TDRFree: a software tool to visualize automatically measured soil moisture content in an arbitrary-shaped soil profile.

Theory, user manual and examples.

Version 1.0

Jan G. Wesseling and Klaas Oostindie

Date: April 23, 2013

Contents

Abstract

Software for two dimensional visualization of values that have been automatically measured with in place sensors is difficult to find. Usually these programs assume a regular area and a regular grid of measuring points. In practice, however, both the shape of the area and the position of the sensors are often irregular. This manual describes the program TDRFree, which visualizes the soil moisture content that has been automatically measured with a set of measuring devices distributed over an irregular 2-dimensional soil profile. Both the applied methods and a description of the program will be described, together with some examples of input and output. The output consists of a series of contour-plots which can be easily combined and presented as an animation. It is also possible to generate values for a derived property such as soil water repellency, which depends on moisture content.

1 Introduction

In recent decades a number of computer programs have been developed to simulate two- or three-dimensional flow of groundwater in both unsaturated and saturated soils ([6]; [7]; [3]). These models include a module to graphically visualize the results of the simulations in 2-dimensional contour-plots. To validate the model results and gain more insight into the soil moisture flow processes, moisture contents are measured continuously in the field, yielding a lot of data. Presentation of the measured data is usually done using contour-plots to show the values at different measurement dates/times. Charting features of spreadsheets like Excel (MicroSoft Office, http://www.microsoft.com/) or Calc (http://www.openoffice.org/) are often employed for this purpose. These packages can only handle data measured with a regular x-y grid and can only present one plot at a time. If irregular grids are to be represented, more sophisticated packages like MathLab (http://www.mathworks.com/) can be applied, or one of the packages from R (http://www.r-project.org/).

A few years ago the program TDRProcessor ([10], [9], [8]) was developed to create animations from measured soil moisture data. The 'movies' generated with that software were presented at a lot of national and international scientific conferences and symposia. Though the software has been successfully and extensively used, some desirable extensions have also been identified, such as (Figure 1):

1. Capability to visualize data from irregular shaped soil profiles (e.g. potato fields);
2. Capability to account for irregularly installed sensors;
3. Option to show a larger number of soil profiles simultaneously;
4. Visualization of additional moisture related soil properties. -

In 2009 we got the opportunity to install a number of tdr-sensors in a potato-field. This was part of an experimental setup where the differences in soil moisture and the growth of potatoes were measured under different conditions. Automatic measurements of soil moisture content were performed at 4 different plots. The sensors were installed in a rather irregular pattern (Figure 2). A few sensors were placed in the ridges the potato plants were grown on. Moisture content was measured and registered automatically every 2 hours. Due to these irregularities the original program could not be used to process the measured moisture contents. A quick internet scan revealed that while there are publications available on the processing of data measured on an irregular grid (e.g. [11]; [5]), no user-friendly programs to work with this kind of data are available. Therefore we decided to create

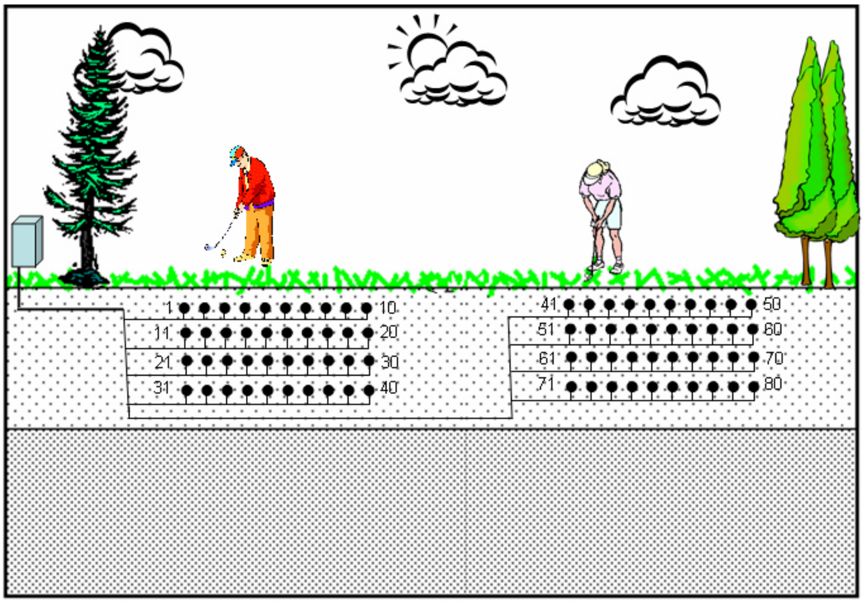


Figure 1: The setup of measuring devices as required by the first version of the program ([9]).

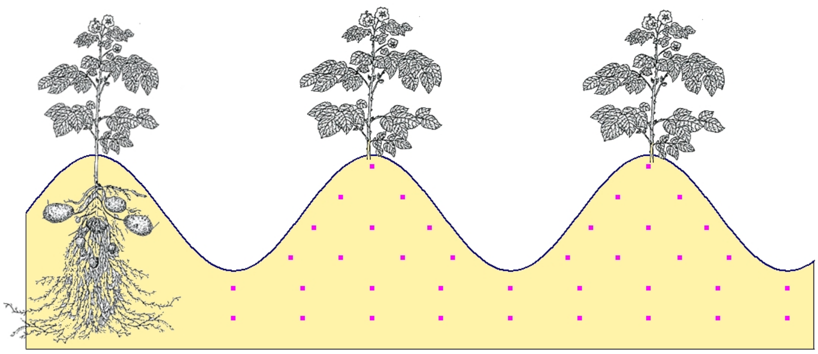


Figure 2: The positions of tdr-sensors under the ridges and furrows of a potato-field.

our own program and named it TDRFree.

This report first presents a few paragraphs describing some of the applied theories. Next the required input files are described, followed by a description of the program and a users-guide, including some examples of output.

2 Applied theory

2.1 Linear interpolation.

2.1.1 General

One of the most widely applicable mathematical methods in solving differential equations numerically is the finite element method (FEM). When applying this method for a 2-dimensional problem, the considered area is divided into a number of elements (usually triangular or quadrilateral shapes) that are defined by specifying the coordinates of the corner points (usually called nodes) (see e.g. [4]). One of the basic techniques in the FEM is the transformation of a quadrilateral or triangular shape in the real (global) coordinate system into a standard shape (square or triangle) in a local coordinate system. These shapes and transformations are then used to solve the governing (partial) differential equations. The transformation technique applied in the FEM will be used here to interpolate moisture content values in the areas between tdr-sensors.
Suppose we have a function Ξ that is 1 in one of the corners of the polygon and 0 in the other corners. Then we can approximate any value inside the polygon by

$$f(\bar{x}) = \sum_{i=0}^{N_e-1} \Xi_i(\bar{x}) f(\bar{x}_i) \tag{1}$$

where N_e is the number of corners of the polygon. Assuming a linear function, then Ξ can be written as

$$\Xi = a_0 + a_1 x + a_2 y + a_3 xy \tag{2}$$

This equation can be applied as well to transform an irregular quadrilateral in the global coordinate system (x,y) to a square in the local coordinate system (ϵ, η) and vice versa (See Figure 3).

2.1.2 From local to global coordinates

For simplicity we derive 2 separate transforms: one for the horizontal coordinate and one for the vertical coordinate. Introducing a parameter vector $\bar{\alpha}$ for the horizontal coordinate and a vector $\bar{\beta}$ for the vertical coordinates then the following sets of equations can be distinguished:

$$\Upsilon \bar{\alpha} = \bar{x} \tag{3}$$

and

$$\Upsilon \bar{\beta} = \bar{y} \tag{4}$$

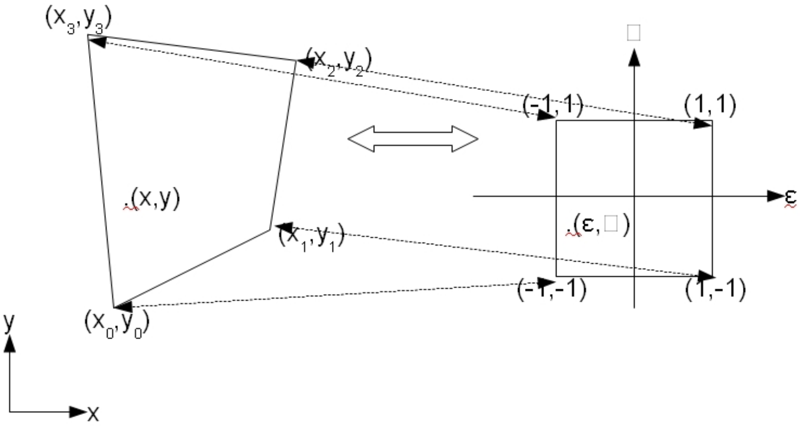


Figure 3: Transformation from global to local coordinates and vice versa.

Substituting the elements of Υ with the coordinates of the corners in the local coordinate system, the following sets of equations are created:

$$\begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \tag{6}$$

When we consider triangles as perturbed quadrilaterals, the values of $\bar{\alpha}$ and $\bar{\beta}$ can be computed for each quadrilateral and triangular shape by solving these sets of linear equations, e.g. by Gaussian elimination. Then the corresponding point in the global coordinate system can be computed for an arbitrary point in the local system by

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_0 & \beta_1 & \beta_2 & \beta 3 \end{pmatrix} \begin{pmatrix} 1 \\ \epsilon \\ \eta \\ \epsilon\eta \end{pmatrix} \tag{7}$$

2.1.3 From global to local coordinates

To convert from the global to the local coordinate system, similar functions can be derived:

$$\Upsilon\bar{\gamma} = \bar{\epsilon} \tag{8}$$

and

$$\Upsilon\bar{\delta} = \bar{\eta} \tag{9}$$

Substituting the elements of Υ with the coordinates of the corners in the global coordinate system, the following sets of equations are created:

$$\begin{pmatrix} 1 & x_0 & y_0 & x_0y_0 \\ 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} \tag{10}$$

$$\begin{pmatrix} 1 & x_0 & y_0 & x_0y_0 \\ 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \end{pmatrix} \begin{pmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \tag{11}$$

Now the values of $\bar{\gamma}$ and $\bar{\delta}$ can be computed for each quadrilateral and triangular shape by solving these sets of linear equations, e.g. by Gaussian elimination. Then the corresponding point in the local coordinate system can be computed for an arbitrary point in the global system by

$$\begin{pmatrix} \epsilon \\ \eta \end{pmatrix} = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 \\ \delta_0 & \delta_1 & \delta_2 & \delta 3 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ y \\ xy \end{pmatrix} \tag{12}$$

2.2 The inner area of a closed polygon.

The inner area of a closed polygon can be computed as

$$A_e = 0.5 * \sum_{i=0}^{N_e-1} (x_i y_j - x_j y_i) \tag{13}$$

with

$$j = (i+1) \diamond N_e \tag{14}$$

where the $\diamond$ stands for the modulus operator.

2.3 Determining whether a point is located within a polygon.

In order to see in which element an arbitrary point is located, a procedure was implemented to determine whether a point is located in the interior or in the exterior of a polygon. This procedure has been described earlier by [1]. Suppose an area is bounded by a closed polygon. This means the endpoint of the polygon is the same as the start point. Now let us assume all corners of the polygon are numbered counter-clockwise (see Fig. 4).

Given a line segment between the subsequent corners of the polygon P_0 (x_0, y_0) and P_1 (x_1, y_1), another point P (x,y) has the following relationship to the line segment:

$$I = (y - y_0)(x_1 - x_0) - (x - x_0)(y_1 - y_0) \tag{15}$$

if I is less than 0 then P is to the right of the line segment, if I is greater than 0 it is to the left, if I is equal to 0 then it lies on the line segment. So, if I is positive for all boundaries of a polygon, then P lies within the polygon.

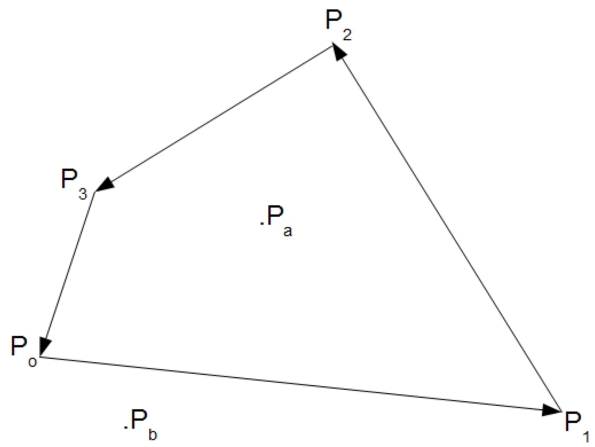


Figure 4: A point inside (P_a) and a point outside (P_b) a polygon.

2.4 Interpolating θ.

To create completely colored soil moisture profiles, it is necessary to assign a color to every pixels of the plot. Therefore all pixels are treated separately. First the x- and y-value of the pixel are converted to global coordinates. These coordinates are then used to check whether the point is located within the profile or not. If so, the element in which it is located is determined. Then the global coordinates of the point under consideration are transformed to the local coordinate system (see previous paragraphs) and the θ-value is computed from the moisture content in the corresponding nodes of the (local) element according to

$$\theta = \sum_{i=0}^{N_e-1} \frac{A_i \theta_i}{A_e} \tag{16}$$

where N_e is the number of corners of the element, A_e is the area of the element, A_i is the fractional area opposite node i and θ_i is the moisture content of node i (see Figure 5).

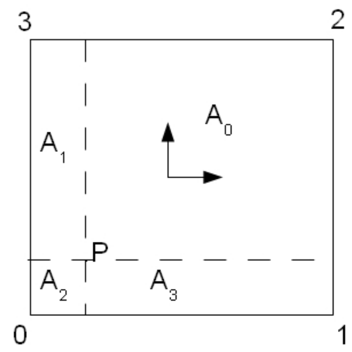


Figure 5: Interpolation of θ-values for a point P in an element in the local coordinate system.

Finally the color corresponding to the computed moisture content is determined and assigned to the pixel.

2.5 Virtual nodes.

In order to obtain the moisture contents in each point of the considered soil profile by interpolation, values are required to interpolate between. Especially near the boundaries of the profile and in the ridges this may cause a problem. If the moisture content in each node of the grid (at the same position as a tdr-sensor) is known, how can the values between the position of the sensor and the boundary of the profile be obtained? Extrapolation of data is risky, because that technique may yield moisture contents that are physically nonsense. Therefore the 'Virtual Node' has been introduced. A virtual node is a node that does not have a physical meaning but is just meant for interpolation purposes. When creating the grid, one is supposed to fill all open spaces outside the soil profile with virtual nodes. In Figure 6 first the grid was created using the positions of the sensors and applying quadrilateral and triangular elements.

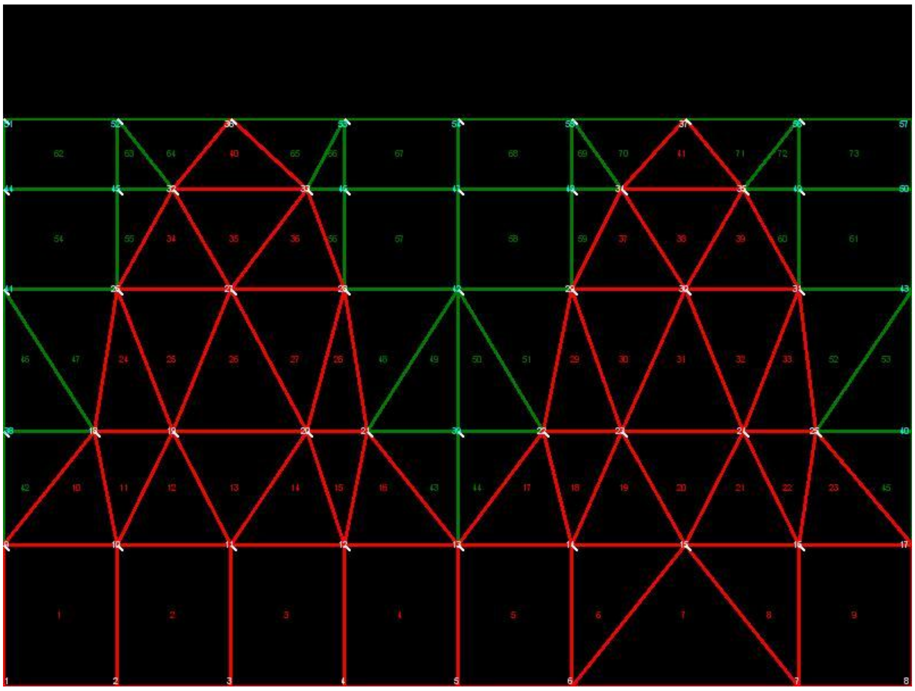


Figure 6: The real and virtual nodes and elements of the plot with potatoes.

These (real) elements are drawn in the red color, the node numbers are presented in white. The next step is to fill the 'holes' in the grid with virtual elements and nodes. These virtual nodes are presented in aqua, the virtual elements are green.

To perform a correct interpolation, first the moisture contents in the virtual nodes should be computed. These values can be computed from the surrounding real and virtual nodes. Let's take a closer look at the area between the ridges in Figure 6. For this purpose, Figure 7 zooms in at this area.

The procedure for computing the moisture content in each virtual node is quite simple: just look around the considered node and compute the θ-value from the nodes that surround the node under consideration. Maximally 3 surrounding nodes can be considered. The value of node 39 in Figure 7 is computed from the nodes 21, 22 and 13. When computing the new values, the inverse of the distance between the nodes is considered as a weight factor.

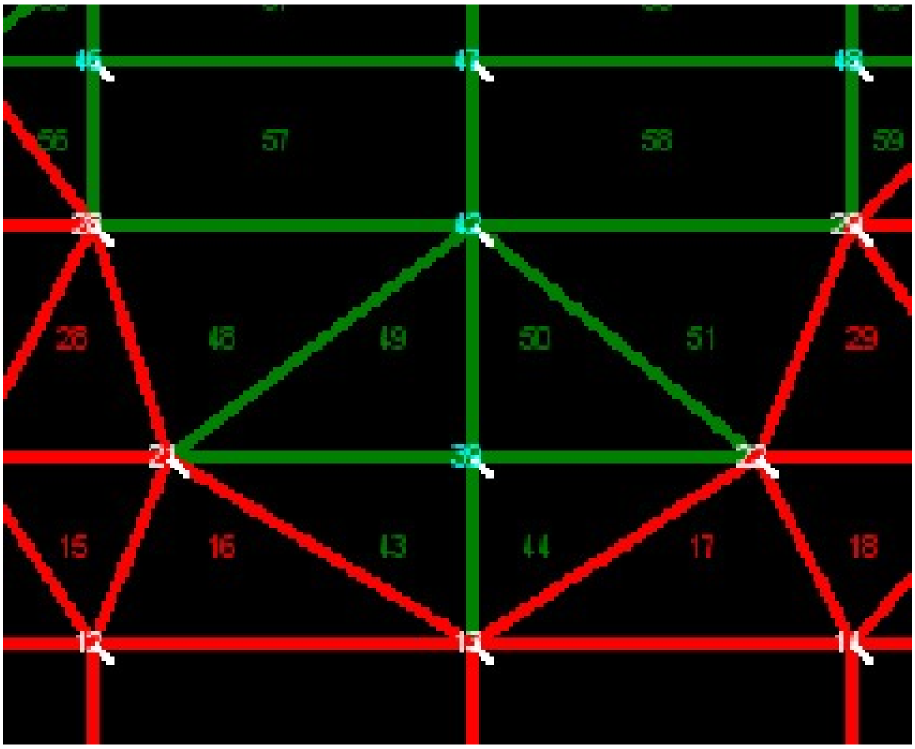


Figure 7: A magnified view of the applied grid with virtual and real nodes and elements.

Written in a more mathematical way:

$$\theta_i = \frac{\sum_{j=0}^{N_k-1} \frac{\theta_j}{\delta_j}}{\sum_{j=0}^{N_k-1} \frac{1}{\delta_j}} \tag{17}$$

where i = number of node considered, N_k= number of nodes with known moisture content surrounding node i and δ_j = distance between node i and node j. If this procedure is followed for each node in the correct order, all virtual nodes will get a moisture content value. The order in which the nodes are to be computed has to be specified by the user and will be described in one of the following sections of this manual.

2.6 Derived properties

This version of the program is not only capable of presenting measured soil moisture content, but also some variable that is directly linked to the moisture content. One example is a so-called drought-class. Imagine that the plants can uptake water when moisture content is higher than e.g. 35 Vol.%, and do not extract any water when the moisture content is lower than 10%. Then we can define 3 classes: dry (0-10%), moderately dry (10-35%) and wet (35% and higher). By assigning a color to each class, similar pictures as were made for moisture content can be made for drought classes. Another property that can be shown this way is water repellancy. This value also depends upon the prevailing moisture content ([2]). The limits of each class can vary with depth in the soil profile. Therefor, the possibility has been created to define a table with limits and depth, thus creating boundaries of classes that vary with depth. The values are interpolated linear between the specified points (Figure 8).

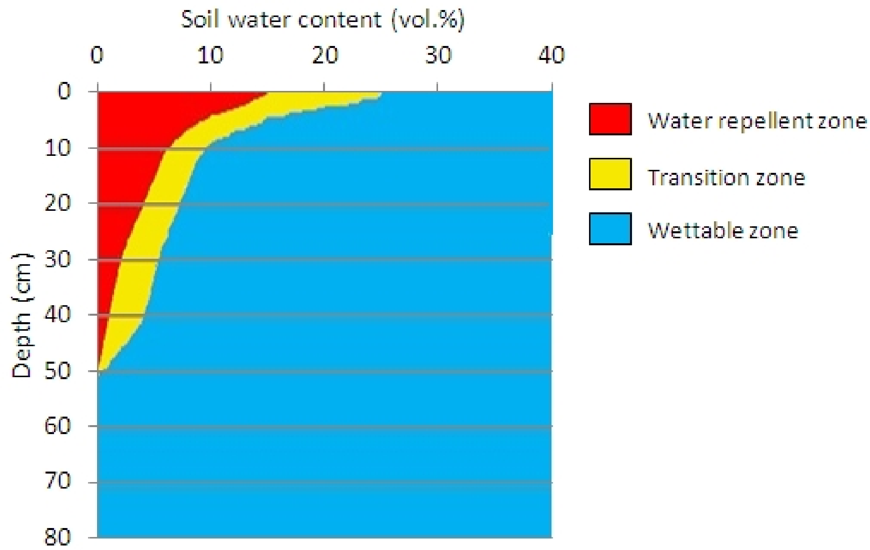


Figure 8: An example of a derived property: the water repellancy as a function of depth and moisture content.

3 Input data

3.1 General

3.2 Precipitation

It is assumed that precipitation has been measured with automatic rain-gauges. These gauges sends a pulse after a certain amount of precipitation (usually 0.2 mm) has been caught. The built-in software then records the time of the pulse and adds the amount of precipitation to its memory. This way the file with measured precipitation data will consist of records with only three items: a date, a time and a cumulative precipitation value. After reading the data from the automatic gauge to a computer, the gauge's memory will be reset, causing the cumulative precipitation to get the value of zero again. In the file, the time and the precipitation value may be separated by a tab-character or by a comma. In the latter case a file with data may look as follows:

Date Time Event (Events 0.2 mm)
26/07/04 2:21:55.0 0
26/07/04 15:10:48.5 0.2
26/07/04 15:11:06.0 0.4
26/07/04 15:11:11.5 0.6
26/07/04 15:12:13.0 1
26/07/04 15:12:26.5 1.2
26/07/04 15:12:32.0 1.4
26/07/04 15:12:37.5 1.6
26/07/04 15:13:14.5 1.8
26/07/04 15:13:47.5 2
26/07/04 15:13:57.0 2.2
26/07/04 15:14:03.0 2.4
26/07/04 15:15:00.0 2.6
26/07/04 15:15:22.5 2.8
26/07/04 15:15:32.5 3
26/07/04 15:16:35.5 3.2
26/07/04 15:16:43.5 3.4
⋮

To process these precipitation data, the file RainProcessor.xls should be read into Microsoft Excel. Now 13 tabsheets will be available: one control-sheet and 12 month-sheets. See Table 1.

Table 1: The worksheets in the precipitation file.

Number	Name	Description
0	Control	Control page
1	January	Precipitation for January
2	February	Precipitation for February
3	March	Precipitation for March
4	April	Precipitation for April
5	May	Precipitation for May
6	June	Precipitation for June
7	July	Precipitation for July
8	August	Precipitation for August
9	September	Precipitation for September
10	October	Precipitation for October
11	November	Precipitation for November
12	December	Precipitation for December

The tabsheet 'Control' is the one where you give some details about the files to be read and where you can start the processing. There are three entry-fields on this sheet as presented in Table 2 and Figure 9.

Table 2: Cells for input in the precipitation file.

Cell	Contents
A1	Directory where files are located
A2	First characters of file
A3	Notation of date

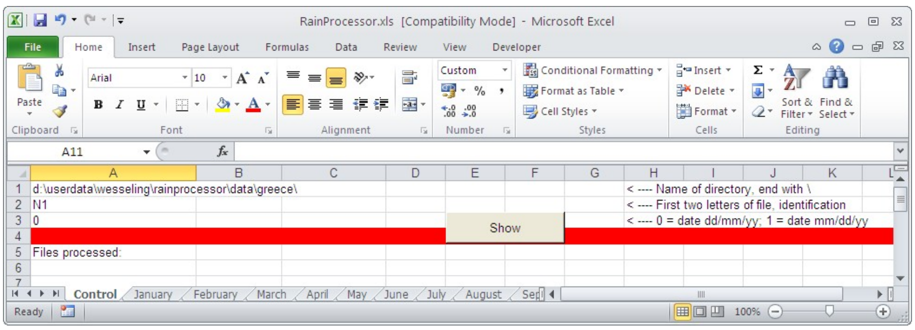


Figure 9: The control sheet for processing the precipitation data.

It is assumed that all files with precipitation from the meteo station as located at the same directory. It is also assumed that the first characters of the filename are the same, indicating the name of the station. (Be careful: when you edit a file, often the old file gets a new extension (often .bak). As

RainProcessor reads **all** files, not considering extensions, the .bak-file should be deleted before running RainProcessor.. Cell A3 may contain either a 0 or a 1. In the first case the date is specified as dd/mm/yy (Dutch system), in the latter case it is specified as mm/dd/yy (US system). Once these fields are filled correctly, clicking the 'Show'-button starts reading the files. This is performed in a number of steps:

1. Read the names of all files in the specified directory.
2. Does the name of the file include the specified string?
3. if so, process the data
4. If there is another file, go to step 2

The processing of a file consists of different steps as well:

1. Read a line
2. Extract the date
3. Put the date and value on the correct month-sheet
4. If there is another line, go to step 1
5. Close the file

After all files have been read, the post-processing takes place for every month with data:

1. Sort the data on date/time
2. Check for resets and process them
3. Calculate intensities (mm/h)
4. Make the graphical presentation of the data

In step 2 of the post-processing procedure it is checked whether there are resets in this month. A reset means the logger has been emptied and the value of the precipitation is put to 0 again. Corrections will be made to create a sheet with continuous data. Finally every month-sheet consists of three columns of data: the date-time, the cumulative precipitation (mm) and the calculated intensities. Every sheet also has a chart showing these values graphically:

After reading all files the control-sheet will look like

The tdr-program only reads the first two columns (the date-time and the cumulative values) and processes these data to obtain precipitation values over the desired period.

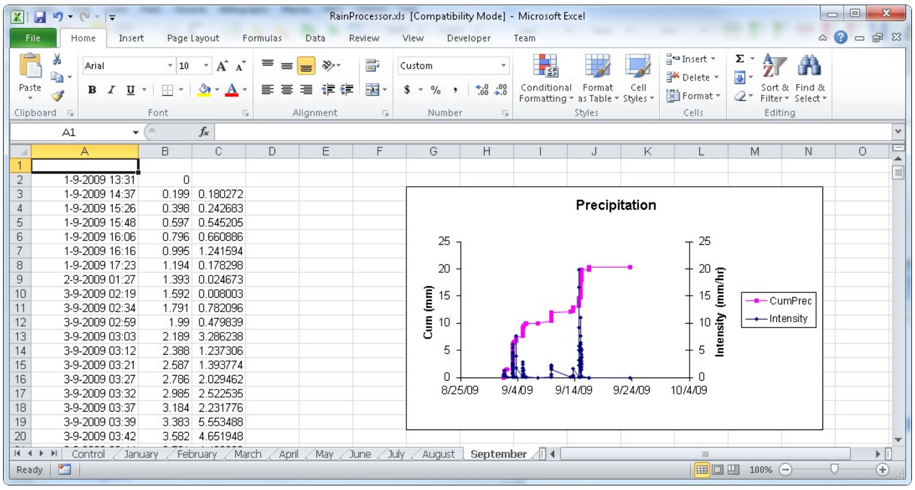


Figure 10: An example of monthly precipitation data.

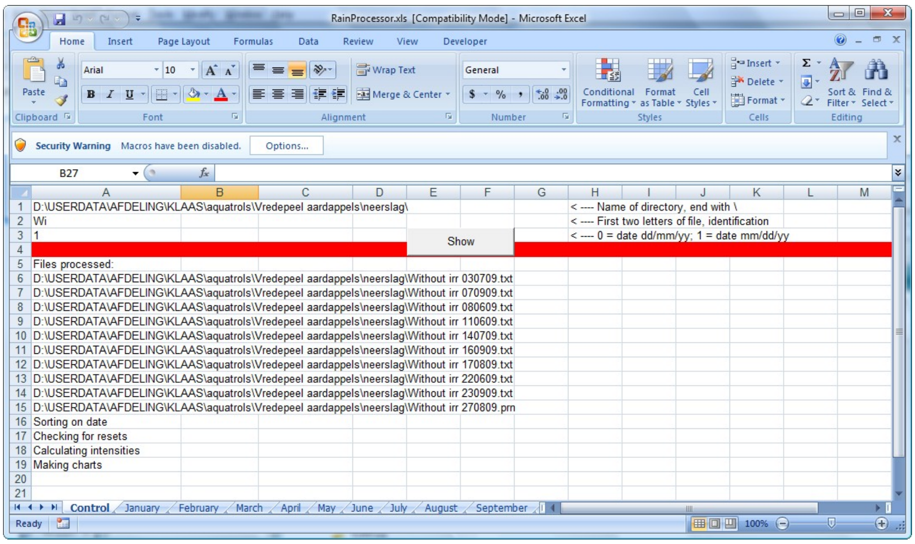


Figure 11: The Control worksheet after reading the data.

3.3 Irrigation data

The applied irrigation should be stored in an Excel-file as well. There is only one worksheet required in this file, a sheet with the name "Irrigation". This sheet should contain at least two columns with data: the A and the B-column. In the A-column the dates (and, if they are known, the times) of irrigation supplies should be stated. The B-column then contains the size of the irrigation supply (in mm). Data will be read starting from row 2, allowing column headers in row 1. An example is presented in Figure 12.

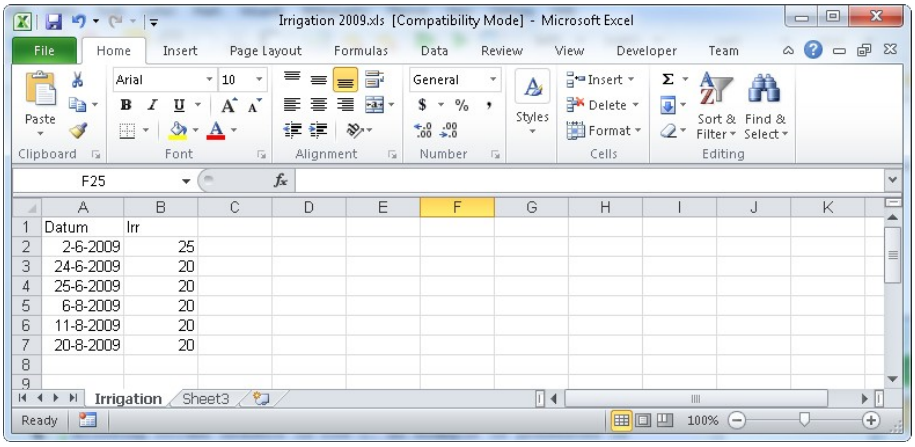


Figure 12: The Excel spreadsheet with irrigation data.

3.4 TDR data

3.4.1 General

The Excel-file with TDR data also serves as the control file for the program. The file should at least have the 8 worksheets listed in Table 3.

Table 3: Worksheets required in the Excel-file with TDR data.

Worksheet	Contents
Control	General data
Nodes	Locations of sensors
Elements	Construction of elements required for interpolation
PlotBoundaries	Boundaries of the soil profile with sensors
Virtual	The virtual nodes for each plot
Colors	The RGB-values for soil moisture and derived classes
Moisture	The measured moisture contents
Derive	Functions to derive values from moisture content

These worksheets will be discussed in the following paragraphs.

3.4.2 Worksheet Control

In the worksheet "Control" the user may specify what to show and how to show it. See Figure 13 for an example. In this figure the red ellipses point to the required input fields. These fields will be discussed below.

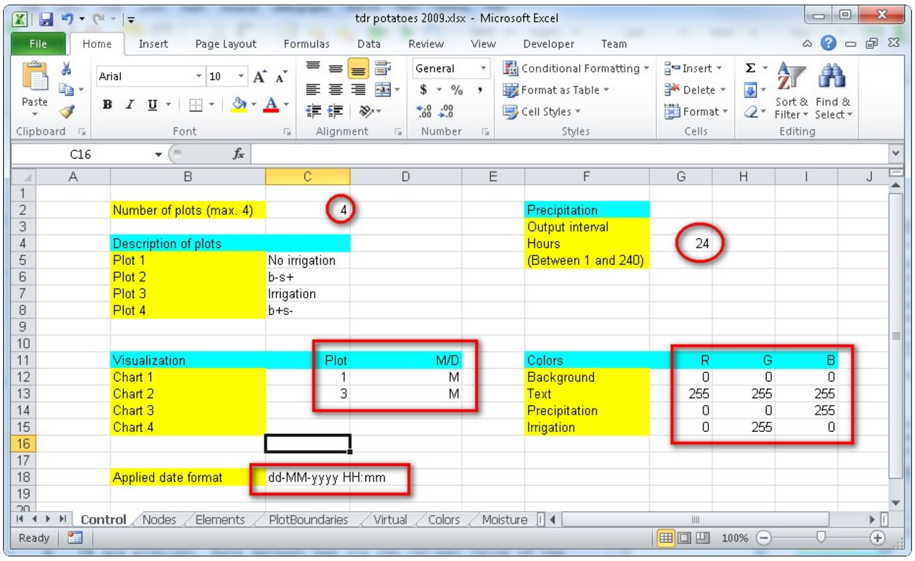


Figure 13: The worksheet 'Control' of the Excel file with tdr-measurements.

The program TDRFree is able to show soil moisture data and/or derived values of up to 4 different plots. These plots may be designed and constructed to gain information about the effects of different treatments on soil moisture flow. It is assumed that the moisture content at all plots is measured at the same time. The number of plots for which moisture content is measured, is specified in cell C2 of this worksheet. Each plot can be described by a few words in the cells C5-C8.

It is possible to show up to 4 charts with data at a time. These data may either be soil moisture contents or derived values. In the cells C12 - C15 the user can specify the plotnumber which data should be shown in the corresponding chart. The cells D12-D15 may only contain either a M (moisture content) or a D (Derived data). The number of charts to be shown is derived from the number of values in the C-column. If cell C2 is empty, only one chart will be shown.

In one of the preceding sections of this manual the Excel-file with precipitation data has been described. It was seen there that the precipitation data are recorded at various moments of time, usually after 0.2 mm has been fallen. To present these data in a more user-friendly way, the program integrates the precipitation data over a user-specified period. The length of this period (in hours) can be specified in cell G4[1]. The recommended size of this value varies between 4 and 24. If the value is 4, then 6 precipitation bars will be shown for each day of measurement. If it is set to 24, only one value will be presented. The most correct value depends on the frequency of measurements of soil moisture content. If these data are measured only once a day, it is an overkill to present the precipitation on an hourly basis. On the other hand, when precipitation is presented on a daily basis and soil moisture is measured every hour, it may be very annoying to wait for the effect of a known precipitation during 23 steps when the rain has fallen between 11 PM and midnight. Each dataset has its own optimal value of the integration period.

Because the selection of the optimal color of the background and the text fields and axes of the precipitation chart is a matter of personal taste and has lead to discussions a lot of times, it was decided to enable the user to select his own colors. These colors may be specified as RGB-values in the columns G-I for the background (row 15), the text (row 16), the bars with precipitation data (row 17) and the irrigation data, if any (row 18).

At the lower left of the tabsheet the applied date format is specified. This indicates the way all dates are presented in the spreadsheets for moisture contents, precipitation and irrigation [2]. The coding of this format is the same as is used in Microsoft Windows.

3.4.3 Worksheet Nodes

As this program is based upon a derivative of the Finite Element Method, a grid of nodes should be defined by the user [3]. As discussed before, the method applied here uses so-called Virtual Nodes to make the grid complete. The positions of the real (=non-virtual) nodes are the same as the positions of the tdr-sensors. A grid has to be defined for each plot under consideration. The x- and z-coordinates of each node are specified on the worksheet Nodes (Figure 14).

Each plot has 4 columns of data: the first column contains the Id of the node, the second column contains the x-coordinate, the third column contains the z-coordinate and in the fourth column it is indicated whether the node is real (0) or virtual (1). In the upper 2 rows only the plot number and the meaning of the column are presented. These rows are not used by the program and are inserted only to improve the readability of the data. For the same reason an empty red-colored column has been inserted at the right-hand side of the data for each plot. To distinguish between the data for each plot, different colors have been used: light blue, yellow, orange and

[1] Be careful: the program does not check if this value is realistic or possible.

[2] It is not possible to use a different type of date in each file!

[3] In the future some network generator will be created to perform this labor-intensive and boring work.

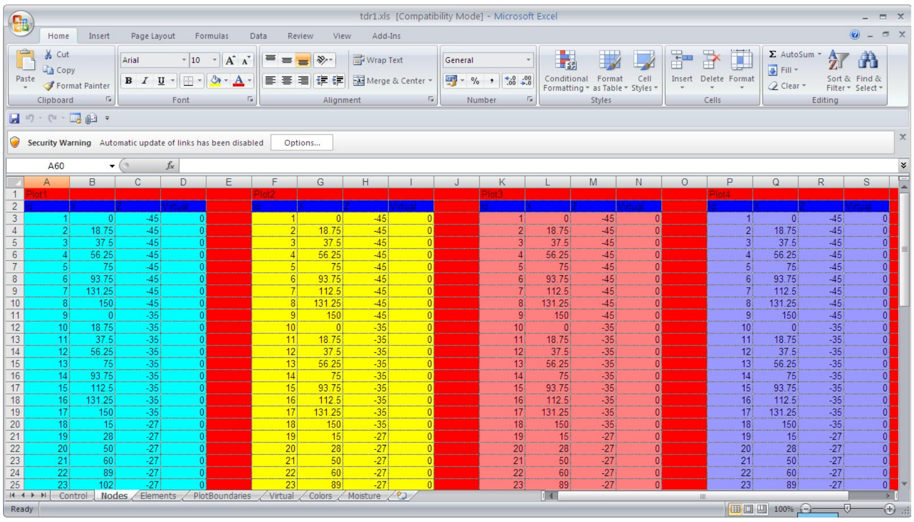


Figure 14: The worksheet 'Nodes' of the Excel-file with tdr-measurements.

violet. These colors will be applied throughout the entire Excel workbook.

3.4.4 Worksheet Elements

From the nodes described in the previous paragraph, elements can be made that are used for interpolation as described in the Finite Element Method. In general these elements are quadrilaterals with a regular shape: rectangles. If the moisture sensors are installed at irregular positions however, the shape of the quadrilateral may be irregular. When elements are applied to follow some irregular boundary or a change in sensor position, triangles may be used instead of quadrilaterals. These triangles can be seen as perturbed quadrilaterals and can be entered by setting two sequential corners to the same node. See Figure 15.

In the worksheet the same colors as in the worksheet for nodes have been applied for the different plots. Each plot now has 6 columns: Id, lower-left corner, lower-right corner, upper-right corner and upper-left corner. The last column indicates whether this is a real (0) or virtual (1) element. See Figure 16) for an example.

3.4.5 Worksheet PlotBoundaries

In the previous version of this program ([10], [8], [9]) it was assumed that the profile to be plotted was bounded by the locations of the tdr-sensors. In the present version the boundary has to be specified by the user. This should be accomplished in worksheet PlotBoundaries. See Figure 17. On

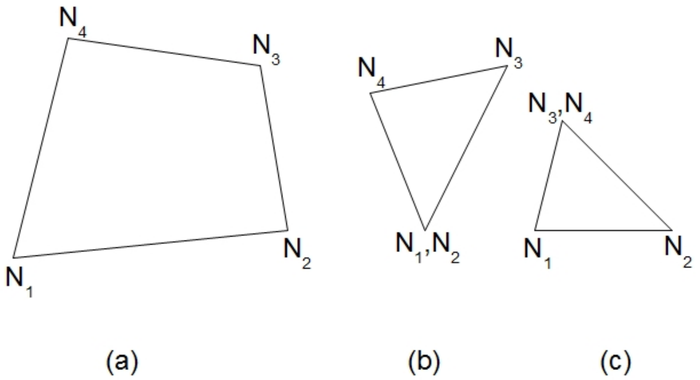


Figure 15: Examples of quadrilateral (a) and perturbed (b,c) elements.

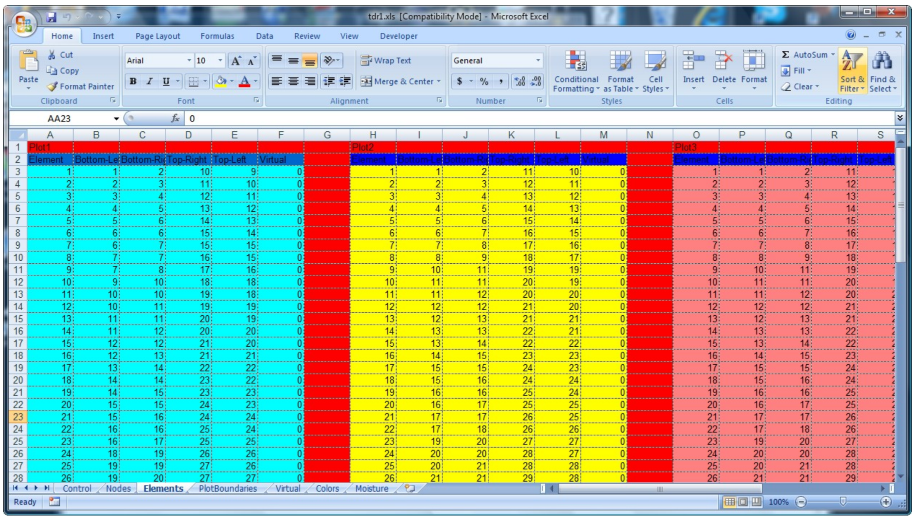


Figure 16: The worksheet Elements of the Excel-file with tdr-measurements.

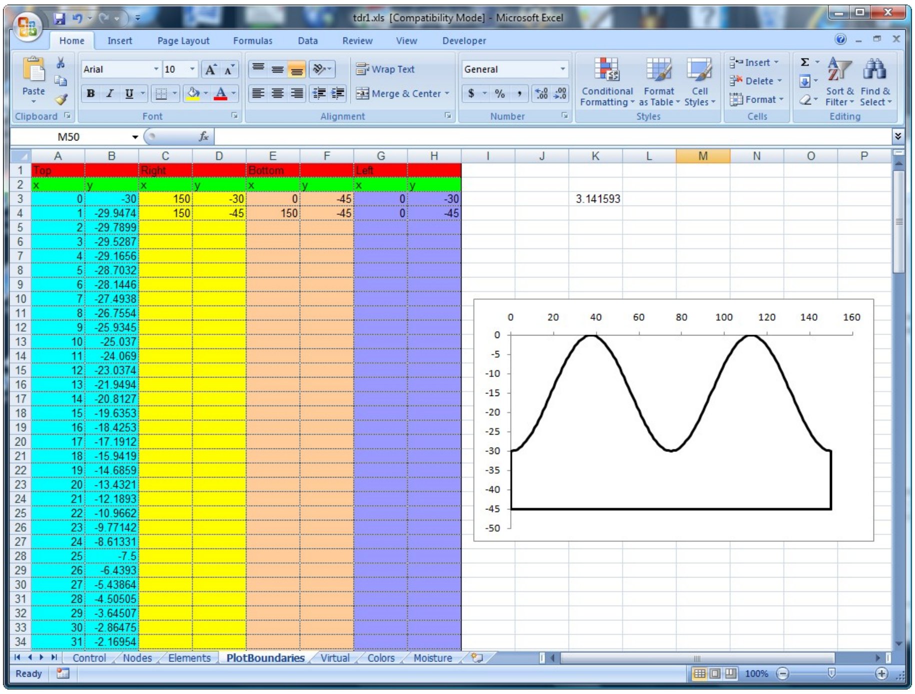


Figure 17: The worksheet 'PlotBoundaries' of the Excel-file with tdr-measurements.

this page each side of the plots to be drawn [4] are specified by 2 or more pairs of (x,y)-coordinates. The program will perform a linear interpolation to obtain the values between the specified points. From left to right two columns are used for respectively the top boundary (blue), the right-hand side boundary (yellow), the bottom boundary (orange) and the left-hand side boundary (purple). The upper 2 rows of the worksheet are not read by the program. At the right-hand side of the Figure a simple plot can be seen that has been drawn from the specified data.

3.4.6 Worksheet Virtual

In one of the previous chapters of this manual the principle of the Virtual Nodes has been explained. In this worksheet the relationship between the virtual node and its neighbors is defined. The same color coding has been applied for the different plots that was applied on the previous worksheets. For each plot four columns with numbers are reserved (Figure 18): one for the Id of the virtual node, the other 3 contain the Ids of the surrounding nodes that are used to compute the moisture content of the virtual node.

The user should ALWAYS specify numbers for all three of the neighbor-

[4] It is assumed all plots have the same contours.

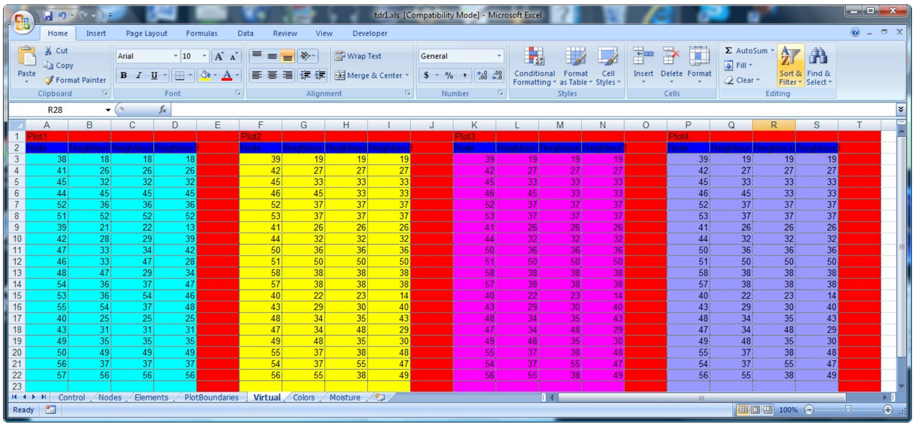


Figure 18: The worksheet 'Virtual' of the Excel-file with tdr-measurements.

ing nodes. If there are less nodes to be applied, multiple fields may contain the same values. The program will then only consider one of these and ignore the other(s). The order in which the moisture content of the virtual nodes will be calculated is the same as the order in which they are specified on this worksheet. This implies that, when a virtual node i requires the value of another virtual node j, node j should be higher in the list than node i.

3.4.7 Worksheet Colors

As the program should know what color should be assigned to each moisture content and derived value, the worksheet Colors (Figure 19) has been created. At the left-hand side of this worksheet the soil moisture classes to

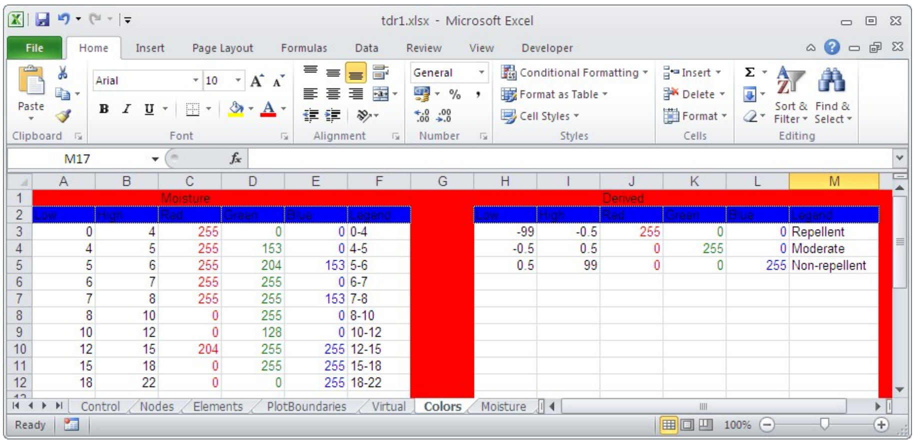
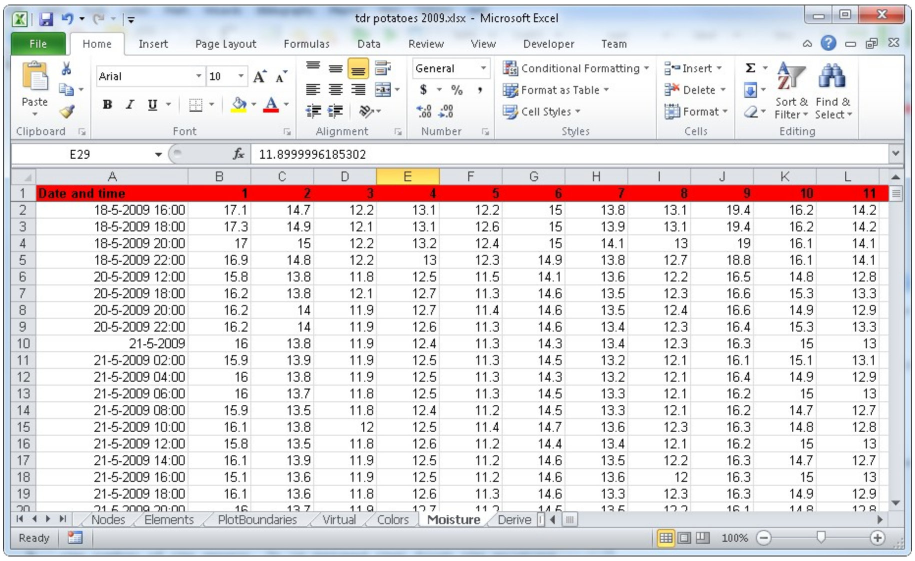


Figure 19: The worksheet 'Colors' of the Excel-file with tdr-measurements.

be distinguished are defined. This is accomplished by defining a lower and upper moisture content of each class and the required color, specified by its R(ed)G(reen)B(lue) components. At the right-hand side the same is done for the derived classes (water repellency in the example). It is advised to define a limited number of classes only (5 to 10 for moisture content).

3.4.8 Worksheet Moisture

On the worksheet called 'Moisture' the actual (tdr-measured) moisture data is presented (Figure 20).



Date and time	1	2	3	4	5	6	7	8	9	10	11
18-5-2009 16:00	17.1	14.7	12.2	13.1	12.2	15	13.8	13.1	19.4	16.2	14.2
18-5-2009 18:00	17.3	14.9	12.1	13.1	12.6	15	13.9	13.1	19.4	16.2	14.2
18-5-2009 20:00	17	15	12.2	13.2	12.4	15	14.1	13	19	16.1	14.1
18-5-2009 22:00	16.9	14.8	12.2	13	12.3	14.9	13.8	12.7	18.8	16.1	14.1
20-5-2009 12:00	15.8	13.8	11.8	12.5	11.5	14.1	13.6	12.2	16.5	14.8	12.8
20-5-2009 18:00	16.2	13.8	12.1	12.7	11.3	14.6	13.5	12.3	16.6	15.3	13.3
20-5-2009 20:00	16.2	14	11.9	12.7	11.4	14.6	13.5	12.4	16.6	14.9	12.9
20-5-2009 22:00	16.2	14	11.9	12.6	11.3	14.6	13.4	12.3	16.4	15.3	13.3
21-5-2009	16	13.8	11.9	12.4	11.3	14.3	13.4	12.3	16.3	15	13
21-5-2009 02:00	15.9	13.9	11.9	12.5	11.3	14.5	13.2	12.1	16.1	15.1	13.1
21-5-2009 04:00	16	13.8	11.9	12.5	11.3	14.3	13.2	12.1	16.4	14.9	12.9
21-5-2009 06:00	16	13.7	11.8	12.5	11.3	14.5	13.3	12.1	16.2	15	13
21-5-2009 08:00	15.9	13.5	11.8	12.4	11.2	14.5	13.3	12.1	16.2	14.7	12.7
21-5-2009 10:00	16.1	13.8	12	12.5	11.4	14.7	13.6	12.3	16.3	14.8	12.8
21-5-2009 12:00	15.8	13.5	11.8	12.6	11.2	14.4	13.4	12.1	16.2	15	13
21-5-2009 14:00	16.1	13.9	11.9	12.5	11.2	14.6	13.5	12.2	16.3	14.7	12.7
21-5-2009 16:00	15.1	13.6	11.9	12.5	11.2	14.6	13.6	12	16.3	15	13
21-5-2009 18:00	16.1	13.6	11.8	12.6	11.3	14.6	13.3	12.3	16.3	14.9	12.9

Figure 20: The worksheet 'Moisture' of the file with tdr-measurements.

The first column of this worksheet contains the date and time at which the moisture contents are measured. The format of the date and time should agree with the format specified on tabsheet 'Control'. The first row contains the number of the sensor. It is assumed that first the moisture contents measured on plot 1 are presented, followed by those measured at plot 2 (if that plot exists), etc. Usually the values of moisture content will be given in volumetric percentages or fractions. For the program itself it is irrelevant what units are applied, as long as the dimensions on the 'Moisture'-worksheet comply with those at the 'Colors'-worksheet.

3.4.9 Worksheet Derive

The last worksheet in the workbook contains the definitions of the derived values (Figure 21). Once more we see the colors of the different plots on this worksheet. It is possible to define different functions for each plot, though they will all be shown with the classification defined on the 'Colors' worksheet. Row 1 only contains the plotnumber and is not read by the program.

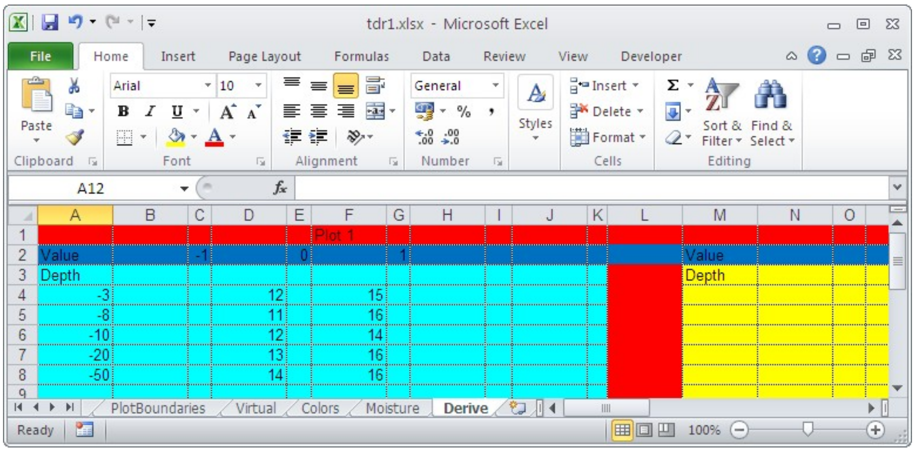


Figure 21: The worksheet 'Derive' of the file with tdr-measurements. Only the definitions for plot 1 are shown.

In the second row the values are presented that the derived function can have. These values start in the third column. The second value is in the fifth column, the third in column 7, and so on. Maximally 5 function values can be specified for each plot. The program reads the cells in this row and determines the number of boundaries from the number of filled cells: 2 values mean 1 boundary, 3 values mean 2 boundaries, etc. It is assumed that the first value is valid below the first boundary, the second value is valid between the first and second boundary and the last value is valid for moisture contents beyond the last boundary. The boundaries are presented as a number of moisture contents at a certain depth [5]. The depths are to be specified in the first column, starting at row 4. Data are read until an empty cell is discovered.

[5] Remember: positions are negative numbers, starting with 0 at the soil surface and decreasing with depth.

4 Running the program

4.1 General

The program TDRFree.exe can be started in one of the usual Windows-ways. When it has been loaded, the user will see the form presented in Figure 22.

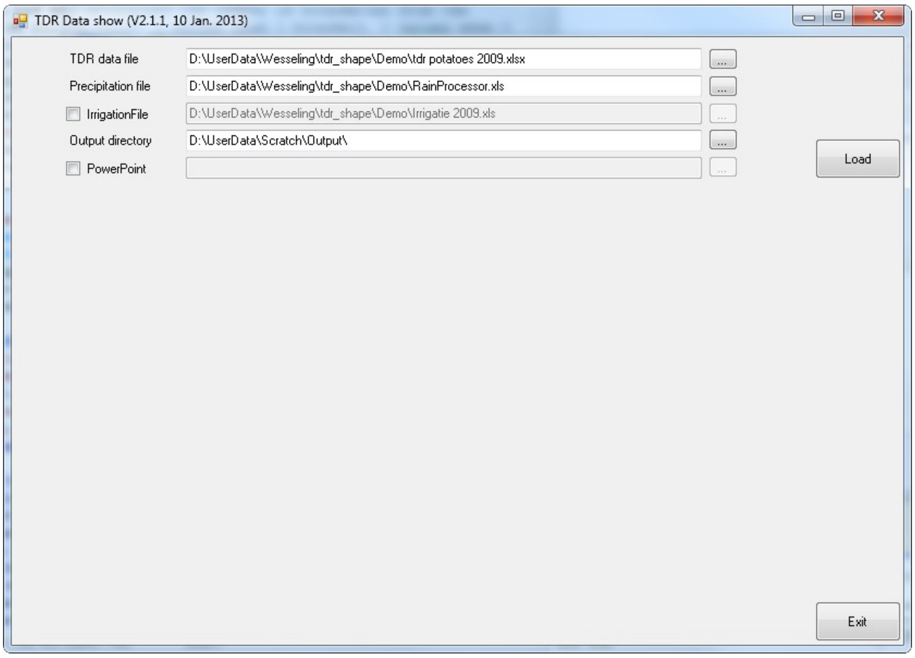


Figure 22: The initial screen of the program TDRFree.

At the top of the screen 5 textboxes are visible, each of them meant to specify an input- or output file or a directory. In the top text box the path and name of the Excel file with TDR-measurements should be specified. The second text box should contain the path and name of the file with precipitation data. Below that, the full name of the Excel file with irrigation data can be specified. As this file is optional, the box before the name has to be checked to indicate irrigation data should be shown as well. If the box is checked, specifying a name will be possible. If it is not checked, the corresponding text box will be greyed out and typing in it is disabled. At the right side of these three text boxes there is a small button. Clicking on one of these buttons enables the user to browse for a file. After selecting a file, the name of the selected file will be shown in the corresponding text box.
The output of the program will be stored as a number of jpg-images on disk. The user can specify the directory where these images should be stored. [6] As

[6]The directory should always end with a \

the program is able to create Powerpoint presentations as well, a checkbox is provided to indicate whether a Powerpoint presentation is required or not. If this is the case, the path and name of the file should be specified.
After specifying the necessary input and output data, the button 'Load' should be pressed. The program now reads all measured moisture contents, precipitation data and (if required) irrigation data. When all data is read [7], a PageControl will be shown with four tabpages (see Figure 23).

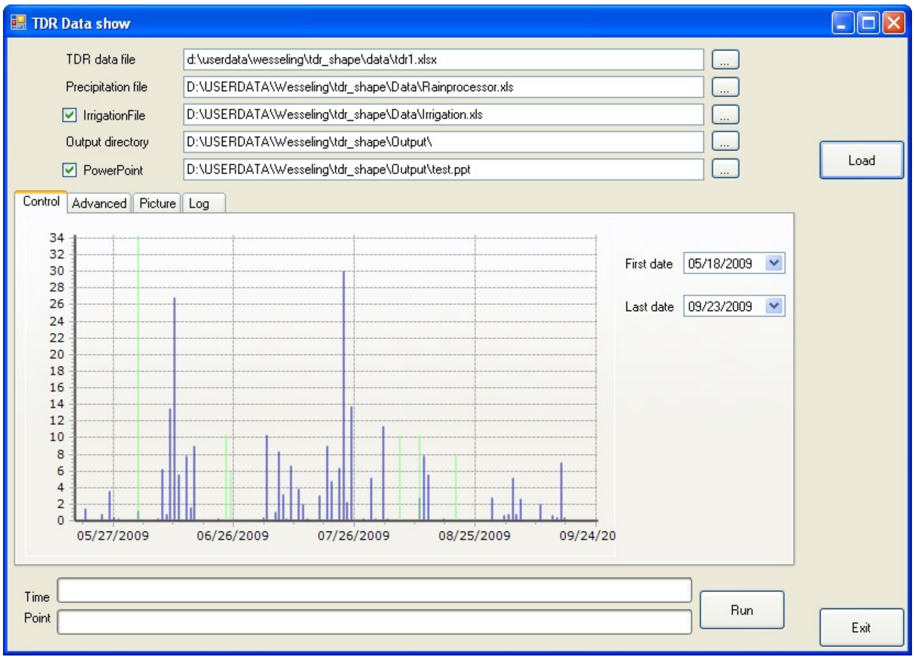


Figure 23: The screen after reading the input data.

The first tabsheet is called 'Control'. It shows a graph with the precipitation (blue) and irrigation data (green). The precipitation is given 4 times for each day, so it is in mm/6h. The irrigation is assumed to be an instantaneous supply and is shows only once a day. The irrigation and precipitation data are presented for the entire period during which soil moisture data is present. Ath the right side of the tabsheet two date-controls are shown. These controls are set to the first day and the last day with tdr-measurements. As a default the data of the entire period will be shown as an animation. If you only want part of the period, it is possible to set the first and last day of animation by selecting the required dates from the calendars that appear when the date-fields are clicked. The graph at the left side will be adapted according to the selected period. See Fig. 24

The second tabsheet is called 'Advanced' (Figure 25). Here the user may

[7]It may take some time before the program reacts after clicking the 'Load'-button. All data in the Excel-files is read into memory at once. Just be patient!

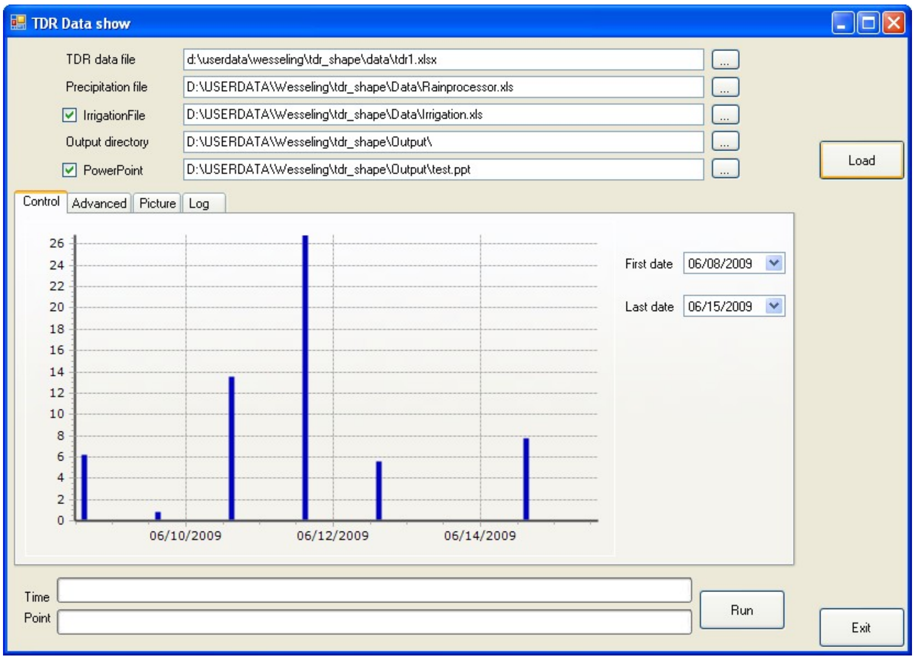


Figure 24: The screen after selecting a period of 1 week.

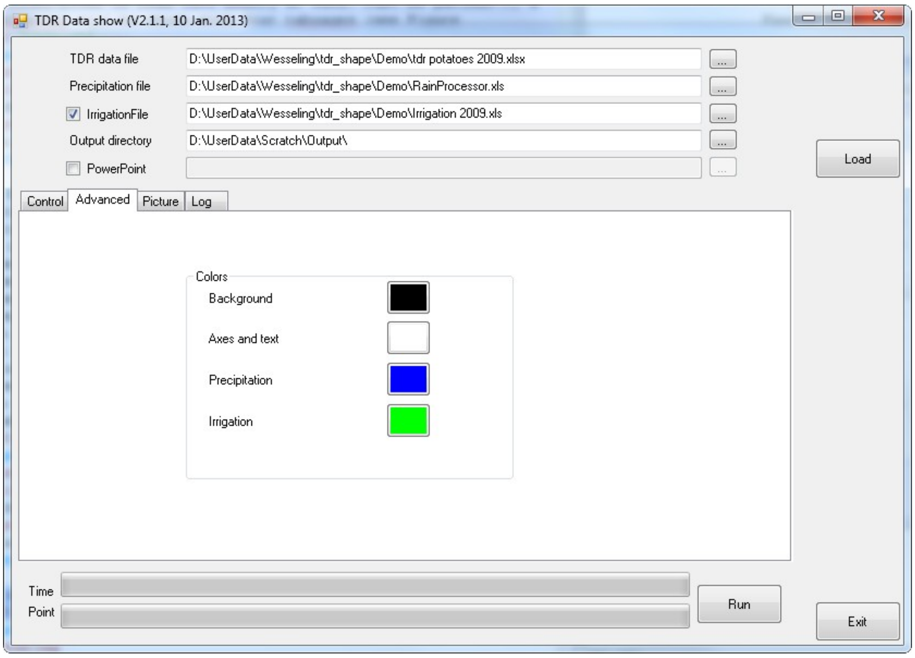


Figure 25: The page where the colors may be adjusted.

select his own colors to be used in the charts. Initially these colors are read from the file with tdr data, tabsheet 'Control'. The colors for background, text, precipitation and irrigation may be changed at runtime by clicking on the colored squares. Clicking these squares will open a color-selection dialog. The third tabsheet contains only one graphical box. This tabsheet may be selected when running the program, as it will show the last created image. See Fig. 26. Finally there is a tabsheet called 'Log'. Here the begin-time

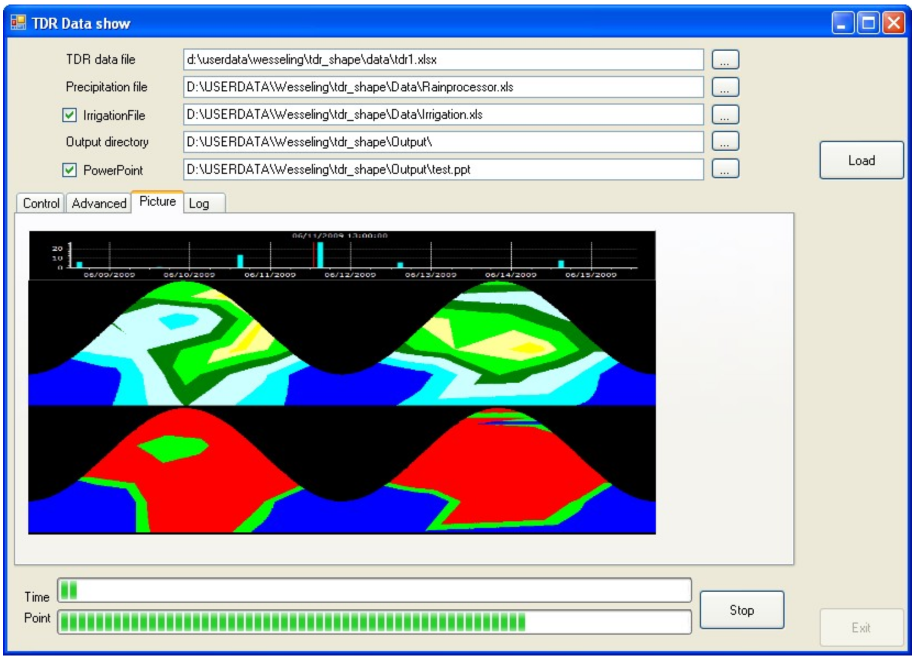


Figure 26: The screen while making the pictures.

and end-time of the creation process will be stored, as well as any error messages, if they occur.

At the bottom of the form two progress bars are visible. The upper one shows the overall progress. This is just the fraction of processed measuring moments over the total number of desired date/times to be processed. The lower progress bar shows the progress of the number of pixels in the profile processed. So if 4 plots are to be processed for each moment, this statusbar has to go from 0 to 100% 4 times before the upper status bar will be updated. At the right-hand side of the progress bars the 'Run'-button can be found. Clicking it will start the processing. The 'Exit'-button will be disabled when the program is running and the 'Run"-button wil change into a 'Stop'-button. In case the process should be stopped before it is finished, just press 'Stop'. The 'Run'-button will appear again. Processing will continue however, until the current picture has been created. Then the 'Exit'-button will be enabled as well. When processing is started, the 'Picture' tab can be selected to show the pictures on screen.

4.2 The output

The output of the program consists of a lot of files:

- A file with the grids applied in creating the pictures with moisture content distribution (grid.png).
- A file with the legend of the colors applied in showing the soil moisture content(legendm.png).
- A file with the legend of the colors applied in showing the derived values (legendd.png),
- Files with the 2-dimensional distribution of the moisture contents and/or the derived values for each required time (plot????.png).
- A PowerPoint file with the legend and all the files with moisture contents.

4.2.1 The grids

An example of the graph has been presented already in Figure 6 for the case of 1 plot to show. As another illustration we present the grids for the case of 4 plots in Figure 27. In this figure the elements are drawn in red, the node numbers are presented in white. Virtual nodes are presented in aqua, virtual elements are green. This file is not included in the PowerPoint presentation. It is meant as an aid for the user to check the construction of the grid.

4.3 The legend files

The colors corresponding to different moisture classes and derived values are specified in the Excel data file as described earlier in this report. When presenting the pictures to an audience, one should explain what the meaning of the different colors is. To assist in this, two charts have been made that can function as legends in the presentation (see Figures 28 and 29). They are included in the PowerPoint presentation.

4.4 The moisture content distributions

The program creates a graphical png-file with a moisture distribution and/or derived value for each moment of input in the Excel-input file that has a value between the selected first and last datetime. The filename starts with plot00001 and the numbers are subsequently increased. The output pictures

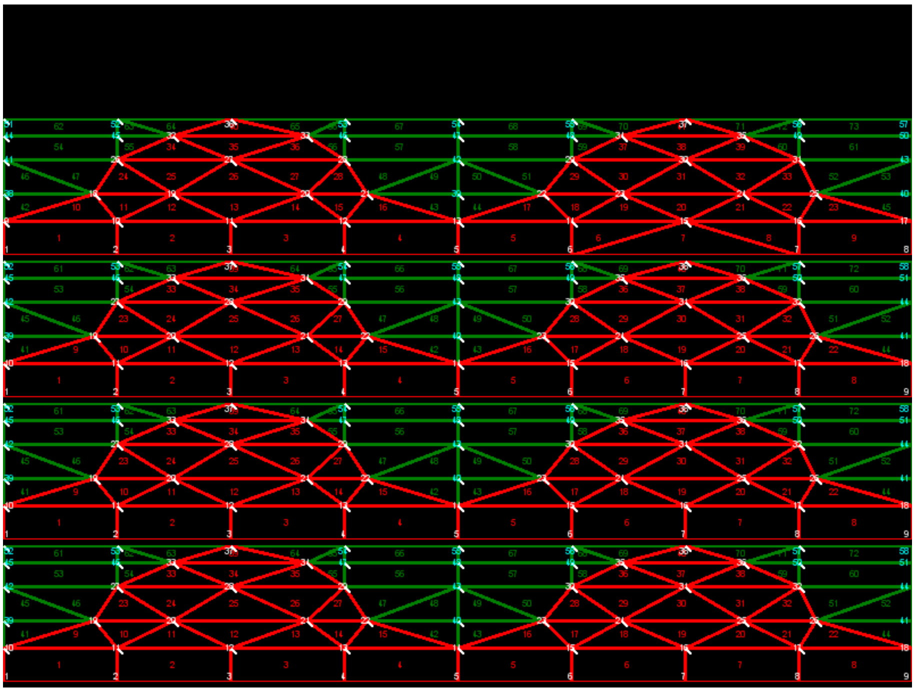


Figure 27: An example of the grids applied when 4 different plots are considered.

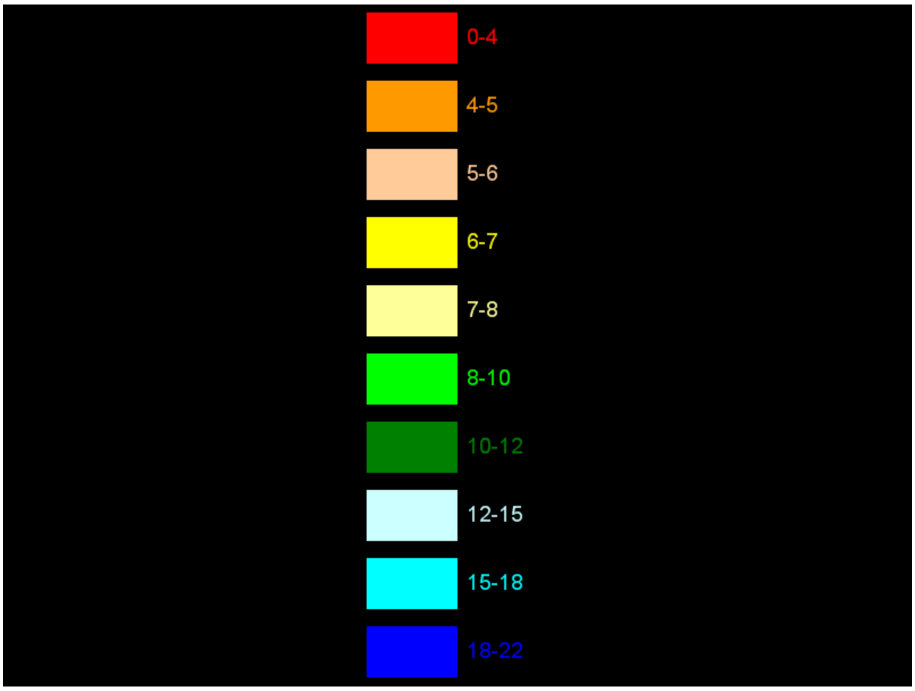


Figure 28: An example of the legend with colors applied to show the different moisture contents.

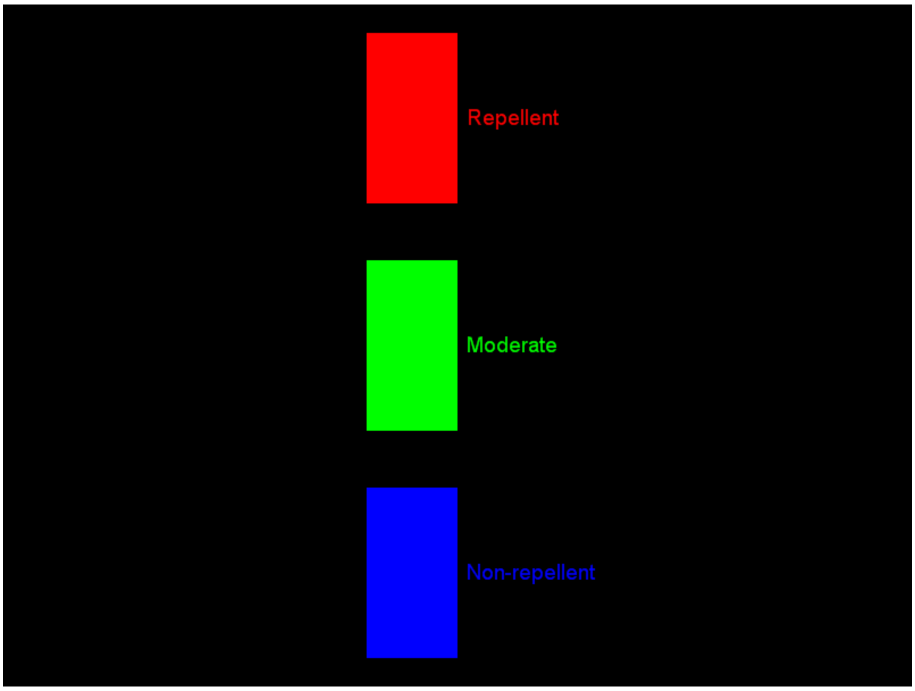


Figure 29: An example of the legend with classes of the derived values. In this case the derived value is the water repellency of the soil.

are the same as the ones discussed in the output section of the program. Here 4 examples will be presented with data for 1 (Figure 30), 2 (Figure 31), 3 (Figure 32) and 4 plots (Figure 33) respectively.

4.5 The PowerPoint file

If the option is selected by the user, all pictures with soil moisture distributions are placed in the PowerPoint file. The first pictures will be the legends. After creating all pictures, the program will close the PowerPoint file and save it under the name the user specified. Before showing the contents of the file, the user should set the slide transition to automatic and specify the time each slide should be shown as 0 seconds. Then PowerPoint will show the slides as fast as it can, thus creating some kind of a moisture content movie.

4.6 The ini-file

When processing the same input files for a number of times (e.g. when you are checking it for errors), it is convenient when the program remembers the settings you use. For that reason the ini-file was created. This file contains the settings the user has entered before clicking the 'Run'-button.

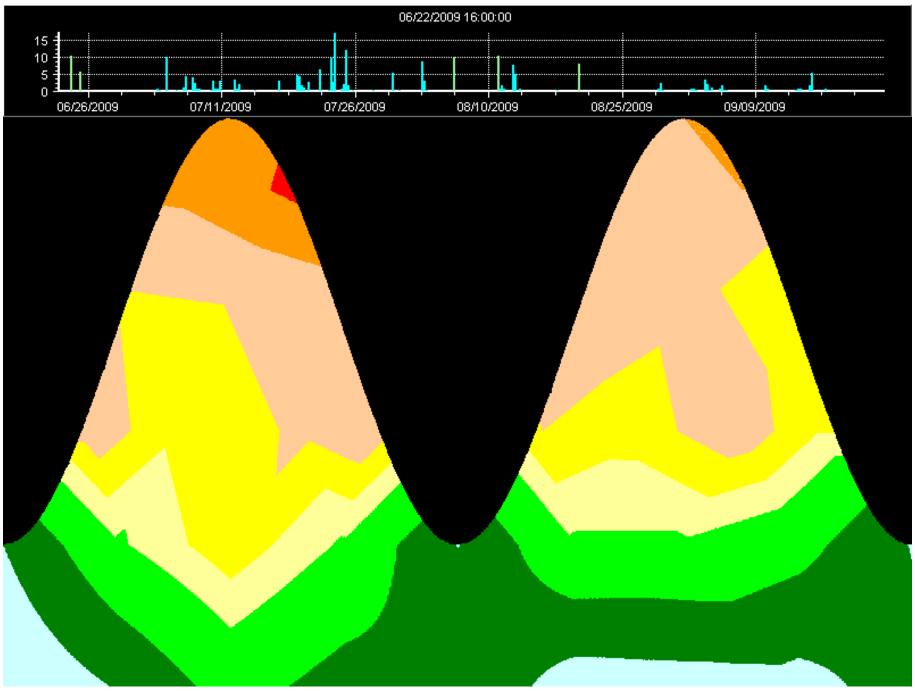


Figure 30: An example with moisture contents of 1 plot.

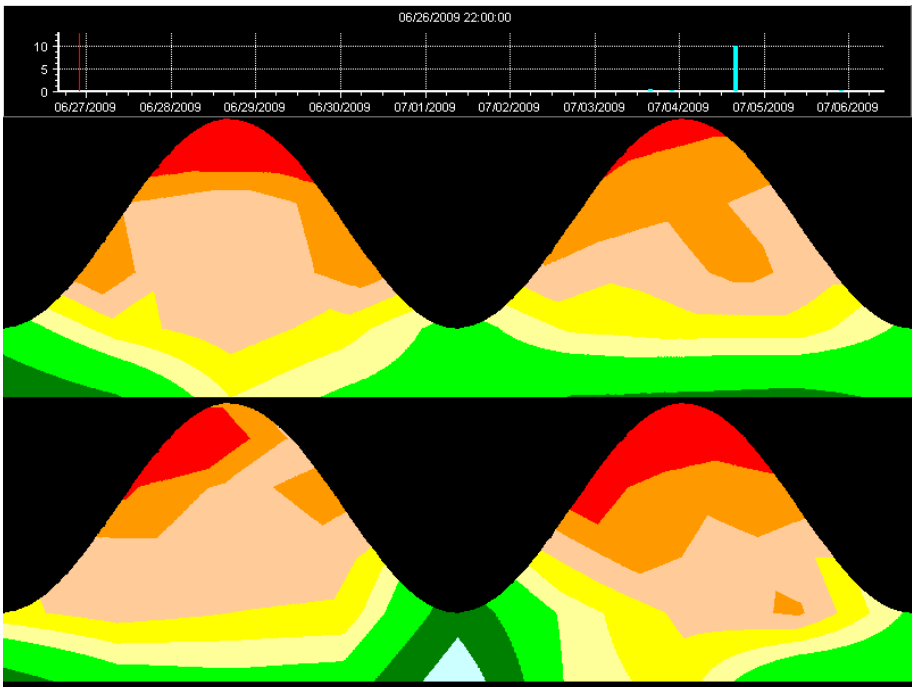


Figure 31: An example with moisture contents of 2 plots.

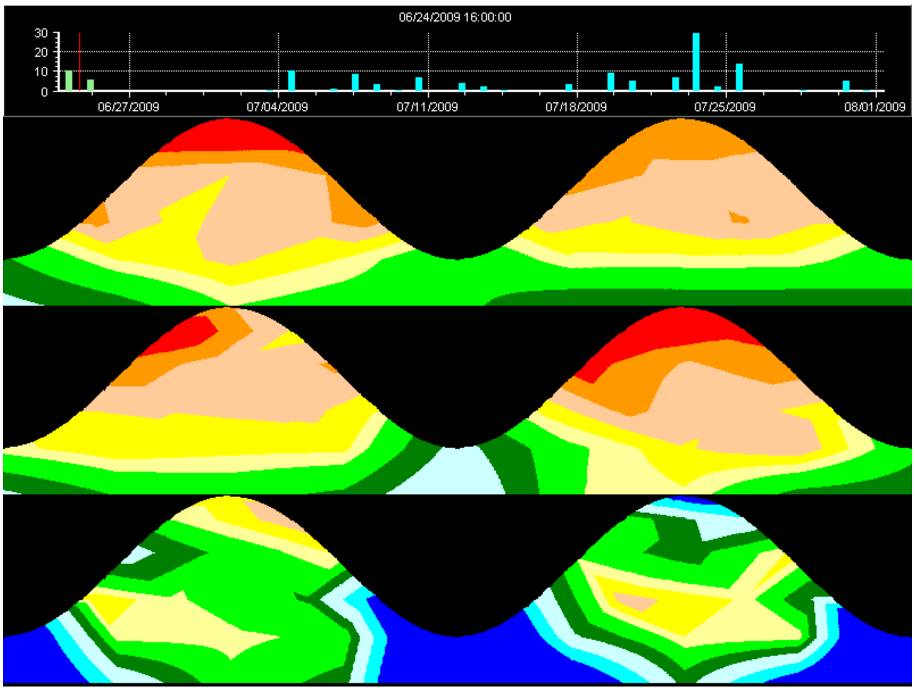


Figure 32: An example with moisture contents of 3 plots.

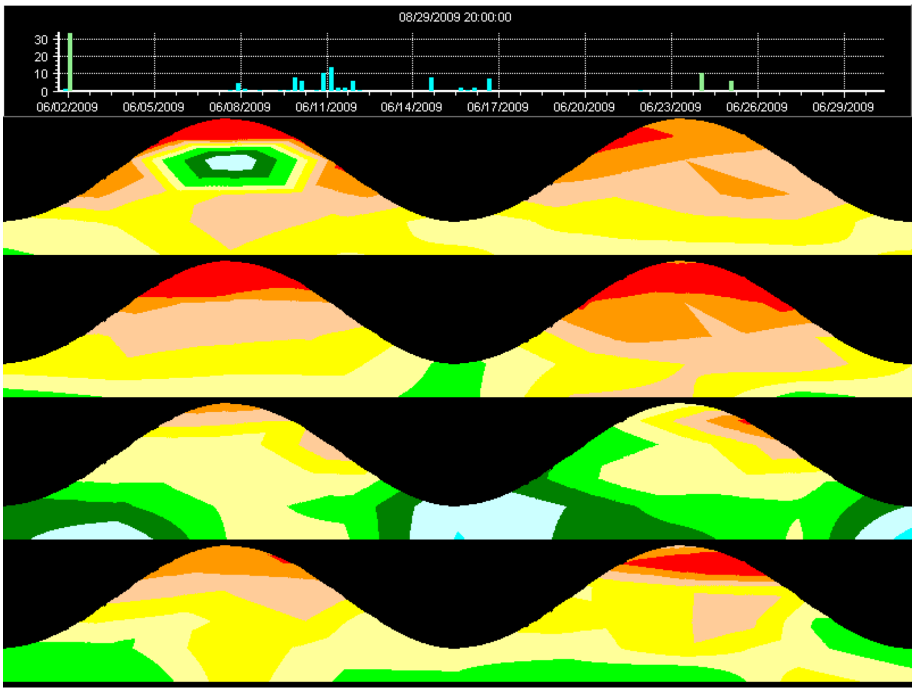


Figure 33: An example with moisture contents of 4 plots.

An example of the ini-file is presented below. As all fields are self-explaining, it wil not be discussed in more detail.

```
[Excel]
Moisture=D:\UserData\Wesseling\tdr_shape\Demo\tdr potatoes 2009.xlsx
Precipitation=D:\UserData\Wesseling\tdr_shape\Demo\RainProcessor.xls
Irrigation=D:\UserData\Wesseling\tdr_shape\Demo\Irrigation 2009.xls

[Output]
Dir=D:\UserData\Scratch\Output\

[Switch]
Irrigation=1
[PowerPoint]

Required=0
File=
```

5 System requirements

- Windows XP (servicepack 2 or higher), Windows Vista or Windows 7 as operating system.
- MicroSoft .Net, preferably Version 3.5 or higher (download free at www.microsoft.com)
- MicroSoft Office 2010. Though the program works with Office 2003 as well, the supplied dll's are based on 2010. We did not test it on Office 2007.
- An Intel Pentium processor (or compatible), at least 1.6 GHz. A slower processor would cause the program to take quite some time building the figures.
- Preferably 1 GB of memory or more.
- Sufficient disk space. One picture takes about 30 kB of spaces. This implies that the creation of the movie of a growing season of 150 days with a measuring frequency of 12 times a day will require about 54 MB of diskspace to store all pictures. If a PowerPoint presentation is required, a similar amount of additional disk space should be available.

6 Final Remarks

- In the present version of the program there is no check on input-values. It is completely left to the user to use data that is realistic and valid.
- It should be noted that the program writes its output files to the directory specified by the user without checking the existence of files. If there are files with the same name, these files will be overwritten without any notification.
- With a graphical conversion program like ImageMagick (freeware, see www.imagemagick.org) it is possible to convert the png-files to an animated gif-file, thus creating an animation suitable for publication on a website.

References

[1] P. Bourke. Determining if a point lies on the interior of a polygon, 2010.

[2] L.W. Dekker, C.J. Ritsema, K. Oostindie, D. Moore, and J.G. Wesseling. Methods for determining soil water repellency on field-moist samples. *Water Resources Research*, 45, 2009.

[3] D.N. Graham and M.B.Butts. *Flexible, integrated watershed modelling with MIKE SHE.* In: Watershed Models, Eds. V.P. Singh and D.K. Frevert Pages 245-272. CRC Press. ISBN: 0849336090, 2005.

[4] T.J.R. Hughes. *The finite element method. Linear static and dynamic finite element analysis.* Prentice-Hall Inc., Englewood Cliffs, N.J.07632, 1987.

[5] J. Leven, J.J. Corso, J.D. Cohen, and S. Kumar. Interactive visualization of unstructured grids using hierarchical 3d textures. In *Proceedings of IEEE/SIGGRAPH Symposium on Volume Visualization and Graphics 2002, pages 37-44.*

[6] M. Sejna, J. Simunek, and M.Th. van Genuchten. The Hydrus software package for simulating two- and three-dimensional movement of water, heat and multiple solutes in variably-saturated media. user manual, version 2.0. Technical report, PC Progress, Prague, Czech Republic, 2011.

[7] J. Simunek, M.Th. van Genuchten, and M. Sejna. The hydrus software package for simulating two- and three-dimensional movement of water, heat and multiple solutes in variably-saturated media. technical manual, version 2.0. Technical report, PC Progress, Prague, Czech Republic, 2011.

[8] J. G. Wesseling. *Soil physical data and modeling soil moisure flow.* PhD thesis, Wageningen University, 2009.

[9] J. G. Wesseling, K. Oostindie, L. W. Dekker, and H. G. M. Van den Elsen. Animating measured precipitation and soil moisture data. *Computers and Geoscience*, 34:658–666, 2008.

[10] J.G. Wesseling. Animating measured precipitation and soil moisture data. Technical report, Alterra Wageningen, 2005.

[11] C.M. Wittenbrink. *IFS Fractal interpolation for 2D and 3D Visualization.* Published in: VIS95 Proceedings of the 6th conference on Visualization. IEEE Computer Society Washington, DC, USA., 1995.