Chapter 3: Grid and Boundary Settings:

Get_Grid_and_Boundary Module

Yakov Pachepsky and Dennis Timlin

Values of the soil and root state variables are recorded and calculated for specific locations within a soil profile, and these locations have to be the same for all modules. The spatial reference system in 2DSOIL is represented by a spatial grid (Fig. 3.1) which has a polygonal geometric structure representing a vertical plane cross-section of the soil. The grid illustrated in Fig. 3.1 is for a soil system where the soil surface has the shape of a ridge and furrow. The spatial locations where values of the state variables and fluxes are known are called nodes. The nodes provide the necessary spatial reference for interactions among soil and root processes, and for the boundary interface with plant and atmosphere models. The nodal coordinates, therefore, have to be available to all modules. The nodal coordinate system is also needed for the numerical solution of transport equations that describe the movement of water, solutes, heat, and gases in the soil. The nodal grid is set by the control module **Get _Grid_and_Boundary**.

3.1 <u>Grid setting</u>

The 2DSOIL code does not generate the grid, it simply reads data defining the grid from the input file **'Grid_bnd.dat'**. Since the program is very much oriented to the use of the finite element method for numerical estimation of intrasoil transport, data for nodes (grid nodes) and elements (grid cells) are both required. If the user replaces a soil transport process module with another that uses the finite difference numerical technique, then the information on elements is not used.

The coordinate system of 2DSOIL uses a vertical coordinate that is zero at the base of the profile and increases upwards. Both radial and planar geometry can be considered. The grid may consist of quadrilateral and triangular elements. The only definite requirement is the continuity and non-intersection of horizontal grid lines (Neuman, 1972; Šimůnek et al., 1992). For practical purposes this means that there is a layer of quadrilateral elements between any pair of horizontal lines. Triangular elements may exist at the boundaries. An example of a simple grid with triangular and quadrilateral elements is shown in Fig. 3.1.

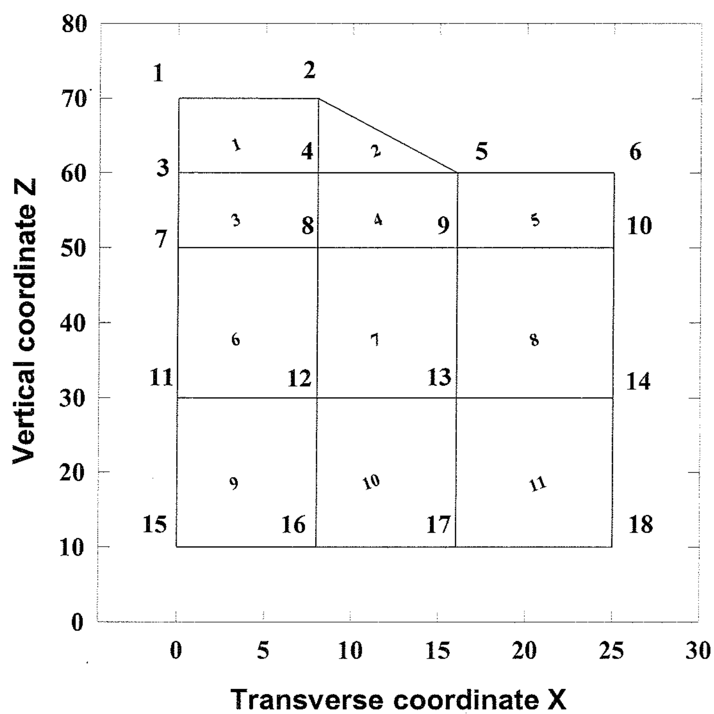

Figure 3.1 An example of the grid for a soil domain with numbered nodes and elements (soil cells).

It is preferable from a computational point of view, but not essential, to number nodes along the direction (vertical or horizontal) in which the number of nodes is the least. If, for example, the grid has a maximum of 25 nodes horizontally and a maximum of 10 nodes vertically, then the numbering should be along the vertical lines. Node number 1 must be in the upper left corner of the grid. In general, element numbering is in the same direction as node numbering as illustrated in Fig. 3.1. There are no definite recommendations for selecting sizes and total number of elements although it is desirable to have neighboring elements with dimensions that differ by no more than about 1.5 times. It is also important to have the smallest elements in the zones of high gradients (Šimůnek et al., 1992). In general, a 200-node grid should be able to give a reasonably good description of the transport processes in soil.

The two-dimensional soil domain may contain several subdomains containing soils with different properties. The subdomains typically represent soil genetic horizons, but may also represent soil plow layers and plow-pans, and different soil materials in row and interrow positions, among other features. Subdomains must be numbered and the number of each subdomain will be referenced for every node and element that lies in the subdomain. Coordinates and a soil subdomain number are given for every node. The subdomain is referred to by a material number that is stored in the public variables *MatNumN(I)*, with *i* varying from one to *NumNP*, the total number of nodes. Elements of the grid each have an identifying number, and four corner nodes *KX(e,1), KX(e,2), KX(e,3), KX(e,4)* associated with them. The corner nodes are numbered counter-clockwise. The third and fourth node number are the same for triangular elements. In addition to corner node numbers, the elements also have soil subdomain material numbers *MatNumE(e), e=1,2,...., NumEl*; where *NumEl* is the total number of elements.

3.2 Codes for boundary nodes

A code must be associated with every boundary node where matter or energy enters or leaves the system. The codes used in 2DSOIL are given in Table 3.1. The source numbering system is used in several older finite element models (Šimůnek et al., 1992).

In general, 2DSOIL uses three types of boundary conditions for water, solute, heat, and gas movement:

(1) First-type boundary conditions, where the state variable is constant, i.e., the pressure head, solute concentration, temperature, or gas content is constant at the boundary node during the entire simulation time, or the state variable has constant values during several time intervals;

(2) Second-type boundary conditions where the flux of water, solute, heat, or gas is prescribed during the simulation time, or where this flux has constant values during several time intervals;

(3) Third-type boundary conditions at the soil-atmosphere boundary where the flux is dependent also on the soil surface condition.

Nodes that are coded with numbers -4 and +4 are at the external boundaries of the soil domain. Nodal codes +6 and -6 are applied to intrasoil sources and sinks of energy or matter (for example, a drainage tile or hot water line). Water, solute, heat, and gas transport codes are stored in the public variables *CodeW(I), CodeS(I), CodeH(I)*, and *CodeG(I)*, respectively, where index *i* varies from one to *NumBP*, the total number of boundary nodes. Codes of boundaries for different transport processes do not have to coincide. For example, if leaching of solute from the soil is considered and the pressure head on the column surface is constant but the concentration of solute in the leaching water is time-dependent, then the nodes of this upper boundary will have code 1 for water movement and code 3 for solute movement. If some boundary nodes for water flow correspond to an impermeable layer, then these nodes are coded by zeroes for solute (heat or gas) boundaries. Thus, the total number of boundary nodes for water, solute, heat, and gas are always the same.

Table 3.1. Codes for boundary nodes.

code	Type of boundary			
	Water movement	Solute movement	Heat movement	Gas Movement
0	Impermeable surface	Impermeable surface	Impermeable surface	Impermeable surface
1	Constant pressure Constant moisture content	Constant solute concentration	Constant temperature	Constant gas content in soil air
2	Saturated seepage face	-	-	-
-2	Unsaturated seepage face	-	-	-
3	Time-dependent pressure head	Time-dependent concentration	Time-dependent temperature	Time-dependent gas content
-3	Time-dependent water flux	Time-dependent solute flux	Time-dependent heat flux	Time-dependent gas flux
4	Soil-atmosphere boundary with prescribed pressure head	Soil-atmosphere boundary with prescribed concentration	Soil atmosphere boundary with temperature prescribed	Soil atmosphere boundary with gas content prescribed
-4	Soil-atmosphere boundary with prescribed water flux	Soil-atmosphere boundary with prescribed solute flux	Soil atmosphere boundary with heat flux prescribed	Soil atmosphere boundary with gas flux prescribed
6	Time dependent pressure within soil domain	Time dependent concentration within the soil domain	Time-dependent temperature within soil domain	Time-dependent gas content within soil domain
-6	Time dependent water flux within soil domain	Time-dependent solute flux within soil domain	Time-dependent heat flux within soil domain	Time-dependent gas flux within soil domain

At the soil-atmosphere boundary, one must initially set *CodeW*=-4. The **WaterMover** module will later distinguish between the case of -4 for a period of evaporation and +4 for a period of precipitation. Note that irrigation should be coded as +3 or -3, i.e., time dependent pressure or water flux. Sprinkler irrigation, however, can also be set in the weather file or **SetSurf.dat** as rainfall.

The soil boundary can include seepage faces, i.e., segments where water can flow out of a saturated soil under hydrostatic pressure. Seepage faces are coded 2 if they are initially saturated and -2 if they are initially unsaturated. Presumably seepage takes place along the whole seepage face. Therefore, additional information is required to show which nodes belong to a particular seepage face. The variable *NSeep* gives the number of seepage faces and is an input parameter. If *NSeep=0*, then no further information on seepage faces is necessary. If $NSeep \neq 0$, then the number of nodes *NSP* and a list of nodes *NS* are to be given for every seepage face (Table 3.1).

Every boundary node must also have an associated width that is necessary for flux calculations. In the case of planar flow the variable *Width* has the unit of length, whereas for axisymmetric flow the unit in area. The strip associated with the boundary node is perpendicular to the direction of the prescribed flux. The method to determine a value for *Width* is illustrated in Fig. 3.2. In Fig. 3.2a boundary nodes N_1, N_2, and N_3 lay on the horizontal soil-atmosphere surface where fluxes are in the vertical plane. The length of segment AB is the width associated with node N_2, here $AB=½(x_2-x_1)+½(x_3-x_2)$. In Fig. 3.2b, boundary nodes N_4, N_5 and N_6 are located at the vertical boundary of the soil domain where the fluxes are in the horizontal plane. The length of segment CD is the width associated with node N_5, $CD=½(y_4-y_5)+½(y_5-y_6)$. In Fig. 3.2c, boundary nodes N_7, N_8, and N_9 are situated on the horizontal soil-atmosphere surface and flow is axisymmetrical. The area of the dashed annulus is the width associated with node N_8. The internal radius of this ring is equal to $½(x_8-y_7)$ and the external radius is equal to $½(x_9-x_8)$. In Fig 3.2d the nodes N_{10}, N_{11}, and $N_{12}$are positioned on the vertical boundary and flow is now axisymmetrical. The area of the dashed strip equals the width of the strip associated with node N_{11} and is equal to $½(y_{10}-y_{11})+½(y_{11}-y_{12})$.

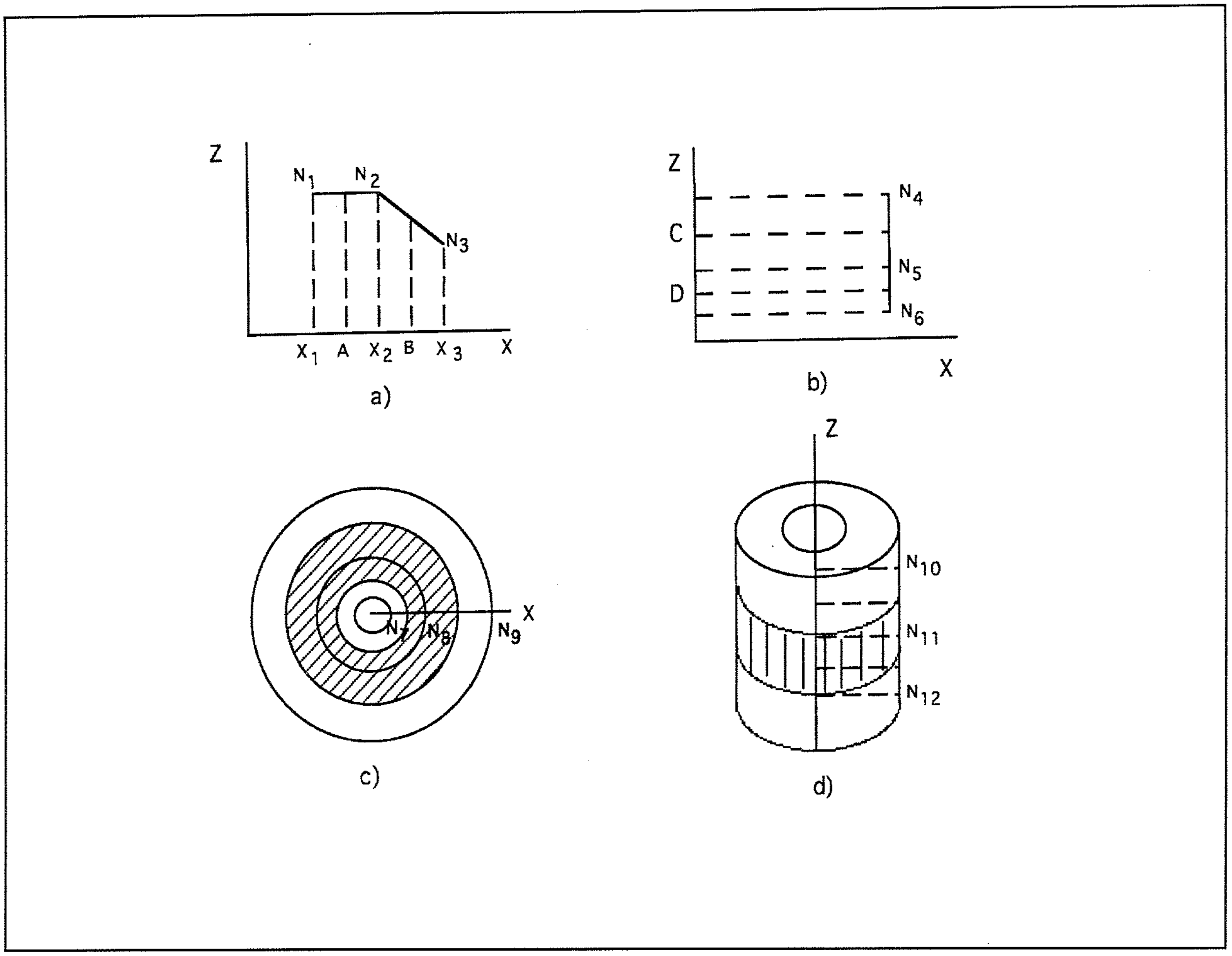

Figure 3.2 Derivation of the *Width* value: (a) transversal boundary, planar flow, (b) vertical boundary, planar flow, (c) transversal boundary, axisymmetrical flow, and (d) vertical boundary, axisymmetrical flow. Additional comments are in the text.

3.3 Grid Data file *Grid_bnd.dat*

A description of the file content is given in Table 3.2. This table is followed by Example 3.3, an illustration of a grid data file which has been constructed for the grid in Fig. 3.1 to show several features of the code. Other examples of grids can be found in Chapter 14 of this manual.

In the grid file given in Example 3, soil properties are considered to differ for the layers above and below the line z=50 cm. The example file listing illustrates the use of the values of *MatMumN* and *MatNumE*. Nodes 5 and 6 are at the bottom of a ditch. The level of water in the ditch is constant and therefore *CodeW* is 1 for these nodes. At the same time it is desired to change the concentration of the solute in the water over time, so that *CodeS* is 3 for these nodes. Heat transport is not considered; hence, the value of *CodeH* is set to zero. Since the bottom of the ditch is impermeable to gases, *CodeG* is zero there also. Nodes 1 and 2 are at the soil-atmosphere boundary; therefore, *CodeW* and *CodeS* have the value of -4. For purposes of this example, the gas fluxes will be determined as known time-dependent variables without any relation to atmospheric conditions. Thus, *CodeG* is equal to -3 and not to -4. Node 12 represents a tile drain where water is subject to positive hydrostatic pressure. The pressure head and concentration of the solute will be constant (*CodeW*=1 and *CodeS*=1). The pressure head is specified in the file used to initialize pressure heads (**Nodal_w.dat**) and described in a later chapter (6). Nodes 15-18 represent an initially unsaturated seepage face and, hence, the gas content in soil air is assumed to be constant there since air are in contact with the atmosphere outside the soil boundary. The value of *Width* at node 2 is equal to 8.0 because the fluxes to/from the atmosphere are parallel to the z direction.

Table 3.2. Format of the file **'Grid_bnd.dat'**.

Record	Variable	Description
1	- -	Comment line.
	<u>Dimensions</u>	
2	- -	Comment line.
3	*KAT*	Type of flow system to be analyzed 1 for axisymmetrical flow 2 for vertical flow in a cross-section.
3	*NumNP*	Number of nodes.
3	*NumEl*	Number of elements (quadrilateral and/or triangular).
3	*NumBP*	Number of boundary nodes for which boundary conditions are prescribed.
3	*IJ*	Maximum number of nodes along a transverse grid line (this is number of nodes along the horizontal or vertical transverse line that has the largest number of nodal points).
3	*NMat*	Total number of soil subdomains in the soil domain.
	<u>Nodal Information</u>	
4	- -	Comment line.
5	*n*	Nodal number.
5	*x(n)*	*x* - coordinate of node *n* [units=L] (always a horizontal coordinate).
5	*z(n)*	*z* - coordinate of node *n* [units=L] (always a vertical coordinate).
5	*MatNumN(n)*	Index for soil subdomain associated with node *n*.
	Record 5 is required for each node *n*, starting with *n*=1 and continuing sequentially until *n=NumNP*.	
	<u>Element Information</u>	
6,7	- -	Comment lines.
8	*e*	Element number.
8	*KX(e,1)*	number of corner node *1*.
8	*KX(e,2)*	number of corner node *2*.
8	*KX(e,3)*	number of corner node *3*.
8	*KX(e,4)*	number of corner node *4*.
	Indices *1,2,3,* and *4* refer to corner nodes of an element *e* taken in a counter-clockwise direction. For triangular elements *KX(e,4)* must be equal to *KX(e,3)*.	
8	*MatNumE(e)*	Index of soil subdomain that is associated with element *e*.
	Record 8 is required for each element *e*, starting with *e*=1 and continuing sequentially until *e=NumEl*.	

Table 3.2 - continued.

Record	Variable	Description
		<u>Boundary Information</u>
9,10	-	Comment lines.
11	*n*	Boundary node number.
11	*CodeW(i)*	Code specifying the type of *water* movement boundary condition applied to the node. Permissible values are 0, 1, ±2, ±3, ±4, ±6 (see Section 3.2). Index *i* denotes the sequential number on the boundary node *n* in this file.
11	*CodeS(i)*	Code specifying the type of *solute* movement boundary condition applied to the node. Permissible values are 0, 1, ±3 (see Section 3.2). Index *i* denotes the sequential number of the boundary node *n* in this file.
11	*CodeH(i)*	Code specifying the type of *heat* movement boundary condition applied to the node. Permissible values are 0, 1, ±3, ±4, ±6 (see Section 3.2). Index *i* denotes the sequential number of the boundary node *n* in this file.
11	*CodeG(i)*	Code specifying the type of *gas* movement boundary condition applied to the node. Permissible values are 0, 1, ±3, ±4, ±6 (see Section 3.2). Index *i* denotes the sequential number of the boundary node *n* in this file.
11	*Width(i)*	Width [units=L] of the boundary associated with boundary node *n*. In the case of planar geometry (*KAT*=2) *Width(i)* includes half of the boundary length of each element connected to the node *n* along the boundary, cm. In the case of axisymmetric flow (*KAT*=1) *Width(i)* represents the area of the boundary strip, cm^2, associated with node *n*. Set this value to zero for internal nodes.

Record 11 is required for each boundary node.

<u>Seepage Face Information</u>

Record	Variable	Description
12,13	-	Comment lines.
14	*Nseep*	Number of seepage faces expected to develop. The file ends here if *NSeep*=0.
15	-	Comment line.
16	*NSP(1)*	Number of nodes on the first seepage face.
16	*NSP(2)*	Number of nodes on the second seepage face.
.	.	.
.	.	.
16	*NSP(Nseep)*	Number of nodes on the last seepage face.
17	-	Comment line.
18	*NP(1,1)*	Sequential global number of the first node on the first seepage face.
18	*NP(1,2)*	Sequential global number of the second node on the first seepage face.
.	.	.
.	.	.
18	*NP(1,NSP(1))*	Sequential global number of the last node on the first seepage face.

Records 17 through 18 are needed for each seepage face.

```
**** Example 3.3:  GRID & BOUNDARY INFORMATION: 'GRID_BND.DAT'
KAT     NumNP    NumEl   NumBP IJ
2       18       11      9     4
 n        x(n)     z(n)    MatNumN(n)
 1        0.0      70.0    1
 2        8.0      70.0    1
 3        0.0      60.0    1
 4        8.0      60.0    1
 5       16.0      60.0    1
 6       25.0      60.0    1
 7        0.0      50.0    1
 8        8.0      50.0    1
 9       16.0      50.0    1
10       25.0      50.0    1
11        0.0      30.0    2
12        8.0      30.0    2
13       16.0      30.0    2
14       25.0      30.0    2
15        0.0      10.0    2
16        8.0      10.0    2
17       16.0      10.0    2
18       25.0      10.0    2
Element information
 e     KX(e,1)    KX(e,2)  KX(e,3)  KX(e,4) MatNumE(e)
 1     1           3         4        2        1
 2     2           4         5        5        1
 3     3           7         8        4        1
 4     4           8         9        5        1
 5     5           9        10        6        1
 6     7          11        12        8        2
 7     8          12        13        9        2
 8     9          13        14       10        2
 9    11          15        16       12        2
10    12          16        17       13        2
11    13          17        18       14        2
Boundary information
 n      CodeW(I)     CodeS(I)     CodeH(I)     CodeG(I)     Width(I)
 1      -4              -4           0            -3            4.0
 2      -4              -4           0            -3            8.0
 5       1               3           0             0            8.5
 6       1               3           0             0            7.0
12       6               1           0             0            0.0
15      -2               0           0             1            4.0
16      -2               0           0             1            8.0
17      -2               0           0             1            8.5
18      -2               0           0             1            4.5
Seepage face information
NSeep
1
NSP(1)   ........ NSP(NSeep)
4
NP(NSP,1)   NP(NSP,2)    NP(NSP,3)    NP(NSP,4)
15          16           17           18
```