University of Natural Resources
and Life Sciences, Vienna

Department of Water, Atmosphere, Environment
Institute of Hydraulics and Rural Water Management

Development of an Automatic Calibration Tool for the Soil Water and Nitrogen Balance Model SIMWASER/STOTRASIM by Coupling it with the Multiparameter Optimization Method AMALGAM

Master's Thesis
for attaining the academic degree of 'Diplomingenieurin'
at the University of Natural Resources and Life Sciences, Vienna

Submitted by
Lisa Huber

Supervisors: Univ.Prof. Dipl.-Ing. Dr. Willibald Loiskandl
Dipl.-Ing. Dr. Peter Strauß

Vienna, December 2016

Acknowledgements

I want to thank my supervisor Prof. WILLIBALD LOISKANDL for the trust and the freedom he allowed me during the project's evolution.

I want to convey my deepest gratitude to the INSTITUTE FOR LAND AND WATER MANAGEMENT RESEARCH. Not only for their scientific support, but also for the many friendships I could establish during the time in Petzenkirchen. It has been a memorable time with you! My special thanks and recognition goes to the head of the institute and my co-supervisor Dr. PETER STRAUSS as well as to Dipl.-Ing. ALEXANDER EDER for their unending support. I sincerely thank Dipl.-Ing. FRANZ FEICHTINGER for introducing me to the world of the simulation program SIMWASER/STOTRASIM.

I'd also like to express my gratitude to the partnering organization JOANNEUM RESEARCH. They provided data for the test site 'Wagna' and informative exchanges about the application of SIMWASER/STOTRASIM.

Finally I would like to thank my loved ones, my family and friends who gave me support throughout my studies and never let me down.

Abstract

The application of soil water and nitrogen balance model SIMWASER/STOTRASIM (Stenitzer, 1988; Feichtinger, 1998) requires detailed knowledge of the water retention and unsaturated soil hydraulic conductivity properties of the considered soil profile as well as plant characteristics of the considered crop rotation. Before the model can be used for scenario testing or prediction, it needs to be calibrated, which means the adjustment of the input parameters to match the model to measured data. The objective of this master thesis was the development of a tool for the automatic calibration of SIMWASER/STOTRASIM. For this purpose, the model written in C# was coupled with the Multiparameter Optimization Tool AMALGAM, coded in MatLab. The task was performed by developing systems of data transfer between MatLab and C#, working out an alternative for the representation of the hydraulic conductivity and water retention characteristics with model parameters, defining objective functions, setting starting values and ranges and developing simple structures for data input and result output. The software package is called Calisto (‘Calibration Stotrasim’). Application tests show that the software is executable and can be used for the calibration of the parameters ‘*potfak*’ (mineralization multiplier), ‘*rs*’ (evaporation coefficient) and for the parameters that describe plant development. The occurrence of two systematic errors, however, impeded optimization of the soil parameters. A solution to fix these problems would require further studies.

Keywords: simulation model, automatic calibration, parameter optimization, STOTRASIM

Zusammenfassung

Zur Anwendung des Bodenwasserhaushalt- und Stickstofftransportmodells SIMWASER/STOTRASIM (Stenitzer, 1988; Feichtinger, 1998) sind Charakterisierungen der Wasserretentionsfunktion und der ungesättigten hydraulischen Leitfähigkeit des betrachteten Bodenprofils sowie des Wachstumsverhaltens der betrachteten Fruchtfolge notwendig. Bevor Testszenarien oder Vorhersagen mit dem Simulationsmodell berechnet werden können, ist eine standorts- und fruchtfolgespezifische Modellkalibrierung notwendig. Das Ziel dieser Masterarbeit war die Entwicklung eines Werkzeuges zur automatischen Kalibrierung von SIMWASER/STOTRASIM. SIMWASER/STOTRASIM, programmiert in C#, wurde mit dem Multiparameter Optimierungsalgorithmus AMALGAM gekoppelt, welches als MatLab-Code zur Verfügung stand. Für die Optimierungssoftware Calisto ("Calibration Stotrasim") waren die Entwicklung von Datenaustauschsystemen zwischen C# und Matlab, die Darstellung der Wasserretentionskurve und der Leitfähigkeitskurve mithilfe von Modellparametern, die Formulierung von Zielfunktionen, die Definition von Start- und Grenzwerten für die zu optimierenden Parameter und die Entwicklung von einfachen Strukturen zum Dateninput und der Ergebniszusammenfassung und -darstellung notwendig. Anwendungstests konnten beweisen, dass die Software lauffähig ist und zur Kalibrierung des Mineralisierungsmultiplikators „*potfak*", des Evaporationskoeffizienten „*rs*" und der Pflanzenparameter verwendet werden kann. Die Kalibrierung der Bodenparameter wird jedoch in gehäuftem Maße durch das Auftreten zweier systematischer Fehler behindert. Die Lösung der Probleme würde weitere Studien und eventuelle grundlegende Eingriffe in die Modellformulierung von SIMWASER/STOTRASIM erfordern.

Stichworte: Simulationsmodell, automatische Kalibrierung, Parameteroptimierung, STOTRASIM

Table of Contents

1. Introduction and Objectives 1

2. Theory and Material 4

2.1 SIMWASER/STOTRASIM 4

2.1.1 Model concept 4

2.1.2 Model implementation 9

2.1.3 Model inputs and outputs 10

2.2 AMALGAM 12

2.2.1 Method description 12

2.2.2 AMALGAM as MatLab tool 14

2.3 Models for Water Retention and Hydraulic Conductivity 15

2.4 Measures of goodness-of-fit 17

2.5 The Programming Languages and Development Environments 18

3. Calisto – The Calibration Program 21

3.1 Calling C# with MatLab 22

3.2 pF and K_u Curves 24

3.3 Calisto Inputs 27

3.3.1 Excel file 'Parametrisierung.xls' 27

3.3.2 Pop up windows 32

3.4 Starting Values and Ranges 33

3.5 Data Transfer between MatLab and C# 36

3.5.1 Transfer of parameters from MatLab to C# 36

3.5.2 Transfer of model outputs from C# to MatLab 39

3.6 Objective Functions 41

3.7 Calisto Results and Output 42

4. Application of Calisto 48

4.1 Setting up a test site 48

4.2 System Testing on Generated Data 57

4.2.1 Optimization of parameters 'potfak' and 'rs' 57

4.2.2 Optimization of plant parameters 59

4.2.3 Optimization of soil parameters 64

5. Discussion 73

6. Conclusion 78

7. References 79

8. List of Figures 86

9. List of Tables 89

1. Introduction and Objectives

In times of rising public and political awareness of environmental problems, environmental simulation models are an indispensable instrument for research, management and decision-making. Modelling is used across many environmental fields: hydrology, air pollution, ecology, hazard assessment and climate dynamics, to name a few. Environmental simulation tools reconstruct nature events based on mathematical equations. The models are usually set up with preliminary surveys under natural conditions and can then be used for scenario testing and - with limitations - for extrapolation and prediction. It is much easier, time-saving, practical and less risky to create computer models to run certain experiments than to conduct them in the field (Peng, 2002).

Soil water and nitrogen balance models are instruments to represent and simulate correlations of soil water movement, nitrogen transport and the related crop yields. They deliver insight into relevant soil and plant processes and can therefore be a tool for exploring best agricultural practices with regard to agronomical optimization as well as to soil and groundwater protection. The capability of extrapolation and prediction theoretically even gives them power to assess the impacts of future land and climate changes. For more than 30 years simulation models have been developed and applied in the research on nitrate leaching (Groenendijk et al., 2014; Eder et al., 2016). Well-known examples for soil water and nitrogen balance models are DAISY (Hansen et al., 1990), ANIMO (Rijtema and Kroes, 1991), SACFARM (Addiscott et al., 1991), the model package DSSAT (Jones et al., 1998; Jones et al., 2003), EPIC (Williams et al., 1984; Williams et al., 1989) and SIMWASER/STOTRASIM (Stenitzer, 1988; Feichtinger, 1998). The Institute for Land and Water Management Research (IKT) from the Federal Agency for Water Management has been using the physically based model SIMWASER/STOTRASIM for decades to assess nitrogen cycles on arable land.

Accurate modeling of the vadose zone flow and transport processes usually requires detailed knowledge of water retention and unsaturated soil hydraulic conductivity properties of the considered soil profile. The soil property parameters needed for SIMWASER/STOTRASIM are usually determined with field measurements or experiments on undisturbed soil samples in the laboratory or estimated from particle size distributions using pedotransfer functions. But there are several difficulties arising when it comes to the use of hydraulic properties in models. Numerous studies have shown that the measured soil hydraulic parameters estimated from laboratory or in-situ point observations may not be adequate or representative for a larger spatial domain (Mertens et al., 2005; Mertens et al., 2006;

Woehling et al., 2008). Talking about hydraulic conductivity values for example, Mallants et al. (1997) showed that values measured in the laboratory with open ended columns often increase with decreasing column length because at smaller scales greater pore continuity can occur. The components of a model are rather conceptual descriptions of real processes. As a consequence, the model parameters do not equal to physical parameters or they are technically not measurable (Gupta et al, 2005). Moreover, the simulation by plant-soil-models needs plant parameters, which describe the development of the specific crops. In the first version of SIMWASER, Stenitzer (1988) fixed data sets for winter crop, summer crop, maize, sugar beet and clover-grass mixtures. For the simulation of further plant species, more data have to be collected. Partly, the necessary input parameters can be derived from literature or taken from physiologically similar plants, but sometimes they lack to adequately predict the individual plant growth and its impact on water and nutrient transport.

Overall it can be concluded that the parameters needed for model simulations often cannot be derived from literature or measured directly. The models rather have to be calibrated, which means the adjustment of the parameters to match the model to measured data (Mertens et al, 2005; Mertens et al., 2006). Because of the lack of automatic software, the calibration of SIMWASER/STOTRASIM has always been practiced by visual inspection and a manual trial-and-error procedure. First, estimates of the parameters from laboratory experiments and from literature were used as starting values. The simulation run was executed with these starting values and the output was visually compared with measured time series. Occurring deviations were corrected by manual adjustment of soil parameters as well as of plant parameters. The disadvantage of a manual calibration like this is being subjective, tedious and time consuming (Mertens et al., 2005; Ndiritu, 2009). Alternatives may be found in automatic calibration methods in form of software, which are relying on systematic search approaches. They are set up to find the best fitting parameter values according to predefined objective functions (Woehling et al., 2008; Kamali et al., 2012). In contrast to manual calibration procedures, automatic calibration methods are faster, reproducible and objective in theory. As calibration of a model is seen as ‘the key factor for a successful application’ (Kamali et al., 2012), the establishment of an automatic calibration tool for SIMWASER/STOTRASIM is desirable.

In former times automated calibration methods have focused mainly on the selection of a single-objective measure of a deviation between model-output and measured data (Yapo et al., 1998; Woehling et al., 2008). The use of a single measure, however, is often inadequate to properly cover all the important characteristics of the system (Yapo et al., 1998). An alternative is the use of multiple objectives in the optimization problem. The existence of multiple objectives leads to a set of Pareto-optimal solutions instead of a single optimal solution (Vrugt and Robinson, 2007). These ‘good’ parameter combinations can be distributed all over the parameter space. Nonlinear models - like most of the soil water and nitrogen balance models are - often have multimodal response surfaces, which means that there are several locations in parameter space where the values of the functions are equally good (Peng, 2002). The phenomenon is called ‘equifinality’ (Beven and Binley, 1992). Evolutionary algorithms have proved themselves as very powerful approaches to deal with these special features of multiobjective models (Mertens et al, 2005; Gupta and Sorooshian, 1998; Vrugt and Robinson, 2007). Evolutionary algorithms are able to search large spaces for Pareto-optimal solutions, to maintain a diverse set of solutions and exploit similarities of solutions by recombination in a single optimization run (Vrugt and Robinson, 2007). However, while currently available evolutionary algorithms typically implement only one algorithm for population evolution, theory has proven that it is impossible to develop a single algorithm that is always efficient for different types of optimization problems (Wolpert and Macready, 1997).

AMALGAM (Vrugt and Robinson, 2007) is an optimization tool which is suitable for solving optimization problems with multiple conflicting objectives and is also able to search large parameter spaces to find optimal parameter sets all over the search area. Moreover AMALGAM combines two concepts, the simultaneous use of several optimization methods and self-adaptive offspring creation. According to the specific difficulties and peculiarities of the particular optimization problem, the method automatically changes preferences to individual search algorithms during the course of the optimization run. So the method merges the strengths of different search strategies to increase the speed of evolution (Vrugt and Robinson, 2007). AMALGAM seems to be a strong optimization tool which could help optimizing the input parameters for SIMWASER/STOTRASIM.

The main objectives of the master project are

- to implement a tool for the automatic calibration of ‘STOTRASIM C#’
- to couple SIMWASER/STOTRASIM with the Multiparameter Calibration Algorithm AMALGAM
- to test the calibration tool for its functionality

2. Theory and Material

2.1 SIMWASER/STOTRASIM

2.1.1 Model concept

The deterministic model SIMWASER/STOTRASIM (Stenitzer, 1988; Stenitzer, 2007 and Feichtinger, 1998) describes one-dimensional vertical flow of water and nitrate-nitrogen within a soil profile, neglecting interflow and preferential flow. Soil water fluxes and plant growth are calculated by the sub-model SIMWASER (Stenitzer, 1988), the nitrogen balance is calculated by the sub-model STOTRASIM (Feichtinger, 1998). In the model, plant growth is limited by water and/or nitrogen availability. The response of plant growth to the supply with other nutrients and to pests is not taken into account.

SIMWASER simulates the daily water balance and the crop yield of any number of crop rotations and years. The water balance and the plant growth are interrelated through the physiological interaction of assimilation and transpiration. For the calculation of plant development and growth as well as of the associated water consumption the various cultivated plants need to be characterized. The upper boundary condition for the water balance is given by precipitation, transpiration and evaporation, whereby evaporation and transpiration are calculated as a function of the stage of plant development, air saturation and water availability in the soil. Interception is considered, too. Additional to the plant parameters, also soil layering, thickness, water retention characteristics and hydraulic conductivity of each layer have to be known. The relevant processes are infiltration of precipitation, capillary rise and plant uptake (Figure 1) (Stenitzer, 1988).

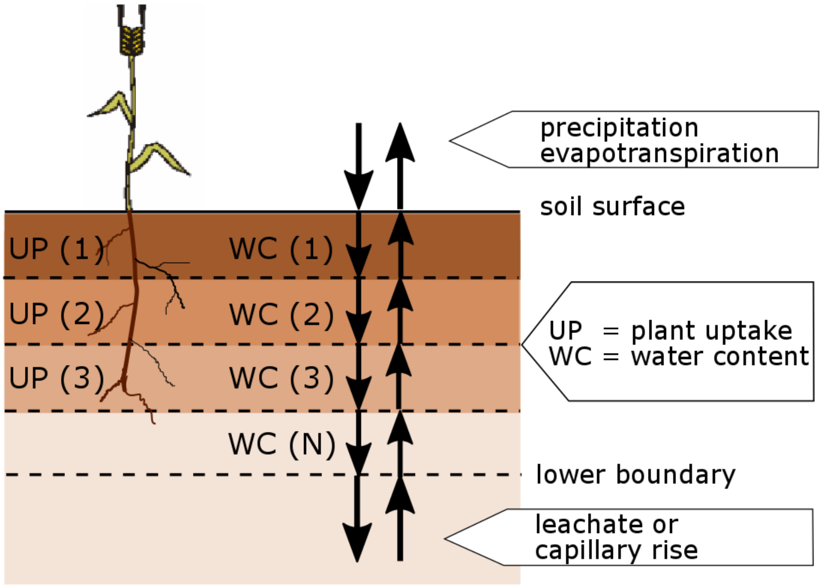


Figure 1: Factors and parameters of water balance considered in SIMWASER; source: adapted from Stenitzer (1988)

The lower boundary of the soil profile is given by the groundwater surface or set in a sufficient depth to exclude root activity. The calculation of the water movement between soil layers (compartments) is based on Richards equation (Richards, 1931). At first, the filtration velocity for the predefined starting water content distribution is calculated:

$$V_i = \frac{(K_i + K_{i+1})}{2} * (\frac{\psi_{i+1} - \psi_i}{z_i} + 1) \tag{1}$$

whereby V_i is the filtration velocity, K_i and K_{i+1} is the hydraulic conductivity in soil compartments i and $i+1$, ψ_i and ψ_{i+1} is the matric potential in soil compartments i and $i+1$ and z_i is the distance from the compartment midpoint of i to the compartment midpoint of $i + 1$.

Afterwards, a time step Δt is calculated that prevents the change of the water content from exceeding 0.1 percent by volume. For the selected Δt the water movement on the top edge and the bottom edge (q_o and q_u) of all soil compartments has to be calculated. q_o of the upper compartment i_1 is determined by:

$$q_o(i_1) = (N - E_{akt}) * \Delta t \tag{2}$$

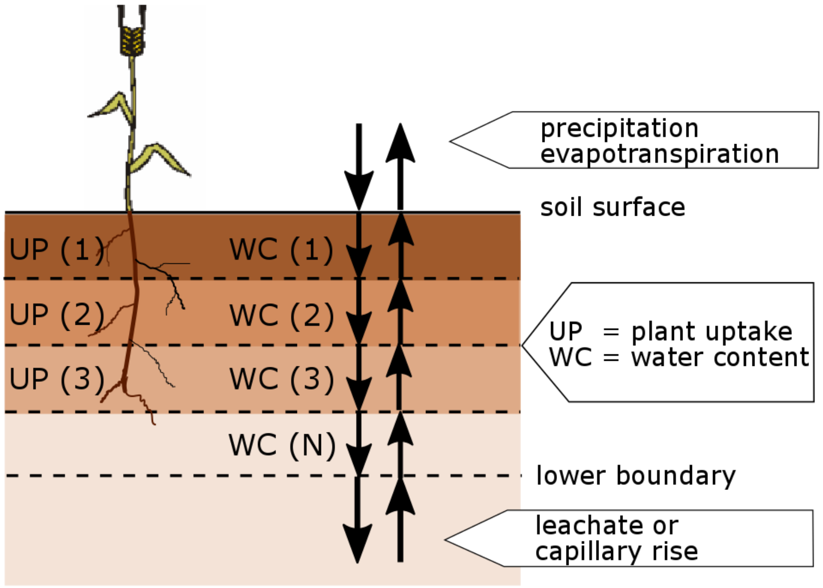


Figure 1: Factors and parameters of water balance considered in SIMWASER; source: adapted from Stenitzer (1988)

The lower boundary of the soil profile is given by the groundwater surface or set in a sufficient depth to exclude root activity. The calculation of the water movement between soil layers (compartments) is based on Richards equation (Richards, 1931). At first, the filtration velocity for the predefined starting water content distribution is calculated:

$$V_i = \frac{(K_i + K_{i+1})}{2} * (\frac{\psi_{i+1} - \psi_i}{z_i} + 1)$$

(1)

whereby V_i is the filtration velocity, K_i and K_{i+1} is the hydraulic conductivity in soil compartments i and $i + 1$, ψ_i and ψ_{i+1} is the matric potential in soil compartments i and $i + 1$ and z_i is the distance from the compartment midpoint of i to the compartment midpoint of $i + 1$.

Afterwards, a time step Δt is calculated that prevents the change of the water content from exceeding 0.1 percent by volume. For the selected Δt the water movement on the top edge and the bottom edge (q_o and q_u) of all soil compartments has to be calculated. q_o of the upper compartment i_1 is determined by:

$$q_o(i_1) = (N - E_{akt}) * \Delta t$$

(2)

whereby N is the precipitation and E_{akt} is the actual evapotranspiration. The subtraction of them represents the water balance at the site. q_o of deeper compartments is evaluated by

$$q_o(i) = q_u(i-1) \tag{3}$$

The water movement at the bottom edge q_u of the first layer, as well as of the other layers, can be calculated by

$$q_u(i) = V_i * \Delta t \tag{4}$$

The 'new' water content at the end of the time step is subsequently defined by

$$W_{i,new} = W_{i,before} + \frac{(q_o(i) - q_u(i))}{h_i} \tag{5}$$

whereby h_i is the thickness of the layer i . For the 'new' water content of each compartment, a corresponding matric potential and hydraulic conductivity can be assigned according to the predefined characteristics of each soil layer. With the new values for water content, matric potential and hydraulic conductivity, the process can be repeated for the next time step starting from equation 1 (Stenitzer, 1988).

The water flux at the lower boundary of the soil profile may either be free groundwater recharge or capillary rise for sites with groundwater connection or free drainage for sites without groundwater connection (Stenitzer, 1988).

A more detailed description of the model characteristics can be found in Stenitzer (1988) and Stenitzer (2007). Here also the equations for the plant development, assimilation, evaporation and transpiration can be found. Only one more specific formula should be mentioned exclusively at this point, as it contains one parameter that plays an important role in the model setup and calibration: The potential evapotranspiration is based on Penman-Monteith (Monteith, 1965) and defined by

$$E_p = \frac{f_t Q + 0.864\, H_0/r_a}{f_t + 1 + (r_s + r_p)/r_a} \tag{6}$$

whereby E_p is the potential evapotranspiration, f_t is the temperature factor, Q is net energy (radiation balance), 0.864 is a dimension factor, H_0 is the saturation deficit of the air, r_a is the aerodynamic resistance, r_p is the stomatal resistance and r_s is the resistance of the soil surface. In literature, r_p and r_s are often summarized in the term '(bulk) surface resistance'. The soil evaporation coefficient r_s can be defined by the user when starting the simulation model or can be calculated by the program during the application and it has decisive influence on the evaporation of the soil, especially of the upper soil layers.

The sub-model STOTRASIM extends SIMWASER for nitrogen dynamics. Fertilization, precipitation, irrigation and nitrogen fixation from the atmosphere are considered as nitrogen inputs. Denitrification and ammonium volatilization are nitrogen outputs. Nitrogen uptake by plants and return of plant material (leaves, straw, roots) are also taken into account. Nitrogen turnover happens in the form of mineralization, nitrification, immobilization and denitrification (Figure 2).

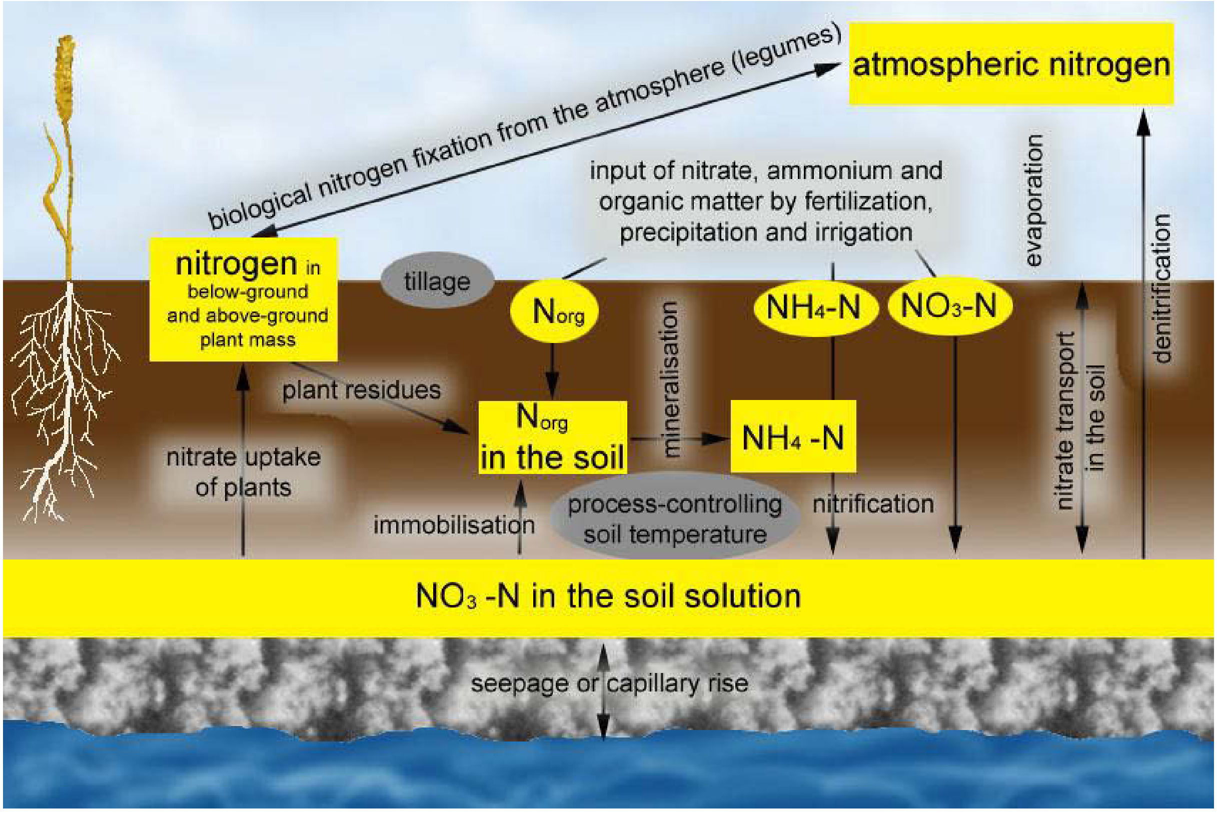

biological nitrogen fixation from the atmosphere (legumes)
atmospheric nitrogen
input of nitrate, ammonium and organic matter by fertilization, precipitation and irrigation
evaporation
denitrification
nitrogen in below-ground and above-ground plant mass
tillage
N_{org}
mineralisation
NH_4-N
NO_3-N
plant residues
N_{org} in the soil
NH_4 -N
nitrate uptake of plants
nitrate transport in the soil
immobilisation
process-controlling soil temperature
nitrification
NO_3 -N in the soil solution
seepage or capillary rise

Figure 2: Processes and factors of the nitrogen cycle considered in STOTRASIM; source: adapted from Feichtinger (1998)

According to Franko (1997) the organic nitrogen stock in the soil is subdivided into four fractions with different decay rates (Figure 3). STOTRASIM assumes that the nitrogen transport in the soil (linked to the water transport) only takes place in form of nitrate. For each soil compartment the following organic nitrogen pools are considered:

- FOS (fresh organic substance): Includes the organic import in the soil (plant residues, mineral fertilizer and compost), which are broken down in a few months.
- AOS (active organic substance): This soil-born fraction is fast decomposable and mineralized within a few years. There can be soil organisms living in it.
- SOS (stabilized organic substance): This soil-born fraction is slowly decomposable and mineralized within a few decades.
- IOS (inert organic substance): This soil-born fraction is inert and its mineralization takes place within several centuries or even millennia.

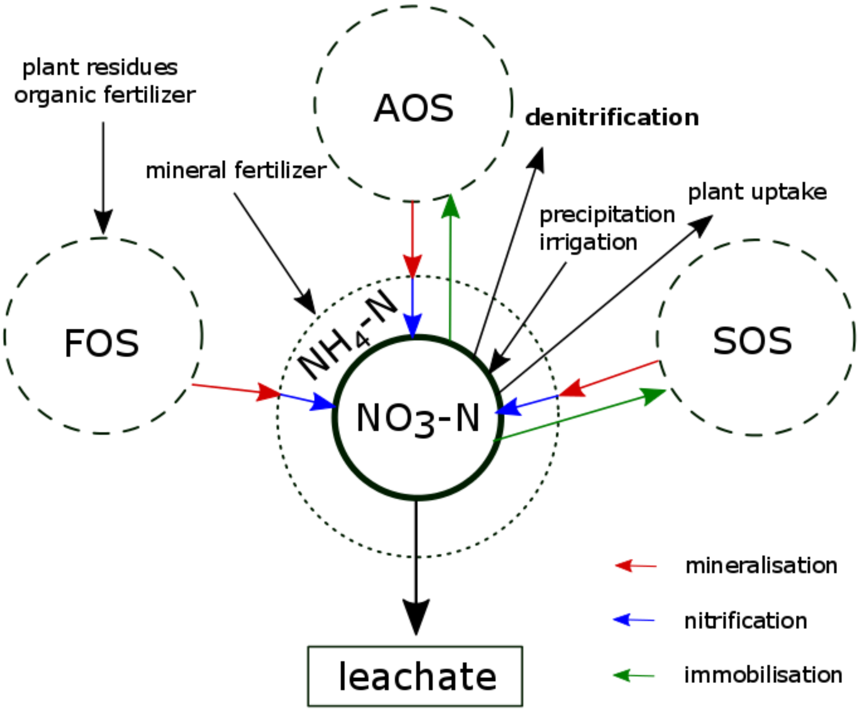


Figure 3: Overview of the nitrogen fractions considered in STOTRASIM

Mineralisation of humus is heavily influenced by the variable '*potfak*'. The mineralization multiplier '*potfak*' can be defined by the user when starting the simulation model or can be determined by the program itself. The nitrate-nitrogen is transported by advection and diffusion/dispersion. According to the potential gradient either nitrogen leaching or capillary rise takes place. The main focus of this model is set on the leachate into groundwater. For a more detailed description of the model characteristics I refer to Feichtinger (1998).

2.1.2 Model implementation

SIMWASER/STOTRASIM had originally been programmed in the higher-level language Fortran 77. In 2013 the model was reengineered in C# (Hobisch, 2014).

The program starts with reading in the project file, which includes configuration parameters (including '*rs*' and '*potfak*', 2.1.1) and the variants that should be simulated, also called runs or projects. Afterwards the first loop over the variants is started, followed by the crop rotation loop, the day loop and the time step loop. The length of the time step cannot be defined by the user, but is chosen by the program according to the flow velocity in the soil. An overview of the program process is given in Figure 4.

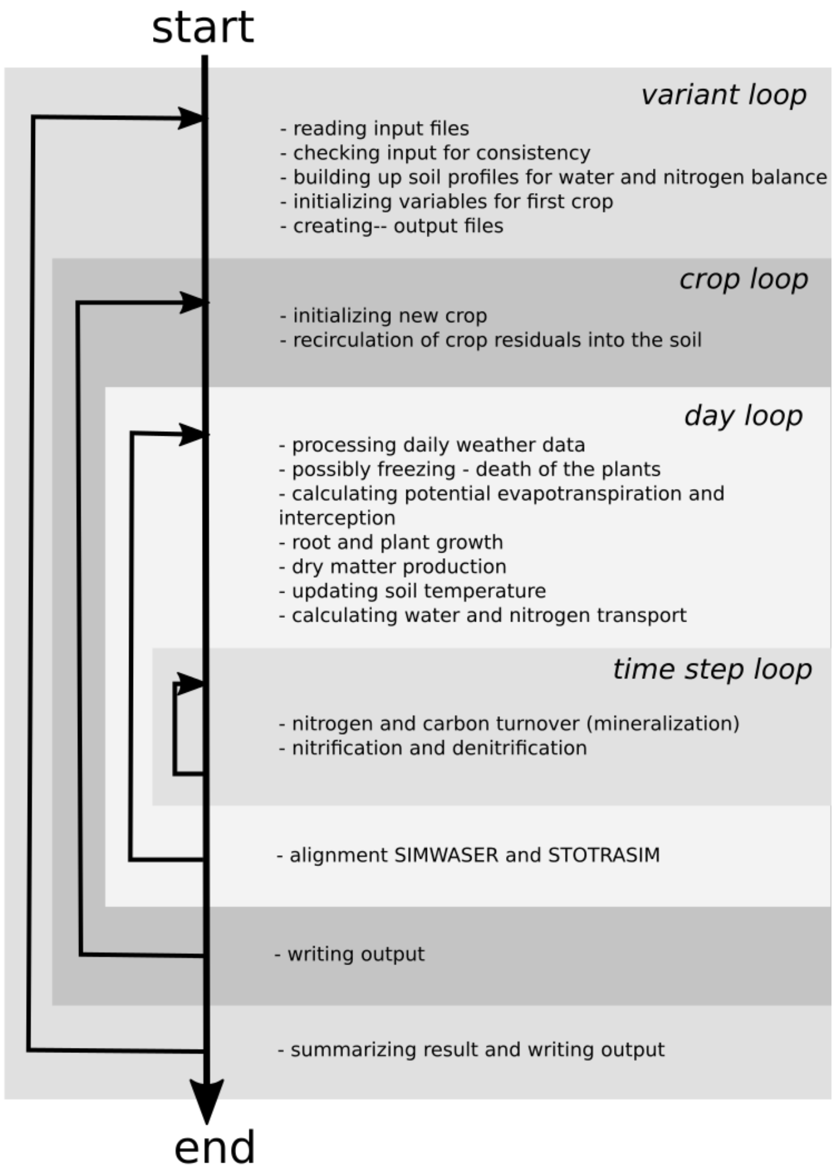

start
variant loop
- reading input files
- checking input for consistency
- building up soil profiles for water and nitrogen balance
- initializing variables for first crop
- creating-- output files
crop loop
- initializing new crop
- recirculation of crop residuals into the soil
day loop
- processing daily weather data
- possibly freezing - death of the plants
- calculating potential evapotranspiration and interception
- root and plant growth
- dry matter production
- updating soil temperature
- calculating water and nitrogen transport
time step loop
- nitrogen and carbon turnover (mineralization)
- nitrification and denitrification
- alignment SIMWASER and STOTRASIM
- writing output
- summarizing result and writing output
end

Figure 4: Overview of the program process cycle SIMWASER/STOTRASIM; source: adapted from Hobisch (2014)

2.1.3 Model inputs and outputs

The required model inputs for SIMWASER/STOTRASIM C# can be grouped into

1) General configurations
2) Soil information
3) Plant parameters
4) Climatic conditions
5) Land management practices including fertilization

This information has to be provided as binary respectively ASCII-files or .csv-files in a folder called 'Input' for being read in by the simulation model:

1)

- 'Project file': gives the user the opportunity to choose program options and specify output; also '*rs*' and '*potfak*' (2.1.1) should be defined here

2)

- 'Soil profile file' called 'BODPROF.csv': defines the layering of the soil profile
- 'Soil layers': each layer specified in the soil profile file has a separate *.dat file, which contains the water retention characteristics and unsaturated hydraulic conductivity properties in form of 'standard curves'. Each 'standard curve' consists of 20 discrete water tension values with the corresponding water contents (θ), unsaturated hydraulic conductivities (k) and penetrometer resistances specific for the regarded soil. The declaration of penetrometer resistances is a leftover of former STOTRASIM versions. In the current version the resistances are not used for calculation.
- 'Soil dispersion/diffusion file' called BOD_DIFF.DAT: contains parameters for the calculation of diffusion-, dispersion, ad- and desorption for all soil layers.

3)

- 'Parameter file' called kennwerte.dat: 14 plant parameters for each plant that control plant growth and development
  - extinction coefficient for global radiation
  - leaf area per weight of leaf dry matter
  - potential plant height
  - minimum stomatal resistance against loss of water

    - assimilation = maximum photosynthetic rate of leaf
    - temperature class (1-5)
    - potential root length
    - potential root length density
    - root strength class
    - potential leaf width
    - sum of accumulated photothermal units at riping
    - leaf area index of emergence
    - minimum air content
    - critical day length
- 'Biological conversion file' called umsetz.dat: information about microorganism activity, mineralization activity, nitrification activity and denitrification activity

4)

- 'Weather file': daily weather information (duration of day light, maximum and minimum air temperature, air temperature at 7 am, 2 pm and 7 pm, relative humidity at 7 am, 2 pm and 7pm, mean relative humidity, saturation deficit at 7 am, 2 pm and 7 pm, mean saturation deficit, mean wind speed, precipitation sum, global radiation sum)

5)

- 'Crop rotation file': lists all management measures including the crop rotation, tillage, irrigation as well as mineral and organic fertilization
- 'Groundwater file': groundwater fluctuation and its nitrogen concentration
- 'Irrigation file': date and amount of irrigation

The groundwater file and the irrigation file are optional; they are only needed if a scenario influenced by groundwater should be calculated or if irrigation took place during the calculated period.

The output is written automatically in a folder called 'Erg'. In the project file the user can select which output files should be created. The following output files are available:

- Soil temperature (‘BODTEMP.ERG’): daily values of soil temperature of each compartment
- Water flux (‘H2OFLUX.ERG’): daily water flux (mm) in each compartment
- Summary of averages and sums (‘MITODSUM.ERG’): a summary of the project period, for example summaries of the plant development (average dry matter production, maximal root depth, water storage available for plants, nitrogen uptake by plants, nitrogen removal at harvest), summaries of the water balance (actual evapotranspiration, groundwater recharge, groundwater uprise) and summaries of the nitrogen balance (nitrogen leachate, nitrogen uprise, average nitrogen concentration)
- Nitrogen fluxes (‘NFLUX.ERG’): daily nitrogen flux (kg N/ha) in each compartment
- Mineralized Nitrogen (‘Nmin.ERG’): daily values of Nmin in 10 cm intervals from soil surface to 100 cm depth
- Nitrate concentration (‘NO3KONZ.ERG’): daily values of NO_3 concentration in the soil water of each compartment
- Matric potential (‘SAUGSP.ERG’): daily matric potential of each compartment
- Summary soil water (‘STB.ERG’): water balance and plant development measures for each plant period of the crop rotation
- Summary soil nitrogen (‘STN.ERG’): nitrogen balance measure for each plant period of the crop rotation
- Summary stotrasim (‘STO.ERG’): both nitrogen and water balance measures for each plant period of the crop rotation as well as starting values of the soil profile.
- Daily values (‘TAGWERTE.ERG’): daily values of a series of nitrogen and water balance measures
- Water contents (‘WASSERAN.ERG’): daily values of water content of each compartment

2.2 AMALGAM

AMALGAM (Vrugt and Robinson, 2007) is a global, multialgorithm, genetically adaptive and multiobjective optimization method.

2.2.1 Method description

The algorithm starts by using an initial population P_0 of size *N* (whereby N has to be a minimum of 20), which can either be generated randomly by using Latin hypercube sampling (McKay, 1979) or starting values can be defined. Then, each parent is assigned a rank using the fast non-dominated sorting (FNS) algorithm. FNS is an approach to identify all non-dominated fronts. At first two entities have to be calculated for each solution:

- the domination count n_p = the number of solutions which dominate this solution p
- S_p= a set of solutions that the solution p dominates

All solutions in the first non-dominated front have their domination count n_p as zero. As a next step, for each solution in the first non-dominated front, the n_t of each member (t) of its set S_t is reduced by one. All those members t for which $\mathrm{n_t}$ becomes zero are put in a separate list T, that contains the second non-dominated front. This procedure is continued with each member of T and the third front is identified. The process can be continued until all non-dominated fronts are identified (Deb et al., 2002).

A population of offspring (Q_0 of size *N*) is subsequently created by simultaneously using different algorithms. By default four algorithms are used: the non-dominated sorted genetic algorithm II (Deb et al, 2002), particle swarm optimization (Kennedy and Eberhart, 1995), adaptive metropolis search (Metropolis et al., 1953) and differential evolution (Storn and Price, 1997). They were chosen because they were considered as ‘mutually consistent and complementary’ (Vrugt and Robinson, 2007). The users of the AMALGAM MatLab tool, however, can choose themselves which algorithms to use for offspring creation.

After creation of the offspring, a combined population of parents (P_0) and offspring (Q_0) is created (R_0, size *2N*) and ranked using FNS. All previous non-dominated members will always be included in R. This comparison of the current offspring with the previous generation ensures that it is always the best individuals that are carried over and so the solution quality will increase from one generation to the next. The members for the next population P_1 are chosen from the non-dominated fronts of R_0 based on their rank (FNS) and crowding distance. For the computation of the crowding distance, the population is sorted according to each objective value function in ascending order of magnitude. For each objective function, the solutions with the smallest and largest results are assigned an infinite distance value. All solutions in-between are assigned a distance value equal to the normalized difference in the function values of two adjacent solutions. This procedure is done for all objective functions and then the distance values are summed up to conclude the overall crowding-distance (Vrugt and Robinson, 2007).

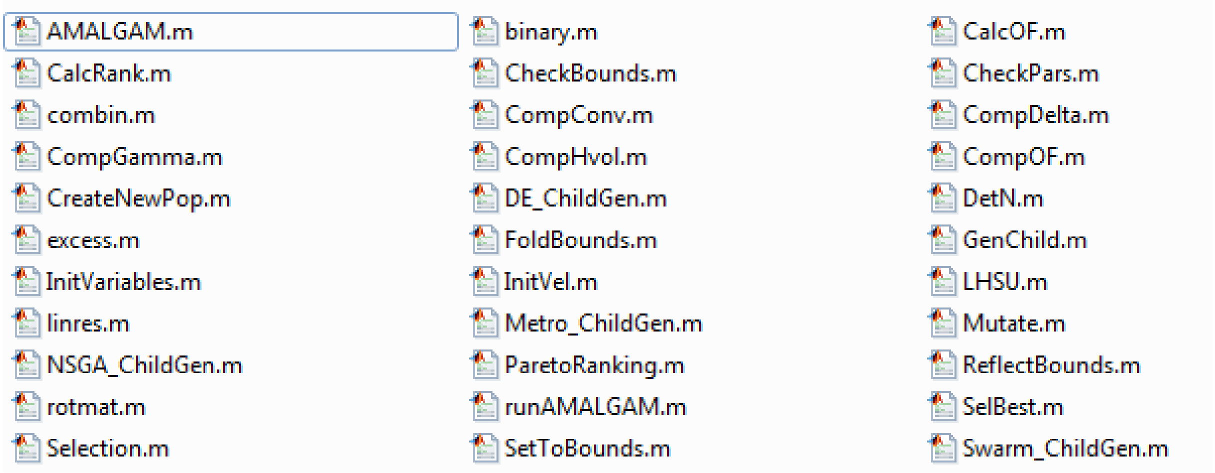
The new population P_1 is then used to create offspring again. AMALGAM favors individual algorithms that show the highest success. Therefore the number of offspring points an algorithm contributes to the new population is counted and set in relation to the corresponding number of offspring the algorithm created in the in the previous generation.

With this process the algorithms can be weighted, so that the best offspring creation methods contribute more offspring to the new population (Vrugt and Robinson, 2007).

All the steps are repeated until convergence is achieved (only possible for synthetic problems) or the predefined maximum number of function evaluations is reached (for real life problems) (Vrugt and Robinson, 2007). For the practical use of AMALGAM it is suggested to implement this method in multiple trials with a relatively small number of model runs rather than run it once with long iterations (Zhang et al, 2009).

2.2.2 AMALGAM as MatLab tool

AMALGAM is designed as a user-friendly MatLab tool with predefined areas and variable names where the specific data can be included. The AMALGAM MatLab package, provided by Jasper Vrugt, consists of 34 m-files (Figure 5).



AMALGAM.m
binary.m
CalcOF.m
CalcRank.m
CheckBounds.m
CheckPars.m
combin.m
CompConv.m
CompDelta.m
CompGamma.m
CompHvol.m
CompOF.m
CreateNewPop.m
DE_ChildGen.m
DetN.m
excess.m
FoldBounds.m
GenChild.m
InitVariables.m
InitVel.m
LHSU.m
linres.m
Metro_ChildGen.m
Mutate.m
NSGA_ChildGen.m
ParetoRanking.m
ReflectBounds.m
rotmat.m
runAMALGAM.m
SelBest.m
Selection.m
SetToBounds.m
Swarm_ChildGen.m

Figure 5: Overview of the MatLab scripts (m-files) of AMALGAM

The calibration process can be started with the script '*runAMALGAM.m*'. Here the algorithms to use in AMALGAM can be chosen ('*Extra.Alg*') and the dimension of the problem (number of parameters to be optimized; '*AMALGAMPar.q*'), the population size ('*AMALGAMPar.N*'), the maximum number of function evaluations ('*AMALGAMPar.ndraw*') and the number of objectives can be specified ('*AMALGAMPar.nobj*'), whereby the objective functions themselves have to be defined in '*CalcOF.m*'. Moreover in '*runAMALGAM.m*' the type of population initialization can be chosen ('*Extra.InitPopulation*'; 'LHS' for Latin Hypercube Sampling or 'PRIOR' for defining starting values). If chosen 'PRIOR', the particular starting values can be stored in the variable '*Extra.prior*'. In '*ParRange.minn*' and '*ParRange.maxn*' the parameter ranges can be defined. The measured data, to which the model output should be calibrated, can be uploaded here in '*runAMALGAM.m*' as well (variable '*Measurement.MeasData*').

When starting ‘*runAMALGAM.m’,* it automatically calls ‘*AMALGAM.m’*, which is the core file of the program. At first the algorithmic variables and necessary properties are initialized (‘*InitVariables.m’*). Then the first population is created by Latin Hypercube Sampling (‘*LHS.m’*) or the defined starting values are passed on. For each member of the population a function value is created with ‘*CompOF.m’.* These function results are then ranked in ‘*CalcRank.m’,* which separates individuals with and without constraint violation and sorts them in ‘*ParetoRanking.m’.* After saving of the first population in a matrix called ‘*ParSet*’, the iteration is started:

- Determination of the best parameter set for the Particle Swarm Optimization (‘*SelBest.m’*)
- Offspring generation (‘*GenChild.m’*) and check if parameters are in bound according to ParRange (‘*CheckPars.m’*)
- Then again, for each member of the population a function value is calculated (‘*CompOF.m’*)
- Merging of the parent and the child population and generation of a new population (‘*CreateNewPop.m’*). For this the children are appended to the parents and they are all ranked together by the procedure in the script ‘*CalcRank.m’*. Then a new population is created by all the algorithms selected in ‘*Extra.Alg’*. By default the algorithms NSGA-II (script ‘*NSGA_ChildGen.m’*), adaptive metropolis search (script ‘*Metro_ChildGen.m’*), particle swarm optimization (script ‘*Swarm_ChildGen.m’*) and differential evolution (script ‘*DE_ChildGen.m’*) are used.
- Determination of the new number of offspring points for the different optimization algorithms (‘*DetN.m’*)
- Saving of the new population to the matrix *‘ParSet’*

These steps are repeated until the maximum number of iterations, defined in the variable ‘*AMALGAMPar.ndraw’,* is reached. All the parameter sets and the corresponding function evaluation values can then be looked up in the matrix ‘*ParSet’.*

2.3 Models for Water Retention and Hydraulic Conductivity

The water retention curve is the relationship between the water content and the matric potential. The curve is characteristic for different soil types. There are a few models describing this relationship, for example the van Genuchten model (van Genuchten, 1980) or the Brooks and Corey model (Brooks and Corey, 1964). Van Genuchten described his model in 1980 and enabled the users to derive closed-form analytical expressions for the relative hydraulic conductivity *K* when substituted in the predictive conductivity models of Burdine (1953) or Mualem (1976). Such a predictive model that couples the conductivity function with the retention function is of particular advantage because it minimizes the number of needed parameters (Durner, 1994).

The van Genuchten water retention curve is generally described with

$$\theta(\psi) = \theta_r + \frac{\theta_s - \theta_r}{[1 + (\alpha|\psi|)^n]^m} \tag{7}$$

where

θ is the water content (L^3L^{-3});

$|\psi|$ is the matric potential head (L);

θ_s is the saturated water content (L^3L^{-3});

θ_r is the residual water content (L^3L^{-3});

α is a form parameter and related to the inverse of the air entry value, $\alpha > 0$ (L^{-1});

n is a form parameter and a measure of the pore-size distribution, $n > 1$ (dimensionless);

m is for the Mualem model $m = 1 - 1/n$ (dimensionless) (van Genuchten, 1980).

If one wants to describe the effective soil water content S_e which is defined by

$$S_e(\theta) = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{8}$$

the formula changes to

$$S_e(\psi) = \frac{1}{[1 + (\alpha|\psi|)^n]^m} \tag{9}$$

The following equation 10 was derived by Mualem (1976) for predicting the relative hydraulic conductivity *(K_r)* from knowledge of the soil-water retention curve:

$$K_r(S_e) = {S_e}^{\tau}\left[\frac{\int_0^{S_e}\left(\frac{1}{\psi}\right)dS_e}{\int_0^{1}\left(\frac{1}{\psi}\right)dS_e}\right]^2 \tag{10}$$

The parameter τ describes the correlation between the pores and the tortuosity of the flow path. τ, however, can also be seen as an additional degree of freedom that can be estimated by fitting the curve to measurements.

Durner (1994) extended the van Genuchten model because undisturbed soils frequently have pore systems that are different from the unimodal, approximately normal distributed type. These pore systems might be the result of specific particle-size distributions or to the formation of secondary pore systems by various soil genetic processes. Durner solved the problem by dividing the porous medium into two (or even more) overlapping regions and using a van Genuchten-Mualem type function for each of these regions. The following formula represents a two regions model (bimodal van Genuchten-Mualem bvGM):

$$S_e = \frac{w}{[1+(\alpha_1|\psi|)^{n_1}]^{m_1}} + \frac{(1-w)}{[1+(\alpha_2|\psi|)^{n_2}]^{m_2}} \tag{11}$$

Combining this retention model with Mualem (1976) pore-size distribution model leads to a hydraulic conductivity K:

$$K(S_e) = K_s \frac{\left[wS_{e_1} + (1-w)S_{e_2}\right]^{\tau}\left[w\alpha_1\left[1-\left(1-{S_{e_1}}^{\frac{1}{m_1}}\right)^{m_1}\right] + (1-w)\alpha_2\left[1-\left(1-{S_{e_2}}^{\frac{1}{m_2}}\right)^{m_2}\right]\right]^2}{[w\alpha_1 + (1-w)\alpha_2]^2} \tag{12}$$

2.4 Measures of goodness-of-fit

The goodness of fit describes the performance quality of a model. Measures of goodness of fit typically summarize the discrepancy between observed values and simulated values. The Nash Sutcliffe Efficiency Coefficient (NSE; Nash and Sutcliffe, 1970) is a commonly used measure for the goodness of fit amongst hydrologists (Schaefli and Gupta, 2007). It is well suited to describe the predictive accuracy of models when there exist observed time series to compare the model results to, for example discharge, and is generally given by:

$$NSE = 1 - \frac{\sum_{t=1}^{n}(O_t - P_t)^2}{\sum_{t=1}^{n}(O_t - \bar{O})^2} \tag{13}$$

where n is the number of observations, O_t is the observed value at time step t, P_t is the corresponding model predicted value, and $\bar{O}$ is the mean value of all observations (Nash and Sutcliffe, 1970). The dimensionless NSE ranges between 1 and $-\infty$, where $NSE = 1$ gives a perfect model fit and for $NSE = 0$ the average of the observations would be a better predictor than the model.

As another - more general - measure, the least squares method (SSE) defines an optimum when the sum of squared residuals r is a minimum.

$$SSE = \sum_{j=1}^{m} {r_j}^2 \tag{14}$$

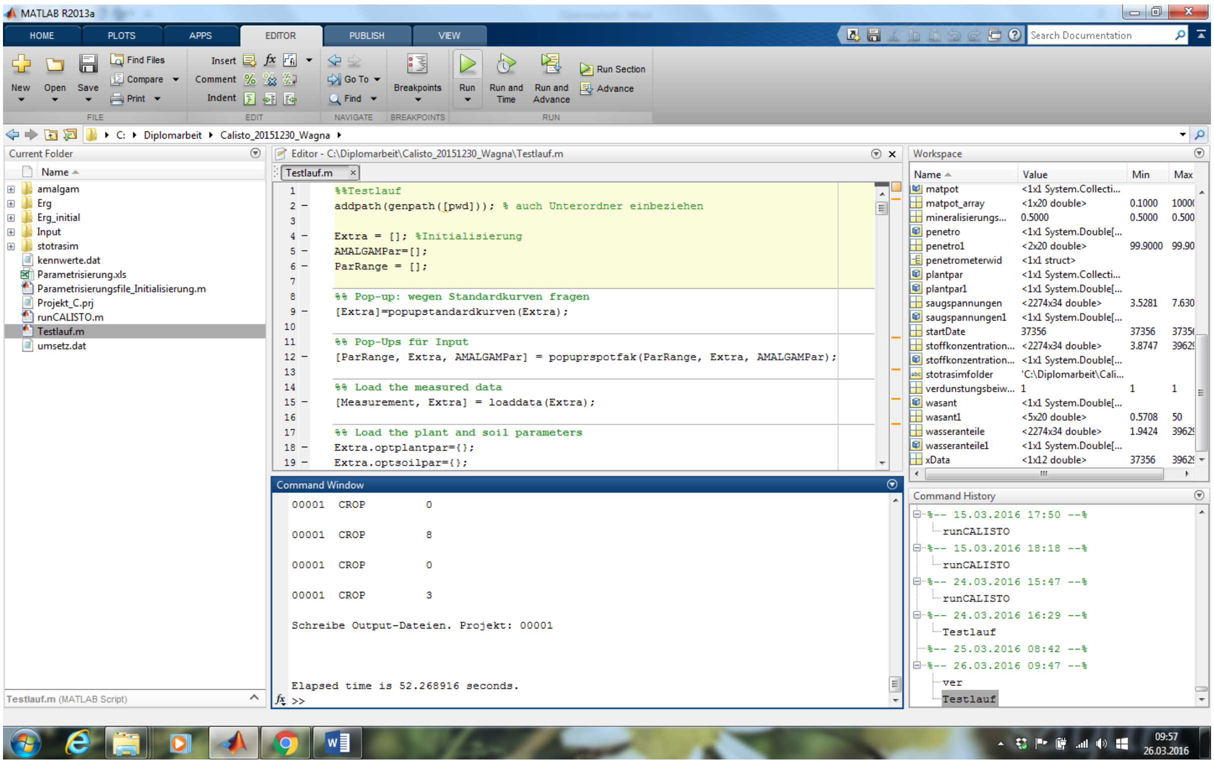
The residual r is defined as the difference between the measured value of a variable (O_j) and the value predicted by the model (P_j).

$$r_j = O_j - P_j \tag{15}$$

2.5 The Programming Languages and Development Environments

AMALGAM (2.2) is coded in MatLab, whereas SIMWASER/STOTRASIM (2.1) is written in the higher programming language C#.

MatLab is a proprietary programming language and a computing environment software of the company MathWorks. It is designed for solving mathematical problems, for graphical representation of the results and the creation of user interfaces. MatLab is created especially for the numerical calculation with the help of matrices, therefore its name is derived from Matrix Laboratory. Add-on toolboxes extend the MatLab environment to solve particular classes of problems in these application areas (Schweizer, 2013). Figure 6 shows the screenshot of the MatLab environment.



amalgam
Erg
Erg_initial
Input
stotrasim
kennwerte.dat
Parametrisierung.xls
Parametrisierungsfile_Initialisierung.m
Projekt_C.prj
runCALISTO.m
Testlauf.m
umsetz.dat

```
%%Testlauf
addpath(genpath([pwd])); % auch Unterordner einbeziehen

Extra = []; %Initialisierung
AMALGAMPar=[];
ParRange = [];

%% Pop-up: wegen Standardkurven fragen
[Extra]=popupstandardkurven(Extra);

%% Pop-Ups für Input
[ParRange, Extra, AMALGAMPar] = popuprspotfak(ParRange, Extra, AMALGAMPar);

%% Load the measured data
[Measurement, Extra] = loaddata(Extra);

%% Load the plant and soil parameters
Extra.optplantpar={};
Extra.optsoilpar={};
```

```
00001   CROP          0

00001   CROP          8

00001   CROP          0

00001   CROP          3

Schreibe Output-Dateien. Projekt: 00001



Elapsed time is 52.268916 seconds.
>>
```

Name	Value	Min	Max
matpot	<1x1 System.Collecti...		
matpot_array	<1x20 double>	0.1000	1000(
mineralisierungs...	0.5000	0.5000	0.500
penetro	<1x1 System.Double[...		
penetro1	<2x20 double>	99.9000	99.90
penetrometerwid	<1x1 struct>		
plantpar	<1x1 System.Collecti...		
plantpar1	<1x1 System.Double[...		
saugspannungen	<2274x34 double>	3.5281	7.630
saugspannungen1	<1x1 System.Double[...		
startDate	37356	37356	3735(
stoffkonzentration...	<2274x34 double>	3.8747	3962(
stoffkonzentration...	<1x1 System.Double[...		
stotrasimfolder	'C:\Diplomarbeit\Cali...		
verdunstungsbeiw...	1	1	1
wasant	<1x1 System.Double[...		
wasant1	<5x20 double>	0.5708	50
wasseranteile	<2274x34 double>	1.9424	3962(
wasseranteile1	<1x1 System.Double[...		
xData	<1x12 double>	37356	3962(

```
%-- 15.03.2016 17:50 --%
  runCALISTO
%-- 15.03.2016 18:18 --%
  runCALISTO
%-- 24.03.2016 15:47 --%
  runCALISTO
%-- 24.03.2016 16:29 --%
  Testlauf
%-- 25.03.2016 08:42 --%
%-- 26.03.2016 09:47 --%
  ver
  Testlauf
```

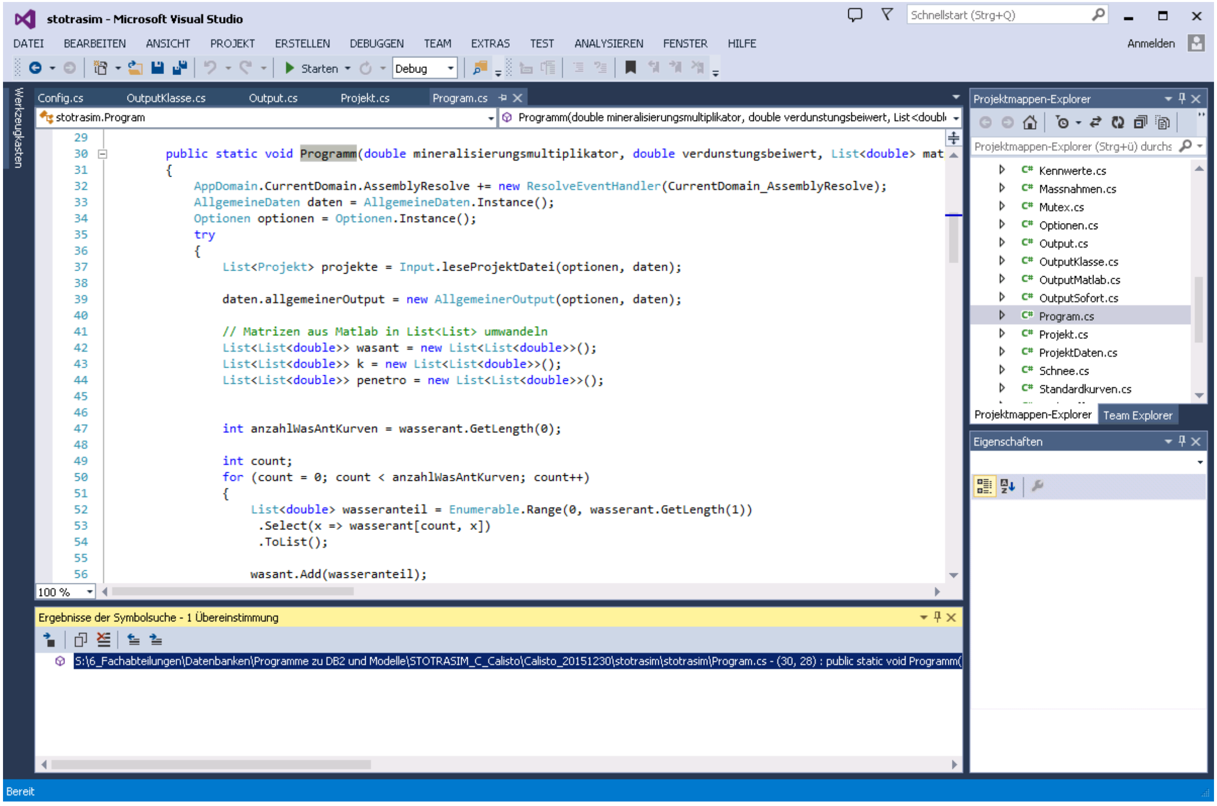
Figure 6: Screenshot of the MatLab environment

The environment usually has the following panels:

- Current folder: to access the project folders and files
- Command window: main area where commands can be entered
- Command history: shows or reruns commands that are entered in the command window
- Workspace: lists the variables
- Editor: to show script files with the extenstion ‘.m’, where the commands can be placed and saved (Schweizer, 2013)

There are new releases of the program every half of a year, named with the year adding the letter a for the first half of the year or b for the second half. This master thesis was developed by using the versions R2013a and R2015a. It was crucial to use a version after 2010 because in earlier releases it had not been possible to call .NET applications, but which is necessary to couple MatLab with C#. For the use of AMALGAM (2.2.2) the Statistics Toolbox is obligatory.

C# is a higher level programming level that was developed by Microsoft within its .NET initiative that was released in 2002. .NET stands for a software framework that runs primarily on Microsoft Windows. The framework allows the creation of language independent code. C# was the only language that was developed completely new for the .NET initiative (Kühnel, 2013). For this master thesis the C# code of SIMWASER/STOTRASIM (2.1) was edited with the programming environment Visual Studio 2010 Shell by Microsoft (Figure 7). The environment supports writing, debugging and testing codes in the languages C#, F#, C++ and VB.NET.


stotrasim - Microsoft Visual Studio
Schnellstart (Strg+Q)
DATEI BEARBEITEN ANSICHT PROJEKT ERSTELLEN DEBUGGEN TEAM EXTRAS TEST ANALYSIEREN FENSTER HILFE
Anmelden
Starten
Debug
Werkzeugkasten
Config.cs OutputKlasse.cs Output.cs Projekt.cs Program.cs
stotrasim.Program
Programm(double mineralisierungsmultiplikator, double verdunstungsbeiwert, List<doubl
100 %
Projektmappen-Explorer
Projektmappen-Explorer (Strg+ü) durchs
Kennwerte.cs
Massnahmen.cs
Mutex.cs
Optionen.cs
Output.cs
OutputKlasse.cs
OutputMatlab.cs
OutputSofort.cs
Program.cs
Projekt.cs
ProjektDaten.cs
Schnee.cs
Standardkurven.cs
Projektmappen-Explorer Team Explorer
Eigenschaften
Ergebnisse der Symbolsuche - 1 Übereinstimmung
Bereit

```
        public static void Programm(double mineralisierungsmultiplikator, double verdunstungsbeiwert, List<double> mat
        {
            AppDomain.CurrentDomain.AssemblyResolve += new ResolveEventHandler(CurrentDomain_AssemblyResolve);
            AllgemeineDaten daten = AllgemeineDaten.Instance();
            Optionen optionen = Optionen.Instance();
            try
            {
                List<Projekt> projekte = Input.leseProjektDatei(optionen, daten);

                daten.allgemeinerOutput = new AllgemeinerOutput(optionen, daten);

                // Matrizen aus Matlab in List<List> umwandeln
                List<List<double>> wasant = new List<List<double>>();
                List<List<double>> k = new List<List<double>>();
                List<List<double>> penetro = new List<List<double>>();


                int anzahlWasAntKurven = wasserant.GetLength(0);

                int count;
                for (count = 0; count < anzahlWasAntKurven; count++)
                {
                    List<double> wasseranteil = Enumerable.Range(0, wasserant.GetLength(1))
                     .Select(x => wasserant[count, x])
                     .ToList();

                    wasant.Add(wasseranteil);
```

```
S:\6_Fachabteilungen\Datenbanken\Programme zu DB2 und Modelle\STOTRASIM_C_Calisto\Calisto_20151230\stotrasim\stotrasim\Program.cs - (30, 28) : public static void Programm(
```

Figure 7: Screenshot of the Visual Studio environment.

Visual Studio includes a code editor that supports syntax highlighting and code completions for variables, functions, methods and loops. The code editor also supports setting bookmarks in the code for a quick navigation. Moreover, it includes a debugger that allows setting breakpoints which enables execution to be stopped temporarily at a certain position (Kühnel, 2013).

3. Calisto – The Calibration Program

The main objective of this master project is to implement a tool for the automatic calibration of 'STOTRASIM C#' with AMALGAM. Therefore SIMWASER/STOTRASIM (programmed in C#) and AMALGAM, (programmed in MatLab) had to be linked. The resulting software shall henceforward be called Calisto ('Calibration Stotrasim').

The general optimization procedure is that SIMWASER/STOTRASIM receives parameter sets that were created within MatLab, executes the simulation and passes back the simulation results to MatLab, where the simulated data series can be compared to the observed behavior of the system. AMALGAM ranks the parameter sets according to their ability to reproduce the measured system behaviour and creates a new generation of data sets (as described in 2.2.1). This is then passed to SIMWASER/STOTRASIM in a loop process (Figure 8).

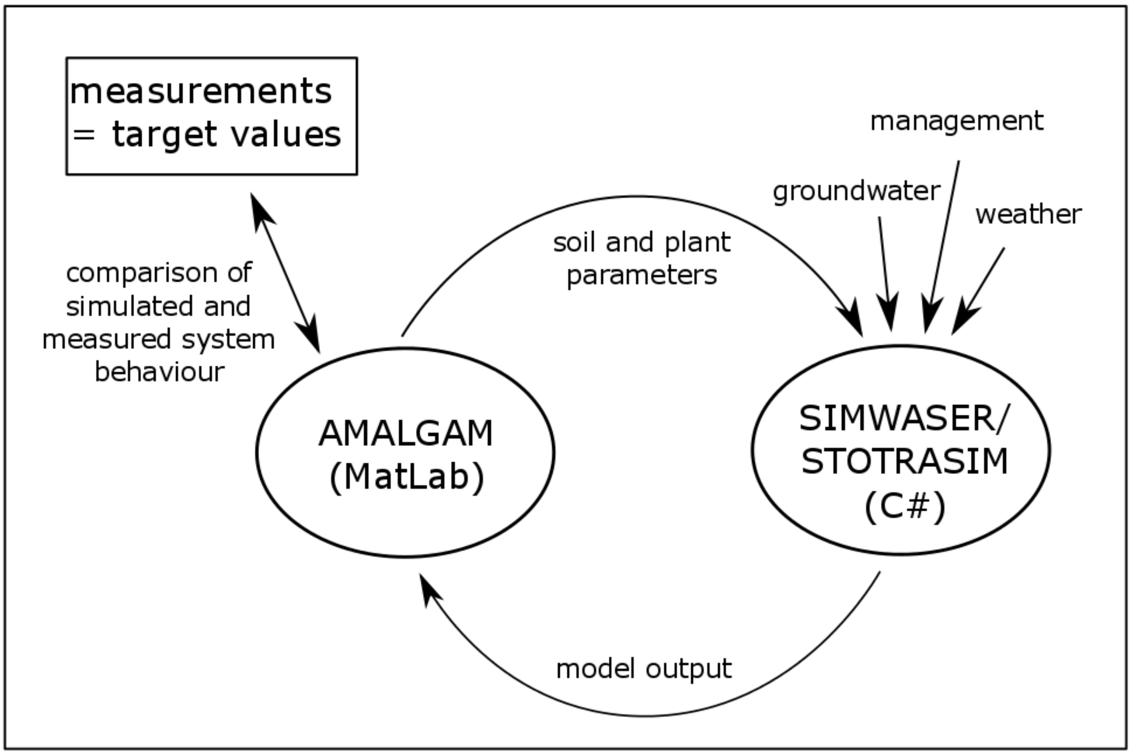


Figure 8: The operation principle of Calisto

In the past, all input files needed (2.1.3) had been read in and output files had been written by SIMWASER/STOTRASIM C# in form of text files. For Calisto, however, all the input parameters that should be calibrated have to be passed on from AMALGAM (MatLab) to SIMWASER/STOTRASIM (C#). In just the same way, the model results have to be passed from C# to MatLab so that AMALGAM can compare these results with the measurements. Therefore communication and ways of reciprocal data exchange between AMALGAM (MatLab) and SIMWASER/STOTRASIM (C#) had to be established. For the input of the soil characteristics, model parameters had to be defined (3.2). These can then be transformed into ‘standard curves’, which are necessary for SIMWASER/STOTRASIM calculation (2.1.3). Moreover there is a need for the user to enter measured values and the parameters that do not need to be calibrated. Figure 9 shows the operation principle of Calisto extended with the input and output matrices and variables that are transferred between AMALGAM (MatLab) and SIMWASER/STOTRASIM (C#).

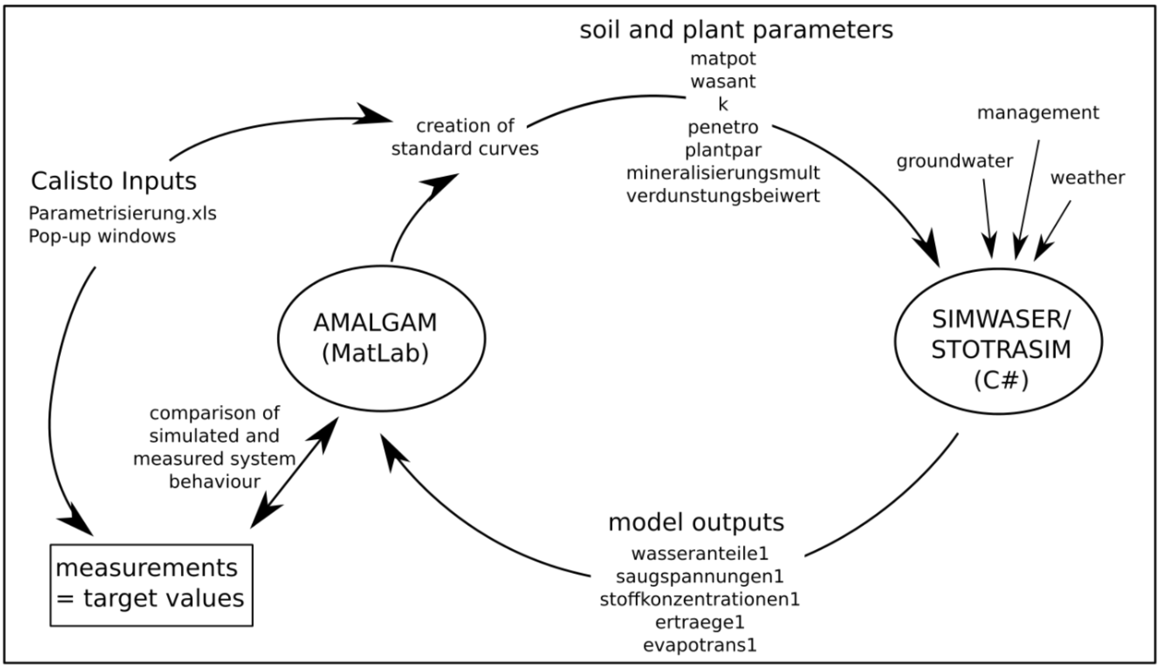


Figure 9: Inputs and Outputs transferred between AMALGAM (MatLab) and SIMWASER/STOTRASIM (C#)

Moreover it is necessary to define starting values and ranges for the input parameters and to write an output of the optimization process. For the performance of the optimization it is crucial to formulate appropriate objective functions. All relevant work steps are explained subsequently.

3.1 Calling C# with MatLab

SIMWASER/STOTRASIM, which had been a C# console application before, is saved as .NET assembly. The ‘Main Function’ that specifies the starting point of the program execution is renamed and attributed as ‘public’, so that it can be accessed externally. The .NET assembly is compiled to a .DLL-file, so the application is not self-launching any more. The command *NET.addAssembly(‘path//to//assembly’)* specifies which assembly should be accessed. The STOTRASIM .NET assembly is usually stored in the same Windows folder with the MatLab scripts and can therefore be added to MatLab by:

```
stotrasimfolder = [pwd '\stotrasim\stotrasim\bin\Debug\stotrasim.dll'];

NET.addAssembly(fullfile(stotrasimfolder));
```

The pwd command displays the current folder. SIMWASER/STOTRASIM can be called with the command *Namespace.Class.Function():*

```
[wasseranteile1, saugspannungen1, stoffkonzentrationen1,  ertraege1, evapotrans1] =
        stotrasim.Program.Programm (mineralisierungsmult, verdunstungsbeiwert, matpot, wasant, k,
        penetro, plantpar)
```

The matrices and variables

- ‘*wasseranteile1*’ (= daily values of water contents in all compartments)
- ‘*saugspannungen1*’ (= daily values of matric potentials in all compartments)
- ‘*stoffkonzentrationen1*’ (=daily values of nitrate concentration in all compartments)
- ‘*ertraege1*’ (= yields of all crops)
- ‘*evapotrans1*’ (= mean annual evapotranspiration)

are the results from SIMWASER/STOTRASIM which are passed on to MatLab.

The variables

- ‘*mineralisierungsmult*’ (mineralisation multiplier ‘*potfak*’)
- ‘*verdunstungsbeiwert*’ (evaporation coefficient ‘*rs*’)
- ‘*matpot*’ (20 values of matric potential for the ‘standard curve’ – for all soil types)
- ‘*wasant*’ (20 values of water content for the ‘standard curve’ – for all soil types)
- ‘*k*’ (20 values of hydraulic conducticity for the ‘standard curve’ – for all soil types)
- ‘*penetro*’ (20 values of penetrometer resistance for the ‘standard curve’ – for all soil types)
- ‘*plantpar*’ (plant parameters for all crops)

are passed from MatLab as input for SIMWASER/STOTRASIM.

3.2 pF and K_u Curves

All calculations in SIMWASER/STOTRASIM are based on ‘standard curves’ (20 discrete water tension values with the corresponding water contents and unsaturated hydraulic conductivities; 2.1.3). For the optimization of the water retention and unsaturated soil hydraulic functions, however, it is not feasible to adjust 20 points. Therefore in Calisto the soil moisture and hydraulic characteristics should be characterized with a different approach. For this purpose the bimodal van Genuchten-Mualem (bvGM) model was chosen (2.3).

This model is described with 11 parameters that can form both the water retention and the hydraulic conductivity curve:

- θ_s (saturated water content)
- θ_r (residual water content)
- w (weighing factor)
- $\propto_1$ (form parameter of the first ‘subcurve’)
- n_1 (form parameter of the first ‘subcurve’)
- m_1= 1 – 1/n_1
- $\propto_2$ (form parameter of the second ‘subcurve’)
- n_2 (form parameter of the second ‘subcurve’)
- m_2 = 1 – 1/n_2
- K_s (saturated hydraulic conductivity)
- τ_s (tortuosity factor)

The parameter values for each soil type can either be created by AMALGAM or read in from the Excel file ‘*Parametrisierung.xls’* (3.3).

The bvGM parameters are only used for the optimization within AMALGAM. SIMWASER/STOTRASIM still works with ‘standard curves’; so every time before the SIMWASER/STOTRASIM calculation is initiated, ‘standard curves’ need to be created out of the parameters. For this purpose the following formulas are used in MatLab to calculate the twenty values of the matric potential ψ (from 0.1 to 10000000 hPa) to form the ‘standard curve’:

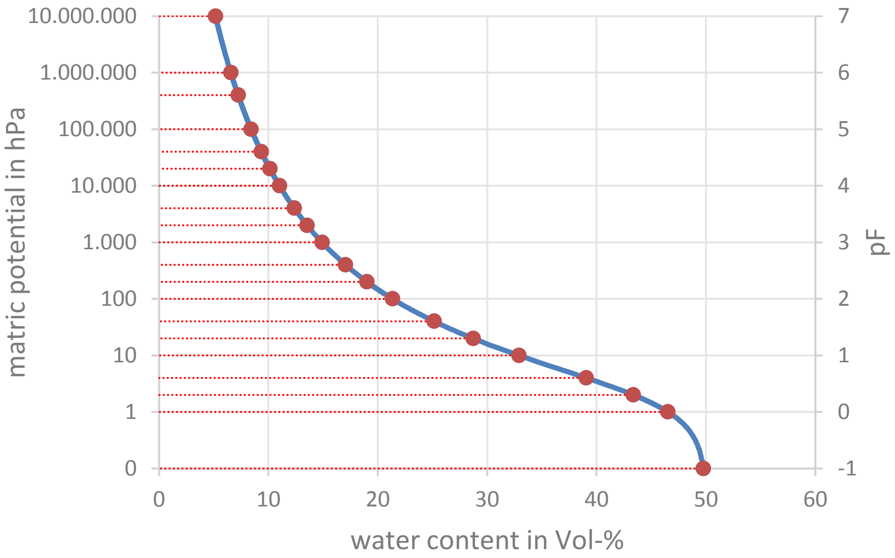
$$\theta(\psi) = \Big(\big((1-w) * (1+(\alpha_1 * \psi)^{n_1})^{m_1 * -1} + w * (1+(\alpha_2 * \psi)^{n_2})^{m_2 * -1}\big) * (\theta_s - \theta_r) + \theta_r\Big) * 100 \qquad (16)$$

$$k(\psi) = ((1 + (\alpha 1 * \psi)^{n1})^{m1*-1})^{\tau} * \frac{\left(\alpha 1*(1-(\alpha_1*\psi)^{n_1-1}*\left(\left(1+(\alpha 1*\psi)\right)^{n1}\right)^{m1*-1})\right)^2}{(\alpha 1)^2} * Ks$$

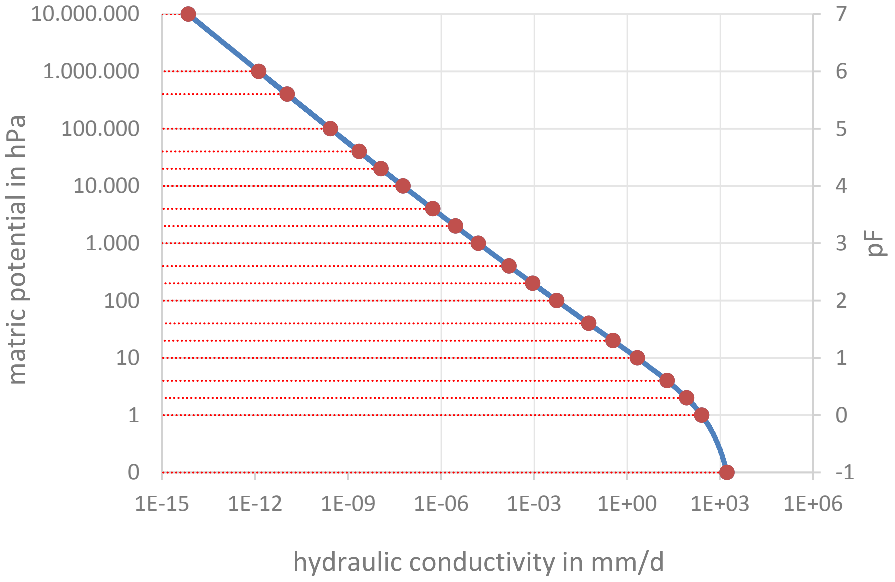
(17)

The ‘standard curves’ are stored in MatLab matrices (3.5). Figure 10 shows the example of a water retention curve and the hydraulic conductivity curve modelled by bvGM, with the specified parameter values. The red dots in the diagram as well as the matrices on the right hand side show the corresponding points of the ‘standard curve’.

$\theta_s = 0.5$ $\theta_r = 0.0$ $w = 0.5$ $\alpha_1 = 0.8$ $n_1 = 1.1$ $\alpha_2 = 0.5$ $n_2 = 1.4$ $K_s = 1000$ $\tau = 0.5$



hPa	Vol-%
10000000	5.2
1000000	6.6
400000	7.2
100000	8.4
40000	9.3
20000	10.1
10000	11.0
4000	12.3
2000	13.5
1000	14.9
400	17.0
200	19.0
100	21.3
40	25.2
20	28.7
10	32.9
4	39.0
2	43.4
1	46.5
0.1	49.8



hPa	mm/d
10000000	7.0E-15
1000000	1.3E-12
400000	1.1E-11
100000	2.7E-10
40000	2.3E-09
20000	1.2E-08
10000	6.0E-08
4000	5.4E-07
2000	2.9E-06
1000	1.6E-05
400	1.6E-04
200	9.1E-04
100	5.4E-03
40	5.8E-02
20	3.6E-01
10	2.1E+00
4	2.0E+01
2	8.4E+01
1	2.6E+02
0.1	1.7E+03

Figure 10: Example for bimodal van Genuchten-Mualem retention and hydraulic conductivity curves and the corresponding ‘standard curves’

3.3 Calisto Inputs

There are a few inputs that need to be provided for Calisto, first of all, soil and plant parameters. Calisto has to read in predefined values for the parameters as well as tags where parameters should be created and optimized by AMALGAM. Moreover Calisto needs to load measured values for calibration. Most input information is passed on with the Excel File '*Parametrisierung.xls'*. Further information (number of iteration, the parameters '*rs*' and '*potfak*') is queried after the start of Calisto with the help of pop-up windows.

3.3.1 Excel file 'Parametrisierung.xls'

The Excel file '*Parametrisierung.xls'* consists of seven sheets:

- '*Pflanzenparameter*' (plant parameters)
- '*Bodenparameter*' (soil parameters)

parameters

- '*Erträge*' (yields)
- '*Evapotranspiration*' (evapotranspiration)
- '*MatPot*' (matric potential)
- '*WasAnt*' (water content)
- '*Nitratkonzentrationen*' (nitrate concentration)

measurements

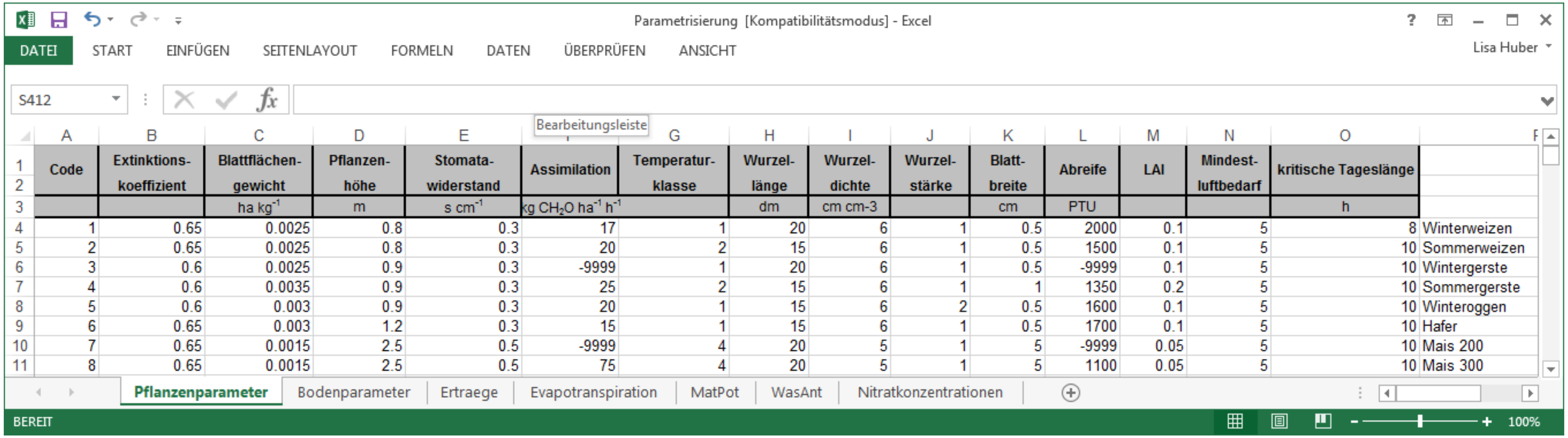
In sheet '*Pflanzenparameter'* there are sixty rows with the plant parameters as explained in 2.1.3 (Figure 11). It corresponds to the former section DATA PLANTF in the parameter file. In sheet *'Bodenparameter'* the soil parameters θ_r, θ_s, $\propto_1$, $\propto_2$, n_1, n_2, w, K_s and τ for the soil layers are stored (Figure 12). The parameters can be modified as required and parameters that should be optimized have to be replaced by '-9999'. The sheets for the yields (above-ground dry mass in kg/ha; Figure 13), evapotranspiration (in mm/a; Figure 14), matric potential (in hPa; Figure 15), water content (in Vol-%; Figure 16) and nitrate concentration (in mg/l; Figure 17) have to be filled with the measured data to which the model should be fitted.

Before filling the Excel file with values for the Calisto application, it can be prepared by executing the MatLab script '*Parametrisierungsfile_Initialisierung.m*'. Afterwards, all the soil types used in the project are listed automatically in the Excel file and if standardized parameters for those layers exist, they are written in the specific rows (Figure 12). Moreover, the project relevant time periods are filled in the grey columns of the sheets '*Erträge', 'MatPot', 'WasAnt'* and '*Nitratkonzentrationen'*. The automatic specification of the dates should be a convenience for the program user and could be reached by programming a

MatLab code that reads the information in the ‘soil profile file’ and the ‘crop rotation file’ (2.1.3) of the input folder and writes it into the Excel file ‘Parametrisierung’. As an example, the following MatLab code section shows the reading of the crop names (‘*fruchtname’*) and the sowing and harvest dates (‘*beginndatum’* and ‘*enddatum’*) of the crop cultivation:

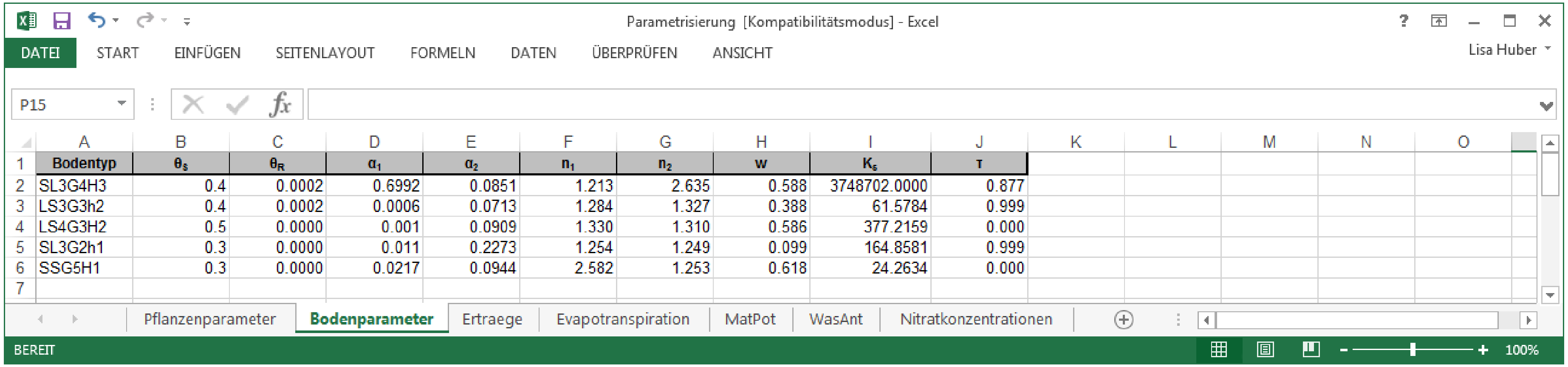
```
while ~feof(fid) %loop till end of file
zeile3=fgetl(fid);
if (zeile3(4)~=' ')
fruchtname=[fruchtname; zeile3(6:25)];          %all crop names
fruchtname = strrep(fruchtname, ' ', ");
beginndatum=[beginndatum; zeile3(27:37)];   %all beginning dates
enddatum=[enddatum; zeile3(39:49)];             %all harvest dates
end
```

This data could then be written into the sheet ‘*Ertraege*’ (Figure 13). Moreover, the first sowing date and the last harvest date could be used for determining the total simulation period and fill the grey columns in Figure 15, Figure 16 and Figure 17. If there are measurements available, they are filled in the sheets; individual missing measurements are replaced by ‘-9999’. For matric potential, water content and nitrate concentration several columns for different soil depths are created.



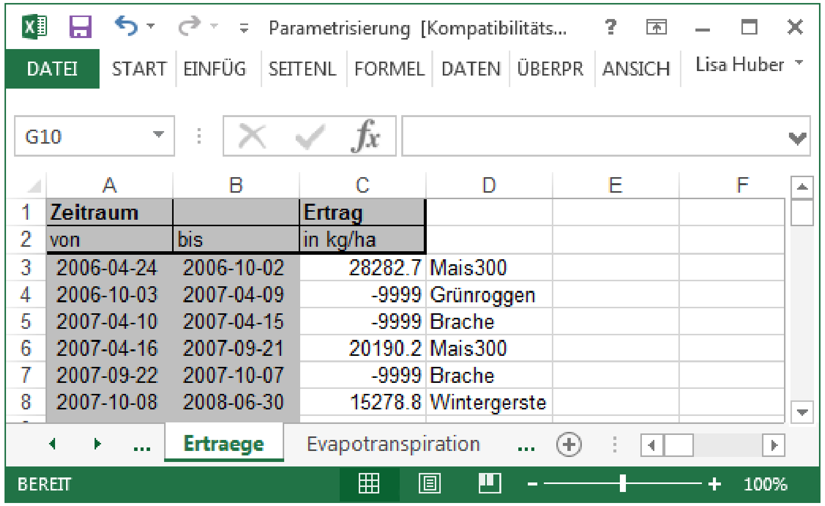
Code	Extinktions-koeffizient	Blattflächen-gewicht	Pflanzen-höhe	Stomata-widerstand	Assimilation	Temperatur-klasse	Wurzel-länge	Wurzel-dichte	Wurzel-stärke	Blatt-breite	Abreife	LAI	Mindest-luftbedarf	kritische Tageslänge	
		ha kg^{-1}	m	s cm^{-1}	kg CH$_2$O ha^{-1} h^{-1}		dm	cm cm-3		cm	PTU			h	
1	0.65	0.0025	0.8	0.3	17	1	20	6	1	0.5	2000	0.1	5	8	Winterweizen
2	0.65	0.0025	0.8	0.3	20	2	15	6	1	0.5	1500	0.1	5	10	Sommerweizen
3	0.6	0.0025	0.9	0.3	-9999	1	20	6	1	0.5	-9999	0.1	5	10	Wintergerste
4	0.6	0.0035	0.9	0.3	25	2	15	6	1	1	1350	0.2	5	10	Sommergerste
5	0.6	0.003	0.9	0.3	20	1	15	6	2	0.5	1600	0.1	5	10	Winteroggen
6	0.65	0.003	1.2	0.3	15	1	15	6	1	0.5	1700	0.1	5	10	Hafer
7	0.65	0.0015	2.5	0.5	-9999	4	20	5	1	5	-9999	0.05	5	10	Mais 200
8	0.65	0.0015	2.5	0.5	75	4	20	5	1	5	1100	0.05	5	10	Mais 300

Figure 11: Some of the 60 entries of the sheet '*Pflanzenparameter*' in '*Parametrisierung.xls'*. In this example the assimilation and the riping parameters of the plants with the codes 3 and 7 should be optimized, therefore they are replaced by '-9999'.



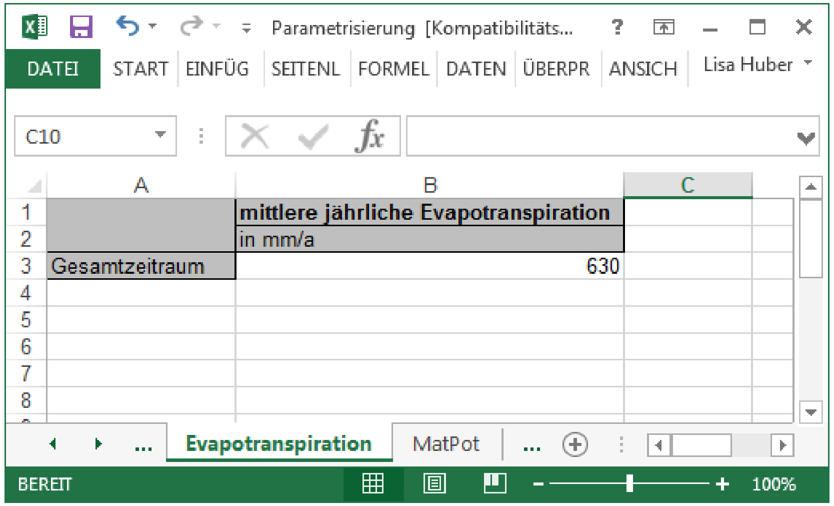
Bodentyp	θ_S	θ_R	α_1	α_2	n_1	n_2	w	K_s	τ
SL3G4H3	0.4	0.0002	0.6992	0.0851	1.213	2.635	0.588	3748702.0000	0.877
LS3G3h2	0.4	0.0002	0.0006	0.0713	1.284	1.327	0.388	61.5784	0.999
LS4G3H2	0.5	0.0000	0.001	0.0909	1.330	1.310	0.586	377.2159	0.000
SL3G2h1	0.3	0.0000	0.011	0.2273	1.254	1.249	0.099	164.8581	0.999
SSG5H1	0.3	0.0000	0.0217	0.0944	2.582	1.253	0.618	24.2634	0.000

Figure 12: The sheet '*Bodenparameter*' in '*Parametrisierung.xls'*: the soil parameters of 6 soil types. In this example, the soil parameters are given and should not be optimized. Therefore no parameter is replaced by '-9999'.



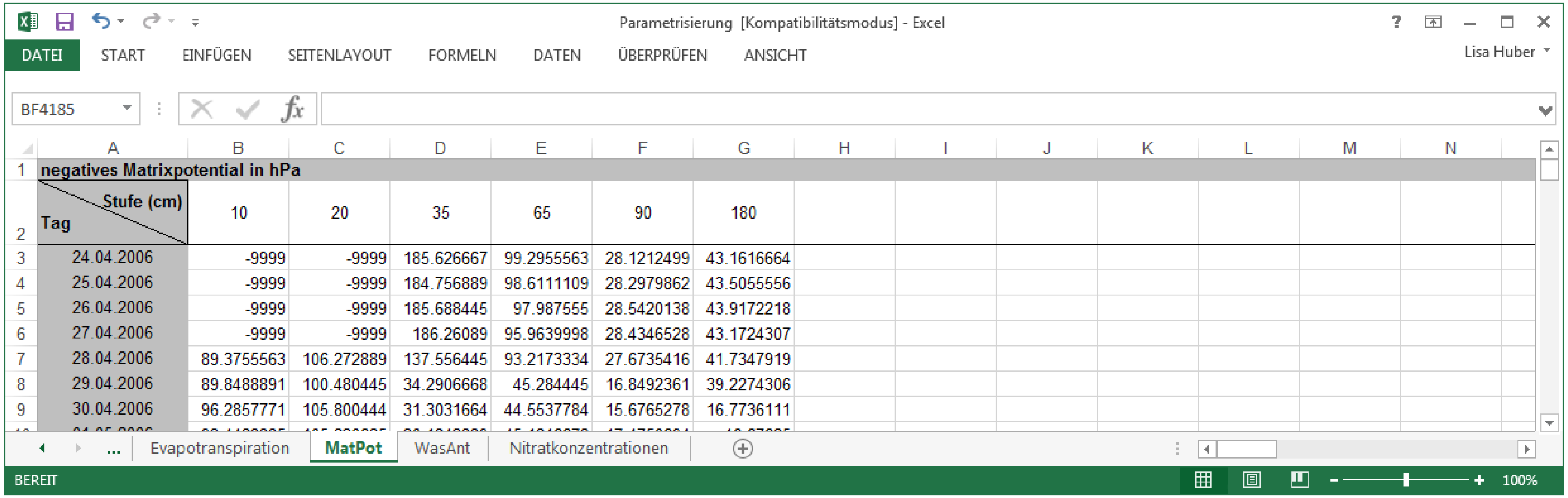
Zeitraum		Ertrag	
von	bis	in kg/ha	
2006-04-24	2006-10-02	28282.7	Mais300
2006-10-03	2007-04-09	-9999	Grünroggen
2007-04-10	2007-04-15	-9999	Brache
2007-04-16	2007-09-21	20190.2	Mais300
2007-09-22	2007-10-07	-9999	Brache
2007-10-08	2008-06-30	15278.8	Wintergerste

Figure 13: The sheet ‘*Ertraege*’ in ‘*Parametrisierung.xls*’



	mittlere jährliche Evapotranspiration
	in mm/a
Gesamtzeitraum	630

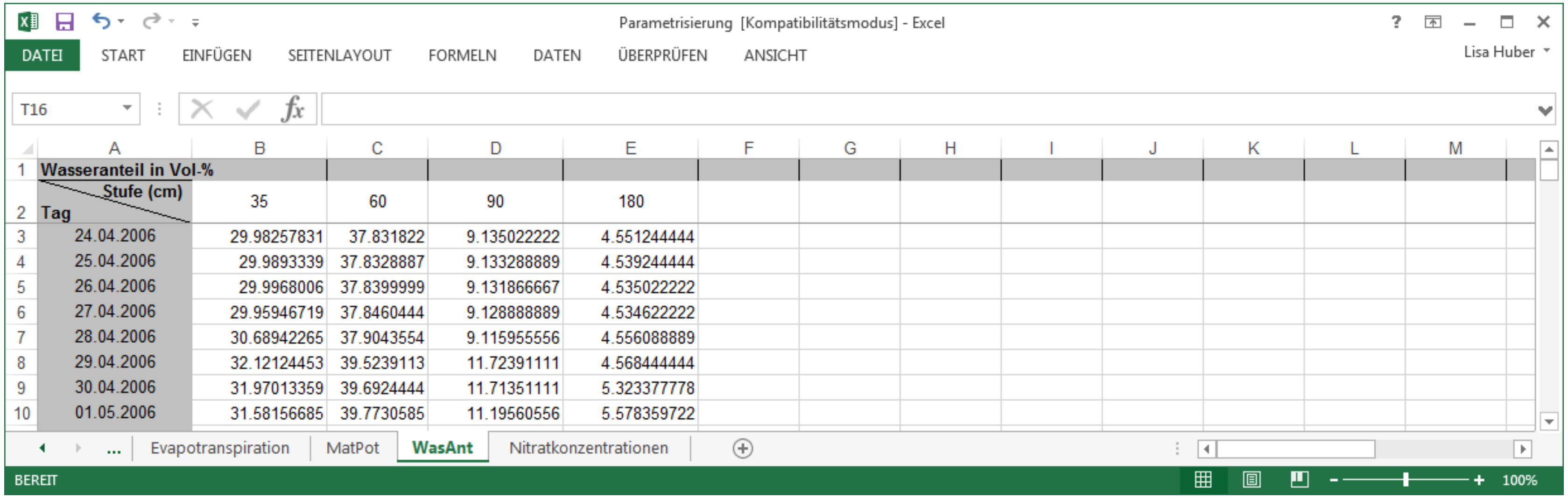
Figure 14: The sheet ‘*Evapotranspiration*’ in ‘*Parametrisierung.xls*’



negatives Matrixpotential in hPa

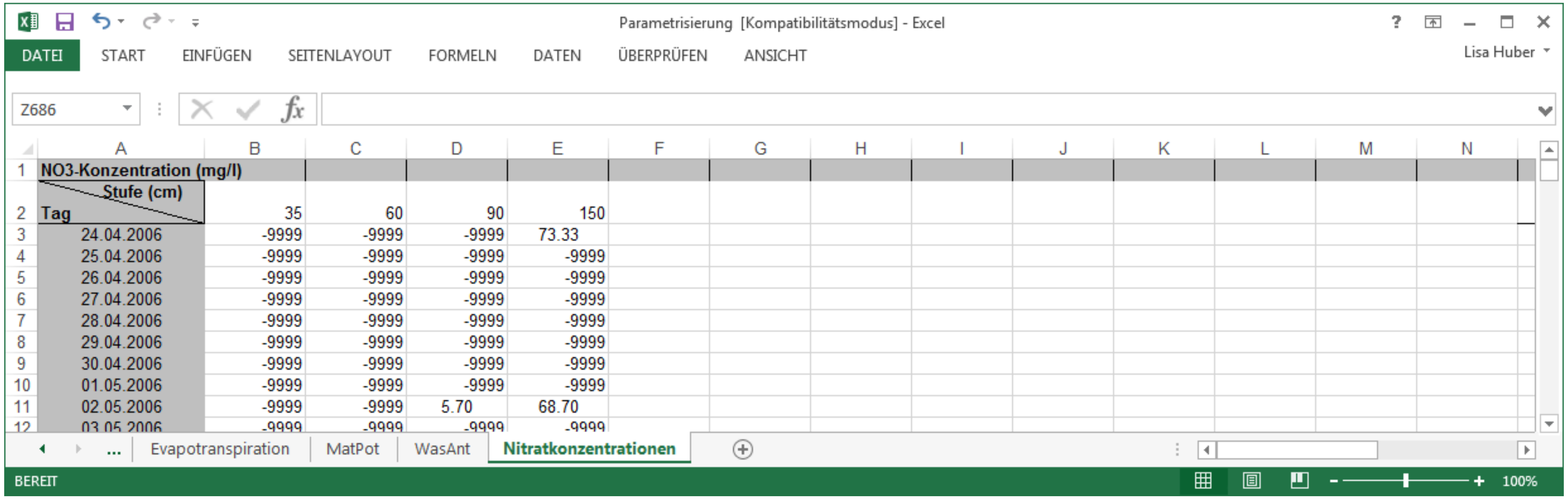
Stufe (cm) / Tag	10	20	35	65	90	180
24.04.2006	-9999	-9999	185.626667	99.2955563	28.1212499	43.1616664
25.04.2006	-9999	-9999	184.756889	98.6111109	28.2979862	43.5055556
26.04.2006	-9999	-9999	185.688445	97.987555	28.5420138	43.9172218
27.04.2006	-9999	-9999	186.26089	95.9639998	28.4346528	43.1724307
28.04.2006	89.3755563	106.272889	137.556445	93.2173334	27.6735416	41.7347919
29.04.2006	89.8488891	100.480445	34.2906668	45.284445	16.8492361	39.2274306
30.04.2006	96.2857771	105.800444	31.3031664	44.5537784	15.6765278	16.7736111

Figure 15: The sheet ‘*MatPot*’ in ‘*Parametrisierung.xls*’



	A	B	C	D	E
1	**Wasseranteil in Vol-%**				
2	**Stufe (cm)** **Tag**	35	60	90	180
3	24.04.2006	29.98257831	37.831822	9.135022222	4.551244444
4	25.04.2006	29.9893339	37.8328887	9.133288889	4.539244444
5	26.04.2006	29.9968006	37.8399999	9.131866667	4.535022222
6	27.04.2006	29.95946719	37.8460444	9.128888889	4.534622222
7	28.04.2006	30.68942265	37.9043554	9.115955556	4.556088889
8	29.04.2006	32.12124453	39.5239113	11.72391111	4.568444444
9	30.04.2006	31.97013359	39.6924444	11.71351111	5.323377778
10	01.05.2006	31.58156685	39.7730585	11.19560556	5.578359722

Figure 16: The sheet ‘*WasAnt*’ in ‘*Parametrisierung.xls’*



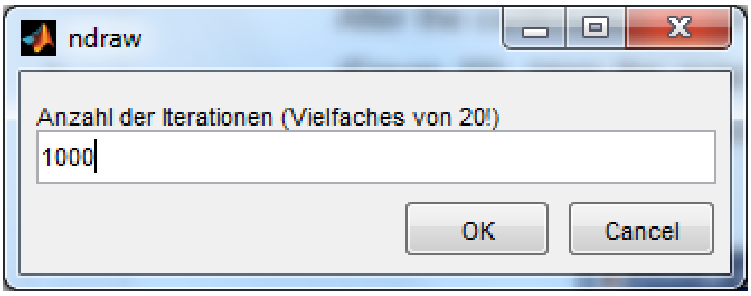
	A	B	C	D	E
1	**NO3-Konzentration (mg/l)**				
2	**Stufe (cm)** **Tag**	35	60	90	150
3	24.04.2006	-9999	-9999	-9999	73.33
4	25.04.2006	-9999	-9999	-9999	-9999
5	26.04.2006	-9999	-9999	-9999	-9999
6	27.04.2006	-9999	-9999	-9999	-9999
7	28.04.2006	-9999	-9999	-9999	-9999
8	29.04.2006	-9999	-9999	-9999	-9999
9	30.04.2006	-9999	-9999	-9999	-9999
10	01.05.2006	-9999	-9999	-9999	-9999
11	02.05.2006	-9999	-9999	5.70	68.70
12	03.05.2006	-9999	-9999	-9999	-9999

Figure 17: The sheet ‚*Nitratkonzentrationen*‘ in ‚*Parametrisierung.xls*‘

The Excel file *'Parametrisierung.xls'* has to be completed and saved before Calisto is started.

3.3.2 Pop up windows

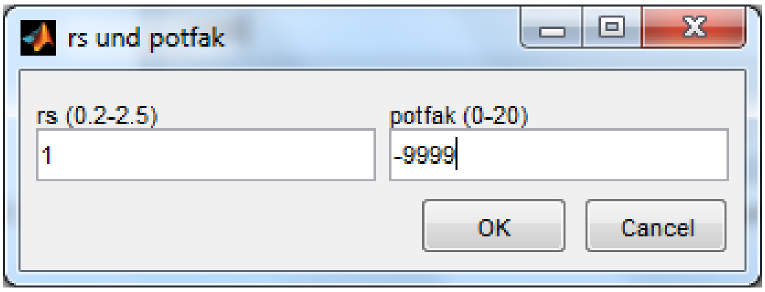
The user can start Calisto by executing the MatLab script *'runCalisto.m'*. Next, a pop-up window 'ndraw' appears (Figure 18). Here the parameter '*AMALGAMPar.ndraw'* can be defined, which describes the number of function evaluations (2.2.2). In the text box a preconfigured number of a maximum of 1000 function evaluations ('Iterationen') is shown. This number can be changed as required, but needs to be a multiple of the population size N (usually 20; 2.2.2).



Anzahl der Iterationen (Vielfaches von 20!)
1000

Figure 18: Pop-up window for the specification of the maximum number of function evaluations

After the confirmation of the input with 'OK', another pop-up window '*potfak und rs'* appears (Figure 19). Here the evaporation coefficient '*rs'* and the mineralization multiplier '*potfak'* (2.1.1 and 2.1.3) are specified. If '*rs*' or '*potfak*' shall be optimized by Calisto a value '-9999' has to be inserted.



rs (0.2-2.5)
1
potfak (0-20)
-9999

Figure 19: Pop-up window for the specification of the evaporation coefficient '*rs*' and the mineralization multiplier '*potfak*'. In this example '*rs*' should be optimized; therefore it is marked with '-9999'.

Moreover, the user can optionally define parameter search ranges as well as starting values for the optimization of the parameters. Therefore, more pop windows are integrated into the Calisto start, which are described in the following.

3.4 Starting Values and Ranges

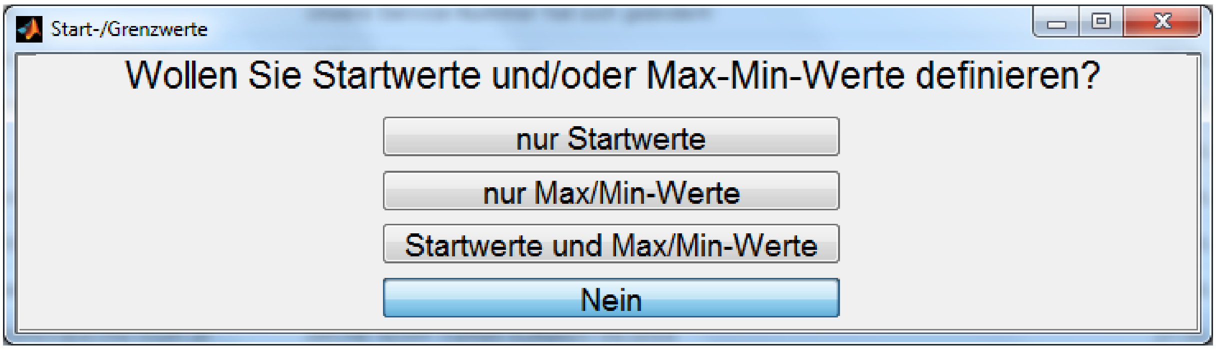
In Calisto, the starting values for the parameter estimation can either be generated by using Latin hypercube sampling. Or, if prior information is available (for example first estimation from transient experiments with soil samples under controlled conditions), starting values can be specified (2.2.1). Moreover, to guarantee the fitting of model parameters that make physical sense, realistic search range limits should be set (Ndiritu, 2009). As default there are standard ranges used for the calibration (Table 1). The lower and upper bounds of the soil parameters were specified based on available literature (Wöhling et al., 2008; Vrugt and Robinson, 2007; Wöhling and Vrugt, 2011; Mertens et al., 2006). The ranges of the plant parameters were set according to expert knowledge (Feichtinger, 2015).

If no further details are specified, Calisto uses the default values from Table 1 and automatically generates starting values within these bounds.

Table 1: Default values for the parameter ranges in Calisto

	parameter	**Abbreviation/ name in code**	**min**	**max**	**unit**
	Evaporation coefficient	rs	0.2	2.5	-
	Mineralisation multiplier	potfak	0	20	-
soil parameters	θ_s	thetaS	0.2	0.8	$m^3\ m^{-3}$
	θ_r	thetaR	0.0	0.3	$m^3\ m^{-3}$
	$\propto_1, \propto_2$	alpha1,2	0.0001	20	m^{-1}
	n_1, n_2	n1,2	1.1	9.0	-
	w	w	0	1	-
	K_s	Ks	0.1	100000	mm d^{-1}
	τ	tau	-3	3	-
plant parameters	Extinction coefficient	ext	0.3	0.75	-
	Leaf area	bfgw	0.0015	0.0050	ha kg^{-1}
	Pot. Plant height	hgt	0.3	2.5	m
	Stomatal resistance	rs	0.1	2.0	s cm^{-1}
	Assimilation	as	7.0	90	kg CH_2O ha^{-1} h^{-1}
	Pot. Root length	rlg	4	40	dm
	Pot. Root density	rdf	1.0	9.0	cm cm^{-3}
	Pot. Leaf width	lfw	0.5	20.0	cm
	sum of accumulated photothermal units at riping	ripe	500	4500	PTU
	LAI at emergence	lai0	0.01	0.80	-
	Critical day length	cdayl	8	10	h

However, Calisto also offers the possibility to enter specific starting values and / or ranges for the actual project. After the pop up windows asking for '*ndraw*' and '*potfak und rs'* (Figure 18 and Figure 19), there appears another window asking if the user wants to define starting values and/or ranges (Figure 20):



Start-/Grenzwerte

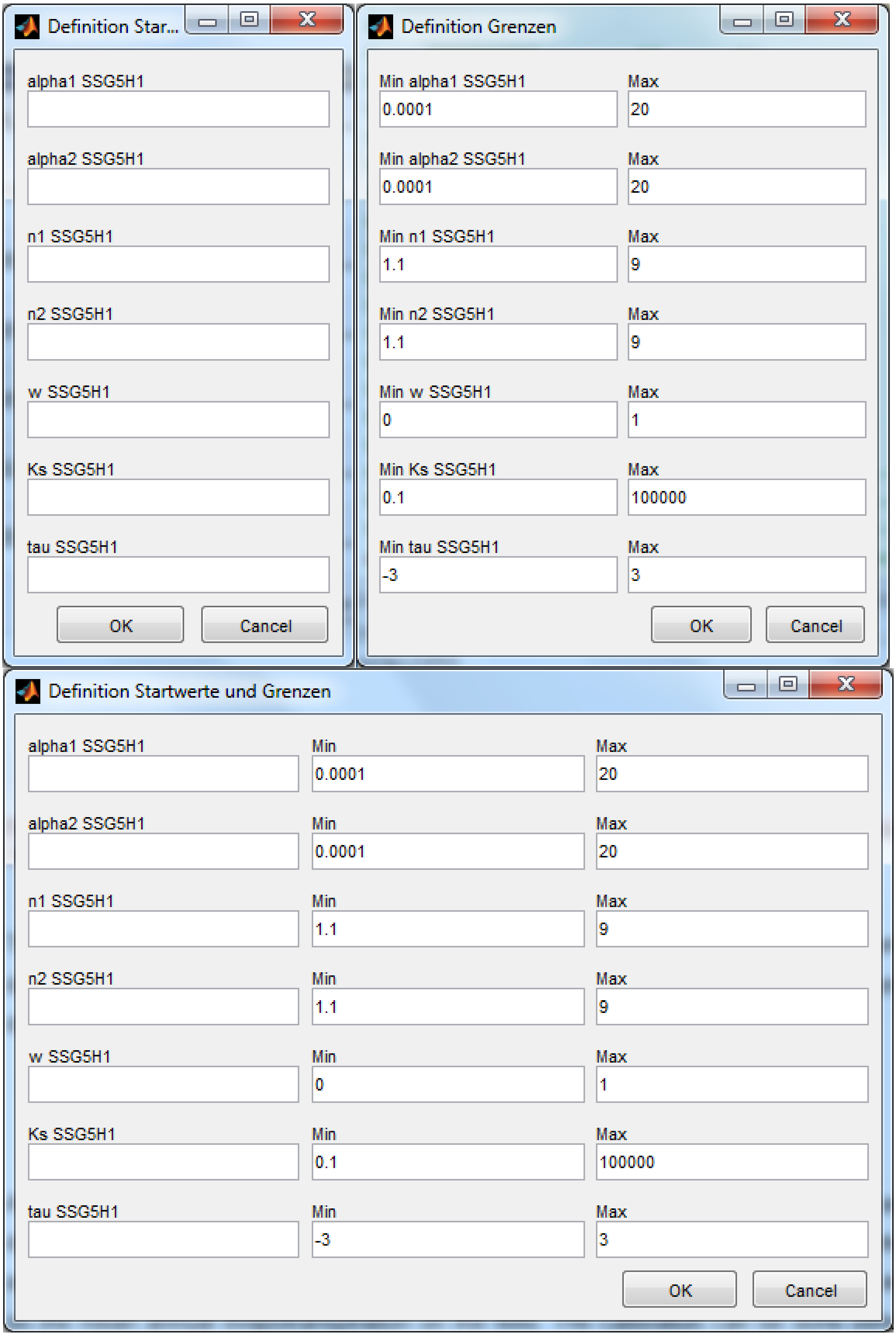
Wollen Sie Startwerte und/oder Max-Min-Werte definieren?

Figure 20: Pop-Up window asking if starting values and/or parameter ranges should be predefined

The window offers four choices:

- 'nur Startwerte' = definition of starting values only
- 'nur Max/Min-Werte' = definition of ranges only
- 'Startwerte und Max/Min-Werte' = definition of both starting values and ranges
- 'Nein' = no definition

Clicking on an option leads to the popping up of different dialog boxes as shown in Figure 21. Clicking 'Nein' denies the question if starting values and/or parameter values should be predefined, therefore the default values are used and no further window appears.



Definition Star...

alpha1 SSG5H1

alpha2 SSG5H1

n1 SSG5H1

n2 SSG5H1

w SSG5H1

Ks SSG5H1

tau SSG5H1

Definition Grenzen

Min alpha1 SSG5H1	Max
0.0001	20
Min alpha2 SSG5H1	Max
0.0001	20
Min n1 SSG5H1	Max
1.1	9
Min n2 SSG5H1	Max
1.1	9
Min w SSG5H1	Max
0	1
Min Ks SSG5H1	Max
0.1	100000
Min tau SSG5H1	Max
-3	3

Definition Startwerte und Grenzen

alpha1 SSG5H1	Min	Max
	0.0001	20
alpha2 SSG5H1	Min	Max
	0.0001	20
n1 SSG5H1	Min	Max
	1.1	9
n2 SSG5H1	Min	Max
	1.1	9
w SSG5H1	Min	Max
	0	1
Ks SSG5H1	Min	Max
	0.1	100000
tau SSG5H1	Min	Max
	-3	3

Figure 21: MatLab dialog boxes for the definition of starting values and/or ranges of the parameters that should be optimized in Calisto

The windows list all parameters that have been marked with ‘-9999’. For each parameter definition there is one or more input field(s) for defining the starting values and/or ranges. By default the ranges from Table 1 are inserted and can be adjusted manually.

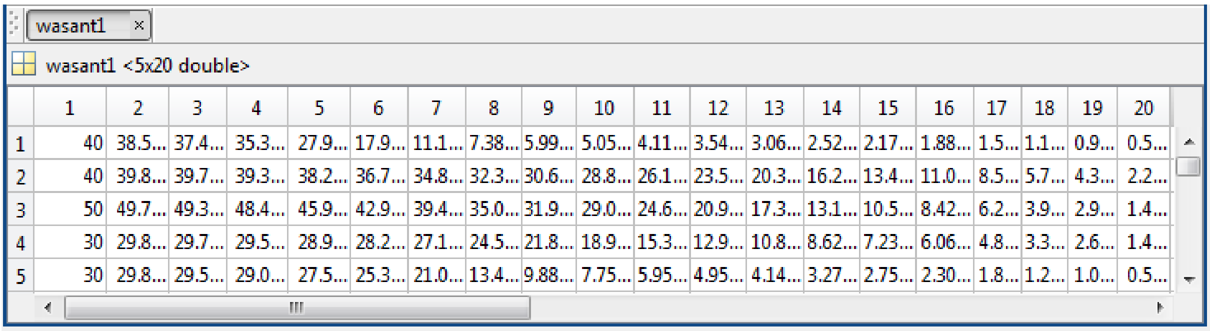
3.5 Data Transfer between MatLab and C#

The inputs for SIMWASER/STOTRASIM are defined in MatLab in special formats that are readable for C#: Parameters ‘*wasant’* and ‘*k’* are .NET arrays, ‘*matpot’, ‘penetro’* and ‘*plantpar’* are .NET lists and ‘*mineralisierungsmult’* and ‘*verdunstungsbeiwert’* are doubles. In C#, the inputs are partly restructured and handed on between several methods (3.5.1).

The model outputs ‘*wasseranteile1’, ‘saugspannungen1’, ‘stoffkonzentrationen1’, ‘ertraege1’* and *‘evapotrans1’* are handed over from C# to MatLab as .NET arrays, which have to be transformed into MatLab arrays.

3.5.1 Transfer of parameters from MatLab to C#

In the following Figure 22 there is an example of a MatLab matrix (wasant1) containing the **water contents** of the soil types SL3G4H3, LS3G3h2, LS4G3H2, SL3G2h1 and SSG5H1. Each row represents a soil type and gives the water content (in Vol-%) for 20 steps of water tension (from 0.01 to 1000000 cbar), which have been calculated by using formula 16 (3.2).



wasant1

wasant1 <5x20 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	40	38.5...	37.4...	35.3...	27.9...	17.9...	11.1...	7.38...	5.99...	5.05...	4.11...	3.54...	3.06...	2.52...	2.17...	1.88...	1.5...	1.1...	0.9...	0.5...
2	40	39.8...	39.7...	39.3...	38.2...	36.7...	34.8...	32.3...	30.6...	28.8...	26.1...	23.5...	20.3...	16.2...	13.4...	11.0...	8.5...	5.7...	4.3...	2.2...
3	50	49.7...	49.3...	48.4...	45.9...	42.9...	39.4...	35.0...	31.9...	29.0...	24.6...	20.9...	17.3...	13.1...	10.5...	8.42...	6.2...	3.9...	2.9...	1.4...
4	30	29.8...	29.7...	29.5...	28.9...	28.2...	27.1...	24.5...	21.8...	18.9...	15.3...	12.9...	10.8...	8.62...	7.23...	6.06...	4.8...	3.3...	2.6...	1.4...
5	30	29.8...	29.5...	29.0...	27.5...	25.3...	21.0...	13.4...	9.88...	7.75...	5.95...	4.95...	4.14...	3.27...	2.75...	2.30...	1.8...	1.2...	1.0...	0.5...

Figure 22: The MatLab water content matrix (wasant1) for the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1

To make the matrix ‘*wasant1’* transferrable to C#, it has to be converted into a .NET array (‘*wasant’*):

```
wasant = NET.convertArray(wasant1, 'System.Double');
```

The same procedure is done for the unsaturated hydraulic conductivity values of every soil type. Formula 17 (3.2) is used for calculating the unsaturated hydraulic conductivities of each soil type and creating a matrix similar to Figure 22, but filled with hydraulic conductivities in hPa (‘*k1*’). This matrix is then converted into a .NET array (‘*k*’) as well.

The **matric potentials** (20 points from 0.01 to 1000000 cbar) are the same for every project and every soil type, therefore they do not need to be read in and there is no necessity for a two dimensional matrix. In this case a .NET List can be used. At first a MatLab array with the values is created. The MatLab array cannot be converted to a .NET List directly, so a .NET List of 20 empty elements in double (64-bit floating-point values) precision has to be created first and is then filled with the data afterwards. The corresponding MatLab code is:

```
matpot_array = [0.01, 0.1, 0.2, 0.4, 1.0, 2.0, 4.0, 10.0, 20.0, 40.0, 100.0, 200.0, 400.0, 1000.0,
2000.0, 4000.0, 10000.0, 40000.0, 100000.0, 1000000.0];

A_matpot = NET.convertArray(matpot_array, 'System.Double');

matpot = NET.createGeneric('System.Collections.Generic.List', {'System.Double'},20);

matpot.AddRange(A_matpot);
```

As already mentioned, the **penetrometer resistances** are not relevant for the current version of SIMWASER/STOTRASIM. Therefore the MatLab list ‘*penetro’* is filled with placeholder numbers (f.e. ‘99.9’). The **plant parameters** are read in from the sheet ‘Pflanzenparameter’ in the Excel file ‘*Parametrisierung.xls*’, not predefined parameters are created by AMALGAM and all values are stored in a .NET list similar to the list of matric potentials. The mineralization multiplier ‘*potfak’* and the evaporation coefficient *‘rs’* are stored as numeric variables in double precision.

When calling the method SIMWASER/STOTRASIM by MatLab, all arrays (water content ‘*wasant’*, hydraulic conductivity ‘*k*’), lists (matric potential ‘*matpot’*, penetrometer resistance ‘*penetro’,* plant parameters ‘*plantpar*’) and double values (mineralization multiplier ‘*mineralisierungsmult’* and evaporation coefficient *‘verdunstungsbeiwert’)* can be passed as variables (Figure 9).

After being passed into C#, the .NET matrices ‘*wasant’, ‘k’* and ‘*penetro’* are restructured to Generic Lists of Lists. Each list represents one soil type and is then subsumed to three Lists of Lists, one for the water content, one for the hydraulic conductivity and one for the penetrometer resistance. The Lists of Lists ‘*wasant’, ‘k’* and ‘*penetro’* as well as the List ‘*matpot’* and the doubles ‘*rs’* and ‘*potfak’* are then handed over starting with the method

‘*Programm’,* from method to method to areas in the code where they can replace values that would otherwise have been read in from input files.

For example *‘matpot’*, *‘wasant’*, ‘*k*’ and ‘*penetro*’ are handed over from method ‘*Programm’* to the method ‘*einfacheAusführung’* to the method ‘*initProjekt’* to the method ‘*leseBodenkennwertedateien’* in the file ‘*Input.cs’* where originally the soil layer files had been read in. The C# code section for reading in the .dat files of the soil layers was deleted and instead the values of the List ‘*matpot*’ and the Lists of Lists ‘*wasant*’, ‘*k*’ and ‘*penetro*’ are saved as variables there. The corresponding C# code is:

Previous Code:

```
public static void
leseBodenkennwertedateien(Bodenprofil bp)




 {
        System.Console.WriteLine("Lese
Bodenkennwertedateien");

        foreach (BodenWasserhaushalt boden in
bp.schichten)
        {
           string zeile;
           try
           {
              System.IO.StreamReader datei = new
System.IO.StreamReader(Config.pfadDaten +
Config.inputPfad + boden.bodenart + ".dat",
System.Text.Encoding.Default);
              datei.ReadLine(); //Name Bodenart
              zeile =
mehrfacheLeerzeichenEntfernen(datei.ReadLine());
              string[] parts = zeile.Split(':');
              int standardkurven =
Convert.ToInt32(parts[1]);

              for (int j = 0; j < standardkurven; j++)
              {
                 Standardkurven kurve = new
Standardkurven();
                 zeile =
mehrfacheLeerzeichenEntfernen(datei.ReadLine());
                 parts = zeile.Split(':');
                 kurve.porenvolumen =
Convert.ToInt32(parts[1]);
                 boden.standardkurven.Add(kurve);
              }


              datei.ReadLine();
              datei.ReadLine();
              datei.ReadLine();
              while ((zeile = datei.ReadLine()) != null)
              {
                 parts = zeile.Split(' ');
```

New Code:

```
public static void
leseBodenkennwertedateien(Bodenprofil bp,
List<double> matpot, List<List<double>> wasant,
List<List<double>> k, List<List<double>> penetro)


     {
        System.Console.WriteLine("Lade
Bodenkennwerte");

        foreach (BodenWasserhaushalt boden in
bp.schichten)
        {

           try
           {
              Standardkurven kurve = new
Standardkurven();
              boden.standardkurven.Add(kurve);









              for (int i = 0; i < 20; i++)
              {
                 boden.psi_input.Add(matpot[i]);

                 if (boden.isoil == i + 1)
                 {
```

boden.psi_input.Add(toDouble(parts[0]));
for (int j = 0; j < standardkurven; j++)
{	

boden.standardkurven[j].wassergehalt.Add(toDouble(parts[1 + 3 * j]));	boden.standardkurven[0].wassergehalt.AddRange(wasant[i]);
boden.standardkurven[j].kapillareLeitfaehigkeit.Add(toDouble(parts[2 + 3 * j]));	boden.standardkurven[0].kapillareLeitfaehigkeit.AddRange(k[i]);
boden.standardkurven[j].penetrometerwiderstand.Add(toDouble(parts[3 + 3 * j]));	boden.standardkurven[0].penetrometerwiderstand.AddRange(penetro[i]);
	boden.standardkurven[0].porenvolumen = wasant[i][0];
	boden.porenvolumen = Convert.ToInt16(wasant[i][0]);

}
}
}
catch (Exception)
{
throw new Exception("Fehler beim Lesen der Datei ('" + boden.bodenart + ".dat')!");
}
}
}
	}
	}
	}
	catch (Exception)
	{
	throw new Exception("Fehler beim Lesen der Datei ('" + boden.bodenart + ")");
	}
	}
	}

Also, the plant parameters, the mineralization multiplier ‘*potfak*’ and the evaporation coefficient ‘*rs*’ are handed on in the same way.

3.5.2 Transfer of model outputs from C# to MatLab

In the public class ‘*OutputKlasseProjekt*’ the calculated model output of water content, matric potential and nitrate concentrations are prepared for being written into the output files.

In the class ‘*OutputKlasseProjekt’* three new methods (one for water content, matric potential and nitrate concentration each) were established to create matrices of the simulated values and then carry them on to the method *‘Programm’* (vice versa to the process in 3.5.1). The code below shows the C# code for the **water contents**. First a list of lists (‘*ListOfListsWasAnt’*) is created, whereby each of the sublists includes the water contents for all compartment depths (‘*Stufen’*) for one day (‘*OutputKlasseTag’ ‘tag’*). The final list of lists includes the whole time period of the project.

```
public double[,] schreibeWasseranteile()
        {
            List<List<double>> ListOfListsWasAnt = new List<List<double>>();
            Stufen = bodenprofilStufe;
          //  Stufen.Insert(0, 0);
            ListOfListsWasAnt.Add(Stufen);

            foreach (OutputKlasseTag tag in tagwerte)
            {
                double datum = tag.datum.ToOADate();
                tag.wasseranteile.Insert(0, datum);
                ListOfListsWasAnt.Add(tag.wasseranteile);
            }
            wasseranteile = new double[ListOfListsWasAnt.Count, ListOfListsWasAnt[1].Count];

            for (int i = 0; i < ListOfListsWasAnt.Count; i++)
            {
                for (int j = 0; j < ListOfListsWasAnt[i].Count; j++)
                {
                    wasseranteile[i, j] = ListOfListsWasAnt[i][j];
                }
            }
            return wasseranteile;
        }
```

The list of lists is then transformed to a matrix ‘*wasseranteile*’ that has the appropiate format to be handed over to MatLab. The matrix ‘*wasseranteile’* is handed on from method ‘*OuputklasseProjekt’* to method ‘*ProjektOutput’* to class ‘*Projekt’* to class ‘*Program’* and finally to method *Programm* where it serves as an output variable. Similar procedure is done for the **matric potentials** and **nitrate concentrations**.

In the same way, the variables ‘*ertraege*’ (**yields**) and ‘*mittlereEvapotrans*’ (**average annual evapotranspiration**) are taken from the code and handed on from method to method so that they can finally be called by the method ‘*Programm*’.

3.6 Objective Functions

As AMALGAM is a multiobjective optimization tool, several objective functions can be defined. The model can be calibrated on the time series of water content, matric potential and nitrate concentration of the leachate at various depths, on the yield of the crops and/or on the mean annual evapotranspiration on the field. The calibration can be done either on one or a multiple of these variables. Calisto always calibrates on all data that is available in the sheets '*Erträge*', '*Evapotranspiration*', '*MatPot*', '*WasAnt'* and '*Nitratkonzentrationen*' of the Excel file '*Parametrisierung.xls'* (3.3.1). Measurements that should not be used for calibration have to be deleted from the file. The definition of the objective functions is coded in the Calisto script '*CalcOF.m',* which was taken over directly from AMALGAM (2.2.2).

The Nash-Sutcliffe efficiency (NSE; 2.4) was chosen as the objective function (OF) for the calibration on water content, matric potential and nitrate concentrations. As AMALGAM is designed for minimizing the objective function, the formula had to be reformulated slightly (NSE$_{adapt}$).

$$NSE_{adapt} = \frac{\sum_{t=1}^{n}(O_t - P_t)^2}{\sum_{t=1}^{n}(O_t - \bar{O})^2} \tag{18}$$

where n is the number of observations, O_t is the observed value at time step t, P_t is the corresponding model predicted value, and $\bar{O}$ is the mean value of all observations.

So for each observed variable i and for each observed soil depth s that should be considered for optimization the following function is minimized:

$$OF\ (\beta, i, s) = \frac{\sum_{t=1}^{n}(O_{ist} - P_{ist})^2}{\sum_{t=1}^{n}(O_{ist} - \overline{O_{\iota s}})^2} \tag{19}$$

where OF is the objective function, β is the vector of input parameters, O_{ist} is the measured value of the i th variable at the s th location (depth) and the t th time and P_{ist} is the corresponding simulated value and $\overline{O_{\iota s}}$ is the mean value of the observed variable i in this depth s over the time period. The observed variables i can be measured matric potentials, water contents and/or nitrate concentrations of the soil water in various depths s. The time t defines the observed moment. The outputs are given daily in SIMWASER/STOTRASIM, but only days for which proper measurement values are available are considered in the calculation.

For the calibration on the yield and the evapotranspiration, the method of least squares was chosen as the objective function. So for the crop rotation the following function is minimized:

$$OF(\beta) = \sum_{c=1}^{m} (O_c - P_c)^2 \tag{20}$$

where OF is the objective function, β is the vector of input parameters, O_c is the measured value of the yield of crop c and P_c is the corresponding simulated value.

3.7 Calisto Results and Output

As mentioned in 2.2.2, all computed parameter sets and the corresponding function evaluation values are saved in the MatLab matrix '*ParSet'.* The result provided by AMALGAM is not only one parameter set, but a population of parameter sets which in the best case should form a Pareto front.

The user of Calisto, however, wants to receive a result of one parameter set that he/she can use for the model simulation. Therefore Calisto has to choose one of the resulting parameter sets in ParSet as solution. For this master thesis a common approach was taken, which is to normalize each objective function and select a solution by minimizing the Euclidian distances (Wöhling and Vrugt, 2011). The MatLab code for the implementation of this approach is shown below:

```
[x,y]=size(ParSet);
minF = min(ParSet(:,AMALGAMPar.n+2:end)); minF = minF(ones(x,1),:);
maxF = max(ParSet(:,AMALGAMPar.n+2:end)); maxF = maxF(ones(x,1),:);
spanneF = maxF-minF;

% Now determine Euclidean distance from these optimal solutions
[T] = sqrt(sum((((ParSet(:,AMALGAMPar.n+2:end)-minF))./spanneF.^2),[2]));

% Sort T and determine idx
[dummy,idx] = sort(T);

% Now sort ParSet
OPT = ParSet(idx(1),1:AMALGAMPar.n);
```

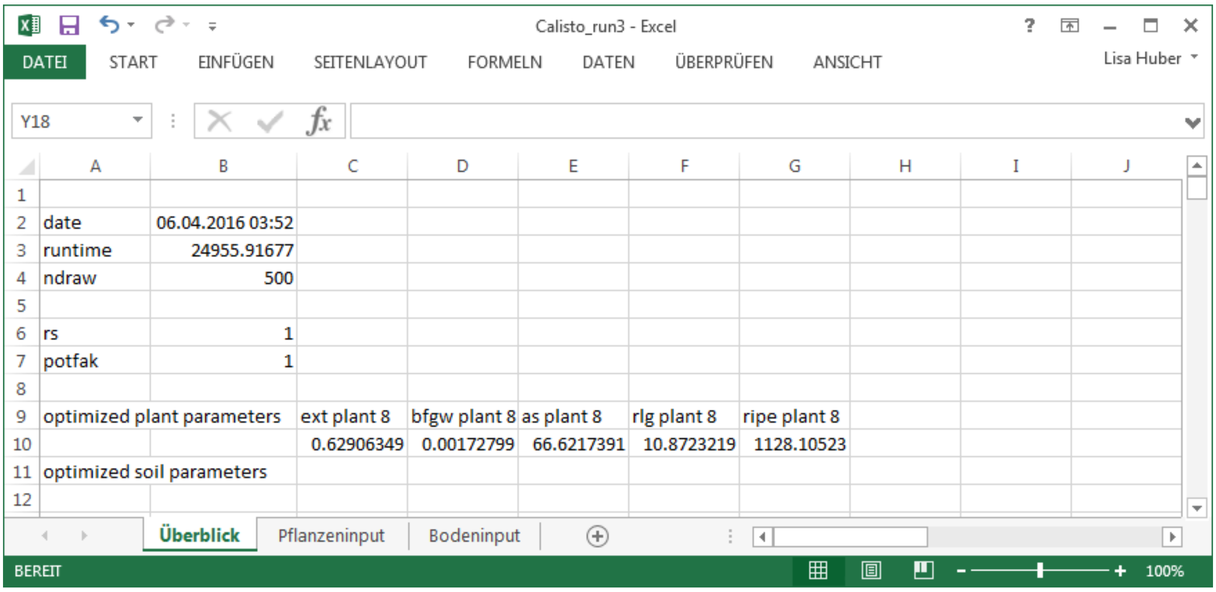
```
ErgebnisOF = ParSet(idx(1),1:(AMALGAMPar.n+1+AMALGAMPar.nobj));

evalstr = ['ModPred = ',ModelName,'(OPT,Extra, Measurement);']; eval(evalstr);
```

At first, the minimum results for each respective objective function are determined (‘*minF’*). ‘*minF’* forms a set of the best function evaluations that could be reached. Then, the Euclidean distance (straight-line distance between two points in Euclidean space) between the normalized objective function evaluations of each parameter set in ‘*ParSet’* and ‘*minF’* is calculated. ‘*minF* ‘can be considered as the best reachable solution. ‘*minF*’ is the zero position in an Euclidean space where each objective function forms a dimension. The Euclidean distances are sorted and the shortest distance is considered as the best one. So the parameter set that results in objective functions that are overall closest to ‘*minF’*, the best reachable solution or the point zero, is chosen (‘*OPT’*). After determination of this best parameter set ‘*OPT’*, a simulation with this parameter set is started once again (evaluated with the command *eval()*). The STOTRASIM results of this simulation is stored in the folder ‘ERG’ (2.1.3).

Moreover, Calisto creates an Excel file with the result summary (Figure 23). The Excel file is given the file name ‘*Calisto_run1’* with sequential numbering. The result file consists of three sheets:

- ‘*Überblick*’ (overview)
- ‘*Pflanzeninput*’ (plant inputs)
- ‘*Bodeninput*’ (soil inputs)

The overview sheet ‘*Überblick*’ shows the actual date of the optimization run as well as the run time, the selected amount of iterations and the parameter values for ‘*rs*’ and ‘*potfak*’. ‘*Rs*’ and ‘*potfak*’ can either be predefined in the pop-up window (Figure 18 and Figure 19) or it may be optimized by AMALGAM. Optimized ‘*potfak*’ and ‘*rs*’ can usually be recognized by their high number of decimal digits. In Figure 23 ‘*rs*’ and ‘*potfak*’ had been predefined. The following lines show the optimized plant parameters and the optimized soil parameters.



	A	B	C	D	E	F	G	H	I	J
1										
2	date	06.04.2016 03:52								
3	runtime	24955.91677								
4	ndraw	500								
5										
6	rs	1								
7	potfak	1								
8										
9	optimized plant parameters		ext plant 8	bfgw plant 8	as plant 8	rlg plant 8	ripe plant 8			
10			0.62906349	0.00172799	66.6217391	10.8723219	1128.10523			
11	optimized soil parameters									
12										

Figure 23: Screenshot of the sheet ‘*Überblick*’ of the Excel file ‘*Calisto_run1.xls’*

Moreover the sheet ‘*Überblick*’ contains graphs showing the discrepancies between the outputs of the optimized simulation and the measurements. Depending on whether measurement values are given in the file ‘*Parametrisierung.xls’* (3.3.1), graphs are drawn for the yields (Figure 24), the mean annual evapotranspiration, the water content in several depths (Figure 25), matric potentials in several depths and the nitrate concentrations of the soil water in several depths.

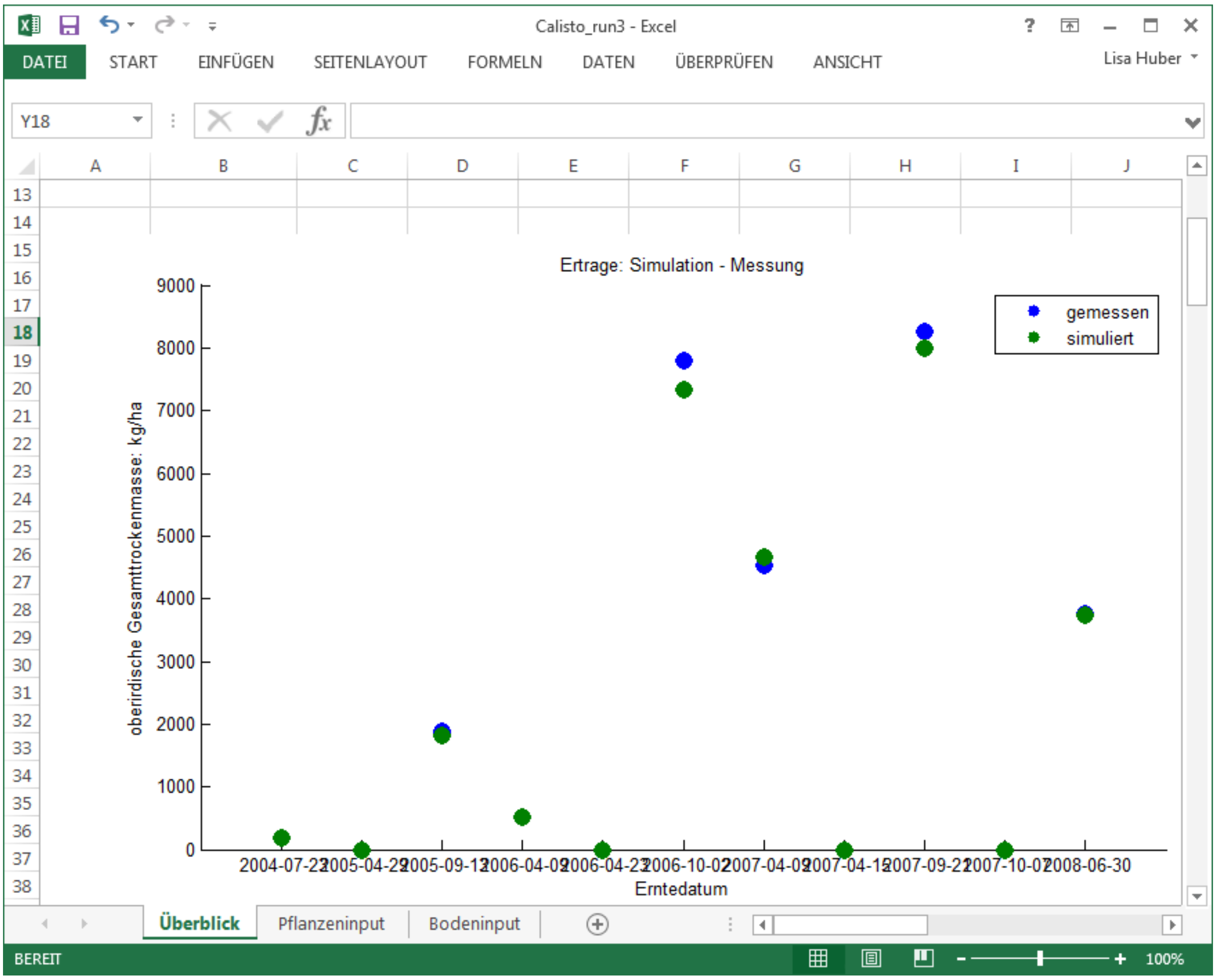


Figure 24: Graph with comparison of simulated (green) and measured (blue) yields (dry matter) in the sheet ‘*Überblick*’ in ‘*Calisto_run1.xls’*

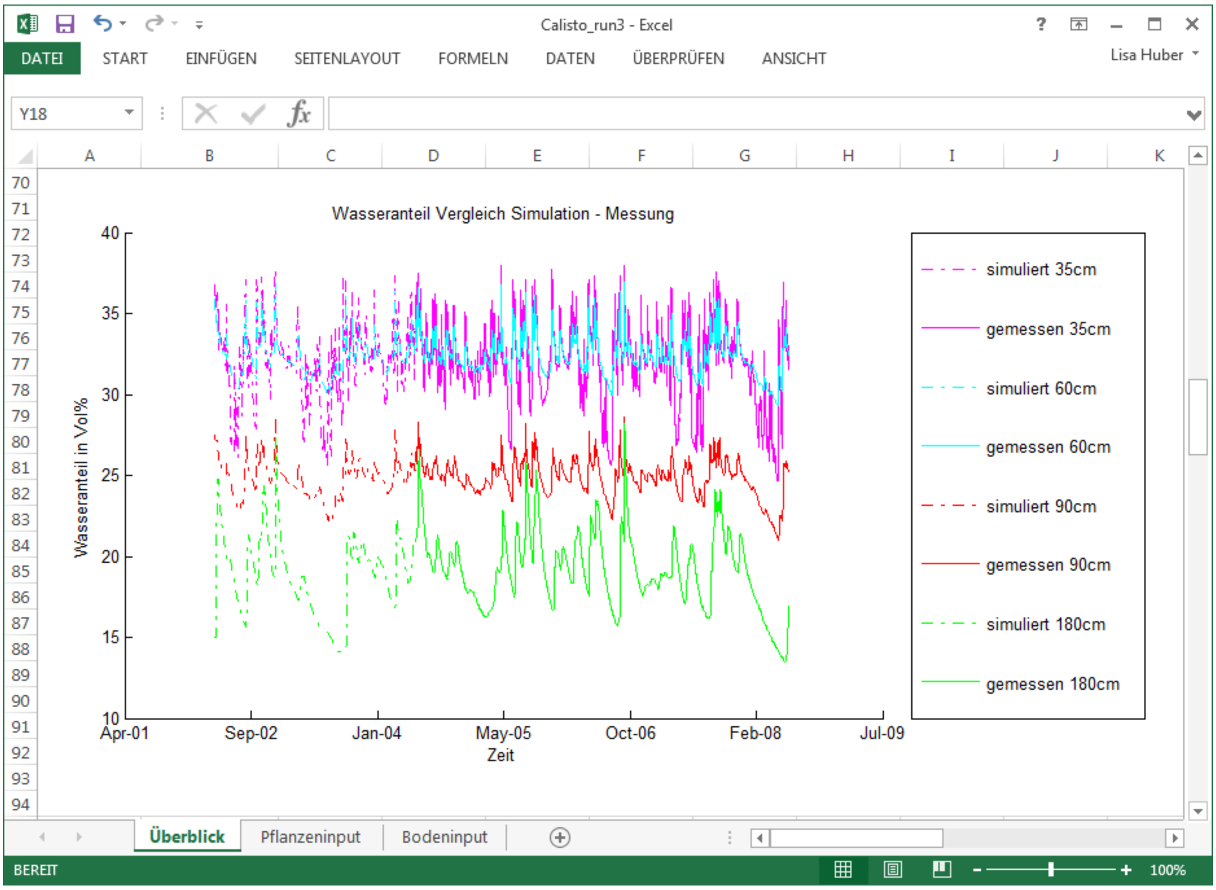
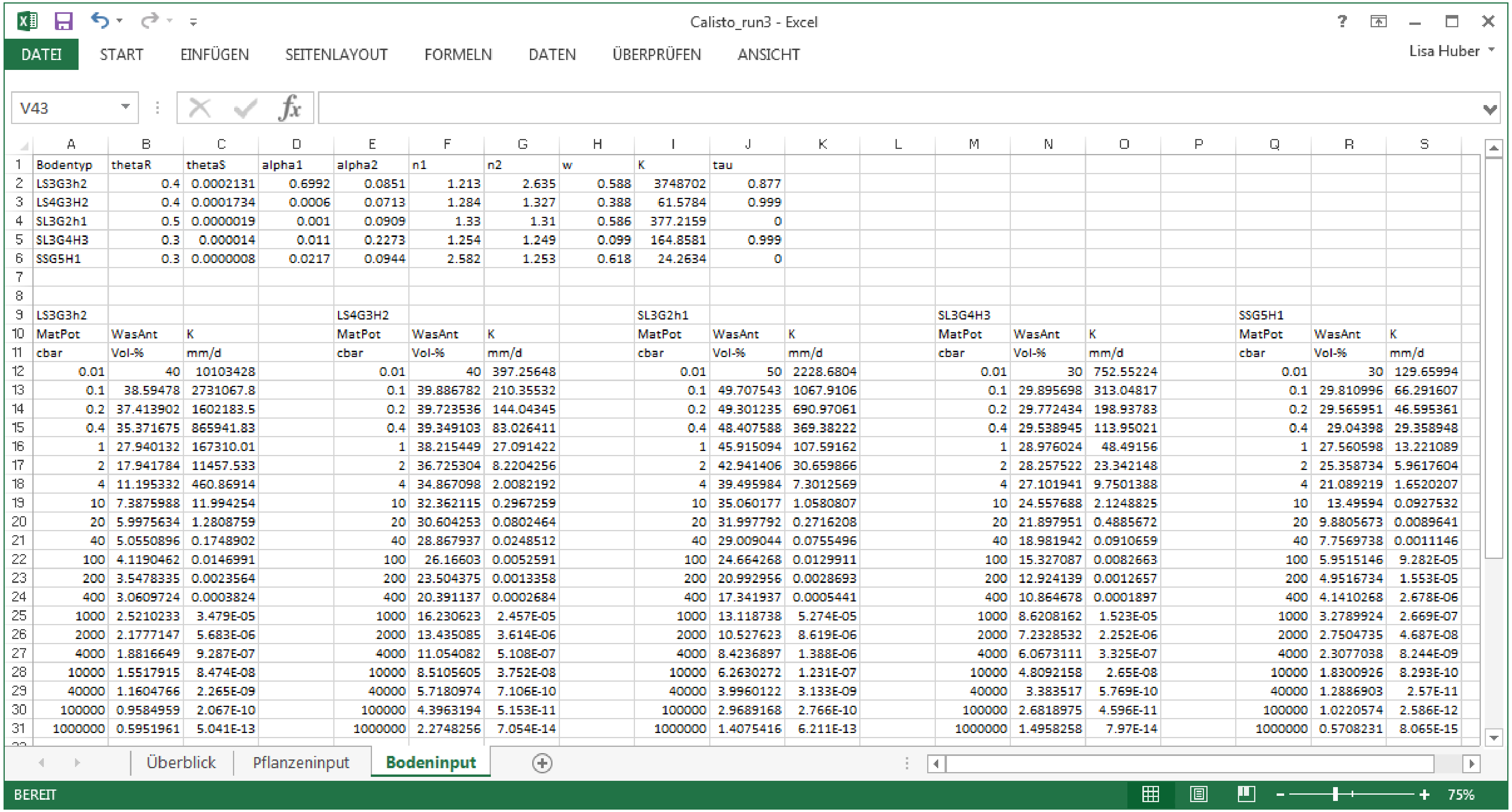


Figure 25: Graph with comparison of simulated (dotted) and measured (solid) time series of water content in the sheet '*Überblick*' in '*Calisto_run1.xls'*

The sheets of plant and soil parameters contain the complete final set of plant and soil parameters including the optimized parameters that are used for the final simulation (Figure 11). The sheet of the soil parameters contains in addition to the bvGM parameters also the corresponding 'standard curves' (Figure 26).



	A	B	C	D	E	F	G	H	I	J
1	Bodentyp	thetaR	thetaS	alpha1	alpha2	n1	n2	w	K	tau
2	LS3G3h2	0.4	0.0002131	0.6992	0.0851	1.213	2.635	0.588	3748702	0.877
3	LS4G3H2	0.4	0.0001734	0.0006	0.0713	1.284	1.327	0.388	61.5784	0.999
4	SL3G2h1	0.5	0.0000019	0.001	0.0909	1.33	1.31	0.586	377.2159	0
5	SL3G4H3	0.3	0.000014	0.011	0.2273	1.254	1.249	0.099	164.8581	0.999
6	SSG5H1	0.3	0.0000008	0.0217	0.0944	2.582	1.253	0.618	24.2634	0
7										
8										

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
9	LS3G3h2				LS4G3H2				SL3G2h1				SL3G4H3				SSG5H1		
10	MatPot	WasAnt	K		MatPot	WasAnt	K		MatPot	WasAnt	K		MatPot	WasAnt	K		MatPot	WasAnt	K
11	cbar	Vol-%	mm/d		cbar	Vol-%	mm/d		cbar	Vol-%	mm/d		cbar	Vol-%	mm/d		cbar	Vol-%	mm/d
12	0.01	40	10103428		0.01	40	397.25648		0.01	50	2228.6804		0.01	30	752.55224		0.01	30	129.65994
13	0.1	38.59478	2731067.8		0.1	39.886782	210.35532		0.1	49.707543	1067.9106		0.1	29.895698	313.04817		0.1	29.810996	66.291607
14	0.2	37.413902	1602183.5		0.2	39.723536	144.04345		0.2	49.301235	690.97061		0.2	29.772434	198.93783		0.2	29.565951	46.595361
15	0.4	35.371675	865941.83		0.4	39.349103	83.026411		0.4	48.407588	369.38222		0.4	29.538945	113.95021		0.4	29.04398	29.358948
16	1	27.940132	167310.01		1	38.215449	27.091422		1	45.915094	107.59162		1	28.976024	48.49156		1	27.560598	13.221089
17	2	17.941784	11457.533		2	36.725304	8.2204256		2	42.941406	30.659866		2	28.257522	23.342148		2	25.358734	5.9617604
18	4	11.195332	460.86914		4	34.867098	2.0082192		4	39.495984	7.3012569		4	27.101941	9.7501388		4	21.089219	1.6520207
19	10	7.3875988	11.994254		10	32.362115	0.2967259		10	35.060177	1.0580807		10	24.557688	2.1248825		10	13.49594	0.0927532
20	20	5.9975634	1.2808759		20	30.604253	0.0802464		20	31.997792	0.2716208		20	21.897951	0.4885672		20	9.8805673	0.0089641
21	40	5.0550896	0.1748902		40	28.867937	0.0248512		40	29.009044	0.0755496		40	18.981942	0.0910659		40	7.7569738	0.0011146
22	100	4.1190462	0.0146991		100	26.16603	0.0052591		100	24.664268	0.0129911		100	15.327087	0.0082663		100	5.9515146	9.282E-05
23	200	3.5478335	0.0023564		200	23.504375	0.0013358		200	20.992956	0.0028693		200	12.924139	0.0012657		200	4.9516734	1.553E-05
24	400	3.0609724	0.0003824		400	20.391137	0.0002684		400	17.341937	0.0005441		400	10.864678	0.0001897		400	4.1410268	2.678E-06
25	1000	2.5210233	3.479E-05		1000	16.230623	2.457E-05		1000	13.118738	5.274E-05		1000	8.6208162	1.523E-05		1000	3.2789924	2.669E-07
26	2000	2.1777147	5.683E-06		2000	13.435085	3.614E-06		2000	10.527623	8.619E-06		2000	7.2328532	2.252E-06		2000	2.7504735	4.687E-08
27	4000	1.8816649	9.287E-07		4000	11.054082	5.108E-07		4000	8.4236897	1.388E-06		4000	6.0673111	3.325E-07		4000	2.3077038	8.244E-09
28	10000	1.5517915	8.474E-08		10000	8.5105605	3.752E-08		10000	6.2630272	1.231E-07		10000	4.8092158	2.65E-08		10000	1.8300926	8.293E-10
29	40000	1.1604766	2.265E-09		40000	5.7180974	7.106E-10		40000	3.9960122	3.133E-09		40000	3.383517	5.769E-10		40000	1.2886903	2.57E-11
30	100000	0.9584959	2.067E-10		100000	4.3963194	5.153E-11		100000	2.9689168	2.766E-10		100000	2.6818975	4.596E-11		100000	1.0220574	2.586E-12
31	1000000	0.5951961	5.041E-13		1000000	2.2748256	7.054E-14		1000000	1.4075416	6.211E-13		1000000	1.4958258	7.97E-14		1000000	0.5708231	8.065E-15

Figure 26: Screenshot of the sheet '*Bodeninput*' with the optimized soil parameters in form of bimodal van Genuchten-Mualem parameters and 'standard curve'

4. Application of Calisto

4.1 Setting up a test site

To apply Calisto, data from the agricultural test site Wagna (located in Eastern Austria, within the Mur valley between Graz and Bad Radkersburg) were used (Klammler and Fank, 2014). The recorded data from one of the two monolithic lysimeters on the site was used. The SCIENCE-lysimeter (UMS, 2016) zahwith a depth of 2 m and an area of 1 m² is equipped with soil temperature probes, soil water samplers (suction cups), soil moisture (TDR) probes and tensiometers/matric sensors at four measuring depths (35, 60, 90 and 180 cm). Additionally, matric sensors are installed in the depths of 10 and 20 cm.

During the installation of the lysimeter in 2004, detailed soil samples were taken and analyzed by the Institute for Land and Water Management Research of the Austrian Federal Agency for Water Management. Based on particle size distributions and pF curves of the test site, the profile was subdivided into 5 layers.

Table 2 gives an overview of the soil horizons, soil textures and estimated diffusion/dispersion parameters of the layers in the lysimeter. The corresponding soil type was determined by the particle size distributions according to the Austrian texture triangle, combined with the classification according to the content of organic carbon (H) and gravel (G) (Ad-hoc-Arbeitsgruppe Boden, 2005).

Table 2: Soil horizons, textures and estimated diffusion/dispersion parameters for the layers in the lysimeter

depth	**horizon**	**soil type**	**average soil texture**				**diffusion/dispersion parameter**		
			clay	*silt*	*sand*	*gravel*	*a*[a]	*b*[b]	*λ*[c]
(cm)			(% mass)						
0-30	Ap	SL3G4H3	11	22	27	40	0.00560	10	2.53
30-60	B	LS3G3H2	20	34	25	22	0.00503	10	2.78
60-80	B	LS4G3H2	20	28	30	21	0.00500	10	3.00
80-120	B	SL3G2H1	9	26	57	9	0.00576	10	1.58
120-300	C	SSG5H1	0	1	36	62	0.00631	10	2.39

[a] a = 0.00633 – 0.000066 * clay content (in % mass) according to Duynisveld (1983)

[b] fixed with 10 according to Duynisveld (1983)

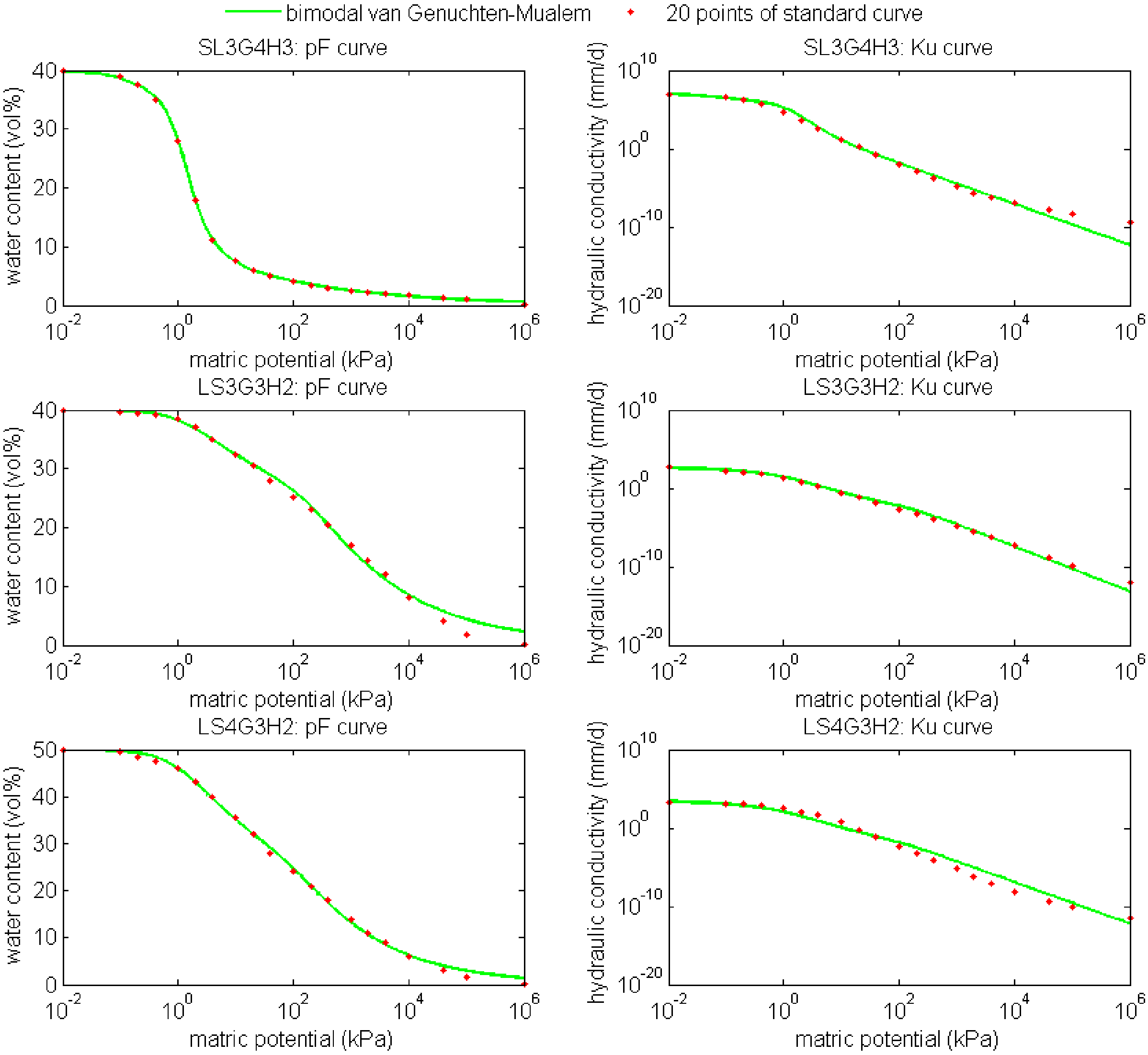
[c] λ = 0.0123 * sand content – 0.0178 * silt content + 0.122 * clay content + 0.0312 * gravel (in % mass) according to Murer (1998)

For the five soil types standardized soil characteristics were available from the Institute for Land and Water Management in form of ‘standard curves’. These standard curves are based on original studies of Stenitzer within the establishment of SIMWASER (Stenitzer, 1988). For the application in Calisto, bvGM parameters were created that approximate the ‘standard curves’ of the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1 (Table 3).

Table 3: Standardized bimodal van Genuchten-Mualem parameters for the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1

Soil type	**thetaS**	**thetaR**	**alpha1**	**alpha2**	**n1**	**n2**	**w**	**Ks**	**τ**
	m^3 *m^-	m^3 *m^{-3}	m^{-1}	m^{-1}	-	-	-	mm *d^{-1}	-
SL3G4H3	0.40	0.0002131	0.6992	0.0851	1.213	2.635	0.588	3748702	0.877
LS3G3h2	0.40	0.0001734	0.0006	0.0713	1.284	1.327	0.388	61	0.999
LS4G3H2	0.50	0.0000019	0.0010	0.0909	1.330	1.310	0.586	377	0.000
SL3G2h1	0.30	0.0000140	0.0110	0.2273	1.254	1.249	0.099	165	0.999
SSG5H1	0.30	0.0000008	0.0217	0.0944	2.582	1.253	0.618	24	0.000

Not all of the obtained model values are physically realistic. Figure 27, however, shows the accordance of the standard curves by Stenitzer (red dots) and the curves created by bvGM parameters (Table 3; green lines) for the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1.



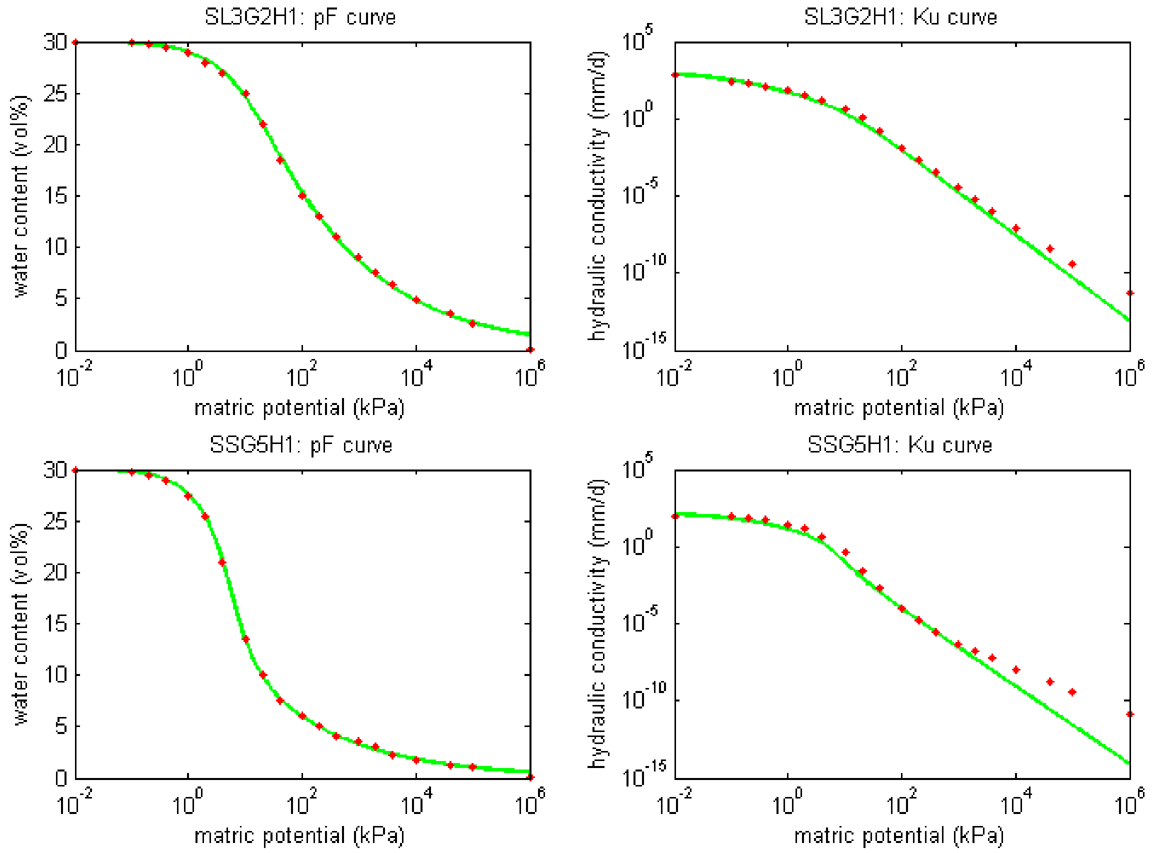


Figure 27: Accordance of standard curves (red dots) and curves created by standardized bimodal van Genuchten-Mualem parameters (green lines) for the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1

For plants that are grown on the study site standardized plant parameters provided from the Institute for Land and Water Management (Table 4) could be used. Main parameters for maize (plant code 8), pumpkin (plant code 42), and winter barley (plant code 3) are available. For the triticale in 2010/2011 the parameters for winter wheat (plant code 1) could be used, as the plant development was considered quite similar. The crop rotation also contained nitrogen fixating crops, which are very difficult to parameterize (Groenendijk et al, 2014). In general for all periods with nitrogen fixating seed mixtures the parameters for grassland (plant code 15) were used. The only exception is the mixture in the winter of 2006/2007, where the forage rye was dominant. Here the parameters for winter rye (plant code 5) were used. Another tough challenge was the parametrization of the rye grass that is being undersown in the pumpkin field. In SIMWASER/STOTRASIM it is not possible to use two different plant parameter sets at once, the sawing date of the rye grass can only be set after the harvesting of the pumpkin. Therefore the development stages of the rye grass could not be reconstructed realistically. One workaround was to set a period of grassland seed (plant code 15) after the harvest of the pumpkin.

Table 4: Standardized plant parameters for the use in SIMWASER/STOTRASIM for the plants winter wheat, winter barley, winter rye, maize, grass and pumpkin

code	name	ext	bfgw	hgt	rs	as	temp	rlg	rdf	root	lfw	ripe	lai0	luftbed	cdayl
		-	ha kg^{-1}	m	s cm^{-1}	kg CH_2O ha^{-1} h^{-1}	-	dm	cm cm^{-3}	-	cm	PTU	-	-	h
1	w.wheat	0.65	0.0025	0.8	0.3	17	1	20	6	1	0.5	2000	0.1	5	8
3	w.barley	0.60	0.0025	0.9	0.3	17	1	20	6	1	0.5	1500	0.1	5	10
5	w.rye	0.60	0.0030	0.9	0.3	20	1	15	6	2	0.5	1600	0.1	5	10
8	maize	0.65	0.0015	2.5	0.5	75	4	20	5	1	5.0	1100	0.05	5	10
15	grass	0.65	0.0035	0.9	0.3	7	1	35	6	2	0.5	2100	0.1	5	8
42	pumpkin	0.65	0.0025	0.5	0.9	11	2	15	5	2	20	1750	0.1	5	10

Ext = extinction coefficient, bfgw = leaf area, hgt = potential plant height, rs = stomatal resistance, as = assimilation, temp = temperature class, rlg = potential root length, rdf = potential root density, root = root density class, lfw = potential leaf width, ripe = sum of accumulated photothermal units at riping, lai0 = leaf area index at emergence, luftbed = minimum air content, cdayl=critical day length

For the test site standard values for parameters ‘*rs*’ and ‘*potfak*’ of 1 were used. As suggested in Groenendijk et al (2014), the calibration period was set from 2004 to 2008, whereas the validation period was set from 2008 to the end of 2011. Before the calibration period a lead time of two years was defined to help stabilizing the system.

The comparisons of the corrected time series of the measurement and the simulation results of SIMWASER/STOTRASIM with the standardized parameters from Table 3 and Table 4 and a standard ‘*potfak*’ of 1 and a ‘*rs*’ of 1 are visualized in Figure 28, Figure 29, Figure 30 and Figure 31 for the calibration period.

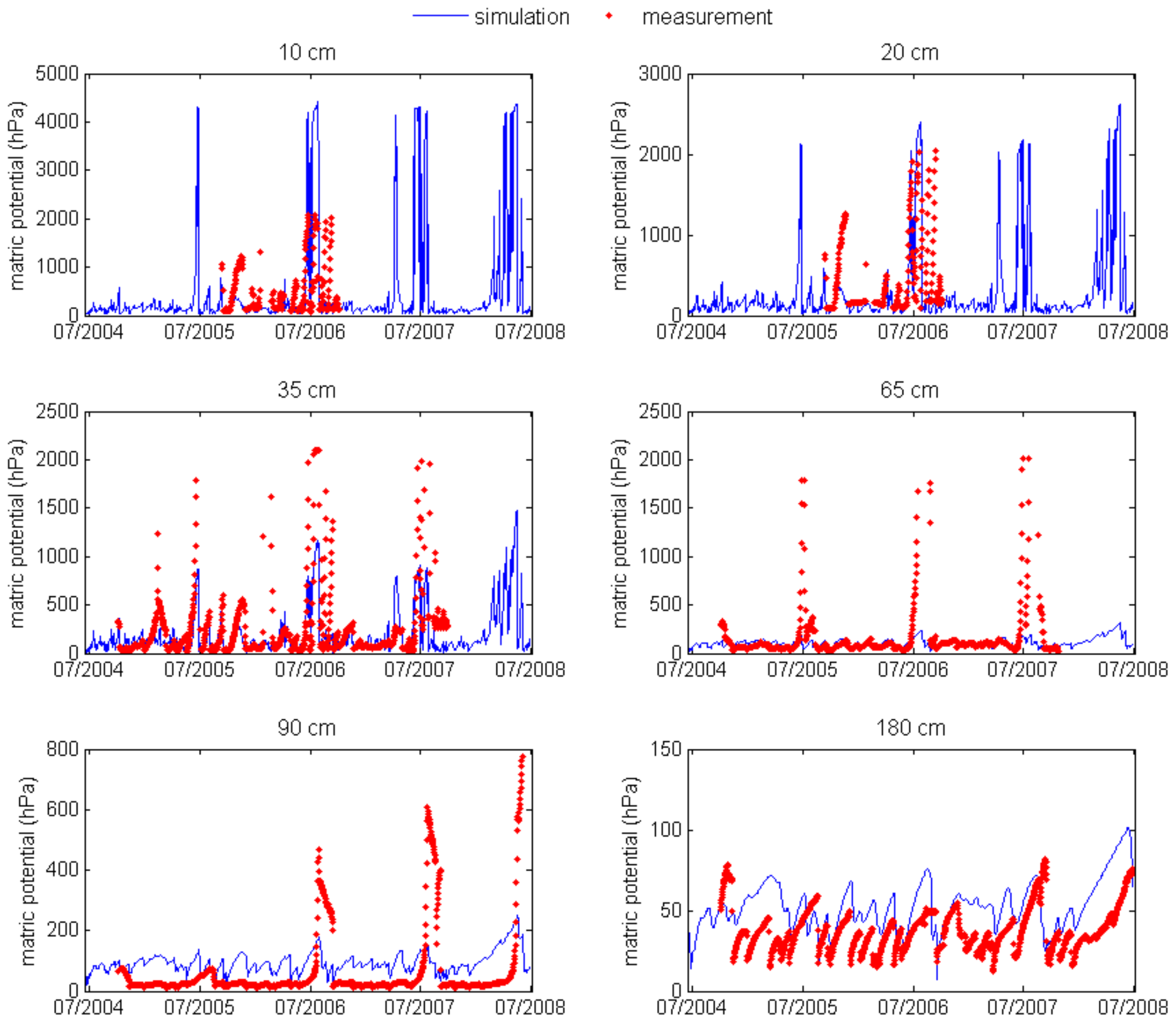


Figure 28: Comparison of measured (red dots) and simulated (blue lines) matric potentials in 10, 20, 35, 60, 90 and 180 cm depth

The comparison of the measured and simulated matric potentials (Figure 28) show that even the standardized parameters are able to represent the dynamic of the system. The NSE, however, are not satisfying (Table 5). When considering that the matric sensors are only able to provide valid measurements up to 2000 hPa and removing all the dates when the simulated values are exceeding 2000 hPa, the NSE of 10 and 20 cm depth can be improved to -0.23 and -0.06, respectively.

The comparison of the measured and simulated matric potentials (Figure 29) show unsatisfying results. Especially the high discrepancies in 90 and 180 cm depths indicate a fundamentally wrong soil parametrization.

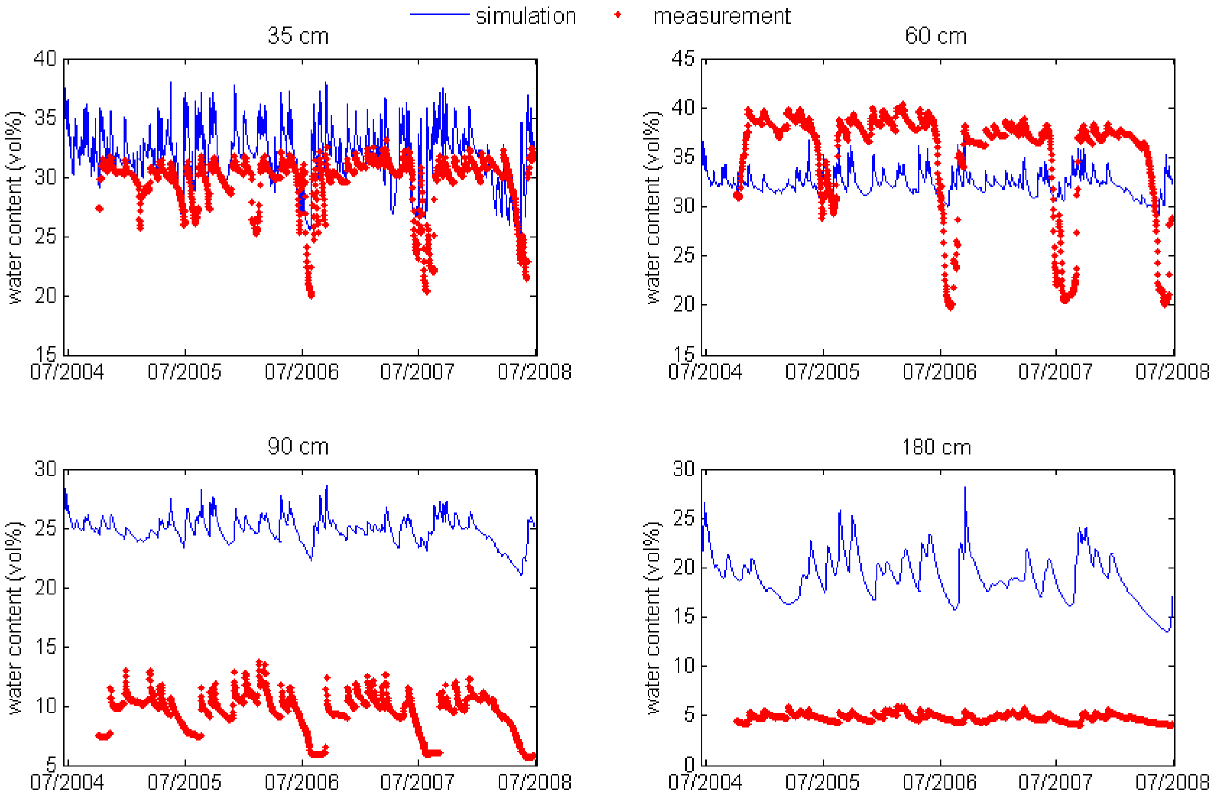


Figure 29: Comparison of measured (red dots) and simulated (blue lines) water contents in 35, 60, 90 and 180 cm depth

Also, the comparisons of the measured and simulated nitrate concentrations (Figure 30) and the comparisons of the measured and simulated yields (Figure 31) show unsatisfying results.

Table 5 summarizes the results of the objective functions (Nash Sutcliffe NSE and Sum of Squared Errors SSE, 3.6) for the comparisons of measured values and simulation results obtained with the standard values visualized above.

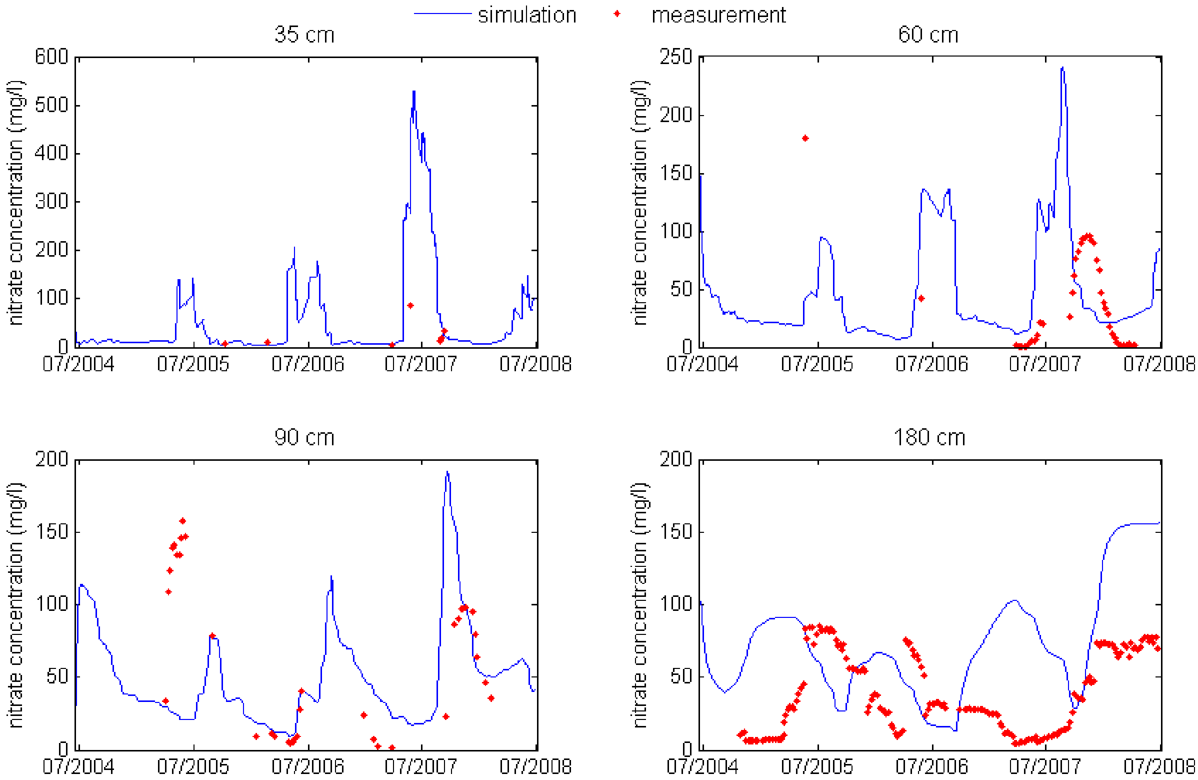


Figure 30: Comparison of measured (red dots) and simulated (blue lines) nitrate concentration in 35, 60, 90 and 180 cm depth

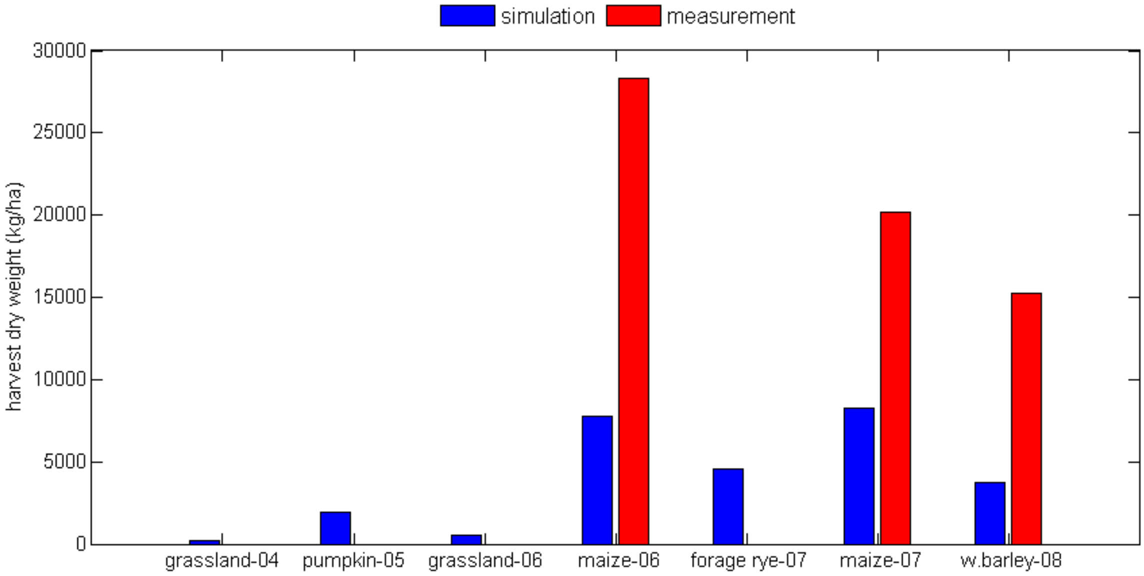


Figure 31: Comparison of measured (red) and simulated (blue) yields (dry weight) of grassland in 2004, pumpkin in 2005, grassland in 2006, maize 300 in 2006, green rye in 2007, maize 300 in 2007 and winter barley in 2008

Table 5: Nash Suftcliffe efficiencies of matric potential (10, 20, 35, 60, 90 and 180 cm), water contents (35, 60, 90, 180 cm), nitrate concentration (35, 60, 90 and 150cm) and the sum of squared errors of the yields (above-ground dry mass)

NSE														**SSE**
Matric potential						*Water content*				*Nitrate concentration*				*Yields in kg/ha*
10	20	35	60	90	180	35	60	90	180	35	60	90	180	
-1.90	-0.20	0.36	0.04	0.03	-1.74	-1.19	-0.26	-85.71	-1324.90	-8.04	-0.50	-0.56	-3.53	703,000000

The results obtained by straight forward simulation with SIMWASER/STOTRASIM are a typical example, where Calisto could be used for project completion. All important inputs are known, including pre-parametrized soil and plant information. The simulation is executable, but the parametrization is not satisfactory yet. At this point of a simulation Calisto should help to improve fine calibration of the parameters.

4.2 System Testing on Generated Data

The calibration system should be tested on its basic functionality without dealing with the peculiarities of a real test site. Therefore, for a first testing, all measurements of the test site were disregarded. Instead, simulation output obtained with the standardized starting parameters described in 4.1 were taken as artificial calibration targets. There appear to be several advantages in choosing this approach:

- The basic functionality of Calisto could be tested without the influence of peculiarities of a real test site.
- The data set to be used was complete.
- The data set to be used was not subject to potential measurement errors.
- It was possible to enable different parameter values for the optimization procedure while leaving others fixed because their real values were known.

The results of the simulation were filled into the file '*Parametrisierung.xls'* (3.3). So the simulated values of yields, water contents, matric potential and nitrate concentration from 16.6.2004 until 30.6.2008 as well as the mean annual evapotranspiration were taken as reference values the model should be calibrated on. All other input (meteorological data, management information,...) was taken unchanged from the Wagna test site. Like mentioned in 4.1, the calibration period was set from 2004 to 2008 with a lead time of two years for stabilizing the system.

4.2.1 Optimization of parameters 'potfak' and 'rs'

As the artificial calibration targets are created with standard values of '*rs*' and '*potfak*' of 1 and 1 (specified for SIMWASER/STOTRASIM in the 'project file' (2.1.3)), it is assumed that they will approximate 1 when enabling them for optimization. Theory suggests that other parameter combinations could lead to similar results ('equifinality', see chapter 1). But as the adjustment to the generated data does not allow any scope for fitting and the mineralization multiplier '*rs*' and the evaporation coefficient '*potfak*' are not able to balance each other out, it is unlikely that other combinations of them can produce equally good simulation outputs. The default ranges for the two parameters were used (Table 1), no starting values were defined and the calibration was done on all objective functions.

Table 6 shows the performance of 11 optimization runs with different population sizes and number of evaluations.

Table 6: Calibration performance for optimizing parameters '*rs*' and '*potfak*'; parameter units are defined in Table 1

	Calisto configurations		**parameters**			
run	Population size	Iteration number	standard values/ ranges/ results	'potfak'	'rs'	run time
-	-	-	standard values	1	1	-
1-11	-	-	ranges	0-20 *	0.2-2.5 *	-
1	100	2000	results	1.02	1.00	102345 s
2	20	500	results	1.07	1.00	32903 s
3	20	500	results	1.02	1.00	26719 s
4	20	500	results	0.86	1.01	25032 s
5	20	500	results	0.97	1.00	29533 s
6	20	500	results	0.97	1.00	29360 s
7	20	200	results	0.86	0.98	10080 s
8	20	200	results	0.89	0.97	10312 s
9	20	200	results	1.29	1.02	10410 s
10	20	200	results	0.81	0.95	9884 s
11	20	200	results	0.92	1.01	12394 s

* ranges comply with the default values according to Table 1

As the population size is recommended with 100 (Zhang et al, 2009), an optimization run with a population size of 100 and 2000 iterations was started. Parameter '*rs'* could be determined on two digits exactly 1.00, parameter '*potfa*k' was determined as 1.02. One evaluation needed on average more than 50 seconds, which lead to a total time of 28.5 hours. To save time, also runs with the lowest possible size of 20 and 500 iterations (average eight hours) and 200 iterations (average three hours) were tested. Figure 32 shows the results of the objective functions of all 11 Calisto runs. The objective functions of the yields and the mean annual evapotranspiration are presented as the sum of squared errors, the objective functions of the time series (matric potential in six depths, water content in four depths and nitrate concentrations in four depths) are presented in the adapted Nash Sutcliffe Coefficient (3.6).

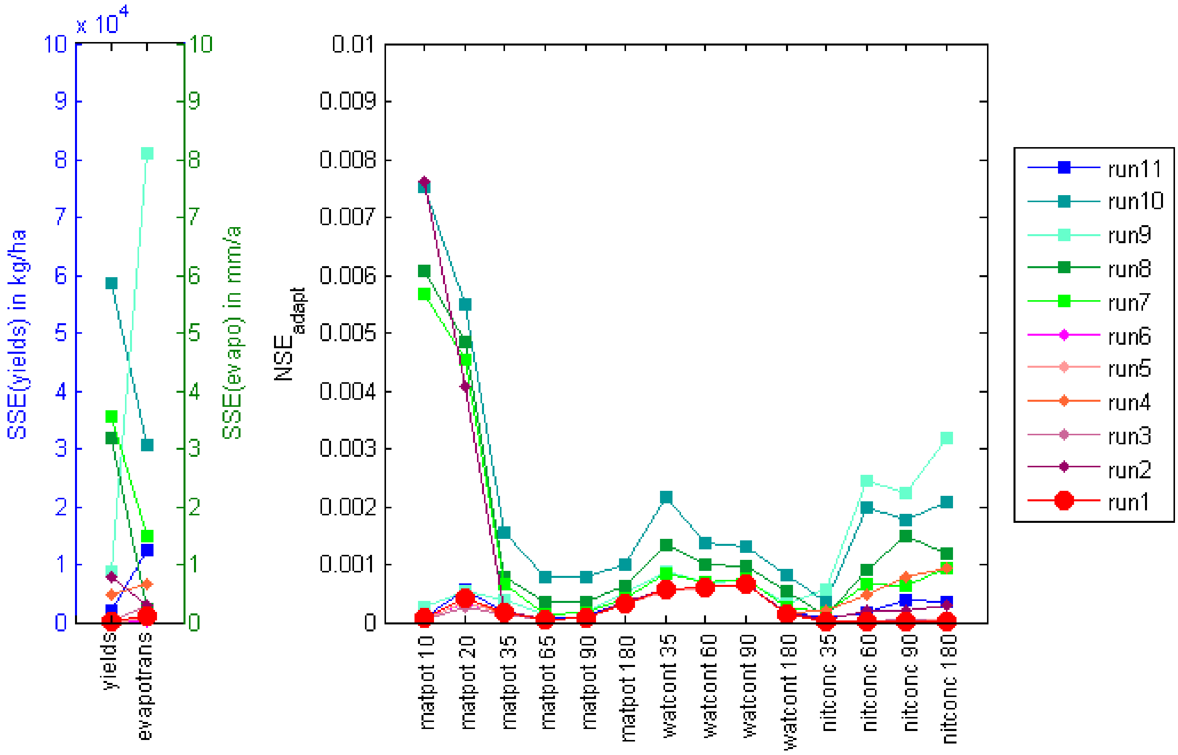


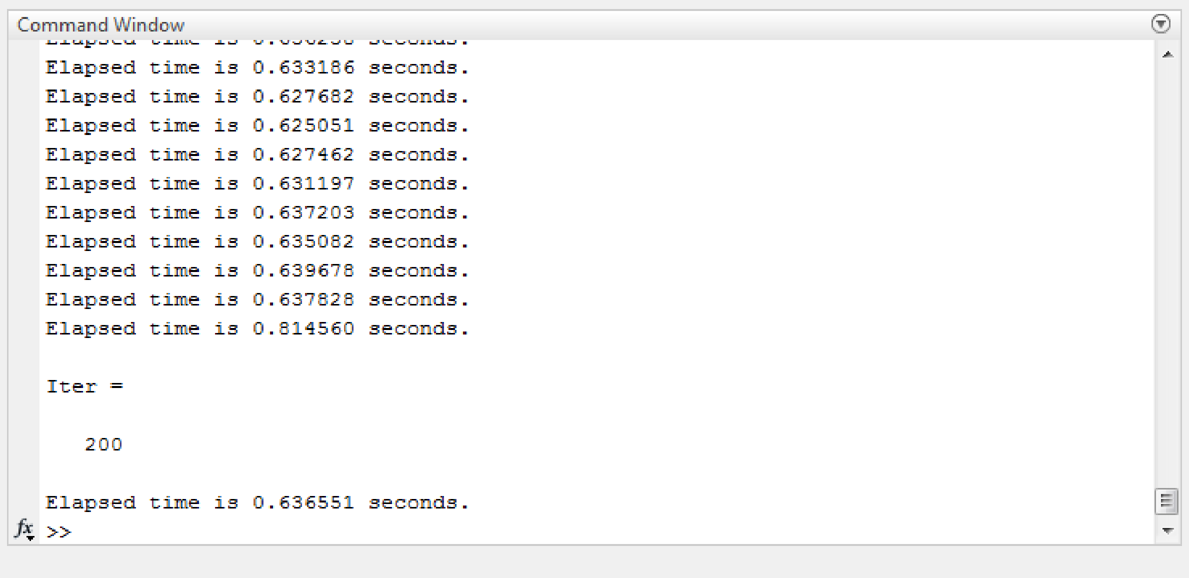
Figure 32: Performance of different evaluation configurations of Calisto for optimizing parameters '*rs*' and *'potfak',* visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentrations in four depths)

The graphs visually prove that run 1, with a population size of 100 and an iteration number of 2000, was able to create results with the best fitting (red), followed by the runs with 500 iterations (shades of orange/purple). The results obtained with the runs with only 200 iterations (shades of green/blue) form the poorest accordance to measurements.

4.2.2 Optimization of plant parameters

For the optimization of plant parameters, two different tests were made. In the first test, several parameters of one crop species only were optimized. In the second test, all five plant species of the calibration period were calibrated at once. To reduce the total number of parameters, only two parameters for each plant were calibrated.

In total, 28 optimization runs were performed. In six cases, Calisto could not finish. After 40 to 60 iterations without incident, all the iteration times were reduced to some seconds. In this cases only empty matrices were handed over from SIMWASER/STOTRASIM to MatLab (3.5.2). MatLab creates matrices with accurate dimensions, but filled with zeros only. So the predicted values are all zero and form a bad match with the measured values they should be calibrated to. According to the principle of AMALGAM, this parameter set was evaluated as poor and the next generation is oriented at parameter sets with better fittings (2.2). In these six cases, however, without exception all the iterations could not provide a result. Figure 33 shows a screenshot of the command window displaying the run times of the iteration runs. Each line represents the execution time of one SIMWASER/STOTRASIM calculation. The Calisto run was finished without presenting an optimized parameter set.



```
Elapsed time is 0.633186 seconds.
Elapsed time is 0.627682 seconds.
Elapsed time is 0.625051 seconds.
Elapsed time is 0.627462 seconds.
Elapsed time is 0.631197 seconds.
Elapsed time is 0.637203 seconds.
Elapsed time is 0.635082 seconds.
Elapsed time is 0.639678 seconds.
Elapsed time is 0.637828 seconds.
Elapsed time is 0.814560 seconds.

Iter =

   200

Elapsed time is 0.636551 seconds.
>>
```

Figure 33: Command Window displaying the run times of a failed Calisto optimization

After a failed optimization, Calisto was started again with the same configurations, which could then lead to results. Despite six failed optimizations in total, 22 optimizations could be finished successfully: 11 for optimizing several parameters of one crop species and 11 for optimizing ‘rs’ and ‘potfak’ for all five plant species. Table 7 shows the calibration results for the optimization of the parameters extinction coefficient *ext*, leaf area *bfgw*, assimilation *as*, potential root length *rlg* and the sum of accumulated photothermal units at riping *ripe* of maize.

Table 7: Calibration performance for optimizing parameters ‘ext’, ‘bfgw’, ‘as’, ‘rlg’ and ‘ripe’ of maize, excluding failed optimization runs; parameter units are defined in Table 1

	Calisto config.		**parameters**						
run(s)	Population size	Iteration number	standard values/ ranges/ results	‘ext’	‘bfgw’	‘as’	‘rlg’	‘ripe’	run time
-	-	-	standard	0.65	0.0015	75	20	1100	-
1-11	-	-	ranges	0.30-0.75*	0.0015-0.0050*	7-90*	4-40*	500-4500*	-
1	100	2000	results	0.59	0.0020	60	27	1116	101159s
2	20	500	results	0.63	0.0016	69	15	1116	24881s
3	20	500	results	0.63	0.0017	67	11	1128	24956s
4	20	500	results	0.59	0.0020	65	28	1145	26159s
5	20	500	results	0.66	0.0016	67	4	1107	24321s
6	20	500	results	0.61	0.0028	44	17	1130	24259s
7	20	200	results	0.47	0.0050	33	40	1210	10187s
8	20	200	results	0.49	0.0032	47	31	1175	9633s
9	20	200	results	0.54	0.0048	26	13	1152	10019s
10	20	200	results	0.57	0.0017	74	8	1089	9579s
11	20	200	results	0.72	0.0017	63	25	1089	9775s

* ranges comply with the default values according to Table 1

Like for the optimization of parameters ‘*rs*’ and ‘*potfak*’, one run with a population size of 100 and 2000 iterations (run time 28 hours), five runs with a population size of 20 and 500 iterations (average run time 6.9 hours) and five runs with a population size of 20 and 200 iterations (average run time 2.75 hours) were made. Figure 34 shows the corresponding function evaluations of the eleven optimization runs.

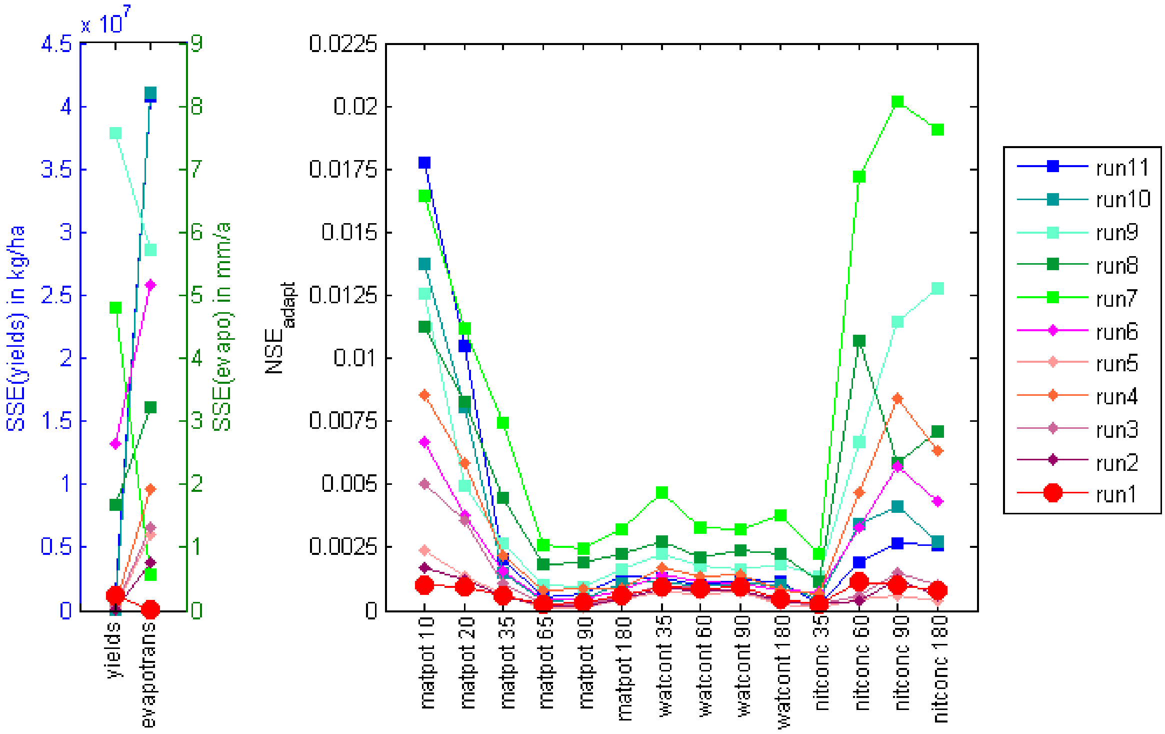


Figure 34: Performance of different evaluation configurations of Calisto for optimizing parameters *'ext'*, '*bfgw', '*as*', '*rlg*' and '*ripe*' of maize, visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentration in four depths)

Figure 34 proves that the number of iterations positively correlates with the goodness of fit of the obtained optimized parameter values. Run 1 (red), with a population size of 100 and 2000 iterations could gain the best results: the lowest values of NSE$_{adapt}$ and of SSE. It is followed by the runs with a population size of 20 and 500 iterations (shades of orange/purple). The runs with only 200 iterations (shades of green/blue) form the poorest accordance to measurements.

Table 8 shows the calibration results for optimizing the parameters assimilation *'as'* and the sum of accumulated photothermal units at ripening '*ripe'* of the plants winter barley, winter rye, maize, grassland and pumpkin. Figure 35 shows the corresponding functions evaluations. Here again, the number of iterations positively correlates with the goodness of fit of the optimization results.

Table 8: Calibration performance for optimizing parameters ‘as’ and ‘ripe’ of winter barley, winter rye, maize, grassland and pumpkin; excluding failed optimization runs; parameter units are defined in Table 1

	Calisto configurations		**parameters**											
run(s)	Population size	Iteration number	standard values/ ranges/ results	winter barley		winter rye		maize		grassland		pumpkin		run time
				‘as’	‘ripe’	‘as’	‘ripe’	‘as’	‘ripe’	‘as’	‘ripe’	‘as’	‘ripe’	
-	-	-	standard values	17	1500	20	1600	75	1100	7	2100	11	1750	-
1-11	-	-	ranges	7-90*	500-4500*	7-90*	500-4500*	7-90*	500-4500*	7-90*	500-4500*	7-90*	500-4500*	-
1	100	2000	results	50	1608	49	1294	69	1162	7	1663	12	1541	108252s
2	20	500	results	68	1894	7	3169	67	1099	7	2426	11	1520	27440s
3	20	500	results	41	1789	58	3534	77	1103	7	1413	17	1688	25436s
4	20	500	results	53	1828	76	1828	49	1064	7	893	8	1708	26192s
5	20	500	results	52	2117	27	3072	58	1148	40	500	25	1815	26005s
6	20	500	results	26	1581	79	4340	85	1100	21	4488	17	1683	25083s
7	20	200	results	32	2337	78	2078	74	1237	52	569	52	3707	11707s
8	20	200	results	7	1862	8	1301	66	1116	14	4500	9	2073	10410s
9	20	200	results	51	2177	38	3610	86	1187	7	4500	58	4500	10361s
10	20	200	results	11	1536	90	2594	90	1473	20	1314	22	1542	11615s
11	20	200	results	25	1840	61	1818	69	1190	16	3608	8	3688	11652s

* ranges comply with the default values according to Table 1

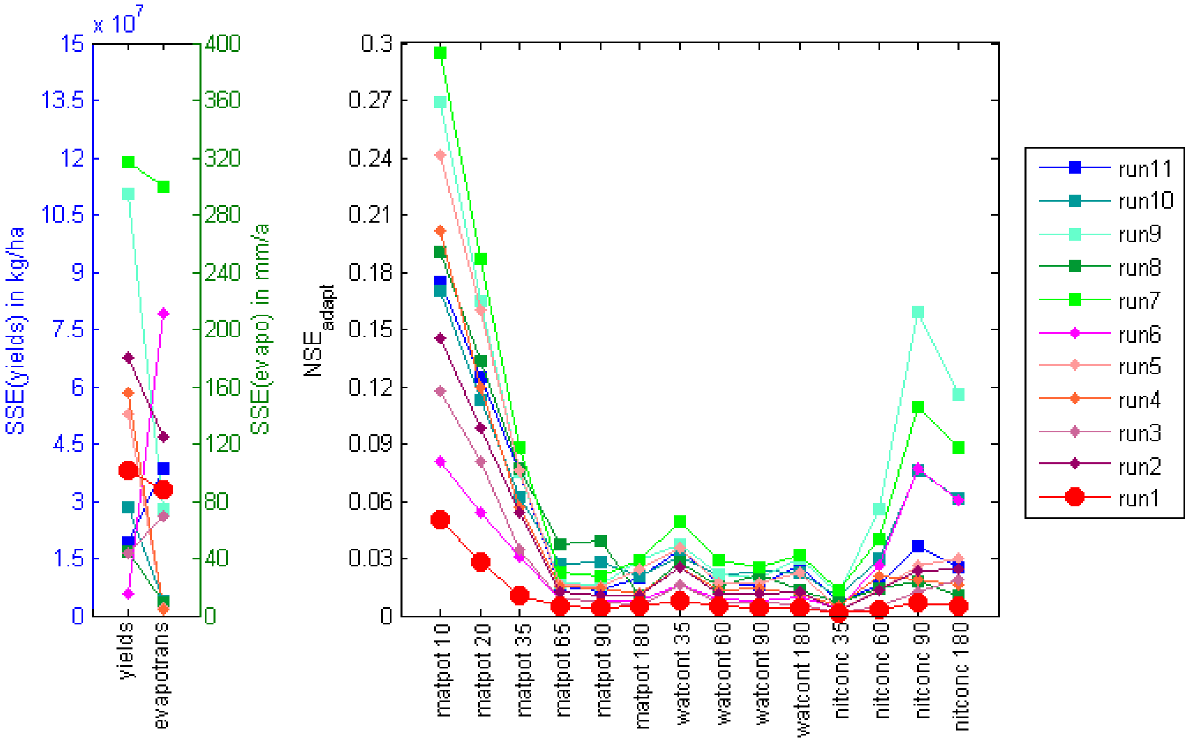
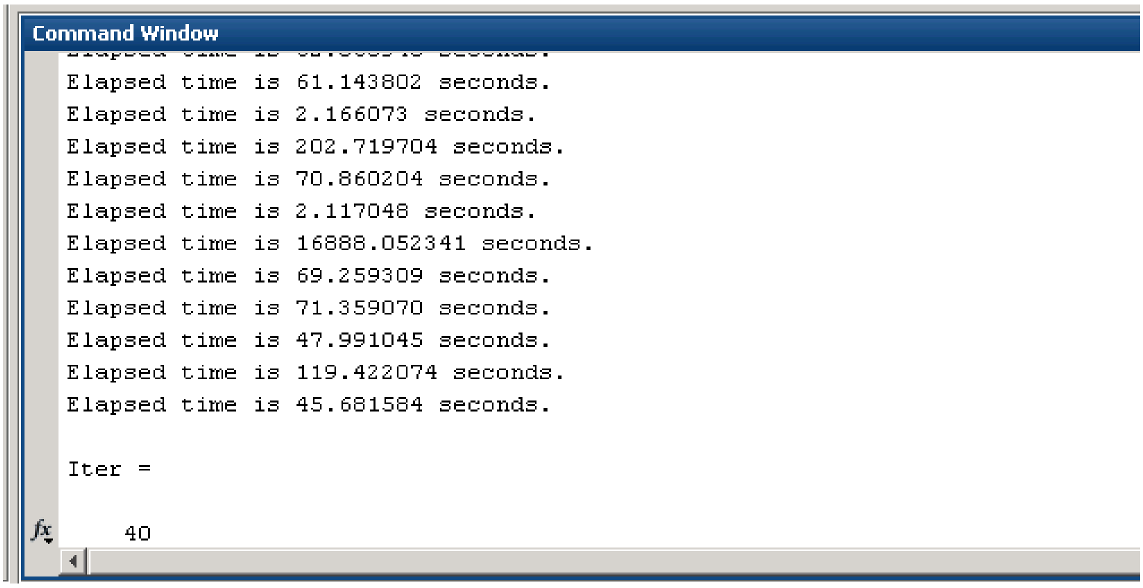


Figure 35: Performance of different evaluation configurations of Calisto for optimizing parameters ‘as’ and ‘ripe’ of the five plants winter barley, winter rye, maize, grassland and pumpkin, visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentration in four depths)

4.2.3 Optimization of soil parameters

When optimizing soil parameters (different combinations of parameters and layers), very often problems occurred. Figure 36 shows a screenshot of the command window displaying characteristic run times of the iteration runs for optimizing the soil parameters.



```
Elapsed time is 61.143802 seconds.
Elapsed time is 2.166073 seconds.
Elapsed time is 202.719704 seconds.
Elapsed time is 70.860204 seconds.
Elapsed time is 2.117048 seconds.
Elapsed time is 16888.052341 seconds.
Elapsed time is 69.259309 seconds.
Elapsed time is 71.359070 seconds.
Elapsed time is 47.991045 seconds.
Elapsed time is 119.422074 seconds.
Elapsed time is 45.681584 seconds.

Iter =

    40
```

Figure 36: Command Window displaying the run times

When observing these iteration run times, two unusual and unwanted cases can be spotted. Usually, the calculation takes between 40 and 70 seconds, as observed when optimizing ‘rs’ and ‘potfak’ and the plant parameters (4.2.1 and 4.2.2). For the case where only 2 seconds are displayed, no SIMWASER/STOTRASIM result could be received. As long as there are only single occurrences of aborted iterations, they do not influence the performance of Calisto. In the case where the iteration in Figure 36 displays 16888 seconds (4.7 hours), the SIMWASER/STOTRASIM calculation became very slow.

These phenomena can lead to two errors that hinder Calisto from finishing its optimization. Two cases can occur:

- *error 1:* The run time of all the iterations are reduced to less than a second or some seconds and Calisto cannot find an optimized parameter set, as described in in 4.2.2.
- *error 2*: The run time of one or more iteration(s) exceeds several hours and has to be cancelled manually

When a Calisto optimization is configured with 200, 500 or even more iterations, only a few runs with several hours lead to a tremendous proliferation of the total Calisto run time. It is not possible for MatLab to terminate SIMWASER/STOTRASIM after a certain amount of time. Although timer can be included in the MatLab code, MatLab waits until the .NET application is finished and only then it proceeds to the next code line where a possible time break could be implemented. Therefore, a time limitation needs to be set within SIMWASER/STOTRASIM itself. The Visual Studio Debugger indicated that the time often was lost within the method ‘*berechneWasserbewegung’* in the script ‘*Grundwasser.cs’*. Therefore, a workaround was developed to minimize this problem by integrating a stopwatch into the Timestep Loop. This forces the loop to stop after a million milliseconds, which is equivalent to 0.28 hours.

```
Stopwatch sw = new Stopwatch();
          sw.Start();

        while (summeZeit < dauer)
        {
          if (sw.ElapsedMilliseconds > 1000000) throw new TimeoutException();
….
```

This measure reduced the number of iterations with escalating run time in a few cases. Nevertheless, there were still cases where the run time exceeded several hours. A physical reason of this problem could not be identified and thus no solution provided. Therefore, all optimization trials that exceeded approximately 8 hours were terminated by forcing Windows via the task manager to cancel MatLab.

Most of the soil parameter optimization trials eventually ended in one of the two errors described. Most success could be reached by setting narrow parameter ranges. As an example, Table 9 shows the Calisto calibration performance for optimizing the deepest soil layer SSG5H1 of the test data.

Table 9: Calibration performance for optimizing soil parameters ‘α1’, ‘α2’, ‘n1’, ‘n2’, ‘w’, ‘Ks’ and ‘τ’ of the deepest soil layer SSG5H1; parameter units are defined in Table 1

	Calisto configurations		parameters								
run	Population size	Iteration number	standard values/ ranges/ starting values/ results	α1	α2	n1	n2	w	Ks	τ	run time
-	-	-	standard values	0.0217	0.0944	2.58	1.25	0.6	24	0.00	-
1-10	-	-	ranges	0.0001 - 20*	0.0001 - 20*	1.1 - 9.0*	1.1 - 9.0*	0-1*	0.1-10000*	-3- 3*	-
1	20	200	results	error 2							> 20h
2	20	200	results	error 2							> 7h
3	20	200	results	error 2							> 8h
4	20	200	results	error 2							> 10h
5	20	200	results	error 1							
6 - 10			starting values	0.03	0.1	2.3	1.4	0.6	25	0	
6	20	200	results	error 1							
7	20	200	results	error 1							
8	20	200	results	error 2							> 8h
9	20	200	results	error 1							
10	20	200	Results	error 2							> 10h

11 - 15			ranges	0.0001 - 20*	0.0001 - 20*	1.1 - 9.0*	1.1 - 9.0*	0- 1*	10-100	0 - 1	
11	20	200	results	error 2							> 20h
12	20	200	results	error 2							> 14h
13	20	200	results	error 2							> 8h
14	20	200	results	error 2							> 10h
15	20	200	results	error 2							> 8h
16-20			ranges	0.0001 - 1	0.0001 - 1	2.0 - 3.0	1.1 - 2.0	0.4 - 0.8	10 - 100	0 - 1	
16	20	200	results	error 1							
17	20	200	results	error 1							
18	20	200	results	0.0001	0.098	3	2	0.8	100	0	10922s
19	20	200	results	error 2							> 8h
20	20	200	results	error 1							
21-25			ranges	0.0001 - 1	0.0001 - 1	1.1 - 5.0	1.1 - 5.0	0.4 - 0.8	10 - 100	0 - 1	
21	20	200	results	0.029	0.053	2.72	1.59	0.63	10	0.085	11031s
22	20	200	results	0.043	0.077	2.59	1.12	0.56	43	0.035	11626s
23	20	200	results	0.0004	0.037	1.10	4.90	0.46	10	0.39	11423s
24	20	200	results	0.0001	0.106	1.63	1.58	0.78	100	0.028	10672s

25	20	200	results	error 1							
26-30			ranges	0.0001 - 20*	0.0001 - 20*	1.1 - 5.0	1.1 - 5.0	0.4 - 0.8	0.1- 10000*	-3 -3*	
26	20	200	results	error 1							
27	20	200	results	error 2							> 7h
28	20	200	results	error 2							> 12h
29	20	200	results	error 1							
30	20	200	results	error 2							> 14h
31-35			ranges	0.0001 - 1	0.0001 - 1	1.1 - 9.0*	1.1 - 9.0*	0- 1*	0.1- 10000*	-3 -3*	
31	20	200	results	error 1							
32	20	200	results	error 1							
33	20	200	results	error 1							
34	20	200	results	0.3758	0.0331	1.28	3.70	0.37	4051	2.67	59631s
35	20	200	results	error 2							> 8h
36-40			ranges	0.0001 - 1	0.0001 - 1	1.1 - 5.0	1.1 - 5.0	0- 1*	0.1- 10000*	-3 -3*	
36	20	200	results	error 1							
37	20	200	results	error 1							
38	20	200	results	error 2							> 10h

39	20	200	results	error 1							
40	20	200	results	error 2							> 9h
41-45			ranges	0.0001 - 1	0.0001 - 1	1.1 - 5.0	1.1 - 5.0	0.4 - 0.8	0.1 - 10000*	-3 -3*	
41	20	200	results	error 1							
42	20	200	results	error 1							
43	20	200	results	error 2							> 7h
44	20	200	results	error 1							
45	20	200	results	error 2							> 11h
46-50			ranges	0.0001 -20*	0.0001 -20*	1.1 - 5.0	1.1 - 5.0	0.4 - 0.8	0.1- 10000*	0 - 1	
46	20	200	results	error 1							
47	20	200	results	error 1							
48	20	200	results	error 1							
49	20	200	results	error 1							
50	20	200	results	error 1							

* ranges comply with the default values according to Table 1

Runs 1-5 were started with the default ranges and no starting values. They could not obtain any results. Even the setting of fictional starting values, that were very close to the ‘standard values’ (Table 3), could not help Calisto to finish its optimization (runs 6-10). Runs 11-15 tested the assumption that the setting of very narrow ranges for the hydraulic conductivity parameters ‘w’ and ‘Ks’ would improve calibration as it could possibly prevent the water flow from getting unrealistically slow or fast. This assumption could not be confirmed, as all five runs had to be terminated by hand after 8 or more hours. In runs 16-20 narrower ranges for all 7 parameters were set. One of the five runs could be finished, but showed results directly at the upper and lower limits. There is no plausible explanation why the runs 21-25, with the same ranges except for ‘n1’ and ‘n2’ – which had even wider ranges – were more successful, although the results were within the ranges of runs 16-20. These runs, however, with ranges of 0.0001 to 1 for ‘α1’ and ‘α2’, 1.1 to 5.0 for ‘n1’ and ‘n2’, 0.4 to 0.8 for ‘w’, 10 to 100 for ‘Ks’ and 0 to 1 for ‘τ’, could obtain results within the expected time frame of approximately 3 hours in four of five cases. The calibration performance of the four successful optimization runs is visualized in Figure 37. Runs 26-50 indicate that it is really necessary to narrow the ranges of all seven parameters, as the narrowing of only few of them could not lead to results. The only run where results could be obtained was run 34, but only after 16.5 hours.

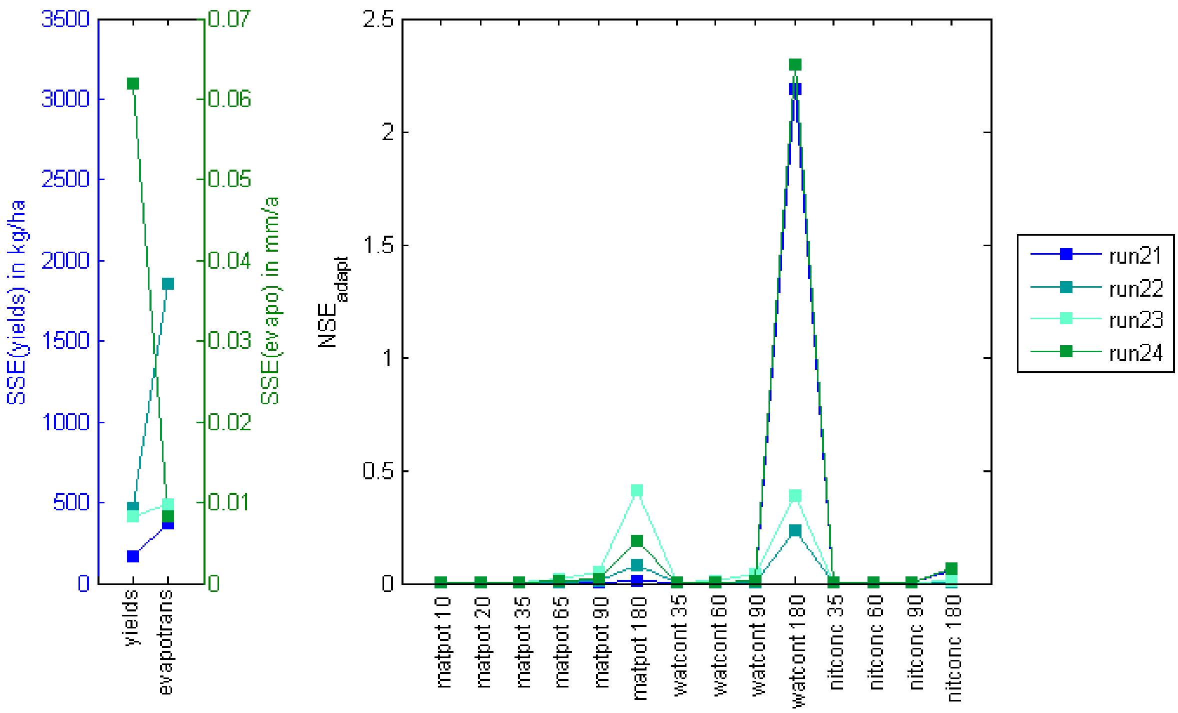


Figure 37: Performance of different evaluation configurations of Calisto for optimizing parameters ‘α1’, ‘α2’, ‘n1’, ‘n2’, ‘w’, ‘Ks’ and ‘τ’ of the soil layer SSG5H1, visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentration in four depths)

It appears that setting of narrower parameter ranges was a successful way to avoid both errors 1 and 2. Unfortunately, the same range definitions could not be used for the other soil layers as well. A comparison with the ‘standard values’ of all layers in Table 3 shows that not all of them are located in the ranges that could be obtained as the most successful for layer SSG5H1. For the other layers, other ranges had to be found that allow the proper functioning of Calisto without errors.

5. Discussion

The main task of this master thesis was the development of the software 'Calisto'. Calisto is an application, executed in MatLab, that combines the soil water and nitrogen balance model SIMWASER/STOTRASIM with the optimization tool AMALGAM to allow automatic calibration. The task was performed by developing systems of data and information transfer between MatLab (AMALGAM) and C# (SIMWASER/STOTRASIM) (3.1 and 3.5), defining objective functions (3.6), starting values and parameter ranges (3.4) and developing structures for data input and result output (3.3 and 3.7).

Moreover, an alternative for the representation of the hydraulic conductivity and water retention characteristics was worked out (3.2). SIMWASER/STOTRASIM is designed to use 'standard curves' of soil hydraulic functions (2.1.3). Theoretically, it would have been possible to use 'standard curves' for soil hydraulic functions for calibration, too. In this case an optimization of twenty monotonic decreasing values for each soil layer would have been necessary. To reduce the number of parameters and to guarantee a link between the hydraulic conductivity and water retention functions, a model was used instead. The bimodal van Genuchten-Mualem (bvGM) model is able to describe both curves with only 9 parameters in total. This formulation has the big advantage that it couples the conductivity function with the retention function and therefore minimizes the parameter number (Durner, 1994). In tests, the bimodal formulation seemed to be the most successful for approximating the 'standard curves' developed by Stenitzer (1988; Figure 27). SIMWASER/STOTRASIM, however, is originally not designed for working with curves obtained with the bvGM model. Also, the bvGM parameter values obtained when fitted to the standard curves, do not in all cases provide realistic values. Looking at the parameter values for soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1 (Table 3), especially the values for 'thetaR' (very small values) and 'Ks' (very large value for SL3G4H3) are not all physically meaningful. Moreover, there are a few general points of criticism on the model mentioned in literature. Ippisch et al. (2005) theoretically showed conditions for which the vGM model leads to wrong prediction of relative hydraulic conductivity and suggested the introduction of an air-entry value for these cases. It would have been conceivable to describe the water retention with the modified van Genuchten model by Ippisch et al., or with another model like Brooks and Corey model (Brooks and Corey, 1964). They could also be coupled with another conductivity model like the model of Burdine (1953).

Another big challenge of setting up an automatic calibration was the formulation of the objective functions. At the moment, Calisto is able to calibrate on the measured yields, the mean annual evapotranspiration and the time series of water content, matric potential and nitrate concentrations in different depths. For a further development of the program also calibrations on the leachate (e.g. outflow of a lysimeter), the nitrate concentration in the leachate or the N_{min} or N_{ges} contents in several depths of the soil profile could be considered. AMALGAM is a multiparameter optimization tool that enables the user to calibrate on different objective functions at once. Most authors, however, have been using the same measure for the quality of fit for all objective functions (Zhang et al, 2009; Wöhling and Vrugt, 2011). In Calisto Nash Sutcliffe Efficiency (NSE) and Sum of Squared Errors (SSE) were mixed for the optimization of time series of water content, matric potential and nitrate concentration on the one hand and the yields and evapotranspiration on the other hand. To make the mixing of two different measures possible, the objective functions had to be normalized in the evaluation process of the best parameter set (3.7). No further tests of the effects of mixing and normalizing have been done. To resolve the uncertainty, the Calisto user can choose to calibrate on one of the different measures only. The parameters can only be optimized on measurements that are put into the Excel file '*Parametrisierung.xls'*, so the user has control by adding or deleting values. If the user wants to calibrate the plant parameters only, it could be reasonable to calibrate on yields only. For the calibration of parameter '*potfak'*, the time series of nitrate concentrations are probably necessary for calibration. There is a need of further studies and experience with the application of Calisto to evaluate the effects of the normalization and the best choice of measurements to optimize on. Another approach could be the introduction of weights for the different objectives that can be defined by the user. The weight definition as well as the selection of single objective functions by the user, however, is very subjective. This would contradict the claim of the automatic calibration approach of being objective and repeatable.

For the evaluation of yields and evapotranspiration the SSE was chosen from a wide range of measures. Possible alternatives would have been Root Mean Squared Error or an Error without squaring, amongst many others. Available literature indicates problems with both kinds of error measure, whether r or r², as their magnitudes are not consistently related to the accuracy of the prediction (Willmott, 1982). The time series of the water contents, matric potential and nitrate concentration are evaluated with the NSE. The NSE is often used to evaluate water content or matric potential in the vadose zone (Zhang et al, 2009; Groh eh al, 2013; Groenendijk et al., 2014), amongst other measures. Groenendijk et al. (2014) for example compare different nitrate leaching models with four different measures: The Mean Absolute Error, the Root Mean Squared Error, the Index of Agreement (Willmott, 1982) and the NSE. All those measures have deficiencies when it comes to describing the overall dynamic of the system. In the present case, the use of the mean observed value as a reference in the NSE can be a very poor predictor because it is evaluating time series with a strongly seasonal fluctuation (Schaefli and Gupta, 2007). Moreover, the NSE heavily weights large deviations. As it can be observed in the accordance of the measured and predicted matric potential values in 35 and 65 cm depth on the test site (Figure 28) the overall dynamics is predicted quite well. The NSE value, however, is poor according to Table 5 because the model is not able to represent a few measurement peaks. When doing a manual calibration, the researcher could take the fact into consideration, that the model performance is excellent for most of the time, whereas the NSE cannot differentiate between different time periods.

The calibration of the soil is limited, as the user of Calisto needs to define in advance the layering of the soil profile. The layering is not changed during the calibration process, only the parameters of the single layers are adapted. Also the diffusion parameters are defined in advance and are not changed during the calibration process. The simultaneous adaption of the diffusion parameters could be done quite easily and should be included in future upgrades of Calisto.

Calisto optimization relies on the principle that SIMWASER/STOTRASIM outputs are compared to measurements and then, based on the goodness of fit, new input parameters are created. This obviously relies on the assumption that SIMWASER/STOTRASIM is able to produce a valid output. To ensure the workability, it is strongly recommended that there exists a valid pre-parametrization of the model even before the optimization process is started. The Wagna test site was chosen as example for such a case where simulation is executable, but the parametrization is not satisfactory yet (4.1). First application tests showed that the Calisto system is executable and able to calibrate a few parameters enabled for optimization (4.2.1). However, three big shortcomings of Calisto became apparent:

1) long execution time
2) *error 1:* Calisto can not optimize because only empty matrices are handed over from SIMWASER/STOTRASIM to MatLab
3) *error 2*: iteration runs take several hours and have to be stopped manually

The problem of the long execution times does not arise from the construction of Calisto itself, but from the run time of SIMWASER/STOTRASIM. When assuming the run time of one SIMWASER/STOTRASIM calculation of about one minute, the Calisto application with a population number of 100 and 2000 iterations takes approximately 30 hours. Proposals to implement AMALGAM in multiple trials by Zhang et al. (2009), the total application time would exceed several days. The optimization of parameters '*rs'* and '*potfak'* only (4.2.1) showed, that a population number of 20 and an iteration number of 200 or 500 could be adequate as well (Table 6) which reduced the run time significantly (approximately 3 hours for 200 iterations or 8 hours for 500 iterations). The objective function evaluations in Figure 32 show that runs two to six with 500 iterations could lead to comparably satisfying results than the run with 2000 iterations. All the results, whether 500 or 200 iterations, showed a good fit with very small SSE and NSE_{adapt}. Optimizing a larger amount of parameters at once required a greater number of population and iterations, which was observed when optimizing the plant parameters (4.2.2). But also when optimizing five plant parameters at once, the combinations of a population size of 20 and 500 iterations could reach acceptable results. The sufficiency of such small numbers of population size and iterations is quite surprising, as usually in literature a minimum population number of 100 and several thousand iterations is proposed (Mertens et al., 2006; Zhang et al., 2009; Woehling and Vrugt, 2011).

In the present case the parametrization was split into the optimization of parameters '*potfak'* and '*rs',* of plant parameters and of soil parameters. The adjustment of a number of parameters as small as possible at once can serve as strategy for reducing the execution time. The splitting into calibration of 'potfak' and 'rs', plant parameters and soil parameters can be one attempt to reduce the parameter number. As another strategy, it is also conceivable to do the soil parametrization on a short period of fallow land only. In this period there is no influence of the plants, therefore the plant parameters can be neglected and a calibration restricted to such periods may lead to shorter execution time. However, it can be challenging to find a short period where a wide range of matric potentials and water contents is covered. An argument against this method of parameter reduction is that an overall parametrization of the whole model dynamics should be sought, not only of the submodels. Another possibility to reduce the number of parameters is to concentrate on the most sensitive parameters (Woehling and Vrugt, 2011). In 4.2.2, the two plant parameters '*as*' and '*ripe*' were chosen because they are the most sensitive according to expert knowledge (Feichtinger, 2015). For a scientifically correct manner, a sensitivity analysis should be done before the calibration to reduce parameters (Mertens et al., 2005; van Griensven et al, 2006; Kamali et al, 2012; Groh et al, 2013).

Long execution times are a problem when there is time pressure for the completion of the model calibration, but they can be accepted as long as they stay in expected time limits. The even bigger problem, however, is, when Calisto is not able to do a calibration at all. The problem of the optimizations did not occur when optimizing parameters '*potfak'* and '*rs*', but when optimizing plant parameters and soil parameters. Experience showed that if there is a mismatch of the input parameters, SIMWASER/STOTRASIM is not able to perform a calculation. It stops calculation and so the premise of Calisto - that SIMWASER/STOTRASIM is able to produce outputs – cannot be met. At present it cannot be explained why in some cases at a sudden point of usually 50 to 60 iterations, SIMWASER/STOTRASIM cannot provide results any more (error 1). As long as the number of failed optimizations does not get out of hand, they do not hinder Calisto from its successful application for model calibration. Usually a new start of Calisto, even with the same configurations, leads to results (4.2.2).

Another challenging problem is the phenomenon of iterations exceeding all expected time bounds, which happened when the soil parameters were optimized (error 2). The very long execution times of several hours for one iteration can eventually be explained by the fact that SIMWASER/STOTRASIM chooses its time steps itself, depending on the calculated water flux. To solve the problem the attempt of including a timer into the time step loop in C# was made (4.2.2), however without success. Another approach could be the external definition of a minimum time step in the calculation procedure. This would require profound changes within the coding of the program SIMWASER/STOTRASIM and could not be tackled within the framework of this master thesis.

A workaround of the problem may be the setting of narrower ranges for the calibration parameters. For the optimization of the parameters of the soil layer SSG5H1 on generated data the setting of a narrow range was successful (4.2.3). However, no overall valid parameter ranges could be found. Matters are complicated by the fact that some of the obtained model parameter values are not physically realistic (see above). The workability of the ranges seem to depend on the specific soil layer in the specific project and have to be found out individually by a trial-and-error procedure, which is tedious and time consuming. Moreover the workaround of setting narrower ranges works against the claim of objectivity and against the idea of AMALGAM to exploit a wide parameter space and find solution sets all over the parameter space.

To meet scientific standards, a model validation would be necessary after calibration (e.g. Groenendijk et al., 2014). As suggested by Groenendijk et al. (2014) the validation period would be set after the calibration period, from 2008 to the end of 2011.

6. Conclusion

It is important to note that the calibration of SIMWASER/STOTRASIM with an automatic tool instead of manual calibration still requires a considerable amount of expert knowledge. The user needs to have an understanding of the model structure and the underlying processes as well as of the character of the input parameters and their expected search ranges. Calisto is only suitable for the fine calibration of SIMWASER/STOTRASIM, therefore it is necessary to have sufficient prior knowledge about the soil profile as well as rough first estimates of the soil and plant parameters.

Application tests showed that the Calisto system is executable and able to calibrate a few parameters enabled for optimization. However, three big shortcomings of Calisto became apparent: At first, the general execution time of the Calisto optimizations is very long which makes the application inconvenient. The even bigger problems are the two types of error that arose when optimizing the plant and soil parameters. In a few Calisto applications all the iteration times were reduced to some seconds and only empty matrices were handed over from SIMWASER/STOTRASIM to AMALGAM. Calisto could not finish and not find an optimized parameter set (error 1). When parametrizing soil parameters, the phenomenon occurred that single iteration runs took several hours and the Calisto process had to be stopped manually after exceeding certain time limits (error 2).

As the system testing on generated data did already reveal several deficiencies, no calibration on the measured values of yields, annual evapotranspiration, water contents, matric potential and nitrate concentrations was started. The solving of the errors is crucial in any case, before the calibration of a real project can be started. The solving, however, would require profound changes within the optimization approach or even within the coding of the program SIMWASER/STOTRASIM. Therefore it could not be tackled within the framework of this master thesis.

To conclude, there are several unresolved questions and deficiencies in the application of Calisto. There is a need to carry out further studies or even question the general suitability of the optimization tool for calibrating SIMWASER/STOTRASIM.

7. References

Addiscott T.M., Bailey N.J., Bland G.J. and Whitmore A.P. (1991) Simulation of nitrogen in soil and winter wheat crops: a management model that makes the best use of limited information. *Fert Res* 27: 305-312.

Ad-Hoc-Arbeitsgruppe Boden (2005) Bodenkundliche Kartieranleitung. Bundesanstalt für Geowissenschaften und Rohstoffe in Zusammenarbeit mit den Staatlichen Geologischen Diensten, 5th edition, 438p. ISBN: 3-510-95920-5

Beven K. and Binley A. (1992) The future of distributed models: Model calibration and uncertainty prediction. *Hydrol Process* 6: 265-277.

Brooks R. H. and Corey A. T. (1964) Hydraulic properties of porous media. Hydrol Pap 3, Colo. State Univ., Fort Collins.

Burdine N.T. (1953) Relative permeability calculation from pore size distribution data. *Trans Am Inst Min Eng* 198: 71-78.

Deb K., Pratap A., Agarwal, S. and Meyarivan T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans Evol Comp* 6: 182-197.

Durner W. (1994) Hydraulic conductivity estimation for soils with heterogeneous pore structure. *Water Resour Res* 30(2): 211-223.

Duynisveld, W.H.M. (1983) Entwicklung von Simulationsmodellen für den Transport von gelösten Stoffen in wasserungesättigten Böden und Lockersedimenten. Umweltbundesamt, Texte 17/83, Berlin.

Eder A., Schmid G., Huber L., Bloeschl G. and Strauss P. (2016) Nitrogen leaching of Austrian’s arable areas under the perspective of greenhouse gas emissions. Manuscript submitted for publication.

Feichtinger F. (1988) Stotrasim – Ein Modell zur Simulation der Stickstoffdynamik in der ungesättigten Zone eines Ackerstandortes. In: Bundesamt für Wasserwirtschaft (eds) Modelle für die gesättigte und ungesättigte Bodenzone. Schriftenreihe d Bundesamtes f Wasserwirtschaft 7: 14-41.

Feichtinger F. (2015) personal communication, March 3, 2015

Franko U. (1997) Modellierung des Umsatzes der organischen Bodensubstanz. *Arch Acker Pfl Boden* 41: 527-547. Doi: 10.1080/03650349709366019

Groenendijk P., Heinen M., Klammler G., Fank J., Kupfersberger H., Pisinaras V., Gemitzi A., Peña-Haro S., García-Prats A., Pulido-Velazquez M., Perego A., Acutis M., Trevisan M. (2014) Performance assessment of nitrate leaching models for highly vulnerable soils used in low-input farming based on lysimeter data. *Sci Total Environ* 499: 463-480. Doi: 10.1016/j.scitotenv.2014.07.002

Gupta H.V. and Sorooshian S. (1998) Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resour Res* 34 (4): 751-763. Doi: 10.1029/97WR03495

Gupta H.V., Beven K.J. and Wagener T. (2005) Model calibration and uncertainty estimation. In: Anderson M. and McDonnel J. (eds) Encyclopedia of Hydrologic Sciences, Wiley, Chichester, UK.

Hansen S., Jensen H.E., Nielsen N.E. and Svendsen H. (1990) DAISY: soil plant atmosphere system model. NPO Report No. A 10. Copenhagen: The National Agency for Environmental Protection.

Hobisch M. (2014) Reengineering of a legacy complex FORTRAN 77 simulation system. Master’s thesis, Graz: University of Technology.

Ippisch O., Vogel H.J. and Bastian P. (2006) Validity limits for the van Genuchten-Mualem model and implications for parameter estimation and numerical simulation. *Adv Water Resour* 29: 1780-1789.

Jones J.W., Tsuji G.Y., Hoogenboom G., Hunt L.A., Thornton P.K., Wilkens P.W., Imamura D.T., Bowen W.T. and Singh U. (1998) Decision support system for agrotechnology transfer: DSSAT v3. In: Tsuji G.Y., Hoogenboom G. and Thornton P.K. (eds) Systems Approaches for Sustainable Agricultural Development, 7. Springer Netherlands: 157-177.

Jones J.W., Hoogenboom G., Porter C.H., Boote K.J., Batchelor W.D., Hunt L.A., Wilkens P.W., Singh U., Gijsman A.J. and Ritchie J.T. (2003) The DSSAT cropping system model. *Europ J Agronomy* 18: 235-265.

Kamali B., Mousavi S.J. and Abbaspour K.C. (2012) Automatic calibration of HEC-HMS using single-objective and multi-objective PSO algorithms. *Hydrol Process 27,* 4028-4042. Doi: 10.1002/hyp.9510

Kennedy J. and Eberhart R. (1995) Particle Swarm Optimization. *Proc IEEE Inter Conf on Neural Networks* 4: 1942-1948.

Klammler G. and Fank J. (2014) Determining water and nitrogen balances for beneficial management practices using lysimeters at Wagna test site (Austria). *Sci Tot Environ* 499: 448-462. Doi:10.1016/j.scitotenv.2014.06.009

Kühnel, Andreas (2013). Visual C# Das umfassende Handbuch. 6. Auflage. Bonn: Galleo Press. ISBN: 978-3-8362-1997-6.

Mallants D., Mohanty B.P., Vervoort A. and Feyen, J. (1997) Spatial analysis of saturated hydraulic conductivity in a soil with macropores. *Soil Technol* 10: 115-131. Doi: 10.1016/S0933-3630(96)00093-1

McKay M.D., Beckman R.J., Conover W.J. (1979) A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 21 (2): 239–245. Doi: 10.2307/1268522.

Mertens J., Madsen H., Kristensen M., Jacques D. and Feyen J. (2005) Sensitivity of soil parameters in unsaturated zone modelling and the relation between effective, laboratory and in situ estimates. *Hydrol Process* 19: 1611–1633. Doi: 10.1002/hyp.5591

Mertens J., Stenger R and Barkle G.F. (2006) Multiobjective Inverse Modeling for Soil Parameter Estimation and Model Verification. *Vadose Zone J*. 5: 917–933. Doi:10.2136/vzj2005.0117

Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H. and Teller, E. (1953) Equation of State Calculations by Fast Computing Machines. *J Chem Phys* 21: 1087-1091.

Monteith J.L. (1965) Evaporation and environment. *Sym Soc Exp Biol* 19: 205-224.

Mualem Y. (1976) A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resour Res* 12: 513-522.

Murer E. (1998) Die Ableitung der Parameter eines Bodenwasserhaushalts- und Stofftransportmodelles aus den Ergebnissen der Bodenkartierung. Schriftenreihe d Bundesamtes f Wasserwirtschaft 7: 89-103.

Nash, J. E. and Sutcliffe J.V. (1970) River flow forecasting through conceptual models part I - A discussion of principles, *J Hydrol* 10 (3), 282–290.

Ndiritu J.G. (2009) Automatic calibration of the Pitman model using the shuffled complex evolution method. Report to Water Research Commission School of Civil and Environmental Engineering. KV 229/09, University of the Witwatersrand.

Peng G. (2002) Environmental modelling and prediction, Springer, Berlin. ISBN: 9783540674221

Richards L.A. (1931) Capillary Conduction of Liquids through Porous Mediums. *Physics* 1 (5): 318-333. Doi: 10.1063/1.1745010

Rijtema P.E. and Kroes J.G. (1991) Some results of nitrogen simulations with the model ANIMO. *Fert Res* 27: 189-198.

Schaefli B. and Gupta H.V. (2007) Do Nash values have value? *Hydrol Process* 21: 2075 – 2080. Doi: 10.1002/hyp.6825

Schweizer, Wolfgang (2013). MATLAB kompakt. Oldenbourg Verlag, München. ISBN: 978-3-486-72114-0.

Stenitzer E. (1988) SIMWASER – Ein numerisches Modell zur Simulation des Bodenwasserhaushaltes und des Pflanzenertrages eines Standortes. Mitteilungen aus der Bundesanstalt für Kulturtechnik und Bodenwasserhaushalt 31, Petzenkirchen.

Stenitzer E., Diestel H., Franko U., Schwartengräber R., Zenker T. (2007) Performance of the model SIMWASER in two contrasting case studies on soil water movement. In: Kersebaum K.C. Hecker J.M., Mirschel W. and Wegehenkel M (eds) Modelling water and nutrient dynamics in soil-crop systems: 27-36. Doi: 10.1007/978-1-4020-4479-3_3

Storn R. and Price K. (1997) Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J Global Optim* 11: 341-359.

UMS (Umwelt-Monitoring-Systeme) (2016) Science-lysimeter http://www.ums-muc.de/en/lysimeter/science-lysimeter/ (accessed April 6, 2016)

van Genuchten, M.T. (1980) A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci. Soc. Am. J.* 44: 892–898. Doi:10.2136/sssaj1980.03615995004400050002x van Griensven A., Meixner T., Grunwald S., Bishop T., Diluzio M. and Srinivasan R. (2006). A global sensitivity analysis tool for the parameters of multi-variable catchment models. *J Hydrol* 326(2006): 10-23.

Vrugt J.A. and Robinson B.A. (2007) Improved evolutionary optimization from genetically adaptive multimethod search. *P. Natl. Acad. Sci. USA* 104(3): 708-711. Doi:10.1073/pnas.0610471104

Williams J.R, Jones C.A and Dyke, P.T. (1984) A modelling approach to determining the relationship between erosion and soil productivity. *Trans Am Soc Agric Eng 27*(1): 129-144.

Williams J.R., Jones C.A., Kiniry J.R. and Spanel, D.A. (1989) EPIC crop growth model. *Trans Am Soc Agric Eng* 32(2): 497-511.

Willmott C.J. (1982) Some comments on the evaluation of model performance. *Bull Am Meteorol Soc* 63: 1309-1313.

Woehling T., Vrugt J.A. and Barkle G.F. (2008) Comparison of Three Multiobjective Optimization Algorithms for Inverse Modeling of Vadose Zone Hydraulic Properties. *Soil Sci Soc Am J* 72: 305-319. Doi:10.2136/sssaj2007.017

Woehling T. and Vrugt J.A. (2011) Multiresponse multilayer vadose zone model calibration using Markov chain Monte Carlo simulation and field water retention data. *Water Resour Res* 47, W04510. Doi: 10.1029/2010WR009265

Wolpert D.H., Macready W.G. (1997) No Free Lunch Theorems for Optimization. *IEEE T Evolut Comput* 1: 67-82.

Yapo P.O., Gupta H.V. and Sorooshian S. (1998) Multi-objective global optimization for hydrologic models. *J Hydrol* 204: 83-97.

Zhang X, Srinivasan R and Van Liew M (2009) On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the SWAT model. *Hydrol Process* 24: 955-969. Doi: 10.1002/hyp.7528

8. List of Figures

Figure 1: Factors and parameters of water balance considered in SIMWASER; source: adapted from Stenitzer (1988) 5

Figure 2: Processes and factors of the nitrogen cycle considered in STOTRASIM; source: adapted from Feichtinger (1998) 7

Figure 3: Overview of the nitrogen fractions considered in STOTRASIM 8

Figure 4: Overview of the program process cycle SIMWASER/STOTRASIM. Source: adapted from Hobisch (2014) 9

Figure 5: Overview of the MatLab scripts (m-files) of AMALGAM 14

Figure 6: Screenshot of the MatLab environment 19

Figure 7: Screenshot of the Visual Studio environment. 20

Figure 8: The operation principle of Calisto 21

Figure 9: Inputs and Outputs transferred between AMALGAM (MatLab) and SIMWASER/STOTRASIM (C#) 22

Figure 10: Example for bimodal van Genuchten-Mualem retention and hydraulic conductivity curves and the corresponding 'standard curves' 26

Figure 11: Some of the 60 entries of the sheet '*Pflanzenparameter*' in '*Parametrisierung.xls'* 29

Figure 12: The sheet '*Bodenparameter*' in '*Parametrisierung.xls'*: the soil parameters of 6 soil types 29

Figure 13: The sheet '*Ertraege*' in '*Parametrisierung.xls'* 30

Figure 14: The sheet '*Evapotranspiration*' in '*Parametrisierung.xls'* 30

Figure 15: The sheet '*MatPot*' in '*Parametrisierung.xls'* 30

Figure 16: The sheet '*WasAnt*' in '*Parametrisierung.xls'* 31

Figure 17: The sheet ‚*Nitratkonzentrationen*' in ‚*Parametrisierung.xls*' 31

Figure 18: Pop-up window for the specification of the maximum number of function evaluations 32

Figure 19: Pop-up window for the specification of the evaporation coefficient '*rs*' and the mineralization multiplier '*potfak*'. 32

Figure 20: Pop-Up window asking if starting values and/or parameter ranges should be predefined 34

Figure 21: MatLab dialog boxes for the definition of starting values and/or ranges of the parameters that should be optimized in Calisto 35

Figure 22: The MatLab water content matrix (wasant1) for the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1 36

Figure 23: Screenshot of the sheet '*Überblick*' of the Excel file '*Calisto_run1.xls'* 44

Figure 24: Graph with comparison of simulated (green) and measured (blue) yields (dry matter) in the sheet '*Überblick*' in '*Calisto_run1.xls'* 45

Figure 25: Graph with comparison of simulated (dotted) and measured (solid) time series of water content in the sheet '*Überblick*' in '*Calisto_run1.xls'* 46

Figure 26: Screenshot of the sheet '*Bodeninput*' with the optimized soil parameters in form of bimodal van Genuchten-Mualem parameters and 'standard curve' 47

Figure 27: Accordance of standard curves (red dots) and curves created by standardized bimodal van Genuchten-Mualem parameters (green lines) for the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1 51

Figure 28: Comparison of measured (red dots) and simulated (blue lines) matric potentials in 10, 20, 35, 60, 90 and 180 cm depth 53

Figure 29: Comparison of measured (red dots) and simulated (blue lines) water contents in 35, 60, 90 and 180 cm depth 54

Figure 30: Comparison of measured (red dots) and simulated (blue lines) nitrate concentration in 35, 60, 90 and 180 cm depth 55

Figure 31: Comparison of measured (red) and simulated (blue) yields (dry weight) of grassland in 2004, pumpkin in 2005, grassland in 2006, maize 300 in 2006, green rye in 2007, maize 300 in 2007 and winter barley in 2008....................... 55

Figure 32: Performance of different evaluation configurations of Calisto for optimizing parameters '*rs*' and *'potfak',* visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentrations in four depths) 59

Figure 33: Command Window displaying the run times of a failed Calisto optimization .. 60

Figure 34: Performance of different evaluation configurations of Calisto for optimizing parameters *'ext'*, '*bfgw', '*as*'*, '*rlg*' and '*ripe*' of maize, visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentration in four depths)... 62

Figure 35: Performance of different evaluation configurations of Calisto for optimizing parameters 'as' and 'ripe' of the five plants winter barley, winter rye, maize, grassland and pumpkin, visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentration in four depths) 64

Figure 36: Command Window displaying the run times 65

Figure 37: Performance of different evaluation configurations of Calisto for optimizing parameters ‘α1’, ‘α2’, ‘n1’, ‘n2’, ‘w’, ‘Ks’ and ‘τ’ of the soil layer SSG5H1, visualized by the results of 11 objective functions (yields, evapotranspiration, matric potential in six depths, water content in four depths and nitrate concentration in four depths) 71

9. List of Tables

Table 1: Default values for the parameter ranges in Calisto 33

Table 2: Soil horizons, textures and estimated diffusion/dispersion parameters for the layers in the lysimeter 49

Table 3: Standardized bimodal van Genuchten-Mualem parameters for the soil types SL3G4H3, LS3G3H2, LS4G3H2, SL3G2H1 and SSG5H1 49

Table 4: Standardized plant parameters for the use in SIMWASER/STOTRASIM for the plants winter wheat, winter barley, winter rye, maize, grass and pumpkin 52

Table 5: Nash Suftcliffe efficiencies of matric potential (10, 20, 35, 60, 90 and 180 cm), water contents (35, 60, 90, 180 cm), nitrate concentration (35, 60, 90 and 150cm) and the sum of squared errors of the yields (above-ground dry mass).. 56

Table 6: Calibration performance for optimizing parameters '*rs*' and '*potfak*'; parameter units are defined in Table 1 58

Table 7: Calibration performance for optimizing parameters 'ext', 'bfgw', 'as', 'rlg' and 'ripe' of maize, excluding failed optimization runs; parameter units are defined in Table 1 61

Table 8: Calibration performance for optimizing parameters 'as' and 'ripe' of winter barley, winter rye, maize, grassland and pumpkin; excluding failed optimization runs; parameter units are defined in Table 1 63

Table 9: Calibration performance for optimizing soil parameters 'α1', 'α2', 'n1', 'n2', 'w', 'Ks' and 'τ' of the deepest soil layer SSG5H1; parameter units are defined in Table 1 67

- FOS (fresh organic substance): Includes the organic import in the soil (plant residues, mineral fertilizer and compost), which are broken down in a few months.
- AOS (active organic substance): This soil-born fraction is fast decomposable and mineralized within a few years. There can be soil organisms living in it.
- SOS (stabilized organic substance): This soil-born fraction is slowly decomposable and mineralized within a few decades.
- IOS (inert organic substance): This soil-born fraction is inert and its mineralization takes place within several centuries or even millennia.

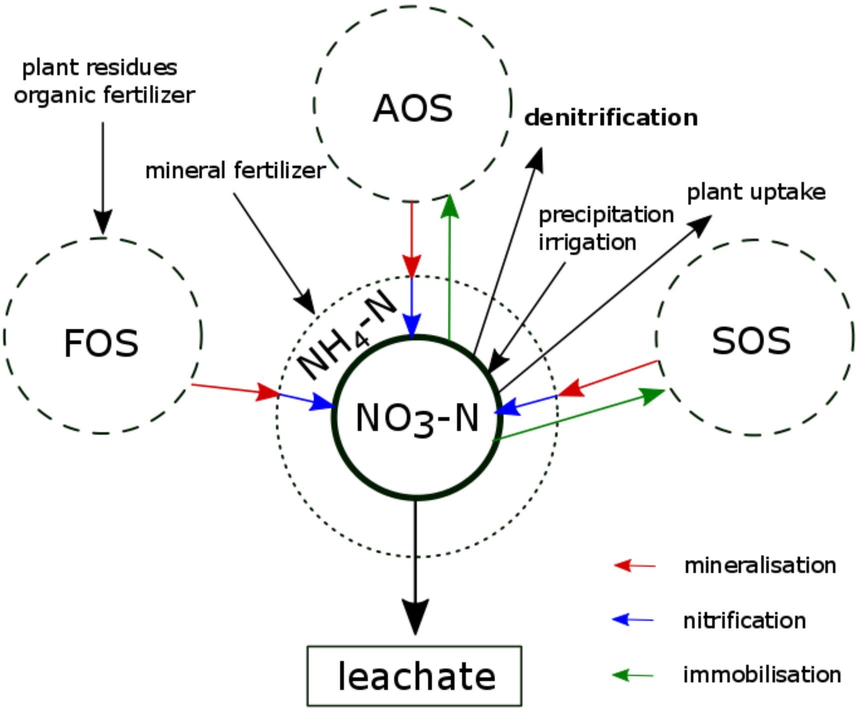


Figure 3: Overview of the nitrogen fractions considered in STOTRASIM

Mineralisation of humus is heavily influenced by the variable ‘*potfak*’. The mineralization multiplier ‘*potfak*’ can be defined by the user when starting the simulation model or can be determined by the program itself. The nitrate-nitrogen is transported by advection and diffusion/dispersion. According to the potential gradient either nitrogen leaching or capillary rise takes place. The main focus of this model is set on the leachate into groundwater. For a more detailed description of the model characteristics I refer to Feichtinger (1998).

2.1.2 Model implementation

SIMWASER/STOTRASIM had originally been programmed in the higher-level language Fortran 77. In 2013 the model was reengineered in C# (Hobisch, 2014).

The program starts with reading in the project file, which includes configuration parameters (including '*rs*' and '*potfak*', 2.1.1) and the variants that should be simulated, also called runs or projects. Afterwards the first loop over the variants is started, followed by the crop rotation loop, the day loop and the time step loop. The length of the time step cannot be defined by the user, but is chosen by the program according to the flow velocity in the soil. An overview of the program process is given in Figure 4.

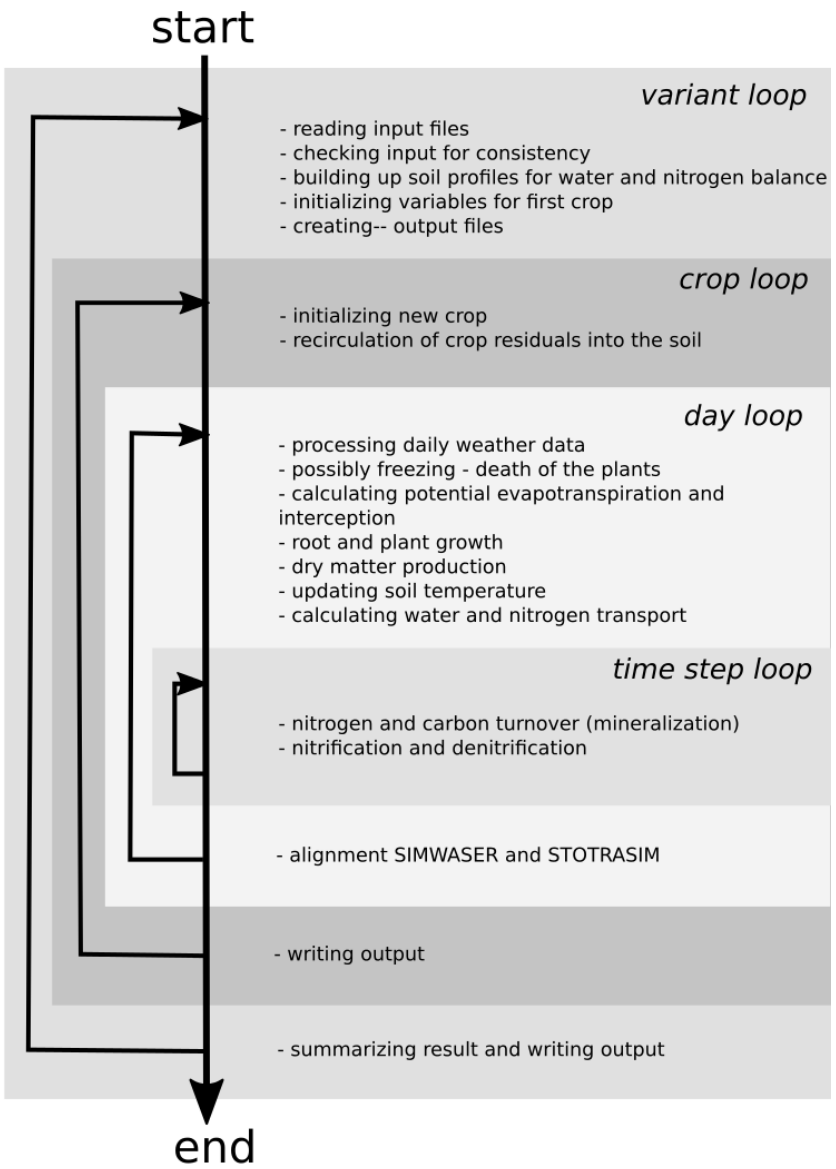

start
variant loop
- reading input files
- checking input for consistency
- building up soil profiles for water and nitrogen balance
- initializing variables for first crop
- creating-- output files
crop loop
- initializing new crop
- recirculation of crop residuals into the soil
day loop
- processing daily weather data
- possibly freezing - death of the plants
- calculating potential evapotranspiration and interception
- root and plant growth
- dry matter production
- updating soil temperature
- calculating water and nitrogen transport
time step loop
- nitrogen and carbon turnover (mineralization)
- nitrification and denitrification
- alignment SIMWASER and STOTRASIM
- writing output
- summarizing result and writing output
end

Figure 4: Overview of the program process cycle SIMWASER/STOTRASIM; source: adapted from Hobisch (2014)

3. Calisto – The Calibration Program

The main objective of this master project is to implement a tool for the automatic calibration of 'STOTRASIM C#' with AMALGAM. Therefore SIMWASER/STOTRASIM (programmed in C#) and AMALGAM, (programmed in MatLab) had to be linked. The resulting software shall henceforward be called Calisto ('Calibration Stotrasim').

The general optimization procedure is that SIMWASER/STOTRASIM receives parameter sets that were created within MatLab, executes the simulation and passes back the simulation results to MatLab, where the simulated data series can be compared to the observed behavior of the system. AMALGAM ranks the parameter sets according to their ability to reproduce the measured system behaviour and creates a new generation of data sets (as described in 2.2.1). This is then passed to SIMWASER/STOTRASIM in a loop process (Figure 8).

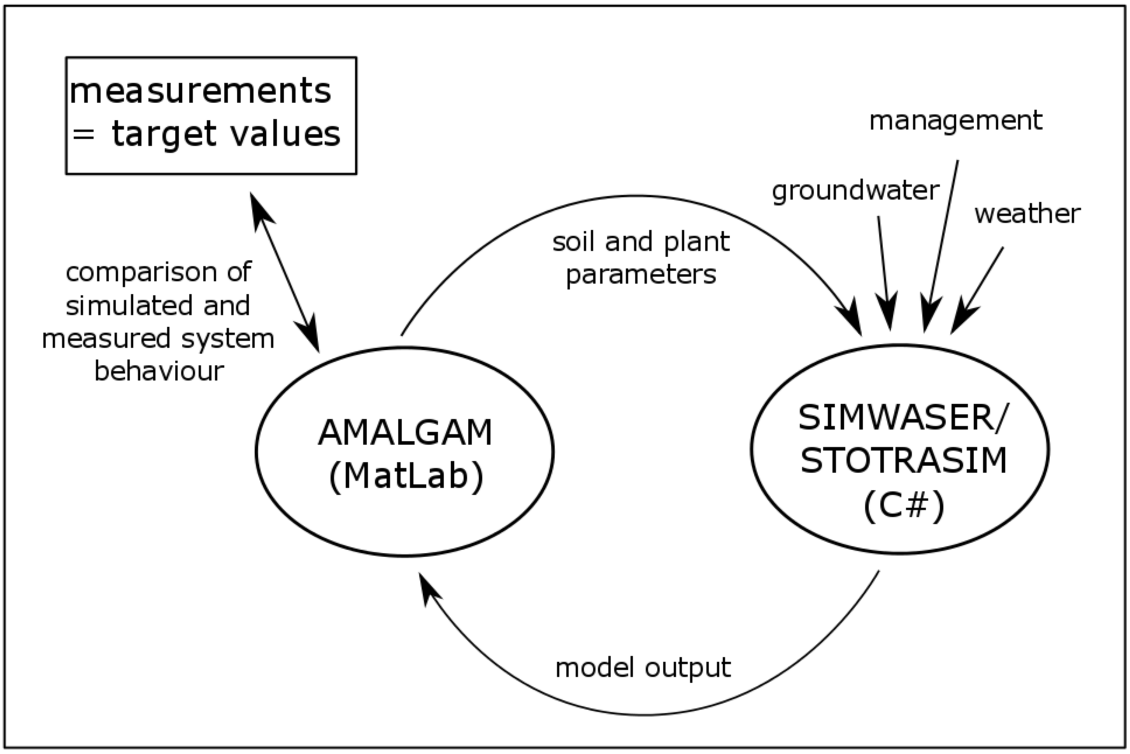


Figure 8: The operation principle of Calisto

In the past, all input files needed (2.1.3) had been read in and output files had been written by SIMWASER/STOTRASIM C# in form of text files. For Calisto, however, all the input parameters that should be calibrated have to be passed on from AMALGAM (MatLab) to SIMWASER/STOTRASIM (C#). In just the same way, the model results have to be passed from C# to MatLab so that AMALGAM can compare these results with the measurements. Therefore communication and ways of reciprocal data exchange between AMALGAM (MatLab) and SIMWASER/STOTRASIM (C#) had to be established. For the input of the soil characteristics, model parameters had to be defined (3.2). These can then be transformed into ‘standard curves’, which are necessary for SIMWASER/STOTRASIM calculation (2.1.3). Moreover there is a need for the user to enter measured values and the parameters that do not need to be calibrated. Figure 9 shows the operation principle of Calisto extended with the input and output matrices and variables that are transferred between AMALGAM (MatLab) and SIMWASER/STOTRASIM (C#).

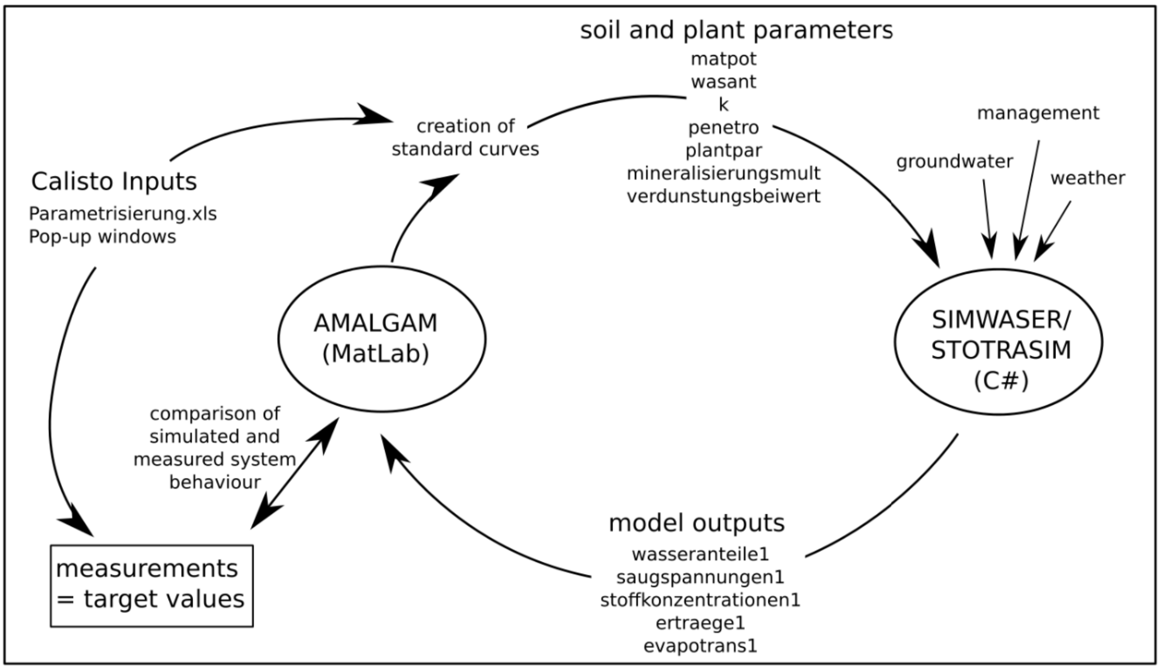


Figure 9: Inputs and Outputs transferred between AMALGAM (MatLab) and SIMWASER/STOTRASIM (C#)

Moreover it is necessary to define starting values and ranges for the input parameters and to write an output of the optimization process. For the performance of the optimization it is crucial to formulate appropriate objective functions. All relevant work steps are explained subsequently.

3.1 Calling C# with MatLab

SIMWASER/STOTRASIM, which had been a C# console application before, is saved as .NET assembly. The ‘Main Function’ that specifies the starting point of the program execution is renamed and attributed as ‘public’, so that it can be accessed externally. The .NET assembly is compiled to a .DLL-file, so the application is not self-launching any more. The