INSTITUT FÜR INFORMATIK

Surrogate-Based Optimization of Climate Model Parameters Using Response Correction

M. Prieß, S. Koziel, T. Slawig

Bericht Nr. 1104
March 30, 2011

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

ZU KIEL

Institut für Informatik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Surrogate-Based Optimization of Climate Model Parameters Using Response Correction

M. Prieß, S. Koziel, T. Slawig

Bericht Nr. 1104
March 30, 2011

e-mail: mpr@informatik.uni-kiel.de, koziel@ru.is,
ts@informatik.uni-kiel.de

Surrogate-Based Optimization of Climate Model Parameters Using Response Correction

M. Prieß[a,1,*], S. Koziel[b], T. Slawig[a]

[a] *Institute for Computer Science, Cluster The Future Ocean, Christian-Albrechts Universität zu Kiel, 24098 Kiel, Germany*
[b] *Engineering Optimization & Modeling Center, School of Science and Engineering, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland*

Abstract

We present a computationally efficient methodology for the optimization of climate model parameters applied to a (one-dimensional) representative of a class of marine ecosystem models. We use a response correction technique to create a surrogate from a temporarily coarser discretized physics-based low-fidelity model. We demonstrate that replacing the direct parameter optimization of the high-fidelity ecosystem model by iteratively updating and re-optimizing the surrogate leads to a very satisfactory solution while yielding significant cost saving - about 84% when compared to the direct high-fidelity model optimization.

Keywords: Climate models, marine ecosystem models, surrogate-based optimization, parameter optimization, response correction

1. Introduction

In this paper we present the application of a *Surrogate-based Optimization* approach, based on a multiplicative response correction, on parameter identification problems in a climate model.

Surrogate-based optimization [1–4] is a methodology to efficiently optimize complex, so-called *high-fidelity* models, that require substantial computational effort already for a model evaluation. High-fidelity models are typically evaluated through computer simulation and evaluation times of several hours, days or even weeks are not uncommon. As a consequence, optimization and control problems for such models are often still beyond the capability of modern numerical algorithms and computer power. The idea of surrogate-based optimization is to replace the high-fidelity in focus by a computationally cheaper and yet reasonably accurate representation, so-called surrogate. The surrogate can be created by approximating sampled high-fidelity model data or by employing a physically-based *low-fidelity* or coarse model. In this work, we use the latter approach. The coarse model is normally less accurate, therefore, it has to be iteratively corrected by suitable methods. The correction (or alignment) can be realized using a limited number (in many cases, only one)

*Corresponding author (*phone:* +49-(0)431 880 7452, *fax:* +49-(0)431 880 7618)
Email addresses: mpr@informatik.uni-kiel.de (M. Prieß), koziel@ru.is (S. Koziel), ts@informatik.uni-kiel.de (T. Slawig)
URL: http://www.informatik.uni-kiel.de/co2/mitarbeiterinnen/dipl-phys-malte-priess/ (M. Prieß), http://koziel.ru.is (S. Koziel), http://www.informatik.uni-kiel.de/co2/mitarbeiterinnen/prof-dr-thomas-slawig/ (T. Slawig)
[1]Research supported by DFG Cluster The Future Ocean

evaluations of the high-fidelity model and possibly also its derivatives. Surrogate-based optimization is widely and very successfully used in engineering sciences, compare [1–4]. The application on parameter optimization in climate models is rather new.

Climate models are typically given as time-dependent partial differential or differential algebraic equations (PDEs/DAEs) [5–7]. Since the number of processes that have to be included and the needed temporal and spatial resolution is quite high, so is the computational effort. As a result, many processes on small temporal or spatial scales are, as denoted in the climate community, *parameterized*, i.e., they are represented by simpler models that usually include a number of parameters that have to be properly chosen or adjusted. A typical example – not only used in climate models for ocean or atmosphere simulations – is turbulence modeling [8]. There are also processes in the climate system where even without much simplification several quantities or parameters are unknown or very difficult to measure. This is for example the case for growth and dying rates in marine ecosystem models [9, 10], one of which is taken as a test case for the surrogate-based optimization approach we analyze in this paper. Marine ecosystem models describe photosynthesis and other biogeochemical processes in the marine ecosystem that are important, e.g., to compute and predict the oceanic uptake of carbon dioxide (CO_2) as part of the global carbon cycle [9].

The aim of parameter optimization is to adjust or identify the model parameters so that the model output fits given measurement data [11]. The mathematical task thus can be classified as a least-squares type optimization or inverse problem [12]. The number of optimization parameters range from about 10 to 100 discrete real-valued ones in marine ecosystem models (where they are growth and dying rates etc.) up to distributed functions (or thousands and more discrete values after discretization), for example when an initial model state or boundary condition is unknown and target of the optimization. The optimization parameters and the model state are coupled by the constraint of the time-dependent PDE, i.e., the climate model. Additionally, constraints on the parameters (e.g., non-negativity of growth-rates in ecosystem models etc.) and on the state variables (non-negativity of concentrations of biological species as algae etc. or of temperature) might be given.

This optimization process requires a substantial number of (typically expensive) function and optionally sensitivity or gradient or even Hessian matrix evaluations. If the latter are computed by finite difference approximations, the critical quantity determining the computational effort of the optimization is that of the cost function evaluation, which is basically a single model simulation. Hence, decreasing the effort related to the function evaluations (or, equivalently, cutting down the number of function calls necessary to find the optimum) is of primary importance to reduce the overall optimization cost. This becomes particularly significant for computationally expensive three-dimensional coupled models, as for example global climate models [7].

In this paper we analyze the application of a *multiplicative response correction technique* to create a surrogate for one specific type of a climate model, a one-dimensional marine ecosystem model that uses pre-computed ocean circulation data [13]. This model was chosen because here extensive optimization runs

with different methods including local, gradient-based and so-called global, genetic algorithms have been performed, see [14]. The underlying physically-based low-fidelity model is obtained from a temporarily coarser discretization of the high-fidelity one. We verify our approach by using synthetic target data and by comparing the results of surrogate-based optimization to those obtained from the direct fine model optimization. The application on real data is performed as a next step. Furthermore, this exemplary application shall serve as a test for three-dimensional model runs, which are much more costly with respect to computing time.

The structure of the paper is as follows: The general form of climate models and the parameter optimization problem considered is described in Section 2. We point out that the mathematical formulation of the climate models we use is quite general, such that our approach is not limited to them but remains applicable for a wide range of time-dependent models. We first recall the basic idea of surrogate-based optimization in Section 3. The ecosystem model, which is taken as an example in this paper, is introduced in Section 4, and its low-fidelity counterpart that we use as a basis for the surrogate is described in Section 5. The response correction, the construction of the surrogate model and the quality of the surrogate are described and analyzed in Section 6. The setup of the optimization which is used to compare the results is given in Section 7. Numerical results and discussion of an exemplary test run are provided in Section 8. Section 9 concludes the paper with a summary and an outlook.

2. Model Equations and Optimization Problem

In this section we give the formulations of what we call a *model* and of the corresponding parameter optimization problem. Our formulations are quite general and appropriate for a big class of applications, for which climate models are only one example.

2.1. Continuous and discrete Model Formulation

We start from an initial boundary value problem (IBVP) for a system of time-dependent partial differential or differential algebraic equations (PDEs/DAEs) of the following form:

$$\left.\begin{aligned} E\,\frac{\partial y}{\partial t} &= f(y,u) && \text{in } \Omega\times(0,T)\\ y(x,0) &= y_{init}(x) && \text{in } \Omega\\ y(x,t) &= y_{bdr}(x,t) && \text{on } \partial\Omega\times(0,T). \end{aligned}\right\} \tag{1}$$

Here y is the vector of the *state variables*, and E is a matrix with the size of y, typically being the identity matrix for a PDE while having rank deficiency for a DAE [15]. We include DAEs in this formulation since in climate models, e.g., ocean circulation models, the Navier-Stokes equations [16] are an important part, and – after space discretization – take the form of a DAE system. Then y may for example consist of velocity field, pressure, temperature and salinity. In our example of a marine ecosystem model (which is formulated as PDE system), the matrix E can be set to the identity and thus omitted. In this case the state vector y contains all relevant biogeochemical tracers as phyto- and zooplankton etc., see Section 4 for the details.

The spatial domain Ω is an open subset of $\mathbb{R}^d, d \in \{1,2,3\}$, with boundary $\partial\Omega$. We assume that the initial time is $t = 0$, and define the end of the considered time interval as $T > 0$. The right-hand side of the system is given by a vector-valued and usually nonlinear function f which includes spatial differential operators. In climate models, it often additionally depends explicitly on the space and time variables x and t, respectively, which is skipped in the notation. Moreover f depends on a number of model parameters which are summarized in the vector u. The vector-valued functions y_{init} and y_{bdr} are given initial and boundary values. They can also be subject to optimization, but for simplicity of notation and since in our example this is not the case, we do not formulate (1) this way. The IBVP given by (1) can be called a *continuous model*, since its solution – if it exists – is mathematically given in an appropriate function space Y.

After spatial and temporal discretization, we obtain a discretized or *discrete model*. We will consider here a time discretization which is performed by a sequential integration at time steps $0 = t_0 < \ldots < t_j < \ldots < t_M = T$. In many applications (among them climate models), this integration is at least partially implicit, i.e., it involves the solution of linear or non-linear systems. With respect to the spatial discretization, we are quite general, i.e., it may be either finite volumes or differences (as in most climate models) or finite elements etc. We assume that we have K discrete values for the discrete approximation $\mathbf{y}_j$ of the state y at time t_j, i.e., $\mathbf{y}_j \in \mathbb{R}^K$ and $\mathbf{y}_j \approx y(\cdot, t_j)$. Thus, after discretization we end up with the following discrete model:

$$\left.\begin{array}{rcl} \mathbf{y}_0 & \approx & y_{init}, \\ \mathbf{y}_{j+1} & = & \Phi_j(\mathbf{y}_j, \mathbf{u}), \quad j = 1, \ldots, M-1. \end{array}\right\} \tag{2}$$

Again, only for simplicity of notation, we do not include multi-step time integration schemes here, but they can be easily treated similarly. The first line in (2) indicates that already the discrete initial value usually is only an approximation (or restriction) of the exact initial data y_{init} given in the continuous model (1). We use the boldface notation for discrete vectors, e.g., here for $\mathbf{y}_j \in \mathbb{R}^I$. The whole discrete state $\mathbf{y} := (\mathbf{y}_j)_{j=1,\ldots,M}$ then is in $\mathbb{R}^{MI}$. Note that I not necessarily equals the number of points in the spatial grid, but is often a multiple of it since we have a system of equations in (1) and y consists of several components. We moreover introduced a discrete vector $\mathbf{u} \in \mathbb{R}^n$ of parameters, either after their discretization, or (as in our model example), since they have been real numbers, from the beginning. Here we assume that the parameters do not depend on the time t_j and thus have no index j. The right-hand sides Φ_j in (2) depend on f, the actual time t_j (since f depends on it), the time discretization scheme and its step size.

If the discretization schemes in space and time are chosen properly, we can regard the discrete model as a mapping

$$\mathbf{u} \quad \mapsto \quad \mathbf{y}(\mathbf{u})$$

from the parameters to the discrete space state, i.e., from $\mathbb{R}^n =: U$ to $\mathbb{R}^{MI} =: Y$.

2.2. Optimization Problem

In this subsection we formulate the optimization problem for the discrete model. Omitting the boldface notation, the same formulation holds for the continuous model, but naturally would require further analysis, which is beyond the scope of this paper.

The key task in parameter optimization is to minimize a least-squares type cost function measuring the misfit between the discrete model output $\mathbf{y} = \mathbf{y}(\mathbf{u})$, i.e., the solution of (2), and given observational data $\mathbf{y}_d$ [11, 12]. We assume that $\mathbf{y}_d \in Y$, otherwise an appropriate observation/restriction operator has to be introduced. In most cases, the cost function is constrained by parameter bounds. Thus the parameter optimization problem can be written as

$$\min_{\mathbf{u} \in U_{ad}} J(\,\mathbf{y}(\mathbf{u})\,) \tag{3}$$

where

$$J(\,\mathbf{y}\,) \;:=\; \frac{1}{2}\,||\,\mathbf{y} - \mathbf{y}_d\,||_Y^2 \quad , \quad U_{ad} := \{\mathbf{u} \in \mathbb{R}^n : \mathbf{b}_l \leq \mathbf{u} \leq \mathbf{b}_u\} \quad , \qquad \mathbf{b}_l, \mathbf{b}_u \in \mathbb{R}^n \quad , \quad \mathbf{b}_l < \mathbf{b}_u.$$

The inequalities in the definition of the set U_{ad} of admissible parameters are meant component-wise. The functional J may additionally include a regularization term for the parameters, which was not necessary in our case.

Additional constraints on the state variable $\mathbf{y}$ might be necessary, e.g., to ensure non-negativity of the temperature or of the concentrations of biogeochemical quantities. In our example model however, by using appropriate parameter bounds $\mathbf{b}_l$ and $\mathbf{b}_u$, non-negativity of the state variables can be ensured. This was already observed and used in [14].

3. Surrogate-Based Optimization

For many nonlinear optimization problems, a high computational cost of evaluating the objective function and its sensitivity, and, in some cases, the lack of sensitivity information, is a major bottleneck. The need for decreasing the computational cost of the optimization process is especially important while handling complex three-dimensional models.

Surrogate-based optimization [1–4] addresses these issues by replacing the original high-fidelity model $\mathbf{y}$ by its surrogate model $\mathbf{s}$. The surrogate should be computationally cheap and analytical tractable. It can be obtained by approximating the sampled high-fidelity model data using a suitable technique, e.g., polynomial regression [1], kriging [17] or support-vector regression [18].

Another possibility, explored in this paper, is to construct the surrogate through correction of a coarse or low-fidelity model, a less accurate but computationally cheap representation of $\mathbf{y}$. The surrogate model is updated at each iteration of the optimization algorithm, typically using available high-fidelity model data. In particular, the surrogate model $\mathbf{s}_k$ at iteration k can be constructed by only using the high-fidelity model output $\mathbf{y}(\mathbf{u}_k)$ at the current optimization variable vector $\mathbf{u}_k$ and the corresponding low-fidelity model output.

The low-fidelity model correction aims at reducing misalignment between the low- and high-fidelity models. The specific correction technique exploited in this work is described in detail in Section 6. The next iterate, $\mathbf{u}_{k+1}$, is obtained by optimizing the surrogate $\mathbf{s}_k$, i.e.,

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}\in U_{ad}} J(\,\mathbf{s}_k(\mathbf{u})\,). \tag{4}$$

Then the updated surrogate $\mathbf{s}_{k+1}$ is determined by re-aligning the low-fidelity model at $\mathbf{u}_{k+1}$ and optimized again as in (4). The process of aligning the coarse model to obtain the surrogate and subsequent optimization of this surrogate is repeated until a user-defined termination condition is satisfied, which can use certain convergence criteria, assumed level of cost function value or on a specific number of iterations (particularly if the computational budget of the optimization process is limited). A discussion of termination condition used in this work can be found in Section 8.

A well performing surrogate-based algorithm is capable of yielding a satisfactory solution at a low computational cost, typically corresponding to only a few evaluations of the high-fidelity model. The key prerequisites to ensure this are a cheap and yet reasonably accurate coarse model as well as a properly selected and low-cost alignment procedure (i.e., using a limited number of high-fidelity model evaluations, preferably just one).

If the surrogate $\mathbf{s}_k$ satisfies so-called 0-order and 1st-order consistency conditions [19, 20] with the high-fidelity model at $\mathbf{u}_k$, i.e.,

$$\mathbf{s}_k(\mathbf{u}_k) = \mathbf{y}(\mathbf{u}_k) \quad , \quad \mathbf{s}_k'(\mathbf{u}_k) = \mathbf{y}'(\mathbf{u}_k), \tag{5}$$

the surrogate-based scheme (4) is provable convergent to at least a local optimum of (3), provided that both the low- and high-fidelity models are sufficiently smooth, and the surrogate optimization step is enhanced by the trust-region (TR) safeguard [19, 20], i.e.,

$$\mathbf{u}_{k+1} = \operatorname*{argmin}_{\mathbf{u}\in U_{ad}, ||\mathbf{u}-\mathbf{u}_k||\leq\delta_k} J(\,\mathbf{s}_k(\mathbf{u})\,),$$

with δ_k being the trust-region radius updated according to the TR rules.

Note that the 1st-order consistency requires high-fidelity sensitivity data, which is not utilized here. In this work, the surrogate is defined to satisfy the 0-order consistency only which is sufficient to ensure good performance as demonstrated in Subsection 6.3 and Section 8.

4. Example: A Marine Ecosystem Model

In this section we briefly describe the model – both in continuous and discrete form – that we used to analyze the application of a surrogate-based optimization approach on a climate model. The considered example is a one-dimensional marine ecosystem model driven by pre-computed ocean circulation data [13].

4.1. The Continuous Model

Simulating the marine ecosystem has become a key tool for understanding the ocean carbon cycle and its variability. The marine ecosystem contains several biogeochemical quantities (called *tracers*), for example nutrients, phyto- and zooplankton which interact and are moreover transported by the ocean circulation and influenced by temperature and salinity. Thus ecosystem simulations require modeling and computation both of ocean circulation and biogeochemistry. The underlying continuous models are governed by coupled systems of nonlinear, parabolic PDEs or DAEs, for ocean circulation (ocean models, i.e., Navier-Stokes equations with additional temperature and salinity transport equations) and transport of biogeochemical tracers (marine ecosystem models, i.e., convection- or advection-diffusion-reaction type equations) [9]. Thus they fit in our general formulation (1) and its discrete counterpart (2).

In ecosystem models, the parameters to be optimized – summarized in the vector $\mathbf{u}$ in (2) – are for example growth and dying rates of the tracers and thus appear in the usually nonlinear coupling or interaction terms in the model.

Our example ecosystem model was developed by Oschlies and Garcon [13] and simulates the interaction of dissolved inorganic nitrogen, phytoplankton, zooplankton and detritus (thus also called $NPZD$ model). One aim was to reproduce observations $\mathbf{y}_d$ at different North Atlantic locations by the optimization of model parameters within credible limits. Figure 4.1 shows the model output and target data, respectively, as illustration for the tracer detritus for a certain depth and a part of the time interval. The model uses pre-computed ocean circulation and temperature data from an ocean model (in a sometimes called *off-line modus*), i.e., no feedback by the biogeochemistry on the circulation and temperature is modeled [13]. Thus the continuous model (1) here just contains the biochemistry, whereas all circulation data are hidden in the right-hand side f.

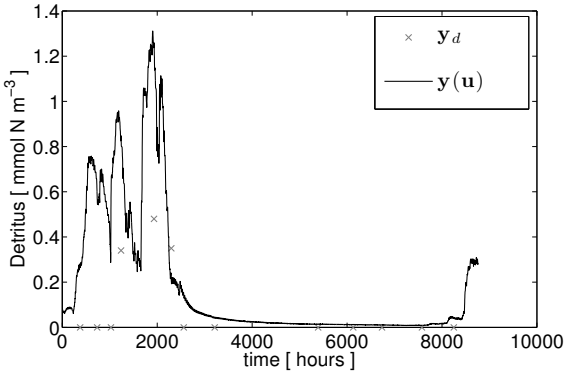

Figure 1: Model output $\mathbf{y}^{(D)}$ (detritus) and observation data $\mathbf{y}_d^{(D)}$ for one year at depth $z \simeq -25\,\mathrm{m}$.

As a test case and since biogeochemistry – except for sinking processes – mainly happens locally in space, we use here a one-dimensional version of the model. This version simulates one water column at a given horizontal position. This is additionally motivated by the fact that there have been special time series studies at fixed locations. Clearly the computational effort in a one-dimensional simulation is significantly smaller than in the three-dimensional case. Thus, before going to 3-D, this model serves as a good test example for the applicability of surrogate-based optimization approaches, since it includes all significant features of ecosystem models.

In the $NPZD$ model, the concentrations (in mmol N m^{-3}) of dissolved inorganic nitrogen N, phytoplankton P, zooplankton Z, and detritus (i.e., dead material) D are summarized in the vector $y = (y^{(l)})_{l=N,P,Z,D}$

and described by the following coupled PDE system

$$\left.\begin{aligned} \frac{\partial y^{(l)}}{\partial t} &= \frac{\partial}{\partial z}\left(\kappa \frac{\partial y^{(l)}}{\partial z}\right) + Q^{(l)}(y, u_2, \ldots, u_n), && l = N, P, Z \\ \frac{\partial y^{(D)}}{\partial t} &= \frac{\partial}{\partial z}\left(\kappa \frac{\partial y^{(D)}}{\partial z}\right) + Q^{(D)}(y, u_2, \ldots, u_n) - \frac{\partial y^{(D)}}{\partial z} u_1, && l = D \end{aligned}\right\} \quad \text{in } (-H, 0) \times (0, T) \tag{6}$$

with additional appropriate initial values. Here, z denotes the only remaining, vertical spatial coordinate, and H the depth of the water column. The terms $Q^{(l)}$ are the biogeochemical coupling (or *source-minus-sink*) terms for the four tracers and $\mathbf{u} = (u_1, \ldots, u_n)$ is the vector of unknown physical and biological parameters. The sinking term is only apparent in the equation for detritus. In the one-dimensional model no advection term is used, since a reduction to vertical advection would make no sense. Thus, the circulation data (taken from an ocean model) are the turbulent mixing coefficient $\kappa = \kappa(z, t)$ and the temperature $\Theta = \Theta(z, t)$, which goes into the nonlinear coupling terms $Q^{(l)}$ but is omitted in the notation.

4.2. Discretization Scheme and Discretized Model

The continuous model (6) is discretized and solved using an operator splitting method, which for a given a time-step τ reads

$$\underbrace{\left[I - \tau A_j^{\text{diff}}\right]}_{:= B_j^{\text{diff}}} \mathbf{y}_{j+1} = \underbrace{\left[I + \tau A^{\text{sink}}\right]}_{:= B^{\text{sink}}} B_j^Q \circ B_j^Q \circ B_j^Q \circ B_j^Q(\mathbf{y}_j), \quad j = 1, \ldots, M. \tag{7}$$

Recall that by $\mathbf{y}_j$ we denote the discrete solution in time step j given as

$$\mathbf{y}_j = (y_{ji})_{i=1,\ldots,I}, \quad j = 1, \ldots, M. \tag{8}$$

at the discrete spatial points. Since in our case the model output consists of four tracers, I denotes the number of spatial discrete points times 4. If the discrete state $\mathbf{y}_j$ is given in such a way that the four discrete tracer vectors at the time step j are concatenated, the matrices $A_j^{\text{diff}}, A^{\text{sink}}$ in (7) are (4×4)-block-diagonal matrices. They represent the discretization of the diffusion (with second order central differences) and the sinking (discretized by an upstream scheme), respectively.

In every time step $j \to j + 1$, at first the nonlinear coupling operators Q_j (that depend on t_j directly and/or via the temperature field Θ) are computed at every spatial grid point and integrated by four explicit Euler steps, each of which is described by the nonlinear operator

$$B_j^Q(\mathbf{y}_j) := \left[\mathbf{y}_j + \frac{\tau}{4} Q_j(\mathbf{y}_j)\right].$$

Note that, for simplicity, we omitted the additional arguments of the term Q_j in the formulation above. Then, an explicit Euler step with full step-size τ is performed for the sinking term. This step is represented by the matrix B^{sink}. Since the sinking velocity is temporarily constant, this matrix does not depend on the time step j. Finally, an implicit Euler step for the diffusion operator is applied. Due to $\kappa = \kappa(z, t)$ the

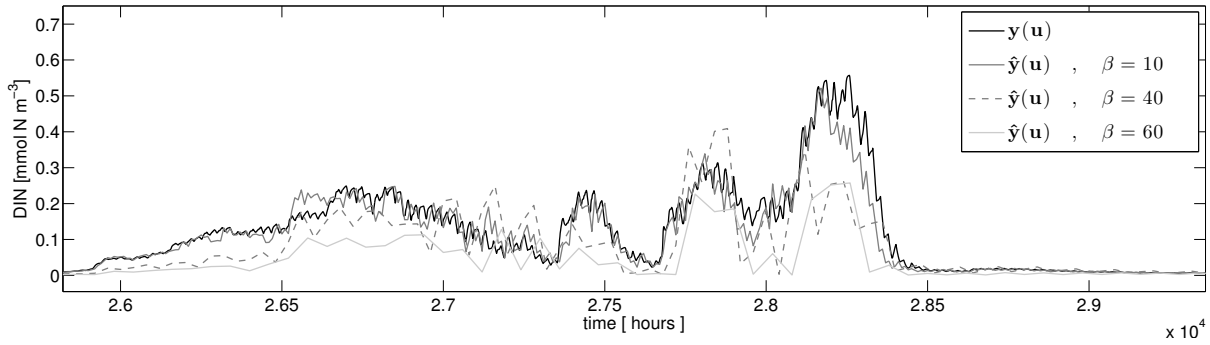

Figure 2: High- and low-fidelity model output $\mathbf{y}, \hat{\mathbf{y}}$, respectively, for the state dissolved inorganic nitrogen at depth $z \approx -2.68\,\mathrm{m}$ for different values of the coarsening factor β and the same randomly chosen parameter vector $\mathbf{u}$. For simplicity we skip super- and subscripts in the legends of all figures.

resulting matrix B_j^{diff} depends on j and is non-symmetric [21, Section 5]. It is tridiagonal, and the system is solved directly by splitting it up into the four blocks. Writing this last step formally as a matrix inversion, formulation (7) corresponds to (2).

In the original discrete model (6) the time step τ is chosen as one hour, and this version is from now on what in surrogate-based optimization is called the high-fidelity or fine model.

5. The Low-Fidelity Model

Surrogates can be either based upon an approximation of the sampled high-fidelity model data (functional surrogates) or on a physical low-fidelity model. Functional surrogates are constructed without any particular knowledge of the system and will not be addressed further in this paper. In contrast, surrogates based upon a physical low-fidelity model (also known as *physically based surrogates* [22]) inherit more characteristics of the fine model under consideration. Possible ways to create such a physical low-fidelity model are by using a coarser discretization (while employing the same simulation tool as for the high-fidelity model), simplified physics or different ways of describing the same physical phenomenon or even by using analytical formulas if available. In this paper, we use a low-fidelity model which has a coarser time discretization which we will explain below.

5.1. Coarser Time Discretization

The low-fidelity model is obtained by using a coarser time discretization with

$$\hat{\tau} = \beta\tau$$

with a *coarsening factor* $\beta \in \mathbb{N} \setminus \{0, 1\}$, while keeping the spatial discretization fixed. The state variable for this coarser discretized model will be denoted by $\hat{\mathbf{y}}$, the corresponding number of discrete time steps by $\hat{M} = M/\beta$. Note that the parameters $\mathbf{u}$ for this coarse model are the same as for the fine model. Figure 2 shows the fine and coarse model output $\mathbf{y}, \hat{\mathbf{y}}$ for the state dissolved inorganic nitrogen, for different values of β and at the same randomly chosen parameter vector $\mathbf{u}$.

It is important to keep in mind that choosing β too large could lead to a numerically unstable scheme [23]. The condition on stability is determined by the ratio h/u_1 and the nonlinear coupling term Q, where h denotes the spatial step-size. All computations in this paper were performed with parameters that guarantee stability.

6. The Surrogate

The surrogate model is constructed here in a simple way using a multiplicative response correction of the low-fidelity model. The correction term is calculated at the beginning of each iteration of the algorithm (4) using a single high-fidelity model evaluation. It turns out that this way of correcting the low-fidelity model is quite suitable for the considered problem because the relation between the low- and high-fidelity model response values is rather well preserved for various sets of parameters $\mathbf{u}$, at least locally.

6.1. Smoothing

As the low-fidelity model output is very noisy (cf. Figure 2), it is necessary to smoothen the coarse and, consequently, also the fine model output before calculating the multiplicative correction factors. Initial experiments indicated (details omitted for the sake of brevity) that the surrogate-based optimization exploiting the unsmoothed model outputs is not able to yield a reasonable solution.

For the smoothening of the fine and coarse model output $\hat{\mathbf{y}}, \mathbf{y}$, respectively, we use a walking average with span $\pm n$ given as:

$$\begin{aligned}
\tilde{\hat{y}}_{ji} &:= \frac{1}{2n+1} \sum_{m=j-n}^{j+n} \left[\frac{1}{2n+1} \sum_{p=m-n}^{m+n} \hat{y}_{pi} \right] \\
\tilde{y}^{\beta}_{ji} &:= \frac{1}{2n+1} \sum_{m=j-n}^{j+n} \left[\frac{1}{2n+1} \sum_{p=m-n}^{m+n} y^{\beta}_{pi} \right] \\
j &= 1, \ldots, \hat{M}, \quad i = 1, \ldots, I,
\end{aligned} \tag{9}$$

where j, i are the temporal and spatial indices, respectively (cf. (8)) and where we used the *down-sampled* fine model output, denoted by $\mathbf{y}^{\beta} \in \mathbb{R}^{\hat{M}I}$, which is given by

$$y^{\beta}_{ji} := y_{\beta j, i}, \quad j = 1, \ldots, \hat{M}, \quad i = 1, \ldots, I, \tag{10}$$

to be commensurable with the coarse model output. In this paper, we use $n = 3$. Also, the smoothing is performed twice. It was observed by visual inspection of the model outputs that this procedure allows us to remove the numerical noise and identify the main characteristics of the traces of interest. It turns out, also by visual inspection, that the chosen value of $n = 3$ and "double" smoothing are suitable for the considered problem. Figures 2 and 3 show the corresponding fine and coarse model outputs without (Figure 2) and with smoothing (Figure 3), with increasing coarsening factor β (cf. Subsection 5.1) for one representative tracer.

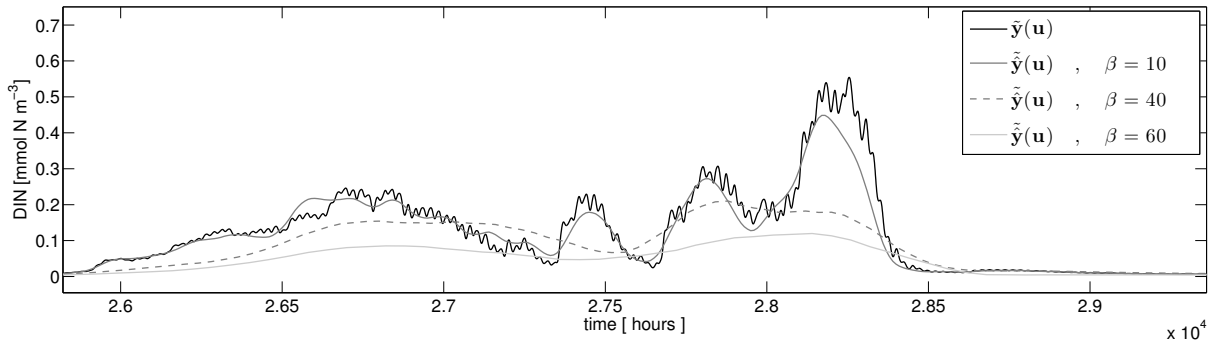

Figure 3: Same as in Figure 2 but now using smoothing (cf. (9)) for both the coarse and the fine model. Smoothing helps removing the numerical noise in the model outputs so that the optimization process is able to identify and track relevant changes of the traces of interest.

6.2. Response Correction

In this work, the surrogate model output is generated, at iteration k of the optimization process, by multiplicative correction of the low-fidelity model output (cf. Section 3). The *correction factor*, denoted as A_{kji}, is defined by pointwise division of the smoothed fine by the smoothed coarse model output at the point $\mathbf{u}_k$, i.e.,

$$\left.\begin{array}{rcl} s_{kji}(\mathbf{u}) & := & A_{kji}\,\tilde{\tilde{y}}_{ji}(\mathbf{u}), \\ A_{kji} & := & \dfrac{\tilde{y}^{\beta}_{ji}(\mathbf{u}_k)}{\tilde{\tilde{y}}_{ji}(\mathbf{u}_k)} \end{array}\right\} \begin{array}{l} k = 1, 2, \ldots, \\ j = 1, \ldots, \hat{M}, \quad i = 1, \ldots, I, \end{array} \tag{11}$$

where $\tilde{\mathbf{y}}^{\beta}$ is given by (9). We call $A_k := (A_{kji})_{j,i} \in \mathbb{R}^{\hat{M}\times I}$ the *correction matrix* in step k. We use it to write the correction step in iteration k on the whole discrete state vector as

$$\mathbf{s}_k(\mathbf{u}) := A_k \circ \tilde{\tilde{\mathbf{y}}}(\mathbf{u}), \quad \mathbf{s}_k \in \mathbb{R}^{\hat{M}I}$$

where the operation "$\circ$" is defined by (11).

Note that the surrogate model is constructed using just one evaluation of the high-fidelity model. This simple correction scheme is justified by the fact that the overall "shape" of the low-fidelity model output resembles that of the high-fidelity one. In particular, the high-value outputs for both models are corresponding to each other on the time scale, which is the consequence of the low-fidelity model being physically-based. Also, the relative changes of the outputs while changing the model parameters are similar for both coarse and fine models so that the multiplicative correction seems to be a natural choice.

It should be emphasized that our surrogate model does not use high-fidelity model sensitivity data. Still, as demonstrated in Section 8, it is able to yield remarkably good results, not only with respect to the quality of the final solution, but, most importantly, in terms of the low computational cost of the optimization process.

6.3. Consistency Conditions and Generalization Capability

By definition, the surrogate model (11) satisfies the 0-order consistency condition (cf. (5)) in the point of alignment $\mathbf{u}_k$, i.e.,

$$\mathbf{s}_k(\mathbf{u}_k) = \mathbf{y}^\beta(\mathbf{u}_k).$$

As we do not use sensitivity information, the 1st-order consistency condition cannot be satisfied exactly. Nevertheless, our surrogate model exhibits quite good generalization capability, which means that the surrogate provides a reasonable approximation of the high-fidelity one in the neighborhood of $\mathbf{u}_k$. This is of primary importance from the point of view of the robustness of the optimization process.

In the following, in order to demonstrate the generalization property of the surrogate model, we analyze the quality of the surrogate during the optimization run.

For this purpose we consider an iterate, say $\mathbf{u}_k$, and $\bar{\mathbf{u}}_k \in B_\epsilon(\mathbf{u}_k)$, i.e., in the ball around $\mathbf{u}_k$ with the radius ϵ which is an estimate for the step size or trust-region radius in this iteration step (cf. Section 3). The surrogate model $\mathbf{s}_k$ is established at $\mathbf{u}_k$ so that we have $\mathbf{s}_k(\mathbf{u}_k) = \mathbf{y}^\beta(\mathbf{u}_k)$. We would like to have at least approximate satisfaction of the 0-order consistency condition at $\bar{\mathbf{u}}_k$, i.e., $\mathbf{s}(\bar{\mathbf{u}}_k) \approx \mathbf{y}^\beta(\bar{\mathbf{u}}_k)$.

We performed this test of the generalization property in every iteration of our surrogate optimization and show here the results for one iteration at the beginning and one towards the end. Since the step size within the optimization usually decreases at the end we chose a smaller ϵ in the second case. Figure 4 shows the smoothed surrogate's, fine and coarse model output at those two iterations.

Obviously, since the surrogate's model output at $\mathbf{u}_k$ exactly reproduces the fine model output at $\mathbf{u}_k$, the surrogate exactly satisfies the 0-order consistency condition (6.3) in the point of alignment. Furthermore, in the neighborhood, the surrogate's model output still provides a very reasonable approximation of the fine one. This is due to the fact that the relation between the low- and high-fidelity model output is rather well preserved for various sets of parameters $\mathbf{u}$ (here $\mathbf{u}_k$ and $\bar{\mathbf{u}}_k$), which, in turn, is a consequence of both the coarse and fine model enjoying the same underlying physics. This is also verified by Figure 4 (upper) comparing the low- and high-fidelity model at these two parameter vectors.

Specifically, this setup demonstrates that towards the end of the optimization run, our surrogate has even better generalization properties, which improves the robustness of the optimization process.

Occasionally, there might occur a situation where the coarse model output is close to zero (and maybe even negative due to approximation errors) and a few magnitudes smaller than the fine one, which leads to large (possibly negative) entries in the corresponding correction tensor A_k. If this is only true in one iterate $\mathbf{u}_k$ but not anymore in the neighborhood, the resulting surrogate might provide a poor approximation there. These issues can be dealt with using some simple means, e.g., by introducing upper and lower bounds for the multiplication factors as well as non-negative bounds for the model outputs (the negative output is non-physical and is a result of numerical errors due to using large time steps in the coarse model).

However, even without addressing these issues, our approach yields good results both in terms of the quality of the final solution and, most importantly, in terms of the relative reduction in the total optimization

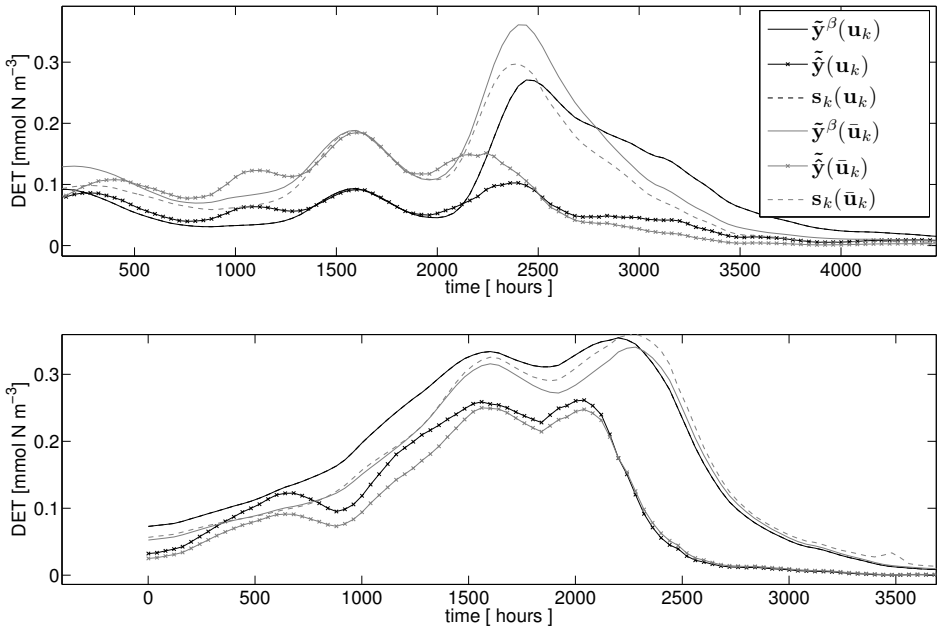

Figure 4: Surrogate's, fine (down-sampled) and coarse model output $\tilde{\mathbf{y}}^{\beta}, \tilde{\hat{\mathbf{y}}}, \mathbf{s}_k$ for the state detritus at depth $z \approx -2.68\,\mathrm{m}$ and at two iterates $\mathbf{u}_k$ and with different neighborhood radii ϵ, see the text for details. The surrogate obviously provides a reasonable approximation of the fine model at the point and in the neighborhood. Shown are the smoothed model outputs and for illustration only for some representative tracers and a part of the whole time interval only.

cost.

7. Optimization Setup

The optimization approach proposed in this work has been tested using synthetic target data. We compare the quality of the solution and the computational cost of the surrogate-based optimization to those obtained by direct fine and coarse model optimization. For all optimizations we used the MATLAB[2] function `fmincon`, exploiting the active-set algorithm.

At a randomly chosen parameter vector $\mathbf{u}_d \in U_{ad}$ we computed the fine model output $\mathbf{y}(\mathbf{u}_d)$ and down-sampled it to be commensurable with the coarse and surrogate model outputs. The resulting data set is used as our synthetic target data $\mathbf{y}_d$ and given as:

$$(y_d)_{ji} := y_{ji}^{\beta}(\mathbf{u}_d), \quad j = 1, \ldots, \hat{M}, \quad i = 1, \ldots, I,$$

where $\mathbf{y}^{\beta}$ was defined in (10).

[2]MATLAB is a registered trademark of The MathWorks, Inc., http://www.mathworks.com

7.1. Cost Functions

Now we define the following cost functions:

$$J(\mathbf{z}) \quad := \quad ||\,\mathbf{z}-\mathbf{y}_d\,||^2 = \sum_{i=1}^{I}\sum_{j=1}^{\hat{M}} (z_{ji}-(y_d)_{ji})^2\,, \tag{12}$$

$$\tilde{J}(\mathbf{z}) \quad := \quad ||\,\mathbf{z}-\tilde{\mathbf{y}}_d\,||^2 = \sum_{i=1}^{I}\sum_{j=1}^{\hat{M}} (z_{ji}-(\tilde{y}_d)_{ji})^2\,, \quad \mathbf{z}\in\mathbb{R}^{\hat{M}I}. \tag{13}$$

Note that, since for the optimization of the coarse model and the surrogate we have to consider the smoothed model output (see Section 6.1), we also have to consider the smoothed target data, yielding $\tilde{\mathbf{y}}_d$.

In order to yield a fair comparison between the results obtained from direct fine model optimization and those obtained from coarse model and surrogate optimization we also consider the sampled fine model output $\mathbf{y}^\beta$ given by (10) for the fine model optimization. We thus have the following three optimization problems:

- fine model optimization:
$$\mathbf{u}^* := \arg\min_{\mathbf{u}\in U_{ad}} J(\,\mathbf{y}^\beta(\mathbf{u})\,)$$
- coarse model optimization:
$$\hat{\mathbf{u}}^* := \arg\min_{\mathbf{u}\in U_{ad}} \tilde{J}(\,\tilde{\mathbf{y}}(\mathbf{u})\,)$$
- surrogate optimization:
$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}\in U_{ad}} \tilde{J}(\,\mathbf{s}_k(\mathbf{u})\,), \quad k=0,1,\ldots$$

We furthermore constitute that using smoothing of the fine model output is not essential. However it actually turned out that even when using smoothing, results of the fine model optimization are not significantly affected.

7.2. Optimization Cost

We will denote the total optimization cost of the surrogate, the fine and of the coarse model optimization by C_s, C_f and C_c, respectively.

This cost is given in terms of the total number of *equivalent* fine model evaluations. For example, β evaluations of the coarse model used here with a coarsening factor β are equivalent to (or, as expensive as) one fine model evaluation. On the other hand, the cost of one iteration of the surrogate-based optimization procedure (in terms of equivalent fine model evaluations) equals to the number of coarse model evaluations necessary to optimize the surrogate model divided by β, and increased by one (since there is only one fine mode evaluation necessary per iteration).

8. Results and Discussion

The operation and performance of the proposed algorithm is illustrated through the results of an exemplary test run with $I = 33\cdot 4, M = 8760\cdot 5$ and $\beta = 40$, which means that we obtain $\hat{M} = M/\beta = 1095$ discrete

time steps for the coarse model.

Below, we consider the following quantities:

(i) The target $\mathbf{y}_d$, i.e., the down-sampled fine model output at a randomly chosen parameter vector $\mathbf{u}_d$,

(ii) the down-sampled fine model output $\mathbf{y}^\beta$ at another randomly chosen parameter vector $\mathbf{u}_0$, serving as initial value of the optimization runs,

(iii) at the result $\hat{\mathbf{u}}^*$ of a coarse model optimization,

(iv) at the result $\mathbf{u}_s^*$ of the surrogate-based optimization and

(v) at the output of a fine model optimization yielding $\mathbf{u}^*$.

For the comparison of the results of the three optimization approaches we use the value of the cost function J given in (12), i.e., the one using the sampled fine model output and unsmoothed target data.

For the results provided in the following paragraph, 13 re-alignments (cf. Section 3) of the surrogate were required to satisfy the termination condition $J(\,\mathbf{y}^\beta(\mathbf{u}_k)\,) \leq 50$. This particular value was selected as it ensures good visual agreement between the fine model output and the target. Furthermore, we used a specific number of iterations within each surrogate optimization run (4), here 7. The reason is that it is not necessary to run the surrogate model optimization until convergence: an approximate solution is sufficient as the surrogate model is not perfectly accurate so that using a fixed (and rather limited) number of function calls allows us to reduce the computational cost of the optimization process. We should also point out that choosing the above termination condition is up to the user and it is generally problem dependent. We refer the reader to Subsection 8.2 for a more thorough discussion.

Altogether, a good agreement between (i) and (v) would indicate a high quality of the algorithm exploited to optimize the model (fine/coarse/surrogate), whereas a good agreement between (iv) and (v) would mean that the surrogate-based optimization works well.

8.1. Numerical Results

Figure 5 illustrates the results of the fine, coarse model and surrogate optimization for the randomly chosen initial parameter vector $\mathbf{u}_0$. Corresponding parameters and values of the cost function J are given in Table 1. Furthermore, the table shows the total optimization cost of the high-fidelity (C_f), the low-fidelity (C_f) and of the surrogate optimization (C_s) as were described in Subsection 7.2.

Note that Figure 5 shows selected (representative) tracers for a part of the whole time interval at some distinct depth layers only. The total number of depth layers considered in the optimization process is 33 and the entire time scale is 43800 so that it is impossible to present a full model output here. We emphasize that the qualitative behavior of the other tracers and at different times and spatial layers is similar.

Figure 5 indicates that the direct fine model optimization yields a very reasonable final solution $\mathbf{y}^\beta(\mathbf{u}^*)$ of the target data $\mathbf{y}_d$. This corresponds to a cost function value of $J(\,\mathbf{y}^\beta(\mathbf{u}^*)\,) = 1.267e-02$ obtained after 983 function evaluations (cf. Table 1), hence leading to the optimization cost of $C_f = 983$. Furthermore, in can be observed that the parameters $\hat{\mathbf{u}}^*$ obtained by coarse model optimization provide only a rough

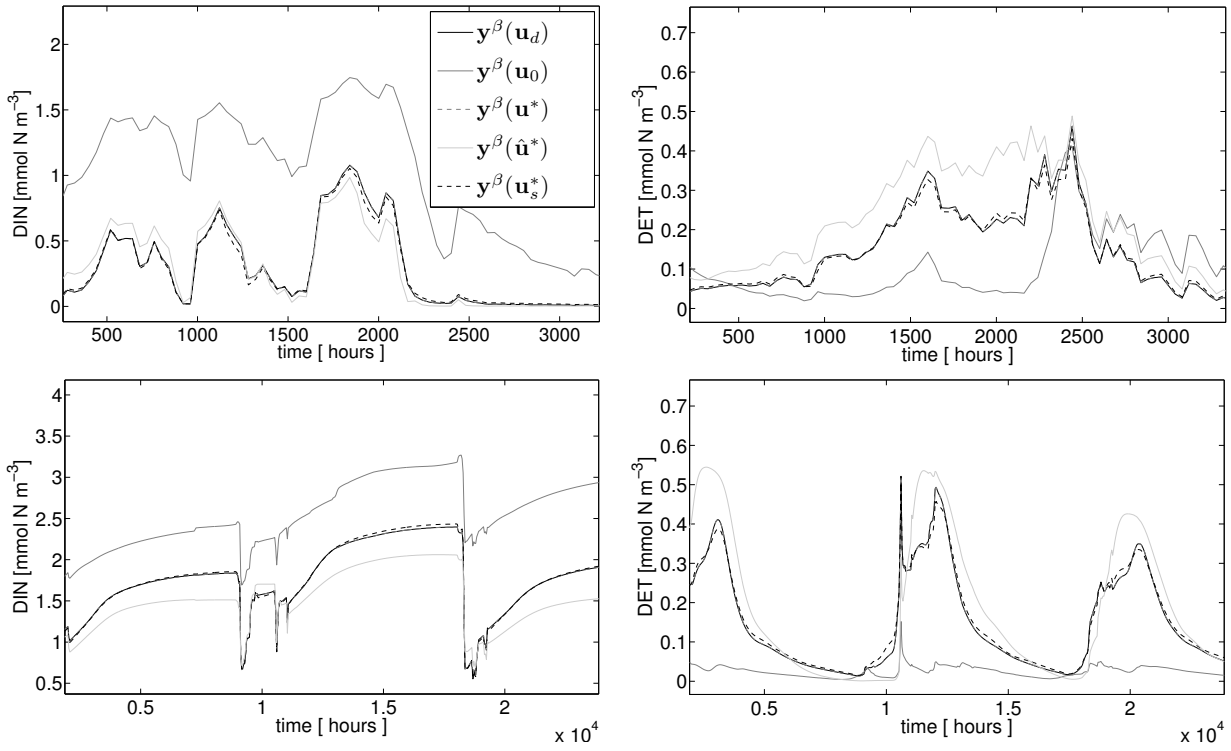

Figure 5: Fine model output $\mathbf{y}^\beta$ (down-sampled) for state dissolved inorganic nitrogen (left) and the state detritus (right) at depth $z \approx -2.68\,\mathrm{m}$ (top) and $z \approx -184.32\,\mathrm{m}$ (bottom). Shown are, in the legend from top to bottom: (i) Target $\mathbf{y}_d$, i.e., the sampled fine model output at a randomly chosen parameter vector $\mathbf{u}_d$, (ii) fine model output at the initial value $\mathbf{u}_0$, (iii) at the result of the direct fine model optimization yielding $\mathbf{u}^*$, (iv) at the coarse model optimum $\hat{\mathbf{u}}^*$ and (v) at the optimum $\mathbf{u}_s^*$ obtained by surrogate optimization. Curves corresponding to (i), (iii) and (v) are very close. For clarity, the sampled fine model output is only shown at the selected (representative) time intervals. In the lower figures, a greater section can be shown since the model output at this deeper depth layer is not as noisy as in upper layers.

approximation $\mathbf{y}^\beta(\hat{\mathbf{u}}^*)$ of the target data corresponding to $J(\,\mathbf{y}^\beta(\hat{\mathbf{u}}^*)\,) = 2.96e+03$. The optimization cost is only $C_c = 11.275$ equivalent fine model evaluations. Optimization of the surrogate finally provides a solution $\mathbf{u}_s^*$ with a remarkably good optimal fit $\mathbf{y}^\beta(\mathbf{u}_s^*)$ and parameter match corresponding to a cost function of $J(\,\mathbf{y}^\beta(\mathbf{u}_s^*)\,) = 48.527$.

The key point is that the computational cost of the surrogate-based optimization is low: only $C_s = 59.575$ equivalent fine model evaluations were required to yield $\mathbf{u}_s^*$. Roughly the same cost function value $J \approx 48$ was obtained by direct fine model optimization after $C_f = 375$ model evaluations. Altogether, a reduction in the total optimization cost of about 84% could be obtained by using this surrogate-based optimization approach.

We point out that the performance looks similar for other initial conditions $\mathbf{u}_0$ as well as for other target data. It is also worth noticing that although using different routines for fine/surrogate model optimization might yield different results, the relative reduction in the total optimization cost using the surrogate in the optimization run would probably be maintained. For example, in [14] better cost function values were

iterate	$u_{k,1}$	$u_{k,2}$	...								$u_{k,12}$	$J(\mathbf{y}^\beta(\mathbf{u}))$	C_i
$\mathbf{u}_0$	0.718	0.314	0.018	0.06	0.026	1.992	0.839	0.001	0.152	0.079	0.661	3.823	6.609e+04	
	Fine model optimization: $\mathbf{u}^* := \text{argmin}_{\mathbf{u}\in U_{ad}} J\left(\mathbf{y}^\beta(\mathbf{u})\right)$													
$\mathbf{u}^*$	0.747	0.596	0.025	0.01	0.03	0.999	2.046	0.01	0.203	0.02	0.493	4.31	1.267e-02	983
	Coarse model optimization: $\hat{\mathbf{u}}^* := \text{argmin}_{\mathbf{u}\in U_{ad}} \tilde{J}\left(\tilde{\mathbf{y}}(\mathbf{u})\right)$													
$\hat{\mathbf{u}}^*$	0.3	1.066	0.036	0.065	0.064	0.025	0.04	0.065	0.01	0.012	0.73	3.449	2.96e+03	11.275
	Surrogate optimization: $\mathbf{u}_s^* := \text{argmin}_{\mathbf{u}\in U_{ad}} \tilde{J}\left(\mathbf{s}_k(\mathbf{u})\right)$													
$\mathbf{u}_s^*$	0.705	0.626	0.044	0.015	0.06	0.937	1.908	0.016	0.147	0.02	0.629	4.237	48.527	59.575
$\mathbf{u}_d$	0.75	0.6	0.025	0.01	0.03	1.0	2.0	0.01	0.205	0.02	0.5	4.32	$\sim 84\%$ reduction	

Table 1: Initial and optimal parameters $\mathbf{u}_0, \mathbf{u}^*, \hat{\mathbf{u}}^*, \mathbf{u}_s^*$, the corresponding values of the cost function J (which we use for comparison, cf. (12)) as well as the computational cost $C_i \in \{C_f, C_c, C_s\}$ (cf. Subsection 7.2) for a fine, coarse model and surrogate optimization run. The cost is given in terms of the total number of equivalent fine model evaluations required to obtain the given cost function value (see the text for details). Corresponding results in terms of the model output are given in Figure 5.

obtained by direct fine model optimization using a different optimization method (other than MATLAB's `fmincon`) for the same problem and the same model.

8.2. Appropriate Choice of Number of Alignment Steps

It should be emphasized again that the surrogate-based optimization method presented in this paper does not use sensitivity information and that the surrogate model satisfies exactly only the 0-order consistency condition with the high-fidelity model (cf. Subsection 6.3). Because of the specific choice of the model alignment method that is tailored to the relationship between the low- and high-fidelity model, our algorithm is able to yield a rapid improvement of the cost function. On the other hand, the algorithm convergence can be quite slow in the vicinity of the optimal solution. Both points are illustrated in the following paragraphs.

Results are presented in Figure 6 showing the value of the cost function J (cf. (12)) calculated at the single iterates of the fine and coarse model optimization runs (Figure 5 and Table 1) and at those of this extended surrogate optimization run. The x-axis represents the number of equivalent fine model evaluations which were required to reach the given value of the cost function. The same figure indicates several points corresponding to the specific values of the reduction in the total optimization cost.

The point showing 84% reduction marks the result $\mathbf{u}_s^*$ which we presented in the previous paragraph corresponding to a value of the cost function $J(\mathbf{y}^\beta(\mathbf{u}_s^*)) \approx 48$ (cf. Figure 5, Table 1).

The figure also shows that approximately 95% reduction could be achieved after only 4 equivalent fine model evaluations corresponding to a termination condition of $J(\mathbf{y}^\beta(\mathbf{u}_k)) \leq 2780$). Of course the quality of the final solution at this point is not as good as the quality of the solution given above in Figure 5 and Table 1, i.e., the one obtained after approximately equivalent 60 fine model evaluations. It is worth noticing that with even more than those 60 model evaluations, no significant reduction in the cost function value J can be

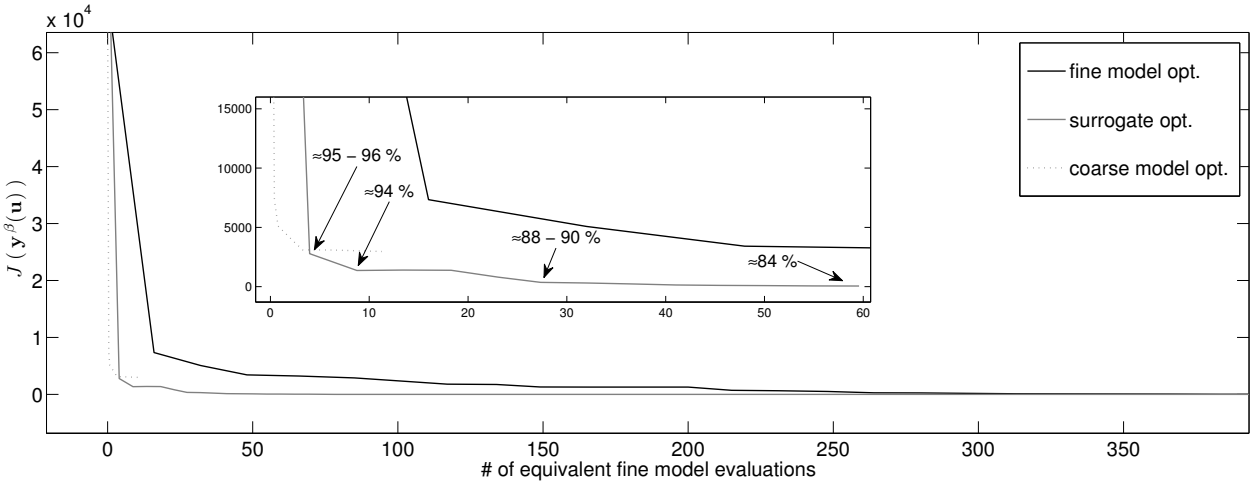

Figure 6: The values of the cost function J (cf. (12)) versus the equivalent number of fine model evaluations for the fine, coarse and the surrogate-based optimization run. Several points corresponding to various values of the relative reduction in the total optimization cost (surrogate-based optimization versus straightforward fine model optimization) are also indicated. Results of fine model and surrogate optimization given in Figure 5 and Table 1 correspond to the point marked as ∼84%.

further achieved by the surrogate optimization process. Decreasing the threshold value in the termination condition to $J(\mathbf{y}^\beta(\mathbf{u}_k)) \leq 0.1$ leads to a significant increase of the number of surrogate optimization steps of approximately 400.

On the other hand, optimization of the coarse model yields a solution $\hat{\mathbf{u}}^*$, which was obtained after approximately 11 equivalent fine model evaluations (cf. Table 1) corresponding to $J(\mathbf{y}^\beta(\hat{\mathbf{u}}^*)) \approx 2960$. This result is much worse than that obtained using surrogate models.

9. Conclusions

Parameter optimization in climate models can be very expensive in terms of the cost function and gradient evaluations, especially for three-dimensional cases. Therefore, methods that aim at reducing the optimization cost, including surrogate-based optimization techniques, are highly desirable.

In this paper, we successfully applied a surrogate optimization technique to the optimization of a one-dimensional coupled marine ecosystem model. We use a physically-based surrogate constructed from a low-fidelity model that is the same as the original, high-fidelity one, but utilizes a coarser time discretization. The surrogate is constructed through a simple multiplicative response correction of the low-fidelity model. We demonstrated that the relation between the low- and high-fidelity model response values is rather well preserved for various sets of parameters, which shows that our correction method is quite suitable for the considered problem.

The optimization approach proposed in this work has been verified using synthetic target data. We furthermore compared the results, both in terms of the quality of the solution and the computational cost,

to those obtained by direct fine and coarse model optimization. Although the direct fine model optimization yields an almost exact fit of the target data, its computational cost is high. On the other hand, the surrogate-based optimization produces a remarkably good results (both in terms of the quality of the final solution and the corresponding parameters) within a very few number of fine model evaluations only, resulting in a significant reduction of the total optimization cost of over 84%.

10. Acknowledgments

The authors would particularly like to thank Andreas Oschlies, IFM Geomar, Kiel and Johannes Rückelt, Institute of Computer Science, Kiel.

References

[1] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization," *Prog. Aerosp. Sci.*, vol. 41, no. 1, pp. 1 – 28, 2005.

[2] A. I. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Prog. Aerosp. Sci.*, vol. 45, no. 1-3, pp. 50 – 79, 2009.

[3] J. W. B, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Søndergaard, "Space mapping: The state of the art," *IEEE T. Microw. Theory.*, vol. 52, no. 1, 2004.

[4] L. Leifsson and S. Koziel, "Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction," *Journal of Computational Science*, vol. 1, no. 2, pp. 98 – 106, 2010.

[5] K. McGuffie and A. Henderson-Sellers, *A Climate Modelling Primer*. Wiley, 3nd ed., 2005.

[6] A. Majda, *Introduction to PDE's and Waves for the Atmosphere and Ocean*. AMS, 2003.

[7] A. E. Gill, *Atmosphere - Ocean Dynamics*, vol. 30 of *International Geophysics Series*. Academic Press, 1982.

[8] D. C. Wilcox, *Turbulence Modeling for CFD*. DCW Industries, 2nd ed., 1998.

[9] J. Sarmiento and N. Gruber, *Ocean Biogeochemical Dynamics*. Princeton University Press, 2006.

[10] W. Fennel and T. Neumann, *Introduction to the Modelling of Marine Ecosystems*. Elsevier, 2004.

[11] H. T. Banks and K. Kunisch, *Estimation Techniques for Distributed Parameter Systems*. Birkhäuser, 1989.

[12] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[13] A. Oschlies and V. Garcon, "An eddy-permitting coupled physical-biological model of the north atlantic. 1. sensitivity to advection numerics and mixed layer physics," *Global Biogeochem. Cy.*, vol. 13, pp. 135–160, 1999.

[14] J. Rückelt, V. Sauerland, T. Slawig, A. Srivastav, B. Ward, and C. Patvardhan, "Parameter optimization and uncertainty analysis in a model of oceanic co2-uptake using a hybrid algorithm and algorithmic differentiation," *Nonlinear Analysis: Real World Applications*, vol. online, 2010.

[15] P. Kunkel and V. Mehrmann, *Differential-algebraic equations: analysis and numerical solution.* EMS, 2006.

[16] R. Temam, *Navier-Stokes Equations.* Amsterdam: North-Holland, 1979.

[17] T. Simpson, J. Poplinski, P. N. Koch, and J. Allen, "Metamodels for computer-based engineering design: Survey and recommendations," *Eng. Comput.*, vol. 17, pp. 129–150, 2001. 10.1007/PL00007198.

[18] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, pp. 199–222, 2004. 10.1023/B:STCO.0000035301.49549.88.

[19] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-region methods.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 2000.

[20] S. Koziel, J. Bandler, and Q. Cheng, "Robust trust-region space-mapping algorithms for microwave design optimization," *IEEE T. Microw. Theory.*, vol. 58, pp. 2166 –2174, Aug. 2010.

[21] W. Hackbusch, *Elliptic Differential Equations: Theory and Numerical Treatment.* Springer Series in Computational Mathematics, Springer Berlin, 2010.

[22] J. Søndergaard, *Optimization using surrogate models - by the space mapping technique.* PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2003. Supervisor: Kaj Madsen.

[23] C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics*, vol. 1. Springer, 2nd ed., 1991.