MIKE
Powered by DHI

AutoCal

Auto Calibration Tool

User Guide

MIKE 2025

PLEASE NOTE

COPYRIGHT

LIMITED LIABILITY

The liability of DHI is limited as specified in your DHI Software Licence Agreement:

In no event shall DHI or its representatives (agents and suppliers) be liable for any damages whatsoever including, without limitation, special, indirect, incidental or consequential damages or damages for loss of business profits or savings, business interruption, loss of business information or other pecuniary loss arising in connection with the Agreement, e.g. out of Licensee's use of or the inability to use the Software, even if DHI has been advised of the possibility of such damages.

This limitation shall apply to claims of personal injury to the extent permitted by law. Some jurisdictions do not allow the exclusion or limitation of liability for consequential, special, indirect, incidental damages and, accordingly, some portions of these limitations may not apply.

Notwithstanding the above, DHI's total liability (whether in contract, tort, including negligence, or otherwise) under or in connection with the Agreement shall in aggregate during the term not exceed the lesser of EUR 10.000 or the fees paid by Licensee under the Agreement during the 12 months' period previous to the event giving rise to a claim.

Licensee acknowledge that the liability limitations and exclusions set out in the Agreement reflect the allocation of risk negotiated and agreed by the parties and that DHI would not enter into the Agreement without these limitations and exclusions on its liability. These limitations and exclusions will apply notwithstanding any failure of essential purpose of any limited remedy.

CONTENTS

1 Auto Calibration Tool . . . 7
1.1 New AutoCal Dialog . . . 9
1.2 Simulation Specifications . . . 9
1.2.1 Model simulation sequence . . . 10
1.2.2 Model parameter files . . . 11
1.2.3 Simulation option . . . 12
1.2.4 Simulation title . . . 12
1.3 Model Parameters . . . 12
1.4 Objective Functions . . . 15
1.4.1 Output measures . . . 16
1.4.2 Objective functions . . . 18
1.4.3 Evaluation period . . . 19
1.4.4 Aggregation of objective functions . . . 19
1.5 Scenario Runs . . . 20
1.5.1 Scenario type . . . 21
1.5.2 Parameters . . . 21
1.6 Sensitivity Analysis . . . 21
1.6.1 Sensitivity analysis method . . . 22
1.6.2 Local sensitivity analysis . . . 22
1.7 Parameter Optimisation . . . 24
1.7.1 Optimisation method . . . 24
1.7.2 Shuffled Complex Evolution method . . . 25
1.7.3 Population Simplex Evolution method . . . 28
1.8 Save Output Files . . . 30
1.9 Simultaneous simulations . . . 31
1.10 Start AutoCal Simulation . . . 33
1.11 Output . . . 33
1.11.1 General output files . . . 33
1.11.2 Sensitivity analysis output . . . 33
1.11.3 Optimisation output . . . 34
1.12 Error handling . . . 35

2 Examples . . . 37
2.1 MIKE 21 . . . 37
2.1.1 Example 1 . . . 37
2.2 MIKE SHE . . . 40
2.2.1 Example 1 . . . 40
2.2.2 Example 2 . . . 42

Index 47

1 Auto Calibration Tool

AutoCal is a generic tool for performing automatic calibration, parameter optimisation, sensitivity analysis and scenario management of the numerical modelling engines under MIKE Zero. It interfaces the modelling engines using the MIKE Zero PFS (parameter files system) format for model input and the DFS (data file system) format for model output. AutoCal can also be linked to other modelling engines by interfacing ASCII model input files and creating a bridge between the model output and the DFS file format.

The core of the AutoCal tool consists of the following steps:

1. Provision of a set of model parameter values to the numerical model to be used in a specific model run.
2. Execution of the simulation model using the specified set of model parameter values.
3. Calculation of statistical performance measures of the model output.

The model parameter interface is made via a so-called template file. This file is simply a replica of the model input file in which parameter identification tags are placed at the locations where the numerical values of the model parameters are given. Thus, when AutoCal has to provide a new set of model parameters to the model, the parameter identification tags in the template file are substituted by the numerical values and saved in the model input file to be used when running the model. It is possible to manipulate model parameters in different model input files. For each model input file a corresponding template file is defined.

NOTE: The character “!” is used as a delimiter in the template file to identify the parameters to be manipulated during the simulation. I.e. the character “!” is a reserved character for AutoCal and cannot be used anywhere else in the model input, e.g. in text strings or file names.

AutoCal allows parameters to be defined as functions of other parameters that are manipulated in the AutoCal run. In this case general equations can be specified that defines the relations between the dependent model parameter and the other parameters. AutoCal provides the set of independent parameter values and the dependent parameters are then calculated automatically from these values using the defined equations. Both dependent and independent parameter values are then substituting their corresponding parameter identification tags in the template files. Besides the standard arithmetic operators, the equation tool includes a large number of mathematical functions.

A parameter may also be defined as a constant. In this case the constant value defined in AutoCal is substituting the corresponding parameter identification tag in the template file. This feature is especially useful when a sensitivity analysis is performed prior to the parameter optimisation. In this case the sensitivity analysis typically includes a long list of parameters. From the results of the sensitivity analysis the most sensitive parameters are retained in the subsequent parameter optimisation. Instead of making a new AutoCal setup, the same setup as used in the sensitivity analysis can be applied simply by setting the insensitive parameters to constant values.

A model simulation in AutoCal can be defined as a sequence of individual model runs. AutoCal supports execution of all the MIKE Zero model engines. In this case the model simulation setup file is given as input. In addition, AutoCal enables execution of any executable (.exe) or batch (.bat) file. In this case the executable as well as the corresponding list of arguments are given as input.

Whether AutoCal is used for parameter optimisation, sensitivity analysis or scenario management, the performance of the model simulation given the specified parameter set should be assessed. This is done by calculating statistical performance measures. These measures are typically comparison statistics that compare measurements or, in general, target values with corresponding simulated values.

For calculating the comparison statistics AutoCal requires that simulation results and corresponding observations are given as time series in DFS0 files. If the output from a model engine is not explicitly given in DFS0 format, a processing of simulation results is required to transform the simulation results at measurement locations into DFS0 format. This post-processing is then part of the sequence of model runs defined in AutoCal. Typically post-processing is required for MIKE 11 result files (.res11) and for models that provide results in 1D, 2D or 3D grid files (.dfs1, .dfs2 and.dfs3). For application of AutoCal with modelling engines that are not included in MIKE Zero the COM components included in the MIKE Objects Time Series Package can be used to create DFS0 time series files from the model output files.

Before setting up AutoCal, the simulation model should be properly tested. At least one model run should be performed to create the output files that are needed in the AutoCal setup.

Folder structure

AutoCal requires that the setup of the model and any linked sub-models to be used by AutoCal is placed in a sub-folder of the folder with the AutoCal setup file.

Note: All files which are referenced as a parameter file in the .auc setup must be written within one folder (i.e. the same sub-folder as the original setup file, e.g. .she).

Example:

C:\temp\MyAutocal.auc
C:\temp\MyModel\

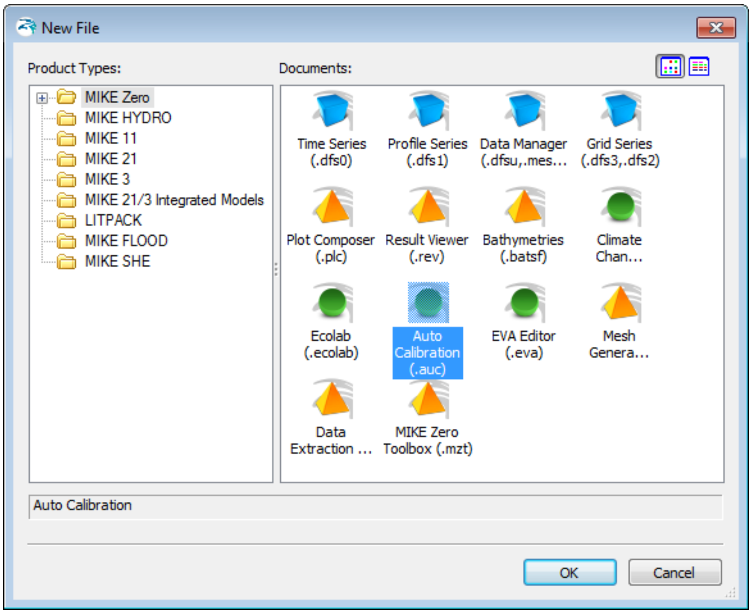
When AutoCal is launched, copies of the model setup folder will be created (one for each simultaneous simulation) and used for individual model simulations by AutoCal.

For example, if the number of simultaneous simulations is set to four, the following sub-folders will be created:

C:\temp\MyAutocal.auc
C:\temp\MyModel_Process1\
C:\temp\MyModel_Process2\
C:\temp\MyModel_Process3\
C:\temp\MyModel_Process4\

1.1 New AutoCal Dialog

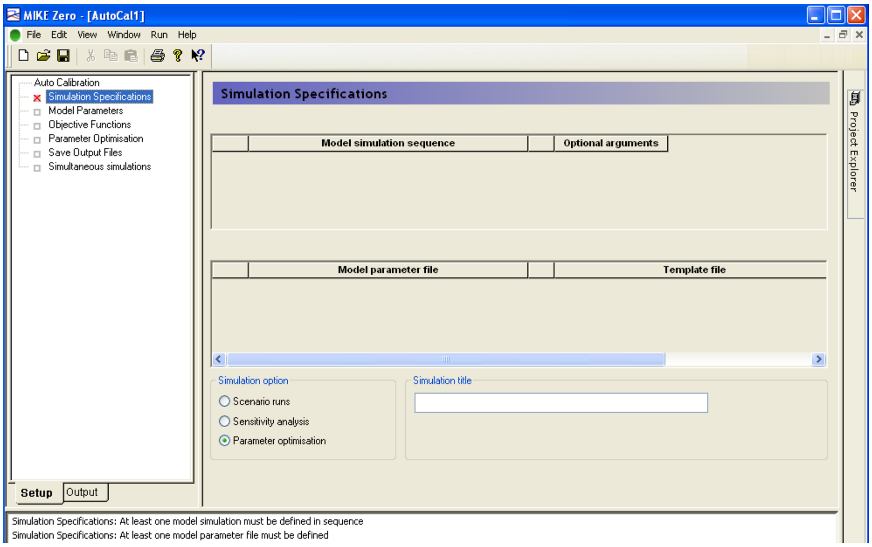
To create a new AutoCal setup, use the File | New | File... top menu item, or the New icon. This will open the New File dialog where you can select Auto Calibration (.auc) from the MIKE Zero file types.



Auto Calibration

1.2 Simulation Specifications

On the *Simulation Specifications* page, the model simulation sequence, the model parameter files, and the simulation option are specified.



Simulation Specifications

	Model simulation sequence		Optional arguments

	Model parameter file		Template file

Simulation option
○ Scenario runs
○ Sensitivity analysis
● Parameter optimisation

Simulation title

Simulation Specifications: At least one model simulation must be defined in sequence
Simulation Specifications: At least one model parameter file must be defined

Figure 1.1 Simulation specifications page

1.2.1 Model simulation sequence

A model simulation in AutoCal can be defined as a sequence of individual model runs. For example, a modelling sequence may consist of a hydrodynamic (HD) model simulation followed by an advection-dispersion (AD) model simulation that uses the HD output. In this case two individual model runs must be defined in the *Model simulation sequence* table, including the HD run in the first row and the AD run in the second row. Another typical example is a simulation followed by the extraction and savings of simulation results at measurement points to DFS0 files.

AutoCal will execute MIKE Zero based numerical calculation kernels as well as any executable (.exe) or batch (.bat) file. In this case the model simulation file for the numerical engine (e.g. a .sim11 file for a MIKE 11 run and a .m21fm file for a MIKE 21 run) or the name of the .exe or .bat file must be specified in the *Model simulation sequence* column.

If the executable file requires arguments, such as an input file name or some simulation options, these are written in the *Optional arguments* column.

For post-processing of model results, AutoCal can extract time series from 1D (dfs1), 2D (dfs2) and 3D(dfs3) result files using the MIKE Zero Toolbox. The toolbox action is saved as a .mzt file and can be executed by AutoCal by specifying the .mzt file in the *Model simulation sequence* table.

In MIKE SHE and MIKE 11 HD you can specify time series output that can be used directly by AutoCal.

To extract time series of other MIKE 11 variables from a MIKE 11 result file, the *res11read.exe* program can be used. The *Model simulation sequence* column should then include the res11read input file (.res11read), which specifies the data to be extracted from the res11 file and saved in a dfs0 file. This file should contain the following:

-silent -someresSpecFileName -MakeDfs0 Res11FileName Dfs0FileName

where *SpecFileName* is the name of the file that contains specifications of the data to be extracted from the .res11 result file (item, chainage and rivername), *Res11FileName* is the name of the .res11 result file, and *Dfs0FileName* is the root name of the dfs0 file where the extracted time series are to be saved.

Note the *Dfs0FileName* file name must not include the .dfs0 extension.

Note the relative path names should be used for the specified files.

More information about the res11read application program is given in the MIKE 11 User Guide, Appendix B.

1.2.2 Model parameter files

In the *Model parameter files* table, the names of the files containing the model parameters to be manipulated by AutoCal are specified.

AutoCal can manipulate parameters in the model parameter files associated with the numerical calculation kernels. For some simulation models only one model parameter file is associated, but for some models (e.g. MIKE 11 and MIKE SHE) several model parameter files can be manipulated. AutoCal can also manipulate parameters in any ASCII file.

When a model parameter file is selected, a corresponding template file is automatically created -with *_Autocal* added to the file name. This template file is simply a replica of the model parameter file. It is used for placing parameter identification tags at the locations where the values of the model parameters to be manipulated in AutoCal are given. The template file can be directly edited from AutoCal. For MIKE Zero PFS files, the corresponding editor is opened. For ASCII files, Notepad is opened.

Identification tags for model parameters

In AutoCal, reserved floating point values are used as identification tags. The identification tags are the values 1.01e-35, 1.02e-35, ..., 9.99e-35. In the model editor, the identification tag is placed in the field where the numerical value of the model parameter is given. Individual parameters must be given unique identification tags.

If several model parameters should be given the same parameter value in AutoCal, the same identification tag can be used. If different identification tags are used, then the parameters can be coupled using the parameter options on the *Model Parameters* tab.

When the template file has been created and the parameter identification tags have been entered, a table with the model parameters is automatically created on the *Model Parameters* tab.

When running AutoCal, a second template file is created (.tpl is added to the file name) that is used internally by AutoCal.

Units used in AutoCal

AutoCal optimizes the parameters stored in the pfs file. However, parameters stored in these files are stored using the base units for the parameter type. The base units are always standard SI units. Thus, if you are not using the base parameter units (e.g. feet instead of metres), AutoCal will ignore these unit changes. Thus, you must always use the base units if you want to use AutoCal - or at least be aware that the optimized values output by AutoCal are in the base SI units.

1.2.3 Simulation option

AutoCal supports three different simulation options:

- Scenario runs
- Sensitivity analysis
- Parameter optimisation

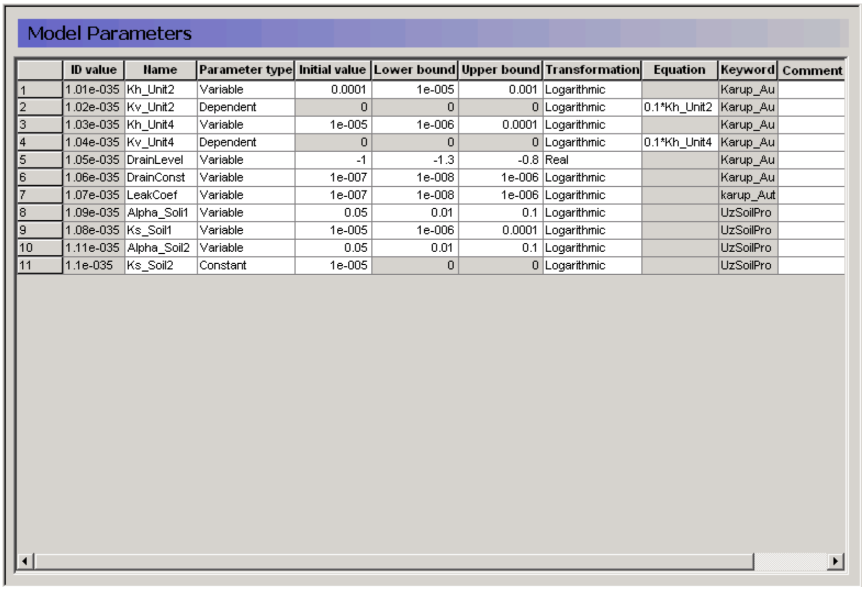
Depending on the choice a corresponding property page is shown in the tree view.

1.2.4 Simulation title

Optionally a simulation title can be specified.

1.3 Model Parameters

On the *Model Parameters* page (Figure 1.2) the properties of the model parameters that are defined in the model parameter files are specified. The parameter table is automatically created by AUTOCAL based on the specifications given in the template files on the *Simulation Specifications* page.



Model Parameters

	ID value	Name	Parameter type	Initial value	Lower bound	Upper bound	Transformation	Equation	Keyword	Comment
1	1.01e-035	Kh_Unit2	Variable	0.0001	1e-005	0.001	Logarithmic		Karup_Au	
2	1.02e-035	Kv_Unit2	Dependent	0	0	0	Logarithmic	0.1*Kh_Unit2	Karup_Au	
3	1.03e-035	Kh_Unit4	Variable	1e-005	1e-006	0.0001	Logarithmic		Karup_Au	
4	1.04e-035	Kv_Unit4	Dependent	0	0	0	Logarithmic	0.1*Kh_Unit4	Karup_Au	
5	1.05e-035	DrainLevel	Variable	-1	-1.3	-0.8	Real		Karup_Au	
6	1.06e-035	DrainConst	Variable	1e-007	1e-008	1e-006	Logarithmic		Karup_Au	
7	1.07e-035	LeakCoef	Variable	1e-007	1e-008	1e-006	Logarithmic		karup_Aut	
8	1.09e-035	Alpha_Soli1	Variable	0.05	0.01	0.1	Logarithmic		UzSoilPro	
9	1.08e-035	Ks_Soil1	Variable	1e-005	1e-006	0.0001	Logarithmic		UzSoilPro	
10	1.11e-035	Alpha_Soil2	Variable	0.05	0.01	0.1	Logarithmic		UzSoilPro	
11	1.1e-035	Ks_Soil2	Constant	1e-005	0	0	Logarithmic		UzSoilPro	

Figure 1.2 Model Parameters page

The following properties are specified in the table:

ID value

The ID value is the identification tag value given for the parameter in the template file.

Keyword/Line no

The keyword is an identification string that shows the location of the parameter in the PFS model input file. The first part of the string is the name of the template file. This is followed by the hierarchy of PFS sections separated by dots. The last part of the string is the PFS keyword. If the parameter file is not a MIKE Zero PFS file, the line No. where the parameter is located is shown.

For each parameter, the following has to be specified:

Name

The user must specify a unique name for each parameter. This name must not include white spaces. In addition, if the parameter is included as an independent parameter in an equation, arithmetic symbols and function names used by the equation parser must not be used as part of the parameter name.

Parameter type

The parameter may be defined as either a *Variable* parameter, a *Constant* parameter or a *Dependent* parameter. A variable parameter is a parameter that is changed by AutoCal according to the chosen simulation option. For a variable parameter the *Initial value*, *Lower bound* and *Upper bound* need to be defined. A constant parameter is set to the value defined in the *Initial value* field. A dependent parameter is defined as a function of the other parameters. In this case the *Equation* must be specified.

Initial value

The *Initial value* is the value used by AutoCal for performing a single scenario run. If the *Local sensitivity analysis* option is chosen, the sensitivity coefficients are evaluated around the initial parameter set.

Lower bound

The *Lower bound* specifies the lower limit of the feasible parameter values in the parameter optimisation.

Upper bound

The *Upper bound* specifies the upper limit of the feasible parameter values in the parameter optimisation.

Transformation

The parameter may be used in AutoCal as its native value by setting the transformation field to *Real* or as its logarithmic transformed value by setting the transformation field to *Logarithmic*. A logarithmic transformation is generally recommended if the feasible range of the parameter varies over orders of magnitude.

Equation

If a parameter is defined as a dependent parameter, an equation must be given to define the parameter as a function of the available variable parameters. AutoCal uses an equation parser that supports the general arithmetic operators (+,-,*,/) as well as a number of mathematical functions. The list of available mathematical functions is given in Table 1.1.

Table 1.1 Mathematical functions used by the equation parser (X and Y are variable names)

Syntax	**Function**
SQR(X)	Square function
SQRT(X)	Square root function
SIN(X)	Sinus function. SIN returns the sine of the angle X in radians
COS(X)	Cosinus function. COS returns the cosine of the angle X in radians
TAN(X)	Tangent function. TAN returns the tangent of the angle X in radians
COTAN(X)	Cotangent function. COTAN returns the cotangent of the angle X in radians
ATAN(X)	ArcTangent function

Table 1.1 Mathematical functions used by the equation parser (X and Y are variable names)

Syntax	Function
EXP(X)	Exponential function
LN(X)	Natural logarithmic function
LOG(X)	10 based logarithmic function
SINH(X)	Sinus Hyperbolic function
COSH(X)	Cosinus Hyperbolic function
INTPOW(X,Y)	The INTPOW function raises X to an integer power Y, e.g. INTPOW(2, 3) = 8. Note that the result of INTPOW(2, 3.4) = 8 as well
POW(X,Y)	The POW function raises X to any power Y
ABS(X)	Absolute value
SIGN(X)	SIGN(X) returns -1 if X<0; +1 if X>0, 0 if X=0
TRUNC(X)	Discards the fractional part of a number, e.g. TRUNC(3.2) is 3
MIN(X,Y)	Minimum of X and Y, e.g. MIN(2,3) is 2
MAX(X,Y)	Maximum of X and Y, e.g. Max(2,3) is 3

As an example, suppose Y is to be expressed as a function of the variables X1, X2 and X3 as 5 times variable X1 minus the square of variable X2 plus 2 times the natural logarithm of X3, the *Equation* field for Y should be written:

5*X1-SQR(X2)+2*LN(X3)

Comment

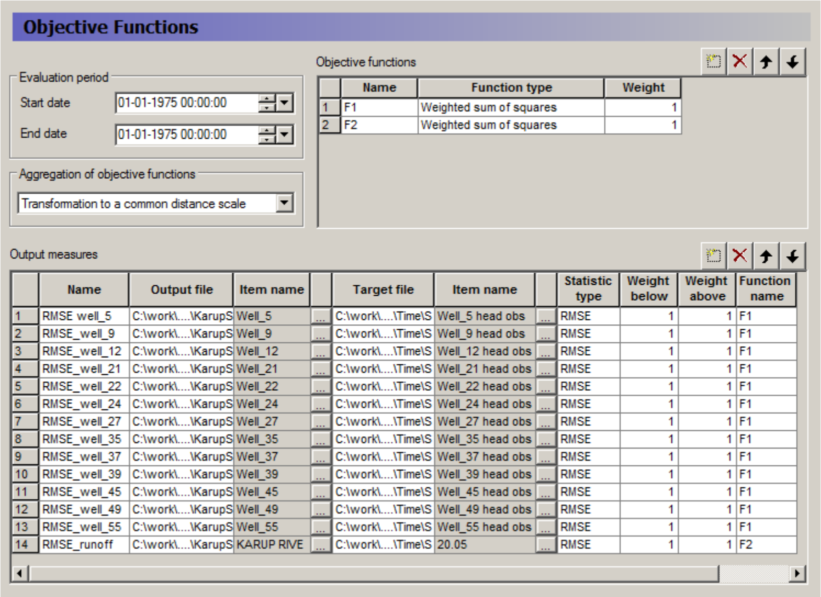
Optionally a comment can be written for the parameter.

Model parameter update

If parameter values are updated or deleted in the model parameter files, the model parameter table is automatically updated.

1.4 Objective Functions

On the *Objective Functions* page (Figure 1.3) the properties for calculation of comparison statistics are specified. The basic statistics used by AutoCal are the *Output measures* that include a single comparison statistic between an observed and a simulated time series. These basic measures can then be aggregated into different *Objective functions*, for instance according to spatial location, type of variable, or type of statistic. Finally, the defined objective functions are aggregated into a single statistic that is used by the optimisation algorithm.



Objective Functions

Evaluation period

Start date: 01-01-1975 00:00:00

End date: 01-01-1975 00:00:00

Aggregation of objective functions: Transformation to a common distance scale

Objective functions

	Name	Function type	Weight
1	F1	Weighted sum of squares	1
2	F2	Weighted sum of squares	1

Output measures

	Name	Output file	Item name		Target file	Item name		Statistic type	Weight below	Weight above	Function name
1	RMSE well_5	C:\work\....\KarupS	Well_5	...	C:\work\....\Time\S	Well_5 head obs	...	RMSE	1	1	F1
2	RMSE_well_9	C:\work\....\KarupS	Well_9	...	C:\work\....\Time\S	Well_9 head obs	...	RMSE	1	1	F1
3	RMSE_well_12	C:\work\....\KarupS	Well_12	...	C:\work\....\Time\S	Well_12 head obs	...	RMSE	1	1	F1
4	RMSE_well_21	C:\work\....\KarupS	Well_21	...	C:\work\....\Time\S	Well_21 head obs	...	RMSE	1	1	F1
5	RMSE_well_22	C:\work\....\KarupS	Well_22	...	C:\work\....\Time\S	Well_22 head obs	...	RMSE	1	1	F1
6	RMSE_well_24	C:\work\....\KarupS	Well_24	...	C:\work\....\Time\S	Well_24 head obs	...	RMSE	1	1	F1
7	RMSE_well_27	C:\work\....\KarupS	Well_27	...	C:\work\....\Time\S	Well_27 head obs	...	RMSE	1	1	F1
8	RMSE_well_35	C:\work\....\KarupS	Well_35	...	C:\work\....\Time\S	Well_35 head obs	...	RMSE	1	1	F1
9	RMSE_well_37	C:\work\....\KarupS	Well_37	...	C:\work\....\Time\S	Well_37 head obs	...	RMSE	1	1	F1
10	RMSE_well_39	C:\work\....\KarupS	Well_39	...	C:\work\....\Time\S	Well_39 head obs	...	RMSE	1	1	F1
11	RMSE_well_45	C:\work\....\KarupS	Well_45	...	C:\work\....\Time\S	Well_45 head obs	...	RMSE	1	1	F1
12	RMSE_well_49	C:\work\....\KarupS	Well_49	...	C:\work\....\Time\S	Well_49 head obs	...	RMSE	1	1	F1
13	RMSE_well_55	C:\work\....\KarupS	Well_55	...	C:\work\....\Time\S	Well_55 head obs	...	RMSE	1	1	F1
14	RMSE_runoff	C:\work\....\KarupS	KARUP RIVE	...	C:\work\....\Time\S	20.05	...	RMSE	1	1	F2

Figure 1.3 Objective Functions page

1.4.1 Output measures

Name

Name of the output measure.

Output file and item name

File name and corresponding item name of the time series of the simulation results.

Target file and item name

File name and corresponding item name of the target time series.

Statistic type

AutoCal includes three basic comparison statistics:

Average error (*Avg. Error*):

$$AE = \frac{1}{N}\sum_{i=1}^{N} w_i(SIM_i - TARGET_i) \qquad (1.1)$$

Root mean square error (*RMSE*):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} w_i(SIM_i - TARGET_i)^2} \tag{1.2}$$

Standard deviation of residuals (*St.Dev.*):

$$STD = \sqrt{\frac{1}{N}\sum_{i=1}^{N} w_i(SIM_i - TARGET_i - AE)^2} \tag{1.3}$$

where *TARGET*$_i$ and *SIM*$_i$, *i* = 1,..,*N* are the observed and the corresponding simulated time series, respectively, and *w*$_\text{i}$ is a user specified weight (weight below and weight above). Before calculation of the statistics, the time series are synchronised; that is, simulated values are extracted at the same time instants as the available target values using linear interpolation.

The three statistics are linked via the equation:

$$RMSE^2 = AE^2 + STD^2 \tag{1.4}$$

The statistic *AE* is a measure of the general offset between targets and simulations (bias), whereas *STD* is a measure of the dynamical correspondence. *RMSE* is an aggregated measure that includes both bias and dynamical correspondence.

Besides the basic statistics, AutoCal includes two event-based statistics:

Error of maximum value (*Error of max.*):

$$ErrMax = w_i(Max\{SIM_i\} - Max\{TARGET_i\}) \tag{1.5}$$

Error of minimum value (*Error of min.*):

$$ErrMin = w_i(Min\{SIM_i\} - Min\{TARGET_i\}) \tag{1.6}$$

The maximum and minimum target and simulated values are extracted in the period defined in the target file.

Weight below and weight above

The weight below is the weight assigned when the simulated value is below the target, and the weight above is the weight assigned when the simulated value is above the target. The weights should reflect the relative importance of positive and negative deviations from the target. In addition, the weights should specify the relative importance of the different output measures. For instance, in the model calibration, the assigned weights should reflect the measurement uncertainties, and the correlation between the measurements. That is, smaller weights should be given to more uncertain measurements, and if clusters of measurement points exist, these points should be given a lower weight than single point measurements in other parts of the modelling domain in order not to put undue emphasis on model performance in certain areas.

Function name

The function name of the objective function in which the output measure should be included. In the drop-down menu the function names from the *Objective functions* table are given.

1.4.2 Objective functions

Name

Name of the objective function.

Function type

AutoCal uses three different functions for aggregation of the defined output measures:

Weighted sum:

$$F_{pool,i} = \sum_{j=1}^{n} F_j \tag{1.7}$$

Weighted sum of absolute values:

$$F_{pool,i} = \sum_{j=1}^{n} |F_j| \tag{1.8}$$

Weighted sum of squares:

$$F_{pool,i} = \sum_{j=1}^{n} F_j^2 \tag{1.9}$$

where F_j is the output measure and n is the number of measures that are pooled.

Typically, output measures within a certain area that measure the same statistic for the same physical variable are pooled to evaluate the average model performance for that variable in the specified area with respect to bias (*Avg. Error*), dynamical behaviour (*St. Dev.*) or an overall goodness-of-fit (*RMSE*). The event-based statistics are typically pooled into an aggregate error of maximum and minimum values, respectively.

Weight

The weight assigned to the objective function in the aggregation of the different objective functions into one aggregate measure to be optimised. The assigned weights should reflect the relative priorities given to the different objectives, depending on the specific model application being considered. For investigating the entire Pareto front between the objective functions in a multi-objective optimisation, the aggregated measure can be adopted by performing several optimisation runs using different weights.

1.4.3 Evaluation period

Start date

The start date of the time series for which the output measures are calculated. It is generally recommended to set the start date after the start date of the model simulation in order to include a certain warm-up period in the simulation to minimise the influence from the initial conditions in the calculation of the output measures.

End date

The end date of the time series for which the output measures are calculated. This is usually set to the end date of the model simulation.

1.4.4 Aggregation of objective functions

The defined objective functions are aggregated into one measure:

$$F = \sum_{i=1}^{M} w_i g_i(F_{pool,\,i}) \tag{1.10}$$

where M is the number of objective functions that are aggregated, w_i, i = 1,2,..,M are the weights, and g_i(.), i = 1,2,..,M are transformation functions assigned to each objective function.

Three different transformation options are available:

No transformation:

$$g_i(F_{pool,\,i}) = F_{pool,\,i} \tag{1.11}$$

Transformation to a common distance scale:

$$g_i(F_{pool,\,i}) = \frac{F_{pool,\,i}}{\sigma_i} + \varepsilon_i \tag{1.12}$$

where σ_i is the standard deviation of the *i*'th objective function of the initial population used in the optimisation algorithm, and ε_i is a transformation constant given by:

$$\varepsilon_i = max\left\{min\left\{\frac{F_j}{\sigma_j}, j = 1, 2, \ldots, M\right\}\right\} - min\left\{\frac{F_i}{\sigma_i}\right\} \tag{1.13}$$

Transformation to a common probability scale:

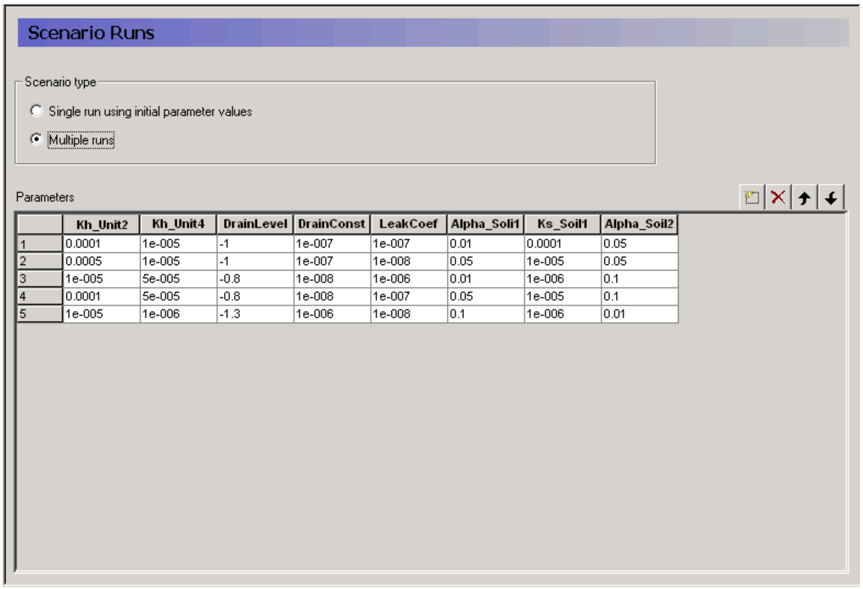
$$g_i(F_{pool,\,i}) = \Phi\left(\frac{F_{pool,\,i} - \mu_i}{\sigma_i}\right) \tag{1.14}$$

where $\Phi(.)$ is the cumulative distribution function of the standard normal distribution, and μ_i and σ_i are the mean and the standard deviation of the *i*'th objective function of the initial population.

The transformation functions that are applied in the transformation to a common distance scale and a common probability scale are introduced to compensate for differences in the magnitudes of the different measures so that all $g_i(.)$ have about the same influence on the aggregated objective function near the optimum. When using a population-based optimisation algorithm, such as the Shuffled Complex Evolution method and the Population Simplex Evolution, an initial population within the feasible region is evaluated. From this initial population, the transformation functions are calculated.

1.5 Scenario Runs

If the scenario run option has been chosen, the scenario run properties must be specified on the *Scenario Runs* page (Figure 1.4).



Scenario type

○ Single run using initial parameter values

● Multiple runs

Parameters

	Kh_Unit2	Kh_Unit4	DrainLevel	DrainConst	LeakCoef	Alpha_Soli1	Ks_Soil1	Alpha_Soil2
1	0.0001	1e-005	-1	1e-007	1e-007	0.01	0.0001	0.05
2	0.0005	1e-005	-1	1e-007	1e-008	0.05	1e-005	0.05
3	1e-005	5e-005	-0.8	1e-008	1e-006	0.01	1e-006	0.1
4	0.0001	5e-005	-0.8	1e-008	1e-007	0.05	1e-005	0.1
5	1e-005	1e-006	-1.3	1e-006	1e-008	0.1	1e-006	0.01

Figure 1.4 Scenario Runs page

1.5.1 Scenario type

AutoCal includes two different options for performing scenario analysis:

- Single run using initial parameter values. In this case a single model run is performed using the initial parameter values given in the table on the *Model Parameters* page. When the model parameters and the objective functions have been specified, it is recommended to carry out a single run in order to check the setup
- Multiple runs. In this case multiple model runs are performed using the parameter values given in the *Parameters* table.

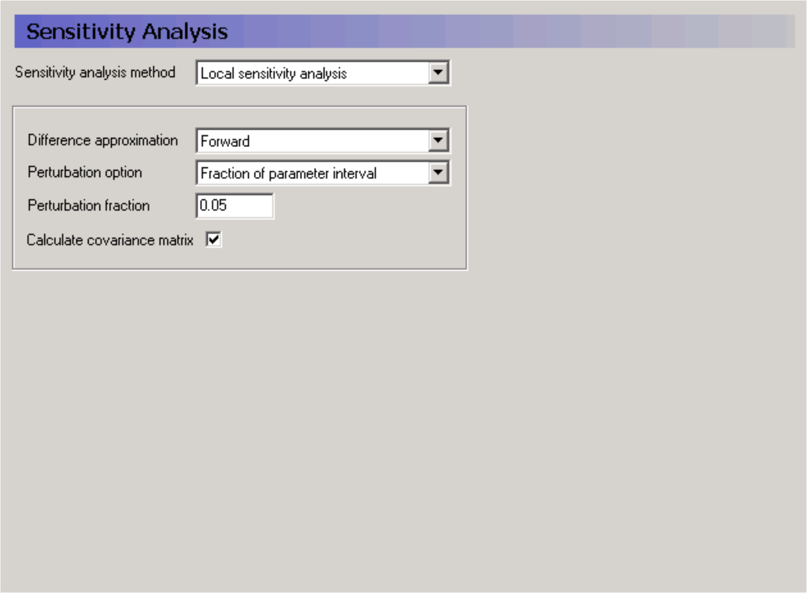
1.5.2 Parameters

In the *Parameters* table the set of variable parameters to be used in the scenario runs are specified.

1.6 Sensitivity Analysis

If the sensitivity analysis option has been chosen, the properties must be specified on the *Sensitivity Analysis* page (Figure 1.5).

Sensitivity analysis is often done as a first step to identify the most important model parameters to be fine-tuned in the succeeding parameter optimisation.



Sensitivity Analysis

Sensitivity analysis method: Local sensitivity analysis

Difference approximation: Forward

Perturbation option: Fraction of parameter interval

Perturbation fraction: 0.05

Calculate covariance matrix ☑

Figure 1.5 Sensitivity Analysis page

1.6.1 Sensitivity analysis method

The present version of AutoCal includes one sensitivity analysis method:

- Local sensitivity analysis

1.6.2 Local sensitivity analysis

Local sensitivity analysis provides the sensitivity of the model parameters around a specified parameter set, and hence gives information about the importance of the parameters only at that location in parameter space. If the simulation model is highly non-linear in its parameter-output interaction, sensitivity measures may vary considerably in the parameter space. Thus, parameters that are insensitive for certain parameter sets may be highly sensitive for other parameter sets and vice versa.

The local sensitivity measures are calculated around the initial parameter set specified on the *Model Parameters* page.

Difference approximation

The sensitivity of a parameter with respect to a model response (output measure) is defined as

$$S_i = \frac{\partial F}{\partial \theta_i} \tag{1.15}$$

where F is the output measure, and θ_i is the considered model parameter. The sensitivity measure is evaluated around a specified parameter set $(\theta_1,\theta_2,..,\theta_n)$.

In AutoCal a finite difference approximation is used to evaluate the sensitivity coefficients. Three different options are available:

Forward difference approximation:

$$S_i = \frac{F(\theta_1, \theta_2, \ldots, \theta_i + \Delta\theta_i, \ldots, \theta_n) - F(\theta_1, \theta_2, \ldots, \theta_n)}{\Delta\theta_i} \tag{1.16}$$

Backward difference approximation:

$$S_i = \frac{F(\theta_1, \theta_2, \ldots, \theta_n) - F(\theta_1, \theta_2, \ldots, \theta_i - \Delta\theta_i, \ldots, \theta_n)}{\Delta\theta_i} \tag{1.17}$$

Central difference approximation:

$$S_i = \frac{F(\theta_1, \theta_2, \ldots, \theta_i + \Delta\theta_i, \ldots, \theta_n) - F(\theta_1, \theta_2, \ldots, \theta_i - \Delta\theta_i, \ldots, \theta_n)}{2\Delta\theta_i} \tag{1.18}$$

where $\Delta\theta_i$ is the parameter perturbation.

The calculation of the sensitivity coefficients require n + 1 model evaluations in the case of forward and backward difference approximations, and 2n +1 model evaluations when the central difference approximation is applied.

Perturbation option

The parameter perturbation can be calculated as:

A fraction of the initial parameter value:

$$\Delta\theta_i = f_c\theta_i \tag{1.19}$$

A fraction of the parameter interval:

$$\Delta\theta_i = f_c(\theta_{i,\,upper} - \theta_{i,\,lower}) \tag{1.20}$$

where $\theta_{i,upper}$ and $\theta_{i,lower}$ are the specified upper and lower limits of the parameter.

Perturbation fraction

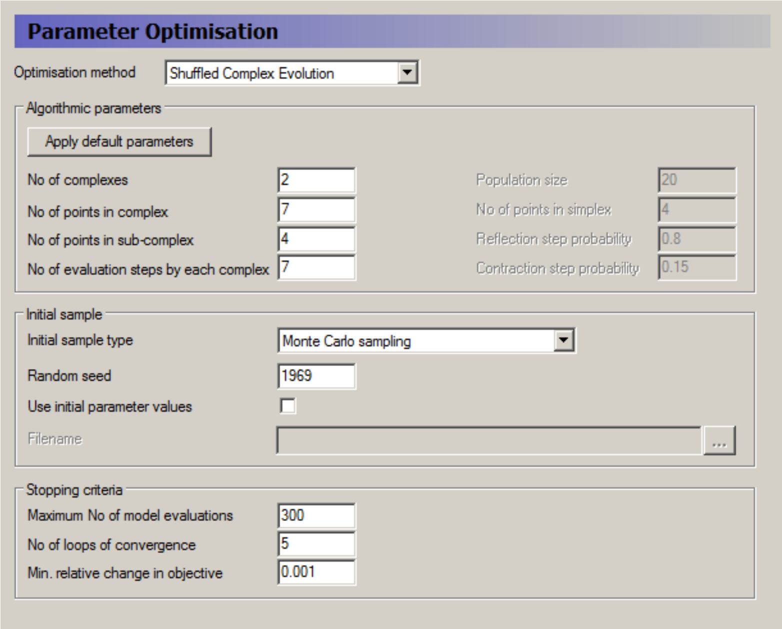
The perturbation fraction is the fraction f_c of the initial parameter value or the parameter interval depending on the choice of parameter perturbation.

Calculate covariance matrix

If this option is selected, the covariance matrix of the parameters evaluated around the initial parameter set is calculated. This matrix is derived based on the sensitivities of the simulated values corresponding to each of the target values with respect to each of the parameters. The matrix can only be calculated in the case a weighted least squares aggregated objective function is specified, i.e. each *Output measure* on the *Objective Functions* page is defined as a *RMSE* statistic, each *Objective function* is defined as a *Weighted sum of squares*, and the *Aggregation of objective functions* is set to *No transformation*.

1.7 Parameter Optimisation

If the parameter optimisation option has been chosen, the properties must be specified on the *Parameter Optimisation* page (Figure 1.6).



Parameter Optimisation

Optimisation method: Shuffled Complex Evolution

Algorithmic parameters

No of complexes	2	Population size	20
No of points in complex	7	No of points in simplex	4
No of points in sub-complex	4	Reflection step probability	0.8
No of evaluation steps by each complex	7	Contraction step probability	0.15

Initial sample

Initial sample type: Monte Carlo sampling

Random seed: 1969

Use initial parameter values: □

Filename:

Stopping criteria

Maximum No of model evaluations: 300

No of loops of convergence: 5

Min. relative change in objective: 0.001

Figure 1.6 Parameter Optimisation page with Shuffled Complex Evolution option chosen

1.7.1 Optimisation method

AutoCal includes two parameter optimisation methods:

- Shuffled Complex Evolution
- Population Simplex Evolution

1.7.2 Shuffled Complex Evolution method

The Shuffled Complex Evolution (SCE) method is a global optimisation algorithm that combines various search strategies, including (i) competitive evolution, (ii) controlled random search, (iii) the simplex method, and (iv) complex shuffling.

The SCE algorithm includes the following steps:

1. *Initialisation*. An initial sample of parameter sets θ_i are randomly generated from the feasible parameter space defined by the lower and upper limits of each parameter on the *Model Parameters* page. For each parameter set the objective function value $F_i = F(\theta_i)$ is calculated. The initial sample has the size $s = pm$ where p is the number of complexes and m is the number of points in each complex.
2. *Partitioning into complexes*. The s points are ranked in order of increasing objective function value ($F(1) < F(2) <... < F(s)$). The s points are partitioned into p complexes, such that points corresponding to function values {$F(1)$, $F(p+1)$, .., $F((s\text{-}1)p+1)$} form the 1st complex, points corresponding to function values {$F(2)$, $F(p+2)$, .., $F((s\text{-}1)p+2)$} form the 2nd complex, etc.
3. *Evolution*. A sub-complex of size q is formed from the complex by randomly choosing q points from the p points in the complex. A triangular probability distribution is used for assigning the probability of a point to be included in the sub-complex (i.e. larger probability for points with smaller objective function value). The sub-complex is evolved (offspring generation) according to the simplex algorithm. Each complex is evolved β times.
4. *Complex shuffling*. The new sample of s points is shuffled, cf. step 2.
5. Steps 2-4 are repeated until a stopping criterion is met.

Algorithmic parameters

The algorithmic parameters of the SCE algorithm, their feasible range and recommended values are shown in Table 1.2.

Table 1.2 Algorithmic parameters for the SCE algorithm (n = No. of optimisation parameters), their range and recommended values

Parameter	Description	Range	Recommended value
p	No. of complexes	$p \geq 1$	-
m	No. of points in a complex	$m \geq 2$	$2n + 1$

Table 1.2 Algorithmic parameters for the SCE algorithm (n = No. of optimisation parameters), their range and recommended values

Parameter	Description	Range	Recommended value
q	No. of points in a sub-complex	$2 \leq q \leq m$	$n + 1$
b	No. of evolution steps taken by each complex before shuffling	$\beta \geq 1$	$2n + 1$

If one complex is chosen in SCE and the number of points in the complex as well as the sub-complex are set equal to *n* + 1, the local search simplex method is obtained as a special case.

No. of complexes

Number of complexes *p* applied in the SCE algorithm. This is the most important parameter of the SCE algorithm. Sensitivity tests show that the dimensionality of the optimisation problem (No. of calibration parameters) is the primary factor determining the proper choice of *p*. In general, the larger value of *p* is chosen the higher the probability of converging into the global optimum but at the expense of a larger number of model simulations (the number of model simulations is virtually proportional to *p*), and vice versa. One should choose *p* to balance the trade-off between the robustness of the algorithm and the computing time.

No. of points in complex

Number of points in a complex. A recommended value for this parameter is 2*n* + 1 where *n* is the number of optimisation parameters.

No. of points in sub-complex

Number of points in a sub-complex. A recommended value for this parameter is n + 1 where *n* is the number of optimisation parameters.

No. of evaluation steps by each complex

Number of evaluation steps taken by each complex before complex shuffling. A recommended value for this parameter is 2*n* + 1 where *n* is the number of optimisation parameters.

Apply default parameters

If the *Apply default parameters* option is selected, the default algorithmic parameters in Table 1.2 are used.

Initial sample

Initial sample type

Three different options are available for generation of the initial sample in the SCE algorithm:

- Monte Carlo sampling. In this case the initial parameter sets are randomly generated within the feasible parameter range specified on the *Model Parameters* page assuming a uniform distribution.
- Latin hypercube sampling. In this case the individual parameters are sampled according to a stratified sampling scheme where the feasible parameter interval is divided into *s* equal intervals (*s* being the sample size) and a point is then randomly selected within each interval.
- Initial sample from previous optimisation run. This option allows continuing the optimisation from the last iteration loop of a previous optimisation run.

Random seed

Random seed used in the optimisation. Can be set to any positive integer value. Since the SCE method is a probabilistic search procedure, different optimisation results will be obtained by using different random seeds.

Use initial parameter values

Option to specify if the initial parameter values specified on the Model Parameters page should be included in the initial sample.

File name

File name of the file containing the optimisation results from a previous optimisation run to be used as initial conditions. The file to be specified is the AutoCal SCE optimisation output file (see Section 1.11.3).

Stopping criteria

Three stopping criteria are defined:

- Maximum number of model evaluations.
- Convergence in objective function space. In this case the optimisation terminates if the objective function of the best parameter set has not changed more than a user-defined minimum value in a given number of shuffling loops.
- Convergence in parameter space. In this case the optimisation terminates if the range of parameter values of the entire population in the parameter space is less than a given value (not user-defined).

The search terminates when one of these criteria is met.

Maximum No. of model evaluations

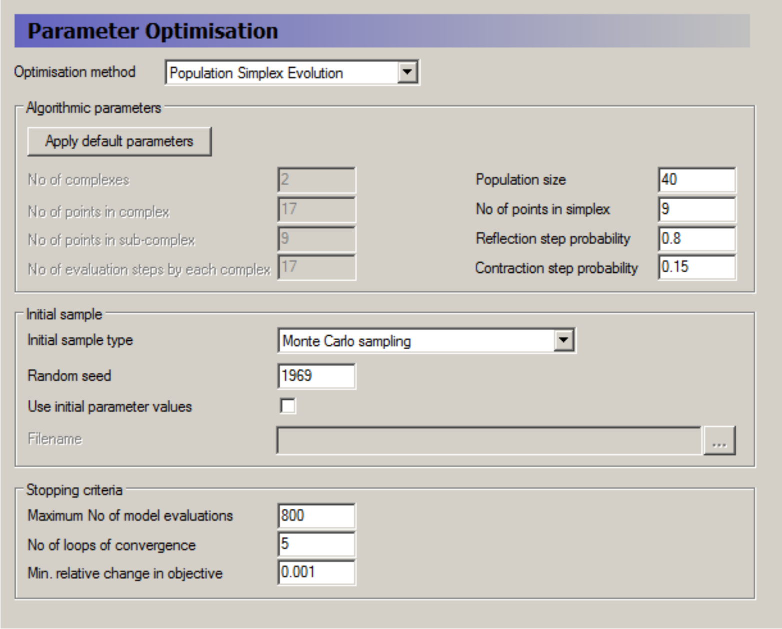
The maximum number of model evaluations allowed in the optimisation.

No. of loops of convergence

The number of iteration loops in which the objective function value of the best parameter set has not changed more than the *Minimum relative change in objective function value*.

Minimum relative change in objective function value

Minimum relative change allowed in the best objective function value in the last *No. of loops of convergence*.



Parameter Optimisation

Optimisation method: Population Simplex Evolution

Algorithmic parameters

No of complexes	2	Population size	40
No of points in complex	17	No of points in simplex	9
No of points in sub-complex	9	Reflection step probability	0.8
No of evaluation steps by each complex	17	Contraction step probability	0.15

Initial sample

Initial sample type: Monte Carlo sampling

Random seed: 1969

Use initial parameter values ☐

Filename:

Stopping criteria

Maximum No of model evaluations: 800

No of loops of convergence: 5

Min. relative change in objective: 0.001

Figure 1.7 Parameter Optimisation page with Population Simplex Evolution option chosen

1.7.3 Population Simplex Evolution method

The Population Simplex Evolution (PSE) method is a global optimisation algorithm that is especially suited for parallel execution using the simultaneous simulation option in AutoCal. The method evolves a population of points using the reflection and contraction operators included in the simplex method. In addition a mutation component is added to minimise the risk of premature convergence.

The PSE algorithm can be summarised as follows:

1. *Initialisation*. An initial sample of size *s* of parameter sets θ_i are randomly generated from the feasible parameter space defined by the lower and upper limits of each parameter on the *Model Parameters* page. For each parameter set the objective function value $F_i = F(\theta_i)$ is calculated.
2. *Evolution*. From the population *s* simplexes of size *q* are formed. A triangular probability distribution is used for assigning the probability of a point to be included in the simplex (i.e. larger probability for points with smaller objective function value). The simplex is evolved according to the simplex operators and a mutation operator. Thus, a simplex can be

evolved by reflection (with probability p_r), contraction (with probability p_c) or mutation (with probability 1-p_c-p_r). For each simplex a random value ε between 0 and 1 is generated and this value then determines which operator should be applied for the evolution, i.e. reflection if $\varepsilon < p_r$, contraction if ε is in the interval [p_r, $p_r + p_c$], and mutation if $\varepsilon > p_c + p_r$.

3. *Selection of new population*. The algorithm applies the principle of elitism to select the new population. The old population and the evolved population of points are sorted with respect to their objective function value. The best (i.e. smallest objective function value) *s* points defines the new population.
4. Steps 2-3 are repeated until a stopping criterion is met.

Algorithmic parameters

The algorithmic parameters of the PSE algorithm, their feasible range and recommended values are shown in Table 1.3.

Table 1.3 Algorithmic parameters for the PSE algorithm (n = No. of optimisation parameters), their range and recommended values

Parameter	Description	Range	Recommended value
s	Population size	$s > n$	-
q	No. of points in a simplex	$2 \leq q < s$	n + 1
p_r	Reflection step probability	$0 \leq p_r \leq 1$	0.70-0.90
p_c	Contraction step probability	$0 \leq p_c \leq 1$	0.05-0.20
p_m	Mutation probability determined as p_m = 1-p_r-p_c	$0 \leq p_m \leq 1$	0.01-0.10

Population size

Population size *s* applied in the PSE algorithm. This parameter is important for the convergence properties of the algorithm. In general, the larger value of *s* is chosen the higher the probability of converging into the global optimum but at the expense of a larger number of required model evaluations. One should choose *s* to balance the robustness of the algorithm and the computing time. The proper choice of *s* depends on the dimensionality of the problem.

No. of points in simplex

Number of points in a simplex. A recommended value for this parameter is *n* + 1 where *n* is the number of optimisation parameters.

Reflection step probability

The probability of performing a reflection step of the simplex. In this case the new point is found by reflecting the worst point of the simplex in the centroid of the remaining points. A recommended value for this parameter is in the range 0.70-0.90.

Contraction step probability

The probability of performing a contraction step of the simplex. In this case the new point is found as the mean between the worst point of the simplex and the centroid of the renaming points. A recommended value for this parameter is in the range 0.05-0.20. The sum of the reflection and contraction probabilities should be less than one. The remaining portion ($1-p_r-p_c$) is assigned a mutation probability with a recommended value in the range 0.01-0.10.

Initial sample

The same options as used for the SCE algorithm are applied.

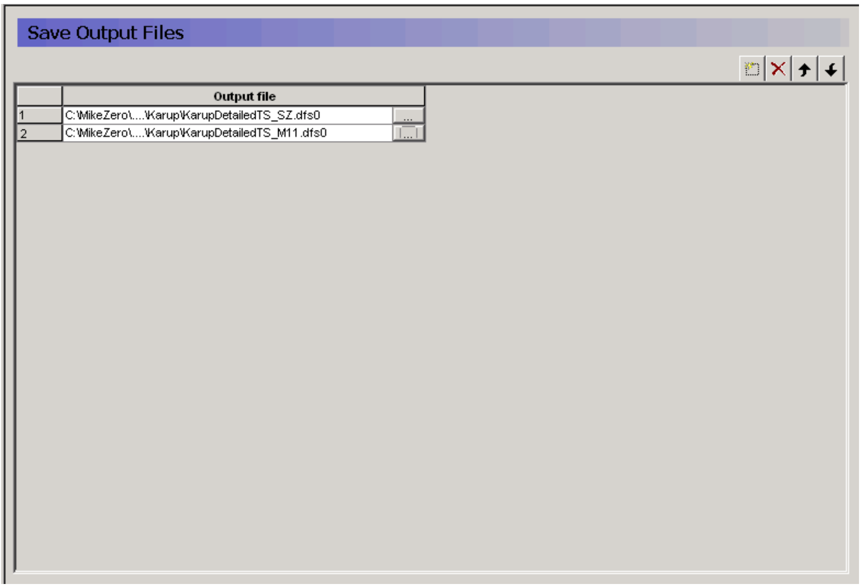
Stopping criteria

The same stopping criteria as used for the SCE algorithm are applied.

1.8 Save Output Files

AutoCal allows output files from the individual model simulation runs to be saved. The names of the files to be saved are specified in the *Output files* table (Figure 1.8). The saved output files from the individual simulations are given an extension in the file name that contains the simulation number. For instance, if the simulation model produces a result file called MyResults.dfs2, the saved files are named MyResults_1.dfs2, MyResults_2.dfs2, etc.

The files are saved in the individual model sub-folders: _Process1, _Process2, etc.



Save Output Files

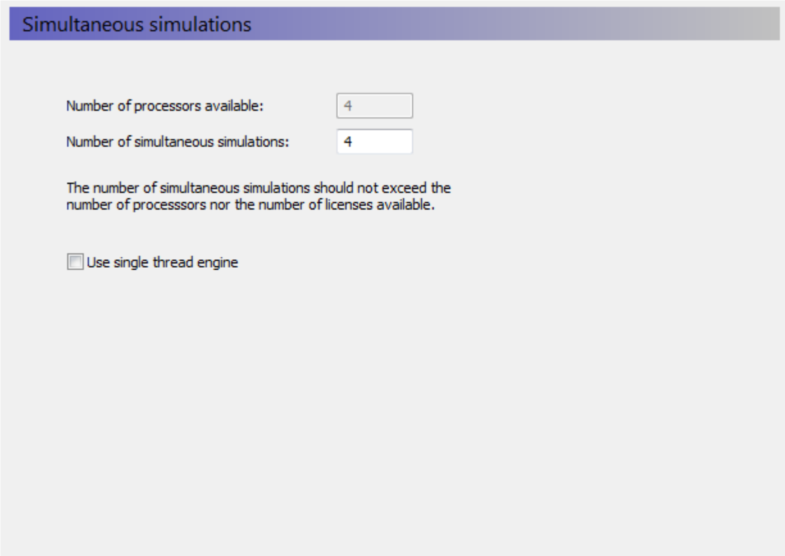
	Output file
1	C:\MikeZero\....\Karup\KarupDetailedTS_SZ.dfs0
2	C:\MikeZero\....\Karup\KarupDetailedTS_M11.dfs0

Figure 1.8 Save Output Files page

1.9 Simultaneous simulations

AutoCal gives the user the power to utilize multi core PC's for carrying out multiple simulations simultaneously. Hereby an ensemble of simulations may be completed faster than the standard sequential method. Thus, the multi core functionality reduces the overall simulation time when doing scenario runs or optimisation runs.

The simultaneous simulations may be used with all the simulation options. In the case of *Scenario runs* or *Sensitivity analysis* all the parameter sets to be evaluated by the model are submitted for execution at the beginning of the simulations. When all parameter sets are evaluated, the AutoCal simulation is finished. In the case of *Parameter optimisation* only the *Population Simplex Evolution* algorithm can be applied for simultaneous execution. For each iteration step in the algorithm the population of parameter sets to be evaluated are submitted for execution. When all the parameter sets of the population are evaluated on the individual processes, the main job processes the results and determines a new population of parameter sets to be evaluated, which are then submitted to the job queue for execution. The AutoCal simulation is finished when one of the stopping criteria of the algorithm has been reached.



Simultaneous simulations

Number of processors available: 4

Number of simultaneous simulations: 4

The number of simultaneous simulations should not exceed the number of processsors nor the number of licenses available.

□ Use single thread engine

Figure 1.9 Simultaneous simulations page

Number of processes available

Informs the user of the number of processors available

Number of simultaneous simulations

The user selected number of simultaneous simulations to be carried out

Use single thread engine

By default auto calibration will only use one core per individual simulation i.e. the individual runs in the ensemble will not distribute the load on multiple cores. The latter is only of importance if the engine being launched can utilize multiple processors.

Selecting the number of simultaneous simulations

The book keeping of which cores are used when is handled by the auto calibration tool. All that is required on the user side is to decide how many of the cores on the PC should be used. This number should be chosen based on

1. The number of cores available on the PC. There is no gain in launching more simultaneous simulations than the number of cores available on the PC. The interface informs the user of the cores available.
2. The number of simulations that the license grants you access to. This number may vary depending on the contents of your license file. If you are uncertain of the number of simulations you can carry out, the information may be retrieved through the license administration interface, The latter is located under your MIKE Powered by DHI installation short cut in the start menu. Look for the model in question and find the number in the Max User column.

The number should be selected as the smallest of the two numbers.

1.10 Start AutoCal Simulation

To start an AutoCal simulation choose:

Run | Simulation

See also Figure 1.10.

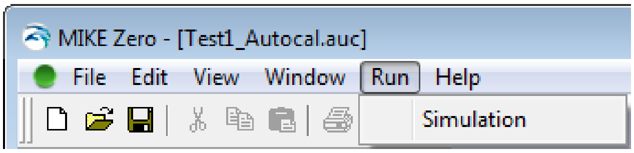


Figure 1.10 Start AutoCal simulation

1.11 Output

1.11.1 General output files

AutoCal produces two general output files:

- A log file that contains summary information about the AutoCal simulation. The log file is named MyAutocalRun.log where MyAutocalRun is the user-specified name for the AutoCal .auc input file. The file is saved in the working directory.
- A file that contains results from the individual model simulations, including the evaluated parameter set and the calculated output measures and objective functions. The file is named MyAutocalRun.dat and is saved in the working directory. The result file is tab delimited and can be directly imported into a spreadsheet.

1.11.2 Sensitivity analysis output

If the sensitivity analysis option is chosen, AutoCal produces a result file that contains the calculated sensitivity coefficients of each parameter with respect to the different output measures and objective functions. The file is named MyAutocalRun_Sensitivity.dat and is saved in the working directory. The result file is tab delimited and can be directly imported into a spreadsheet.

Local sensitivity analysis

In the case of local sensitivity analysis the result file contains the following:

- The parameter set and corresponding output measures and objective functions for the point where the local sensitivity analysis has been performed.
- Scaled sensitivity coefficients for each parameter of the specified output measures and objective functions.
- Parameter covariance matrix in terms of the standard deviations of the parameters and the correlation matrix. It should be noted that the calculated parameter covariance is related to the transformed parameter values and not their native values.

In order to compare the local sensitivity coefficients between parameters of different scales of magnitude scaled sensitivity values are calculated as

$$S_{i,\,scale} = S_i(\theta_{i,\,upper} - \theta_{i,\,lower}) \tag{1.21}$$

where S_i is the calculated un-scaled sensitivity coefficient described in Section 1.6.2, and $\theta_{i,upper}$ and $\theta_{i,lower}$ are the specified upper and lower limits of the parameter.The scaled sensitivities provide a ranking of the parameters with respect to the importance of the parameters for the considered output measure or objective function. Higher scaled sensitivity values (absolute values) indicate more sensitive parameters.

As a rule of thumb, parameters are said to be insensitive if their scaled sensitivity value is less than about 0.01-0.02 times the maximum scaled sensitivity value (absolute value). However, care should be taken using this as a strict threshold measure. Since the sensitivity coefficients are evaluated only around the initial parameter set, they reflect the local sensitivities only. At other locations in the parameter space the sensitivity coefficients may be very different, especially if the simulation model is highly non-linear in its parameter-output interaction.

1.11.3 Optimisation output

If the parameter optimisation option is chosen, AutoCal produces a result file that contains the results of the optimisation. The file is named MyAutocalRun_Optimisation.dat and is saved in the working directory. The result file is tab delimited and can be directly imported into a spreadsheet.

In the log file a summary of the optimisation results is written, including the best parameter set and corresponding objective function value for each iteration loop.

When the stopping criterion of the optimisation run is met, AutoCal performs a final model run using the optimised parameters. The result of this run is written in the last line of the MyAutocalRun.dat file. The model parameter files contain the optimised parameter values.

The results using the optimised parameter set is performed in model subfolder _Process1.

Shuffled Complex Evolution and Population Simplex Evolution

For each iteration loop performed in the SCE or PSE optimisation the result file MyAutocalRun_Optimisation.dat contains the population of parameter sets and corresponding objective functions sorted with respect to the aggregated objective function value. This file should be used as the initial sample file for a subsequent SCE or PSE optimisation that continues the optimisation.

1.12 Error handling

When AutoCal performs a model simulation, the return code from the called program is checked. In the case an error in the model simulation has occurred for a given parameter set, e.g. by creating instabilities, an error return code is received by AutoCal (a return value different from 0). In such a case the output measures and objective functions are given a delete value of -99, which are written to the output files.

Depending on the simulation option chosen in AutoCal, error return codes from the called program are handled differently:

- In the case the *Scenario runs* option is chosen AutoCal continues with the next scenario.
- In the case the *Sensitivity analysis* options is chosen AutoCal is halted. In this case the sensitivity coefficients and the parameter covariance matrix cannot be calculated.
- In the case the *Parameter optimisation* option is chosen, the optimisation continues and the parameter set that gave an error in the simulation model is penalised in the optimisation by assigning the aggregated objective function a penalty value of 10^{30}. However, if a large number of model runs return error codes, a general model failure is suspected and AutoCal is halted.

If the simulation model does not provide a return code, AutoCal offers another possibility for error handling. When a model run is started, AutoCal creates an empty file called AutocalSimErr.dat that is placed in the process directory. When the model run finishes, AutoCal checks the file, and if it is not empty it is assumed that the model simulation has failed. In this case AutoCal handles the simulation in the same way as when it receives an error return code as described above.

To use this facility, it is necessary to include as part of the model simulation an option that writes an error message (or any dummy string) to the AutocalSimErr.dat file if an error occurs during the model simulation. If an error message is written to the file, the file can be saved as part of the AutoCal run for later inspection by using the *Save Output Files* option.

2 Examples

The examples contained herein have been sorted according to the model applied. The first example is fairly simple and the following examples have increasing complexity.

Recommendation

In general, when you start setting up an Autocal setup, you should begin with 1 parameter only and then test it before inserting several parameters. Add only 2-3 parameters in one go and do test runs again (ideally you always add only 1 new parameter). During tests of parameters you can select to use only part of the full simulation period.

Note: The AutoCal tool cannot be executed from within a MIKE Zero project. The .mzp project files in the examples are created for overview only. To execute an .auc setup open the file in a MIKE Zero shell without a project.

2.1 MIKE 21

2.1.1 Example 1

In this example AutoCal performs Parameter Optimisation.

The setup used in the example is a simple MIKE 21 HD FM model describing the increase of salinity in a large shallow reclamation basin that is suddenly subjected to a tidal flush due to a local failure in the bank next to the sea with a tidal water level variation. See the setup for details.

The initial salinity in the basin is low, but after 24 hours the salinity in the corner is supposedly measured to 15 PSU, see Figure 2.1 for location. The objective of this example is to calibrate the dispersion parameter such that the model setup can predict the increase of salinity correctly.

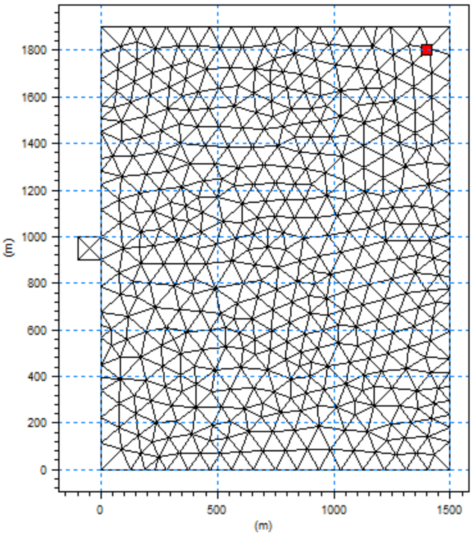


Figure 2.1 Bathymetry of reclamation basin, measuring point is indicated

The example is included as Test1_Autocal.auc. Two subfolders, called "Setup" and "Measurements", are present. The first one contains all the relevant files for the auto-calibration simulation while the second one includes all the measurements/observed data (for Example 1 only one Time Series Dfs0 file is present).

Simulation specifications

Under Simulation specifications, model simulation sequence, model parameter file(s) and simulation option must be specified.

In Example 1 the MIKE 21 setup (Test1.m21fm) is selected for both model simulation sequence and model parameter file because besides producing results it also contains the parameter chosen for optimization. Once Test1.m21fm has been specified as model parameter file, AutoCal will automatically produce a template MIKE 21 FM file called Test1_Autocal.m21fm. Test1_Autocal.m21fm is a copy of Test1.m21fm which is then edited to mark the dispersion coefficient for optimization. The horizontal dispersion coefficient is given the value 1.01e-035.

‘Parameter Optimization’ is chosen as simulation option.

Model Parameters

The model parameters table is partly filled out automatically. AutoCal lists all the parameters recognised for optimisation and allows the user to specify Name, Parameter type, Initial Value, Lower bound, Upper bound and Transformation. The parameter is given the Name DispersionC, set the Parameter type to Variable, and let the horizontal dispersion coefficient vary between

0.05 [m^2/s] and 2.0 [m^2/s] with an initial value of 1.0 [m^2/s]. Finally the transformation is set to real.

In Keyword/Line no, which is filled out automatically, you can see where the parameter being optimized is located.

Objective Functions

The objective function specification is used as a measure for quality of the calibration. Example 1 is only a simple illustrative example aiming to illustrate how to use the auto-calibration tool. Measurements in this case are not real but an assumed value. AutoCal evaluates objective functions by comparing dfs0 files therefore real observations should normally be used as target files.

AutoCal minimises an aggregation of one or more objective functions. Each objective function may consist of several so-called output measures. For the present example one objective function ('OF Salinity') is defined. The function is specified as weighted sum of absolute power and it is given a weight value of 1.

'Salinity' is specified as output measure. The measure is specified as the Avg. Error of the difference between the simulated salinity and the measured salinity in the same point within the MIKE 21 domain. By specifying the weight as 1 and the function name as 'OF Salinity' the measure will contribute to the objective function called 'OF Salinity' with a weight of 1.

No aggregation of objective functions is applied.

Parameter Optimisation

Population Simplex Evolution is chosen as optimization method.

The parameters for optimisation can either be left as default values or user-specified. In this case Population size is set to 8, No of points in simplex is set to 2, maximum No of model evaluations is reduced to 80 (default is 500), number of loops of convergence is set to 5 and minimum relative change is set to 0.001.

Normally ending the optimisation by meeting the convergence criteria should be preferred. Convergence criteria are a measure of the relative change in best objective function value in the last number of loops of convergence where the number of loops of convergence is also user-specified.

Save Output Files

AutoCal allows output files from the individual model simulation runs to be saved. The names of the files to be saved are specified in the Output files table. The saved output files from the individual simulation are given an extension in the file name that contains the simulation number. For instance, if the simulation model produces a result file called MyResults.dfs2, the saved files are named MyResults_1.dfs2, MyResults_2.dfs2, etc.

In the current example no Output Files are specified.

Simultaneous Simulations

The number of simultaneous simulations affect the speed of the simulation but not the overall result. You may edit the values according to your hardware setup.

Results

The optimal dispersion coefficient is found to be 1.34. Applying this value in the MIKE 21 model setup will give a salinity of 14.985 PSU at the measuring point which is close to the ‘measured’ value of 15 PSU.

2.2 MIKE SHE

2.2.1 Example 1

Example 1 is the simplest of the examples included in the AutoCal-MIKE She example collection.

The setup used in the example is a simple MIKE SHE model including only the Saturated Zone (SZ), which is composed of one layer (aquifer). Constant uniform precipitation is applied and rainfall completely infiltrates. See the setup for details.

The objective of the example is to calibrate the model with respect to the aquifer horizontal hydraulic conductivity, given a measured SZ head elevation in a random point of the domain.

The example is included as Example1.auc. Two subfolders, called "Setup" and "Measurements", are present. The first one contains all the relevant files for the auto-calibration simulation while the second one includes all the measurements/observed data (for Example 1 only one Time Series Dfs0 file is present).

Simulation specifications

Under Simulation specifications, model simulation sequence, model parameter file(s) and simulation option must be specified.

In Example 1 the MIKE SHE setup (Example1.she) is selected for both model simulation sequence and model parameter file because besides producing results it also contains the parameter chosen for optimization. Once Example1.she has been specified as model parameter file, AutoCal will automatically produce a template MIKE SHE file called Example1_Autocal.she.

Example1_Autocal.she is a copy of Example1.she which is then edited to mark the aquifer horizontal hydraulic conductivity for optimization. The horizontal hydraulic conductivity is given the value 1.01e-035.

Parameter Optimization is chosen as simulation option.

Model Parameters

The model parameters table is partly filled out automatically. AutoCal lists all the parameters recognised for optimisation and allows the user to specify Name, Parameter type, Initial Value, Lower bound, Upper bound and Transformation. The parameter is given the Name HHydrCond, set the Parameter type to Variable, and let the horizontal hydraulic conductivity vary between 0.0001 [m/s] and 0.005 [m/s] with an initial value of 0.000222 [m/s]. Finally the transformation is set to logarithmic.

In Keyword/Line no, which is filled out automatically, you can see where the parameter being optimized is located.

Objective Functions

The objective function specification is used as a measure of comparison of measured results with observations for the calibration. Example 1 is only a simple illustrative example aiming to illustrate how to use the auto-calibration tool. Measurements in this case are not real but consist of output results of a previous MIKE SHE simulation. AutoCal evaluates objective functions by comparing dfs0 files therefore real observations should normally be used as target files.

AutoCal minimises an aggregation of one or more objective functions. Each objective function may consist of several so-called output measures. For the present example one objective function (MyFunc) is defined. MyFunc is specified as weighted sum and it is given a weight value of 1.

MyOutPut is specified as output measure. The measure is specified as the RMSE of the difference between the simulated SZ head elevation and the measured SZ head elevation in the same point within the model domain. By specifying the weight as 1 and the function name as MyFunc the measure will contribute to the objective function called MyFunc with a weight of 1.

No aggregation of objective functions is applied since only one objective function is defined.

Parameter Optimisation

Population Simplex Evolution is chosen as optimization method.

The parameters for optimisation can either be left as default values or user-specified. In this case Population size is set to 10, No of points in simplex is set to 2 and maximum No of model evaluations is reduced to 50 (default is 500).

Normally ending the optimisation by meeting the convergence criteria should be preferred. Convergence criteria are a measure of the relative change in best objective function value in the last number of loops of the optimisation where the number of loops of convergence is also user-specified.

Save Output Files

AutoCal allows output files from the individual model simulation runs to be saved. The names of the files to be saved are specified in the Output files table. The saved output files from the individual simulations are given an extension in the file name that contains the simulation number. For instance, if the simulation model produces a result file called MyResults.dfs2, the saved files are named MyResults_1.dfs2, MyResults_2.dfs2, etc.

In the current example no Output Files are specified.

Simultaneous simulations

Multi-core PC's can be effectively utilised by distributing the individual model runs. The number of simultaneous simulations is set according to the hardware configuration and number of available licenses.

Results

By using 50 model evaluations AutoCal finds a horizontal hydraulic conductivity of 0.00199 [m/s].

Results from the optimisation can be found in the files Example1.dat and Example1_Optimisation.dat. Example1.dat shows the evaluated parameter (HHydrCond), output measure (MyOutput) and objective function (myFunc) for the different model runs in the optimisation. The last line is the optimum solution. Example_Optimisation.dat shows the population of parameter sets and corresponding objective functions for the different loops of the optimisation sorted with respect to the aggregated objective function (in this case myFunc).

2.2.2 Example 2

The setup used in Example 2 is a relatively simple MIKE SHE model including only the Saturated Zone (SZ), which is composed of three layers. Constant uniform precipitation is applied, rainfall completely infiltrates and two pumping wells are included in the SZ. See the setup for details.

The objective of the example is to calibrate the model with respect to horizontal and vertical hydraulic conductivities of layer 1, horizontal hydraulic conductivity of layer 2 and rainfall (recharge), given measured SZ head elevations in correspondence of the two pumping wells.

The example is included as Example2.auc. Two subfolders, called "Setup" and "Measurements", are present. The first one contains all the relevant files for the auto-calibration simulation while the second one includes all the measurements/observed data.

Simulation specifications

Under simulation specifications, model simulation sequence, model parameter file(s) and simulation option must be specified.

In Example 2 the MIKE SHE setup (Example2.she) is selected for both model simulation sequence and model parameter file because besides producing results it also contains the parameters chosen for optimization. Once Example2.she has been specified as model parameter file, AutoCal will automatically produce a template she file called Example2_Autocal.she. Example2_Autocal.she is a copy of Example2.she which is then edited to mark the parameters for optimization. Rainfall is given the value 1.01e-035, horizontal hydraulic conductivity of layer 1 is given the value 1.02e-035, vertical hydraulic conductivity of layer 1 is given the value 1.03e-035 and horizontal hydraulic conductivity of layer 2 is given the value 1.04e-035

Parameter Optimization is chosen as simulation option.

Model Parameters

The model parameters table is partly filled out automatically. AutoCal lists all the parameters recognised for optimisation and allows the user to specify Name, Parameter type, Initial Value, Lower bound, Upper bound and Transformation. The four parameters are given the names Recharge, HHydrCond_layer1, VHydrCond_layer1 and HHydrCond_layer2 respectively.

For Recharge the Parameter type is set to Variable. Its value can vary between 5 [mm] and 12 [mm] with an initial value of 8 [mm]. The transformation is set to real.

For HHydrCond_layer1 the Parameter type is set to Variable. Its value can vary between 0.001 [m/s] and 0.006 [m/s] with an initial value of 0.0022 [m/s]. The transformation is set to logarithmic.

For VHydrCond_layer1 the Parameter type is set to Dependent. In fact, in the current example, the vertical hydraulic conductivity relates to the horizontal hydraulic conductivity according to the following equation 0.1*HHydrCond_layer1. Neither Upper/Lower bound nor initial value need to be specified. The transformation is set to logarithmic.

For HHydrCond_layer2 the Parameter type is set to Variable. Its value can vary between 0.0005 [m/s] and 0.004 [m/s] with an initial value of 0.0015 [m/s]. The transformation is set to logarithmic.

In Keyword/Line no, which is filled out automatically, you can see where the parameter being optimized is located.

Objective Functions

The objective function specification is used as a measure for quality of the calibration. Measurements are not real but consist of output results of a previous MIKE SHE simulation. AutoCal evaluates objective functions by comparing dfs0 files therefore real observations should normally be used as target files.

AutoCal minimises an aggregation of one or more objective functions. Each objective function may consist of several so-called output measures. For the present example one objective function (MyFunc) is defined. MyFunc is specified as weighted sum and it is given a weight value of 1.

Well_01 and Well_02 are specified as output measures. Both measures are specified as the Avg. Error of the difference between the simulated SZ head elevation and the measured SZ head elevation in correspondence of the two pumping wells. By specifying the weight as 1 and the function name as MyFunc the measure will contribute to the objective function called MyFunc with a weight of 1.

No aggregation of objective functions is applied.

Parameter Optimisation

Population Simplex Evolution is chosen as optimization method.

The parameters for optimisation can either be left as default values or user-specified. In this case Population size is set to 16, No of points in simplex is set to 4 and maximum No of model evaluations is reduced to 160 (default is 500).

Normally ending the optimisation by meeting the convergence criteria should be preferred. Convergence criteria are a measure of the relative change in best objective function value in the last number of loops of convergence where the number of loops of convergence is also user-specified.

Save Output Files

AutoCal allows output files from the individual model simulation runs to be saved. The names of the files to be saved are specified in the Output files table. The saved output files from the individual simulations are given an extension in the file name that contains the simulation number. For instance, if the simulation model produces a result file called MyResults.dfs2, the saved files are named MyResults_1.dfs2, MyResults_2.dfs2, etc.

In the current example no Output Files are specified.

Simultaneous simulations

Multi-core PC's can be effectively utilised by distributing the individual model runs. The number of simultaneous simulations is set according to the hardware configuration and number of available licenses.

Results

After 160 evaluations the objective function convergence criteria are met. AutoCal finds the following results:

Recharge = 10 [mm]

HHydrCond_layer1 = 0.00199 [m/s]

HHydrCond_layer2 = 0.00145 [m/s]

VHydrCond_layer1 does not appear in the results because it has been set to "Dependent" on HHydrCond_layer1.

Results from the optimisation can be found in the files Example2.dat and Example2_Optimisation.dat. Example2.dat shows the evaluated parameter (HHydrCond), output measure (Well) and objective function (myFunc) for the different model runs in the optimisation. The last line is the optimum solution. Example_Optimisation.dat shows the population of parameter sets and corresponding objective functions for the different loops of the optimisation sorted with respect to the aggregated objective function (in this case myFunc).

INDEX

A
Aggregate measure 19

C
Comparison statistics 16
Complexes 25
Constant parameter 13

D
Dependent parameter 13

E
Equation parser 14
Evolution 25

F
Finite difference approximation 23

I
Identification tags 11

L
Latin hypercube sampling 27
Log file 33
Logarithmic transformation 14

M
Measurement uncertainties 18
MIKE 11 result file 11
MIKE Zero Toolbox 10
Monte Carlo sampling 27
Multi-objective calibration 19

N
Number of complexes 26

P
Parameter Optimisation 37
Parameter perturbation 23
Pareto front 19
Post-processing of model results 10
PSE algorithm 28

S
Scaled sensitivity values 34
SCE algorithm 25
Sensitivity measures 22
Stopping criteria 27

T
Template file 11
Transformation functions 19

V
Variable parameter 13

W
Weighted sum 18
Weighted sum of absolute values 18
Weighted sum of squares 18