),3)=cAppl/ThNew(nAppl(I))
        16       Enddo
        17       tNext(ModNum)=1.E+32
        18     Endif
        19     Return
        20  20   Stop 'Mngm data error'
        21     End
```

Figure 15.1 Listing of FORTRAN code for a chemical application module.

nAppl(2),..., nAppl(NumAp) is set equal to *cAppl*. The FORTRAN code of the module is shown in Fig. 15.1.

- Line 1 names the module. The module must be invoked in the main program. Since the module must make it's calculations after the end of a time step **Mngm** must be called after all soil process subroutines.
- Line 2 makes all public information available to this module.
- Line 3 defines the common field *Mngmnt* as a storage of private information and lists the private variables: *tAppl* -application time, *cAppl* -applied concentration, *NumAp* - total number of nodes to which fertilizer is applied; and *nAppl* -the number of application nodes.
- Lines 4 - 11 describe actions of the module during the initialization stage at the beginning of the simulation run. Private time-independent information is read from the external unit

40, which is then closed because this unit number may be used by other modules. Any errors during file read statements will lead to termination of the run, as shown in line 18. The module is given a sequential number, *ModNum*, as an identifier and the variable that hold the number of module, *ModNum*, is updated. Because on of the time steps must end exactly at *tAppl*, *tNext(ModNum)* is set to *tAppl.*

- Lines 13 - 18 describe the actions of the module during the simulation run. If simulated time value *Time* coincides with the application time *(tAppl)*, then the concentration of the first solute is altered in given nodes. Because the module will not be invoked again, *tNext(ModNum)* is given an unreachable value.

The use of this module is illustrated in Section 15.4.1

15.1.2 Example 15.2: Simulation of root respiration

Root respiration is a source of CO_2 in soil. This module, called **GasUptake**, will modify the concentration of gas # 1 in the soil atmosphere. The rate of CO_2 production is assumed to be proportional to the increase in root mass in the soil:

$$G_g = \alpha_g \frac{d\rho_R}{dt} \qquad \textbf{(15.1)}$$

where G_g is the rate of CO_2 supply due to root respiration, g cm^{-3} day^{-1}; ρ_R is root mass density; g cm^{-3}; t is time, day; and α_g is respiration rate constant, day^{-1}. Derivation of the model equation and values of α_g may be found in the work of Penning de Fries (1975). For a finite difference approximation of equation 15.1 using discrete time steps we replace the derivative in the right side by the difference:

$$G_g = \alpha_g \frac{m_R^{new} - m_R^{old}}{A \Delta t} \qquad \textbf{(15.2)}$$

Here m_R^{new} is root mass in the soil cell at the new time level, m_R^{old} is root mass in the soil cell at the old time level, A is area of the cell, and Δt is time step.

The FORTRAN code of the module is shown in Fig. 15.2.

```
              *||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||*
        *
Line #  *   Root respiration module - example 2 from section 15.1    *          *
              *||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||*
1                 Subroutine GasUptake()
2                 Include 'public.ins'
3                 Common /GasCom/ alphaR, RTWTOLD(NumElD),ModNum
4                 If (lInput.eq.1) then
5                   Open(40,file='GasUpt.dat')
6                   Read(40,*,ERR=20) alphaR
7                   Close(40)
8                    Do I=1,NumEl
9                       RTWTOLD(I)=RTWT(I)
10                   Enddo
11                   NumMod=NumMod+1
12                   ModNum=NumMod
13                   TNext(ModNum)=1.0E+32
14                Endif
15                Do I=1,NumEl
16                  gSink(i,1)=alphaR*(RTWT(I)-RTWTOLD(I))/Area(I)/Step
17                  RTWTOLD(I)=RTWT(I)
18                Enddo
19                Return
20           20   Stop 'GasUpt data error'
21                End
```

Figure 15.2 Listing of FORTRAN code for a root respiration module

- Line 1 names the module which must be invoked in the main program. **GasUptake** is called after all soil transport process modules because **GasUptake** has a time step, and so must work during every time step.
- Line 2 makes all public information available to this module.
- Line 3 defines the common field *GasCom* as storage of private information and presents the list of private variables: *alphag* and *RTWTOLD*. The latter variable represents root mass in soil cells at the previous or 'old' time level. This value is needed because *RTWT* has a new value each time *GasUptake* is called and the old value is not available in the public data block. Because the *GasUptake* module needs the 'old' values of *RTWT*, it stores it as its own private information.
- Lines 4 - 13 describe actions of the module in the beginning of the simulation run. Private time-independent information is read and the external unit 40 is closed. The module is

given a sequential number, *ModNum*, as an identifier and the variable that hold the number of module, *ModNum*, is updated. This module will not specify limitations on the time steps, and therefore *tNext(ModNum)* is set to an unreachable value of 1.E+32.

- Lines 15 - 18 describe the actions of the module during a simulation run. Because the module is executed at every time step, there is no check of coincidence between simulated time, *Time*, and some specific time as in the previous example. In all nodes, *gSink* receives a new value. The 'new' RTWT will be the 'old' one at the next time step and the necessary assignment is made.

The use of this module is illustrated in Section 15.3.2

15.2 Solute uptake module

This module calculates passive solute uptake by roots as they take up water. The total solute uptake is calculated from the concentration of the solute in the water taken up by the roots and the amount of water. This module was used in example 4 of Chapter 14, the example using nitrogen.

```
*******         Solute Uptake subroutine from Example 14-4************
         subroutine SoluteUptake()
        include 'public.ins'
        IncLude 'Puplant.ins'
        Include 'Puweath.ins'
        real bi(3),ci(3)
        Common /SUP/ TotWSink,TotSSink
        If(lInput.eq.1) then
                TotWSink=0.
                TotSSINK=0.
        Endif
        SincrSink=0.
        Wincrsink=0.
C calculate sum of N in all forms in soil using triangluar elements
        Do n=1,NumEl
                cSink(n,1)=0.
                        NUS=4
                        if(KX(n,3).eq.KX(n,4)) NUS=3
*        Loop on subelements
                AE=0.0
                        do k=1,NUS-2
                                i=KX(n,1)
                                j=KX(n,k+1)
                                l=KX(n,k+2)
                                Ci(1)=x(l)-x(j)
                                Ci(2)=x(i)-x(l)
                                Ci(3)=x(j)-x(i)
                                Bi(1)=y(j)-y(l)
                                Bi(2)=y(l)-y(i)
                                Bi(3)=y(i)-y(j)
                                AE=AE+(Ci(3)*Bi(2)-Ci(2)*Bi(3))/2.
                                Csink(n,1)=CSink(n,1)+(Conc(i,1)+
     &                  Conc(j,1)+Conc(l,1))/3.
                        Enddo
                Csink(n,1)=CSink(n,1)*Sink(n)
                SincrSink=SIncrSink+cSink(n,1)*step*14./62.*AE
                Enddo
        enddo
        TotWSink=TotWSink+WIncrSink
        TotSSINK=TotSSINK+SIncrSink
        return
        end
```

Figure 15.3 Listing of FORTRAN code for the solute uptake module

Table 15.1. Public soil state variables used for the exchange of information among modules.

Variable	Description	Reference chapter
	Parameters	
NumNPD	Maximum number of nodal points	3
NumElD	Maximum number of elements (soil cells)	3
NumBPD	Maximum number of boundary points	3,5
NSeepD	Maximum number of seepage surfaces	3,6
NumSPD	Maximum number of nodal points on the seepage surfaces	3,6
NumSD	Maximum number of solutes	7
NumGD	Maximum number of gases	9
NMatD	Maximum number of different soil layers or horizons	3
NumObjD	Maximum number of soil process modules	2
MBandD	Maximum allowed difference between numbers of corner nodes for any two elements having at least one common node	6,7,8,9
	common /object_public/	
NumObj	Number of modules describing soil processes	2
CObj	Switch to show if the module is included in the present simulation run	2
KAT	Switch to show if axisymmetrical or planar movement is to be simulated	2,6,7,8
NumSol	Actual number of solutes	7
NumG	Actual number of gases	9
	common /grid_public/	
NumNP	Actual total number of nodal points	3
NumEl	Actual total number of elements (soil cells)	3
IJ	Maximum number of nodes along the transverse grid lines	3
MBand	Actual maximum difference between numbers of corner nodes for any two elements having at least one common node	3
x	Transverse coordinates of nodal points	3
y	Vertical coordinates of nodal points	3
KX	Numbers of corner nodes for every element	3

Variable	Description	Reference chapter
Area	Areas of elements (soil cells)	3,6
	common /nodal_public/	
hNew	Nodal pressure heads at the end of current time step	6
ThNew	Nodal moisture contents at the end of current time step	6
Vx	Nodal transverse water velocities at the end	
	of current time step	6
Vz	Same as above for vertical water velocities	6
Q	Nodal water fluxes at the end of current time step	6
Conc	Nodal concentrations of solutes in soil solution at the end of current time step	7,11
Tmpr	Nodal temperature values at the end of current time step	8
g	Nodal gas contents in soil air at the end of current time step	9
MatNumN	Number of soil layer or horizon in which node occur	3
Con	Nodal soil hydraulic conductivity	6
Tcon	Nodal soil thermal conductivity	8
	common /elem_public/	
Sink	Water extraction rates for elements	6,10
cSink	Solute extraction rates for elements	7
gSink	Gas extraction rates for elements	9
MatNumE	Number of soil layer or horizon in which element occurs	3
RTWT	Root mass in the elements (soil cells)	10
	common /bound_public/	
NumBP	Actual total number of boundary points	3,4,5
NSurf	Total number of nodes on the soil-atmosphere surface	5
NVarBW	Total number of boundary nodes where time-dependent water fluxes or pressure heads are prescribed	3,4,6
NVarBS	Total number of boundary nodes where time-dependent solute concentrations are prescribed	3,7
NVarBT	Total number of boundary nodes where time-dependent temperatures are prescribed	3,8
NVarBG	Total number of boundary nodes where time-dependent gas contents are prescribed	3,9

Variable	Description	Reference chapter
NSeep	Actual total number of seepage faces	3
NSP	Total number of nodes at every seepage face	3
NP	Nodal numbers of boundary points at seepage faces	3
KXB	List of boundary node numbers	3
Width	Width of strips associated with boundary nodes	3
CodeW	Codes of boundary condition for water movement	3,4
CodeS	Same as above for the solute movement	3,4
CodeT	Same as above for the heat movement	3,4
CodeG	Same as above for the gas movement	3,4
VarBW	Boundary pressure heads or water fluxes	3,4,5
VarBS	Boundary concentrations or fluxes of solutes	3,4,5
VarBT	Boundary temperatures or heat flux components	3,4,5
VarBG	Boundary gas contents or gas flux components	3,4,5
EO	Potential transpiration rate from the unit of crop area	5,10
Tpot	Potential transpiration from the unit soil area	5,10
	common /time_public/	
lInput	Switch to show if initial data have to be read	2
tNext	Obligatory time step end times	2
Tinit	Time of the beginning of calculations	2
Time	Current value of the time step end time	2
Step	Current time step	2
tAtm	Time of the next soil-atmosphere boundary update	5
Iter	Maximum number of iterations in soil transport modules	2
dtMin	Minimum reasonable time step	2
DtMx	Maximum next time steps allowed by soil transport process modules	2,6,7,8,9
tTDB	Time of the next update of the time-dependent boundary conditions	4
Tfin	Time of the end of calculations	2

Table 15.02. Alteration (*) of public variables by modules of 2DSOIL

Variable	obj_t ime	grid bnd	syn-cron	set srf1	set srf2	set TDB	wat mov	sol mov	heat mov	gas mov	wat upt1	wat upt2	macro chem
Area	-	*	-	-	-	-	-	-	-	-	-	-	-
CObj	*	-	-	-	-	-	-	-	-	-	-	-	-
CodeG	-	*	-	-	-	-	-	-	-	-	-	-	-
CodeS	-	*	-	-	-	-	-	-	-	-	-	-	-
CodeT	-	*	-	-	-	-	-	-	-	-	-	-	-
CodeW	-	*	-	*	-	-	-	-	-	-	-	-	-
Con	-	-	-	-	-	-	*	-	-	-	-	-	*
Conc	-	-	-	-	-	-	-	*	-	-	-	-	-
cSink	-	-	-	-	-	-	-	-	-	-	-	-	-
dtMin	*	-	-	-	-	-	-	-	-	-	-	-	-
DtMx	-	-	-	-	-	-	*	*	*	*	-	-	-
EO	-	-	-	-	-	-	-	-	-	-	*	-	-
g	-	-	-	-	-	-	-	-	-	*	-	-	-
gSink	-	-	-	-	-	-	-	-	-	-	-	-	-
hNew	-	-	-	-	-	-	*	-	-	-	-	-	-
IJ	-	*	-	-	-	-	-	-	-	-	-	-	-
Iter	-	-	-	-	-	-	*	-	-	-	-	-	-
KAT	-	*	-	-	-	-	-	-	-	-	-	-	-
KX	-	*	-	-	-	-	-	-	-	-	-	-	-
KXB	-	*	-	-	-	-	-	-	-	-	-	-	-
IInput	-	-	*	-	-	-	-	-	-	-	-	-	-
MatNumE	-	*	-	-	-	-	-	-	-	-	-	-	-
MatNumN	-	*	-	-	-	-	-	-	-	-	-	-	-
MBand	-	*	-	-	-	-	-	-	-	-	-	-	-
NP	-	*	-	-	-	-	-	-	-	-	-	-	-
NSeep	-	*	-	-	-	-	-	-	-	-	-	-	-
NSP	-	*	-	-	-	-	-	-	-	-	-	-	-
NSurf	-	*	-	-	-	-	-	-	-	-	-	-	-
NumBP	-	*	-	-	-	-	-	-	-	-	-	-	-
NumEl	-	*	-	-	-	-	-	-	-	-	-	-	-
NumG	-	-	-	-	-	-	-	-	-	*	-	-	-

Table 15.02. Alteration (*) of public variables by modules of 2DSOIL

Variable	obj_time	grid bnd	syn-cron	set srf1	set srf2	set TDB	wat mov	sol mov	heat mov	gas mov	wat upt1	wat upt2	macro chem
NumNP	-	*	-	-	-	-	-	-	-	-	-	-	-
NumObj	*	-	-	-	-	-	-	-	-	-	-	-	-
NumSol	-	-	-	-	-	-	-	*	-	-	-	-	-
NVarBG	-	*	-	-	-	-	-	-	-	-	-	-	-
NVarBS	-	*	-	-	-	-	-	-	-	-	-	-	-
NVarBT	-	*	-	-	-	-	-	-	-	-	-	-	-
NVarBW	-	*	-	-	-	-	-	-	-	-	-	-	-
Q	-	-	-	-	-	-	*	-	-	-	-	-	-
RTWT	-	-	-	-	-	-	-	-	-	-	*	-	-
Sink	-	-	-	-	-	-	-	-	-	-	*	*	-
Step	-	-	*	-	-	-	*	-	-	-	-	-	-
tAtm	-	-	-	*	*	-	-	-	-	-	-	-	-
TCon	-	-	-	-	-	-	-	-	*	-	-	-	-
tFin	*	-	-	-	-	-	-	-	-	-	-	-	-
ThNew	-	-	-	-	-	-	*	-	-	-	-	-	-
Time	-	-	*	-	-	-	*	-	-	-	-	-	-
tIinit	*	-	-	-	-	-	-	-	-	-	-	-	-
Tmpr	-	-	-	-	-	-	-	-	*	-	-	-	-
tNext	-	-	-	-	-	-	-	-	-	-	*	*	-
Tpot	-	-	-	*	*	-	-	-	-	-	-	-	-
tTDB	-	-	-	-	-	*	-	-	-	-	-	-	-
VarBG	-	-	-	*	*	*	-	-	-	-	-	-	-
VarBS	-	-	-	*	*	*	-	-	-	-	-	-	-
VarBT	-	-	-	*	*	*	-	-	-	-	-	-	-
VarBW	-	-	-	*	*	*	-	-	-	-	-	-	-
Vx	-	-	-	-	-	-	*	-	-	-	-	-	-
Vz	-	-	-	-	-	-	*	-	-	-	-	-	-
Width	-	*	-	-	-	-	-	-	-	-	-	-	-
x	-	*	-	-	-	-	-	-	-	-	-	-	-
y	-	*	-	-	-	-	-	-	-	-	-	-	-

15.3 The examples: Specific applications and simulation results

15.3.1 Example 15.1. A chemical application module: influence of the root system on chemical transport

This example demonstrates the addition of a new module to carry out a management practice, in this case application of a chemical. The example is a simulation of solute movement in row and interrow zones of a row crop. The roots of a crop planted in rows will dry the soil under the rows more rapidly than between the rows by nature of the uneven root distribution. Because the soil is more wet between the rows than in the row zone, the wetting front during infiltration will penetrate deeper in the interrow zone than in the row zone. As a result, there will also be higher solute fluxes in the interrow zone than in the row zone. The modules used for this simulation are **Mngm**, the management module that applies the chemical at a specified time as described in Section 15.1, **WaterMover**, **SoluteMover**, **RootWaterUptake01**, and **SetSurf01**.

The following are the specific steps taken to incorporate the management module **Mngm** into 2DSOIL.

Changes in program loading. A line to invoke **Mngm** is inserted into the main program (2dmain.For), then the file 'Mngm.for' (Fig. 15.1) is compiled and linked with the other modules.

Changes in data files.

(1) A new file 'Mngm.dat' that provides input data for Mngm ('Mngm.for') is created.

Description of the physical system and parameters. Figure 15.3 is a graphical representation of the finite element grid. A simple rectangular grid with a flat surface boundary was used. Nodes and elements are numbered downward in the z direction with

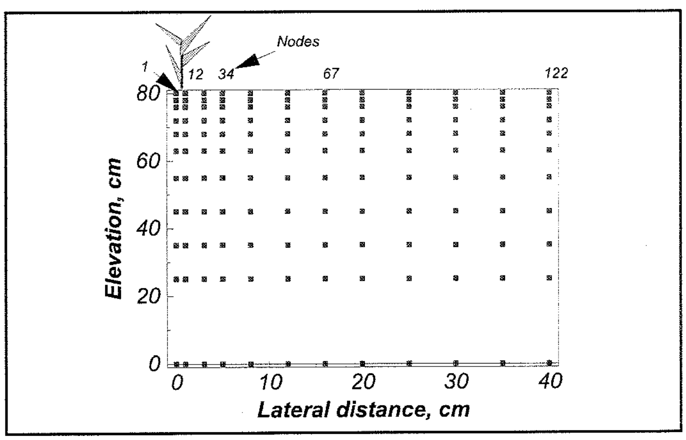

Figure 15.4 Finite element grid for Example 15.2, nodes and elements are numbered in vertical direction

distance between nodes increasing with depth and with lateral distance. The soil consisted of one layer and the soil texture was a loam. The soil hydraulic functions are in Fig. 15.4. The atmospheric boundary data, precipitation and evapotranspiration, were input directly from **SetSurf.Dat**. Heat and gas movement were not simulated. The initial chemical concentration was zero throughout the profile, and there were no chemical gradients across the left or right boundaries. The bottom boundary was impermeable to both chemical and water. The plant was placed at the left end of the grid at x=0 cm, and 5 cm deep (Fig 15.3). Soybean root parameters were used. Simulated time began at day 1 and ran a total of 57 days.

The chemical was a non-reactive tracer similar in properties to bromide. Chemical was applied at day 44 at a rate of 300 $\mu g\ cm^{-2}$. During the preceding 44 days, only 11 cm of rain-fall was applied and evapotranspiration was 29 cm. This simulates a moderately dry period. Immediately after chemical application 3 cm of water was applied. The details of the surface boundary data can be found in the file 'SetSurf.Dat'.

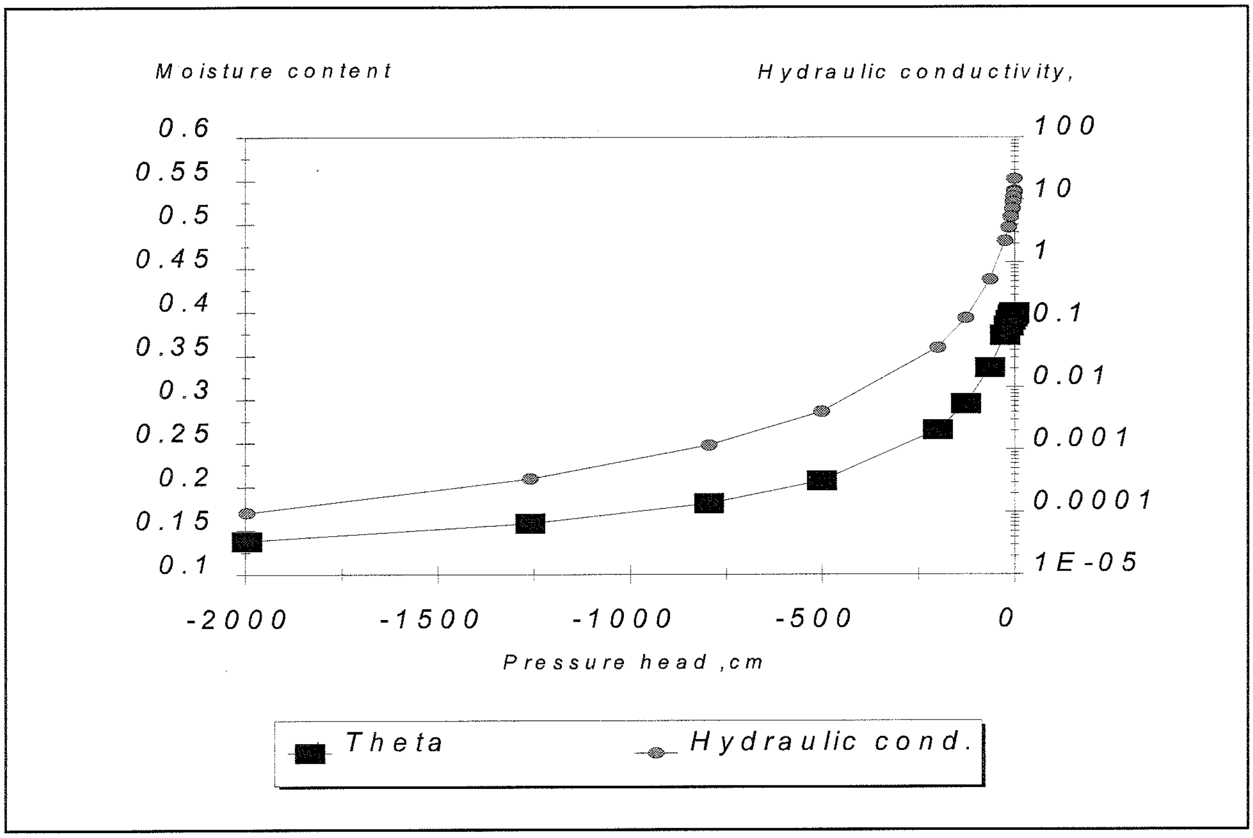

Figure 15.5 Soil hydraulic functions for Example 15.1

<u>Results of simulations.</u> The water content and root distribution before chemical application and water distribution after rainfall are presented in Fig. 15.5. The root distribution is concentrated in the soil under the plant. It can be seen that values of water content are lowest under the plant where the root distribution is high. Water content increases with depth and horizontal distance from the plant. After rainfall, the water contents are still highest in the interrow zone (Fig. 15.5). The chemical concentration directly below the plant row (at x=1) and under the interrow position (x=35) at day 56 are plotted in Fig. 15.6. There is more chemical in the soil under the row position than the interrow position. Furthermore, the chemical concentrations near the surface are still rather high in spite of 2 rain-fall events after chemical application. Evaporation and transpiration act to move the water with chemical toward the soil

surface and toward the plant. This results in concentration of chemical in the soil under the plant and depletion of chemical in the interrow zone.

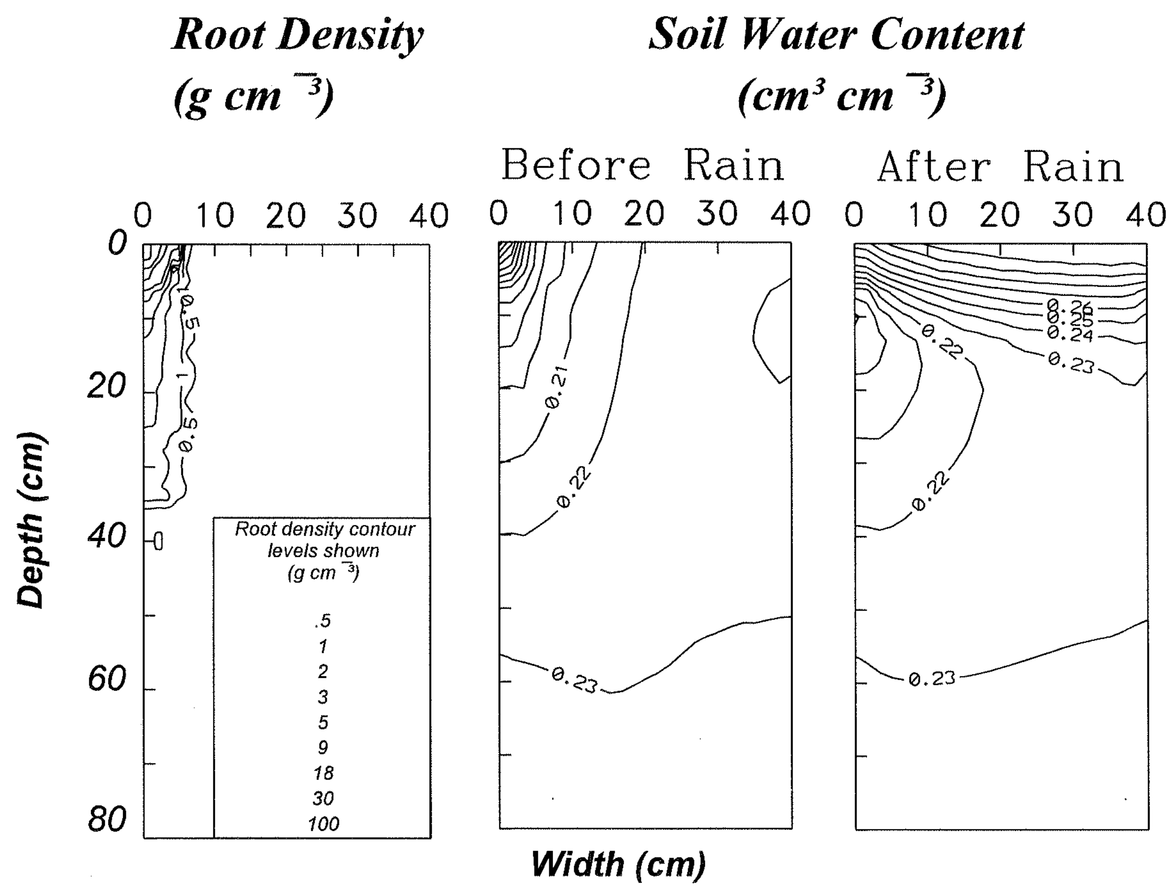

Figure 15.6 Root density and water content at day 44 before chemical application and rainfall, and water content after chemical application and rainfall, day 44.8.

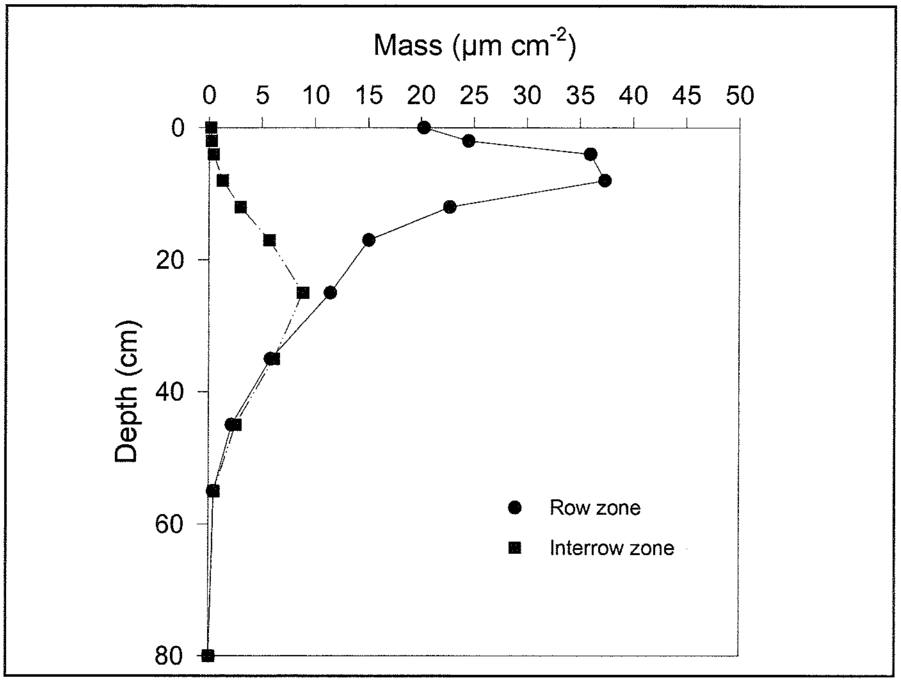

Figure 15.7 Chemical concentration *vs* soil depth in the row zone (x=1 cm) and in the interrow zone (x=35 cm)

It has been suggested that placement of chemicals in areas not conducive to leaching would be a possible management practice to reduce leaching of chemicals out of the root zone. This model with a comprehensive plant simulator can be used to evaluate the relative merits of such management practices. An interesting extension to this simulation would be to consider the fluxes of water as precipitation to be nonhomogeneous along the surface.

15.3.2 Example 15.2: Adding a module of root respiration: a two-dimensional pattern of CO_2 content in soil induced by root respiration

In this section we take the example in Section 14.1 and add the **GasUptake** module described in Section 15.1.

Changes in program loading. A line is added to the main file '**2DSOIL.FOR**' to call **GasUptake** and the file '**GasUpt.for**' is compiled and linked to other modules.

Changes in data files.

(1) Data file '**Grid_bnd.dat**' is modified: codes of gas movement boundaries are included.

(2) Data file '**Weather.Dat**' is modified to include soil-atmosphere gas exchange parameters: parameters *PG* and *GAIR* are added in line 11 according to Table 5.5.

(3) Data file '**GasUpt.dat**' is added.

All these changes can be traced by comparing files in directories EXE13.1R and EXE15.2 on the distribution diskette.

<u>Results of gas movement simulations</u>. Gas content distributions are shown in Fig. 15.6 for 30 and 60 days after emergence. There is little difference in transversal distributions of gas contents. Diffusion coefficients are very high, and diffusion has enough time to redistribute the

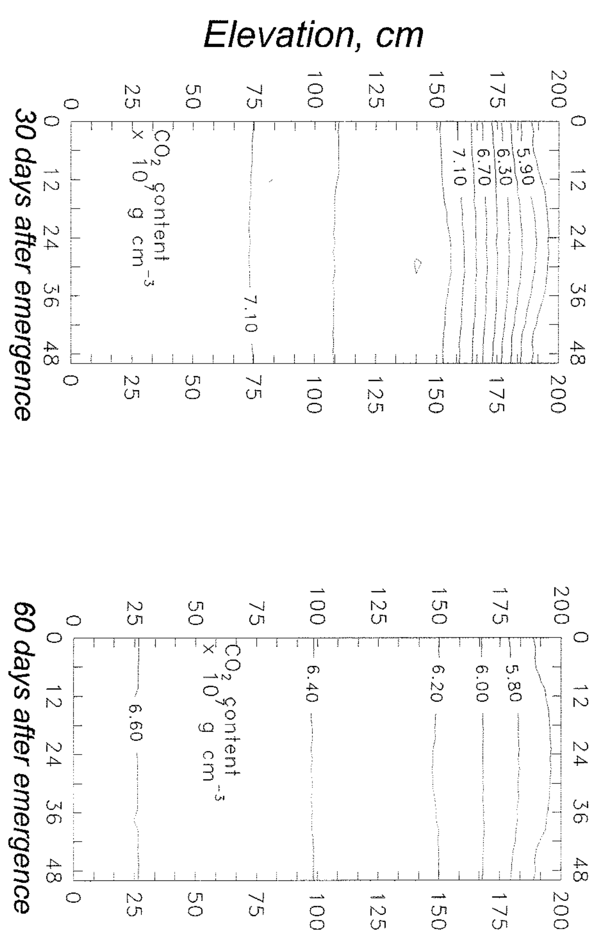

Figure 15.8 Calculated CO_2 contents in soil air due to root respiration, X 10^7, *g* cm^{-3}. (a) 30 days after emergence, (b) 60 days after emergence.

CO_2 produced by roots in spite of the common perception of diffusion as a slow process. At early stages of the simulation, gas contents tend to be higher in the row zone than in interrow position. This may be due to the high intensity of early root growth simulated by the shoot imitator. Diffusion moves gas downwards and simultaneously sends it to the atmosphere. As a result, the maximum concentration of CO_2 is found at approximately 80 cm depth. At later stages of the simulation the roots produce less CO_2 and gas exchange between the soil and atmosphere is able to remove almost all excess gas from the soil. CO_2 transport even occurs from deep horizons of the soil.

In general, the calculated gas contents are very low and do not exceed gas contents in the atmosphere by more than 25 %. In a field soil one can readily obtain gas content values that are 10-30 times larger than contents in the atmosphere. This suggests microbial activity must be taken into account for realistic predictions of CO_2 dynamics in soil. One should also consider the more complex tortuosity patterns in field soil as opposed to repacked soils.