Geoscientific
Model Development

Optimization of model parameters and experimental designs with the Optimal Experimental Design Toolbox (v1.0) exemplified by sedimentation in salt marshes

J. Reimer[1], M. Schuerch[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

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

Abstract. The geosciences are a highly suitable field of application for optimizing model parameters and experimental designs especially because many data are collected.

In this paper, the weighted least squares estimator for optimizing model parameters is presented together with its asymptotic properties. A popular approach to optimize experimental designs called local optimal experimental designs is described together with a lesser known approach which takes into account the potential nonlinearity of the model parameters. These two approaches have been combined with two 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 application is described.

In numerical experiments, the model parameters and experimental design were optimized using this toolbox. Two existing models for sediment concentration in seawater and sediment accretion on salt marshes of different complexity served as an application example. The advantages and disadvantages of these approaches were compared based on these models.

Thanks to optimized experimental designs, the parameters of these models could be determined very accurately with significantly fewer measurements compared to unoptimized experimental designs. The chosen optimization approach played a minor role for the accuracy; therefore, the approach with the least computational effort is recommended.

1 Introduction

Mathematical models often contain roughly known model parameters which are optimized based on measurements. The resulting accuracy of the model parameters depends on the conditions, also called experimental setups or experimental designs, under which these measurements are carried out. These experimental designs can be optimized so that the resulting accuracy is maximized. Thus, the effort and cost of measurements can be significantly reduced.

The optimization of experimental designs is therefore particularly interesting for geosciences, where much money is spent on data collection. However, few application examples exist in this field (see Guest and Curtis, 2009, for an overview). This article aims to promote this approach in geosciences and exemplarily apply it to an existing salt marsh accretion model (Schuerch et al., 2013).

In optimizing experimental design, the main problem is to quantify the information content of the measurements to be planned. In general, this can only be done approximatively. There are several approaches available. In Sect. 2, four different approaches to optimize experimental designs together with the weighted least squares estimator for model parameters are presented. Each of these four approaches makes a different trade-off between accuracy and computational effort.

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 the potential

nonlinearity of the model parameters. Both approaches are combined with two different approaches of solving the underlying discrete optimization problem.

To the author's knowledge, there is no open-source software available that can apply all four of these approaches. The only software using this robust approach is a closed-source software called VPLAN which was introduced in Körkel (2002). For the local optimal approach, several implementations are available, but there is no open-source software written in MATLAB. All four approaches, together with approaches to optimize model parameters, were implemented in a MATLAB toolbox called the *Optimal Experimental Design Toolbox*. Its structure and application is described in Sect. 3.

We have chosen two models as application examples, simulating the suspended sediment concentration on salt marshes during tidal inundation and resulting accretion rates on these marshes (Krone, 1987; Temmerman et al., 2003; Schuerch et al., 2013). Both models are zero-dimensional point models and differ in their complexity and number of parameters. These models can be used as a basis to predict the survival capability of salt marshes under the influence of expected global sea level rise.

To use these models for local salt marshes, their parameters have to be adapted to the local environmental conditions. The required measurements are very time-consuming and costly. Using the presented approaches, these measurements could be carried out much more efficiently. These two models are described together with the attendant numerical experiments and the associated results in Sect. 4.

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 estimated model parameters. These estimated 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 estimator should be derived from the statistical properties of the measurement errors, for example, a maximum likelihood estimator. Often the measurement errors are assumed to be normally distributed; this leads to the least squares estimators. They are the most widely used class of estimators since their introduction by Gauss and Legendre (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 the event of heteroscedastic measurement errors. This estimator and its asymptotic properties are presented in the following sub-

section. The generalized least squares estimator is a further generalization which takes into account the 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

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

Here, $\Omega_x \subseteq \mathbb{R}^{n_x}$ is the set of feasible experimental designs, and $\Omega_p \subset \mathbb{R}^{n_p}$ is the set of feasible model parameters from which the unknown exact parameter vector $\hat{p} \in \Omega_p$ 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 with the expectation $f(x, \hat{p})$ and standard deviation $\sigma_x > 0$.

A1a. $\eta_x \sim \mathcal{N}(f(x, \hat{p}), \sigma_x^2)$ for every $x \in \Omega_x$.

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

A1b. η_x and $\eta_{x'}$ are stochastically independent for every $x, x' \in \Omega_x$.

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 are defined as

$$p_n := P_n(y) := \arg\min_{p \in \Omega_p} \psi_n(y, p), \tag{1}$$

where the misfit function ψ_n is defined as

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

With the following assumptions, the existence of a minimum is ensured.

A2. $f(x, \cdot)$ is continous for every $x \in \Omega_x$.

A3. Ω_p is compact (closed and bounded).

If $\psi_n(y, \cdot)$ is convex, the minimum is also unique.

The parameter estimation p_n in Eq. (1) can be calculated with an optimization method for continuous optimization problems. A possible method is the sequential quadratic programming (SQP) algorithm which is, for example, 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 normally distributed and asymptotically efficient.

These 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,

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

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

For consistency, the following assumptions are sufficient in addition to the previous assumptions A1 to A3 (Seber and Wild, 2003, p. 565).

A4a. $n^{-1}B_n$ converges uniformly with $B_n : \Omega_p \times \Omega_p \to \mathbb{R}, (p, p') \longmapsto \sum\limits_{i=1}^{n} f(x_i, p) f(x_i, p') \sigma_{x_i}^{-2}$.

A4b. $\bar{D}(p, \hat{p}) = 0 \Rightarrow p = \hat{p}$ for all $p \in \Omega_p$ with $\bar{D} := \lim\limits_{n \to \infty} n^{-1} D_n$ and $D_n : \Omega_p \times \Omega_p \to \mathbb{R}, (p, p') \longmapsto$ $\sum\limits_{i=1}^{n} (f(x_i, p) - f(x_i, p'))^2 \sigma_{x_i}^{-2}$ ($\bar{D}$ is well defined by assumption A4a).

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 (Cramér, 1946; Rao, 1945) is a lower bound for the variance of any unbiased estimator.

For asymptotic efficiency, the following assumptions are sufficient in addition to the previous assumptions A1 to A4 (Seber and Wild, 2003, p. 571).

A5. $\hat{p}$ is an interior point of Ω_p. Let $\hat{\Omega}_p \subseteq \Omega_p$ be an open neighborhood of $\hat{p}$.

A6. $f(x_i, \cdot)$ is twice continuously differentiable in $\hat{\Omega}_p$.

A7. $n^{-1} M_n$ converges uniformly with $M_n : \hat{\Omega}_p \to \mathbb{R}^{n_p \times n_p}, p \longmapsto \sum\limits_{i=1}^{n} \nabla_p f(x_i, p) \nabla_p f(x_i, p)^T \sigma_{x_i}^{-2}$.

A8. $n^{-1} H_n$ converges uniformly with $H_n : \hat{\Omega}_p \to \mathbb{R}^{n_p \times n_p}, p \longmapsto \left(\sum\limits_{i=1}^{n} (\frac{\partial^2}{\partial p_i \partial p_j} f(x_i, p))^2 \sigma_{x_i}^{-2} \right)_{i, j = 1, \dots, n_p}$.

A9. $\hat{M}(\hat{p})$ is invertible with $\hat{M} := \lim\limits_{n \to \infty} n^{-1} M_n$.

In this case, the Cramér–Rao bound of the weighted least squares estimator P_n is the inverse of the Fisher information matrix $M_n(\hat{p})$.

Under these assumptions, the asymptotic behavior of the weighted least squares estimator can be summarized by its convergence in distribution as follows:

$$\sqrt{n}(P_n - \hat{p}) \xrightarrow{d} \mathcal{N}(0, M_n(\hat{p})^{-1}n) \text{ as } n \to \infty \quad (2)$$

(see, e.g., Seber and Wild, 2003, chapter 12 and Walter and Pronzato, 1997, chapter 3).

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. 2), this can be approximated by the inverse of the information matrix $M_n(p_n)$, provided the matrix $M_n(p_n)$ is nonsingular, that is,

$$\text{cov}(P_n) \approx M_n(p_n)^{-1}. \quad (3)$$

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

Criteria $\phi : \mathbb{R}^{n_p \times n_p} \to \mathbb{R}^+ \cup \{\infty\}$, 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. 3) is used and $M_n(p_n)$ 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 $x'_1, \dots, x'_m \in \Omega_x$. 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

$$M_n(w, p_n) := M_n(p_n)$$
$$+ \sum_{i=1}^{m} w_i \frac{\nabla_p f(x'_i, p_n) \nabla_p f(x'_i, p_n)^T}{\sigma_{x'_i}^2}.$$

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

$$\underset{w \in \{0, 1\}^m}{\arg \min} \ \phi(M_n(w, p_n)^{-1}). \quad (4)$$

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

Potential constraints on the choice of the designs can be realized by constraints on the weight w. For example, the

number or the cost of the measurements can be limited by linear constraints on $\boldsymbol{w}$. These constraints have to be considered in the above optimization problem (Eq. 4).

The formulation (Eq. 4) is useful if additional experimental designs should be chosen from a finite number of experimental designs. Otherwise, the optimization problem can be reformulated so that the additional optimal design variables have to be optimized directly.

2.4　Calculation of optimal experimental designs

A straight-forward way to solve the optimization problem (Eq. 4) is to test all possible values of $\boldsymbol{w}$. This direct approach is only practical for small m.

For bigger m, the optimization problem (Eq. 4) is solved approximately. For this purpose, it is solved in the continuous rather than the discrete setting; that is, the constraint $\boldsymbol{w} \in \{0, 1\}^m$ is relaxed to $\boldsymbol{w} \in [0, 1]^m$. Accordingly, the problem

$$\underset{\boldsymbol{w} \in [0,1]^m}{\arg\min} \ \phi(M_n(\boldsymbol{w}, \boldsymbol{p}_n)^{-1}) \tag{5}$$

is solved.

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

After the continuous problem (Eq. 5) is solved, its solution is projected onto 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, for example, 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. 5) are already close to integer values. This behavior was also observed, for example, in Körkel (2002) and Körkel et al. (2004).

2.5　Robust optimal experimental designs

The information matrix M_n depends on the estimated parameters $\boldsymbol{p}_n$ if the parameters are nonlinear. This may lead to suboptimal designs if $\nabla_p f(\,\cdot\,, \boldsymbol{p}_n)$ 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 $\boldsymbol{p}_n$ as in Eq. (4), 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

$$G_n(\alpha) := \left\{ \boldsymbol{p} \in \mathbb{R}^{n_p} \mid \|\boldsymbol{p} - \boldsymbol{p}_n\|^2_{M_n(\boldsymbol{p}_n)^{-1}} \leq \gamma(\alpha) \right\}. \tag{6}$$

Here, $\gamma(\alpha)$ is the α quantile of the χ^2 distribution and $\|\boldsymbol{v}\|_A := \sqrt{\boldsymbol{v}^T A \boldsymbol{v}}$ denotes the energy norm of the vector $\boldsymbol{v} \in \mathbb{R}^{n_p}$ 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 $\boldsymbol{p}_n$ and the assumption $P_n \sim \mathcal{N}(\hat{\boldsymbol{p}}, M_n(\boldsymbol{p}_n)^{-1})$.

If the worst case quality in the entire region $G_n(\alpha)$ shall be optimized, the optimization problem (Eq. 4) becomes

$$\underset{\boldsymbol{w} \in \{0,1\}^m}{\arg\min} \ \underset{\boldsymbol{p} \in G_n(\alpha)}{\max} \ \phi(M_n(\boldsymbol{w}, \boldsymbol{p})^{-1}). \tag{7}$$

This minimum–maximum optimization problem can be solved only with considerably more computational effort compared to the optimization problem (Eq. 4). In order to reduce this effort, the function $\phi(M_n(\boldsymbol{w}, \,\cdot\,)^{-1})$ is linearized in point $\boldsymbol{p}_n$ in the following way:

$$\phi(M_n(\boldsymbol{w}, \boldsymbol{p})^{-1}) \approx$$
$$\phi(M_n(\boldsymbol{w}, \boldsymbol{p}_n)^{-1}) + \nabla_p(\phi(M_n(\boldsymbol{w}, \boldsymbol{p})^{-1}))^T (\boldsymbol{p} - \boldsymbol{p}_n).$$

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

$$\underset{\boldsymbol{p} \in G_n(\alpha)}{\max} \ \phi(M_n(\boldsymbol{w}, \boldsymbol{p}_n)^{-1}) + \nabla_p(\phi(M_n(\boldsymbol{w}, \boldsymbol{p})^{-1}))^T (\boldsymbol{p} - \boldsymbol{p}_n)$$

$$= \phi(M_n(\boldsymbol{w}, \boldsymbol{p}_n)^{-1}) + \gamma(\alpha)^{\frac{1}{2}} \|\nabla_p(\phi(M_n(\boldsymbol{w}, \boldsymbol{p}_n)^{-1}))\|_{M_n(\boldsymbol{p}_n)},$$

as can be seen, for example, in Körkel (2002). With this approach the optimization problem (Eq. 7) is replaced by

$$\underset{\boldsymbol{w} \in \{0,1\}^m}{\arg\min} \ \phi(M_n(\boldsymbol{w}, \boldsymbol{p}_n)^{-1})$$
$$+ \gamma(\alpha)^{\frac{1}{2}} \|\nabla_p(\phi(M_n(\boldsymbol{w}, \boldsymbol{p}_n)^{-1}))\|_{M_n(\boldsymbol{p}_n)}. \tag{8}$$

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. 8), the first and second derivatives of the model are used. In contrast, only the first derivative is used for local optimal designs (Eq. 4).

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 $\boldsymbol{w} \in \{0, 1\}^m$ regarding a criterion ϕ and with n previous measurements is defined as follows:

$$E_\phi(\boldsymbol{w}) := \underset{\hat{\boldsymbol{w}} \in \{0,1\}^m}{\min} \frac{\phi(M_n(\hat{\boldsymbol{w}}, \hat{\boldsymbol{p}})^{-1})}{\phi(M_n(\boldsymbol{w}, \hat{\boldsymbol{p}})^{-1})}. \tag{9}$$

```
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
```

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

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

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

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 (MathWorks, 2014) 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 a Git repository (Reimer, 2015) under the GNU General Public License (Foundation, 2007). It includes extensive commented source code, a detailed help integrated in MATLAB and a user manual.

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 are required.

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

The *model* interface prescribes how to provide these functions. They 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, for example, 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*. Listing 1 shows, for example, how a *model_explicit* object is created.

In the event that 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

```
model_object = model_ivp('−y+(t+1)*b', '[a,b]', 'y', 'a', 't', [1,10])
% 1. input: the right hand side of the differential equation
% 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
```

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

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

Listing 3. Set the model.

the parameter dependent initial value problem and calculates the necessary derivatives. Listing 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 MATLAB's *ode23s* function which can also solve stiff problems. Since the arising initial value problems for the derivatives are mutually independent, the solutions of the initial value problems can be calculated in parallel using the *Parallel Computing Toolbox*.

3.2 Setup of the solver

Methods for the optimization of model parameters and experimental designs are provided by the *solver* class. First, a *solver* object has to be created and the necessary information has to be passed.

On the one hand, this is the *model* which has to be set by the *set_model* method (see Listing 3).

On the other hand, an initial guess of the model parameters have to be set by the *set_initial_parameter_estimation* method (see Listing 4).

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. Furthermore, the measurement results themselves have to be passed for a parameter estimation (see Listing 5).

Finally, if an optimization of experimental designs shall be performed, the selectable measurements have to be set by the *set_selectable_measurements* method (see Listing 6). These measurements consist of the experimental designs as well as the variances of the measurement errors.

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

Listing 4. Set the initial parameter estimation.

```
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
```

Listing 5. Set accomplished measurements.

```
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
```

Listing 6. Set selectable measurements.

```
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
```

Listing 7. Optimize experimental designs.

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 Listing 7) or the *get_optimal_parameters* (see Listing 8) method, respectively. 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 approximatively.

For the approximative solving, the continuous problem is solved with the SQP algorithm (see Nocedal and Wright, 1999, Chapter 18) provided by the *fmincon* function of the *Optimization Toolbox*. Its solution is projected onto an integer solution by the second heuristic described in Sect. 2.4.

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 Broyden–Fletcher–Goldfarb–Shanno (BFGS) update (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970).

MATLAB's SQP algorithm can recover from infinity. If an infinite function value is reached during the optimization, the algorithm attempts to take a smaller step. Thus, if the optimization is started with a regular design, singular designs do not make any trouble.

The *get_optimal_parameters* method uses the trust-region-reflective (Coleman and Li, 1994, 1996) or the Levenberg–Marquard algorithm (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 *solver* object. Thus, for example, the increase in quality by adding or removing experimental designs can be determined.

3.4 Execution time and memory consumption

The total time required for the optimization of the model parameters or an experimental design depends crucially on the time required for evaluating the model function and its first and second derivative with respect to the model parameters.

When optimizing model parameters, the model function and its first derivative has to be evaluated several times with different model parameter vectors at the accomplished measuring points. When optimizing experimental designs, the model function and its first and second derivative has to be evaluated for one model parameter vector but at the accomplished and selectable measuring points.

Generally, the execution time increases with the number of parameters, the number of selectable measurements and the number of accomplished measurements.

The implementation of this toolbox favors a low execution time of a low memory consumption. For this reason, (intermediate) results within a method call and between successive method calls are saved and reused. An example is multiple occurring matrix multiplications within a method call. Another example 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 are reused.

Due to the described caching strategy, the total memory consumption depends linearly on the number of (accomplished and selectable) measurements and quadratically on the number of parameters. Nevertheless, it should be possible to solve problems with hundreds of parameters and thousands of measurements on a standard computer.

3.5 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 Listing 9). The desired options can be set using property-value pairs, as already known from MATLAB.

Estimation method: 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

```
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
```

Listing 8. Optimize model parameters.

```
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
```

Listing 9. Change an option.

presented in Sect. 2, are supported. The *region* estimation method is the default setting.

Confidence level: 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: It can be chosen whether a parameter optimization should be performed before optimizing experimental designs. This can 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 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: It can be chosen whether model parameter should be scaled before optimizing experimental designs or the model parameters themselves. Scaling means a uniform impact of all model parameters and is enabled by default. The options are *edo_scale_parameters* and *po_scale_parameters* with the values *yes* and *no*.

Optimization algorithm for experimental design: The exact and the approximative approach for the optimization of an experimental design problem can be chosen with the *edo_algorithm* option and the values *direct* and *local_sqp*. For time reasons, by default the experimental design problem is solved by the approximative approach. Furthermore, the number of function evaluations and iterations by the SQP algorithm can be constrained by the options *edo_max_fun_evals* and *edo_max_iter*.

Optimization algorithm for parameter estimation: The optimization algorithm for the parameter estimation problem can be chosen with the *po_algorithm* option. The trust-region-reflective (Coleman and Li,

1994, 1996) and the Levenberg–Marquard algorithm (Levenberg, 1944; Marquardt, 1963; Moré, 1977) can be chosen with the values *trust-region-reflective* and *Levenberg–Marquardt* The trust-region-reflective algorithm is the default algorithm. Furthermore, the number of function evaluations and iterations can be limited through the options *po_max_fun_evals* and *po_max_iter*.

3.6 Help and documentation

The *Optimal Experimental Design Toolbox* also provides extensive integrated help. Besides system requirements and version information, a user's guide with step-by-step instructions on how to optimize experimental designs and model parameters is included. Demos show how to work with the toolbox in practice. In addition, a detailed description for every class and method is available.

The layout of the help for 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 reoriented with a new layout.

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 of different complexity. Both models describe the sediment concentration in seawater during tidal inundation of 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 role of dissipating current and wave energy and therefore reducing erosional forces at dikes and coastal areas.

With these models, the vertical accretion of coastal salt marshes can be predicted. When considering expected global sea level rise (IPCC, 2013), the future ability of coastal salt marshes to adapt to rising sea levels and thus to survive can be estimated. Depending on these estimates, measures to protect these salt marshes can be taken.

Calibration of the model parameters requires measurements of suspended sediment concentration during tidal inundation, which are time-consuming and laborious. For this reason, it is advantageous to know under which conditions and how many of these measurements should be carried out.

4.1 The models

Both models are zero-dimensional point models, which describe the sediment concentration in seawater during tidal inundation of coastal salt marshes. The first model (C_2-model) has two model parameters, was described in Temmerman et al. (2003) and was adapted for a salt marsh in the Wadden

Sea (southeastern North Sea), located near Hoernum in the southern part of the island of Sylt (Germany), by Schuerch et al. (2013). The second model (C_3-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 in $kg\,m^{-3}$ is modeled by the function $C : [t_S, t_E] \to \mathbb{R}^+$. 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 implicitly as the 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 $C(t_S) = C_0$. (10)

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

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

describes the time-dependent water surface elevation and E the elevation of the marsh both in meters and 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 in meters. The start t_S and end time t_E of the inundation are the points where the height h equals the elevation of the marsh E.

The sediment 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. (10). During the flood phase, the reduced sediment 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. (10).

The values used in the water surface elevation function h, for the local salt marsh, are shown in Table 1. These have been estimated by nonlinear regression analysis using local historic tide gauge data from 1999 to 2009 (at Hoernum Hafen, Germany). The continuous high-resolution (6 min) time series has, therefore, been split into the individual tidal cycles beforehand (Schuerch et al., 2013).

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	1.3m

Table 2. Values for the C_2-model.

	C_0 [$kg\,m^{-3}$]	w_S [$m\,s^{-1}$]
reference value	0.1	10^{-5}
typical range	0.01–0.2	4×10^{-6}–4×10^{-4}
start value	5	2×10^{-7}
optimization bound	10^{-4}–10^4	10^{-8}–1

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. Reference values derived from literature values and typical ranges can be found in Table 2 (see Bartholdy and Aagaard, 2001, for C_0 and Temmerman et al., 2003, for C_0 and w_S).

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. Reference values derived from literature values and typical ranges (where available) can be found in Table 3 (see van Leussen, 1999, and Pejrup and Mikkelsen, 2010, for the settling index s and Temmerman et al., 2004, for k).

In this model, 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 (Temmerman et al., 2003; Schuerch et al., 2013). Additionally, a relationship between the initial sediment concentration and the settling velocity is assumed (Krone, 1987). This is an empirical approximation of the so-called flocculation process (Burt, 1986).

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.

Table 3. Values for the C_3-model.

	k	r	s
reference value	0.25	10^{-5}	0.5
typical range	0.04–0.2		0.5–3.5
start value	12.5	2×10^{-7}	3
optimization bound	10^{-4}–10^4	10^{-8}–1	10^{-1}–5

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 for the calculation of 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 reference parameters in Tables 2 and 3 plus an additive normally distributed measurement error with zero expectation as artificial measurement results. As standard deviation of the measurement error, we chose 10^{-2} once and 10^{-1} once.

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 10 times and averaged the results to minimize the influence of randomness.

For the parameter estimation, the start values and bounds in Tables 2 and 3 were used. The bounds were chosen so that the typical range of values is covered, but also more extreme values are possible. The starting values were chosen slightly outside the typical ranges to represent a poor initial guess.

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 10 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. In the robust approach, a confidence level of 95 % was used. The optimization problems for the experimental designs were once solved exactly and once approximatively (see Sect. 2.4). To evaluate all these methods, we compared the resulting parameter estimations with the reference model parameters.

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 and five measurements. Altogether 50 measurements were simulated at each experiment with the C_2-model. For the C_3-model, 150 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.

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. 1 and 2.

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 independent of the choice of the experimental designs. The accuracy improved faster the more frequently the experimental designs and parameters were optimized. However, the best achieved accuracy was independent of the frequency.

With uniformly distributed experimental designs the best achieved accuracy was slightly worse than with optimized experimental designs. Four to six more measurements were needed compared to optimized experimental designs in order to achieve their accuracy.

Although the parameters occur nonlinearly 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.

The approximate solving of the discrete optimization problem has resulted in slightly worse accuracy at the first iterations compared to the exact solving. Thereafter, the difference was very small. The solutions of the relaxed continuous optimization problems were almost always nearly integer.

The different standard deviations of the measurement errors only influenced the best achieved accuracy which was of course worse at a higher standard deviation. This can be explained by the fact that different constant standard deviations only mean a different scaling of the objective of the experimental design optimization problem. Thus, different constant standard deviations do not affect its solution.

4.3.2 Results for the C_3-model

After 10–25 measurements, the accuracy of the parameter estimations for the C_3-model with optimized experimental designs only improved slightly. Again, the fewer measurements performed per iteration the faster the accuracy improved, and the best achieved accuracy was independent of the number of measurements per iteration.

With uniformly distributed experimental designs, the best accuracy was achieved after 24–60 measurements. Further-
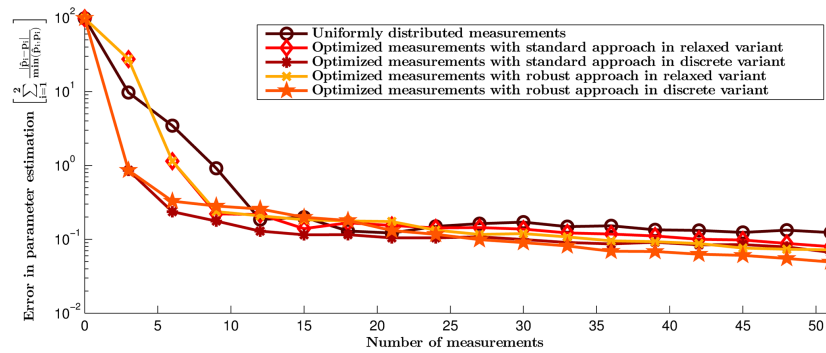
Figure 1. Averaged error in the parameter estimation from 10 optimization runs with the C_2-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.
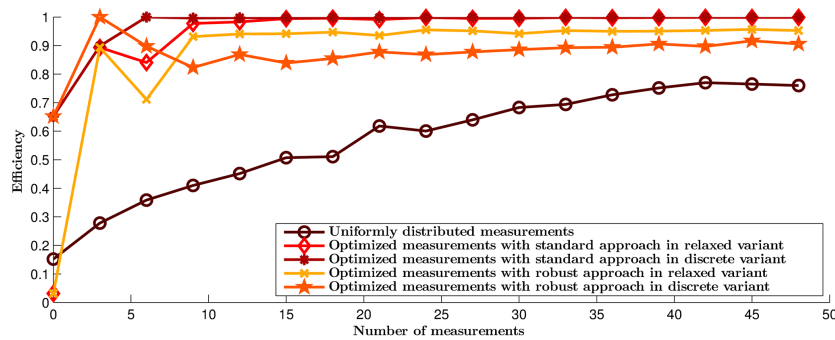
Figure 2. Averaged error in the parameter estimation from 10 optimization runs with the C_3-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

more, the best achieved accuracy was worse by about a factor of 10 compared to the best 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.

For both approaches, the difference between the accuracy achieved with the exact solutions of the discrete optimization problem and the accuracy achieved with the approximate solutions was small but recognizable and almost constant over the iterations. Also in these experiments, the solutions of the relaxed continuous optimization problems were almost all nearly integer.

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

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 experimen-

tal designs. In the more complex model, the difference was even bigger.

The robust approach did not achieved higher accuracy compared to the standard approach. In the complex model, the robust approach was even slightly less accurate. This may indicate that the gain in accuracy by taking into account the nonlinearity is offset by the additional approximations in the robust approach. Since a considerably higher computational effort is associated with the robust approach, the standard approach should be preferred, at least for these models.

The exact solutions of the discrete optimization problems yielded only slightly better accuracy gains compared to its approximate solutions. The fact that the approximate solutions were almost all nearly integer also argues for the approximate solving. This circumstance was also observed in Körkel (2002) and Körkel et al. (2004). For these reasons and because the exact solving requires much more computational effort, the approximate solving should be preferred, 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. 3 and 4.
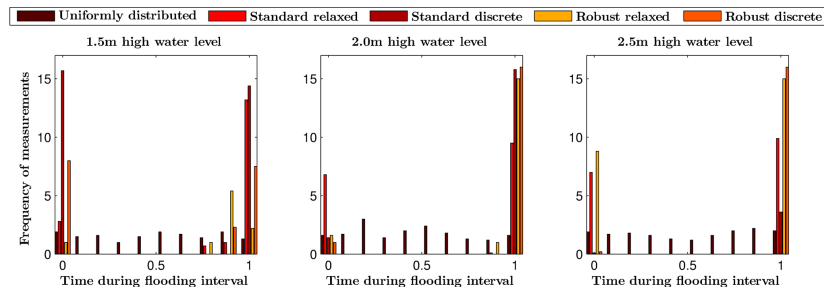
Figure 3. Averaged efficiency for the experimental designs from 10 optimization runs with the C_2-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.
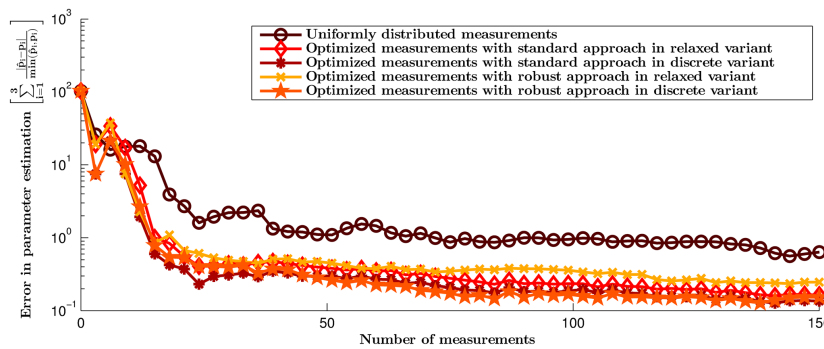
Figure 4. Averaged efficiency for the experimental designs from 10 optimization runs with the C_3-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

The results emphasized the already seen importance of the optimization of the experimental designs. In particular, the advantage in the case of the 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. 5 and 6.

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 exact variant favored lower high water levels unlike both approaches in the approximate 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 re-
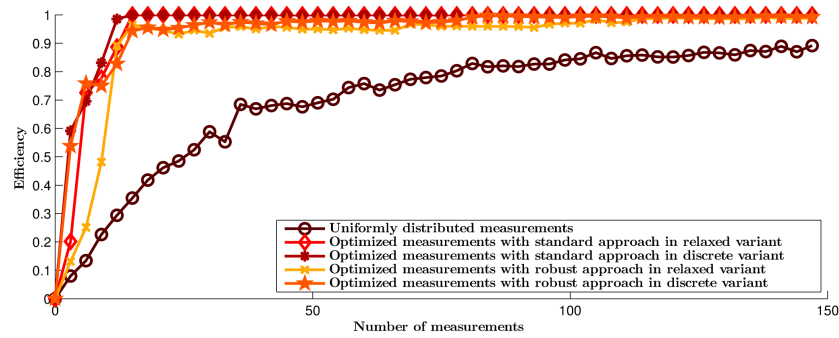
Figure 5. Averaged frequency of measurements from 10 optimization runs with the C_2-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.
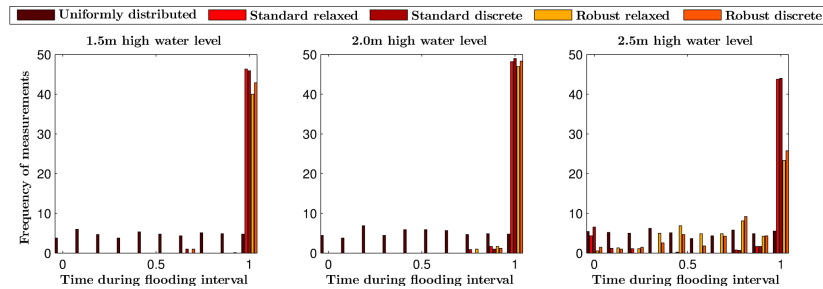
Figure 6. Averaged frequency of measurements from 10 optimization runs with the C_3-model and three measurement per iteration with standard deviation 10^{-2} of the measurement error.

spect 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.

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.

By employing the presented approach for two existing salt marsh models, we showed that model parameters can be determined much more accurately if the corresponding measurement conditions were optimized. In particular for time-consuming or costly measurements, it is useful to optimize the measurement conditions with the *Optimal Experimental Design Toolbox*.

This gain in accuracy is not limited to the application examples. In general, using the implemented methods, the accuracy of the parameters of any model can be maximized while minimizing the measurement cost, provided that the related assumptions are fulfilled. However, the required execution time for the optimization increases with the number of model parameters and (accomplished and selectable) measurements. Parallelization techniques in the optimization as well as in the model evaluation itself can counteract this effect.

In addition to the parallelization, the optimization in the toolbox could also be extended to techniques of globalization, so that the chance of getting into a local minimum is reduced.

The results concerning the application examples have not significantly differed despite the various approaches for optimizing experimental design. For this reason, the approach with the least computational effort is recommended. However, this recommendation can not be applied readily to other

(more complex) models. Here, the performance of the approaches should be compared again if possible.

Furthermore, it has been found that measurements at the beginning and end of the inundation period are particularly important for the application examples. The high water level of the inundation seemed to play a minor role. These results, however, can not be applied easily to other models. Usually, a separate optimization of experimental design makes sense here.

Code availability

The *Optimal Experimental Design Toolbox* is available under the GNU General Public License (Foundation, 2007) at a Git repository (Reimer, 2015). In addition to the toolbox, including commented source code and a user manual, an implementation of the application examples is also available.

Edited by: J. Neal

References

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

Bartholdy, J. and Aagaard, T.: Storm surge effects on a back-barrier tidal flat of the Danish Wadden Sea, Geo-Mar. Lett., 20, 133–141, doi:10.1007/s003670000048, 2001.

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.

Burt, T.: Field Settling Velocities of Estuary Muds, in: Estuarine Cohesive Sediment Dynamics, edited by: Mehta, A., Vol. 14 of Lecture Notes on Coastal and Estuarine Studies, 126–150, Springer New York, doi:10.1007/978-1-4612-4936-8_7, 1986.

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.

Coleman, T. F. and Li, Y.: An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds, SIAM J. Optimiz., 6, 418–445, 1996.

Cramér, H.: Mathematical Methods of Statistics, Almqvist & Wiksells Akademiska Handböcker, Princeton University Press, 1946.

El-Monsef, M. M. E. A., Rady, E. A., and Seyam, M. M.: Relationships among Several Optimality Criteria, Interstat, 2009.

Fletcher, R.: A new approach to variable metric algorithms, Comput. J., 13, 317–322, doi:10.1093/comjnl/13.3.317, 1970.

Foundation, F. S.: GNU General Public License, available at: http://www.gnu.org/licenses/gpl-3.0-standalone.html (last access: 30 July 2014), 2007.

Goldfarb, D.: A family of variable-metric methods derived by variational means, Math. Comput., 24, 23–26, 1970.

Guest, T. and Curtis, A.: Iteratively constructive sequential design of experiments and surveys with nonlinear parameter-data relationships, J. Geophys. Res.-Sol., 114, B04307, doi:10.1029/2008JB005948, 2009.

IPCC: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by: Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S. K., Boschung, J., Nauels, A., Xia, Y., Bex, B., and Midgley, P. M., Cambridge University Press, Cambridge, United Kingdom, New York, NY, USA, doi:10.1017/CBO9781107415324, 2013.

Jennrich, R. I.: Asymptotic Properties of Non-Linear Least Squares Estimators, Ann. Math. Stat., 40, 633–643, doi:10.1214/aoms/1177697731, 1969.

Krone, R. B.: A method for simulating historic marsh elevations, in: Coastal Sediments (1987), pp. 316–323, ASCE, 1987.

Körkel, S.: Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen, PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2002.

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.

Levenberg, K.: A method for the solution of certain problems in least squares, Q. Appl. Math., 2, 164–168, 1944.

Malinvaud, E.: The Consistency of Nonlinear Regressions, Ann. Math. Stat., 41, 956–969, 1970.

Marquardt, D. W.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters, J. Soc. Ind. Appl. Math., 11, 431–441, 1963.

MathWorks: Matlab R2014a Primer, Natick, Massachusetts, release 2014a Edn., available at: http://www.mathworks.de/help/releases/R2014a/pdf_doc/matlab/getstart.pdf (last access: 12 March 2015), 2014.

Moré, J.: The Levenberg-Marquardt algorithm: Implementation and theory, in: Numerical Analysis, edited by: Watson, G., Vol. 630 of Lecture Notes in Mathematics, 105–116, Springer Berlin Heidelberg, doi:10.1007/BFb0067700, 1977.

Nocedal, J. and Wright, S.: Numerical Optimization, Springer series in operations research and financial engineering, Springer, New York, 1999.

Pejrup, M. and Mikkelsen, O. A.: Factors controlling the field settling velocity of cohesive sediment in estuaries, Estuar. Coast. Shelf Sci., 87, 177–185, doi:10.1016/j.ecss.2009.09.028, 2010.

Rao, R. C.: Information and the accuracy attainable in the estimation of statistical parameters, Bull. Calcutta Math. Soc., 37, 81–91, 1945.

Reimer, J.: oed: Optimal Experimental Design Toolbox: Version 1.0, doi:10.5281/zenodo.16004, 2015.

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.-Earth, 118, 84–96, doi:10.1029/2012JF002471, 2013.

Seber, G. A. F. and Wild, C. J.: Nonlinear regression, Wiley series in probability and statistics, Wiley-Interscience, 2003.

Shanno, D. F.: Conditioning of Quasi-Newton Methods for Function Minimization, Math. Comput., 24, 647–656, 1970.

Stigler, S. M.: Gauss and the Invention of Least Squares, Ann. Stat., 9, 465–474, 1981.

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.

Temmerman, S., Govers, G., Wartel, S., and Meire, P.: Modelling estuarine variations in tidal marsh sedimentation: response to changing sea level and suspended sediment concentrations, Mar. Geol., 212, 1–19, doi:10.1016/j.margeo.2004.10.021, 2004.

van Leussen, W.: The variability of settling velocities of suspended fine-grained sediment in the Ems estuary, J. Sea Res., 41, 109–118, doi:10.1016/S1385-1101(98)00046-X, 1999.

Walter, É. and Pronzato, L.: Identification of parametric models from experimental data, Communications and control engineering, Springer, New York, 1997.

White, H.: Nonlinear Regression on Cross-Section Data, Econometrica, 48, 721–746, 1980.

White, H. and Domowitz, I.: Nonlinear Regression with Dependent Observations, Econometrica, 52, 143–162, 1984.

Wu, C.-F.: Asymptotic Theory of Nonlinear Least Squares Estimation, Ann. Stat., 9, 501–513, 1981.