Geosci. Model Dev. Discuss., 7, 6439–6487, 2014
www.geosci-model-dev-discuss.net/7/6439/2014/
doi:10.5194/gmdd-7-6439-2014

Geoscientific Model Development Discussions
Open Access

Optimization of experimental designs and model parameters exemplified by sedimentation in salt marshes

J. Reimer[1], **M. Schürch**[2], **and T. Slawig**[1]

[1]Institute of Computer Science, Future Ocean – Kiel Marine Sciences, Christian-Albrechts-University Kiel, 24098 Kiel, Germany
[2]Institute of Geography, Future Ocean – Kiel Marine Sciences, Christian-Albrechts-University Kiel, 24098 Kiel, Germany

Received: 6 August 2014 – Accepted: 3 September 2014 – Published: 26 September 2014

Correspondence to: J. Reimer (jor@informatik.uni-kiel.de)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

The weighted least squares estimator for model parameters was presented together with its asymptotic properties. A popular approach to optimize experimental designs called local optimal experimental designs was described together with a lesser known approach which takes into account a potential nonlinearity of the model parameters. These two approaches were combined with two different methods to solve their underlying discrete optimization problem.

All presented methods were implemented in an open source MATLAB toolbox called the *Optimal Experimental Design Toolbox* whose structure and handling was described.

In numerical experiments, the model parameters and experimental design were optimized using this toolbox. Two models for sediment concentration in seawater of different complexity served as application example. The advantages and disadvantages of the different approaches were compared, and an evaluation of the approaches was performed.

1 Introduction

Mathematical models are a fundamental concept in science. Often, they contain only roughly known model parameters. A common way to make such models more realistic is to optimize these parameters so that the model output is more consistent with measurement results.

The measurements required for this purpose are often time-consuming or costly. For this reason, it is desirable that the information content of the obtained measurement results is maximal.

Several conditions under which measurements are carried out are controllable. These conditions are also known as experimental setup or experimental design. This can be, e.g., the point in time, the location or the method of the measurements. These

GMDD
7, 6439–6487, 2014

experimental designs can be optimized so that the information content is maximized. Thus, the number of measurements necessary for a certain accuracy of the model parameters and accordingly of the model itself can be considerably reduced.

The main problem in optimizing experimental design is to quantify the information content. In general, this can only be done approximatively. There are several approaches to quantify the information content and hence to optimize experimental designs. See, e.g., Pronzato and Pázman (2013) for an overview. Usually, these approaches are a tradeoff between accuracy and computational effort. In general, it is difficult to say whether a higher computational effort is justified by a higher accuracy.

In this paper, two models for sediment concentration in seawater served as application examples. Their model parameters had to be adapted to the local environmental conditions. The measurements required for this purpose are very time-consuming. For this reason, it should be evaluated which approach is most suitable to optimize their experimental designs.

After this introduction, four different approaches to optimize experimental designs together with the weighted least squares estimator for model parameters are presented in Sect. 2. One approach is based on the linearization of the model with respect to the parameters and is the most common used approach called local optimal experimental design. The second more robust approach takes into account a potential nonlinearity of the model parameters. Both approaches are combined with two different approaches of solving the underlying discrete optimization problem.

The presented methods to optimize experimental designs and model parameters were implemented in an open source MATLAB toolbox called the *Optimal Experimental Design Toolbox*. The structure and handling of this toolbox is described in Sect. 3.

The numerical experiments carried out with the models for sediment concentration and their results are shown in Sect. 4.

GMDD
7, 6439–6487, 2014

2 Optimization of model parameters and experimental designs

The first step to the optimization of model parameters is the choice of the estimator. This maps the measurement results onto optimal model parameters. These optimal parameters are often defined so that they minimize a so-called misfit function. The misfit function quantifies the distance between the measurement results and the model output.

The most widely used class of estimators are the least squares estimators since their introduction by Gauss and Legendre (see, e.g., Stigler, 1981). Their simplest form is the ordinary least squares estimator. Its misfit function is the sum of the squares of the differences between each measurement result and the corresponding model output. A generalization is the weighted least squares estimator which has advantages in case of heteroscedastic measurement errors. This estimator and its asymptotic properties are presented in the following subsection. The generalized least squares estimator is a further generalization which takes into account a stochastic dependence of the measurement errors.

2.1 The weighted least squares estimator

In the following, the weighted least squares estimator is presented. For this purpose, some notations and assumptions are introduced.

The model function is denoted by

$$f : \Omega_x \times \Omega_p \longrightarrow \mathbb{R}.$$

Here, $\Omega_x \subseteq \mathbb{R}^{n_x}$ is the set of feasible experimental designs and $\Omega_p \subset \mathbb{R}^{n_p}$ the set of feasible model parameters from which the unknown exact parameter vector $\hat{\boldsymbol{p}} \in \Omega_p$ is to be determined. Often, these sets are defined by lower and upper bounds.

The measurement result for every design $x \in \Omega_x$ is considered as a realization of a random variable η_x. Each random variable η_x is assumed to be normally distributed

GMDD
7, 6439–6487, 2014

with expectation $f(x, \hat{\boldsymbol{p}})$ and standard deviation $\sigma_x > 0$, i.e.,

$$\eta_x \sim \mathcal{N}(f(x, \hat{\boldsymbol{p}}), \sigma_x^2) \text{ for every } x \in \Omega_x. \tag{1}$$

Furthermore, these random variables are assumed to be pairwise stochastically independent, i.e.,

$$\eta_x \text{ and } \eta_{x'} \text{ stochastically independent for every } x, x' \in \Omega_x. \tag{2}$$

If we consider $n \geq n_p$ measurement results $y = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ with corresponding experimental designs $x_1, \ldots, x_n \in \Omega_x$, the weighted least squares estimation p_n and the corresponding estimator P_n is defined as

$$p_n := P_n(y) := \underset{p \in \Omega_p}{\arg\min}\, \psi_n(y, p) \tag{3}$$

where the misfit function ψ_n is defined as

$$\psi_n : \mathbb{R}^n \times \Omega_p \to \mathbb{R}, (y, p) \mapsto \sum_{i=1}^{n} \left(\frac{y_i - f(x_i, p)}{\sigma_{x_i}} \right)^2.$$

The set of possible model parameters Ω_p is assumed to be compact and the model function $f(x, \cdot)$ is assumed to be continuous for every selectable design $x \in \Omega_x$. In this way, the existence of a minimum is ensured. If $\psi_n(y, \cdot)$ is also assumed to be injective, the minimum is also unique.

The optimal parameters p_n in Eq. (3) can be calculated with an optimization method for continuous optimization problems. A possible method is the SQP algorithm which is, e.g., described in (Nocedal and Wright, 1999, Chapter 18).

2.2 Asymptotic properties

Provided certain regularity conditions are met, the least squares estimators are consistent, asymptotically normal distributed and asymptotically efficient.

GMDD
7, 6439–6487, 2014

This asymptotic properties were first proved by Jennrich (1969) for the ordinary least squares estimator and also discussed in Malinvaud (1970) and Wu (1981). In White (1980), these properties were proved for the weighted least squares estimator and for the generalized least squares estimator in White and Domowitz (1984). A good summary for all three can be found in Amemiya (1983).

Consistency means that the estimated parameters converge in probability to the unknown exact parameters as the number of measurements goes to infinity. That is

$$P_n \xrightarrow{p} \hat{\boldsymbol{p}} \text{ as } n \to \infty$$

for the weighted least squares estimator P_n with the unknown exact model parameters $\hat{\boldsymbol{p}}$.

An estimator is asymptotically efficient if its variance converges to the Cramér–Rao bound as the number of measurements goes to infinity. The Cramér–Rao bound (see Cramér, 1946; Rao, 1945) is a lower bound for the variance of any unbiased estimator.

For the assumed measurement distribution Eqs. (1) and (2) with n measurements, this bound is the inverse of the Fisher information matrix

$$\mathbf{M}_n(\hat{\boldsymbol{p}}) := \sum_{i=1}^{n} \frac{\nabla_p f(x_i, \hat{\boldsymbol{p}}) \nabla_p f(x_i, \hat{\boldsymbol{p}})^T}{\sigma_{x_i}^2}$$

if the inverse exists. Here, $\nabla_p f(x_i, \hat{\boldsymbol{p}})$ denotes the gradient of $f(x_i, \cdot)$ at the point $\hat{\boldsymbol{p}}$.

In this case, the asymptotic behavior of the weighted least squares estimator can be summarized by its convergence in distribution as follows

$$\sqrt{n}(P_n - \hat{\boldsymbol{p}}) \xrightarrow{d} \mathcal{N}(0, \mathbf{M}_n(\hat{\boldsymbol{p}})^{-1} n) \text{ as } n \to \infty. \quad (4)$$

See, e.g., Seber and Wild (2003, Chapter 12) and Walter and Pronzato (1997, Chapter 3).

GMDD
7, 6439–6487, 2014

2.3 Optimal experimental designs

The accuracy of the weighted least square estimator P_n can be described by its covariance matrix. Due to the asymptotic distribution Eq. (4), this can be approximated by the inverse of the information matrix $\mathbf{M}_n(p_n)$, provided the matrix $\mathbf{M}_n(p_n)$ is nonsingular, i.e.,

$$\mathrm{cov}(P_n) \approx \mathbf{M}_n(p_n)^{-1}. \tag{5}$$

Therefore, the unknown model parameters can be determined more accurately the smaller the (approximated) covariance matrix of the estimator is.

Criteria $\phi : \mathbb{R}^{n_p \times n_p} \to \mathbb{R}^+$, such as the trace or determinant, are used in order to compare these matrices. (See, e.g., El-Monsef et al., 2009, for an overview of various criteria.) If the approximation Eq. (5) is used and $\mathbf{M}_n(p_n)$ is singular, the value of ϕ is set to infinity.

In the context of optimizing experimental designs, we assume $n \geq 0$ measurements have been carried out and designs for additional measurements should be selected from m designs $x'_1, \ldots, x'_m \in \Omega_x$. The choice for each design x'_i is expressed by a weight $w_i \in \{0, 1\}$ where 1 indicates the selection and 0 the contrary.

Hence, the resulting information matrix, depending on the choice $w \in \{0,1\}^m$ and the parameter vector $p_n \in \Omega_p$, is defined as

$$\mathbf{M}_n(w, p_n) := \mathbf{M}_n(p_n) + \sum_{i=1}^{m} w_i \frac{\nabla_p f(x'_i, p_n) \nabla_p f(x'_i, p_n)^T}{\sigma^2_{x'_i}}.$$

If the covariance matrix is approximated by the inverse of the information matrix, optimal (additional) designs, with respect to a criterion ϕ, are expressed by a solution of

$$\underset{w \in \{0,1\}^m}{\arg\min}\, \phi(\mathbf{M}_n(w, p_n)^{-1}). \tag{6}$$

GMDD
7, 6439–6487, 2014

These optimal designs are called local optimal designs because these designs are only optimal regarding the previously optimized model parameters p_n and not the unknown exact model parameters $\hat{\boldsymbol{p}}$.

Potential constraints on the choice of the designs can be realized by constraints on the weight w. For example, the number or the costs of the measurements can be limited by linear constraints on w. These constraints have to be considered in the above optimization problem Eq. (6).

2.4 Calculation of optimal experimental designs

A straight-forward way to solve the optimization problem Eq. (6) is to test all possible values of w. This direct approach is only practical for small m.

For bigger m, The optimization problem Eq. (6) is solved approximately. For this purpose, it is solved in the continuous rather than the discrete setting, i.e., the constraint $w \in \{0,1\}^m$ is relaxed to $w \in [0,1]^m$. Accordingly, the problem

$$\arg\min_{w \in [0,1]^m} \phi(\mathbf{M}_n(w, p_n)^{-1}) \tag{7}$$

is solved.

A possible algorithm to solve this continuous optimization problem is the SQP algorithm which is, e.g., described in Nocedal and Wright (1999, Chapter 18).

After the continuous problem Eq. (7) is solved, the solution is projected onto the integers with heuristics. An easy way is to round the continuous solution. Another is to sum up all continuous weights and then to choose as many designs with the highest continuous weights. Potential constraints on w still have to be considered by solving the continuous problem and the following projection onto an integer solution. The second heuristic, e.g., preserves constraints on the number of designs to choose.

Our numerical experiments with the application examples in Sect. 4 have shown that the solutions of the continuous problem Eq. (7) are already close to integer values. This behavior was also observed, for example, in Körkel (2002) and Körkel et al. (2004).

GMDD
7, 6439–6487, 2014

2.5 Robust optimal experimental designs

The information matrix $\mathbf{M}_n$ depends on the estimated parameters p_n if the parameters occur nonlinear in the model. This may lead to suboptimal designs if $\nabla_p f(\cdot, p_n)$ differs strongly from $\nabla_p f(\cdot, \hat{\boldsymbol{p}})$.

For this reason, we now consider a method which takes into account a possible nonlinearity of the parameters. This robust method was presented in Körkel (2002) and Körkel et al. (2004).

The main idea of the method is not to optimize the quality of the covariance matrix for a single parameter vector p_n as in Eq. (6), but to optimize the worst case quality within a whole domain which contains the unknown exact parameter vector $\hat{\boldsymbol{p}}$ with high probability.

For this purpose, a confidence region which contains $\hat{\boldsymbol{p}}$ with probability $\alpha \in (0,1)$ is approximated by

$$G_n(\alpha) := \{p \in \mathbb{R}^{n_p} \mid \|p - p_n\|^2_{\mathbf{M}_n(p_n)^{-1}} \leq \gamma(\alpha)\}. \tag{8}$$

Here, $\gamma(\alpha)$ is the α-quantile of the χ^2-distribution and $\|v\|_A := \sqrt{v^T A\, v}$ denotes the energy norm of the vector $v \in \mathbb{R}^{n_p}$ with respect to the positive definite matrix $A \in \mathbb{R}^{n_p \times n_p}$. The approximation of the confidence region arises from linearization of the model function f in point p_n and the assumption $P_n \sim \mathcal{N}(\hat{\boldsymbol{p}}, \mathbf{M}_n(p_n)^{-1})$.

If the worst case quality in the entire region $G_n(\alpha)$ shall be optimized, the optimization problem Eq. (6) becomes

$$\underset{w \in \{0,1\}^m}{\arg\min} \; \max_{p \in G_n(\alpha)} \phi(\mathbf{M}_n(w,p)^{-1}). \tag{9}$$

This min-max optimization problem can by solved only with considerable more computational effort compared to the optimization problem Eq. (6). In order to reduce this effort, the function $\phi(\mathbf{M}_n(w,\cdot)^{-1})$ is linearized in point p_n in the following way.

$$\phi(\mathbf{M}_n(w,p)^{-1}) \approx \phi(\mathbf{M}_n(w,p_n)^{-1}) + \nabla_p(\phi(\mathbf{M}_n(w,p)^{-1}))^T(p - p_n)$$

GMDD
7, 6439–6487, 2014

The resulting inner maximization problem can be solved analytically. It is

$$\max_{p \in G_n(\alpha)} \phi(\mathbf{M}_n(w,p_n)^{-1}) + \nabla_p(\phi(\mathbf{M}_n(w,p)^{-1}))^T(p-p_n) =$$

$$\phi(\mathbf{M}_n(w,p_n)^{-1}) + \gamma(\alpha)^{\frac{1}{2}} \|\nabla_p(\phi(\mathbf{M}_n(w,p_n)^{-1}))\|_{\mathbf{M}_n(p_n)},$$

as can be seen, e.g., in Körkel (2002). With this approach the optimization problem Eq. (9) is replaced by

$$\arg\min_{w \in \{0,1\}^m} \phi(\mathbf{M}_n(w,p_n)^{-1}) + \gamma(\alpha)^{\frac{1}{2}} \|\nabla_p(\phi(\mathbf{M}_n(w,p_n)^{-1}))\|_{\mathbf{M}_n(p_n)}. \quad (10)$$

This optimization problem again can be solved approximatively by solving the corresponding continuous problem and projecting this solution onto an integer solution as described in the previous subsection.

It should be noted that in this approach Eq. (10), the first and second derivatives of the model is used. In contrast, only the first derivative is used for local optimal designs Eq. (6).

2.6 Efficiency of experimental designs

A common way to describe the benefit of an experimental design is its efficiency. The efficiency of an experimental design $w \in \{0,1\}^m$ regarding a criterion ϕ and with n previous measurements is defined as follows.

$$E_\phi(w) := \min_{\hat{w} \in \{0,1\}^m} \frac{\phi(\mathbf{M}_n(\hat{w},\hat{\boldsymbol{p}})^{-1})}{\phi(\mathbf{M}_n(w,\hat{\boldsymbol{p}})^{-1})} \quad (11)$$

It should be noted that the searched parameter vector $\hat{\boldsymbol{p}}$ is used here. If this is not known, thus the efficiency can not be calculated.

The efficiency is always between 0 and 1 and is larger the better the experimental design is.

GMDD
7, 6439–6487, 2014

3 The Optimal Experimental Design Toolbox

We implemented the methods presented in the previous section for optimization of model parameters and experimental designs as a MATLAB toolbox named the *Optimal Experimental Design Toolbox*.

MATLAB (see MathWorks, 2011) was chosen because it supports vector and matrix operations and provides many numerical algorithms, especially for optimization. Moreover, MATLAB supports object oriented programming and therefore permits a simple structuring, modification and extension of the implementation. Another advantage of MATLAB is that it can easily interact with C and Fortran.

The toolbox is available at the Git repository (see Reimer, 2013) at GitHub under the GNU General Public License (see Foundation, 2007). It includes extensive commented source code and a detailed help integrated in MATLAB.

3.1 Provision of the model function

For the methods described in Sect. 2, the model function and its first and second derivative with respect to the model parameters is required.

Actually, the model function is required for the parameter optimization and, depending on the optimization method, also the first derivative. The first derivative is also required for the experimental design optimization. If the robust method is used also the second derivative is required.

The first step for using the *Optimal Experimental Design Toolbox* is to provide these functions. The *model* interface prescribes how this should be done. The functions need not be written in MATLAB itself, since MATLAB can call functions in C, C++ or Fortran.

The toolbox has several possibilities to provide the derivatives automatically. The *model_fd* class, e.g., provides the derivatives by approximation with finite differences. If the model function is given as an explicit symbolic function, the *model_explicit* class can provide the derivatives by symbolic differentiation with the *Symbolic Math Toolbox*. Figure 1 shows, for example, how a *model_explicit* object is created.

GMDD
7, 6439–6487, 2014

For the case the model function is given as a solution of an initial value problem, the *Optimal Experimental Design Toolbox* contains the *model_ivp* class. This class solves the parameter dependent initial value problem and calculates the necessary derivatives. Figure 2 shows how a *model_ivp* object is created.

The class takes advantage of the fact that the integration and differentiation of the differential equation can be interchanged if the model function is sufficiently often continuously differentiable. Required derivatives of the differential equation and initial value are calculated again by symbolic differentiation with the *Symbolic Math Toolbox*. The resulting initial value problems are solved with MATLABs *ode23s* function which can also solve stiff problems. Because the arising initial value problems for the derivatives are mutually independent, the solution of the initial value problems can be calculated in parallel using the *Parallel Computing Toolbox*.

3.2 Setup of the solver

Another important class in the *Optimal Experimental Design Toolbox* is the *solver* class. This class provides the methods for the optimization of parameter estimations and experimental designs. To perform one of these optimizations, the *solver* class has to be instantiated (see Fig. 3) and the necessary informations have to be passed to the *solver* object.

First of all, the model represented by an object which implements the *model* interface has to be set by the *set_model* method (see Fig. 4).

In addition, an initial estimation of the model parameters have to be set by the *set_initial_parameter_estimation* method (see Fig. 5).

Potential accomplished measurements can be set via the *set_accomplished_measurements* method. These measurements consist of the corresponding experimental designs together with their variances of the measurement errors. Also the measurement results themselves have to be passed for a parameter estimation (see Fig. 6).

GMDD
7, 6439–6487, 2014

Finally, if an optimization of experimental designs shall be performed, the selectable measurements have to be set by the *set_selectable_measurements* method (see Fig. 7). These measurements consist of the experimental designs and the variances of the measurement errors again.

3.3 Optimization of experimental designs and model parameters

Once the *solver* object is configured as described in the previous subsection, experimental designs or model parameters can be optimized via the *get_optimal_measurements* (see Fig. 8) respectively the *get_optimal_parameters* (see Fig. 9) method. Constraints on the experimental designs or model parameters can be passed to the corresponding method.

The *get_optimal_measurements* method can solve the optimization problem directly by trying all possible combinations or by solving the corresponding continuous problem and projecting onto an integer solution. For solving the continuous problem, the implementation of the SQP algorithm (see Nocedal and Wright, 1999, Chapter 18) provided by the *fmincon* function of the *Optimization Toolbox* is used. The first derivative of the objective function is provided in analytical form. This saves much of the computing time compared to derivatives calculated by finite differences. The Hessian matrix is approximated by the BFGS-update (see Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970).

The *get_optimal_parameters* method uses the Trust–Region–Reflective (see Coleman and Li, 1994; Coleman and Li, 1996) or the Levenberg–Marquard algorithm (see Levenberg, 1944; Marquardt, 1963; Moré, 1977) provided by the *lsqnonlin* function of the *Optimization Toolbox* to solve the least squares problem resulting from the parameter estimation. The first derivative of the objective function is also provided analytically.

Furthermore, the expected quality of the resulting parameter estimation for any selection of experimental designs can be calculated using the *get_quality* method of the

GMDD
7, 6439–6487, 2014

solver object. Thus, for example, the increase in quality by adding or removing experimental designs can be determined.

In the methods of the *Optimal Experimental Design Toolbox*, often reusable (intermediate) results occur. The toolbox takes advantage of this by internally saving and reusing appropriate results. Thus, the execution time is significantly reduced. Multiple occurring matrix multiplications within a calculation are an example. Also, reusable results are cached between different method calls. An example scenario is a re-optimization of designs with other constraints, such as another maximum number of allowed measurements. Here, the derivatives of the model function calculated in the previous optimization is reused.

3.4 Changeable options

Many settings for the optimization of experimental designs or model parameters are changeable. These can be altered by the *set_option* method of the *solver* object (see Fig. 10). The desired options can be set using property-value pairs, as already known from MATLAB. This means, the name of the option has to be passed to the method as first argument and the new value as second argument.

Estimation method: For example, the estimation method for the quality of experimental designs can be selected by the *estimation_method* option. The standard *point* estimation method and the robust *region* estimation method, both presented in Sect. 2, are supported. The *region* estimation method is the default setting.

Confidence level: Moreover, the level of confidence for the confidence region at the *region* estimation method, represented by α in Sect. 2.5, can be set by the *alpha* option. The default value is 0.95.

Prior parameter estimation: Furthermore, it can be chosen whether a parameter optimization should be performed before optimizing experimental designs. This would improve the estimations of the quality of experimental designs. This can

GMDD
7, 6439–6487, 2014

be set by the *parameter_estimation* option and the values *yes* or *no*. To save computational time no previous parameter optimization is performed by default.

Quality criterion: The quality criterion, which is applied to the covariance matrix and represented in Sect. 2.1 as ϕ, can also be chosen. Therefore, an object of a class which implements the *criterion* interface have to be passed with the *criterion* option. The *criterion* interface prescribes the syntax of the criterion function and its necessary derivatives. The trace of the covariance is the default criterion and implemented by the *criterion_A* class.

Parameter scaling: Furthermore, it can be chosen whether the covariance matrix should be scaled before applying the quality criterion or not by the *scale_covariance_matrix* option and the values *yes* and *no*. Scaling the covariance matrix allows to optimize the quality of each parameter uniformly and is enabled by default. The model parameters are scaled by default for the parameter optimization, too. This can be changed by the *po_scale_parameter* option and the values *yes* and *no*.

Optimization algorithm for experimental design: Finally, the optimization algorithm for the experimental design problem can be configured. The direct and the relaxed method, described in 2.4, can be chosen as solution algorithm. The corresponding option is *ed_algorithm* and the values are *direct* respectively *local_sqp*. For time reasons by default the experimental design problem is solved by the relaxed method. Furthermore, the number of function evaluations and iterations by the SQP algorithm can be constrained by the options *ed_max_fun_evals* and *ed_max_iter*.

Optimization algorithm for parameter estimation: Similarly, the optimization algorithm for the parameter estimation problem can be configured. The Trust–Region–Reflective (see Coleman and Li, 1994; Coleman and Li, 1996) and the Levenberg–Marquard algorithm (see Levenberg, 1944; Marquardt, 1963; Moré, 1977) can be

GMDD
7, 6439–6487, 2014

chosen as solution algorithm with the option *po_algorithm* and the values *trust-region-reflective* respectively *levenberg-marquardt*. The Trust–Region–Reflective algorithm is the default algorithm. By default the model parameters are scaled for the optimization. This can be influenced by the *po_scale_parameter* option and the values *yes* and *no*. Furthermore, the number of function evaluations and iterations can be limited through the options *po_max_fun_evals* and *po_max_iter*.

3.5 Help and documentation

The *Optimal Experimental Design Toolbox* also provides an extensive integrated help. It can be viewed in the command window by the MATLAB command *help* or in the help browser of MATLAB by its *doc* command (see Fig. 11).

The layout of the help of the *Optimal Experimental Design Toolbox* is based on the design of the help also used by MATLAB and other toolboxes. Thus the user does not have to get used to a new layout. The help includes, besides system requirements and version informations, a user's guide with a step by step instruction how to optimize experimental designs and model parameters. Demos show how to work with the toolbox in practice. In addition, a detailed description for every class and method is available.

4 Application examples

In this section, numerical experiments together with their results regarding the optimization of model parameters and experimental designs are presented for two models from geophysics, namely for sediment concentration in seawater which floods coastal salt marshes.

Coastal salt marshes have an important ecological function with their diverse flora and as a nursery for migratory birds. Furthermore they have the ability of dissipating current and wave energy and therefore reducing erosional forces at dikes and coastal areas.

GMDD
7, 6439–6487, 2014

With these models, the vertical accretion of coastal salt marshes can be predicted. If sea level rise is considered too, the future ability of coastal salt marshes to grow faster as sea increases and thus to survive can be estimated. Depending on this, measures to protect these salt marshes can be taken.

This application example arose in cooperation with the Geographical Institute of the Christian-Albrechts University of Kiel. There, the parameters of these two models should be determined. Carrying out the required measurements of the sediment concentrations is time consuming and laborious. For this reason, it is advantageous to know under which conditions and how many measurements should be carried out.

4.1 The models

Both models are zero-dimensional point models, which describe the sediment concentration in seawater that floods coastal salt marshes within a tidal cycle. The first model has two model parameters, was described in Temmerman et al. (2003) and adapted for the local salt marshes in Schuerch et al. (2013). The second model has three model parameters, is an extension of the first model and subject of current research.

4.1.1 The C_2-model

The first model is called the C_2-model. Here, the sediment concentration is modeled by the function $C : [t_S, t_E) \to \mathbb{R}^+$ and has the unit kg m^{-3}. Furthermore, t_S is the start time of the inundation of the salt marsh and t_E the end time. The concentration C is given implicit as solution of the initial value problem

$$C'(t) = \begin{cases} \frac{-w_s C(t) + (C_0 - C(t))h'(t)}{h(t) - E} & \text{if } h'(t) > 0 \\ \frac{-w_s C(t)}{h(t) - E} & \text{else} \end{cases}$$

for all $t \in (t_S, t_E)$ and $C(t_S) = C_0$. (12)

GMDD
7, 6439–6487, 2014

Here, $C_0 \geq 0$ is the initial sediment concentration at the flooding seawater and $w_S \geq 0$ the settling velocity of the sediment in the unit $m\,s^{-1}$. Moreover, the function

$$h : \mathbb{R} \to \mathbb{R}, t \mapsto \frac{a}{1 + \left(\frac{t - x_0}{b}\right)^2} + h_{HW} - h_{MHW}$$

describes the time-dependent water surface elevation and E the elevation of the marsh both relative to a fixed datum. Here, a, b and x_0 are constants describing the change in the water level, h_{MHW} the mean high water level and h_{HW} the high water level of a certain tidal inundation. The start and end time t_S and t_E of the inundation are the points where the height h equals the elevation of the marsh E.

The concentration C thus decreases continuously within a tidal cycle depending on the settling velocity w_S which is described by the term

$$-\frac{w_s C(t)}{h(t) - E}$$

in Eq. (12). During the flood phase, the reduced concentration is partially compensated by new inflowing sea water. This is described by the term

$$\frac{(C_0 - C(t))h'(t)}{h(t) - E}$$

in the first case of Eq. (12).

The values used in the water surface elevation function h, for the local salt marsh, are shown in Table 1 (see also Schuerch et al., 2013). The high water level h_{HW} of the current tidal inundation is measured or taken from predictions.

The initial sediment concentration C_0 and the settling velocity w_S are only roughly known and therefore model parameters. Initial estimations can be found in Table 2.

GMDD
7, 6439–6487, 2014

4.1.2 The C_3-model

The second model is an extension of the C_2-model and is called the C_3-model. Here the model parameters C_0 and w_S are substituted by

$$C_0 = k(h_{HW} - E)$$
$$w_s = r(C_0)^s = rk^s(h_{HW} - E)^s.$$

Where $k \geq 0$, $r \geq 0$ and $s \geq 0$ are unknown model parameters.

On the one hand, a linear relationship between the initial sediment concentration and the high water level is assumed, where during heavy flooding a higher sediment concentration is assumed. On the other hand, a relationship between the initial sediment concentration and the settling velocity is assumed. This is an empirical approximation of the so-called flocculation effect.

Initial estimations for the parameters in this model can be found in Table 3.

4.2 Numerical experiments

We performed several numerical experiments to compare the benefit of optimized with unoptimized measurement conditions. Also, the benefit of different approaches to optimization measurement conditions was compared. Using these results, an appropriate approach for the optimization of conditions for real measurements was selected.

The approaches introduced in Sect. 2 and implemented by the *Optimal Experimental Design Toolbox* described in Sect. 3 were used for the numerical experiments. For that, we used the *model_ivp* class which allows to calculate the solution of an initial value problem and its first and second derivatives with respect to the model parameters. The C_2-model was implemented by the *model_C2* class and the C_3-model by the *model_C3* class which is a subclass of the *model_C2* class.

For our numerical experiments, we used the model output with the model parameters in Tables 2 and 3 plus an additive normal distributed measurement error with zero

GMDD
7, 6439–6487, 2014

expectation as artificial measurement results. As standard deviation of the measurement error, we once chose 10^{-2} and once 10^{-1}.

In our numerical experiments, we alternately selected a fixed number of experimental designs and estimated the model parameters with corresponding measurement results. We carried out each experiment ten times and averaged the results to minimize the influence of randomness.

For the initial parameter estimation, we used the values presented in Table 4.

Moreover the bounds for the model parameters shown in Table 5 were used for the parameter estimations.

The experimental designs for these models consist of the time point of the measurement and the high water level of the tidal inundation. A set of thirty selectable experimental designs was specified. They were obtained by combining three different high water levels of the tidal inundation (1.5, 2.0 and 2.5 m) with ten time points equidistantly spread over the inundation period.

For choosing the experimental designs, we compared the standard and the robust approach presented in Sect. 3 with the trace as quality criterion together with uniformly distributed experimental designs. The optimization problems for the experimental designs were once solved exact in the discrete variant and once approximatively in the relaxed variant. To evaluate all these methods, we compared the resulting parameter estimations with the correct model parameters in Tables 2 and 3.

We further investigated whether the number of measurements after which new experimental designs are optimized had an impact on the accuracy of the parameter estimation. For this purpose, different numerical experiments were performed where the parameters and experimental designs have been optimized after each one, three resp. five measurements. Altogether fifty measurements were simulated at each experiment with the C_2-model. For the C_3-model, hundred and fifty measurements were simulated at each experiment since the model is more complex and therefore a sufficiently accurate estimation of its parameters might be more difficult.

GMDD
7, 6439–6487, 2014

4.3 Accuracy of the parameter estimations

In this subsection, we compare the accuracy of the parameter estimations resulting from the previously described numerical experiments. Some results are illustrated in Figs. 12 and 13.

4.3.1 Results for the C_2-model

The accuracy of the parameter estimations for the C_2-model only improved marginally after four to twelve measurements independently of the choice of the experimental designs. The maximal accuracy was achieved faster the more frequently the experimental designs and parameters were optimized. However, the maximal achieved accuracy was independent of the frequency.

With uniformly distributed experimental designs the maximum accuracy was slightly worst than with optimized experimental designs. Additional four to six more measurements were needed compared to optimized experimental designs in order to achieve their accuracy.

Although the parameters nonlinearly occur in this model, it made close to no difference whether the standard or the robust approach for the optimization of the experimental designs was used.

In both approaches, the accuracy was slightly better in the first iterations if the discrete optimization problem was solved directly as if the relaxed optimization problem was solved. The solutions of the relaxed optimization problems were almost all nearly integer.

The different standard deviations of the measurement errors only influenced the maximal accuracy achieved which was of course higher at a higher standard deviation.

GMDD
7, 6439–6487, 2014

4.3.2 Results for the C_3-model

After ten to twenty-five measurements, the accuracy of the parameter estimations for the C_3-model with optimized experimental designs only improved slightly. Again, the maximal accuracy was achieved faster, the fewer measurements were performed per iteration and the maximal achieved accuracy was independent of the number of measurements per iteration.

With uniformly distributed experimental designs, the maximum accuracy was achieved after twenty-four to sixty measurements. Furthermore, the maximal accuracy was worse by about a factor of ten compared to the accuracy achieved by (standard) optimized experimental designs.

The standard approach for optimizing experimental designs resulted in a slightly better accuracy compared to the robust approach. Again, if the underlying optimization problem was solved in the discrete rather in the relaxed variant, the accuracy was slightly better for both approaches. Also in these experiments, the solutions of the relaxed optimization problems were almost all nearly integer.

Again, the different standard deviations of the measurement errors only influenced the maximal accuracy achieved.

4.3.3 Conclusions regarding the approach for optimizing experimental designs

Optimized experimental designs provided a much more accurate parameter estimation than uniformly distributed experimental designs independent of the chosen optimization approach. Furthermore, only about half as many measurements were needed to archive the same accuracy with optimized experimental designs as with uniformly distributed experimental designs. In the more complex model, the difference was even greater.

The robust approach achieved no higher accuracy compared to the standard approach. In the complex model, the robust approach was even slightly less accurate. This may indicate that the additional approximations in the robust approach offset the

GMDD
7, 6439–6487, 2014

increase in accuracy, which should be achieved by taking into account the nonlinearity. Since a considerably higher computational effort is associated with the robust approach, the standard approach should be preferred, at least for these models.

The direct solution of the discrete optimization problems compared to the solution of the continuous relaxed optimization problems only resulted in a small increase in accuracy. The fact that the solutions of the relaxed optimization problems were almost all nearly integer was another indication that the difference between both solutions was small. This fact was also observed, for example, in Körkel (2002) and Körkel et al. (2004). For these reasons and because the direct solving requires much more computational effort, the relaxed problem should be solved, at least for these models.

4.4 Efficiency for the experimental designs

We also calculated the efficiencies of the used experimental designs. Some results are illustrated in Figs. 14 and 15.

The results emphasized the already seen importance of the optimization of the experimental designs. In particular, the advantage in the case of few measurements carried out so far was highlighted. Again, the slight advantage of the standard approach over the robust approach was visible. With increasing number of accomplished measurements, the selection strategy of new measurements became less important as the amount and thus the influence of the new measurements compared to those of the accomplished measurements decreased.

4.5 Distribution of optimal measuring points

In this subsection, we compare the distribution of the measuring points optimized in the previously described numerical experiments. Graphical representation of the distribution of the measuring points from some numerical experiments are shown in Figs. 16 and 17.

GMDD
7, 6439–6487, 2014

4.5.1 Distribution for the C_2-model

The optimized measuring points were almost exclusively located at the start and end of the inundation periods. At the start of the inundation period, both approaches in the discrete variant favored lower high water levels unlike both approaches in relaxed variant which favored higher high water levels. At the end of the inundation period, the standard approach in both variants favored lower high water levels unlike the robust approach in both variants which favored higher high water levels.

4.5.2 Distribution for the C_3-model

For the C_3-model the optimized measuring points accumulated at the end of the inundation periods. All approaches favored lower high water levels. With an increasing number of measurements per iteration the robust approach in both variants also preferred measurements in the middle of the inundation periods with the highest high water level.

4.5.3 Conclusions regarding the distribution of optimal measuring points

The numerical experiments showed that measurements at the start and end of the inundation periods should be preferred for the C_2-model.

Measurements at the start of the inundations can be justified by the fact that one parameter of the model is the concentration at the start of the inundation. The fact that the settling velocity as second model parameter most affects the concentration at the end of the inundations justifies measurements here. This can be confirmed by an examination of the ordinary differential equation of the model derived with respect to the settling velocity. The derivative of the model with respect to the settling velocity is zero at the start of the inundation and is getting smaller the further the inundation progresses. Its absolute greatest value it thus reached at the end of the inundation.

GMDD
7, 6439–6487, 2014

The experiments with the C_3-model showed that here measurements at end of the inundation periods should be preferred. In this model, the concentration at the start is no parameter but is affected by a parameter that also influences the settling velocity. For this reason, measurements are not suggested at the start.

For both models the high water level seemed to play a minor role for the choice of measuring points.

As a rule of thumb one can say that measurements should be carried out at the end of an inundation period and also some at the start if the C_2-model is used.

5 Conclusions

In this paper we presented two different approaches for optimizing experimental design for parameter estimations. One method was based on the linearization of the model with respect to its parameters, the other takes into account a possible nonlinearity of the model parameters. Both methods were implemented in our presented *Optimal Experimental Design Toolbox* for MATLAB.

Using application examples, we showed that model parameters can be determined much more accurately if the corresponding measurement conditions were optimized. Especially for time-consuming or costly measurements, it is therefore useful to optimize the measurement conditions with the *Optimal Experimental Design Toolbox*.

Acknowledgements. This project was funded by the Deutsche Forschungsgemeinschaft (DFG) Cluster “The Future Ocean”.

References

Amemiya, T.: Nonlinear regression models, Handbook of Econometrics, 1, 333–389, 1983. 6444

Broyden, C. G.: The convergence of a class of double-rank minimization algorithms: 2. The new algorithm, IMA J. Appl. Math., 6, 222–231, doi:10.1093/imamat/6.3.222, 1970. 6451

GMDD

7, 6439–6487, 2014

Coleman, T. F. and Li, Y.: On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds, Math. Program., 67, 189–224, 1994. 6451, 6453
Coleman, T. F. and Li, Y.: An interior, trust region approach for nonlinear minimization subject to bounds, SIAM J. Optimiz., 6, 418–445, 1996. 6451, 6453
Cramér, H.: Mathematical Methods of Statistics, Almqvist & Wiksells Akademiska Handböcker, Princeton University Press, 1946. 6444
El-Monsef, M. M. E. A., Rady, E. A., and Seyam, M. M.: Relationships among Several Optimality Criteria, Interstat, 2009. 6445
Fletcher, R.: A new approach to variable metric algorithms, Comput. J., 13, 317–322, doi:10.1093/comjnl/13.3.317, 1970. 6451
Foundation, F. S.: GNU General Public License, available at: http://www.gnu.org/licenses/gpl-3.0-standalone.html (last access: 30 July 2014), version 3, 2007. 6449
Goldfarb, D.: A family of variable-metric methods derived by variational means, Math. Comput., 24, 23–26, 1970. 6451
Jennrich, R. I.: Asymptotic properties of non-linear least squares estimators, Ann. Math. Stat., 40, 633–643, doi:10.1214/aoms/1177697731, 1969. 6444
Körkel, S.: Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen, Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg, 2002. 6446, 6447, 6448, 6461
Körkel, S., Kostina, E., Bock, H. G., and Schlöder, J. P.: Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes, Optim. Method. Softw., 19, 327–338, doi:10.1080/10556780410001683078, 2004. 6446, 6447, 6461
Levenberg, K.: A method for the solution of certain problems in least squares, Q. Appl. Math., 2, 164–168, 1944. 6451, 6453
Malinvaud, E.: The consistency of nonlinear regressions, Ann. Math. Stat., 41, 956–969, 1970. 6444
Marquardt, D. W.: An algorithm for least-squares estimation of nonlinear parameters, J. Soc. Ind. Appl. Math., 11, 431–441, 1963. 6451, 6453
MathWorks: Matlab R2011b Getting Started Guide, available at: www.mathworks.de/help/releases/R2011b/pdf_doc/matlab/getstart.pdf (last access: 30 July 2014), 2011. 6449

GMDD
7, 6439–6487, 2014

Moré, J.: The Levenberg–Marquardt algorithm: implementation and theory, in: Numerical Analysis, edited by: Watson, G., Vol. 630 of Lecture Notes in Mathematics, Springer, Berlin Heidelberg, 105–116, doi:10.1007/BFb0067700, 1977. 6451, 6453
Nocedal, J. and Wright, S. J.: Numerical Optimization, Springer series in operations research and financial engineering, 1999. 6443, 6446, 6451
Pronzato, L. and Pázman, A.: Design of Experiments in Nonlinear Models: Asymptotic Normality, Optimality Criteria and Small-Sample Properties, Lecture Notes in Statistics, Springer, New York, NY, 2013. 6441
Rao, R. C.: Information and the accuracy attainable in the estimation of statistical parameters, Bull. Calcutta Math. Soc., 37, 81–91, 1945. 6444
Reimer, J.: Optimal Experimental Design Toolbox, available at: http://github.com/jor-/oed (last access: 30 July 2014), version 0.7, 2013. 6449
Schuerch, M., Vafeidis, A., Slawig, T., and Temmerman, S.: Modeling the influence of changing storm patterns on the ability of a salt marsh to keep pace with sea level rise, J. Geophys. Res., 118, 84–96, doi:10.1029/2012JF002471, 2013. 6455, 6456
Seber, G. A. F. and Wild, C. J.: Nonlinear Regression, Wiley Series in Probability and Statistics, Wiley-Interscience, 2003. 6444
Shanno, D. F.: Conditioning of quasi-Newton methods for function minimization, Math. Comput., 24, 647–656, 1970. 6451
Stigler, S. M.: Gauss and the invention of least squares, Ann. Stat., 9, 465–474, 1981. 6442
Temmerman, S., Govers, G., Meire, P., and Wartel, S.: Modelling long-term tidal marsh growth under changing tidal conditions and suspended sediment concentrations, Scheldt estuary, Belgium, Mar. Geol., 193, 151–169, doi:10.1016/S0025-3227(02)00642-4, 2003. 6455
Walter, É. and Pronzato, L.: Identification of Parametric Models from Experimental Data, Communications and Control Engineering, Springer, 1997. 6444
White, H.: Nonlinear regression on cross-section data, Econometrica, 48, 721–746, 1980. 6444
White, H. and Domowitz, I.: Nonlinear regression with dependent observations, Econometrica, 52, 143–162, 1984. 6444
Wu, C.-F.: Asymptotic theory of nonlinear least squares estimation, Ann. Stat., 9, 501–513, 1981. 6444

GMDD
7, 6439–6487, 2014

Table 1. Values used for the water surface elevation function h.

	a	b	x_0	h_{MHW}	E
local value	3.7506	19447.1	−1301.0	3.75 m	1.3 m

GMDD
7, 6439–6487, 2014

Table 2. Estimated parameter values for the C_2-model.

	C_0	w_S
estimated value	$0.1\,\mathrm{kg\,m^{-3}}$	$10^{-5}\,\mathrm{m\,s^{-1}}$

GMDD
7, 6439–6487, 2014

Table 3. Estimated parameter values for the C_3-model.

	k	r	s
estimated value	0.25	10^{-5}	0.5

GMDD
7, 6439–6487, 2014

Table 4. Initial parameter values.

	C_0	w_S	k	r	s
initial value	5	2×10^{-7}	12.5	2×10^{-7}	3

GMDD
7, 6439–6487, 2014

Table 5. Parameter bounds.

	C_0	w_S	k	r	s
lower bound	10^{-4}	10^{-8}	10^{-4}	10^{-8}	10^{-1}
upper bound	10^{4}	1	10^{4}	1	5

GMDD
7, 6439–6487, 2014

```
model_object = model_explicit('p*t^2', 'p', 't')
% 1. input: the model function as symbolic formula
% 2. input: the parameter variable(s)
% 3. input: the experimental design variable(s)
% return: a model object which implements the model interface
```

Figure 1. Create a model with a symbolic model function.

GMDD
7, 6439–6487, 2014

```
model_object = model_ivp('-y+(t+1)*b', '[a,b]', 'y', 'a', 't', [1,10])
% 1. input: the formula of the derivative of the model function
% 2. input: the model parameter variable(s)
% 3. input: the model function variable
% 4. input: the initial value of the model function
% 5. input: the dependent variable in the model function
% 6. input: the interval of integration
% return: a model object which implements the model interface
```

Figure 2. Create a model with a model function given as solution of an initial value problem.

GMDD
7, 6439–6487, 2014

```
solver_object = solver()
% return: a solver object
```

Figure 3. Create a solver object.

GMDD
7, 6439–6487, 2014

```
solver_object.set_model(model_object)
% input: an object that implements the model interface
```

Figure 4. Set the model.

GMDD
7, 6439–6487, 2014

```
solver_object.set_initial_parameter_estimation([1, 2])
% input: the initial estimation of the model parameters
```

Figure 5. Set the initial parameter estimation.

GMDD
7, 6439–6487, 2014

```
solver_object.set_accomplished_measurements((1:5)', 0.01*ones(5,1), -↩
    exp((1:5)'))
% 1. input: the experimental designs of accomplished measurements
% 2. input: the variances of the associated measurement errors
% 3. input: the associated measurement results
```

Figure 6. Set accomplished measurements.

GMDD
7, 6439–6487, 2014

```
solver_object.set_selectable_measurements((6:10)', 0.01*ones(5, 1))
% 1. input: the selectable experimental designs
% 2. input: the variances of the associated measurement errors
```

Figure 7. Set selectable measurements.

GMDD
7, 6439–6487, 2014

```
optimal_measurements = solver_object.get_optimal_measurements(3)
% input: the maximum number of measurements allowed
% return: the optimal subset of the selectable measurements with a ↩
    number of measurements less or equal to the restriction
```

Figure 8. Optimize experimental designs.

GMDD
7, 6439–6487, 2014

```
optimal_parameters = solver_object.get_optimal_parameters([0,0],[9,9])
% 1. input: the lower bound of the model parameters
% 2. input: the upper bound of the model parameters
% return: a parameter estimation resulting from the accomplished ↩
    measurements which takes into account the passed constraints
```

Figure 9. Optimize model parameters.

GMDD
7, 6439–6487, 2014

```
solver_object.set_option('option_name', option_value)
% 1. input: the name of the option which should be changed
% 2. input: the new value of the option
```

Figure 10. Change an option.

GMDD
7, 6439–6487, 2014

```
doc optimal_experimental_design_toolbox
```

Figure 11. Get the documentation.

GMDD
7, 6439–6487, 2014

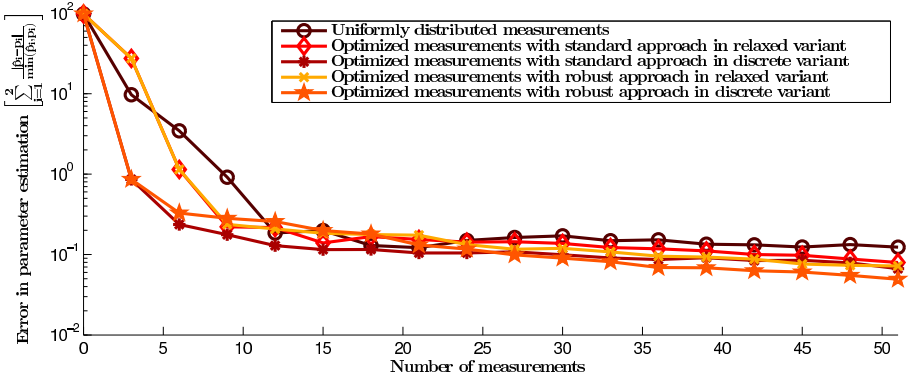

Figure 12. Averaged error in the parameter estimation from ten optimization runs with the C_2-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

GMDD
7, 6439–6487, 2014

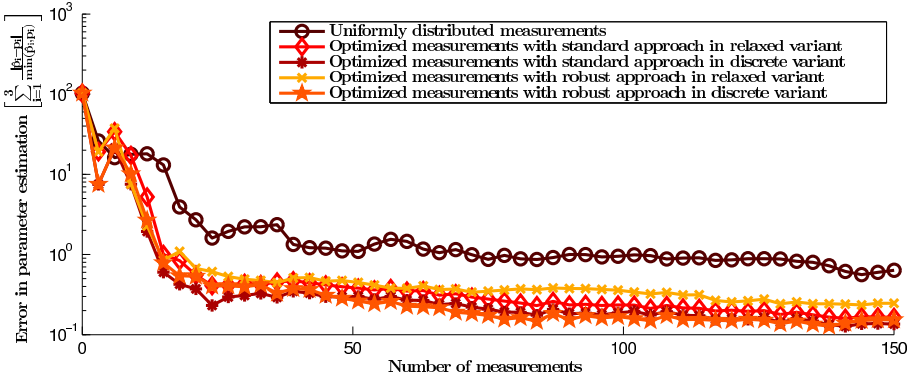

Figure 13. Averaged error in the parameter estimation from ten optimization runs with the C_3-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

GMDD
7, 6439–6487, 2014

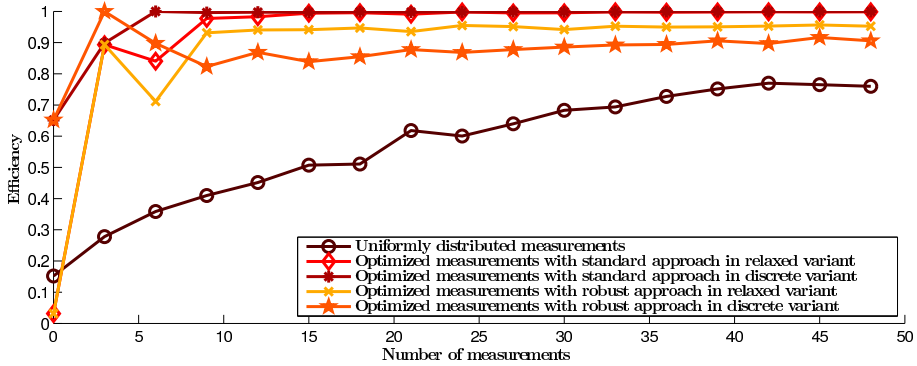

Figure 14. Averaged efficiency for the experimental designs from ten optimization runs with the C_2-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

GMDD
7, 6439–6487, 2014

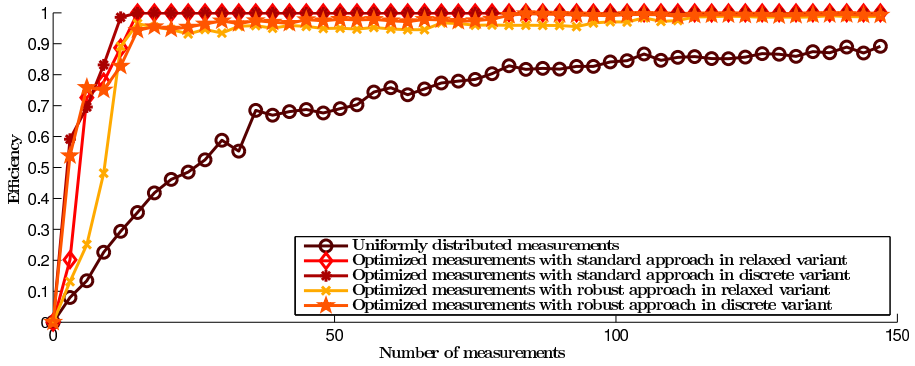

Figure 15. Averaged efficiency for the experimental designs from ten optimization runs with the C_3-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

GMDD
7, 6439–6487, 2014

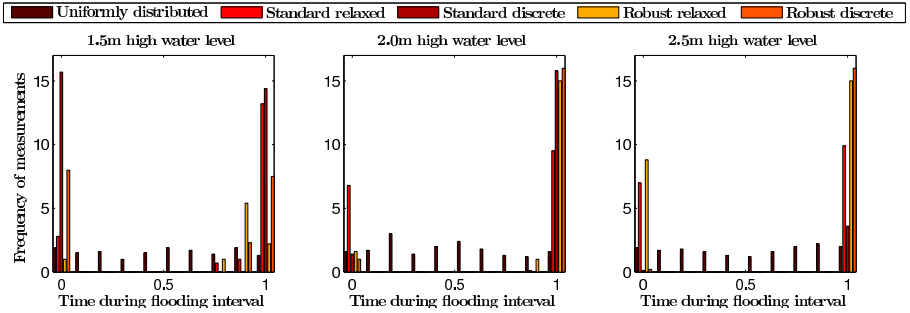

Figure 16. Averaged frequency of measurements from ten optimization runs with the C_2-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

GMDD
7, 6439–6487, 2014

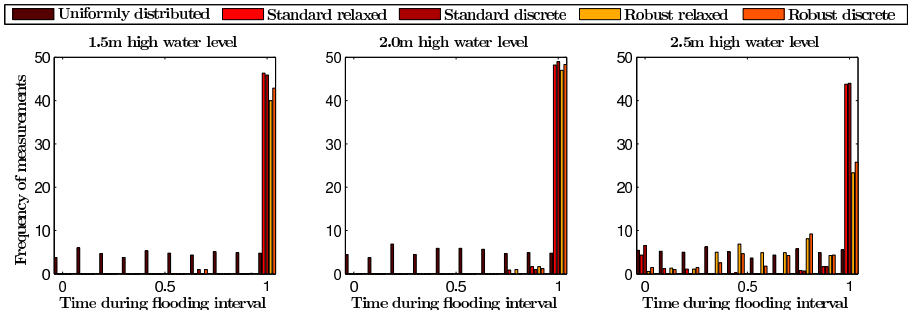

Figure 17. Averaged frequency of measurements from ten optimization runs with the C_3-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

GMDD
7, 6439–6487, 2014